

**DIGITAL WORKS HAVING USAGE RIGHTS AND
METHOD FOR CREATING THE SAME**

RELATED APPLICATIONS

[0001] This application is a continuation of application Ser. No. 09/778,001, filed February 7, 2001, which is a divisional of Ser. No. 08/967,084, filed on Nov. 10, 1997, now U.S. patent 6,236,971, which is a continuation of Ser. No. 08/344,760, filed Nov. 23, 1994, now abandoned, the disclosure of which are being incorporated by reference.

FIELD OF THE INVENTION

[0002] The present invention relates to the field of distribution and usage rights enforcement for digitally encoded works.

BACKGROUND OF THE INVENTION

[0003] A fundamental issue facing the publishing and information industries as they consider electronic publishing is how to prevent the unauthorized and unaccounted distribution or usage of electronically published materials. Electronically published materials are typically distributed in a digital form and recreated on a computer based system having the capability to recreate the materials. Audio and video recordings, software, books and multimedia works are all being electronically published. Companies in these industries receive royalties for each accounted for delivery of the materials, e.g. the sale of an audio CD at a retail outlet. Any unaccounted distribution of a work results in an unpaid royalty (e.g. copying the audio recording CD to another digital medium.)

[0004] The ease in which electronically published works can be "perfectly" reproduced and distributed is a major concern. The transmission of digital works over networks is commonplace. One such widely used network is the Internet. The Internet is a widespread

NVA208575 1

network facility by which computer users in many universities, corporations and government entities communicate and trade ideas and information. Computer bulletin boards found on the Internet and commercial networks such as CompuServ and Prodigy allow for the posting and retrieving of digital information. Information services such as Dialog and LEXIS/NEXIS provide databases of current information on a wide variety of topics. Another factor which will exacerbate the situation is the development and expansion of the National Information Infrastructure (the NII). It is anticipated that, as the NII grows, the transmission of digital works over networks will increase many times over. It would be desirable to utilize the NII for distribution of digital works without the fear of widespread unauthorized copying.

[0005] The most straightforward way to curb unaccounted distribution is to prevent unauthorized copying and transmission. For existing materials that are distributed in digital form, various safeguards are used. In the case of software, copy protection schemes which limit the number of copies that can be made or which corrupt the output when copying is detected have been employed. Another scheme causes software to become disabled after a predetermined period of time has lapsed. A technique used for workstation based software is to require that a special hardware device must be present on the workstation in order for the software to run, e.g., see U.S. Pat. No. 4,932,054 entitled "Method and Apparatus for Protecting Computer Software Utilizing Coded Filter Network in Conjunction with an Active Coded Hardware Device." Such devices are provided with the software and are commonly referred to as dongles.

[0006] Yet another scheme is to distribute software, but which requires a "key" to enable it's use. This is employed in distribution schemes where "demos" of the software are provided on a medium along with the entire product. The demos can be freely used, but in order to use the actual product, the key must be purchased. These scheme do not hinder copying of the software once the key is initially purchased.

[0007] A system for ensuring that licenses are in place for using licensed products is described in PCT Publication WO 93/01550 to Griswold entitled "License Management System and Method." The licensed product may be any electronically published work but is most effective for use with works that are used for extended periods of time such as software

programs. Griswold requires that the licensed product contain software to invoke a license check monitor at predetermined time intervals. The license check monitor generates request datagrams which identify the licensee. The request datagrams are sent to a license control system over an appropriate communication facility. The license control system then checks the datagram to determine if the datagram is from a valid licensee. The license control system then sends a reply datagram to the license check monitor indicating denial or approval of usage. The license control system will deny access in the event that request datagrams go unanswered after a predetermined period of time (which may indicate an unauthorized attempt to use the licensed product). In this system, access is managed at a central location by the response datagrams. So for example if license fees have not been paid, access to the licensed product is terminated.

[0008] It is argued by Griswold that the described system is advantageous because it can be implemented entirely in software. However, the system described by Griswold has limitations. An important limitation is that during the use of the licensed product, the user must always be coupled to an appropriate communication facility in order to send and receive datagrams. This creates a dependency on the communication facility. So if the communication facility is not available, the licensed product cannot be used. Moreover, some party must absorb the cost of communicating with the license server.

[0009] A system for controlling the distribution of digitally encoded books is embodied in a system available from VPR Systems, LTD. of St. Louis, Mo. The VPR system is self-contained and is comprised of: (1) point of sale kiosks for storing and downloading of books, (2) personal storage mediums (cartridges) to which the books are downloaded, and (3) readers for viewing the book. In a purchase transaction, a purchaser will purchase a voucher card representing the desired book. The voucher will contain sufficient information to identify the book purchased and perhaps some demographic information relating to the sales transaction. To download the book, the voucher and the cartridge are inserted into the kiosk.

[0010] The VPR system may also be used as a library. In such an embodiment, the kiosk manages the number of "copies" that may be checked out at one time. Further, the copy of the book is erased from the users cartridge after a certain check-out time has expired. However,

NVA208575.1

individuals cannot loan books because the cartridges may only be used with the owners reader.

[0011] The foregoing distribution and protection schemes operate in part by preventing subsequent distribution of the work. While this certainly prevents unauthorized distributions, it does so by sacrificing the potential for subsequent revenue bearing uses. For example, it may be desirable to allow the lending of a purchased work to permit exposure of the work to potential buyers. Another example would be to permit the creation of a derivative work for a fee. Yet another example would be to permit copying the work for a fee (essentially purchasing it). Thus, it would be desirable to provide flexibility in the manner that the owner of a digital work may allow it to be distributed or used.

[0012] While flexibility in manner of distribution and use is a concern, the owners of a work want to make sure they are paid for such distributions. In U.S. Pat. No. 4,977,594 to Shear, entitled "Database Usage Metering and Protection System and Method," a system for metering and billing for usage of information distributed on a CD-ROM is described. The system requires the addition of a billing module to the computer system. The billing module may operate in a number of different ways. First, it may periodically communicate billing data to a central billing facility, whereupon the user may be billed. Second, billing may occur by disconnecting the billing module and the user sending it to a central billing facility where the data is read and a user bill generated.

[0013] U.S. Pat. No. 5,247,575, Sprague et al., entitled "Information Distribution System", describes an information distribution system which provides and charges only for user selected information. A plurality of encrypted information packages (IPs) are provided at the user site, via high and/or low density storage media and/or by broadcast transmission. Some of the IPs may be of no interest to the user. The IPs of interest are selected by the user and are decrypted and stored locally. The IPs may be printed, displayed or even copied to other storage medias. The charges for the selected IP's are accumulated within a user apparatus and periodically reported by telephone to a central accounting facility. The central accounting facility also issues keys to decrypt the IPs. The keys are changed periodically. If the central accounting facility has not issued a new key for a particular user station, the station is unable to retrieve

NVA208575 1

information from the system when the key is changed.

[0014] A system available from Wave Systems Corp. of Princeton, N.Y., provides for metering of software usage on a personal computer. The system is installed onto a computer and collects information on what software is in use, encrypts it and then transmits the information to a transaction center. From the transaction center, a bill is generated and sent to the user. The transaction center also maintains customer accounts so that licensing fees may be forwarded directly to the software providers. Software operating under this system must be modified so that usage can be accounted.

[0015] Known techniques distributing digital content do not permit the owner or other authorized party to specify a manner of use of the content and to associate the manner of use with the content in a persistent way.

SUMMARY OF THE INVENTION

[0016] A preferred embodiment of the invention is a digital work adapted to be distributed within a system for controlling at least one of the distribution and use of digital works. The digital work comprises: digital content representing a portion of a digital work suitable for being rendered by a rendering device and usage rights associated with the digital content. The usage rights specify a manner of use indicating one or more stated purposes for which the digital work can be at least one of used and distributed by an authorized party.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] FIG. 1 is a flowchart illustrating a simple instantiation of the operation of the currently preferred embodiment of the present invention.

[0018] FIG. 2 is a block diagram illustrating the various repository types and the repository transaction flow between them in the currently preferred embodiment of the present invention.

[0019] FIG. 3 is a block diagram of a repository coupled with a credit server in the currently preferred embodiment of the present invention.

[0020] FIGS. 4a and 4b are examples of rendering systems as may be utilized in the currently preferred embodiment of the present invention.

[0021] FIG. 5 illustrates a contents file layout for a digital work as may be utilized in the currently preferred embodiment of the present invention.

[0022] FIG. 6 illustrates a contents file layout for an individual digital work of the digital work of FIG. 5 as may be utilized in the currently preferred embodiment of the present invention.

[0023] FIG. 7 illustrates the components of a description block of the currently preferred embodiment of the present invention.

[0024] FIG. 8 illustrates a description tree for the contents file layout of the digital work illustrated in FIG. 5.

[0025] FIG. 9 illustrates a portion of a description tree corresponding to the individual digital work illustrated in FIG. 6.

[0026] FIG. 10 illustrates a layout for the rights portion of a description block as may be utilized in the currently preferred embodiment of the present invention.

[0027] FIG. 11 is a description tree wherein certain d-blocks have PRINT usage rights and is used to illustrate "strict" and "lenient" rules for resolving usage rights conflicts.

[0028] FIG. 12 is a block diagram of the hardware components of a repository as are utilized in the currently preferred embodiment of the present invention.

[0029] FIG. 13 is a block diagram of the functional (logical) components of a repository as are utilized in the currently preferred embodiment of the present invention.

[0030] FIG. 14 is diagram illustrating the basic components of a usage right in the currently preferred embodiment of the present invention.

[0031] FIG. 15 lists the usage rights grammar of the currently preferred embodiment of the present invention.

[0032] FIG. 16 is a flowchart illustrating the steps of certificate delivery, hotlist checking and performance testing as performed in a registration transaction as may be performed in the currently preferred embodiment of the present invention.

[0033] FIG. 17 is a flowchart illustrating the steps of session information exchange and clock synchronization as may be performed in the currently preferred embodiment of the present invention, after each repository in the registration transaction has successfully completed the steps described in FIG. 16.

[0034] FIG. 18 is a flowchart illustrating the basic flow for a usage transaction, including the common opening and closing step, as may be performed in the currently preferred embodiment of the present invention.

[0035] FIG. 19 is a state diagram of server and client repositories in accordance with a transport protocol followed when moving a digital work from the server to the client repositories, as may be performed in the currently preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0036] The preferred embodiment is directed to supporting commercial transactions involving digital works. The transition to digital works profoundly and fundamentally changes how creativity and commerce can work. It changes the cost of transporting or storing works because digital property is almost "massless." Digital property can be transported at electronic speeds and requires almost no warehousing. Keeping an unlimited supply of virtual copies on

NVA208575 1

hand requires essentially no more space than keeping one copy on hand. The digital medium also lowers the costs of alteration, reuse and billing.

[0037] There is a market for digital works because creators are strongly motivated to reuse portions of digital works from others rather than creating their own completely. This is because it is usually so much easier to use an existing stock photo or music clip than to create a new one from scratch.

[0038] Herein the phrase "digital work" refers to any work that has been reduced to a digital representation. This would include any audio, video, text, or multimedia work and any accompanying interpreter (e.g. software) that may be required for recreating or rendering the content of the work. The term composite work refers to a digital work comprised of a collection of other digital works. The term "usage rights" or "rights" is a term which refers to rights granted to a recipient of a digital work. Generally, these rights define the manner in which a digital work can be used and distributed. Each usage right may have one or more specified conditions which must be satisfied before the right may be exercised. A Glossary of the terms used herein is provided at the end of the specification.

[0039] Usage rights are permanently associated with the digital work. Copies made of a digital work will also have the associated usage rights. Thus, the usage rights and any associated fees assigned by a creator and subsequent distributor will always remain with a digital work.

[0040] The enforcement elements of the preferred embodiment are repositories. Among other things, repositories are used to store digital works, control access to digital works, bill for access to digital works and maintain the security and integrity of the system. The combination of associated usage rights and repositories enable distinct advantages over prior systems. As noted in the prior art, payment of fees are primarily for the initial access. In such approaches, once a work has been read, computational control over that copy is gone. Metaphorically, "the content genie is out of the bottle and no more fees can be billed." In contrast, the present invention never separates the fee descriptions from the work. Thus, the digital work genie only moves from one trusted bottle (repository) to another, and all uses of copies are potentially controlled and

billable.

[0041] FIG. 1 is a high level flowchart omitting various details but which demonstrates the basic operation of the present invention. Referring to FIG. 1, a creator creates a digital work, step 101. The creator will then determine appropriate usage rights and fees, associate them with the digital work, and store them in Repository 1, step 102. The determination of appropriate usage rights and fees will depend on various economic factors. The digital work remains securely in Repository 1 until a request for access is received. The request for access begins with a session initiation by another repository. Here a Repository 2 initiates a session with Repository 1, step 103. As will be described in greater detail below, this session initiation includes steps which helps to insure that the respective repositories are trustworthy. Assuming that a session can be established, Repository 2 may then request access to the Digital Work for a stated purpose, i.e., manner of use step 104. The purpose may be, for example, to print the digital work or to obtain a copy of the digital work. The purpose will correspond to a specific usage right. In any event, Repository 1 checks the usage rights associated with the digital work to determine if the access to the digital work may be granted, step 105. The check of the usage rights essentially involves a determination of whether a right associated with the access request has been associated with the digital work and if all conditions associated with the right are satisfied. If the access is denied, repository 1 terminates the session with an error message, step 106. If access is granted, repository 1 transmits the digital work to repository 2, step 107. Once the digital work has been transmitted to repository 2, repository 1 and 2 each generate billing information for the access which is transmitted to a credit server, step 108. Such double billing reporting is done to insure against attempts to circumvent the billing process.

[0042] FIG. 2 illustrates the basic interactions between repository types in the present invention. As will become apparent from FIG. 2, the various repository types will serve different functions. It is fundamental that repositories will share a core set of functionality which will enable secure and trusted communications. Referring to FIG. 2, a repository 201 represents the general instance of a repository. The repository 201 has two modes of operation; a server mode and a requester mode. When in the server mode, the repository will be receiving and processing access requests to digital works. When in the requester mode, the repository will be initiating

NVA208575 1

requests to access digital works. Repository 201 is general in the sense that its primary purpose is as an exchange medium for digital works. During the course of operation, the repository 201 may communicate with a plurality of other repositories, namely authorization repository 202, rendering repository 203 and master repository 204. Communication between repositories occurs utilizing a repository transaction protocol 205.

[0043] Communication with an authorization repository 202 may occur when a digital work being accessed has a condition requiring an authorization. Conceptually, an authorization is a digital certificate such that possession of the certificate is required to gain access to the digital work. An authorization is itself a digital work that can be moved between repositories and subjected to fees and usage rights conditions. An authorization may be required by both repositories involved in an access to a digital work.

[0044] Communication with a rendering repository 203 occurs in connection with the rendering of the content of a digital work. As will be described in greater detail below, a rendering repository is coupled with a rendering device (e.g. a printer device) to comprise a rendering system.

[0045] Communication with a master repository 205 occurs in connection with obtaining an identification certificate. Identification certificates are the means by which a repository is identified as "trustworthy." The use of identification certificates is described below with respect to the registration transaction.

[0046] FIG. 3 illustrates the repository 201 coupled to a credit server 301. The credit server 301 is a device which accumulates billing information for the repository 201. The credit server 301 communicates with repository 201 via billing transactions 302 to record billing transactions. Billing transactions are reported to a billing clearinghouse 303 by the credit server 301 on a periodic basis. The credit server 301 communicates to the billing clearinghouse 303 via clearinghouse transactions 304. The clearinghouse transactions 304 enable a secure and encrypted transmission of information to the billing clearinghouse 303.

[0047] A rendering system is generally defined as a system comprising a repository and a rendering device which can render the content of a digital work into its desired form. Examples of a rendering system may be a computer system, a digital audio system, or a printer. A rendering system has the same security features as a repository. The coupling of a rendering repository with the rendering device may occur in a manner suitable for the type of rendering device.

[0048] FIG. 4a illustrates a printer as an example of a rendering system. Referring to FIG. 4, printer system 401 has contained therein a printer repository 402 and a print device 403. It should be noted that the dashed line defining printer system 401 defines a secure system boundary. Communications within the boundary is assumed to be secure. Depending on the security level, the boundary also represents a barrier intended to provide physical integrity. The printer repository 402 is an instantiation of the rendering repository 205 of FIG. 2. The printer repository 402 will in some instances contain an ephemeral copy of a digital work which remains until it is printed out by the print engine 403. In other instances, the printer repository 402 may contain digital works such as fonts, which will remain and can be billed based on use. This design assures that all communication lines between printers and printing devices are encrypted, unless they are within a physically secure boundary. This design feature eliminates a potential "fault" point through which the digital work could be improperly obtained. The printer device 403 represents the printer components used to create the printed output.

[0049] Also illustrated in FIG. 4a is the repository 404. The repository 404 is coupled to the printer repository 402. The repository 404 represents an external repository which contains digital works.

[0050] FIG. 4b is an example of a computer system as a rendering system. A computer system may constitute a "multi-function" device since it may execute digital works (e.g. software programs) and display digital works (e.g. a digitized photograph). Logically, each rendering device can be viewed as having it's own repository, although only one physical repository is needed. Referring to FIG. 4b, a computer system 410 has contained therein a display/execution repository 411. The display/execution repository 411 is coupled to display device, 412 and

NVA208575 1

execution device 413. The dashed box surrounding the computer system 410 represents a security boundary within which communications are assumed to be secure. The display/execution repository 411 is further coupled to a credit server 414 to report any fees to be billed for access to a digital work and a repository 415 for accessing digital works stored therein.

[0051] Usage rights are associated with digital works. Thus, it is important to understand the structure of a digital work. The structure of a digital work, in particular composite digital works, may be naturally organized into an acyclic structure such as a hierarchy. For example, a magazine has various articles and photographs which may have been created and are owned by different persons. Each of the articles and photographs may represent a node in a hierarchical structure. Consequently, controls, i.e. usage rights, may be placed on each node by the creator. By enabling control and fee billing to be associated with each node, a creator of a work can be assured that the rights and fees are not circumvented.

[0052] In the currently preferred embodiment, the file information for a digital work is divided into two files: a "contents" file and a "description tree" file. From the perspective of a repository, the "contents" file is a stream of addressable bytes whose format depends completely on the interpreter or rendering engine used to play, display or print the digital work. The description tree file makes it possible to examine the rights and fees for a work without reference to the content of the digital work. It should be noted that the term description tree as used herein refers to any type of acyclic structure used to represent the relationship between the various components of a digital work.

[0053] FIG. 5 illustrates the layout of a contents file. Referring to FIG. 5, a digital work 509 is comprised of story A 510, advertisement 511, story B 512 and story C 513. It is assumed that the digital work is stored starting at a relative address of 0. Each of the parts of the digital work are stored linearly so that story A 510 is stored at approximately addresses 0-30,000, advertisement 511 at addresses 30,001-40,000, story B 512 at addresses 40,001-60,000 and story C 513 at addresses 60,001-85 K. The detail of story A 510 is illustrated in FIG. 6. Referring to FIG. 6, the story A 510 is further broken down to show text 614 stored at address 0-1500, soldier photo 615 at addresses 1501-10,000, graphics 616 stored at addresses 10,001-25,000 and sidebar

NVA208575.1

617 stored address 25,001-30,000. Note that the data in the contents file may be compressed (for saving storage) or encrypted (for security).

[0054] From FIGS. 5 and 6 it is readily observed that a digital work can be represented by its component parts as a hierarchy. The description tree for a digital work is comprised of a set of related descriptor blocks (d-blocks). The contents of each d-block is described with respect to FIG. 7. Referring to FIG. 7, a d-block 700 includes an identifier 701 which is a unique identifier for the work in the repository, a starting address 702 providing the start address of the first byte of the work, a length 703 giving the number of bytes in the work, a rights portion 704 wherein the granted usage rights and their status data are maintained, a parent pointer 705 for pointing to a parent d-block and child pointers 706 for pointing to the child d-blocks. In the currently preferred embodiment, the identifier 701 has two parts. The first part is a unique number assigned to the repository upon manufacture. The second part is a unique number assigned to the work upon creation. The rights portion 704 will contain a data structure, such as a look-up table, wherein the various information associated with a right is maintained. The information required by the respective usage rights is described in more detail below. D-blocks form a strict hierarchy. The top d-block of a work has no parent; all other d-blocks have one parent. The relationship of usage rights between parent and child d-blocks and how conflicts are resolved is described below.

[0055] A special type of d-block is a "shell" d-block. A shell d-block adds no new content beyond the content of its parts. A shell d-block is used to add rights and fee information, typically by distributors of digital works.

[0056] FIG. 8 illustrates a description tree for the digital work of FIG. 5. Referring to FIG. 8, a top d-block 820 for the digital work points to the various stories and advertisements contained therein. Here, the top d-block 820 points to d-block 821 (representing story A 510), d-block 822 (representing the advertisement 511), d-block 823 (representing story B 512) and d-block 824 (representing story C 513).

[0057] The portion of the description tree for Story A 510 is illustrated in FIG. 9. D-block 925 represents text 614, d-block 926 represents photo 615, d-block 927 represents graphics

616 by and d-block 928 represents sidebar 617.

[0058] The rights portion 704 of a descriptor block is further illustrated in FIG. 10. FIG. 10 illustrates a structure which is repeated in the rights portion 704 for each right. Referring to FIG. 10, each right will have a right code field 1050 and status information field 1052. The right code field 1050 will contain a unique code assigned to a right. The status information field 1052 will contain information relating to the state of a right and the digital work. Such information is indicated below in Table 1. The rights as stored in the rights portion 304 may typically be in numerical order based on the right code.

TABLE 1

DIGITAL WORK STATE INFORMATION

Property	Value	Use
Copies-in-use	Number	A counter of the number of copies of a work that are in use. Incremented when another copy is used; decremented when use is completed.
Loan-Period	Time-Units	Indicator of the maximum number of time-units that a document can be loaned out
Loaner-Copy	Boolean	Indicator that the current work is a loaned out copy of an authorized digital work.
Remaining-Time	Time-Units	Indicator of the remaining time of use on a metered document right.
Document-Descr	String	A string containing various identifying information about a document. The exact format of this is not specified, but it can include information such as a publisher name, author name, ISBN number, and so on.
Revenue-Owner	RO-Descr	A handle identifying a revenue owner for a digital work. This is used for reporting usage fees.
Publication-Date	Date-Descr	The date that the digital work was published.
History-list	History-Rec	A list of events recording the repositories and dates for operations that copy, transfer, backup, or restore a digital work.

[0059] The approach for representing digital works by separating description data from content assumes that parts of a file are contiguous but takes no position on the actual representation of content. In particular, it is neutral to the question of whether content representation may take an object oriented approach. It would be natural to represent content as

objects. In principle, it may be convenient to have content objects that include the billing structure and rights information that is represented in the d-blocks. Such variations in the design of the representation are possible and are viable alternatives but may introduce processing overhead, e.g. the interpretation of the objects.

[0060] Digital works are stored in a repository as part of a hierarchical file system. Folders (also termed directories and sub-directories) contain the digital works as well as other folders. Digital works and folders in a folder are ordered in alphabetical order. The digital works are typed to reflect how the files are used. Usage rights can be associated with folders so that the folder itself is treated as a digital work. Access to the folder would then be handled in the same fashion as any other digital work. As will be described in more detail below, the contents of the folder are subject to their own rights. Moreover, file management rights may be associated with the folder which define how folder contents can be managed.

[0061] The usage rights are treated as part of the digital work. As the digital work is distributed, the scope of the granted usage rights will remain the same or may be narrowed. For example, when a digital work is transferred from a document server to a repository, the usage rights may include the right to loan a copy for a predetermined period of time (called the original rights). When the repository loans out a copy of the digital work, the usage rights in the loaner copy (called the next set of rights) could be set to prohibit any further rights to loan out the copy. The basic idea is that one cannot grant more rights than they have.

[0062] The association of usage rights with a digital content may occur in a variety of ways. If the usage rights will be the same for an entire digital work, they could be associated when the digital work is processed for deposit in the digital work server. In the case of a digital work having different usage rights for the various components, this can be done as the digital work is being created. An authoring tool or digital work assembling tool could be utilized which provides for an automated process of attaching the usage rights.

[0063] As will be described below, when a digital work is copied, transferred or loaned, a "next set of rights" can be specified. The "next set of rights" will be associated with the

NVA208575 1

digital work as it is transported.

[0064] Because each part of a digital work may have its own usage rights, there will be instances where the rights of a "contained part" are different from its parent or container part. As a result, conflict rules must be established to dictate when and how a right may be exercised. The hierarchical structure of a digital work facilitates the enforcement of such rules. A "strict" rule would be as follows: a right for a part in a digital work is sanctioned if and only if it is sanctioned for the part, for ancestor d-blocks containing the part and for all descendent d-blocks. By sanctioned, it is meant that (1) each of the respective parts must have the right, and (2) any conditions for exercising the right are satisfied.

[0065] It also possible to implement the present invention using a more lenient rule. In the more lenient rule, access to the part may be enabled to the descendent parts which have the right, but access is denied to the descendents which do not.

[0066] Example of applying both the strict rule and lenient is illustrated with reference to FIG. 11. Referring to FIG. 11, a root d-block 1101 has child d-blocks 1102-1105. In this case, root d-block represents a magazine, and each of the child d-blocks 1102-1105 represent articles in the magazine. Suppose that a request is made to PRINT the digital work represented by root d-block 1101 wherein the strict rule is followed. The rights for the root d-block 1101 and child d-blocks 1102-1105 are then examined. Root d-block 1101 and child d-blocks 1102 and 1105 have been granted PRINT rights. Child d-block 1103 has not been granted PRINT rights and child d-block 1104 has PRINT rights conditioned on payment of a usage fee.

[0067] Under the strict rule the PRINT right cannot be exercised because the child d-block does not have the PRINT right. Under the lenient rule, the result would be different. The digital works represented by child d-blocks 1102 and 1105 could be printed and the digital work represented by d-block 1104 could be printed so long as the usage fee is paid. Only the digital work represented by d-block 1103 could not be printed. This same result would be accomplished under the strict rule if the requests were directed to each of the individual digital works.

[0068] The present invention supports various combinations of allowing and

disallowing access. Moreover, as will be described below, the usage rights grammar permits the owner of a digital work to specify if constraints may be imposed on the work by a container part. The manner in which digital works may be sanctioned because of usage rights conflicts would be implementation specific and would depend on the nature of the digital works.

[0069] Many of the powerful functions of repositories--such as their ability to "loan" digital works or automatically handle the commercial reuse of digital works--are possible because they are trusted systems. The systems are trusted because they are able to take responsibility for fairly and reliably carrying out the commercial transactions. That the systems can be responsible ("able to respond") is fundamentally an issue of integrity. The integrity of repositories has three parts: physical integrity, communications integrity, and behavioral integrity.

[0070] Physical integrity refers to the integrity of the physical devices themselves. Physical integrity applies both to the repositories and to the protected digital works. Thus, the higher security classes of repositories themselves may have sensors that detect when tampering is attempted on their secure cases. In addition to protection of the repository itself, the repository design protects access to the content of digital works. In contrast with the design of conventional magnetic and optical devices--such as floppy disks, CD-ROMs, and videotapes--repositories never allow non-trusted systems to access the works directly. A maker of generic computer systems cannot guarantee that their platform will not be used to make unauthorized copies. The manufacturer provides generic capabilities for reading and writing information, and the general nature of the functionality of the general computing device depends on it. Thus, a copy program can copy arbitrary data. This copying issue is not limited to general purpose computers. It also arises for the unauthorized duplication of entertainment "software" such as video and audio recordings by magnetic recorders. Again, the functionality of the recorders depends on their ability to copy and they have no means to check whether a copy is authorized. In contrast, repositories prevent access to the raw data by general devices and can test explicit rights and conditions before copying or otherwise granting access. Information is only accessed by protocol between trusted repositories.

[0071] Communications integrity refers to the integrity of the communications

NVA208575 1

channels between repositories. Roughly speaking, communications integrity means that repositories cannot be easily fooled by "telling them lies." Integrity in this case refers to the property that repositories will only communicate with other devices that are able to present proof that they are certified repositories, and furthermore, that the repositories monitor the communications to detect "impostors" and malicious or accidental interference. Thus the security measures involving encryption, exchange of digital certificates, and nonces described below are all security measures aimed at reliable communication in a world known to contain active adversaries.

[0072] Behavioral integrity refers to the integrity in what repositories do. What repositories do is determined by the software that they execute. The integrity of the software is generally assured only by knowledge of its source. Restated, a user will trust software purchased at a reputable computer store but not trust software obtained off a random (insecure) server on a network. Behavioral integrity is maintained by requiring that repository software be certified and be distributed with proof of such certification, i.e. a digital certificate. The purpose of the certificate is to authenticate that the software has been tested by an authorized organization, which attests that the software does what it is supposed to do and that it does not compromise the behavioral integrity of a repository. If the digital certificate cannot be found in the digital work or the master repository which generated the certificate is not known to the repository receiving the software, then the software cannot be installed.

[0073] In the description of FIG. 2, it was indicated that repositories come in various forms. All repositories provide a core set of services for the transmission of digital works. The manner in which digital works are exchanged is the basis for all transaction between repositories. The various repository types differ in the ultimate functions that they perform. Repositories may be devices themselves, or they may be incorporated into other systems. An example is the rendering repository 205 of FIG. 2.

[0074] A repository will have associated with it a repository identifier. Typically, the repository identifier would be a unique number assigned to the repository at the time of manufacture. Each repository will also be classified as being in a particular security class.

NVA208575 1

Certain communications and transactions may be conditioned on a repository being in a particular security class. The various security classes are described in greater detail below.

[0075] As a prerequisite to operation, a repository will require possession of an identification certificate. Identification certificates are encrypted to prevent forgery and are issued by a Master repository. A master repository plays the role of an authorization agent to enable repositories to receive digital works. Identification certificates must be updated on a periodic basis. Identification certificates are described in greater detail below with respect to the registration transaction.

[0076] A repository has both a hardware and functional embodiment. The functional embodiment is typically software executing on the hardware embodiment. Alternatively, the functional embodiment may be embedded in the hardware embodiment such as an Application Specific Integrated Circuit (ASIC) chip.

[0077] The hardware embodiment of a repository will be enclosed in a secure housing which if compromised, may cause the repository to be disabled. The basic components of the hardware embodiment of a repository are described with reference to FIG. 12. Referring to FIG. 12, a repository is comprised of a processing means 1200, storage system 1207, clock 1205 and external interface 1206. The processing means 1200 is comprised of a processor element 1201 and processor memory 1202. The processing means 1201 provides controller, repository transaction and usage rights transaction functions for the repository. Various functions in the operation of the repository such as decryption and/or decompression of digital works and transaction messages are also performed by the processing means 1200. The processor element 1201 may be a microprocessor or other suitable computing component. The processor memory 1202 would typically be further comprised of Read Only Memories (ROM) and Random Access Memories (RAM). Such memories would contain the software instructions utilized by the processor element 1201 in performing the functions of the repository.

[0078] The storage system 1207 is further comprised of descriptor storage 1203 and content storage 1204. The description tree storage 1203 will store the description tree for the

digital work and the content storage will store the associated content. The description tree storage 1203 and content storage 1204 need not be of the same type of storage medium, nor are they necessarily on the same physical device. So for example, the descriptor storage 1203 may be stored on a solid state storage (for rapid retrieval of the description tree information), while the content storage 1204 may be on a high capacity storage such as an optical disk.

[0079] The clock 1205 is used to time-stamp various time based conditions for usage rights or for metering usage fees which may be associated with the digital works. The clock 1205 will have an uninterruptable power supply, e.g. a battery, in order to maintain the integrity of the time-stamps. The external interface means 1206 provides for the signal connection to other repositories and to a credit server. The external interface means 1206 provides for the exchange of signals via such standard interfaces such as RS-232 or Personal Computer Manufacturers Card Industry Association (PCMCIA) standards, or FDDI. The external interface means 1206 may also provide network connectivity.

[0080] The functional embodiment of a repository is described with reference to FIG. 13. Referring to FIG. 13, the functional embodiment is comprised of an operating system 1301, core repository services 1302, usage transaction handlers 1303, repository specific functions, 1304 and a user interface 1305. The operating system 1301 is specific to the repository and would typically depend on the type of processor being used. The operating system 1301 would also provide the basic services for controlling and interfacing between the basic components of the repository.

[0081] The core repository services 1302 comprise a set of functions required by each and every repository. The core repository services 1302 include the session initiation transactions which are defined in greater detail below. This set of services also includes a generic ticket agent which is used to "punch" a digital ticket and a generic authorization server for processing authorization specifications. Digital tickets and authorizations are specific mechanisms for controlling the distribution and use of digital works and are described in more detail below. Note that coupled to the core repository services are a plurality of identification certificates 1306. The identification certificates 1306 are required to enable the use of the

NVA208575.1

repository.

[0082] The usage transactions handler 1303 comprise functionality for processing access requests to digital works and for billing fees based on access. The usage transactions supported will be different for each repository type. For example, it may not be necessary for some repositories to handle access requests for digital works.

[0083] The repository specific functionality 1304 comprises functionality that is unique to a repository. For example, the master repository has special functionality for issuing digital certificates and maintaining encryption keys. The repository specific functionality 1304 would include the user interface implementation for the repository.

[0084] For some digital works the losses caused by any individual instance of unauthorized copying is insignificant and the chief economic concern lies in assuring the convenience of access and low-overhead billing. In such cases, simple and inexpensive handheld repositories and network-based workstations may be suitable repositories, even though the measures and guarantees of security are modest.

[0085] At the other extreme, some digital works such as a digital copy of a first run movie or a bearer bond or stock certificate would be of very high value so that it is prudent to employ caution and fairly elaborate security measures to ensure that they are not copied or forged. A repository suitable for holding such a digital work could have elaborate measures for ensuring physical integrity and for verifying authorization before use.

[0086] By arranging a universal protocol, all kinds of repositories can communicate with each other in principle. However, creators of some works will want to specify that their works will only be transferred to repositories whose level of security is high enough. For this reason, document repositories have a ranking system for classes and levels of security. The security classes in the currently preferred embodiment are described in Table 2.

TABLE 2

NVA208575.1

REPOSITORY SECURITY LEVELS

Level	Description of Security
0 Open system.	Document transmission is unencrypted. No digital certificate is required for identification. The security of the system depends mostly on user honesty, since only modest knowledge may be needed to circumvent the security measures. The repository has no provisions for preventing unauthorized programs from running and accessing or copying files. The system does not prevent the use of removable storage and does not encrypt stored files.
1 Minimal security.	Like the previous class except that stored files are minimally encrypted, including ones on removable storage.
2 Basic security.	Like the previous class except that special tools and knowledge are required to compromise the programming, the contents of the repository, or the state of the clock. All digital communications are encrypted. A digital certificate is provided as identification. Medium level encryption is used. Repository identification number is unforgeable.
3 General security.	Like the previous class plus the requirement of special tools are needed to compromise the physical integrity of the repository and that modest encryption is used on all transmissions. Password protection is required to use the local user interface. The digital clock system cannot be reset without authorization. No works would be stored on removable storage. When executing works as programs, it runs them in their own address space and does not give them direct access to any file storage or other memory containing system code or works. They can access works only through the transmission transaction protocol.
4 High Security	Like the previous class except that high level encryption is used on all communications. Sensors are used to record attempts at physical and electronic tampering. After such tampering, the repository will not perform other transactions until it has reported such tampering to a designated server.
5 Tamper Resistant	Like the previous class except that if the physical or digital attempts at tampering exceed some preset thresholds that threaten the physical integrity of the repository or the integrity of digital and cryptographic barriers, then the repository will save only document description records of history but will erase or destroy any digital identifiers that could be misused if released to an unscrupulous party. It also modifies any certificates of authenticity to indicate that the physical system has been compromised. It also erases the contents of designated documents.
6	Like the previous class except that the repository will attempt wireless communication to report tampering and will employ noisy alarms.

10	This would correspond to a very high level of security. This server would maintain constant communications to remote security systems reporting transactions, sensor readings, and attempts to circumvent security.

[0087] The characterization of security levels described in Table 2 is not intended to be fixed. More important is the idea of having different security levels for different repositories. It is anticipated that new security classes and requirements will evolve according to social situations and changes in technology.

[0088] A user interface is broadly defined as the mechanism by which a user interacts with a repository in order to invoke transactions to gain access to a digital work, or exercise usage rights. As described above, a repository may be embodied in various forms. The user interface for a repository will differ depending on the particular embodiment. The user interface may be a graphical user interface having icons representing the digital works and the various transactions that may be performed. The user interface may be a generated dialog in which a user is prompted for information.

[0089] The user interface itself need not be part of the repository. As a repository may be embedded in some other device, the user interface may merely be a part of the device in which the repository is embedded. For example, the repository could be embedded in a "card" that is inserted into an available slot in a computer system. The user interface may be combination of a display, keyboard, cursor control device and software executing on the computer system.

[0090] At a minimum, the user interface must permit a user to input information such as access requests and alpha numeric data and provide feedback as to transaction status. The user interface will then cause the repository to initiate the suitable transactions to service the request. Other facets of a particular user interface will depend on the functionality that a repository will provide.

In the preferred embodiment, fees may be associated with the exercise of a right. The requirement for payment of fees is described with each version of a usage right in the usage rights language. The recording and reporting of such fees is performed by the credit server. One of the capabilities enabled by associating fees with rights is the possibility of supporting a wide

range of charging models. The simplest model, used by conventional software, is that there is a single fee at the time of purchase, after which the purchaser obtains unlimited rights to use the work as often and for as long as he or she wants. Alternative models, include metered use and variable fees. A single work can have different fees for different uses. For example, viewing a photograph on a display could have different fees than making a hardcopy or including it in a newly created work. A key to these alternative charging models is to have a low overhead means of establishing fees and accounting for credit on these transactions.

[0091] A credit server is a computational system that reliably authorizes and records these transactions so that fees are billed and paid. The credit server reports fees to a billing clearinghouse. The billing clearinghouse manages the financial transactions as they occur. As a result, bills may be generated and accounts reconciled. Preferably, the credit server would store the fee transactions and periodically communicate via a network with billing clearinghouse for reconciliation. In such an embodiment, communications with the billing clearinghouse would be encrypted for integrity and security reasons. In another embodiment, the credit server acts as a "debit card" where transactions occur in "real-time" against a user account.

[0092] A credit server is comprised of memory, a processing means, a clock, and interface means for coupling to a repository and a financial institution (e.g. a modem). The credit server will also need to have security and authentication functionality. These elements are essentially the same elements as those of a repository. Thus, a single device can be both a repository and a credit server, provided that it has the appropriate processing elements for carrying out the corresponding functions and protocols. Typically, however, a credit server would be a card-sized system in the possession of the owner of the credit. The credit server is coupled to a repository and would interact via financial transactions as described below. Interactions with a financial institution may occur via protocols established by the financial institutions themselves.

[0093] In the currently preferred embodiment credit servers associated with both the server and the repository report the financial transaction to the billing clearinghouse. For example, when a digital work is copied by one repository to another for a fee, credit servers

NVA208575 1

coupled to each of the repositories will report the transaction to the billing clearinghouse. This is desirable in that it insures that a transaction will be accounted for in the event of some break in the communication between a credit server and the billing clearinghouse. However, some implementations may embody only a single credit server reporting the transaction to minimize transaction processing at the risk of losing some transactions.

[0094] The preferred embodiment uses statements in a high level "usage rights language" to define rights associated with digital works and their parts. Usage rights statements are interpreted by repositories and are used to determine what transactions, i.e., means of use, can be successfully carried out for a digital work and also to determine parameters for those transactions. For example, sentences in the language determine whether a given digital work can be copied, when and how it can be used, and what fees (if any) are to be charged for that use. Once the usage rights statements are generated, they are encoded in a suitable form for accessing during the processing of transactions.

[0095] Defining usage rights in terms of a language in combination with the hierarchical representation of a digital work enables the support of a wide variety of distribution and fee schemes. An example is the ability to associate multiple versions of a right with a work. So a creator may associate PRINT right to make 5 copies for \$10.00 and a PRINT right to make unlimited copies for \$100.00. A purchaser may then choose which option best fits his needs. Another example is that rights and fees are additive. So in the case of a composite work, the rights and fees of each of the components works is used in determining the rights and fees for the work as a whole. Other features and benefits of the usage rights language will become apparent in the description of distribution and use scenarios provided below.

[0096] The basic contents of a right are illustrated in FIG. 14. Referring to FIG. 14, a right 1450 has a transactional component 1451 and a specifications component 1452. A right 1450 has a label (e.g. COPY or PRINT) which indicate the manners of use that are embodied by the right. The transactional component 1451 corresponds to a particular way in which a digital work may be used or distributed. The transactional component 1451 is typically embodied in software instructions in a repository which implement the manners of use for the right. The

NVA208575.1

specifications components 1452 are used to specify conditions which must be satisfied prior to the right being exercised or to designate various transaction related parameters. In the currently preferred embodiment, these specifications include copy count 1453, Fees and Incentives 1454, Time 1455, Access and Security 1456 and Control 1457. Each of these specifications will be described in greater detail below with respect to the language grammar elements.

[0097] The usage rights language is based on the grammar described below. A grammar is a convenient means for defining valid sequence of symbols for a language. In describing the grammar the notation "[a.vertline.b.vertline.c]" is used to indicate distinct choices among alternatives. In this example, a sentence can have either an "a", "b" or "c". It must include exactly one of them. The braces { } are used to indicate optional items. Note that brackets, bars and braces are used to describe the language of usage rights sentences but do not appear in actual sentences in the language.

[0098] In contrast, parentheses are part of the usage rights language. Parentheses are used to group items together in lists. The notation (x*) is used to indicate a variable length list, that is, a list containing one or more items of type x. The notation (x)* is used to indicate a variable number of lists containing x.

[0099] Keywords in the grammar are words followed by colons. Keywords are a common and very special case in the language. They are often used to indicate a single value, typically an identifier. In many cases, the keyword and the parameter are entirely optional. When a keyword is given, it often takes a single identifier as its value. In some cases, the keyword takes a list of identifiers. In the usage rights language, time is specified in an hours:minutes:seconds (or hh:mm:ss) representation. Time zone indicators, e.g. PDT for Pacific Daylight Time, may also be specified. Dates are represented as year/month/day (or YYYY/MMM/DD). Note that these time and date representations may specify moments in time or units of time Money units are specified in terms of dollars.

Finally, in the usage rights language, various "things" will need to interact with each other. For example, an instance of a usage right may specify a bank account, a digital ticket, etc. Such things need to be identified and are specified herein using the suffix "-ID."

NVA208575.1

[0100] The Usage Rights Grammar of the preferred embodiment is listed in its entirety in FIG. 15 and is described below.

[0101] Grammar element 1501 "Digital Work Rights:=(Rights*)" define the digital work rights as a set of rights. The set of rights associated with a digital work define how that digital work may be transferred, used, performed or played. A set of rights will attach to the entire digital work and in the case of compound digital works, each of the components of the digital work. The usage rights of components of a digital may be different.

[0102] Grammar element 1502 "Right:=(Right-Code {Copy-Count} {Control-Spec} {Time-Spec} {Access-Spec} {Fee-Spec})" enumerates the content of a right. Each usage right must specify a right code. Each right may also optionally specify conditions which must be satisfied before the right can be exercised. These conditions are copy count, control, time, access and fee conditions. In the currently preferred embodiment, for the optional elements, the following defaults apply: copy count equals 1, no time limit on the use of the right, no access tests or a security level required to use the right and no fee is required. These conditions will each be described in greater detail below.

[0103] It is important to note that a digital work may have multiple versions of a right, each having the same right code. The multiple version would provide alternative conditions and fees for accessing the digital work.

[0104] Grammar element 1503 "Right-Code: =Render-Code.vertline.Transport-Code.vertline.File-Management-Code.vertline.Derivative-Works-Code Configuration-Code" distinguishes each of the specific rights into a particular right type (although each right is identified by distinct right codes). In this way, the grammar provides a catalog of possible rights that can be associated with parts of digital works. In the following, rights are divided into categories for convenience in describing them.

[0105] Grammar element 1504 "Render-Code:=[Play: {Player: Player-ID}.vertline.Print:

NVA208575 1

{Printer: Printer-ID}]" lists a category of rights all involving the making of ephemeral, transitory, or non-digital copies of the digital work. After use the copies are erased.

[0106] Play A process of rendering or performing a digital work on some processor. This includes such things as playing digital movies, playing digital music, playing a video game, running a computer program, or displaying a document on a display.

[0107] Print To render the work in a medium that is not further protected by usage rights, such as printing on paper.

[0108] Grammar element 1505 "Transport-Code:=[Copy.vertline.Transfer.vertline.Loan {Remaining-Rights: Next-Set-of-Rights}]{(Next-Copy-Rights: Next-Set of Rights)}" lists a category of rights involving the making of persistent, usable copies of the digital work on other repositories. The optional Next-Copy-Rights determine the rights on the work after it is transported. If this is not specified, then the rights on the transported copy are the same as on the original. The optional Remaining-Rights specify the rights that remain with a digital work when it is loaned out. If this is not specified, then the default is that no rights can be exercised when it is loaned out.

[0109] Copy Make a new copy of a work

[0110] Transfer Moving a work from one repository to another.

[0111] Loan Temporarily loaning a copy to another repository for a specified period of time.

[0112] Grammar element 1506 "File-Management-Code:=Backup {Back-Up-Copy-Rights: Next-Set-of Rights}.vertline.Restore.vertline.Delete.vertline.Folder.vertline.Director y {Name:Hide-Local.vertline.Hide-Remote} {Parts: Hide-Local.vertline.Hide-Remote}" lists a category of rights involving operations for file management, such as the making of backup copies to protect the copy owner against catastrophic equipment failure.

NVA208575.1

[0113] Many software licenses and also copyright law give a copy owner the right to make backup copies to protect against catastrophic failure of equipment. However, the making of uncontrolled backup copies is inherently at odds with the ability to control usage, since an uncontrolled backup copy can be kept and then restored even after the authorized copy was sold.

[0114] The File management rights enable the making and restoring of backup copies in a way that respects usage rights, honoring the requirements of both the copy owner and the rights grantor and revenue owner. Backup copies of work descriptions (including usage rights and fee data) can be sent under appropriate protocol and usage rights control to other document repositories of sufficiently high security. Further rights permit organization of digital works into folders which themselves are treated as digital works and whose contents may be "hidden" from a party seeking to determine the contents of a repository.

[0115] Backup To make a backup copy of a digital work as protection against media failure.

[0116] Restore To restore a backup copy of a digital work.

[0117] Delete To delete or erase a copy of a digital work.

[0118] Folder To create and name folders, and to move files and folders between folders.

[0119] Directory To hide a folder or it's contents.

[0120] Grammar element 1507 "Derivative-Works-Code:
[Extract.vertline.Embed.vertline.Edit {Process: Process-ID}] {Next-Copy-Rights: Next-Set-of Rights}" lists a category of rights involving the use of a digital work to create new works.

[0121] Extract To remove a portion of a work, for the purposes of creating a new work.

[0122] Embed To include a work in an existing work.

[0123] Edit To alter a digital work by copying, selecting and modifying portions of an existing digital work.

[0124] Grammar element 1508 "Configuration-Code:=Install.vertline.Uninstall" lists a category of rights for installing and uninstalling software on a repository (typically a rendering repository.) This would typically occur for the installation of a new type of player within the rendering repository.

[0125] Install: To install new software on a repository.

[0126] Uninstall: To remove existing software from a repository.

[0127] Grammar element 1509 "Next-Set-of-Rights:={{(Add: Set-Of-Rights)} {(Delete: Set-Of-Rights)} {(Replace: Set-Of-Rights)} {(Keep: Set-Of-Rights)}}" defines how rights are carried forward for a copy of a digital work. If the Next-Copy-Rights is not specified, the rights for the next copy are the same as those of the current copy. Otherwise, the set of rights for the next copy can be specified. Versions of rights after Add: are added to the current set of rights. Rights after Delete: are deleted from the current set of rights. If only right codes are listed after Delete:, then all versions of rights with those codes are deleted. Versions of rights after Replace: subsume all versions of rights of the same type in the current set of rights.

[0128] If Remaining-Rights is not specified, then there are no rights for the original after all Loan copies are loaned out. If Remaining-Rights is specified, then the Keep: token can be used to simplify the expression of what rights to keep behind. A list of right codes following keep means that all of the versions of those listed rights are kept in the remaining copy. This specification can be overridden by subsequent Delete: or Replace: specifications.

[0129] Copy Count Specification. For various transactions, it may be desirable to provide some limit as to the number of "copies" of the work which may be exercised

NVA208575.1

simultaneously for the right. For example, it may be desirable to limit the number of copies of a digital work that may be loaned out at a time or viewed at a time.

[0130] Grammar element 1510 "Copy-Count:=(Copies: positive-integer.vertline.0.vertline.unlimited)" provides a condition which defines the number of "copies" of a work subject to the right. A copy count can be 0, a fixed number, or unlimited. The copy-count is associated with each right, as opposed to there being just a single copy-count for the digital work. The Copy-Count for a right is decremented each time that a right is exercised. When the Copy-Count equals zero, the right can no longer be exercised. If the Copy-Count is not specified, the default is one.

[0131] Rights and fees depend in general on rights granted by the creator as well as further restrictions imposed by later distributors. Control specifications deal with interactions between the creators and their distributors governing the imposition of further restrictions and fees. For example, a distributor of a digital work may not want an end consumer of a digital work to add fees or otherwise profit by commercially exploiting the purchased digital work.

[0132] Grammar element 1511 "Control-Spec:=(Control: {Restrictable.vertline.Unrestrictable} {Unchargeable.vertline.Chargeable})" provides a condition to specify the effect of usage rights and fees of parents on the exercise of the right. A digital work is restrictable if higher level d-blocks can impose further restrictions (time specifications and access specifications) on the right. It is unrestrictable if no further restrictions can be imposed. The default setting is restrictable. A right is unchargeable if no more fees can be imposed on the use of the right. It is chargeable if more fees can be imposed. The default is chargeable.

[0133] It is often desirable to assign a start date or specify some duration as to when a right may be exercised. Grammar element 1512 "Time-Spec: =({Fixed-Interval.vertline.Sliding-Interval.vertline.Meter-Time} Until: Expiration-Date)" provides for specification of time conditions on the exercise of a right. Rights may be granted for a specified time. Different kinds of time specifications are appropriate for different kinds of rights. Some rights may be exercised

NVA208575.1

during a fixed and predetermined duration. Some rights may be exercised for an interval that starts the first time that the right is invoked by some transaction. Some rights may be exercised or are charged according to some kind of metered time, which may be split into separate intervals. For example, a right to view a picture for an hour might be split into six ten minute viewings or four fifteen minute viewings or twenty three minute viewings.

[0134] The terms "time" and "date" are used synonymously to refer to a moment in time. There are several kinds of time specifications. Each specification represents some limitation on the times over which the usage right applies. The Expiration-Date specifies the moment at which the usage right ends. For example, if the Expiration-Date is "Jan. 1, 1995," then the right ends at the first moment of 1995. If the Expiration-Date is specified as *forever*, then the rights are interpreted as continuing without end. If only an expiration date is given, then the right can be exercised as often as desired until the expiration date.

[0135] Grammar element 1513 "Fixed-Interval:=From: Start-Time" is used to define a predetermined interval that runs from the start time to the expiration date.

[0136] Grammar element 1514 "Sliding-Interval:=Interval: Use-Duration" is used to define an indeterminate (or "open") start time. It sets limits on a continuous period of time over which the contents are accessible. The period starts on the first access and ends after the duration has passed or the expiration date is reached, whichever comes first. For example, if the right gives 10 hours of continuous access, the use-duration would begin when the first access was made and end 10 hours later.

[0137] Grammar element 1515 "Meter-Time:=Time-Remaining: Remaining-Use" is used to define a "meter time," that is, a measure of the time that the right is actually exercised. It differs from the Sliding-Interval specification in that the time that the digital work is in use need not be continuous. For example, if the rights guarantee three days of access, those days could be spread out over a month. With this specification, the rights can be exercised until the meter time is exhausted or the expiration date is reached, whichever comes first.

Remaining-Use:=Time-Unit

Start-Time:=Time-Unit

Use-Duration:=Time-Unit

[0138] All of the time specifications include time-unit specifications in their ultimate instantiation.

[0139] The present invention provides for various security mechanisms to be introduced into a distribution or use scheme. Grammar element 1516 "Access-Spec:={SC: Security-Class} {Authorization: Authorization-ID*} {Other-Authorization: Authorization-ID*} {Ticket: Ticket-ID})" provides a means for restricting access and transmission. Access specifications can specify a required security class for a repository to exercise a right or a required authorization test that must be satisfied.

[0140] The keyword "SC:" is used to specify a minimum security level for the repositories involved in the access. If "SC:" is not specified, the lowest security level is acceptable.

[0141] The optional "Authorization:" keyword is used to specify required authorizations on the same repository as the work. The optional "Other-Authorization:" keyword is used to specify required authorizations on the other repository in the transaction.

[0142] The optional "Ticket:" keyword specifies the identity of a ticket required for the transaction. A transaction involving digital tickets must locate an appropriate digital ticket agent who can "punch" or otherwise validate the ticket before the transaction can proceed. Tickets are described in greater detail below.

[0143] In a transaction involving a repository and a document server, some usage rights may require that the repository have a particular authorization, that the server have some

NVA208575.1

authorization, or that both repositories have (possibly different) authorizations. Authorizations themselves are digital works (hereinafter referred to as an authorization object) that can be moved between repositories in the same manner as other digital works. Their copying and transferring is subject to the same rights and fees as other digital works. A repository is said to have an authorization if that authorization object is contained within the repository.

[0144] In some cases, an authorization may be required from a source other than the document server and repository. An authorization object referenced by an Authorization-ID can contain digital address information to be used to set up a communications link between a repository and the authorization source. These are analogous to phone numbers. For such access tests, the communication would need to be established and authorization obtained before the right could be exercised.

[0145] For one-time usage rights, a variant on this scheme is to have a digital ticket. A ticket is presented to a digital ticket agent, whose type is specified on the ticket. In the simplest case, a certified generic ticket agent, available on all repositories, is available to "punch" the ticket. In other cases, the ticket may contain addressing information for locating a "special" ticket agent. Once a ticket has been punched, it cannot be used again for the same kind of transaction (unless it is unpunched or refreshed in the manner described below.) Punching includes marking the ticket with a timestamp of the date and time it was used. Tickets are digital works and can be copied or transferred between repositories according to their usage rights. In the currently preferred embodiment, a "punched" ticket becomes "unpunched" or "refreshed" when it is copied or extracted. The Copy and Extract operations save the date and time as a property of the digital ticket. When a ticket agent is given a ticket, it can simply check whether the digital copy was made after the last time that it was punched. Of course, the digital ticket must have the copy or extract usage rights associated therewith.

[0146] The capability to unpunch a ticket is important in the following cases:

[0147] A digital work is circulated at low cost with a limitation that it can be used only once.

NVA208575.1

[0148] A digital work is circulated with a ticket that can be used once to give discounts on purchases of other works.

[0149] A digital work is circulated with a ticket (included in the purchase price and possibly embedded in the work) that can be used for a future upgrade.

[0150] In each of these cases, if a paid copy is made of the digital work (including the ticket) the new owner would expect to get a fresh (unpunched) ticket, whether the copy seller has used the work or not. In contrast, loaning a work or simply transferring it to another repository should not revitalize the ticket.

[0151] The billing for use of a digital work is often fundamental to a commercial distribution system. Grammar Element 1517 "Fee-Spec:={Scheduled-Discount} Regular-Fee-Spec.vertline.Scheduled-Fee-Spec.vertline.Markup-Spec" provides a range of options for billing for the use of digital works.

[0152] A key feature of this approach is the development of low-overhead billing for transactions in potentially small amounts. Thus, it becomes feasible to collect fees of only a few cents each for thousands of transactions.

[0153] The grammar differentiates between uses where the charge is per use from those where it is metered by the time unit. Transactions can support fees that the user pays for using a digital work as well as incentives paid by the right grantor to users to induce them to use or distribute the digital work.

[0154] The optional scheduled discount refers to the rest of the fee specification--discounting it by a percentage over time. If it is not specified, then there is no scheduled discount. Regular fee specifications are constant over time. Scheduled fee specifications give a schedule of dates over which the fee specifications change. Markup specifications are used in d-blocks for adding a percentage to the fees already being charged.

NVA208575 1

[0155] Grammar Element 1518 "Scheduled-Discount:=(Scheduled-Discount: (Time-Spec Percentage)*)" A Scheduled-Discount is a essentially a scheduled modifier of any other fee specification for this version of the right of the digital work. (It does not refer to children or parent digital works or to other versions of rights.). It is a list of pairs of times and percentages. The most recent time in the list that has not yet passed at the time of the transaction is the one in effect. The percentage gives the discount percentage. For example, the number 10 refers to a 10% discount.

[0156] Grammar Element 1519 "Regular-Fee-Spec:={Fee:.vertline.Incentive:} [Per-Use-Spec.vertline.Metered-Rate-Spec.vertline.Best-Price-Spec.vertline .Call-For-Price-Spec] {Min: Money-Unit Per: Time-Spec} {Max: Money-Unit Per: Time-Spec} To: Account-ID)" provides for several kinds of fee specifications.

[0157] Fees are paid by the copy-owner/user to the revenue-owner if Fee: is specified. Incentives are paid by the revenue-owner to the user if Incentive: is specified. If the Min: specification is given, then there is a minimum fee to be charged per time-spec unit for its use. If the Max: specification is given, then there is a maximum fee to be charged per time-spec for its use. When Fee: is specified, Account-ID identifies the account to which the fee is to be paid. When Incentive: is specified, Account-ID identifies the account from which the fee is to be paid.

[0158] Grammar element 1520 "Per-Use-Spec:=Per-Use: Money-unit" defines a simple fee to be paid every time the right is exercised, regardless of how much time the transaction takes.

[0159] Grammar element 1521 "Metered-Rate-Spec:=Metered: Money-Unit Per: Time-Spec" defines a metered-rate fee paid according to how long the right is exercised. Thus, the time it takes to complete the transaction determines the fee.

[0160] Grammar element 1522 "Best-Price-Spec:=Best-Price: Money-unit Max: Money-unit" is used to specify a best-price that is determined when the account is settled. This

specification is to accommodate special deals, rebates, and pricing that depends on information that is not available to the repository. All fee specifications can be combined with tickets or authorizations that could indicate that the consumer is a wholesaler or that he is a preferred customer, or that the seller be authorized in some way. The amount of money in the Max: field is the maximum amount that the use will cost. This is the amount that is tentatively debited from the credit server. However, when the transaction is ultimately reconciled, any excess amount will be returned to the consumer in a separate transaction.

[0161] Grammar element 1523 "Call-For-Price-Spec:=Call-For-Price " is similar to a "Best-Price-Spec" in that it is intended to accommodate cases where prices are dynamic. A Call-For-Price Spec requires a communication with a dealer to determine the price. This option cannot be exercised if the repository cannot communicate with a dealer at the time that the right is exercised. It is based on a secure transaction whereby the dealer names a price to exercise the right and passes along a deal certificate which is referenced or included in the billing process.

[0162] Grammar element 1524 "Scheduled-Fee-Spec:=(Schedule: (Time-Spec Regular-Fee-Spec)*)" is used to provide a schedule of dates over which the fee specifications change. The fee specification with the most recent date not in the future is the one that is in effect. This is similar to but more general than the scheduled discount. It is more general, because it provides a means to vary the fee agreement for each time period.

[0163] Grammar element 1525 "Markup-Spec:=Markup: percentage To: Account-ID" is provided for adding a percentage to the fees already being charged. For example, a 5% markup means that a fee of 5% of cumulative fee so far will be allocated to the distributor. A markup specification can be applied to all of the other kinds of fee specifications. It is typically used in a shell provided by a distributor. It refers to fees associated with d-blocks that are parts of the current d-block. This might be a convenient specification for use in taxes, or in distributor overhead.

[0164] Usage rights can be grouped in predetermined sets that are commonly opposite. The following are examples of sets of usage rights

NVA208575.1

[0165] ((Play) (Transfer(SC: 3)) (Delete) This work can be played without requirements for fee or authorization on any rendering system. It can be transferred to any other repository of security level 3 or greater. It can be deleted.

[0166] ((Play) (Transfer (SC: 3)) (Delete) (Backup) (Restore (Fee: Per-Use: \$5 To: Account-ID-678)))

[0167] Same as the previous example plus rights for backup and restore. The work can be backed up without fee. It can be restored for a \$5 fee payable to the account described by Account-ID-678.

[0168] ((Play) (Transfer (SC: 3)) (Copy (SC:3)(Fee: Per-Use: \$5 To: Account-ID-678)) (Delete (Incentive: Per-Use: \$2.50 To: Account-ID-678)))

[0169] This work can be played, transferred, copied, or deleted. Copy or transfer operations can take place only with repositories of security level three or greater. The fee to make a copy is \$5 payable to Account-ID-678. If a copy is deleted, then an incentive of \$2.50 is paid to the former copy owner.

[0170] ((Play) (Transfer (SC: 3)) Copy (SC: 3) (Fee: Per-Use: \$10 To: Account-ID-678)) Delete) (Backup) (Restore (SC: 3) (Fee: Per-Use: \$5 To: Account-ID-678)))

[0171] Same as the previous example plus fees for copying. The work can be copied digitally for a fee of \$10 payable to Account-ID-678. The repository on which the work is copied or restored must be at security level 3 or greater.

[0172] ((Play) (Transfer (SC: 3)) (Copy Authorization: License-123-ID (SC: 3))) The digital work can be played, transferred, or copied. Copies or transfers must be on repositories of security level 3 or greater. Copying requires the license License-123-ID issued to the copying

repository. None of the rights require fees.

[0173] ((Play) (Print Printer: Printer-567-ID (Fee: Per-Use: \$1 To: Account-ID-678)))

This work can be played for free. It can be printed on any printer with the identifier Printer-567-ID for a fee of \$1 payable to the account described by Account-ID-678.

[0174] ((Play Player: Player-876-ID (From: 94/02/14 Until: 95/02/15) (Fee: Metered: \$0.01 Per: 0:1:0 Min: \$0.25 Per: 0/1/0 To: Account-ID-567))) This work can be played on any player holding the ID Player-876-ID. The time of this right is from Feb. 14, 1994 until Feb. 15, 1995. The fee for use is one cent per minute with a minimum of 25 cents in any day that it is used, payable to the account described by Account-ID-567.

[0175] ((Play) (Transfer) (Delete) (Loan 2(Delete: Transfer Loan))) This work can be played, transferred, deleted, or loaned. Up to two copies can be loaned out at a time. The loaned copy has the same rights except that it cannot be transferred. When both copies are loaned out, no rights can be exercised on the original on the repository.

[0176] ((Play) (Transfer) (Delete) (Backup) (Restore (SC:3)) (Loan 2 Remaining-Copy-Rights: (Delete: Play Transfer) Next-Set-of-Rights: (Delete: Transfer Loan))) Similar to previous example. Rights to Backup and Restore the work are added, where restoration requires a repository of at least security level three. When all copies of the work are loaned out, the remaining copy cannot be played or transferred.

[0177] ((Play) (Transfer) (Copy) (Print) (Backup) (Restore (SC:3)) (Loan 1 Remaining-Copy-Rights: (Add: Play Print Backup) Next-Set-of-Rights: (Delete: Transfer Loan) (Fee: Metered: \$10 Per: 1:0:0 To: Account-ID-567))) (Loan 1 Remaining-Copy-Rights: Add: ((Play Player: Player-876-ID) 2 (From: 94/02/14 Until: 95/02/15) (Fee: Metered: \$0.01 Per: 0:1:0 Min: \$0.25 Per: 0/1/0 To: Account-ID-567))))

[0178] The original work has rights to Play, Transfer, Copy, Print, Backup, Restore, and Loan. There are two versions of the Loan right. The first version of the loan right costs \$10 per

NVA208575 1

day but allows the original copy owner to exercise free use of the Play, Print and Backup rights. The second version of the Loan right is free. None of the original rights are applicable. However a right to Play the work at the specified metered rate is added.

[0179] ((Play Player: Player-Small-Screen-123-ID) (Embed (Fee: Per-Use \$.01 To: Account-678-ID)) (Copy (Fee: Per-Use \$1.00 To: Account-678-ID)))

The digital work can be played on any player with the identifier Player-Small-Screen-123-ID. It can be embedded in a larger work. The embedding requires a modest one cent registration fee to Account-678-ID. Digital copies can be made for \$1.00.

[0180] When a user requests access to a digital work, the repository will initiate various transactions. The combination of transactions invoked will depend on the specifications assigned for a usage right. There are three basic types of transactions, Session Initiation Transactions, Financial Transactions and Usage Transactions. Generally, session initiation transactions are initiated first to establish a valid session. When a valid session is established, transactions corresponding to the various usage rights are invoked. Finally, request specific transactions are performed.

[0181] Transactions occur between two repositories (one acting as a server), between a repository and a document playback platform (e.g. for executing or viewing), between a repository and a credit server or between a repository and an authorization server. When transactions occur between more than one repository, it is assumed that there is a reliable communication channel between the repositories. For example, this could be a TCP/IP channel or any other commercially available channel that has built-in capabilities for detecting and correcting transmission errors. However, it is not assumed that the communication channel is secure. Provisions for security and privacy are part of the requirements for specifying and implementing repositories and thus form the need for various transactions.

[0182] Transactions require that there be some communication between repositories. Communication between repositories occurs in units termed as messages. Because the communication line is assumed to be unsecure, all communications with repositories that are

NVA208575 1

above the lowest security class are encrypted utilizing a public key encryption technique. Public key encryption is a well known technique in the encryption arts. The term key refers to a numeric code that is used with encryption and decryption algorithms. Keys come in pairs, where "writing keys" are used to encrypt data and "checking keys" are used to decrypt data. Both writing and checking keys may be public or private. Public keys are those that are distributed to others. Private keys are maintained in confidence.

[0183] Key management and security is instrumental in the success of a public key encryption system. In the currently preferred embodiment, one or more master repositories maintain the keys and create the identification certificates used by the repositories.

[0184] When a sending repository transmits a message to a receiving repository, the sending repository encrypts all of its data using the public writing key of the receiving repository. The sending repository includes its name, the name of the receiving repository, a session identifier such as a nonce (described below), and a message counter in each message.

[0185] In this way, the communication can only be read (to a high probability) by the receiving repository, which holds the private checking key for decryption. The auxiliary data is used to guard against various replay attacks to security. If messages ever arrive with the wrong counter or an old nonce, the repositories can assume that someone is interfering with communication and the transaction terminated.

[0186] The respective public keys for the repositories to be used for encryption are obtained in the registration transaction described below.

[0187] A usage transaction is carried out in a session between repositories. For usage transactions involving more than one repository, or for financial transactions between a repository and a credit server, a registration transaction is performed. A second transaction termed a login transaction, may also be needed to initiate the session. The goal of the registration transaction is to establish a secure channel between two repositories who know each other's identities. As it is assumed that the communication channel between the repositories is reliable

NVA208575 1

but not secure, there is a risk that a non-repository may mimic the protocol in order to gain illegitimate access to a repository.

[0188] The registration transaction between two repositories is described with respect to FIGS. 16 and 17. The steps described are from the perspective of a "repository-1" registering its identity with a "repository-2". The registration must be symmetrical so the same set of steps will be repeated for repository-2 registering its identity with repository-1. Referring to FIG. 16, repository-1 first generates an encrypted registration identifier, step 1601 and then generates a registration message, step 1602. A registration message is comprised of an identifier of a master repository, the identification certificate for the repository-1 and an encrypted random registration identifier. The identification certificate is encrypted by the master repository in its private key and attests to the fact that the repository (here repository-1) is a bona fide repository. The identification certificate also contains a public key for the repository, the repository security level and a timestamp (indicating a time after which the certificate is no longer valid.) The registration identifier is a number generated by the repository for this registration. The registration identifier is unique to the session and is encrypted in repository-1's private key. The registration identifier is used to improve security of authentication by detecting certain kinds of communications based attacks. Repository-1 then transmits the registration message to repository-2, step 1603.

[0189] Upon receiving the registration message, repository-2 determines if it has the needed public key for the master repository, step 1604. If repository-2 does not have the needed public key to decrypt the identification certificate, the registration transaction terminates in an error, step 1618.

[0190] Assuming that repository-2 has the proper public key the identification certificate is decrypted, step 1605. Repository-2 saves the encrypted registration identifier, step 1606, and extracts the repository identifier, step 1607. The extracted repository identifier is checked against a "hotlist" of compromised document repositories, step 1608. In the currently preferred embodiment, each repository will contain "hotlists" of compromised repositories. If the repository is on the "hotlist", the registration transaction terminates in an error per step 1618. Repositories can be removed from the hotlist when their certificates expire, so that the list does

not need to grow without bound. Also, by keeping a short list of hotlist certificates that it has previously received, a repository can avoid the work of actually going through the list. These lists would be encrypted by a master repository. A minor variation on the approach to improve efficiency would have the repositories first exchange lists of names of hotlist certificates, ultimately exchanging only those lists that they had not previously received. The "hotlists" are maintained and distributed by Master repositories.

[0191] Note that rather than terminating in error, the transaction could request that another registration message be sent based on an identification certificate created by another master repository. This may be repeated until a satisfactory identification certificate is found, or it is determined that trust cannot be established.

[0192] Assuming that the repository is not on the hotlist, the repository identification needs to be verified. In other words, repository-2 needs to validate that the repository on the other end is really repository-1. This is termed performance testing and is performed in order to avoid invalid access to the repository via a counterfeit repository replaying a recording of a prior session initiation between repository-1 and repository-2. Performance testing is initiated by repository-2 generating a performance message, step 1609. The performance message consists of a nonce, the names of the respective repositories, the time and the registration identifier received from repository-1. A nonce is a generated message based on some random and variable information (e.g. the time or the temperature.) The nonce is used to check whether repository-1 can actually exhibit correct encrypting of a message using the private keys it claims to have, on a message that it has never seen before. The performance message is encrypted using the public key specified in the registration message of repository-1. The performance message is transmitted to repository-1, step 1610, where it is decrypted by repository-1 using its private key, step 1611. Repository-1 then checks to make sure that the names of the two repositories are correct, step 1612, that the time is accurate, step 1613 and that the registration identifier corresponds to the one it sent, step 1614. If any of these tests fails, the transaction is terminated per step 1616. Assuming that the tests are passed, repository-1 transmits the nonce to repository-2 in the clear, step 1615. Repository-2 then compares the received nonce to the original nonce, step 1617. If they are not identical, the registration transaction terminates in an error per step

NVA208575.1

1618. If they are the same, the registration transaction has successfully completed.

[0193] At this point, assuming that the transaction has not terminated, the repositories exchange messages containing session keys to be used in all communications during the session and synchronize their clocks. FIG. 17 illustrates the session information exchange and clock synchronization steps (again from the perspective of repository-1.) Referring to FIG. 17, repository-1 creates a session key pair, step 1701. A first key is kept private and is used by repository-1 to encrypt messages. The second key is a public key used by repository-2 to decrypt messages. The second key is encrypted using the public key of repository-2, step 1702 and is sent to repository-2, step 1703. Upon receipt, repository-2 decrypts the second key, step 1704. The second key is used to decrypt messages in subsequent communications. When each repository has completed this step, they are both convinced that the other repository is bona fide and that they are communicating with the original. Each repository has given the other a key to be used in decrypting further communications during the session. Since that key is itself transmitted in the public key of the receiving repository only it will be able to decrypt the key which is used to decrypt subsequent messages.

[0194] After the session information is exchanged, the repositories must synchronize their clocks. Clock synchronization is used by the repositories to establish an agreed upon time base for the financial records of their mutual transactions. Referring back to FIG. 17, repository-2 initiates clock synchronization by generating a time stamp exchange message, step 1705, and transmits it to repository-1, step 1706. Upon receipt, repository-1 generates its own time stamp message, step 1707 and transmits it back to repository-2, step 1708. Repository-2 notes the current time, step 1709 and stores the time received from repository-1, step 1710. The current time is compared to the time received from repository-1, step 1711. The difference is then checked to see if it exceeds a predetermined tolerance (e.g. one minute), step 1712. If it does, repository-2 terminates the transaction as this may indicate tampering with the repository, step 1713. If not repository-2 computes an adjusted time delta, step 1714. The adjusted time delta is the difference between the clock time of repository-2 and the average of the times from repository-1 and repository-2.

[0195] To achieve greater accuracy, repository-2 can request the time again up to a fixed

NVA208575.1

number of times (e.g. five times), repeat the clock synchronization steps, and average the results.

[0196] A second session initiation transaction is a Login transaction. The Login transaction is used to check the authenticity of a user requesting a transaction. A Login transaction is particularly prudent for the authorization of financial transactions that will be charged to a credit server. The Login transaction involves an interaction between the user at a user interface and the credit server associated with a repository. The information exchanged here is a login string supplied by the repository/credit server to identify itself to the user, and a Personal Identification Number (PIN) provided by the user to identify himself to the credit server. In the event that the user is accessing a credit server on a repository different from the one on which the user interface resides, exchange of the information would be encrypted using the public and private keys of the respective repositories.

[0197] Billing Transactions are concerned with monetary transaction with a credit server. Billing transactions are carried out when all other conditions are satisfied and a usage fee is required for granting the request. For the most part, billing transactions are well understood in the state of the art. These transactions are between a repository and a credit server, or between a credit server and a billing clearinghouse. Briefly, the required transactions include the following:

[0198] Registration and LOGIN transactions by which the repository and user establish their bona fides to a credit server. These transactions would be entirely internal in cases where the repository and credit server are implemented as a single system.

[0199] Registration and LOGIN transactions, by which a credit server establishes its bona fides to a billing clearinghouse.

[0200] An Assign-fee transaction to assign a charge. The information in this transaction would include a transaction identifier, the identities of the repositories in the transaction, and a list of charges from the parts of the digital work. If there has been any unusual event in the transaction such as an interruption of communications, that information is included as well.

[0201] An Begin-charges transaction to assign a charge. This transaction is much the

NVA208575.1

same as an assign-fee transaction except that it is used for metered use. It includes the same information as the assign-fee transaction as well as the usage fee information. The credit-server is then responsible for running a clock.

[0202] An End-charges transaction to end a charge for metered use. (In a variation on this approach, the repositories would exchange periodic charge information for each block of time.)

[0203] A report-charges transaction between a personal credit server and a billing clearinghouse. This transaction is invoked at least once per billing period. It is used to pass along information about charges. On debit and credit cards, this transaction would also be used to update balance information and credit limits as needed.

[0204] All billing transactions are given a transaction ID and are reported to the credit servers by both the server and the client. This reduces possible loss of billing information if one of the parties to a transaction loses a banking card and provides a check against tampering with the system.

[0205] After the session initiation transactions have been completed, the usage request may then be processed. To simplify the description of the steps carried out in processing a usage request, the term requester is used to refer to a repository in the requester mode which is initiating a request, and the term server is used to refer to a repository in the server mode and which contains the desired digital work. In many cases such as requests to print or view a work, the requester and server may be the same device and the transactions described in the following would be entirely internal. In such instances, certain transaction steps, such as the registration transaction, need not be performed.

[0206] There are some common steps that are part of the semantics of all of the usage rights transactions. These steps are referred to as the common transaction steps. There are two sets--the "opening" steps and the "closing" steps. For simplicity, these are listed here rather than repeating them in the descriptions of all of the usage rights transactions.

NVA208575.1

[0207] Transactions can refer to a part of a digital work, a complete digital work, or a digital work containing other digital works. Although not described in detail herein, a transaction may even refer to a folder comprised of a plurality of digital works. The term "work" is used to refer to what ever portion or set of digital works is being accessed.

[0208] Many of the steps here involve determining if certain conditions are satisfied. Recall that each usage right may have one or more conditions which must be satisfied before the right can be exercised. Digital works have parts and parts have parts. Different parts can have different rights and fees. Thus, it is necessary to verify that the requirements are met for ALL of the parts that are involved in a transaction For brevity, when reference is made to checking whether the rights exist and conditions for exercising are satisfied, it is meant that all such checking takes place for each of the relevant parts of the work.

[0209] FIG. 18 illustrates the initial common opening and closing steps for a transaction. At this point it is assumed that registration has occurred and that a "trusted" session is in place. General tests are tests on usage rights associated with the folder containing the work or some containing folder higher in the file system hierarchy. These tests correspond to requirements imposed on the work as a consequence of its being on the particular repository, as opposed to being attached to the work itself. Referring to FIG. 18, prior to initiating a usage transaction, the requester performs any general tests that are required before the right associated with the transaction can be exercised, step, 1801. For example, install, uninstall and delete rights may be implemented to require that a requester have an authorization certificate before the right can be exercised. Another example is the requirement that a digital ticket be present and punched before a digital work may be copied to a requester. If any of the general tests fail, the transaction is not initiated, step, 1802. Assuming that such required tests are passed, upon receiving the usage request, the server generates a transaction identifier that is used in records or reports of the transaction, step 1803. The server then checks whether the digital work has been granted the right corresponding to the requested transaction, step 1804. If the digital work has not been granted the right corresponding to the request, the transaction terminates, step 1805. If the digital work has been granted the requested right, the server then determines if the various conditions

for exercising the right are satisfied. Time based conditions are examined, step 1806. These conditions are checked by examining the time specification for the version of the right. If any of the conditions are not satisfied, the transaction terminates per step 1805.

[0210] Assuming that the time based conditions are satisfied, the server checks security and access conditions, step 1807. Such security and access conditions are satisfied if: 1) the requester is at the specified security class, or a higher security class, 2) the server satisfies any specified authorization test and 3) the requester satisfies any specified authorization tests and has any required digital tickets. If any of the conditions are not satisfied, the transaction terminates per step 1805.

[0211] Assuming that the security and access conditions are all satisfied, the server checks the copy count condition, step 1808. If the copy count equals zero, then the transaction cannot be completed and the transaction terminates per step 1805.

[0212] Assuming that the copy count does not equal zero, the server checks if the copies in use for the requested right is greater than or equal to any copy count for the requested right (or relevant parts), step 1809. If the copies in use is greater than or equal to the copy count, this indicates that usage rights for the version of the transaction have been exhausted. Accordingly, the server terminates the transaction, step 1805. If the copy count is less than the copies in use for the transaction the transaction can continue, and the copies in use would be incremented by the number of digital works requested in the transaction, step 1810.

[0213] The server then checks if the digital work has a "Loan" access right, step 1811. The "Loan" access right is a special case since remaining rights may be present even though all copies are loaned out. If the digital work has the "Loan" access right, a check is made to see if all copies have been loaned out, step 1812. The number of copies that could be loaned is the sum of the Copy-Counts for all of the versions of the loan right of the digital work. For a composite work, the relevant figure is the minimal such sum of each of the components of the composite work. If all copies have been loaned out, the remaining rights are determined, step 1813. The remaining-rights is determined from the remaining rights specifications from the versions of the

NVA208575.1

Loan right. If there is only one version of the Loan right, then the determination is simple. The remaining rights are the ones specified in that version of the Loan right, or none if Remaining-Rights: is not specified. If there are multiple versions of the Loan right and all copies of all of the versions are loaned out, then the remaining rights is taken as the minimum set (intersection) of remaining rights across all of the versions of the loan right. The server then determines if the requested right is in the set of remaining rights, step 1814. If the requested right is not in the set of remaining rights, the server terminates the transaction, step 1805.

[0214] If Loan is not a usage right for the digital work or if all copies have not been loaned out or the requested right is in the set of remaining rights, fee conditions for the right are then checked, step 1815. This will initiate various financial transactions between the repository and associated credit server. Further, any metering of usage of a digital work will commence. If any financial transaction fails, the transaction terminates per step 1805.

[0215] It should be noted that the order in which the conditions are checked need not follow the order of steps 1806-1815.

[0216] At this point, right specific steps are now performed and are represented here as step 1816. The right specific steps are described in greater detail below.

[0217] The common closing transaction steps are now performed. Each of the closing transaction steps are performed by the server after a successful completion of a transaction. Referring back to FIG. 18, the copies in use value for the requested right is decremented by the number of copies involved in the transaction, step 1817. Next, if the right had a metered usage fee specification, the server subtracts the elapsed time from the Remaining-Use-Time associated with the right for every part involved in the transaction, step 1818. Finally, if there are fee specifications associated with the right, the server initiates End-Charge financial transaction to confirm billing, step 1819.

[0218] An important area to consider is the transmission of the digital work from the server to the requester. The transmission protocol described herein refers to events occurring

NVA208575.1

after a valid session has been created. The transmission protocol must handle the case of disruption in the communications between the repositories. It is assumed that interference such as injecting noise on the communication channel can be detected by the integrity checks (e.g., parity, checksum, etc.) that are built into the transport protocol and are not discussed in detail herein.

[0219] The underlying goal in the transmission protocol is to preclude certain failure modes, such as malicious or accidental interference on the communications channel. Suppose, for example, that a user pulls a card with the credit server at a specific time near the end of a transaction. There should not be a vulnerable time at which "pulling the card" causes the repositories to fail to correctly account for the number of copies of the work that have been created. Restated, there should be no time at which a party can break a connection as a means to avoid payment after using a digital work.

[0220] If a transaction is interrupted (and fails), both repositories restore the digital works and accounts to their state prior to the failure, modulo records of the failure itself.

[0221] FIG. 19 is a state diagram showing steps in the process of transmitting information during a transaction. Each box represents a state of a repository in either the server mode (above the central dotted line 1901) or in the requester mode (below the dotted line 1901). Solid arrows stand for transitions between states. Dashed arrows stand for message communications between the repositories. A dashed message arrow pointing to a solid transition arrow is interpreted as meaning that the transition takes place when the message is received. Unlabeled transition arrows take place unconditionally. Other labels on state transition arrows describe conditions that trigger the transition.

[0222] Referring now to FIG. 19, the server is initially in a state 1902 where a new transaction is initiated via start message 1903. This message includes transaction information including a transaction identifier and a count of the blocks of data to be transferred. The requester, initially in a wait state 1904 then enters a data wait state 1905.

[0223] The server enters a data transmit state 1906 and transmits a block of data 1907 and then enters a wait for acknowledgement state 1908. As the data is received, the requester enters a data receive state 1909 and when the data blocks is completely received it enters an acknowledgement state 1910 and transmits an Acknowledgement message 1911 to the server.

[0224] If there are more blocks to send, the server waits until receiving an Acknowledgement message from the requester. When an Acknowledgement message is received it sends the next block to the requester and again waits for acknowledgement. The requester also repeats the same cycle of states.

[0225] If the server detects a communications failure before sending the last block, it enters a cancellation state 1912 wherein the transaction is cancelled. Similarly, if the requester detects a communications failure before receiving the last block it enters a cancellation state 1913.

[0226] If there are no more blocks to send, the server commits to the transaction and waits for the final Acknowledgement in state 1914. If there is a communications failure before the server receives the final Acknowledgement message, it still commits to the transaction but includes a report about the event to its credit server in state 1915. This report serves two purposes. It will help legitimize any claims by a user of having been billed for receiving digital works that were not completely received. Also it helps to identify repositories and communications lines that have suspicious patterns of use and interruption. The server then enters its completion state 1916.

[0227] On the requester side, when there are no more blocks to receive, the requester commits to the transaction in state 1917. If the requester detects a communications failure at this state, it reports the failure to its credit server in state 1918, but still commits to the transaction. When it has committed, it sends an acknowledgement message to the server. The server then enters its completion state 1919.

[0228] The key property is that both the server and the requester cancel a transaction if it is interrupted before all of the data blocks are delivered, and commits to it if all of the data

blocks have been delivered.

[0229] There is a possibility that the server will have sent all of the data blocks (and committed) but the requester will not have received all of them and will cancel the transaction. In this case, both repositories will presumably detect a communications failure and report it to their credit server. This case will probably be rare since it depends on very precise timing of the communications failure. The only consequence will be that the user at the requester repository may want to request a refund from the credit services--and the case for that refund will be documented by reports by both repositories.

[0230] To prevent loss of data, the server should not delete any transferred digital work until receiving the final acknowledgement from the requester. But it also should not use the file. A well known way to deal with this situation is called "two-phase commit" or 2PC. Two-phase commit works as follows. The first phase works the same as the method described above. The server sends all of the data to the requester. Both repositories mark the transaction (and appropriate files) as uncommitted. The server sends a ready-to-commit message to the requester. The requester sends back an acknowledgement. The server then commits and sends the requester a commit message. When the requester receives the commit message, it commits the file.

[0231] If there is a communication failure or other crash, the requester must check back with the server to determine the status of the transaction. The server has the last word on this. The requester may have received all of the data, but if it did not get the final message, it has not committed. The server can go ahead and delete files (except for transaction records) once it commits, since the files are known to have been fully transmitted before starting the 2PC cycle.

[0232] There are variations known in the art which can be used to achieve the same effect. For example, the server could use an additional level of encryption when transmitting a work to a client. Only after the client sends a message acknowledging receipt does it send the key. The client then agrees to pay for the digital work. The point of this variation is that it provides a clear audit trail that the client received the work. For trusted systems, however, this

NVA208575 1

variation adds a level of encryption for no real gain in accountability.

[0233] The transaction for specific usage rights are now discussed. A Copy transaction is a request to make one or more independent copies of the work with the same or lesser usage rights. Copy differs from the extraction right discussed later in that it refers to entire digital works or entire folders containing digital works. A copy operation cannot be used to remove a portion of a digital work.

[0234] The requester sends the server a message to initiate the Copy Transaction. This message indicates the work to be copied, the version of the copy right to be used for the transaction, the destination address information (location in a folder) for placing the work, the file data for the work (including its size), and the number of copies requested.

[0235] The repositories perform the common opening transaction steps. The server transmits the requested contents and data to the client according to the transmission protocol. If a Next-Set-Of-Rights has been provided in the version of the right, those rights are transmitted as the rights for the work. Otherwise, the rights of the original are transmitted. In any event, the Copy-Count field for the copy of the digital work being sent right is set to the number-of-copies requested.

[0236] The requester records the work contents, data, and usage rights and stores the work. It records the date and time that the copy was made in the properties of the digital work.

[0237] The repositories perform the common closing transaction steps. A Transfer transaction is a request to move copies of the work with the same or lesser usage rights to another repository. In contrast with a copy transaction, this results in removing the work copies from the server.

[0238] The requester sends the server a message to initiate the Transfer Transaction. This message indicates the work to be transferred, the version of the transfer right to be used in the

transaction, the destination address information for placing the work, the file data for the work, and the number of copies involved.

[0239] The repositories perform the common opening transaction steps. The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, the rights of the original are transmitted. In either case, the Copy-Count field for the transmitted rights are set to the number-of-copies requested.

[0240] The requester records the work contents, data, and usage rights and stores the work.

[0241] The server decrements its copy count by the number of copies involved in the transaction.

[0242] The repositories perform the common closing transaction steps.

[0243] If the number of copies remaining in the server is now zero, it erases the digital work from its memory.

[0244] A loan transaction is a mechanism for loaning copies of a digital work. The maximum duration of the loan is determined by an internal parameter of the digital work. Works are automatically returned after a predetermined time period.

[0245] The requester sends the server a message to initiate the Transfer Transaction. This message indicates the work to be loaned, the version of the loan right to be used in the transaction, the destination address information for placing the work, the number of copies involved, the file data for the work, and the period of the loan.

[0246] The server checks the validity of the requested loan period, and ends with an error if the period is not valid. Loans for a loaned copy cannot extend beyond the period of the original

NVA208575.1

loan to the server.

[0247] The repositories perform the common opening transaction steps. The server transmits the requested contents and data to the requester. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, the rights of the original are transmitted, as modified to reflect the loan period.

[0248] The requester records the digital work contents, data, usage rights, and loan period and stores the work.

[0249] The server updates the usage rights information in the digital work to reflect the number of copies loaned out.

[0250] The repositories perform the common closing transaction steps. The server updates the usage rights data for the digital work. This may preclude use of the work until it is returned from the loan. The user on the requester platform can now use the transferred copies of the digital work. A user accessing the original repository cannot use the digital work, unless there are copies remaining. What happens next depends on the order of events in time.

[0251] Case 1. If the time of the loan period is not yet exhausted and the requester sends the repository a Return message.

[0252] The return message includes the requester identification, and the transaction ID. The server decrements the copies-in-use field by the number of copies that were returned. (If the number of digital works returned is greater than the number actually borrowed, this is treated as an error.) This step may now make the work available at the server for other users.

[0253] The requester deactivates its copies and removes the contents from its memory.

[0254] Case 2. If the time of the loan period is exhausted and the requester has not yet sent a Return message. The server decrements the copies-in-use field by the number digital

NVA208575 1

works that were borrowed.

[0255] The requester automatically deactivates its copies of the digital work. It terminates all current uses and erases the digital work copies from memory. One question is why a requester would ever return a work earlier than the period of the loan, since it would be returned automatically anyway. One reason for early return is that there may be a metered fee which determines the cost of the loan. Returning early may reduce that fee.

[0256] A play transaction is a request to use the contents of a work. Typically, to "play" a work is to send the digital work through some kind of transducer, such as a speaker or a display device. The request implies the intention that the contents will not be communicated digitally to any other system. For example, they will not be sent to a printer, recorded on any digital medium, retained after the transaction or sent to another repository.

[0257] This term "play" is natural for examples like playing music, playing a movie, or playing a video game. The general form of play means that a "player" is used to use the digital work. However, the term play covers all media and kinds of recordings. Thus one would "play" a digital work, meaning, to render it for reading, or play a computer program, meaning to execute it. For a digital ticket the player would be a digital ticket agent.

[0258] The requester sends the server a message to initiate the play transaction. This message indicates the work to be played, the version of the play right to be used in the transaction, the identity of the player being used, and the file data for the work.

[0259] The server checks the validity of the player identification and the compatibility of the player identification with the player specification in the right. It ends with an error if these are not satisfactory.

[0260] The repositories perform the common opening transaction steps. The server and requester read and write the blocks of data as requested by the player according to the transmission protocol. The requester plays the work contents, using the player.

NVA208575.1

[0261] When the player is finished, the player and the requester remove the contents from their memory.

[0262] The repositories perform the common closing transaction steps. A Print transaction is a request to obtain the contents of a work for the purpose of rendering them on a "printer." We use the term "printer" to include the common case of writing with ink on paper. However, the key aspect of "printing" in our use of the term is that it makes a copy of the digital work in a place outside of the protection of usage rights. As with all rights, this may require particular authorization certificates.

[0263] Once a digital work is printed, the publisher and user are bound by whatever copyright laws are in effect. However, printing moves the contents outside the control of repositories. For example, absent any other enforcement mechanisms, once a digital work is printed on paper, it can be copied on ordinary photocopying machines without intervention by a repository to collect usage fees. If the printer to a digital disk is permitted, then that digital copy is outside of the control of usage rights. Both the creator and the user know this, although the creator does not necessarily give tacit consent to such copying, which may violate copyright laws.

[0264] The requester sends the server a message to initiate a Print transaction. This message indicates the work to be played, the identity of the printer being used, the file data for the work, and the number of copies in the request.

[0265] The server checks the validity of the printer identification and the compatibility of the printer identification with the printer specification in the right. It ends with an error if these are not satisfactory.

[0266] The repositories perform the common opening transaction steps.

[0267] The server transmits blocks of data according to the transmission protocol.

[0268] The requester prints the work contents, using the printer.

[0269] When the printer is finished, the printer and the requester remove the contents from their memory.

[0270] The repositories perform the common closing transaction steps. A Backup transaction is a request to make a backup copy of a digital work, as a protection against media failure. In the context of repositories, secure backup copies differ from other copies in three ways: (1) they are made under the control of a Backup transaction rather than a Copy transaction, (2) they do not count as regular copies, and (3) they are not usable as regular copies. Generally, backup copies are encrypted.

[0271] Although backup copies may be transferred or copied, depending on their assigned rights, the only way to make them useful for playing, printing or embedding is to restore them.

[0272] The output of a Backup operation is both an encrypted data file that contains the contents and description of a work, and a restoration file with an encryption key for restoring the encrypted contents. In many cases, the encrypted data file would have rights for "printing" it to a disk outside of the protection system, relying just on its encryption for security. Such files could be stored anywhere that was physically safe and convenient. The restoration file would be held in the repository. This file is necessary for the restoration of a backup copy. It may have rights for transfer between repositories.

[0273] The requester sends the server a message to initiate a backup transaction. This message indicates the work to be backed up, the version of the backup right to be used in the transaction, the destination address information for placing the backup copy, the file data for the work.

[0274] The repositories perform the common opening transaction steps. The server transmits the requested contents and data to the requester. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, a set of default rights for backup files of the original are transmitted by the server.

[0275] The requester records the work contents, data, and usage rights. It then creates a one-time key and encrypts the contents file. It saves the key information in a restoration file.

[0276] The repositories perform the common closing transaction steps. In some cases, it is convenient to be able to archive the large, encrypted contents file to secure offline storage, such as a magneto-optical storage system or magnetic tape. This creation of a non-repository archive file is as secure as the encryption process. Such non-repository archive storage is considered a form of "printing" and is controlled by a print right with a specified "archive-printer." An archive-printer device is programmed to save the encrypted contents file (but not the description file) offline in such a way that it can be retrieved.

[0277] The Restore Transaction A Restore transaction is a request to convert an encrypted backup copy of a digital work into a usable copy. A restore operation is intended to be used to compensate for catastrophic media failure. Like all usage rights, restoration rights can include fees and access tests including authorization checks.

[0278] The requester sends the server a message to initiate a Restore transaction. This message indicates the work to be restored, the version of the restore right for the transaction, the destination address information for placing the work, and the file data for the work.

[0279] The server verifies that the contents file is available (i.e. a digital work corresponding to the request has been backed-up.) If it is not, it ends the transaction with an error.

[0280] The repositories perform the common opening transaction steps.

[0281] The server retrieves the key from the restoration file. It decrypts the work contents, data, and usage rights.

[0282] The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, a set of default rights for backup files of the original are transmitted by the server.

[0283] The requester stores the digital work.

[0284] The repositories perform the common closing transaction steps. A Delete transaction deletes a digital work or a number of copies of a digital work from a repository. Practically all digital works would have delete rights.

[0285] The requester sends the server a message to initiate a delete transaction. This message indicates the work to be deleted, the version of the delete right for the transaction.

[0286] The repositories perform the common opening transaction steps.

[0287] The server deletes the file, erasing it from the file system.

[0288] The repositories perform the common closing transaction steps. A Directory transaction is a request for information about folders, digital works, and their parts. This amounts to roughly the same idea as protection codes in a conventional file system like TENEX, except that it is generalized to the full power of the access specifications of the usage rights language.

[0289] The Directory transaction has the important role of passing along descriptions of the rights and fees associated with a digital work. When a user wants to exercise a right, the user interface of his repository implicitly makes a directory request to determine the versions of the right that are available. Typically these are presented to the user--such as with different choices of billing for exercising a right. Thus, many directory transactions are invisible to the user and

NVA208575 1

are exercised as part of the normal process of exercising all rights.

[0290] The requester sends the server a message to initiate a Directory transaction. This message indicates the file or folder that is the root of the directory request and the version of the directory right used for the transaction.

[0291] The server verifies that the information is accessible to the requester. In particular, it does not return the names of any files that have a HIDE-NAME status in their directory specifications, and it does not return the parts of any folders or files that have HIDE-PARTS in their specification. If the information is not accessible, the server ends the transaction with an error.

[0292] The repositories perform the common opening transaction steps.

[0293] The server sends the requested data to the requester according to the transmission protocol. The requester records the data.

[0294] The repositories perform the common closing transaction steps. A Folder transaction is a request to create or rename a folder, or to move a work between folders. Together with Directory rights, Folder rights control the degree to which organization of a repository can be accessed or modified from another repository.

[0295] The requester sends the server a message to initiate a Folder transaction. This message indicates the folder that is the root of the folder request, the version of the folder right for the transaction, an operation, and data. The operation can be one of create, rename, and move file. The data are the specifications required for the operation, such as a specification of a folder or digital work and a name.

[0296] The repositories perform the common opening transaction steps. The server performs the requested operation--creating a folder, renaming a folder, or moving a work between folders.

NVA208575.1

[0297] The repositories perform the common closing transaction steps. An extract transaction is a request to copy a part of a digital work and to create a new work containing it. The extraction operation differs from copying in that it can be used to separate a part of a digital work from d-blocks or shells that place additional restrictions or fees on it. The extraction operation differs from the edit operation in that it does not change the contents of a work, only its embedding in d-blocks. Extraction creates a new digital work.

[0298] The requester sends the server a message to initiate an Extract transaction. This message indicates the part of the work to be extracted, the version of the extract right to be used in the transaction, the destination address information for placing the part as a new work, the file data for the work, and the number of copies involved.

[0299] The repositories perform the common opening transaction steps. The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the new work. Otherwise, the rights of the original are transmitted. The Copy-Count field for this right is set to the number-of-copies requested.

[0300] The requester records the contents, data, and usage rights and stores the work. It records the date and time that new work was made in the properties of the work.

[0301] The repositories perform the common closing transaction steps. An embed transaction is a request to make a digital work become a part of another digital work or to add a shell d-block to enable the adding of fees by a distributor of the work.

[0302] The requester sends the server a message to initiate an Embed transaction. This message indicates the work to be embedded, the version of the embed right to be used in the transaction, the destination address information for placing the part as a work, the file data for the work, and the number of copies involved.

[0303] The server checks the control specifications for all of the rights in the part and the

NVA208575.1

destination. If they are incompatible, the server ends the transaction with an error.

[0304] The repositories perform the common opening transaction steps. The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the new work. Otherwise, the rights of the original are transmitted. The Copy-Count field for this right is set to the number-of-copies requested.

[0305] The requester records the contents, data, and usage rights and embeds the work in the destination file.

[0306] The repositories perform the common closing transaction steps. An Edit transaction is a request to revise a digital work by copying, selecting and modifying portions of an existing digital work. This operation can actually change the contents of a digital work. The kinds of changes that are permitted depend on the process being used. Like the extraction operation, edit operates on portions of a digital work. In contrast with the extract operation, edit does not effect the rights or location of the work. It only changes the contents. The kinds of changes permitted are determined by the type specification of the processor specified in the rights. In the currently preferred embodiment, an edit transaction changes the work itself and does not make a new work. However, it would be a reasonable variation to cause a new copy of the work to be made.

[0307] The requester sends the server a message to initiate an Edit transaction. This message indicates the work to be edited, the version of the edit right to be used in the transaction, the file data for the work (including its size), the process-ID for the process, and the number of copies involved.

[0308] The server checks the compatibility of the process-ID to be used by the requester against any process-ID specification in the right. If they are incompatible, it ends the transaction with an error.

[0309] The repositories perform the common opening transaction steps. The requester uses the process to change the contents of the digital work as desired. (For example, it can select and duplicate parts of it; combine it with other information; or compute functions based on the information. This can amount to editing text, music, or pictures or taking whatever other steps are useful in creating a derivative work.)

[0310] The repositories perform the common closing transaction steps. The edit transaction is used to cover a wide range of kinds of works. The category describes a process that takes as its input any portion of a digital work and then modifies the input in some way. For example, for text, a process for editing the text would require edit rights. A process for "summarizing" or counting words in the text would also be considered editing. For a music file, processing could involve changing the pitch or tempo, or adding reverberations, or any other audio effect. For digital video works, anything which alters the image would require edit rights. Examples would be colorizing, scaling, extracting still photos, selecting and combining frames into story boards, sharpening with signal processing, and so on.

[0311] Some creators may want to protect the authenticity of their works by limiting the kinds of processes that can be performed on them. If there are no edit rights, then no processing is allowed at all. A processor identifier can be included to specify what kind of process is allowed. If no process identifier is specified, then arbitrary processors can be used. For an example of a specific process, a photographer may want to allow use of his photograph but may not want it to be colorized. A musician may want to allow extraction of portions of his work but not changing of the tonality.

[0312] There are many ways that authorization transactions can be defined. In the following, our preferred way is to simply define them in terms of other transactions that we already need for repositories. Thus, it is convenient sometimes to speak of "authorization transactions," but they are actually made up of other transactions that repositories already have.

[0313] A usage right can specify an authorization-ID, which identifies an authorization object (a digital work in a file of a standard format) that the repository must have and which it

NVA208575 1

must process. The authorization is given to the generic authorization (or ticket) server of the repository which begins to interpret the authorization.

[0314] As described earlier, the authorization contains a server identifier, which may just be the generic authorization server or it may be another server. When a remote authorization server is required, it must contain a digital address. It may also contain a digital certificate.

[0315] If a remote authorization server is required, then the authorization process first performs the following steps:

[0316] The generic authorization server attempts to set up the communications channel. (If the channel cannot be set up, then authorization fails with an error.)

[0317] When the channel is set up, it performs a registration process with the remote repository. (If registration fails, then the authorization fails with an error.)

[0318] When registration is complete, the generic authorization server invokes a "Play" transaction with the remote repository, supplying the authorization document as the digital work to be played, and the remote authorization server (a program) as the "player." (If the player cannot be found or has some other error, then the authorization fails with an error.)

[0319] The authorization server then "plays" the authorization. This involves decrypting it using either the public key of the master repository that issued the certificate or the session key from the repository that transmitted it. The authorization server then performs various tests. These tests vary according to the authorization server. They include such steps as checking issue and validity dates of the authorization and checking any hot-lists of known invalid authorizations. The authorization server may require carrying out any other transactions on the repository as well, such as checking directories, getting some person to supply a password, or playing some other digital work. It may also invoke some special process for checking information about locations or recent events. The "script" for such steps is contained within the authorization server.

NVA208575 1

[0320] If all of the required steps are completed satisfactorily, the authorization server completes the transaction normally, signaling that authorization is granted.

[0321] An Install transaction is a request to install a digital work as runnable software on a repository. In a typical case, the requester repository is a rendering repository and the software would be a new kind or new version of a player. Also in a typical case, the software would be copied to file system of the requester repository before it is installed.

[0322] The requester sends the server an Install message. This message indicates the work to be installed, the version of the Install right being invoked, and the file data for the work (including its size).

[0323] The repositories perform the common opening transaction steps. The requester extracts a copy of the digital certificate for the software. If the certificate cannot be found or the master repository for the certificate is not known to the requester, the transaction ends with an error.

[0324] The requester decrypts the digital certificate using the public key of the master repository, recording the identity of the supplier and creator, a key for decrypting the software, the compatibility information, and a tamper-checking code. (This step certifies the software.)

[0325] The requester decrypts the software using the key from the certificate and computes a check code on it using a 1-way hash function. If the check-code does not match the tamper-checking code from the certificate, the installation transaction ends with an error. (This step assures that the contents of the software, including the various scripts, have not been tampered with.)

[0326] The requester retrieves the instructions in the compatibility-checking script and follows them. If the software is not compatible with the repository, the installation transaction ends with an error. (This step checks platform compatibility.)

[0327] The requester retrieves the instructions in the installation script and follows them. If there is an error in this process (such as insufficient resources), then the transaction ends with an error. Note that the installation process puts the runnable software in a place in the repository where it is no longer accessible as a work for exercising any usage rights other than the execution of the software as part of repository operations in carrying out other transactions.

[0328] The repositories perform the common closing transaction steps. An Uninstall transaction is a request to remove software from a repository. Since uncontrolled or incorrect removal of software from a repository could compromise its behavioral integrity, this step is controlled.

[0329] The requester sends the server an Uninstall message. This message indicates the work to be uninstalled, the version of the Uninstall right being invoked, and the file data for the work (including its size).

[0330] The repositories perform the common opening transaction steps. The requester extracts a copy of the digital certificate for the software. If the certificate cannot be found or the master repository for the certificate is not known to the requester, the transaction ends with an error.

[0331] The requester checks whether the software is installed. If the software is not installed, the transaction ends with an error.

[0332] The requester decrypts the digital certificate using the public key of the master repository, recording the identity of the supplier and creator, a key for decrypting the software, the compatibility information, and a tamper-checking code. (This step authenticates the certification of the software, including the script for uninstalling it.)

[0333] The requester decrypts the software using the key from the certificate and computes a check code on it using a 1-way hash function. If the check-code does not match the tamper-checking code from the certificate, the installation transaction ends with an error. (This

NVA208575.1

step assures that the contents of the software, including the various scripts, have not been tampered with.)

[0334] The requester retrieves the instructions in the uninstallation script and follows them. If there is an error in this process (such as insufficient resources), then the transaction ends with an error.

[0335] The repositories perform the common closing transaction steps. To appreciate the robustness and flexibility of the present invention, various distribution and use scenarios for digital works are illustrated below. These scenarios are meant to be exemplary rather than exhaustive.

[0336] In this scenario, a creator distributes copies of his works to various consumers. Each consumer is a potential distributor of the work. If the consumer copies the digital work (usually for a third party), a fee is collected and automatically paid to the creator.

[0337] This scenario is a new twist for digital works. It depends on the idea that "manufacturing" is just copying and is essentially free. It also assumes that the consumers as distributors do not require a fee for their time and effort in distributing the work.

[0338] This scenario is performed as follows: A creator creates a digital work. He grants a Copy right with fees paid back to himself. If he does not grant an Embed right, then consumers cannot use the mechanism to act as distributors to cause fees to be paid to themselves on future copies. Of course, they could negotiate side deals or trades to transfer money on their own, outside of the system.

[0339] In another scenario, every time a copy of a digital work is sold a fee is paid to the creator and also to the immediate distributor.

[0340] This scenario does not give special status to any particular distributor. Anyone who sells a document has the right to add a fee to the sale price. The fee for sale could be

established by the consumer. It could also be a fixed nominal amount that is contributed to the account of some charity.

[0341] This scenario is performed as follows: A creator creates a digital work. He grants a Copy right with fees to be paid back to himself. He grants an Embed right, so that anyone can add shells to have fees paid to themselves.

[0342] A distributor embeds the work in a shell, with fees specified to be paid back to himself. If the distributor is content to receive fees only for copies that he sells himself, he grants an Extract right on the shell.

[0343] When a consumer buys a copy from the distributor, fees are paid both to the distributor and to the creator. If he chooses, the consumer can extract the work from the distributor's shell. He cannot extract it from the creator's shell. He can add his own shell with fees to be paid to himself.

[0344] In this scenario, a creator wants to protect the reputation and value of his work by making certain requirements on its distributors. He issues licenses to distributors that satisfy the requirements, and in turn, promises to reward their efforts by assuring that the work will not be distributed over competing channels. The distributors incur expenses for selecting the digital work, explaining it to buyers, promoting its sale, and possibly for the license itself. The distributor obtains the right to enclose the digital work in a shell, whose function is to permit the attachment of usage fees to be paid to the distributor in addition to the fees to be paid to the creator.

[0345] This differs from the previous scenario in that it precludes the typical copy owner from functioning as a distributor, since the consumer lacks a license to copy the document. Thus, a consumer cannot make copies, even for free. All copies must come initially from authorized distributors. This version makes it possible to hold distributors accountable in some way for the sales and support of the work, by controlling the distribution of certificates that enable distributors to legitimately charge fees and copy owners to make copies. Since licenses are

themselves digital works, the same mechanisms give the creators control over distributors by charging for licenses and putting time limits on their validity.

[0346] This scenario is performed as follows: A creator purchases a digital distribution license that he will hand out to his distributors. He puts access requirements (such as a personal license) on the Copy and Transfer rights on the distribution license so that only he can copy or transfer it.

[0347] The creator also creates a digital work. He grants an Embed right and a Copy right, both of which require the distribution license to be exercised. He grants a Play right so that the work can be played by anyone. He may optionally add a Transfer or Loan right, so that end consumers can do some non-commercial exchange of the work among friends.

[0348] A distributor obtains the distribution license and a number of copies of the work. He makes copies for his customers, using his distribution license.

[0349] A customer buys and uses the work. He cannot make new copies because he lacks a distribution license.

[0350] This is a variation on the previous scenarios. A distributor can sell to anyone and anyone can sell additional copies, resulting in fees being paid back to the creator. However, only licensed distributors can add fees to be paid to themselves.

[0351] This scenario gives distributors the right to add fees to cover their own advertising and promotional costs, without making them be the sole suppliers. Their customers can also make copies, thus broadening the channel without diminishing their revenues. This is because distributors collect fees from copies of any copies that they originally sold. Only distributors can add fees.

[0352] This scenario is performed similarly to the previous ones. There are two key differences. (1) The creator only grants Embed rights for people who have a Distribution license.

NVA208575 1

This is done by putting a requirement for a distributor's license on the Embed right. Consequently, non-distributors cannot add their own fees. (2) The Distributor does not grant Extract rights, so that consumers cannot avoid paying fees to the Distributor if they make subsequent copies. Consequently, all subsequent copies result in fees paid to the Distributor and the Creator.

[0353] In this scenario, a distributor gets a fee for any copy he sells directly. However, if one of his customers sells further copies, he gets no further fee for those copies.

[0354] This scenario pays a distributor only for use of copies that he actually sold. This scenario is performed similarly to the previous ones. The key feature is that the distributor creates a shell which specifies fees to be paid to him. He puts Extract rights on the shell. When a consumer buys the work, he can extract away the distributor's shell. Copies made after that will not require fees to be paid to the distributor.

[0355] In another scenario, distributors sell to other distributors and fees are collected at each level. Every copy sold by any distributor--even several d-blocks down in the chain--results in a fee being paid back to all of the previous distributors.

[0356] This scenario is like a chain letter or value chain. Every contributor or distributor along the way obtains fees, and is thereby encouraged to promote the sale of copies of the digital work.

[0357] This scenario is performed similarly to the previous ones. The key feature is that the distributor creates a shell which specifies fees to be paid to him. He does not grant Extract rights on the shell. Consequently, all future copies that are made will result in fees paid to him.

[0358] In this scenario, distributors make money according to a distribution tree. The fee that they make depends on various parameters, such as time since their sale or the number of subsequent distributors.

[0359] This is a generalized version of the Distribution Tree scenario, in that it tries to vary the fee to account for the significance of the role of the distributor.

[0360] This scenario is similar to the previous one. The difference is that the fee specification on the distributor's shell has provisions for changes in prices. For example, there could be a fee schedule so that copies made after the passage of time will require lower fees to be paid to the distributor. Alternatively, the distributor could employ a "best-price" billing option, using any algorithm he chooses to determine the fee up to the maximum specified in the shell.

[0361] In this scenario, a first creator creates a work. It is distributed by a first distributor and purchased by a second creator. The second creator extracts a portion of the work and embeds in it a new work distributed by a second distributor. A consumer buys the new work from the second distributor. The first creator receives fees from every transaction; the first distributor receives fees only for his sale; the second creator and second distributor receive fees for the final sale.

[0362] This scenario shows how that flexible automatic arrangements can be set up to create automatic charging systems that mirror current practice. This scenario is analogous to when an author pays a fee to reuse a figure in some paper. In the most common case, a fee is paid to the creator or publisher, but not to the bookstore that sold the book.

[0363] The mechanisms for derived works are the same as those for distribution. In this scenario, several first creators create works. A second creator makes a selection of these, publishing a collection made up of the parts together with some new interstitial material. (For example, the digital work could be a selection of music or a selection of readings.) The second creator wants to continue to allow some of the selected works to be extractable, but not the interstitial material.

[0364] This scenario deals with fine grained control of the rights and fees for reuse.

[0365] This scenario is performed as follows: The first creators create their original

NVA208575 1

works. If they grant extraction and embedding rights, then the second creator can include them in a larger collected work. The second creator creates the interstitial material. He does grant an Extract right on the interstitial material. He grants Extract rights on a subset of the reused material. A consumer of the collection can only extract portions that have that right. Fees are automatically collected for all parts of the collection.

[0366] Commercial libraries buy works with the right to loan. They limit the loan period and charge their own fees for use. This scenario deals with fees for loaning rather than fees for making copies. The fees are collected by the same automatic mechanisms.

[0367] The mechanisms are the same as previous scenarios except that the fees are associated with the Loan usage right rather than the Copy usage right.

[0368] A creator believes that if people try his work that they will want to buy it or use it. Consumers of his work can copy the work for free, and play (or execute) a limited version of the work for free, and can play or use the full featured version for a fee.

[0369] This scenario is performed as follows: The creator creates a digital work and grants various rights and fees. The creator grants Copy and Embed rights without a fee, in order to ensure widespread distribution of the work. Another of the rights is a limited play right with little or no fee attached. For example, this right may be for playing only a portion of the work. The play right can have various restrictions on its use. It could have a ticket that limits the number of times it is used. It could have internal restrictions that limit its functionality. It could have time restrictions that invalidate the right after a period of time or a period of use. Different fees could be associated with other versions of the Play right.

[0370] A consumer buys a digital work together with an agreement that he can upgrade to a new version at a later date for a modest fee, much less than the usual purchase price. When the new version becomes available, he goes to a qualified vendor to make the transaction.

[0371] This scenario deals with a common situation in computer software. It shows how

NVA208575 1

a purchase may include future "rights." Two important features of the scenario are that the transaction must take place at a qualified vendor, and that the transaction can be done only once per copy of the digital work purchased.

[0372] This scenario is performed as follows: The creator creates a digital work, an upgrade ticket, and a distribution license. The upgrade ticket uses the a generic ticket agent that comes with repositories. As usual, the distribution license does not have Copy or Transfer rights. He distributes a bundled copies of the work and the ticket to his distributors as well as distribution licenses.

[0373] The distributor sells the old bundled work and ticket to customers. The customer extracts the work and the ticket. He uses the work according to the agreements until the new version becomes available.

[0374] When the new work is ready, the creator gives it to distributors. The new work has a free right to copy from a distributor if a ticket is available.

[0375] The consumer goes to distributors and arranges to copy the work. The transaction offers the ticket. The distributor's repository punches the ticket and copies the new version to the consumers repository.

[0376] The consumer can now use the new version of the work. A consumer buys a digital work together with an agreement that he can upgrade to a new version at a later date for a modest fee, much less than the usual purchase price. When the new version becomes available, he goes to anyone who has the upgraded version and makes the transaction.

[0377] This scenario is like the previous one in that the transaction can only be done once per copy of the digital work purchased, but the transaction can be accomplished without the need to connect to a licensed vendor.

[0378] This scenario is similar to the previous one except that the Copy right on the new

NVA208575 1

work does not require a distribution license. The consumer can upgrade from any repository having the new version. He cannot upgrade more than once because the ticket cannot work after it has been punched. If desired, the repository can record the upgrade transaction by posting a zero cost bill to alert the creator that the upgrade has taken place.

[0379] A consumer buys a digital work and wants to make a few ephemeral copies. For example, he may want to print out a paper copy of part of a digital newspaper, or he may want to make a (first generation) analog cassette tape for playing in his car. He buys the digital work together with a ticket required for printing rights.

[0380] This scenario is like the common practice of people making cassette tapes to play in their car. If a publisher permits the making of cassette tapes, there is nothing to prevent a consumer from further copying the tapes. However, since the tapes are "analog copies," there is a noticeable quality loss with subsequent generations. The new contribution of the present invention is the use of tickets in the access controls for the making of the analog copies.

[0381] This scenario is performed as follows: The creator sells a work together with limited printing rights. The printing rights specify the kind of printer (e.g., a kind of cassette recorder or a kind of desktop paper printer) and also the kind of ticket required. The creator either bundles a limited number of tickets or sells them separately. If the tickets use the generic ticket agent, the consumer with the tickets can exercise the right at his convenience.

[0382] Professors in a business school want to put together course books of readings selected from scenario studies from various sources. The bookstore wants to be able to print the books from digital masters, without negotiating for and waiting for approval of printing of each of the scenarios. The copyright holders of the scenarios want to be sure that they are paid for every copy of their work that is printed.

[0383] On many college campuses, the hassle of obtaining copy clearances in a timely way has greatly reduced the viability of preparing course books. Print shops have become much more cautious about copying works in the absence of documented permission.

NVA208575 1

[0384] Demand Publishing is performed as follows: the creator sells a work together with printing rights for a fee. There can be rights to copy (distribute) the work between bookstore repositories, with or without fee. The printing rights specify the kind of printer. Whenever a bookstore prints one of the works (either standalone or embedded in a collection), the fee is credited to the creator automatically. To discourage unauthorized copying of the print outs, it would be possible for the printer to print tracer messages discretely on the pages identifying the printing transaction, the copy number, and any other identifying information. The tracer information could be secretly embedded in the text itself (encoded in the grey scale) or hidden in some other way.

[0385] A consumer does not know what music to purchase until he decides whether he likes it. He would like to be able to take it home and listen to it, and then decide whether to purchase. Furthermore, he would like the flexibility of paying less if he listens to it very infrequently.

[0386] This scenario just uses the capability of the approach to have multiple versions of a right on a digital work. Each version of the right has its own billing scheme. In this scenario, the creator of the work can offer the Copy right without fee, and defer billing to the exercise of the Play right. One version of the play right would allow a limited performance without fee--a right to "demo". Another version of the right could have a metered rate, of say \$0.25 per hour of play. Another version could have a fee of \$15.00 for the first play, but no fee for further playing. When the consumer exercises a play right, he specifies which version of the right is being selected and is billed accordingly.

[0387] A designer of type fonts invests several months in the design of special fonts. The most common way of obtaining revenue for this work is to sell copies of the fonts to publishers for unlimited use over unlimited periods of time. A font designer would like to charge a rate that reflects the amount that the font is used.

[0388] This scenario is performed as follows: the font designer creates a font as a digital

NVA208575.1

work. He creates versions of the Play right that bill either for metered use or "per-use". Each version of the play right would require that the player (a print layout program) be of an approved category. The font designer assigns appropriate fees to exercise the Copy right. When a publisher client wants to use a font, he includes it as input to a layout program, and is billed automatically for its use. In this way, a publisher who makes little use of a font pays less than one who uses it a lot.

[0389] Online information retrieval services typically charge for access in a way that most clients find unpredictable and uncorrelated to value or information use. The fee depends on which databases are open, dial-up connect time, how long the searches require, and which articles are printed out. There are no provisions for extracting articles or photographs, no method for paying to reuse information in new works, no distinction between having the terminal sit idly versus actively searching for data, no distinction between reading articles on the screen and doing nothing, and higher rates per search when the centralized facility is busy and slow servicing other clients. Articles can not be offloaded to the client's machine for off-site search and printing. To offer such billing or the expanded services, the service company would need a secure way to account for and bill for how information is used.

[0390] This scenario is performed as follows: The information service bundles its database as files in a repository. The information services company assigns different fees for different rights on the information files. For example, there could be a fee for copying a search database or a source file and a different fee for printing. These fees would be in addition to fees assigned by the original creator for the services. The fees for using information would be different for using them on the information service company's computers or the client's computers. This billing distinction would be controlled by having different versions of the rights, where the version for use on the service company's computer requires a digital certificate held locally. Fees for copying or printing files would be handled in the usual way, by assigning fees to exercising those rights. The distinction between searching and viewing information would be made by having different "players" for the different functions. This distinction would be maintained on the client's computers as well as the service computers. Articles could be extracted for reuse under the control of Extract and Embed rights. Thus, if a client extracts part of an

article or photograph, and then sells copies of a new digital work incorporating it, fees could automatically be collected both by the information service and earlier creators and distributors of the digital work. In this way, the information retrieval service could both offer a wider selection of services and billing that more accurately reflects the client's use of the information.

[0391] In the simplest scenario, when a user wants to print a digital document he issues a print command to the user interface. If the document has the appropriate rights and the conditions are satisfied, the user agrees to the fee and the document is printed. In other cases, the printer may be on a remote repository and it is convenient to spool the printing to a later time. This leads to several issues. The user requesting the printing wants to be sure that he is not billed for the printing until the document is actually printed. Restated, if he is billed at the time the print job is spooled but the job is canceled before printing is done, he does not want to pay. Another issue is that when spooling is permitted, there are now two times at which rights, conditions and fees could be checked: the time at which a print job is spooled and the time at which a print is made. As with all usage rights, it is possible to have rights that expire and to have rights whose fee depends on various conditions. What is needed is a means to check rights and conditions at the time that printing is actually done.

[0392] This scenario is performed as follows: A printing repository is a repository with the usual repository characteristics plus the hardware and software to enable printing. Suppose that a user logs into a home repository and wants to spool print jobs for a digital work at a remote printing repository. The user interface for this could treat this as a request to "spool" prints. Underneath this "spooling" request, however, are standard rights and requests. To support such requests, the creator of the work provides a Copy right, which can be used to copy the work to a printing repository. In the default case, this Copy right would have no fees associated for making the copy. However, the Next-Set-Of-Rights for the copy would only include the Print rights, with the usual fees for each variation of printing. This version of the Copy right could be called the "print spooling" version of the Copy right. The user's "spool request" is implemented as a Copy transaction to put a copy of the work on the printing repository, followed by Print transactions to create the prints of the work. In this way, the user is only billed for printing that is actually done. Furthermore, the rights, conditions and fees for printing the work are determined when the work

NVA208575.1

is about to be printed.

[0393] Thus, a system for enforcing the usage rights of digital works is disclosed. While the embodiments disclosed herein are preferred, it will be appreciated from this teaching that various alternative, modifications, variations or improvements therein may be made by those skilled in the art, which are intended to be encompassed by the appended claims which define the invention.

GLOSSARY

[0394] Authorization Repository - A special type of repository which provides an authorization service. An authorization may be specified by a usage right. The authorization must be obtained before manner of use specified by the right may be exercised.

[0395] Billing Clearinghouse - A financial institution or the like whose purpose is to reconcile billing information received from credit servers. The billing clearinghouse may generate bills to users or alternatively, credit and debit accounts involved in the commercial transactions.

[0396] Billing Transactions - The protocol used by which a repository reports billing information to a credit server.

[0397] Clearinghouse Transactions - The protocol used between a credit server and a clearinghouse.

[0398] Composite Digital Work - A digital work comprised of distinguishable parts. Each of the distinguishable parts is itself a digital work which may have associated usage rights. Content - The digital information that is to be rendered of digital work.

[0399] Copy Owner - A term which refers to the party who owns a digital work stored in a repository. In the typical case, this party has purchased various rights to the document for

NVA208575 1

printing, viewing, transferring, or other specific uses.

[0400] Creator - A term which refers to a party who produces a digital work.

[0401] Credit Server - A device which collects and reports billing information for a repository. In many implementations, this could be built as part of a repository. It requires a means for periodically communicating with a billing clearinghouse.

[0402] Description Tree - A structure which describes the location of content and the usage rights and usage fees for a digital work. A description tree is comprised of description blocks. Each description block corresponds to a digital work or to an interest (typically a revenue bearing interest) in a digital work.

[0403] Digital Work (Work) - Digital content with any associated usage rights. Such digital content may represent music, a magazine or book, or a multimedia composition.
Distributor - A term which refers to a party who legitimately obtains a copy of a digital work and offers it for sale.

[0404] Identification (Digital) Certificate - A signed digital message that attests to the identity of the possessor. Typically, digital certificates are encrypted in the private key of a well-known master repository.

[0405] Master Repository - A special type of repository which issues identification certificates and distributes lists of repositories whose integrity have been compromised and which should be denied access to digital works (referred to as repository "hotlists".)

[0406] Public Key Encryption - An encryption technique used for secure transmission of messages on a communication channel. Key pairs are used for the encryption and decryption of messages. Typically one key is referred to as the public key and the other is the private key. The keys are inverses of each other from the perspective of encryption. Restated, a digital work that is encrypted by one key in the pair can be decrypted only by the other.

[0407] Registration Transactions - The protocol used between repositories to establish a trusted session.

[0408] Rendering Repository - A special type of repository which is typically coupled to a rendering system. The rendering repository will typically be embodied within the secure boundaries of a rendering system.

[0409] Rendering System - The combination of a rendering repository and a rendering device. Examples of a rendering systems include printing systems, display systems, general purpose computer systems, video systems or audio systems.

[0410] Repository - Conceptually a set of functional specifications defining core functionality in the support of usage rights. A repository is a trusted system in that it maintains physical, communications and behavioral integrity.

[0411] Requester Mode - A mode of a repository where it is requesting access to a digital work.

[0412] Revenue Owners - A term which refers to the parties that maintain an interest in collecting fees for document use or who stand to lose revenue if illegitimate copies of the digital work are made.

[0413] Server Mode - A mode of a repository where it is processing an incoming request to access a digital work.

[0414] Sell Description Block - A special type of description block designating an interest in a digital work, but which does not add content. This will typically be added by a distributor of a digital work to add their fees.

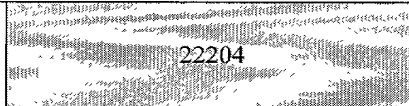
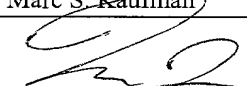
Transactions - A term used to refer to the protocols by which repositories communicate.

[0415] Usage Fees - A fee charged to a requester for access to a digital work. Usage fees are specified within the usage rights language.

[0416] Usage Rights - An indication of the manner of use by which a digital work may be used or distributed, as well as any conditions on which manner of use is premised.

[0417] Usage Transactions - A set of protocols by which repositories communicate in the exercise of a usage rights. Each usage right has it's own transaction steps.

10/015951
12/17/01

UTILITY PATENT APPLICATION TRANSMITTAL <i>(Only for new nonprovisional applications under 37 CFR 1.53(b))</i>		Attorney Docket No. 111325-88	
		First Inventor Mark J. Stefik et al.	
		Title Digital Works Having Usage Rights And Method For Creating The Same	
		Express Mail Label No. _____	
APPLICATION ELEMENTS See MPEP chapter 600 concerning utility patent application contents.		ADDRESS TO: Commissioner for Patents Box Patent Application Washington, DC 20231	
1. ☐ Fee Transmittal Form (e.g., PTO/SB/17) <i>(Submit an original and a duplicate for fee processing)</i> 2. ☐ Applicant claims small entity status. See 37 CFR 1.27. 3. ☒ Specification [Total Pages 88] <i>(preferred arrangement set forth below)</i> - Descriptive title of the invention - Cross Reference to Related Applications <i>(if applicable)</i> - Statement Regarding Fed sponsored R & D <i>(if applicable)</i> - Reference to sequence listing, a table, or a computer program listing appendix <i>(if applicable)</i> - Background of the Invention - Brief Summary of the Invention - Brief Description of the Drawings <i>(if filed)</i> - Detailed Description - Claim(s) - Abstract of the Disclosure 4. ☒ Drawing(s) (35 U.S.C. 113) [Total Sheets 13] 5. Oath or Declaration [Total Pages ☐] a. ☐ Newly executed (original or copy) b. ☐ Copy from a prior application (37 CFR 1.63(d)) <i>(for continuation/divisional with Box 18 completed)</i> i. ☐ DELETION OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b) 6. ☐ Application Data Sheet. See 37 CFR 1.76		7. ☐ CD-ROM or CD-R in duplicate, large table or Computer Program (Appendix) 8. Nucleotide and/or Amino Acid Sequence Submission <i>(if applicable, all necessary)</i> a. ☐ Computer Readable Form (CRF) b. Specification Sequence Listing on: i. ☐ CD-ROM or CD-R (2 copies; or ii. ☐ paper c. ☐ Statements verifying identity of above copies	
ACCOMPANYING APPLICATION PARTS			
9. ☐ Assignment Papers (cover sheet & document(s)) 10. ☐ 37 CFR 3.73(b) Statement ☐ Power of Attorney <i>(when there is an assignee)</i> 11. ☐ English Translation Document <i>(if applicable)</i> 12. ☐ Information Disclosure Statement (IDS)/PTO-1449 ☐ Copies of IDS Citations 13. ☐ Preliminary Amendment 14. ☒ Return Receipt Postcard (MPEP 503) <i>(Should be specifically itemized)</i> 15. ☐ Certified Copy of Priority Document(s) <i>(if foreign priority is claimed)</i> 16. ☐ Nonpublication request under 35 U.S.C. 122(b)(2)(B)(i). Applicant must attach form PTO/SB/35 or its equivalent. 17. ☐ Other: _____			
18. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment, or in an Application Data Sheet under 37 CFR 1.76. ☒ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: <u>09/778,001 filed Feb. 7, 2001, which is a Divisional of Serial No. 08/967,084, filed on Nov. 10, 1997, now U.S. Patent No. 6,236,971, which is a Continuation of Serial No. 08/344,760 filed Nov. 23, 1994, now abandoned.</u> Prior application information: _____ Examiner _____ Group / Art Unit. _____			
For CONTINUATION OR DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 5b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation <u>can only</u> be relied upon when a portion has been inadvertently omitted from the submitted application parts.			
19. CORRESPONDENCE ADDRESS			
☒ Customer Number or Bar Code Label		or ☐ Correspondence address below	
 22204			
Name	Marc S. Kaufman		
Address			
City	State	Zip Code	
Country	Telephone	Fax	
Name (Print/Type)	Marc S. Kaufman	Registration No. (Attorney/Agent)	35,212
Signature		Date	December 17, 2001

What is claimed is:

1. A digital work adapted to be distributed within a system for controlling at least one of the distribution and use of digital works, said digital work comprising:

digital content representing a portion of a digital work suitable for being rendered by a rendering device; and
usage rights associated with said digital content, said usage rights specifying a manner of use indicating one or more stated purposes for which the digital work can be at least one of used and distributed by an authorized party.

2. The digital work as recited in claim 1, wherein said usage rights further specify status information indicating the status of the digital work.

3. The digital work as recited in claim 1, wherein said usage rights further specify a usage fee associated with exercise of the manner of use, said usage fee comprising a fee type and fee parameters.

4. The digital work as recited in claim 3 wherein said fee type is a metered use fee and said fee parameters comprise a fee unit and a time unit.

5. The digital work as recited in claim 3 wherein said fee type is a per use fee and said fee parameter comprises a fee unit.

6. The digital work as recited in claim 3 wherein said fee type is a best price fee and said fee parameters comprise an identifier to identify said best price.

7. The digital work as recited in claim 3 wherein said fee type is a scheduled fee and said fee parameters comprise time units and fee units.

8. The digital work as recited in claim 3 wherein said fee type is a mark-up fee and said fee parameters comprise a mark-up percentage.

NVA208575 1

9. The digital work as recited in claim 1 wherein said digital content and said usage rights are stored on the same physical device.

10. The digital work as recited in claim 1, wherein said digital content and said usage rights are stored on different physical devices.

11. A method for creating a digital work to be used in a system for reporting fees associated with at least one of distribution and use of the digital work, said method comprising:

a) obtaining digital content representing a portion of a digital work suitable for being rendered by a rendering device;

b) associating usage rights with the digital content, the usage rights specifying a manner of use indicating one or more stated purposes for which the digital work can be at least one of distributed; and

c) storing the digital content and the usage rights as the digital work.

12. A method as recited in claim 11, wherein said storing step comprises storing the digital content and the usage rights on the same physical device.

13. A method as recited in claim 11, wherein said storing step comprises storing the digital content and the usage rights on different physical devices.

14. A method as recited in claim 11, wherein said usage rights further specify a usage fee associated with exercise of the manner of use.

15. The method as recited in claim 14, wherein said usage fee comprises usage fee information including a fee type and fee parameters.

16. The method as recited in claim 15 wherein said fee type is a metered use fee and said fee parameters comprise a fee unit and a time unit.

17. The method as recited in claim 15 wherein said fee type is a per use fee and said fee

NVA208575.1

parameters comprise a fee unit.

18. The method as recited in claim 15 wherein said fee type is a best price fee and said fee parameters comprise an identifier to identify the best price.

19. The method as recited in claim 15 wherein said fee type is a scheduled fee and said fee parameters comprise time units and fee units.

20. The method as recited in claim 17 wherein said fee type is a mark-up fee and said fee parameters comprise a mark-up percentage.

21. A digital work adapted to be distributed within a system for controlling at least one of the distribution and use of digital works, said system having a mechanism for reporting status information relating to digital works, said digital work comprising:

digital content representing a portion of a digital work suitable for being rendered by a rendering device; and

usage rights associated with said digital content, said usage rights specifying a manner of use indicating one or more stated purposes for which the digital work can be at least one of used and distributed, said usage rights also including state information indicating the extent of the manner of use exercised by an authorized user.

22. A digital work as recited in claim 21, wherein the state information comprises information indicating the number of copies of the digital work that are in use.

23. A digital work as recited in claim 21, wherein the state information comprises information indicating the number of time units for which the digital work can be used.

24. A digital work as recited in claim 21, wherein the state information comprises information indicating the revenue owner of the digital work.

25. A digital work as recited in claim 21, wherein the state information comprises information indicating the publication date of the digital work.
26. A digital work as recited in claim 21, wherein the state information comprises information indicating the history of at least one of the use and distribution of the digital work.
27. A method for creating a digital work to be used in a system for reporting fees associated with at least one of distribution and use of the digital work, said method comprising:
- a) obtaining digital content representing a portion of a digital work suitable for being rendered by a rendering device;
 - b) associating usage rights with the digital content, the usage rights specifying a manner of use indicating one or more stated purposes for which the digital work can be at least one of distributed, said usage rights also including state information indicating the extent of the manner of use exercised by an authorized user; and
 - c) storing the digital content and the usage rights as the digital work.
28. A method as recited in claim 27, wherein said storing step comprises storing the digital content and the usage rights on the same physical device.
29. A method as recited in claim 27, wherein said storing step comprises storing the digital content and the usage rights on different physical devices.
30. A method as recited in claim 27, wherein the state information comprises information indicating the number of copies of the digital work that are in use.
31. A method as recited in claim 27, wherein the state information comprises information indicating the number of time units for which the digital work can be used.
32. A method as recited in claim 27, wherein the state information comprises information indicating the revenue owner of the digital work.

33. A method as recited in claim 27, wherein the state information comprises information indicating the publication date of the digital work.

34. A method as recited in claim 27, wherein the state information comprises information indicating the history of at least one of the use and distribution of the digital work.

Figure 1

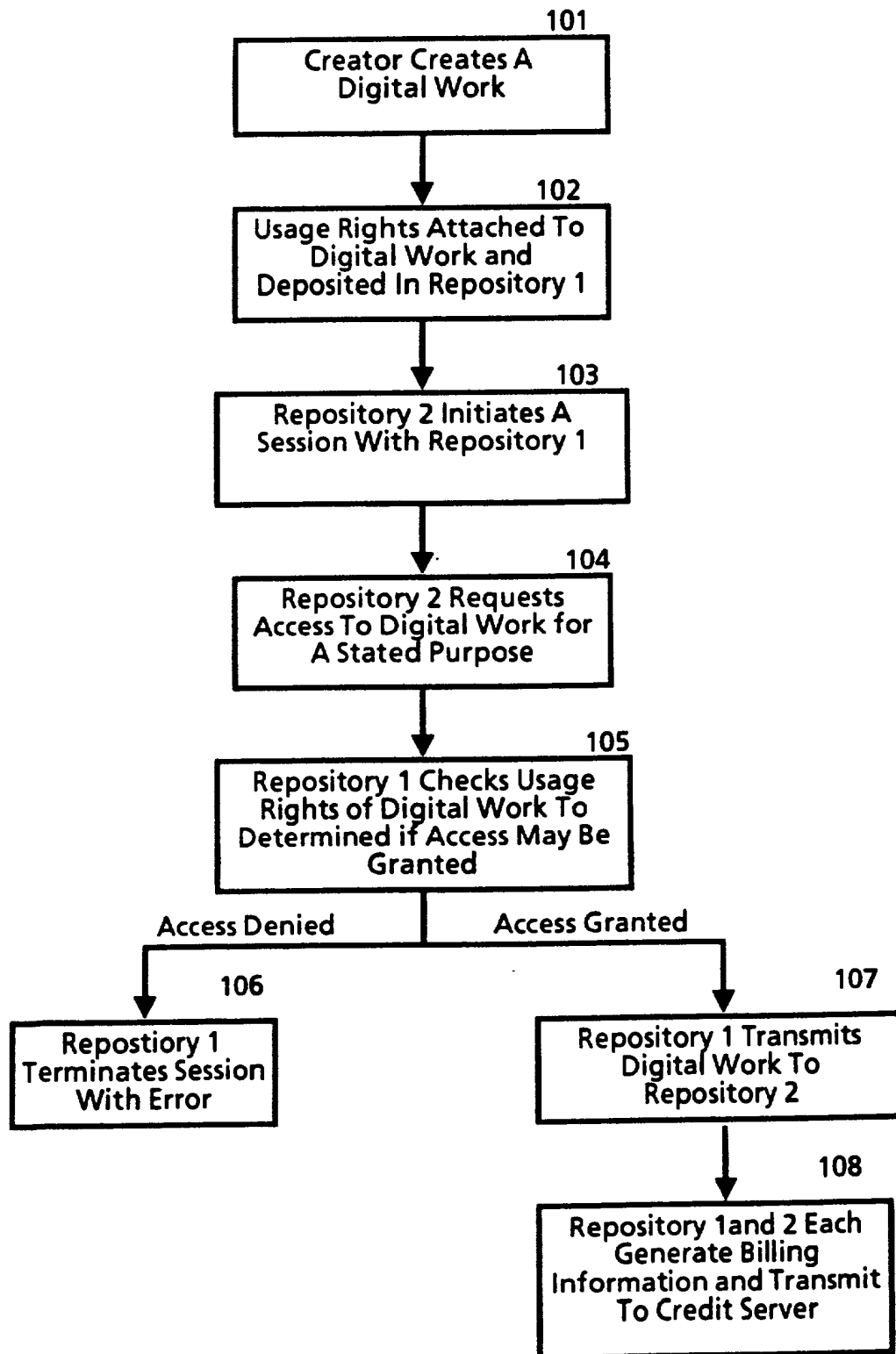


Figure 2

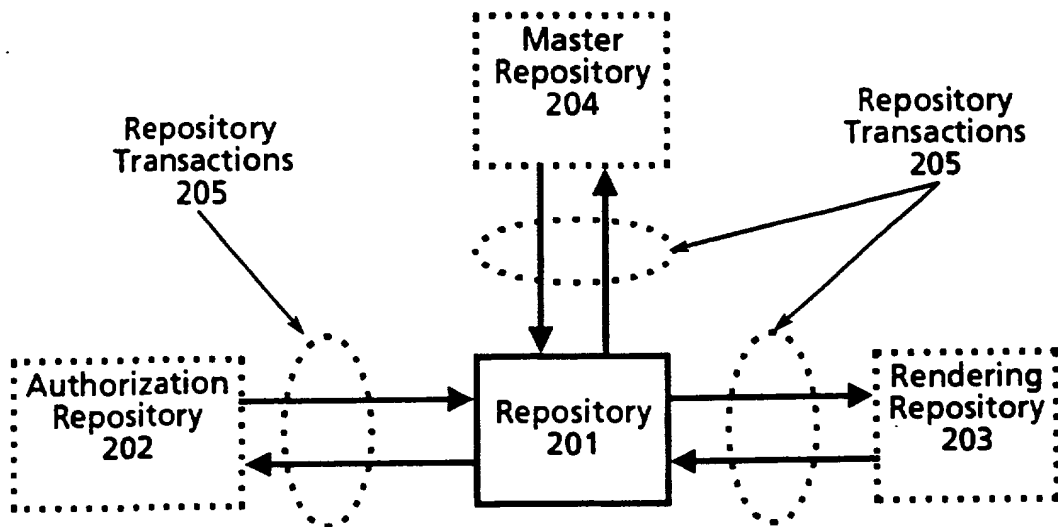


Figure 3

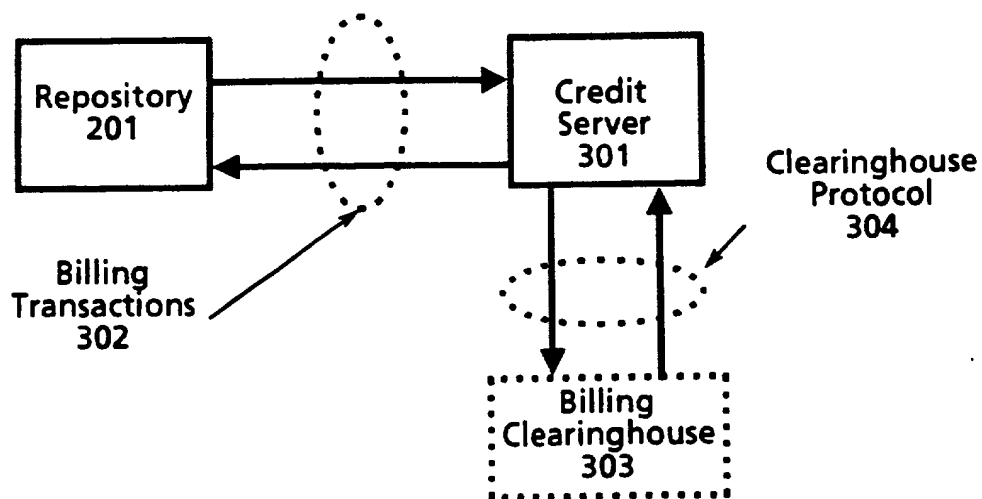


Figure 4a

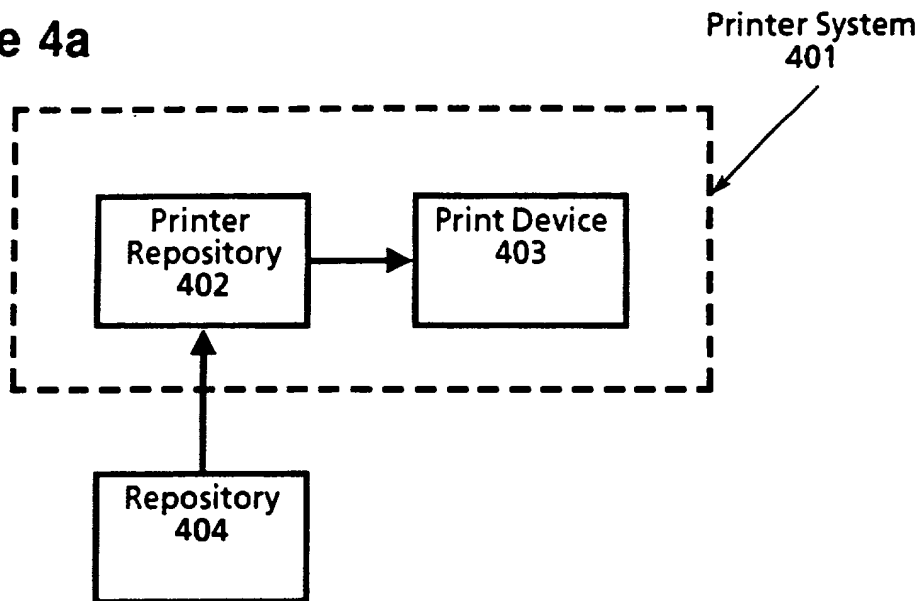
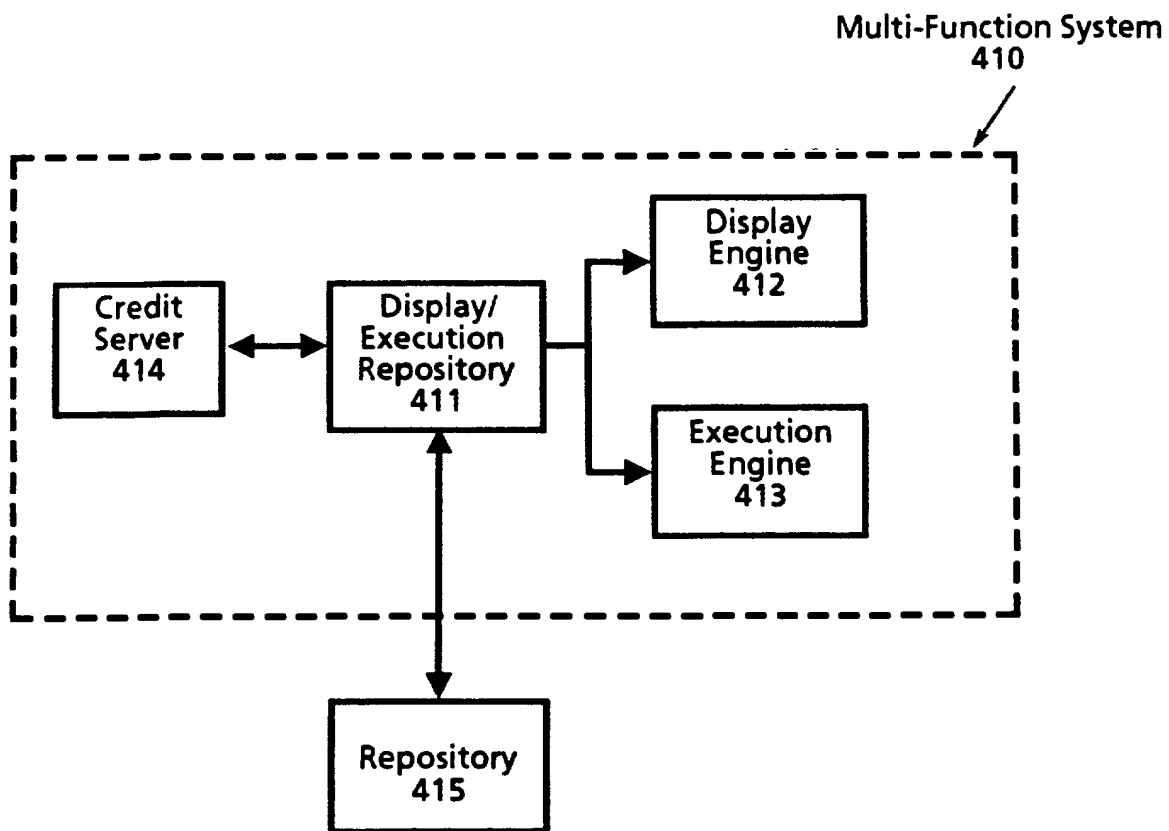


Figure 4b



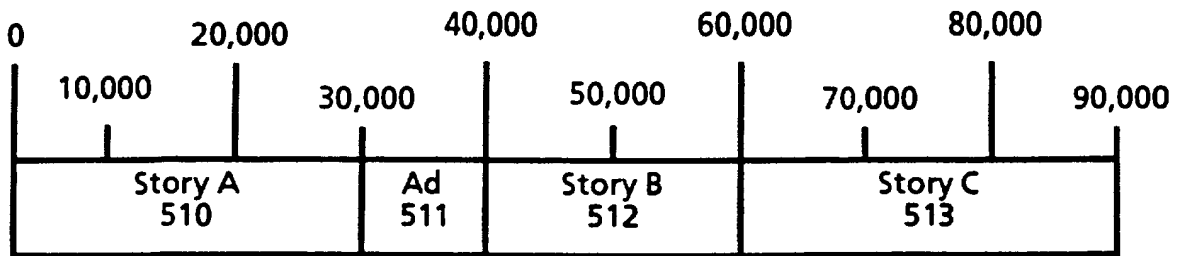


Figure 5

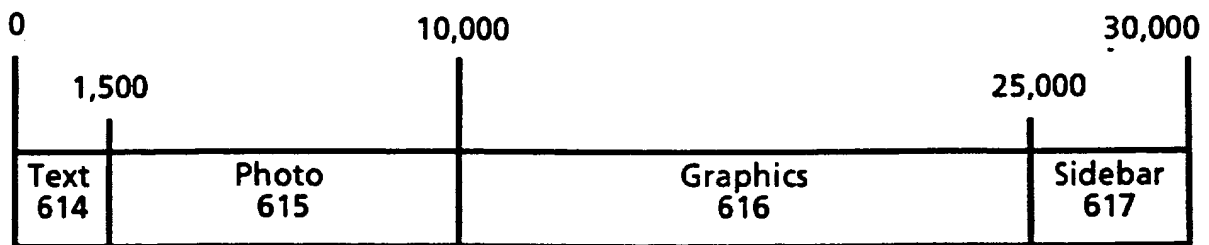


Figure 6

Figure 7

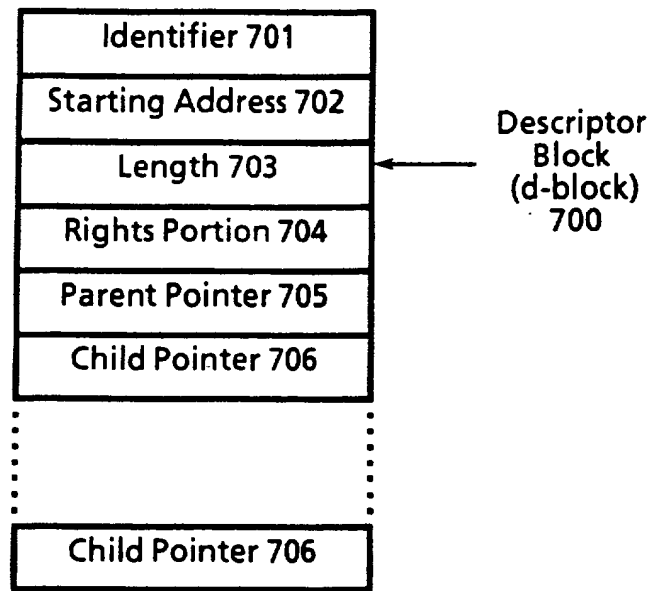


Figure 8

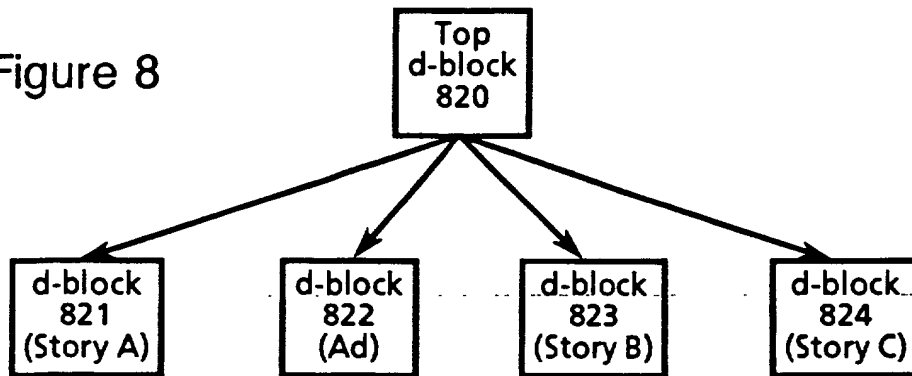


Figure 9

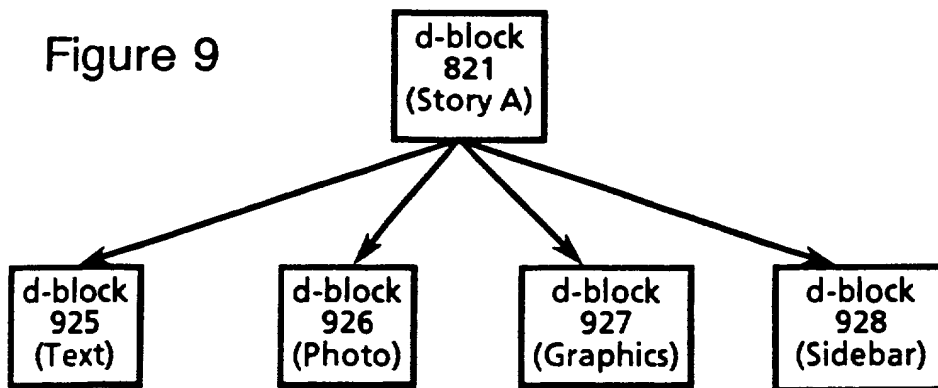


Figure 10

Right Code 1050	Status Information 1052

Figure 14

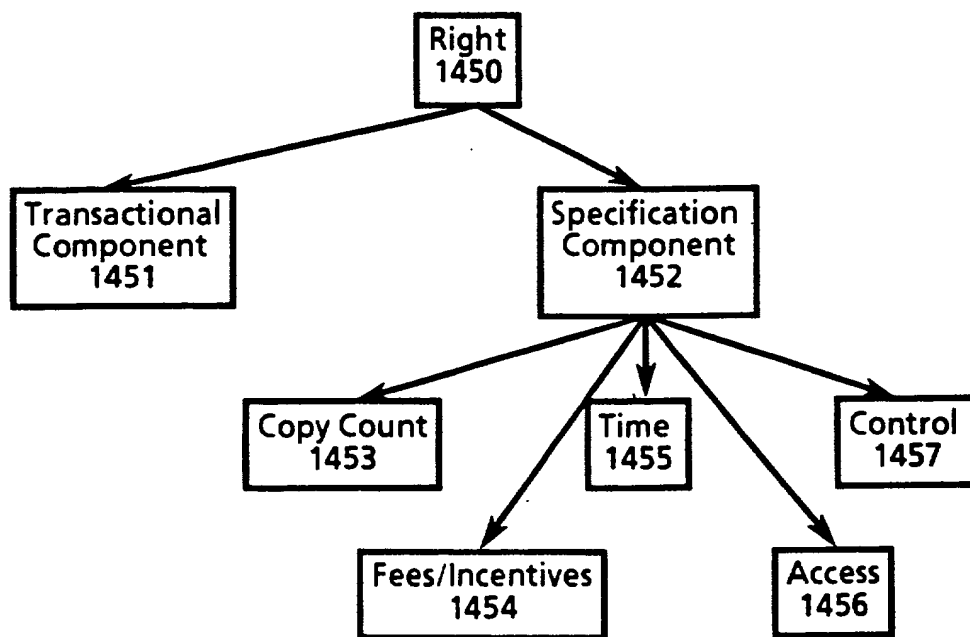


Figure 11

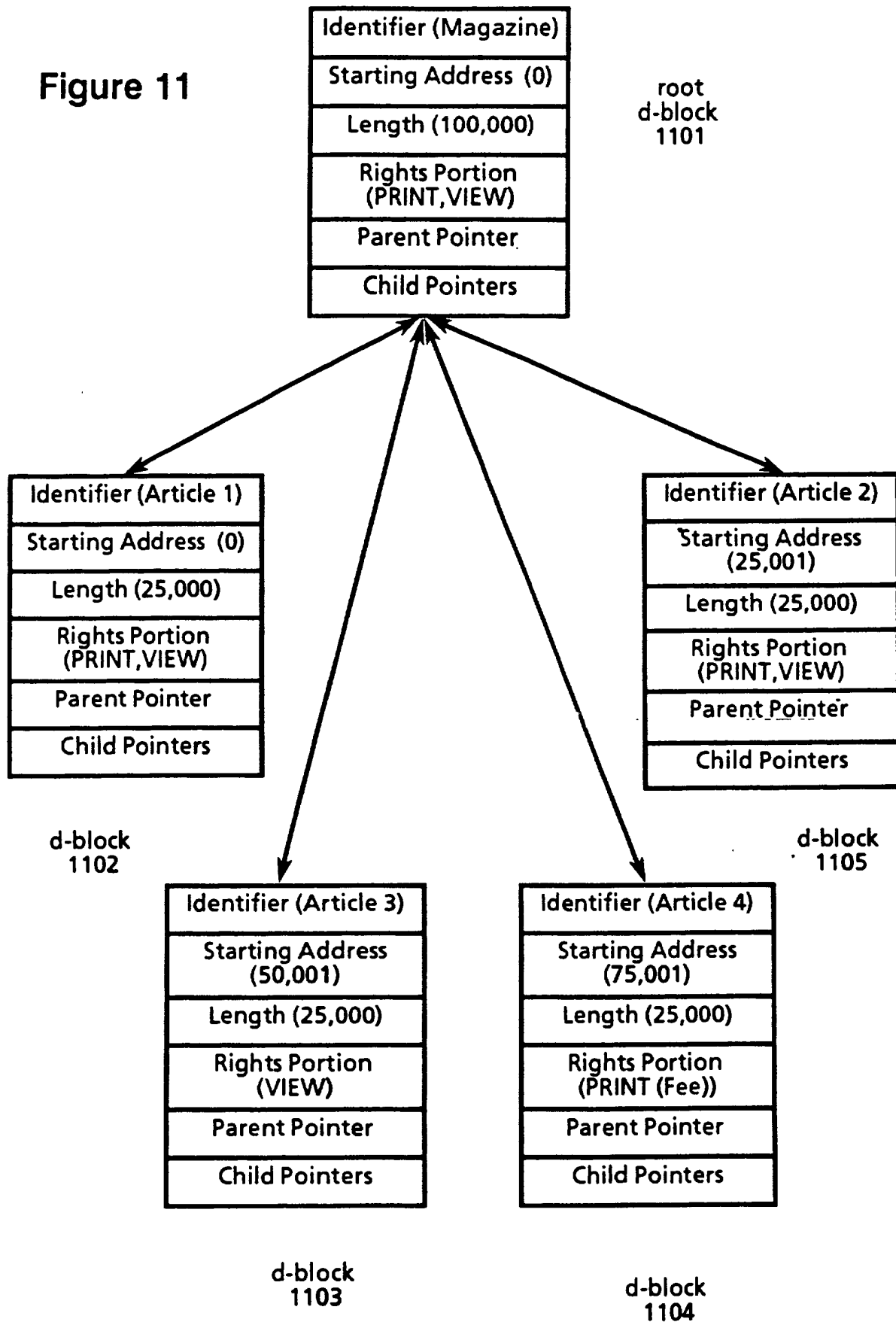


Figure 12

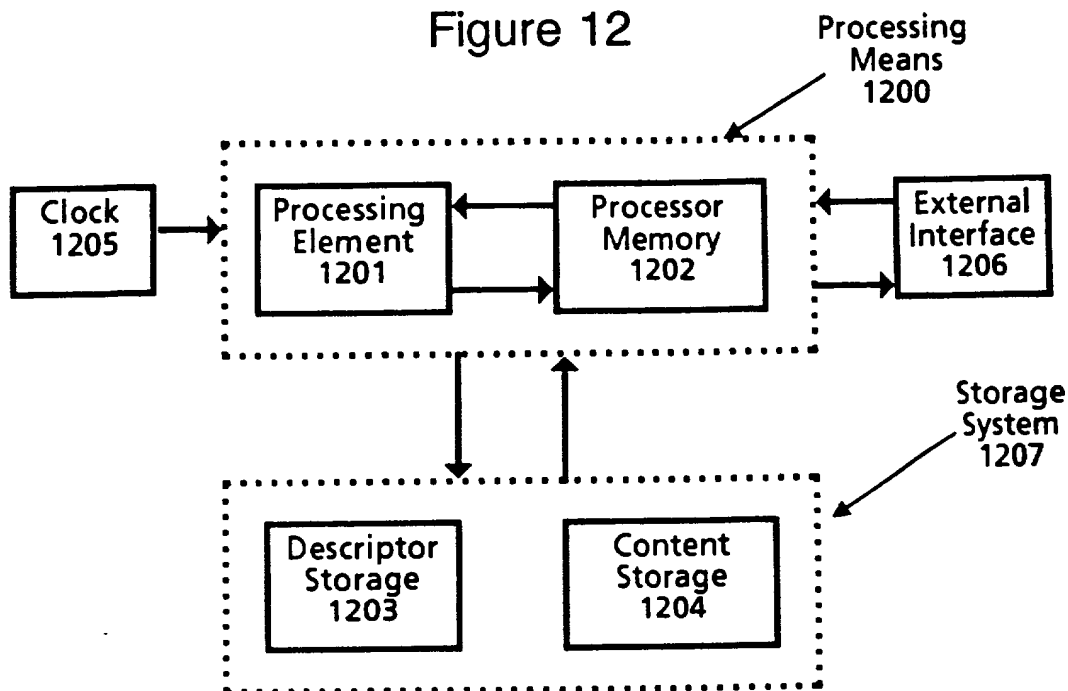


Figure 13

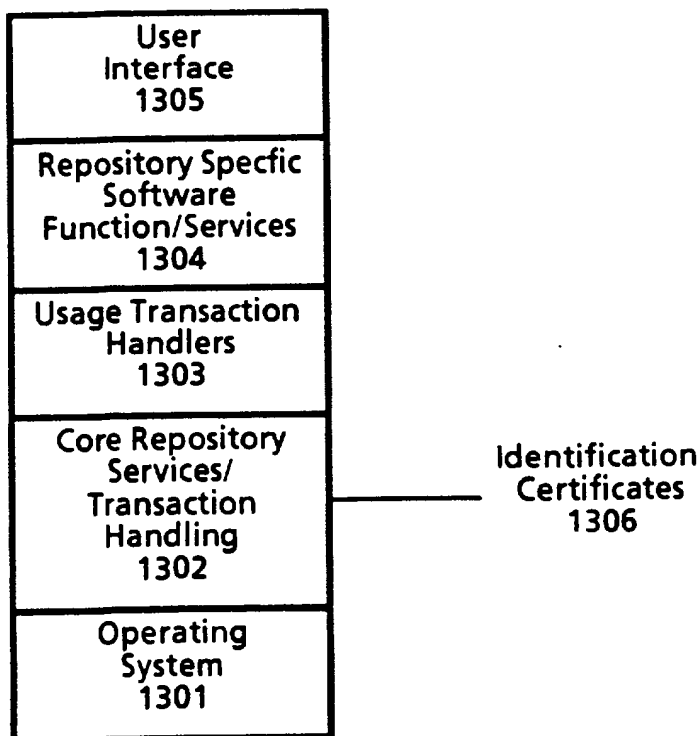


FIGURE 15

1501 ~ Digital Work Rights := (Rights*)

1502 ~ Right := (Right-Code {Copy-Count} {Control-Spec} {Time-Spec} {Access-Spec} {Fee-Spec})

1503 ~ Right-Code := Render-Code | Transport-Code | File-Management-Code | Derivative-Works-Code | Configuration-Code

1504 ~ Render-Code := [Play : {Player: Player-ID} | Print: {Printer: Printer-ID}]

1505 ~ Transport-Code := [Copy | Transfer | Loan {Remaining-Rights: Next-Set-of-Rights}] { (Next-Copy-Rights: Next-Set-of-Rights) }

1506 ~ File-Management-Code := Backup {Back-Up-Copy-Rights: Next-Set-of-Rights} | Restore | Delete | Folder | Directory {Name: Hide-Local | Hide-Remote} {Parts: Hide-Local | Hide-Remote}

1507 ~ Derivative-Works-Code := [Extract | Embed | Edit {Process: Process-ID}] { (Next-Copy-Rights: Next-Set-of-Rights) }

1508 ~ Configuration-Code := Install | Uninstall

1509 ~ Next-Set-of-Rights := { (Add: Set-Of-Rights) } { (Delete: Set-Of-Rights) } { (Replace: Set-Of-Rights) } { (Keep: Set-Of-Rights) }

1510 ~ Copy-Count := (Copies: positive-integer | 0 | Unlimited)

1511 ~ Control-Spec := (Control: {Restrictable | Unrestrictable} {Unchargeable | Chargeable})

1512 ~ Time-Spec := { (Fixed-Interval | Sliding-Interval | Meter-Time) Until: Expiration-Date }

1513 ~ Fixed-Interval := From: Start-Time

1514 ~ Sliding-Interval := Interval: Use-Duration

1515 ~ Meter-Time := Time-Remaining: Remaining-Use

1516 ~ Access-Spec := { (SC: Security-Class) {Authorization: Authorization-ID*} {Other-Authorization: Authorization-ID*} {Ticket: Ticket-ID} }

1517 ~ Fee-Spec := {Scheduled-Discount} Regular-Fee-Spec | Scheduled-Fee-Spec | Markup-Spec

1518 ~ Scheduled-Discount := Scheduled-Discount: (Scheduled-Discount: (Time-Spec Percentage)*)

1519 ~ Regular-Fee-Spec := { (Fee: | Incentive:) [Per-Use-Spec | Metered-Rate-Spec | Best-Price-Spec | Call-For-Price-Spec] {Min: Money-Unit Per: Time-Spec} {Max: Money-Unit Per: Time-Spec} To: Account-ID }

1520 ~ Per-Use-Spec := Per-Use: Money-unit

1521 ~ Metered-Rate-Spec := Metered: Money-Unit Per: Time-Spec

1522 ~ Best-Price-Spec := Best-Price: Money-unit Max: Money-unit

1523 ~ Call-For-Price-Spec := Call-For-Price

1524 ~ Scheduled-Fee-Spec := (Schedule: (Time-Spec Regular-Fee-Spec)*)

1525 ~ Markup-Spec := Markup: percentage To: Account-ID

Figure 16

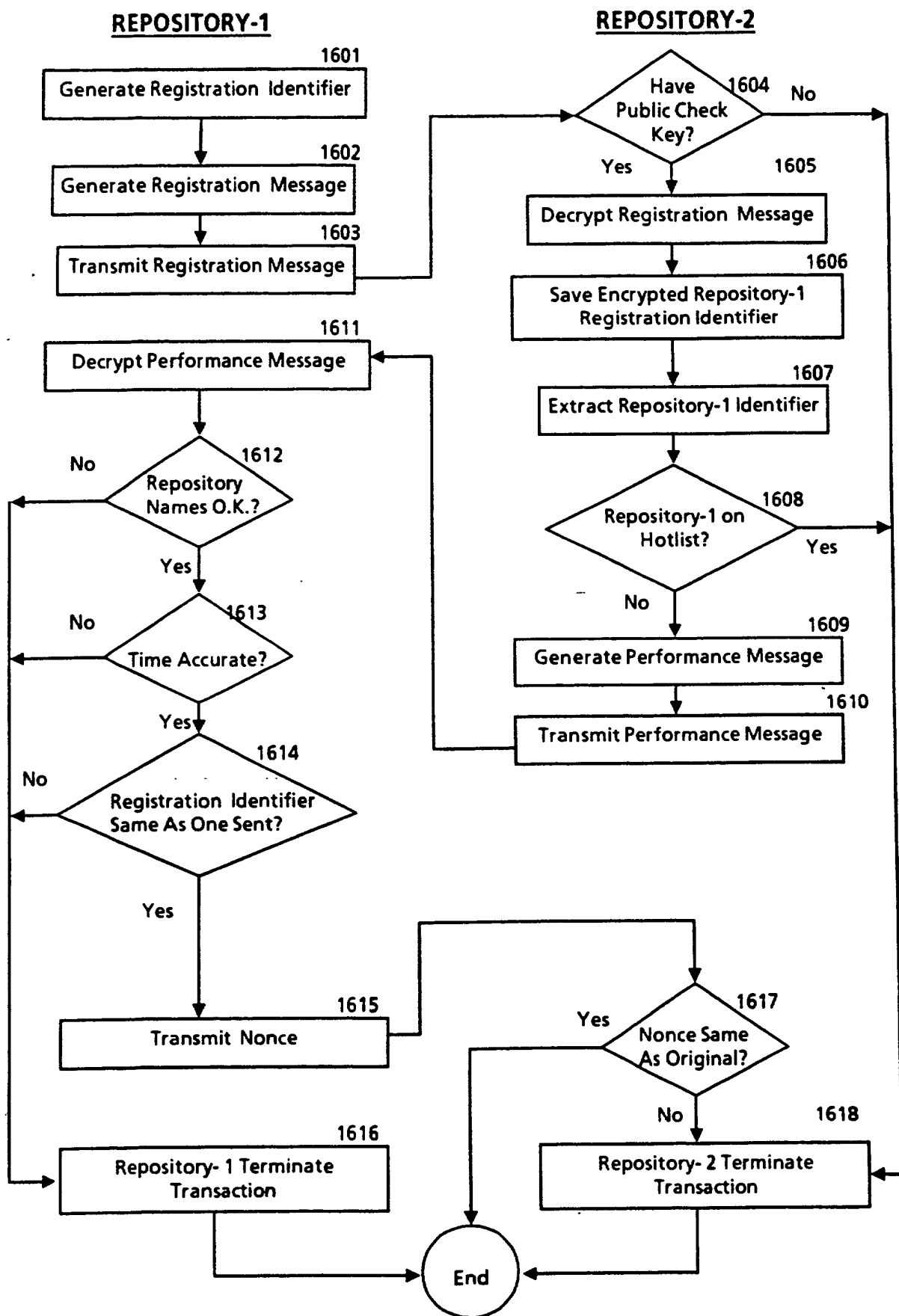


Figure 17

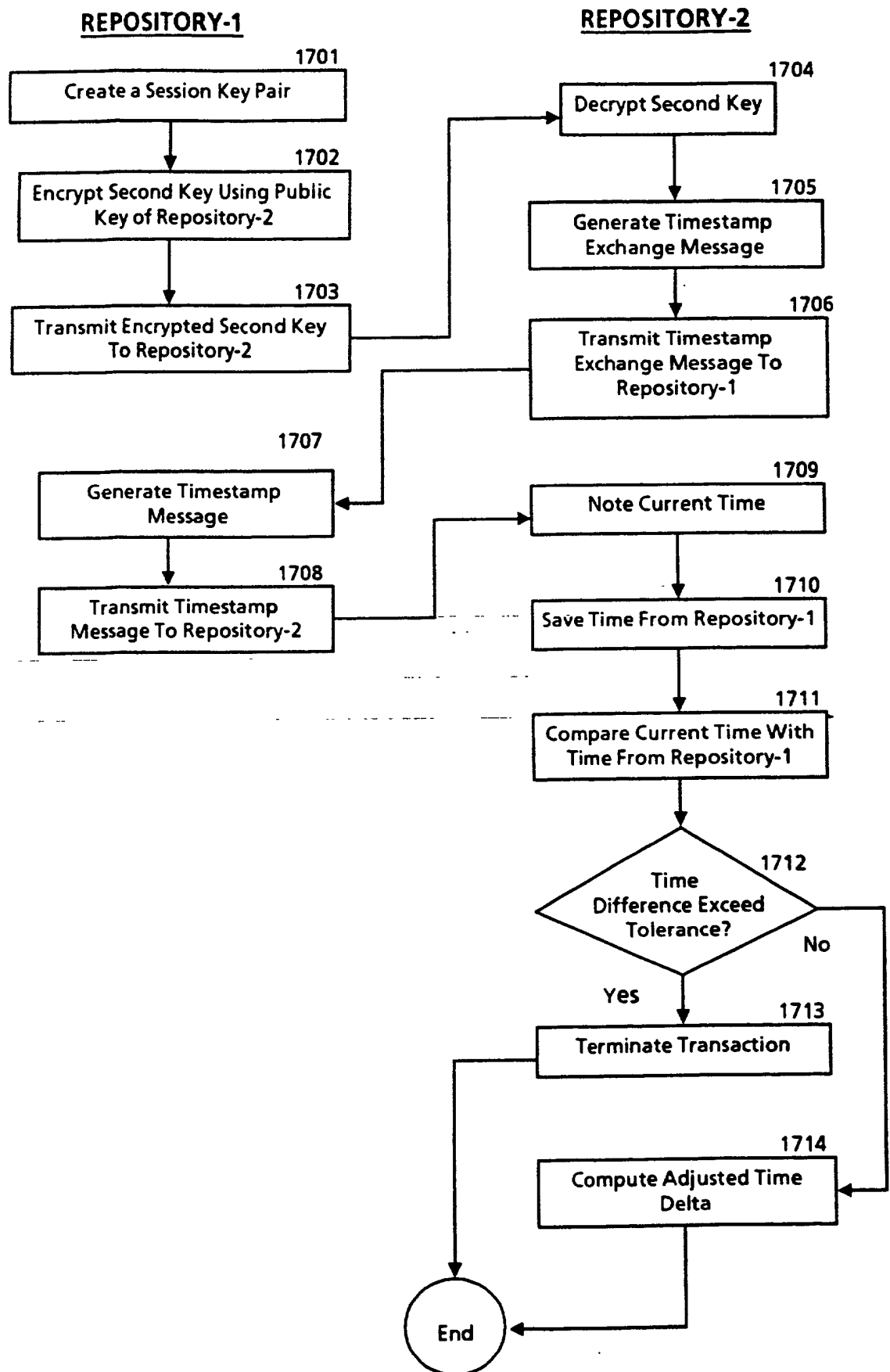


Figure 18

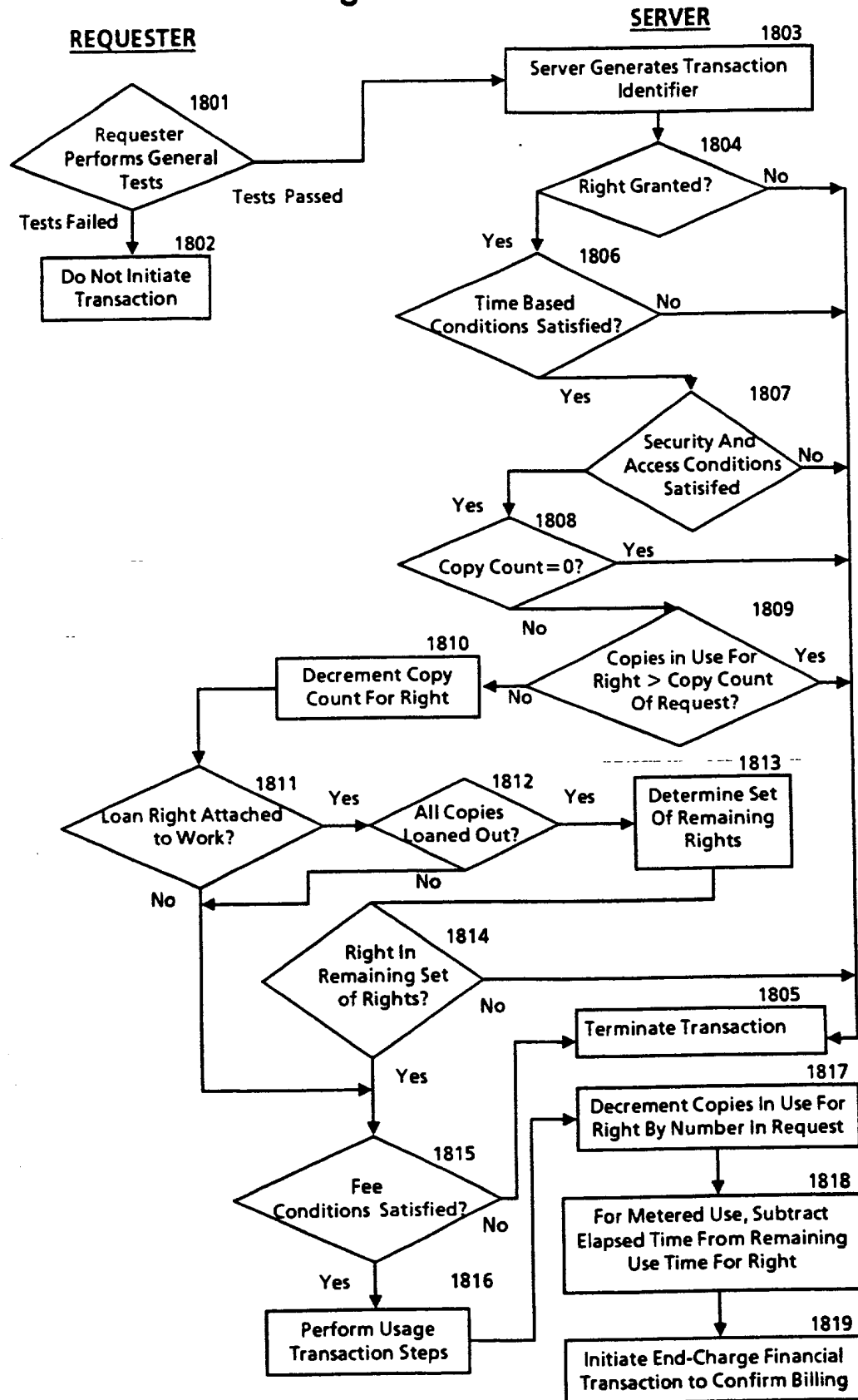
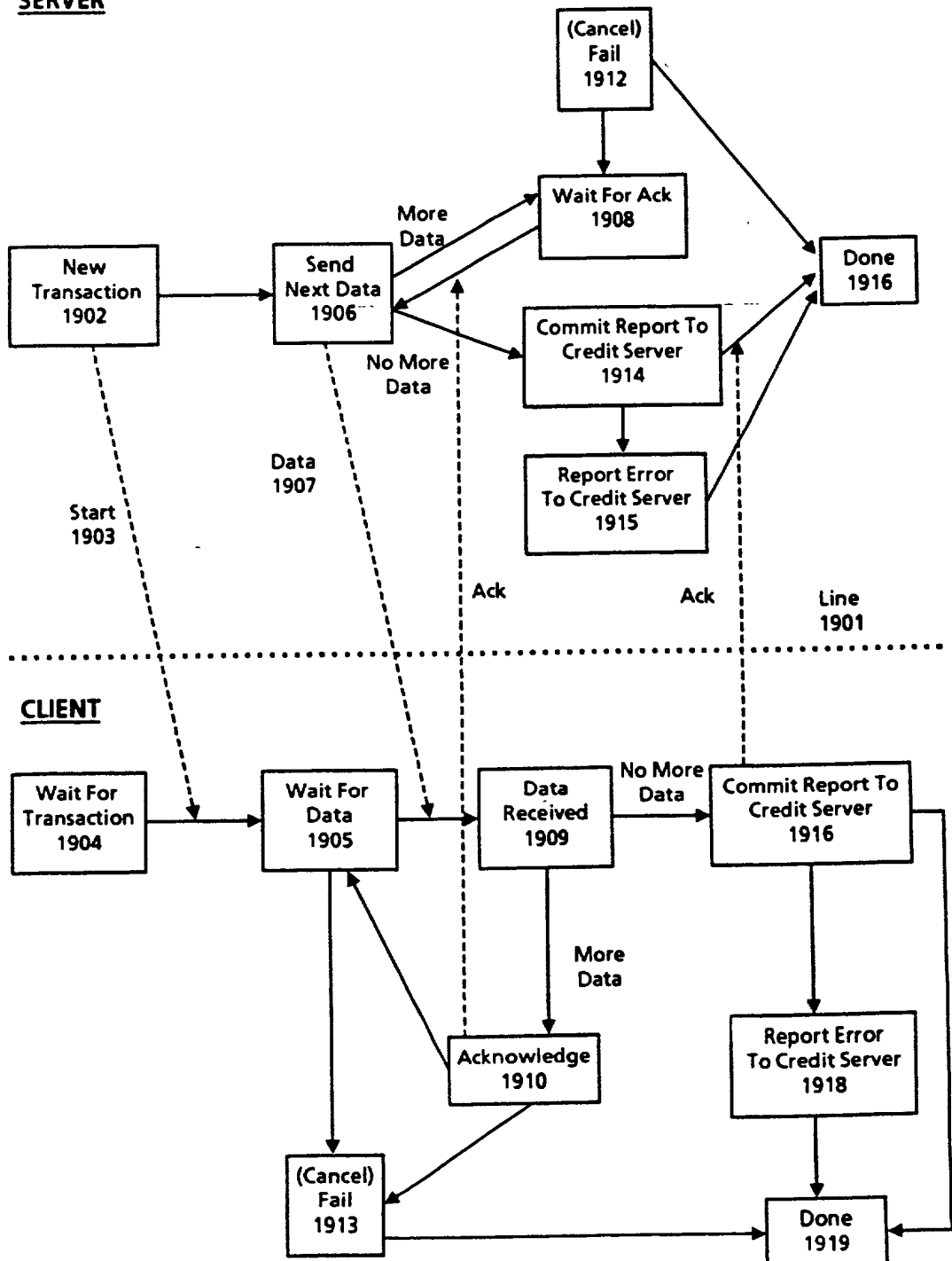


Figure 19

SERVER



ABSTRACT

[0418] Digital work adapted to be distributed within a system for controlling at least one of the distribution and use of digital works. The digital work includes: digital content representing a portion of a digital work suitable for being rendered by a rendering device and usage rights associated with the digital content. The usage rights specify a manner of use indicating one or more stated purposes for which the digital work can be at least one of used and distributed by an authorized party.

Exhibit 1012, p. 102

Figure 1

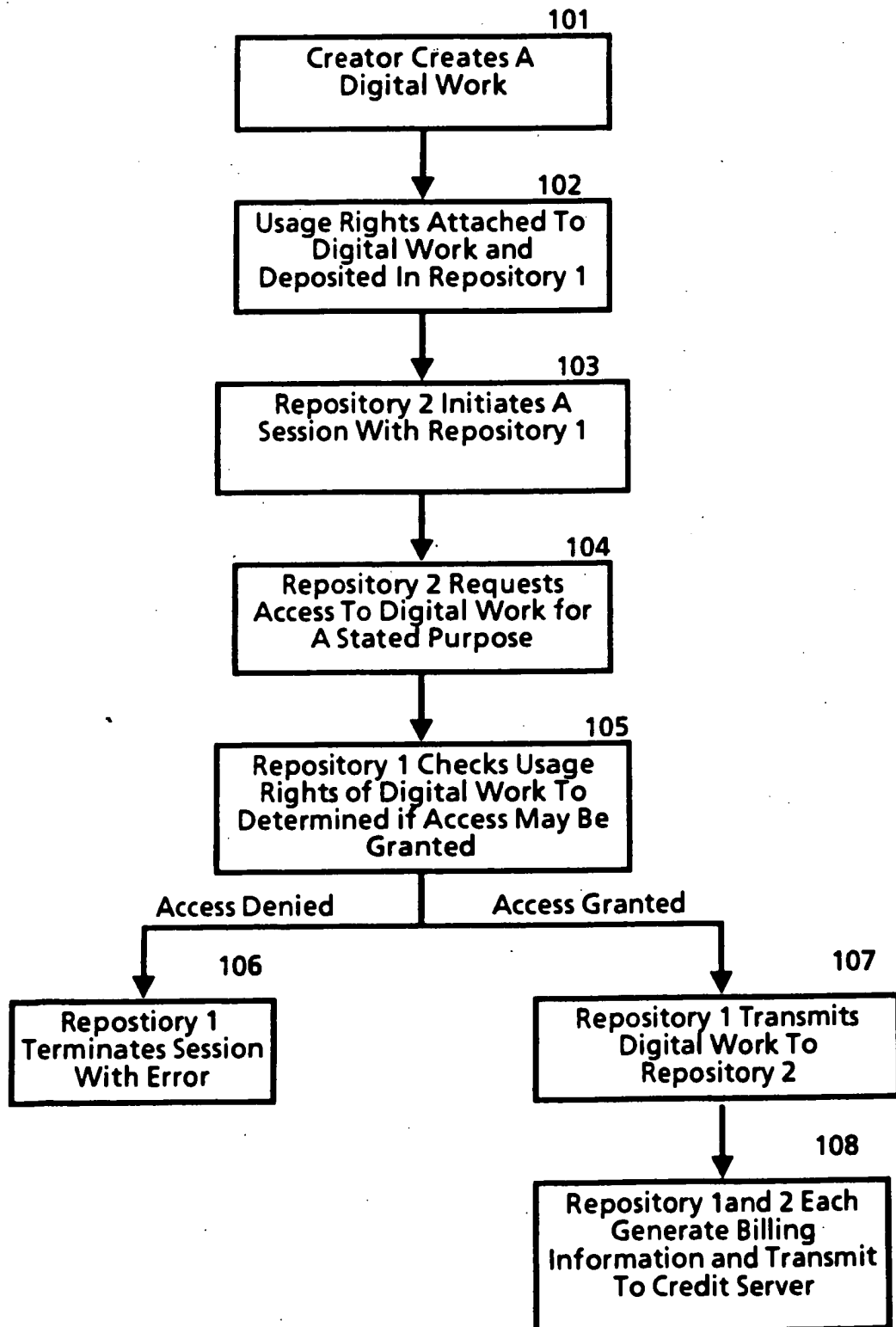


Figure 2

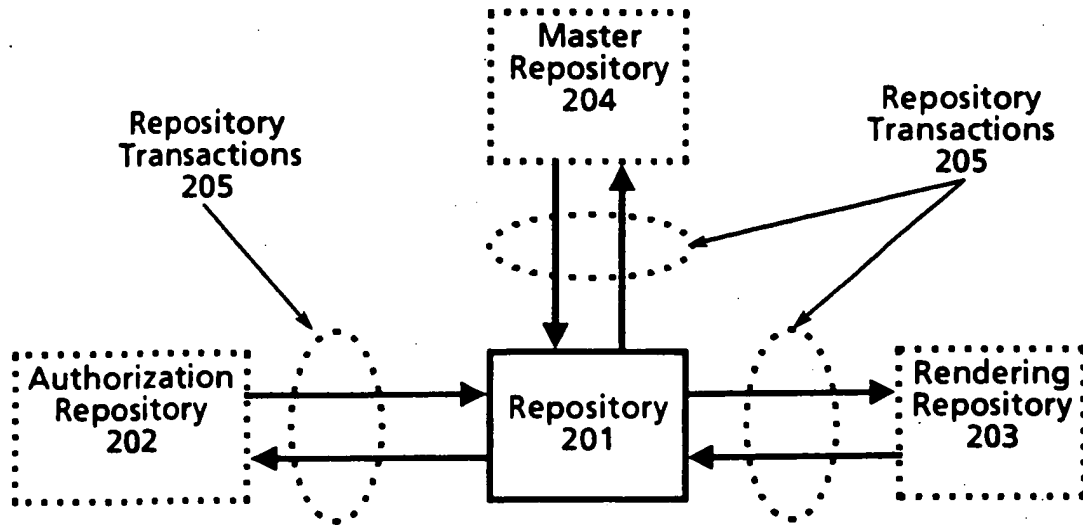


Figure 3

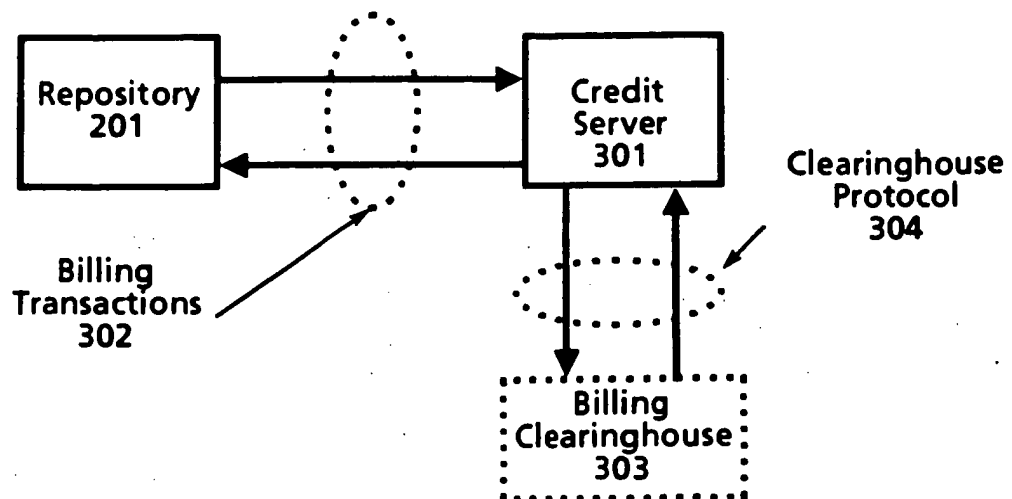


Figure 4a

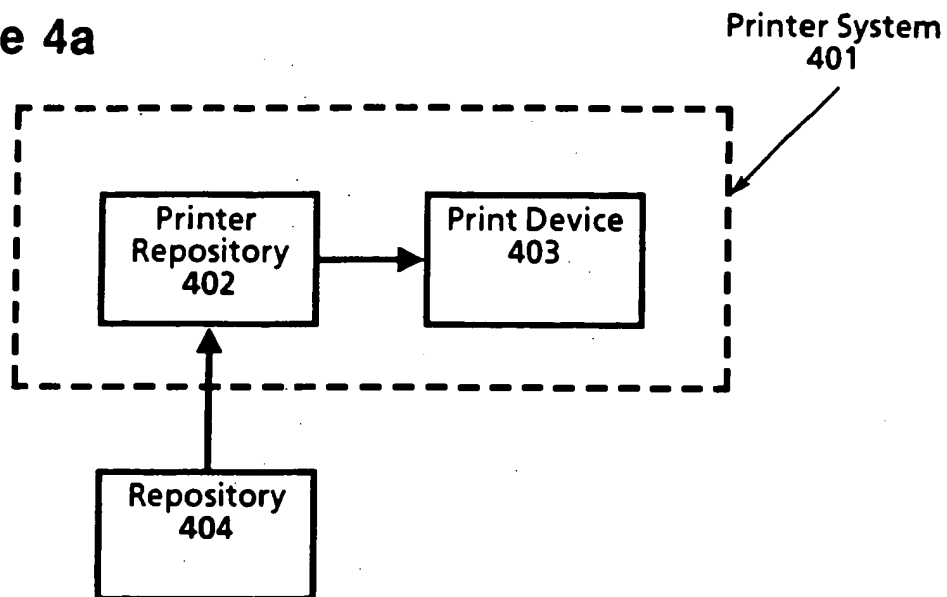
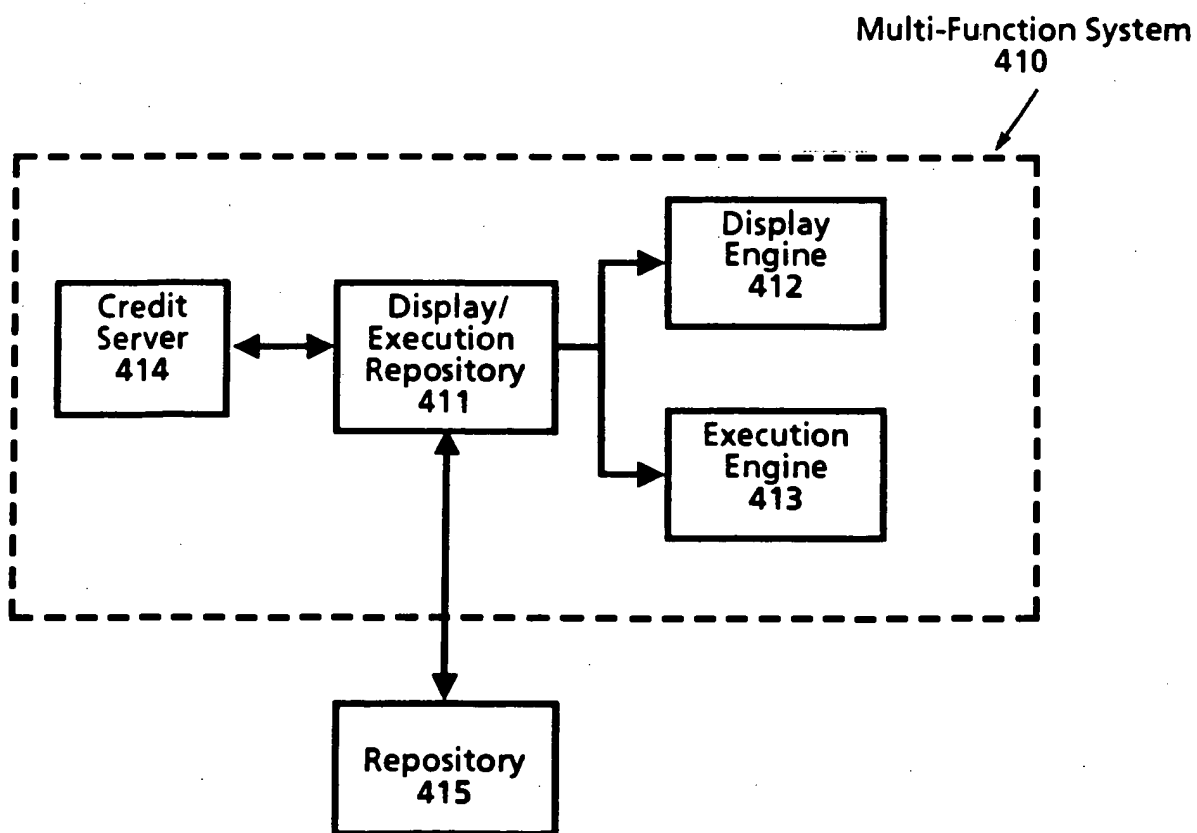


Figure 4b



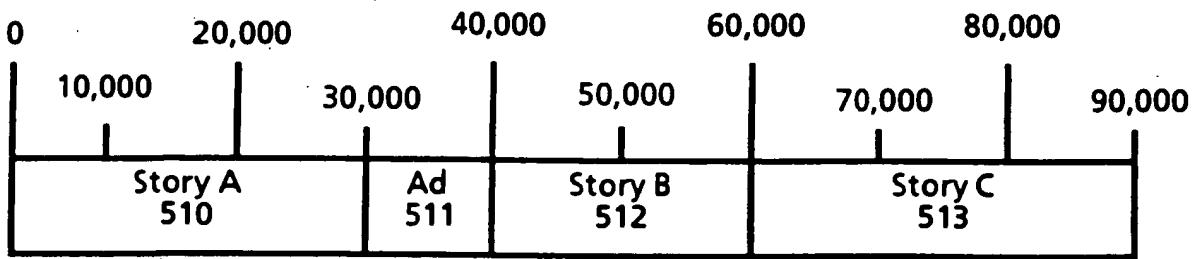


Figure 5

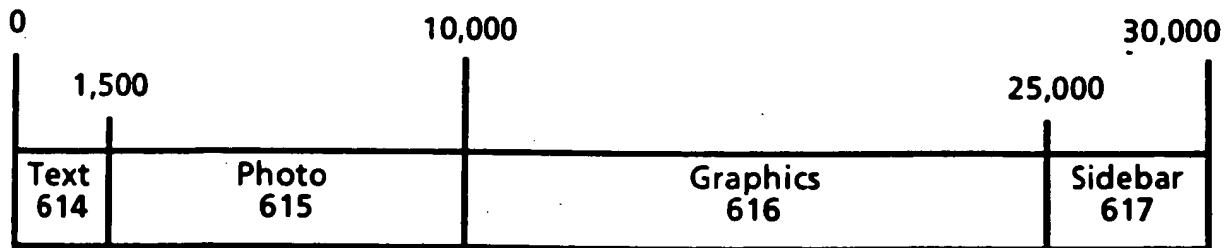


Figure 6

Figure 7

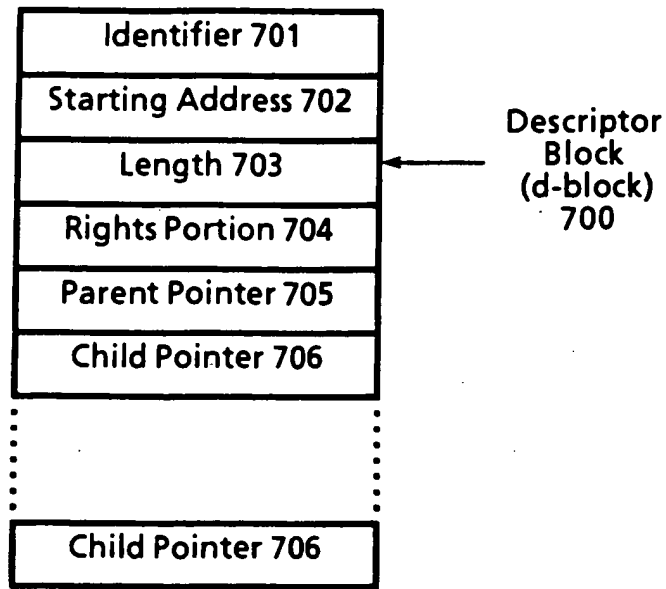


Figure 8

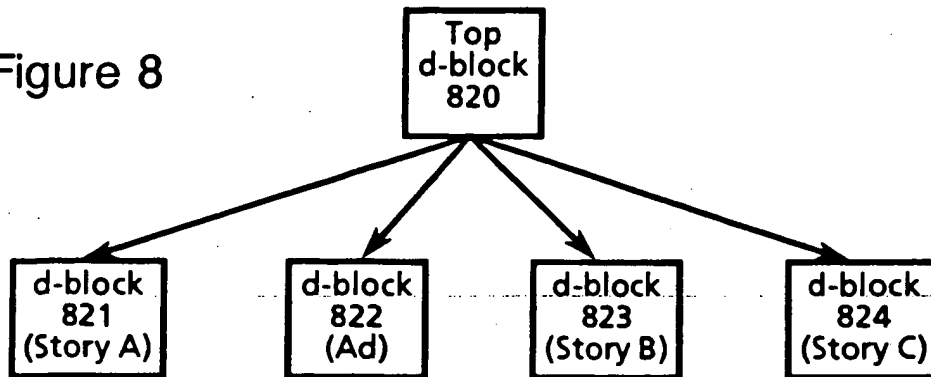


Figure 9

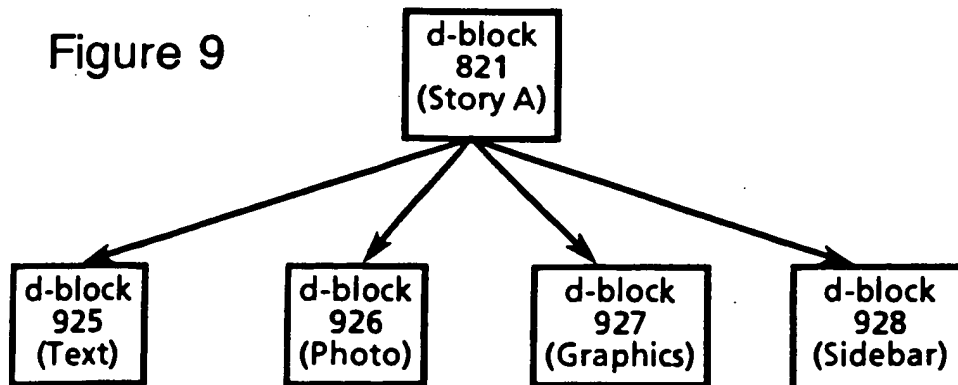


Figure 10



Figure 14

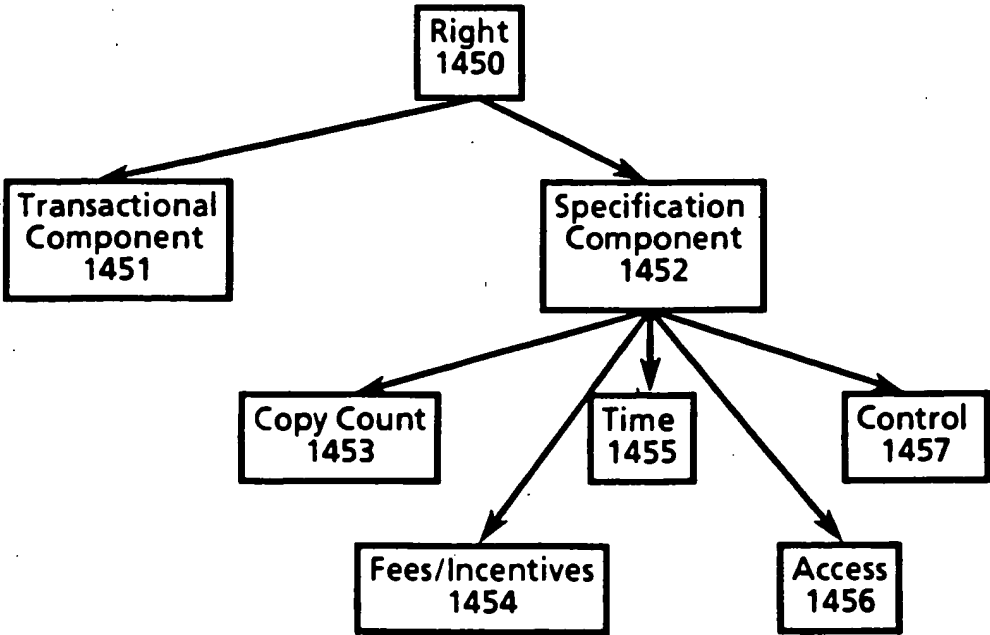


Figure 11

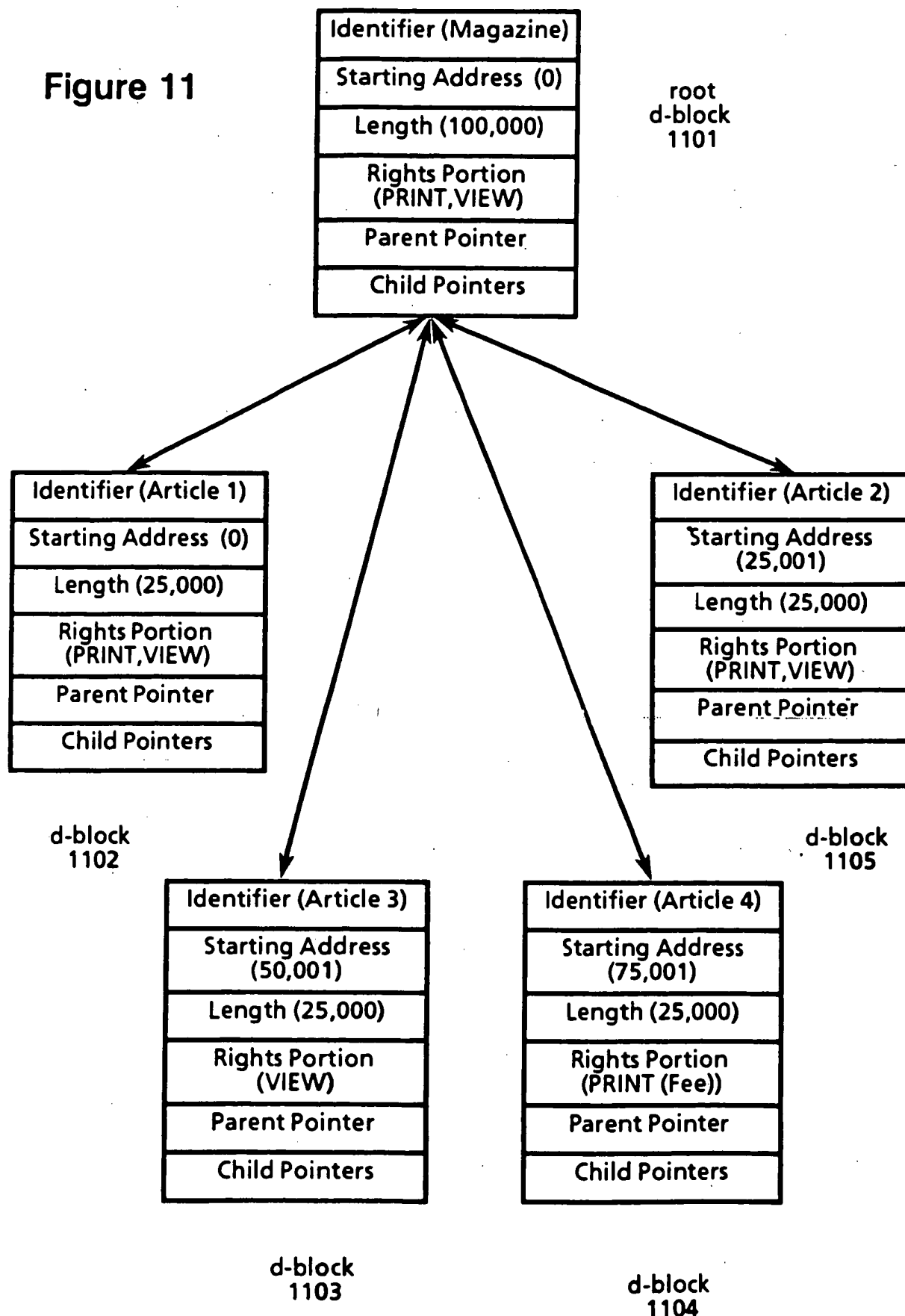


Figure 12

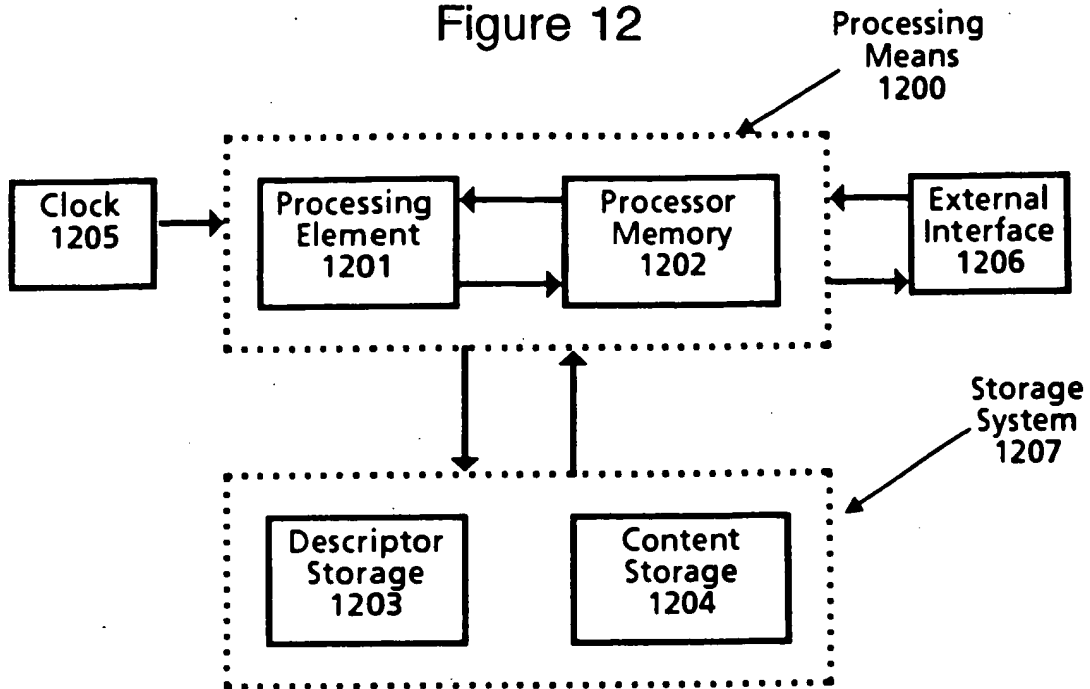


Figure 13

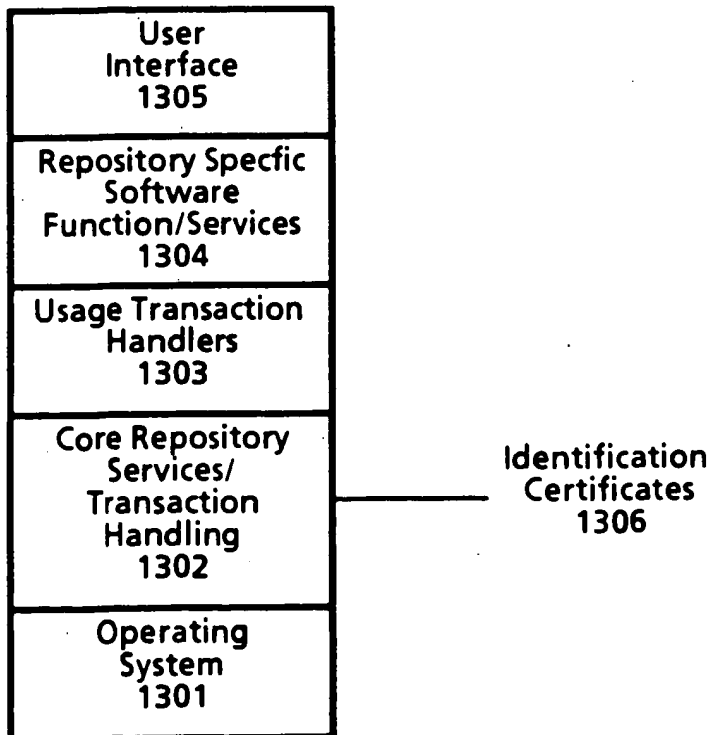


FIGURE 15

1501 ~ Digital Work Rights := (Rights*)

1502 ~ Right := (Right-Code {Copy-Count} {Control-Spec} {Time-Spec} {Access-Spec} {Fee-Spec})

1503 ~ Right-Code := Render-Code | Transport-Code | File-Management-Code | Derivative-Works-Code | Configuration-Code

1504 ~ Render-Code := [Play : {Player: Player-ID} | Print: {Printer: Printer-ID}]

1505 ~ Transport-Code := [Copy | Transfer | Loan {Remaining-Rights: Next-Set-of-Rights}] { (Next-Copy-Rights: Next-Set-of-Rights) }

1506 ~ File-Management-Code := Backup {Back-Up-Copy-Rights: Next-Set-of-Rights} | Restore | Delete | Folder | Directory {Name: Hide-Local | Hide-Remote} {Parts: Hide-Local | Hide-Remote}

1507 ~ Derivative-Works-Code := [Extract | Embed | Edit {Process: Process-ID}] {Next-Copy-Rights: Next-Set-of-Rights}

1508 ~ Configuration-Code := Install | Uninstall

1509 ~ Next-Set-of-Rights := { (Add: Set-Of-Rights) } { (Delete: Set-Of-Rights) } { (Replace: Set-Of-Rights) } { (Keep: Set-Of-Rights) }

1510 ~ Copy-Count := (Copies: positive-integer | 0 | Unlimited)

1511 ~ Control-Spec := (Control: {Restrictable | Unrestrictable} {Unchargeable | Chargeable})

1512 ~ Time-Spec := ({Fixed-Interval | Sliding-Interval | Meter-Time} Until: Expiration-Date)

1513 ~ Fixed-Interval := From: Start-Time

1514 ~ Sliding-Interval := Interval: Use-Duration

1516 ~ Meter-Time := Time-Remaining: Remaining-Use

1516 ~ Access-Spec := { (SC: Security-Class) {Authorization: Authorization-ID*} {Other-Authorization: Authorization-ID*} {Ticket: Ticket-ID} }

1517 ~ Fee-Spec := {Scheduled-Discount} Regular-Fee-Spec | Scheduled-Fee-Spec | Markup-Spec

1518 ~ Scheduled-Discount := Scheduled-Discount: (Scheduled-Discount: (Time-Spec Percentage)*)

1519 ~ Regular-Fee-Spec := ({Fee: | Incentive: } [Per-Use-Spec | Metered-Rate-Spec | Best-Price-Spec | Call-For-Price-Spec] {Min: Money-Unit Per: Time-Spec} {Max: Money-Unit Per: Time-Spec} To: Account-ID)

1520 ~ Per-Use-Spec := Per-Use: Money-unit

1521 ~ Metered-Rate-Spec := Metered: Money-Unit Per: Time-Spec

1522 ~ Best-Price-Spec := Best-Price: Money-unit Max: Money-unit

1523 ~ Call-For-Price-Spec := Call-For-Price

1524 ~ Scheduled-Fee-Spec := (Schedule: (Time-Spec Regular-Fee-Spec)*)

1525 ~ Markup-Spec := Markup: percentage To: Account-ID

Figure 16

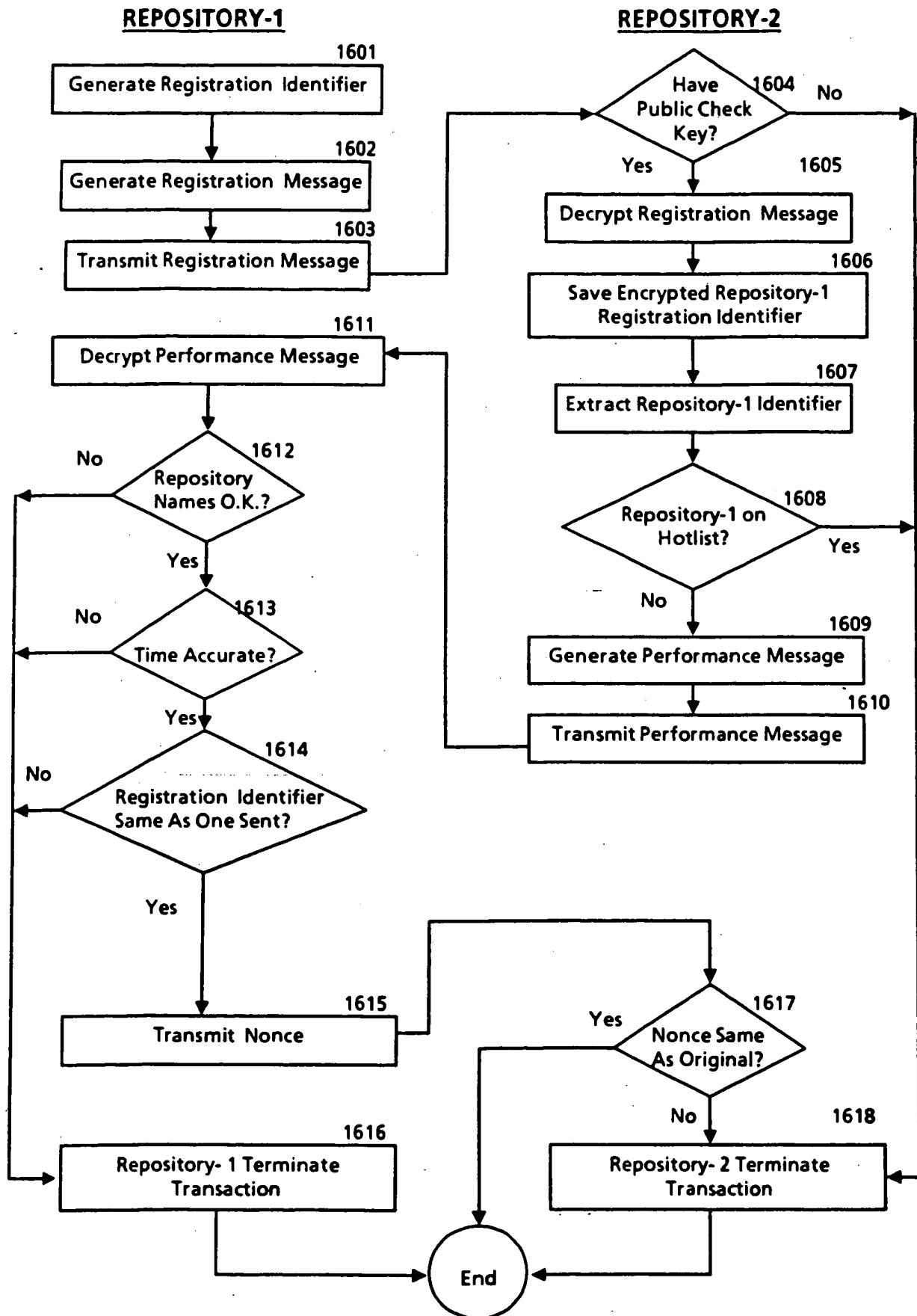


Figure 17

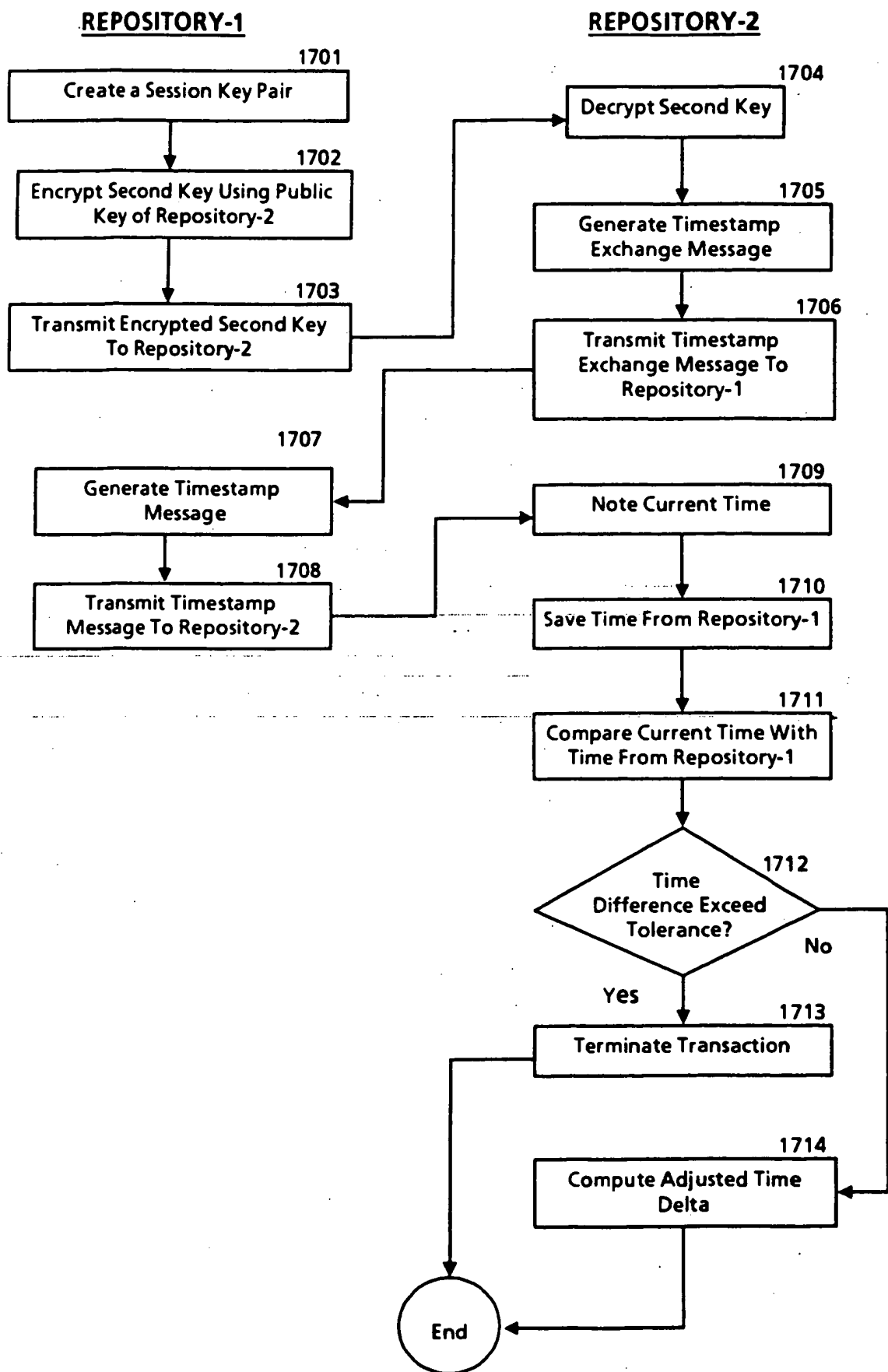


Figure 18

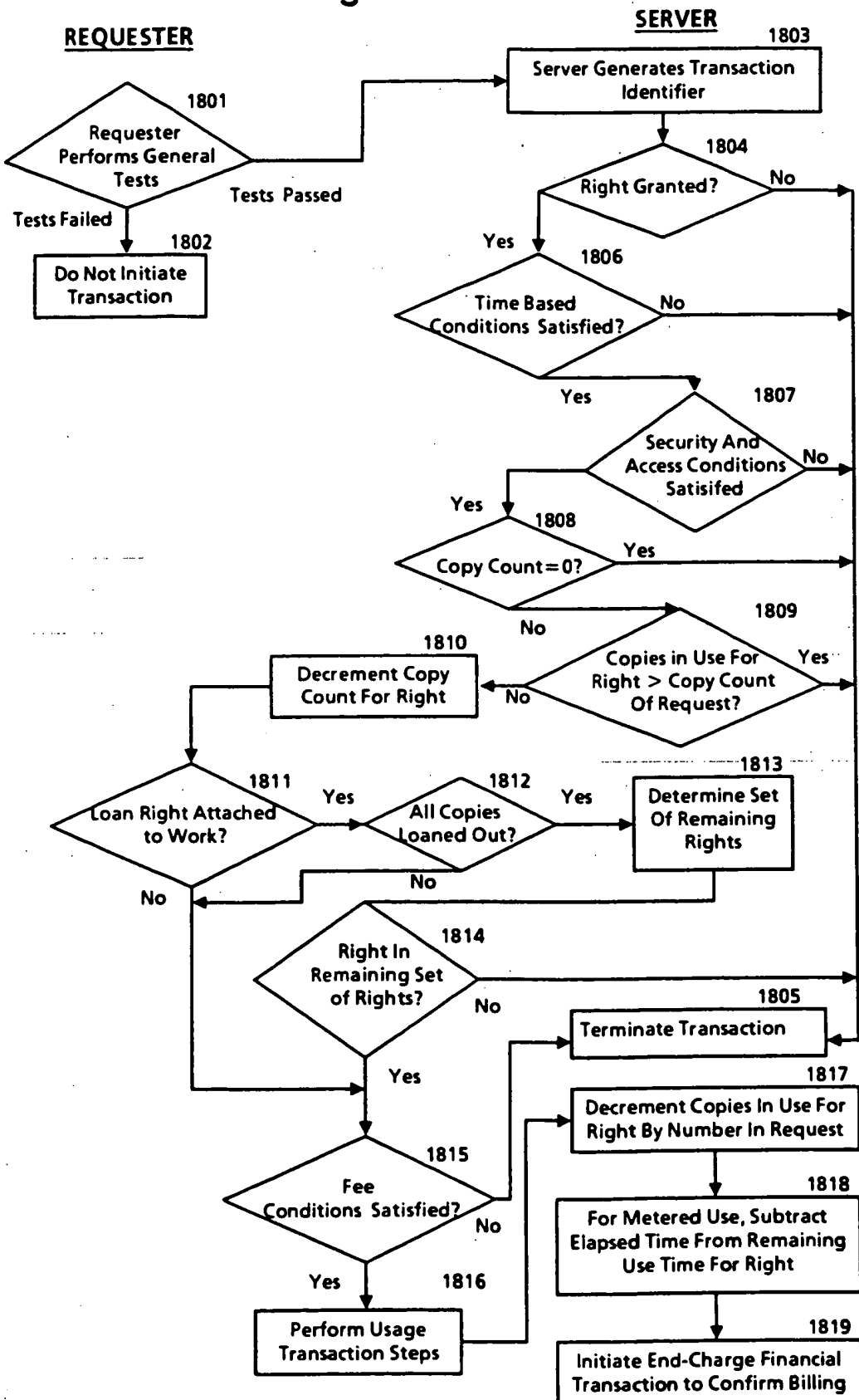
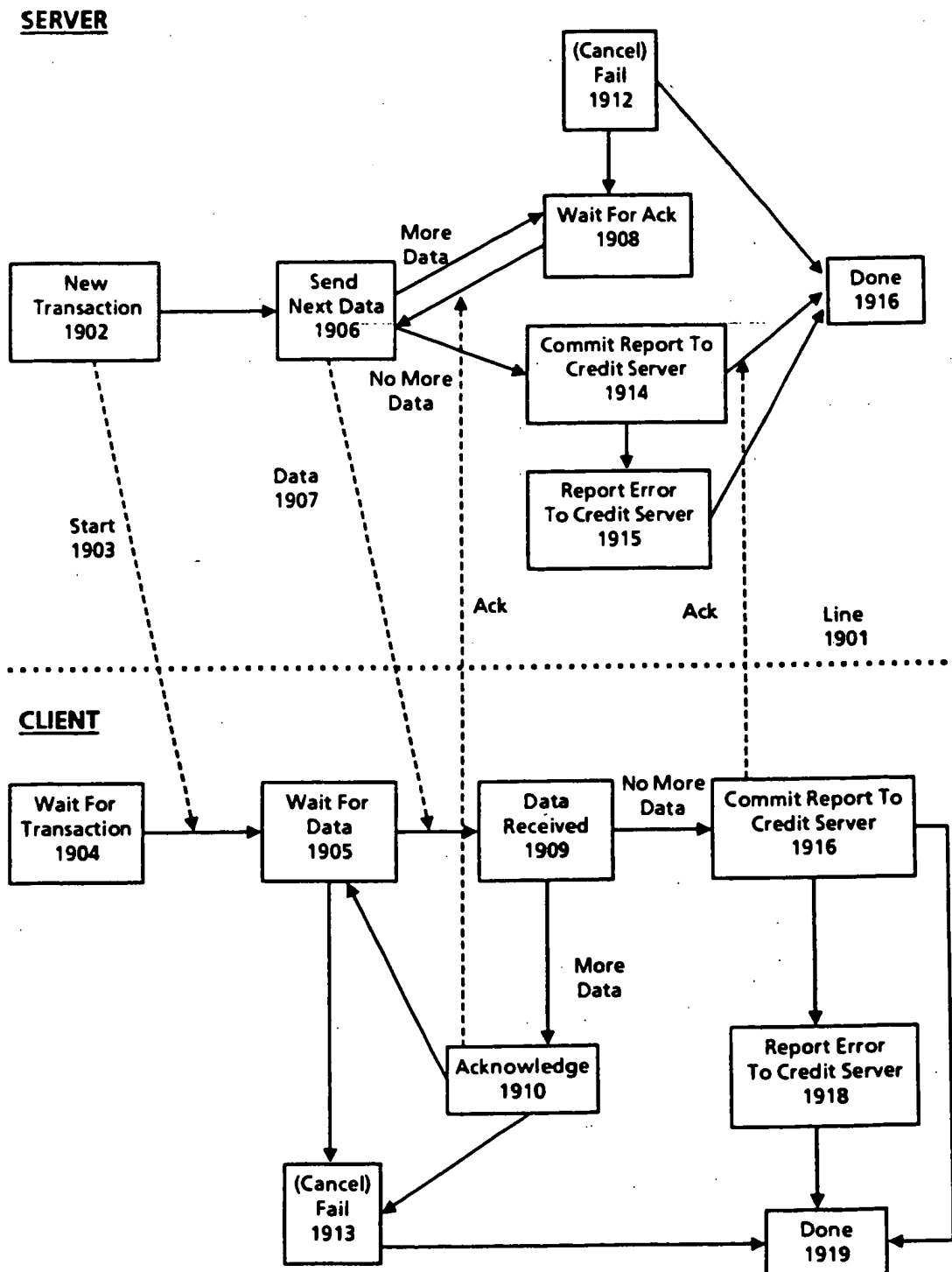


Figure 19



[illegible]

(List databases searched. Attach search strategy inside.)

Search strategy, include, /		
	Date	Exmr.

INTERFERENCE SEARCHED			
Class	Sub.	Date	Exmr.

3035 U.S. PRO
10/015951



U.S. UTILITY Patent Application

PATENT NUMBER and
ISSUE DATE

APPL NUM	FILING DATE	CLASS	SUBCLASS	GAU	EXAMINER
10015951	12/17/2001	705	52	2161	Hewitt

**APPLICANTS: Stefik Mark; Pirolli Peter;

**CONTINUING DATA VERIFIED:

THIS APPLICATION IS A CON OF 09/778,001 02/07/2001 Hewitt
WHICH IS A DIV OF 08/967,084 11/10/1997 PAT 6,236,971
WHICH IS A CON OF 08/344,760 11/23/1994 ABN

** FOREIGN APPLICATIONS VERIFIED:

PG-PUB	DO NOT PUBLISH ☐	RESCIND ☐	
Foreign priority claimed ☐ yes ☐ no		ATTORNEY DOCKET NO	
35 USC 119 conditions met ☐ yes ☐ no		111325-88	
Verified and Acknowledged Examiners's initials			
TITLE : Digital works having usage rights and method for creating the same			
U.S. DEPT. OF COMM./PAT. & TM-PTO-436L (Rev. 12-94)			

NOTICE OF ALLOWANCE MAILED		Assistant Examiner	CLAIMS ALLOWED	
			Total Claims	Print Claim for O.G.
ISSUE FEE		Primary Examiner	DRAWING	
Amount Due	Date Paid		Sheets Drwg.	Figs. Drwg.
☐ TERMINAL DISCLAIMER		PREPARED FOR ISSUE	Application Examiner	
		WARNING: The information disclosed herein may be restricted. Unauthorized disclosure may be prohibited by the United States Code Title 35, Sections 122, 181 and 368, Possession outside the U.S. Patent & Trademark Office is restricted to authorized employees and contractors only.		

FILED WITH:

☐ DISK (CRF)

☐ CD-ROM

(Attached in pocket on right inside flap)

ISSUE SLIP STAPLE AREA (for additional cross-references)

ISSUING CLASSIFICATION

ORIGINAL				CROSS REFERENCE(S)								
CLASS		SUBCLASS		CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)							
INTERNATIONAL CLASSIFICATION												

^ Continued on Issue Slip Inside File Jacket

INDEX OF CLAIMS

✓ Rejected - (Through numeral) ... Canceled N Non-elected A Appeal
 = ... Allowed + ... Restricted I Interference O Objected

Claim	Date
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
40	
41	
42	
43	
44	
45	
46	
47	
48	
49	
50	

Claim	Date
51	
52	
53	
54	
55	
56	
57	
58	
59	
60	
61	
62	
63	
64	
65	
66	
67	
68	
69	
70	
71	
72	
73	
74	
75	
76	
77	
78	
79	
80	
81	
82	
83	
84	
85	
86	
87	
88	
89	
90	
91	
92	
93	
94	
95	
96	
97	
98	
99	
100	

Claim	Date
101	
102	
103	
104	
105	
106	
107	
108	
109	
110	
111	
112	
113	
114	
115	
116	
117	
118	
119	
120	
121	
122	
123	
124	
125	
126	
127	
128	
129	
130	
131	
132	
133	
134	
135	
136	
137	
138	
139	
140	
141	
142	
143	
144	
145	
146	
147	
148	
149	
150	

If more than 150 claims or 9 actions staple additional sheet here

2/11/10

10/015951
12/17/01

UTILITY PATENT APPLICATION TRANSMITTAL <i>(Only for new nonprovisional applications under 37 CFR 1.53(b))</i>		Attorney Docket No. 111325-88	
		First Inventor Mark J. Stefik et al.	
		Title Digital Works Having Usage Rights And Method For Creating The Same	
		Express Mail Label No. _____	

APPLICATION ELEMENTS See MPEP chapter 600 concerning utility patent application contents.	ADDRESS TO: Commissioner for Patents Box Patent Application Washington, DC 20231

1. ☐ Fee Transmittal Form (e.g., PTO/SB/17) <i>(Submit an original and a duplicate for fee processing)</i> 2. ☐ Applicant claims small entity status. See 37 CFR 1.27. 3. ☒ Specification [Total Pages 88] <i>(preferred arrangement set forth below)</i> - Descriptive title of the invention - Cross Reference to Related Applications <i>(if applicable)</i> - Statement Regarding Fed sponsored R & D <i>(if applicable)</i> - Reference to sequence listing, a table, or a computer program listing appendix <i>(if applicable)</i> - Background of the Invention - Brief Summary of the Invention - Brief Description of the Drawings <i>(if filed)</i> - Detailed Description - Claim(s) - Abstract of the Disclosure 4. ☒ Drawing(s) (35 U.S.C. 113) [Total Sheets 13] 5. Oath or Declaration [Total Pages ☐] a. ☐ Newly executed (original or copy) b. ☐ Copy from a prior application (37 CFR 1.63(d)) <i>(for continuation/divisional with Box 18 completed)</i> i. ☐ DELETION OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b) 6. ☐ Application Data Sheet. See 37 CFR 1.76	7. ☐ CD-ROM or CD-R in duplicate, large table or Computer Program (Appendix) 8. Nucleotide and/or Amino Acid Sequence Submission <i>(if applicable, all necessary)</i> a. ☐ Computer Readable Form (CRF) b. Specification Sequence Listing on: i. ☐ CD-ROM or CD-R (2 copies; or ii. ☐ paper c. ☐ Statements verifying identity of above copies ACCOMPANYING APPLICATION PARTS 9. ☐ Assignment Papers (cover sheet & document(s)) 10. ☐ 37 CFR 3.73(b) Statement ☐ Power of Attorney <i>(when there is an assignee)</i> 11. ☐ English Translation Document <i>(if applicable)</i> 12. ☐ Information Disclosure Statement (IDS)/PTO-1449 ☐ Copies of IDS Citations 13. ☐ Preliminary Amendment 14. ☒ Return Receipt Postcard (MPEP 503) <i>(Should be specifically itemized)</i> 15. ☐ Certified Copy of Priority Document(s) <i>(if foreign priority is claimed)</i> 16. ☐ Nonpublication request under 35 U.S.C. 122(b)(2)(B)(i). Applicant must attach form PTO/SB/35 or its equivalent. 17. ☐ Other: _____

18. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment, or in an Application Data Sheet under 37 CFR 1.76:

☒ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: 09/778,001 filed Feb. 7, 2001, which is a Divisional of Serial No. 08/967,084, filed on Nov. 10, 1997, now U.S. Patent No. 6,236,971, which is a Continuation of Serial No. 08/344,760 filed Nov. 23, 1994, now abandoned.

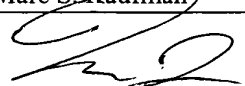
Prior application information: _____ Examiner: _____ Group / Art Unit: _____

For CONTINUATION OR DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 5b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

19. CORRESPONDENCE ADDRESS

☒ Customer Number or Bar Code Label	22204	or ☐ Correspondence address below

Name		Marc S. Kaufman	
Address			
City	State	Zip Code	
Country	Telephone	Fax	

Name (Print/Type)	Marc S. Kaufman	Registration No. (Attorney/Agent)	35,212
Signature		Date	December 17, 2001

CLAIMS ONLY

SERIAL NO.

10015951

FILING DATE

12/17/01

APPLICANT(S)

CLAIMS

	AS FILED		AFTER 1st AMENDMENT		AFTER 2nd AMENDMENT			★		★		★	
	IND.	DEP.	IND.	DEP.	IND.	DEP.		IND.	DEP.	IND.	DEP.	IND.	DEP.
1	/						51						
2		/					52						
3		/					53						
4		/					54						
5		/					55						
6		/					56						
7		/					57						
8		/					58						
9		/					59						
10		/					60						
11	/						61						
12		/					62						
13		/					63						
14		/					64						
15		/					65						
16		/					66						
17		/					67						
18		/					68						
19		/					69						
20		/					70						
21	/						71						
22		/					72						
23		/					73						
24		/					74						
25		/					75						
26		/					76						
27	/						77						
28		/					78						
29		/					79						
30		/					80						
31		/					81						
32		/					82						
33		/					83						
34		/					84						
35							85						
36							86						
37							87						
38							88						
39							89						
40							90						
41							91						
42							92						
43							93						
44							94						
45							95						
46							96						
47							97						
48							98						
49							99						
50							100						
TOTAL IND.	4						TOTAL IND.						
TOTAL DEP.	30						TOTAL DEP.						
TOTAL CLAIMS	34						TOTAL CLAIMS						

* MAY BE USED FOR ADDITIONAL CLAIMS OR ADMENDMENTS

U.S. DEPARTMENT OF COMMERCE
Patent and Trademark Office

FORM PTO-2022 (1-98)

*U.S. Government Printing Office: 1998 - 433-214/70303

**DIGITAL WORKS HAVING USAGE RIGHTS AND
METHOD FOR CREATING THE SAME**

RELATED APPLICATIONS

[0001] This application is a continuation of application Ser. No. 09/778,001, filed February 7, 2001, which is a divisional of Ser. No. 08/967,084, filed on Nov. 10, 1997, now U.S. patent 6,236,971, which is a continuation of Ser. No. 08/344,760, filed Nov. 23, 1994, now abandoned, the disclosure of which are being incorporated by reference.

FIELD OF THE INVENTION

[0002] The present invention relates to the field of distribution and usage rights enforcement for digitally encoded works.

BACKGROUND OF THE INVENTION

[0003] A fundamental issue facing the publishing and information industries as they consider electronic publishing is how to prevent the unauthorized and unaccounted distribution or usage of electronically published materials. Electronically published materials are typically distributed in a digital form and recreated on a computer based system having the capability to recreate the materials. Audio and video recordings, software, books and multimedia works are all being electronically published. Companies in these industries receive royalties for each accounted for delivery of the materials, e.g. the sale of an audio CD at a retail outlet. Any unaccounted distribution of a work results in an unpaid royalty (e.g. copying the audio recording CD to another digital medium.)

[0004] The ease in which electronically published works can be "perfectly" reproduced and distributed is a major concern. The transmission of digital works over networks is commonplace. One such widely used network is the Internet. The Internet is a widespread

NVA208575.1

network facility by which computer users in many universities, corporations and government entities communicate and trade ideas and information. Computer bulletin boards found on the Internet and commercial networks such as CompuServ and Prodigy allow for the posting and retrieving of digital information. Information services such as Dialog and LEXIS/NEXIS provide databases of current information on a wide variety of topics. Another factor which will exacerbate the situation is the development and expansion of the National Information Infrastructure (the NII). It is anticipated that, as the NII grows, the transmission of digital works over networks will increase many times over. It would be desirable to utilize the NII for distribution of digital works without the fear of widespread unauthorized copying.

[0005] The most straightforward way to curb unaccounted distribution is to prevent unauthorized copying and transmission. For existing materials that are distributed in digital form, various safeguards are used. In the case of software, copy protection schemes which limit the number of copies that can be made or which corrupt the output when copying is detected have been employed. Another scheme causes software to become disabled after a predetermined period of time has lapsed. A technique used for workstation based software is to require that a special hardware device must be present on the workstation in order for the software to run, e.g., see U.S. Pat. No. 4,932,054 entitled "Method and Apparatus for Protecting Computer Software Utilizing Coded Filter Network in Conjunction with an Active Coded Hardware Device." Such devices are provided with the software and are commonly referred to as dongles.

[0006] Yet another scheme is to distribute software, but which requires a "key" to enable it's use. This is employed in distribution schemes where "demos" of the software are provided on a medium along with the entire product. The demos can be freely used, but in order to use the actual product, the key must be purchased. These scheme do not hinder copying of the software once the key is initially purchased.

[0007] A system for ensuring that licenses are in place for using licensed products is described in PCT Publication WO 93/01550 to Griswold entitled "License Management System and Method." The licensed product may be any electronically published work but is most effective for use with works that are used for extended periods of time such as software

NVA208575.1

programs. Griswold requires that the licensed product contain software to invoke a license check monitor at predetermined time intervals. The license check monitor generates request datagrams which identify the licensee. The request datagrams are sent to a license control system over an appropriate communication facility. The license control system then checks the datagram to determine if the datagram is from a valid licensee. The license control system then sends a reply datagram to the license check monitor indicating denial or approval of usage. The license control system will deny access in the event that request datagrams go unanswered after a predetermined period of time (which may indicate an unauthorized attempt to use the licensed product). In this system, access is managed at a central location by the response datagrams. So for example if license fees have not been paid, access to the licensed product is terminated.

[0008] It is argued by Griswold that the described system is advantageous because it can be implemented entirely in software. However, the system described by Griswold has limitations. An important limitation is that during the use of the licensed product, the user must always be coupled to an appropriate communication facility in order to send and receive datagrams. This creates a dependency on the communication facility. So if the communication facility is not available, the licensed product cannot be used. Moreover, some party must absorb the cost of communicating with the license server.

[0009] A system for controlling the distribution of digitally encoded books is embodied in a system available from VPR Systems, LTD. of St. Louis, Mo. The VPR system is self-contained and is comprised of: (1) point of sale kiosks for storing and downloading of books, (2) personal storage mediums (cartridges) to which the books are downloaded, and (3) readers for viewing the book. In a purchase transaction, a purchaser will purchase a voucher card representing the desired book. The voucher will contain sufficient information to identify the book purchased and perhaps some demographic information relating to the sales transaction. To download the book, the voucher and the cartridge are inserted into the kiosk.

[0010] The VPR system may also be used as a library. In such an embodiment, the kiosk manages the number of "copies" that may be checked out at one time. Further, the copy of the book is erased from the users cartridge after a certain check-out time has expired. However,

individuals cannot loan books because the cartridges may only be used with the owners reader.

[0011] The foregoing distribution and protection schemes operate in part by preventing subsequent distribution of the work. While this certainly prevents unauthorized distributions, it does so by sacrificing the potential for subsequent revenue bearing uses. For example, it may be desirable to allow the lending of a purchased work to permit exposure of the work to potential buyers. Another example would be to permit the creation of a derivative work for a fee. Yet another example would be to permit copying the work for a fee (essentially purchasing it). Thus, it would be desirable to provide flexibility in the manner that the owner of a digital work may allow it to be distributed or used.

[0012] While flexibility in manner of distribution and use is a concern, the owners of a work want to make sure they are paid for such distributions. In U.S. Pat. No. 4,977,594 to Shear, entitled "Database Usage Metering and Protection System and Method," a system for metering and billing for usage of information distributed on a CD-ROM is described. The system requires the addition of a billing module to the computer system. The billing module may operate in a number of different ways. First, it may periodically communicate billing data to a central billing facility, whereupon the user may be billed. Second, billing may occur by disconnecting the billing module and the user sending it to a central billing facility where the data is read and a user bill generated.

[0013] U.S. Pat. No. 5,247,575, Sprague et al., entitled "Information Distribution System", describes an information distribution system which provides and charges only for user selected information. A plurality of encrypted information packages (IPs) are provided at the user site, via high and/or low density storage media and/or by broadcast transmission. Some of the IPs may be of no interest to the user. The IPs of interest are selected by the user and are decrypted and stored locally. The IPs may be printed, displayed or even copied to other storage medias. The charges for the selected IP's are accumulated within a user apparatus and periodically reported by telephone to a central accounting facility. The central accounting facility also issues keys to decrypt the IPs. The keys are changed periodically. If the central accounting facility has not issued a new key for a particular user station, the station is unable to retrieve

NVA208575.1

information from the system when the key is changed.

[0014] A system available from Wave Systems Corp. of Princeton, N.Y., provides for metering of software usage on a personal computer. The system is installed onto a computer and collects information on what software is in use, encrypts it and then transmits the information to a transaction center. From the transaction center, a bill is generated and sent to the user. The transaction center also maintains customer accounts so that licensing fees may be forwarded directly to the software providers. Software operating under this system must be modified so that usage can be accounted.

[0015] Known techniques distributing digital content do not permit the owner or other authorized party to specify a manner of use of the content and to associate the manner of use with the content in a persistent way.

SUMMARY OF THE INVENTION

[0016] A preferred embodiment of the invention is a digital work adapted to be distributed within a system for controlling at least one of the distribution and use of digital works. The digital work comprises: digital content representing a portion of a digital work suitable for being rendered by a rendering device and usage rights associated with the digital content. The usage rights specify a manner of use indicating one or more stated purposes for which the digital work can be at least one of used and distributed by an authorized party.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] FIG. 1 is a flowchart illustrating a simple instantiation of the operation of the currently preferred embodiment of the present invention.

[0018] FIG. 2 is a block diagram illustrating the various repository types and the repository transaction flow between them in the currently preferred embodiment of the present invention.

[0019] FIG. 3 is a block diagram of a repository coupled with a credit server in the currently preferred embodiment of the present invention.

[0020] FIGS. 4a and 4b are examples of rendering systems as may be utilized in the currently preferred embodiment of the present invention.

[0021] FIG. 5 illustrates a contents file layout for a digital work as may be utilized in the currently preferred embodiment of the present invention.

[0022] FIG. 6 illustrates a contents file layout for an individual digital work of the digital work of FIG. 5 as may be utilized in the currently preferred embodiment of the present invention.

[0023] FIG. 7 illustrates the components of a description block of the currently preferred embodiment of the present invention.

[0024] FIG. 8 illustrates a description tree for the contents file layout of the digital work illustrated in FIG. 5.

[0025] FIG. 9 illustrates a portion of a description tree corresponding to the individual digital work illustrated in FIG. 6.

[0026] FIG. 10 illustrates a layout for the rights portion of a description block as may be utilized in the currently preferred embodiment of the present invention.

[0027] FIG. 11 is a description tree wherein certain d-blocks have PRINT usage rights and is used to illustrate "strict" and "lenient" rules for resolving usage rights conflicts.

[0028] FIG. 12 is a block diagram of the hardware components of a repository as are utilized in the currently preferred embodiment of the present invention.

[0029] FIG. 13 is a block diagram of the functional (logical) components of a repository as are utilized in the currently preferred embodiment of the present invention.

[0030] FIG. 14 is diagram illustrating the basic components of a usage right in the currently preferred embodiment of the present invention.

[0031] FIG. 15 lists the usage rights grammar of the currently preferred embodiment of the present invention.

[0032] FIG. 16 is a flowchart illustrating the steps of certificate delivery, hotlist checking and performance testing as performed in a registration transaction as may be performed in the currently preferred embodiment of the present invention.

[0033] FIG. 17 is a flowchart illustrating the steps of session information exchange and clock synchronization as may be performed in the currently preferred embodiment of the present invention, after each repository in the registration transaction has successfully completed the steps described in FIG. 16.

[0034] FIG. 18 is a flowchart illustrating the basic flow for a usage transaction, including the common opening and closing step, as may be performed in the currently preferred embodiment of the present invention.

[0035] FIG. 19 is a state diagram of server and client repositories in accordance with a transport protocol followed when moving a digital work from the server to the client repositories, as may be performed in the currently preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0036] The preferred embodiment is directed to supporting commercial transactions involving digital works. The transition to digital works profoundly and fundamentally changes how creativity and commerce can work. It changes the cost of transporting or storing works because digital property is almost "massless." Digital property can be transported at electronic speeds and requires almost no warehousing. Keeping an unlimited supply of virtual copies on

NVA208575.1

hand requires essentially no more space than keeping one copy on hand. The digital medium also lowers the costs of alteration, reuse and billing.

[0037] There is a market for digital works because creators are strongly motivated to reuse portions of digital works from others rather than creating their own completely. This is because it is usually so much easier to use an existing stock photo or music clip than to create a new one from scratch.

[0038] Herein the phrase "digital work" refers to any work that has been reduced to a digital representation. This would include any audio, video, text, or multimedia work and any accompanying interpreter (e.g. software) that may be required for recreating or rendering the content of the work. The term composite work refers to a digital work comprised of a collection of other digital works. The term "usage rights" or "rights" is a term which refers to rights granted to a recipient of a digital work. Generally, these rights define the manner in which a digital work can be used and distributed. Each usage right may have one or more specified conditions which must be satisfied before the right may be exercised. A Glossary of the terms used herein is provided at the end of the specification.

[0039] Usage rights are permanently associated with the digital work. Copies made of a digital work will also have the associated usage rights. Thus, the usage rights and any associated fees assigned by a creator and subsequent distributor will always remain with a digital work.

[0040] The enforcement elements of the preferred embodiment are repositories. Among other things, repositories are used to store digital works, control access to digital works, bill for access to digital works and maintain the security and integrity of the system. The combination of associated usage rights and repositories enable distinct advantages over prior systems. As noted in the prior art, payment of fees are primarily for the initial access. In such approaches, once a work has been read, computational control over that copy is gone. Metaphorically, "the content genie is out of the bottle and no more fees can be billed." In contrast, the present invention never separates the fee descriptions from the work. Thus, the digital work genie only moves from one trusted bottle (repository) to another, and all uses of copies are potentially controlled and

billable.

[0041] FIG. 1 is a high level flowchart omitting various details but which demonstrates the basic operation of the present invention. Referring to FIG. 1, a creator creates a digital work, step 101. The creator will then determine appropriate usage rights and fees, associate them with the digital work, and store them in Repository 1, step 102. The determination of appropriate usage rights and fees will depend on various economic factors. The digital work remains securely in Repository 1 until a request for access is received. The request for access begins with a session initiation by another repository. Here a Repository 2 initiates a session with Repository 1, step 103. As will be described in greater detail below, this session initiation includes steps which helps to insure that the respective repositories are trustworthy. Assuming that a session can be established, Repository 2 may then request access to the Digital Work for a stated purpose, i.e., manner of use step 104. The purpose may be, for example, to print the digital work or to obtain a copy of the digital work. The purpose will correspond to a specific usage right. In any event, Repository 1 checks the usage rights associated with the digital work to determine if the access to the digital work may be granted, step 105. The check of the usage rights essentially involves a determination of whether a right associated with the access request has been associated with the digital work and if all conditions associated with the right are satisfied. If the access is denied, repository 1 terminates the session with an error message, step 106. If access is granted, repository 1 transmits the digital work to repository 2, step 107. Once the digital work has been transmitted to repository 2, repository 1 and 2 each generate billing information for the access which is transmitted to a credit server, step 108. Such double billing reporting is done to insure against attempts to circumvent the billing process.

[0042] FIG. 2 illustrates the basic interactions between repository types in the present invention. As will become apparent from FIG. 2, the various repository types will serve different functions. It is fundamental that repositories will share a core set of functionality which will enable secure and trusted communications. Referring to FIG. 2, a repository 201 represents the general instance of a repository. The repository 201 has two modes of operation; a server mode and a requester mode. When in the server mode, the repository will be receiving and processing access requests to digital works. When in the requester mode, the repository will be initiating

NVA208575.1

requests to access digital works. Repository 201 is general in the sense that its primary purpose is as an exchange medium for digital works. During the course of operation, the repository 201 may communicate with a plurality of other repositories, namely authorization repository 202, rendering repository 203 and master repository 204. Communication between repositories occurs utilizing a repository transaction protocol 205.

[0043] Communication with an authorization repository 202 may occur when a digital work being accessed has a condition requiring an authorization. Conceptually, an authorization is a digital certificate such that possession of the certificate is required to gain access to the digital work. An authorization is itself a digital work that can be moved between repositories and subjected to fees and usage rights conditions. An authorization may be required by both repositories involved in an access to a digital work.

[0044] Communication with a rendering repository 203 occurs in connection with the rendering of the content of a digital work. As will be described in greater detail below, a rendering repository is coupled with a rendering device (e.g. a printer device) to comprise a rendering system.

[0045] Communication with a master repository 205 occurs in connection with obtaining an identification certificate. Identification certificates are the means by which a repository is identified as "trustworthy." The use of identification certificates is described below with respect to the registration transaction.

[0046] FIG. 3 illustrates the repository 201 coupled to a credit server 301. The credit server 301 is a device which accumulates billing information for the repository 201. The credit server 301 communicates with repository 201 via billing transactions 302 to record billing transactions. Billing transactions are reported to a billing clearinghouse 303 by the credit server 301 on a periodic basis. The credit server 301 communicates to the billing clearinghouse 303 via clearinghouse transactions 304. The clearinghouse transactions 304 enable a secure and encrypted transmission of information to the billing clearinghouse 303.

[0047] A rendering system is generally defined as a system comprising a repository and a rendering device which can render the content of a digital work into its desired form. Examples of a rendering system may be a computer system, a digital audio system, or a printer. A rendering system has the same security features as a repository. The coupling of a rendering repository with the rendering device may occur in a manner suitable for the type of rendering device.

[0048] FIG. 4a illustrates a printer as an example of a rendering system. Referring to FIG. 4, printer system 401 has contained therein a printer repository 402 and a print device 403. It should be noted that the dashed line defining printer system 401 defines a secure system boundary. Communications within the boundary is assumed to be secure. Depending on the security level, the boundary also represents a barrier intended to provide physical integrity. The printer repository 402 is an instantiation of the rendering repository 205 of FIG. 2. The printer repository 402 will in some instances contain an ephemeral copy of a digital work which remains until it is printed out by the print engine 403. In other instances, the printer repository 402 may contain digital works such as fonts, which will remain and can be billed based on use. This design assures that all communication lines between printers and printing devices are encrypted, unless they are within a physically secure boundary. This design feature eliminates a potential "fault" point through which the digital work could be improperly obtained. The printer device 403 represents the printer components used to create the printed output.

[0049] Also illustrated in FIG. 4a is the repository 404. The repository 404 is coupled to the printer repository 402. The repository 404 represents an external repository which contains digital works.

[0050] FIG. 4b is an example of a computer system as a rendering system. A computer system may constitute a "multi-function" device since it may execute digital works (e.g. software programs) and display digital works (e.g. a digitized photograph). Logically, each rendering device can be viewed as having it's own repository, although only one physical repository is needed. Referring to FIG. 4b, a computer system 410 has contained therein a display/execution repository 411. The display/execution repository 411 is coupled to display device, 412 and

NVA208575.1

execution device 413. The dashed box surrounding the computer system 410 represents a security boundary within which communications are assumed to be secure. The display/execution repository 411 is further coupled to a credit server 414 to report any fees to be billed for access to a digital work and a repository 415 for accessing digital works stored therein.

[0051] Usage rights are associated with digital works. Thus, it is important to understand the structure of a digital work. The structure of a digital work, in particular composite digital works, may be naturally organized into an acyclic structure such as a hierarchy. For example, a magazine has various articles and photographs which may have been created and are owned by different persons. Each of the articles and photographs may represent a node in a hierarchical structure. Consequently, controls, i.e. usage rights, may be placed on each node by the creator. By enabling control and fee billing to be associated with each node, a creator of a work can be assured that the rights and fees are not circumvented.

[0052] In the currently preferred embodiment, the file information for a digital work is divided into two files: a "contents" file and a "description tree" file. From the perspective of a repository, the "contents" file is a stream of addressable bytes whose format depends completely on the interpreter or rendering engine used to play, display or print the digital work. The description tree file makes it possible to examine the rights and fees for a work without reference to the content of the digital work. It should be noted that the term description tree as used herein refers to any type of acyclic structure used to represent the relationship between the various components of a digital work.

[0053] FIG. 5 illustrates the layout of a contents file. Referring to FIG. 5, a digital work 509 is comprised of story A 510, advertisement 511, story B 512 and story C 513. It is assumed that the digital work is stored starting at a relative address of 0. Each of the parts of the digital work are stored linearly so that story A 510 is stored at approximately addresses 0-30,000, advertisement 511 at addresses 30,001-40,000, story B 512 at addresses 40,001-60,000 and story C 513 at addresses 60,001-85 K. The detail of story A 510 is illustrated in FIG. 6. Referring to FIG. 6, the story A 510 is further broken down to show text 614 stored at address 0-1500, soldier photo 615 at addresses 1501-10,000, graphics 616 stored at addresses 10,001-25,000 and sidebar

NVA208575.1

617 stored address 25,001-30,000. Note that the data in the contents file may be compressed (for saving storage) or encrypted (for security).

[0054] From FIGS. 5 and 6 it is readily observed that a digital work can be represented by its component parts as a hierarchy. The description tree for a digital work is comprised of a set of related descriptor blocks (d-blocks). The contents of each d-block is described with respect to FIG. 7. Referring to FIG. 7, a d-block 700 includes an identifier 701 which is a unique identifier for the work in the repository, a starting address 702 providing the start address of the first byte of the work, a length 703 giving the number of bytes in the work, a rights portion 704 wherein the granted usage rights and their status data are maintained, a parent pointer 705 for pointing to a parent d-block and child pointers 706 for pointing to the child d-blocks. In the currently preferred embodiment, the identifier 701 has two parts. The first part is a unique number assigned to the repository upon manufacture. The second part is a unique number assigned to the work upon creation. The rights portion 704 will contain a data structure, such as a look-up table, wherein the various information associated with a right is maintained. The information required by the respective usage rights is described in more detail below. D-blocks form a strict hierarchy. The top d-block of a work has no parent; all other d-blocks have one parent. The relationship of usage rights between parent and child d-blocks and how conflicts are resolved is described below.

[0055] A special type of d-block is a "shell" d-block. A shell d-block adds no new content beyond the content of its parts. A shell d-block is used to add rights and fee information, typically by distributors of digital works.

[0056] FIG. 8 illustrates a description tree for the digital work of FIG. 5. Referring to FIG. 8, a top d-block 820 for the digital work points to the various stories and advertisements contained therein. Here, the top d-block 820 points to d-block 821 (representing story A 510), d-block 822 (representing the advertisement 511), d-block 823 (representing story B 512) and d-block 824 (representing story C 513).

[0057] The portion of the description tree for Story A 510 is illustrated in FIG. 9. D-block 925 represents text 614, d-block 926 represents photo 615, d-block 927 represents graphics

NVA208575.1

616 by and d-block 928 represents sidebar 617.

[0058] The rights portion 704 of a descriptor block is further illustrated in FIG. 10. FIG. 10 illustrates a structure which is repeated in the rights portion 704 for each right. Referring to FIG. 10, each right will have a right code field 1050 and status information field 1052. The right code field 1050 will contain a unique code assigned to a right. The status information field 1052 will contain information relating to the state of a right and the digital work. Such information is indicated below in Table 1. The rights as stored in the rights portion 304 may typically be in numerical order based on the right code.

TABLE 1

DIGITAL WORK STATE INFORMATION

Property	Value	Use
Copies-in-use	Number	A counter of the number of copies of a work that are in use. Incremented when another copy is used; decremented when use is completed.
Loan-Period	Time-Units	Indicator of the maximum number of time-units that a document can be loaned out
Loaner-Copy	Boolean	Indicator that the current work is a loaned out copy of an authorized digital work.
Remaining-Time	Time-Units	Indicator of the remaining time of use on a metered document right.
Document- Descr	String	A string containing various identifying information about a document. The exact format of this is not specified, but it can include information such as a publisher name, author name, ISBN number, and so on.
Revenue-Owner	RO-Descr	A handle identifying a revenue owner for a digital work. This is used for reporting usage fees.
Publication- Date	Date-Descr	The date that the digital work was published.
History-list	History-Rec	A list of events recording the repositories and dates for operations that copy, transfer, backup, or restore a digital work.

[0059] The approach for representing digital works by separating description data from content assumes that parts of a file are contiguous but takes no position on the actual representation of content. In particular, it is neutral to the question of whether content representation may take an object oriented approach. It would be natural to represent content as

objects. In principle, it may be convenient to have content objects that include the billing structure and rights information that is represented in the d-blocks. Such variations in the design of the representation are possible and are viable alternatives but may introduce processing overhead, e.g. the interpretation of the objects.

[0060] Digital works are stored in a repository as part of a hierarchical file system. Folders (also termed directories and sub-directories) contain the digital works as well as other folders. Digital works and folders in a folder are ordered in alphabetical order. The digital works are typed to reflect how the files are used. Usage rights can be associated with folders so that the folder itself is treated as a digital work. Access to the folder would then be handled in the same fashion as any other digital work. As will be described in more detail below, the contents of the folder are subject to their own rights. Moreover, file management rights may be associated with the folder which define how folder contents can be managed.

[0061] The usage rights are treated as part of the digital work. As the digital work is distributed, the scope of the granted usage rights will remain the same or may be narrowed. For example, when a digital work is transferred from a document server to a repository, the usage rights may include the right to loan a copy for a predetermined period of time (called the original rights). When the repository loans out a copy of the digital work, the usage rights in the loaner copy (called the next set of rights) could be set to prohibit any further rights to loan out the copy. The basic idea is that one cannot grant more rights than they have.

[0062] The association of usage rights with a digital contact may occur in a variety of ways. If the usage rights will be the same for an entire digital work, they could be associated when the digital work is processed for deposit in the digital work server. In the case of a digital work having different usage rights for the various components, this can be done as the digital work is being created. An authoring tool or digital work assembling tool could be utilized which provides for an automated process of attaching the usage rights.

[0063] As will be described below, when a digital work is copied, transferred or loaned, a "next set of rights" can be specified. The "next set of rights" will be associated with the

NVA208575.1

digital work as it is transported.

[0064] Because each part of a digital work may have its own usage rights, there will be instances where the rights of a "contained part" are different from its parent or container part. As a result, conflict rules must be established to dictate when and how a right may be exercised. The hierarchical structure of a digital work facilitates the enforcement of such rules. A "strict" rule would be as follows: a right for a part in a digital work is sanctioned if and only if it is sanctioned for the part, for ancestor d-blocks containing the part and for all descendent d-blocks. By sanctioned, it is meant that (1) each of the respective parts must have the right, and (2) any conditions for exercising the right are satisfied.

[0065] It also possible to implement the present invention using a more lenient rule. In the more lenient rule, access to the part may be enabled to the descendent parts which have the right, but access is denied to the descendents which do not.

[0066] Example of applying both the strict rule and lenient is illustrated with reference to FIG. 11. Referring to FIG. 11, a root d-block 1101 has child d-blocks 1102-1105. In this case, root d-block represents a magazine, and each of the child d-blocks 1102-1105 represent articles in the magazine. Suppose that a request is made to PRINT the digital work represented by root d-block 1101 wherein the strict rule is followed. The rights for the root d-block 1101 and child d-blocks 1102-1105 are then examined. Root d-block 1101 and child d-blocks 1102 and 1105 have been granted PRINT rights. Child d-block 1103 has not been granted PRINT rights and child d-block 1104 has PRINT rights conditioned on payment of a usage fee.

[0067] Under the strict rule the PRINT right cannot be exercised because the child d-block does not have the PRINT right. Under the lenient rule, the result would be different. The digital works represented by child d-blocks 1102 and 1105 could be printed and the digital work represented by d-block 1104 could be printed so long as the usage fee is paid. Only the digital work represented by d-block 1103 could not be printed. This same result would be accomplished under the strict rule if the requests were directed to each of the individual digital works.

[0068] The present invention supports various combinations of allowing and

NVA208575.1

disallowing access. Moreover, as will be described below, the usage rights grammar permits the owner of a digital work to specify if constraints may be imposed on the work by a container part. The manner in which digital works may be sanctioned because of usage rights conflicts would be implementation specific and would depend on the nature of the digital works.

[0069] Many of the powerful functions of repositories--such as their ability to "loan" digital works or automatically handle the commercial reuse of digital works--are possible because they are trusted systems. The systems are trusted because they are able to take responsibility for fairly and reliably carrying out the commercial transactions. That the systems can be responsible ("able to respond") is fundamentally an issue of integrity. The integrity of repositories has three parts: physical integrity, communications integrity, and behavioral integrity.

[0070] Physical integrity refers to the integrity of the physical devices themselves. Physical integrity applies both to the repositories and to the protected digital works. Thus, the higher security classes of repositories themselves may have sensors that detect when tampering is attempted on their secure cases. In addition to protection of the repository itself, the repository design protects access to the content of digital works. In contrast with the design of conventional magnetic and optical devices--such as floppy disks, CD-ROMs, and videotapes--repositories never allow non-trusted systems to access the works directly. A maker of generic computer systems cannot guarantee that their platform will not be used to make unauthorized copies. The manufacturer provides generic capabilities for reading and writing information, and the general nature of the functionality of the general computing device depends on it. Thus, a copy program can copy arbitrary data. This copying issue is not limited to general purpose computers. It also arises for the unauthorized duplication of entertainment "software" such as video and audio recordings by magnetic recorders. Again, the functionality of the recorders depends on their ability to copy and they have no means to check whether a copy is authorized. In contrast, repositories prevent access to the raw data by general devices and can test explicit rights and conditions before copying or otherwise granting access. Information is only accessed by protocol between trusted repositories.

[0071] Communications integrity refers to the integrity of the communications

NVA208575.1

channels between repositories. Roughly speaking, communications integrity means that repositories cannot be easily fooled by "telling them lies." Integrity in this case refers to the property that repositories will only communicate with other devices that are able to present proof that they are certified repositories, and furthermore, that the repositories monitor the communications to detect "impostors" and malicious or accidental interference. Thus the security measures involving encryption, exchange of digital certificates, and nonces described below are all security measures aimed at reliable communication in a world known to contain active adversaries.

[0072] Behavioral integrity refers to the integrity in what repositories do. What repositories do is determined by the software that they execute. The integrity of the software is generally assured only by knowledge of its source. Restated, a user will trust software purchased at a reputable computer store but not trust software obtained off a random (insecure) server on a network. Behavioral integrity is maintained by requiring that repository software be certified and be distributed with proof of such certification, i.e. a digital certificate. The purpose of the certificate is to authenticate that the software has been tested by an authorized organization, which attests that the software does what it is supposed to do and that it does not compromise the behavioral integrity of a repository. If the digital certificate cannot be found in the digital work or the master repository which generated the certificate is not known to the repository receiving the software, then the software cannot be installed.

[0073] In the description of FIG. 2, it was indicated that repositories come in various forms. All repositories provide a core set of services for the transmission of digital works. The manner in which digital works are exchanged is the basis for all transaction between repositories. The various repository types differ in the ultimate functions that they perform. Repositories may be devices themselves, or they may be incorporated into other systems. An example is the rendering repository 205 of FIG. 2.

[0074] A repository will have associated with it a repository identifier. Typically, the repository identifier would be a unique number assigned to the repository at the time of manufacture. Each repository will also be classified as being in a particular security class.

NVA208575.1

Certain communications and transactions may be conditioned on a repository being in a particular security class. The various security classes are described in greater detail below.

[0075] As a prerequisite to operation, a repository will require possession of an identification certificate. Identification certificates are encrypted to prevent forgery and are issued by a Master repository. A master repository plays the role of an authorization agent to enable repositories to receive digital works. Identification certificates must be updated on a periodic basis. Identification certificates are described in greater detail below with respect to the registration transaction.

[0076] A repository has both a hardware and functional embodiment. The functional embodiment is typically software executing on the hardware embodiment. Alternatively, the functional embodiment may be embedded in the hardware embodiment such as an Application Specific Integrated Circuit (ASIC) chip.

[0077] The hardware embodiment of a repository will be enclosed in a secure housing which if compromised, may cause the repository to be disabled. The basic components of the hardware embodiment of a repository are described with reference to FIG. 12. Referring to FIG. 12, a repository is comprised of a processing means 1200, storage system 1207, clock 1205 and external interface 1206. The processing means 1200 is comprised of a processor element 1201 and processor memory 1202. The processing means 1201 provides controller, repository transaction and usage rights transaction functions for the repository. Various functions in the operation of the repository such as decryption and/or decompression of digital works and transaction messages are also performed by the processing means 1200. The processor element 1201 may be a microprocessor or other suitable computing component. The processor memory 1202 would typically be further comprised of Read Only Memories (ROM) and Random Access Memories (RAM). Such memories would contain the software instructions utilized by the processor element 1201 in performing the functions of the repository.

[0078] The storage system 1207 is further comprised of descriptor storage 1203 and content storage 1204. The description tree storage 1203 will store the description tree for the

NVA208575.1

digital work and the content storage will store the associated content. The description tree storage 1203 and content storage 1204 need not be of the same type of storage medium, nor are they necessarily on the same physical device. So for example, the descriptor storage 1203 may be stored on a solid state storage (for rapid retrieval of the description tree information), while the content storage 1204 may be on a high capacity storage such as an optical disk.

[0079] The clock 1205 is used to time-stamp various time based conditions for usage rights or for metering usage fees which may be associated with the digital works. The clock 1205 will have an uninterruptable power supply, e.g. a battery, in order to maintain the integrity of the time-stamps. The external interface means 1206 provides for the signal connection to other repositories and to a credit server. The external interface means 1206 provides for the exchange of signals via such standard interfaces such as RS-232 or Personal Computer Manufacturers Card Industry Association (PCMCIA) standards, or FDDI. The external interface means 1206 may also provide network connectivity.

[0080] The functional embodiment of a repository is described with reference to FIG. 13. Referring to FIG. 13, the functional embodiment is comprised of an operating system 1301, core repository services 1302, usage transaction handlers 1303, repository specific functions, 1304 and a user interface 1305. The operating system 1301 is specific to the repository and would typically depend on the type of processor being used. The operating system 1301 would also provide the basic services for controlling and interfacing between the basic components of the repository.

[0081] The core repository services 1302 comprise a set of functions required by each and every repository. The core repository services 1302 include the session initiation transactions which are defined in greater detail below. This set of services also includes a generic ticket agent which is used to "punch" a digital ticket and a generic authorization server for processing authorization specifications. Digital tickets and authorizations are specific mechanisms for controlling the distribution and use of digital works and are described in more detail below. Note that coupled to the core repository services are a plurality of identification certificates 1306. The identification certificates 1306 are required to enable the use of the

repository.

[0082] The usage transactions handler 1303 comprise functionality for processing access requests to digital works and for billing fees based on access. The usage transactions supported will be different for each repository type. For example, it may not be necessary for some repositories to handle access requests for digital works.

[0083] The repository specific functionality 1304 comprises functionality that is unique to a repository. For example, the master repository has special functionality for issuing digital certificates and maintaining encryption keys. The repository specific functionality 1304 would include the user interface implementation for the repository.

[0084] For some digital works the losses caused by any individual instance of unauthorized copying is insignificant and the chief economic concern lies in assuring the convenience of access and low-overhead billing. In such cases, simple and inexpensive handheld repositories and network-based workstations may be suitable repositories, even though the measures and guarantees of security are modest.

[0085] At the other extreme, some digital works such as a digital copy of a first run movie or a bearer bond or stock certificate would be of very high value so that it is prudent to employ caution and fairly elaborate security measures to ensure that they are not copied or forged. A repository suitable for holding such a digital work could have elaborate measures for ensuring physical integrity and for verifying authorization before use.

[0086] By arranging a universal protocol, all kinds of repositories can communicate with each other in principle. However, creators of some works will want to specify that their works will only be transferred to repositories whose level of security is high enough. For this reason, document repositories have a ranking system for classes and levels of security. The security classes in the currently preferred embodiment are described in Table 2.

TABLE 2

NVA208575.1

REPOSITORY SECURITY LEVELS

Level	Description of Security
0 Open system.	Document transmission is unencrypted. No digital certificate is required for identification. The security of the system depends mostly on user honesty, since only modest knowledge may be needed to circumvent the security measures. The repository has no provisions for preventing unauthorized programs from running and accessing or copying files. The system does not prevent the use of removable storage and does not encrypt stored files.
1 Minimal security.	Like the previous class except that stored files are minimally encrypted, including ones on removable storage.
2 Basic security.	Like the previous class except that special tools and knowledge are required to compromise the programming, the contents of the repository, or the state of the clock. All digital communications are encrypted. A digital certificate is provided as identification. Medium level encryption is used. Repository identification number is unforgeable.
3 General security.	Like the previous class plus the requirement of special tools are needed to compromise the physical integrity of the repository and that modest encryption is used on all transmissions. Password protection is required to use the local user interface. The digital clock system cannot be reset without authorization. No works would be stored on removable storage. When executing works as programs, it runs them in their own address space and does not give them direct access to any file storage or other memory containing system code or works. They can access works only through the transmission transaction protocol.
4 High Security	Like the previous class except that high level encryption is used on all communications. Sensors are used to record attempts at physical and electronic tampering. After such tampering, the repository will not perform other transactions until it has reported such tampering to a designated server.
5 Tamper Resistant	Like the previous class except that if the physical or digital attempts at tampering exceed some preset thresholds that threaten the physical integrity of the repository or the integrity of digital and cryptographic barriers, then the repository will save only document description records of history but will erase or destroy any digital identifiers that could be misused if released to an unscrupulous party. It also modifies any certificates of authenticity to indicate that the physical system has been compromised. It also erases the contents of designated documents.
6	Like the previous class except that the repository will attempt wireless communication to report tampering and will employ noisy alarms.

NVA208575.1

10	This would correspond to a very high level of security. This server would maintain constant communications to remote security systems reporting transactions, sensor readings, and attempts to circumvent security.

[0087] The characterization of security levels described in Table 2 is not intended to be fixed. More important is the idea of having different security levels for different repositories. It is anticipated that new security classes and requirements will evolve according to social situations and changes in technology.

[0088] A user interface is broadly defined as the mechanism by which a user interacts with a repository in order to invoke transactions to gain access to a digital work, or exercise usage rights. As described above, a repository may be embodied in various forms. The user interface for a repository will differ depending on the particular embodiment. The user interface may be a graphical user interface having icons representing the digital works and the various transactions that may be performed. The user interface may be a generated dialog in which a user is prompted for information.

[0089] The user interface itself need not be part of the repository. As a repository may be embedded in some other device, the user interface may merely be a part of the device in which the repository is embedded. For example, the repository could be embedded in a "card" that is inserted into an available slot in a computer system. The user interface may be combination of a display, keyboard, cursor control device and software executing on the computer system.

[0090] At a minimum, the user interface must permit a user to input information such as access requests and alpha numeric data and provide feedback as to transaction status. The user interface will then cause the repository to initiate the suitable transactions to service the request. Other facets of a particular user interface will depend on the functionality that a repository will provide.

In the preferred embodiment, fees may be associated with the exercise of a right. The requirement for payment of fees is described with each version of a usage right in the usage rights language. The recording and reporting of such fees is performed by the credit server. One of the capabilities enabled by associating fees with rights is the possibility of supporting a wide

range of charging models. The simplest model, used by conventional software, is that there is a single fee at the time of purchase, after which the purchaser obtains unlimited rights to use the work as often and for as long as he or she wants. Alternative models, include metered use and variable fees. A single work can have different fees for different uses. For example, viewing a photograph on a display could have different fees than making a hardcopy or including it in a newly created work. A key to these alternative charging models is to have a low overhead means of establishing fees and accounting for credit on these transactions.

[0091] A credit server is a computational system that reliably authorizes and records these transactions so that fees are billed and paid. The credit server reports fees to a billing clearinghouse. The billing clearinghouse manages the financial transactions as they occur. As a result, bills may be generated and accounts reconciled. Preferably, the credit server would store the fee transactions and periodically communicate via a network with billing clearinghouse for reconciliation. In such an embodiment, communications with the billing clearinghouse would be encrypted for integrity and security reasons. In another embodiment, the credit server acts as a "debit card" where transactions occur in "real-time" against a user account.

[0092] A credit server is comprised of memory, a processing means, a clock, and interface means for coupling to a repository and a financial institution (e.g. a modem). The credit server will also need to have security and authentication functionality. These elements are essentially the same elements as those of a repository. Thus, a single device can be both a repository and a credit server, provided that it has the appropriate processing elements for carrying out the corresponding functions and protocols. Typically, however, a credit server would be a card-sized system in the possession of the owner of the credit. The credit server is coupled to a repository and would interact via financial transactions as described below. Interactions with a financial institution may occur via protocols established by the financial institutions themselves.

[0093] In the currently preferred embodiment credit servers associated with both the server and the repository report the financial transaction to the billing clearinghouse. For example, when a digital work is copied by one repository to another for a fee, credit servers

NVA208575.1

coupled to each of the repositories will report the transaction to the billing clearinghouse. This is desirable in that it insures that a transaction will be accounted for in the event of some break in the communication between a credit server and the billing clearinghouse. However, some implementations may embody only a single credit server reporting the transaction to minimize transaction processing at the risk of losing some transactions.

[0094] The preferred embodiment uses statements in a high level "usage rights language" to define rights associated with digital works and their parts. Usage rights statements are interpreted by repositories and are used to determine what transactions, i.e., means of use, can be successfully carried out for a digital work and also to determine parameters for those transactions. For example, sentences in the language determine whether a given digital work can be copied, when and how it can be used, and what fees (if any) are to be charged for that use. Once the usage rights statements are generated, they are encoded in a suitable form for accessing during the processing of transactions.

[0095] Defining usage rights in terms of a language in combination with the hierarchical representation of a digital work enables the support of a wide variety of distribution and fee schemes. An example is the ability to associate multiple versions of a right with a work. So a creator may associate PRINT right to make 5 copies for \$10.00 and a PRINT right to make unlimited copies for \$100.00. A purchaser may then choose which option best fits his needs. Another example is that rights and fees are additive. So in the case of a composite work, the rights and fees of each of the components works is used in determining the rights and fees for the work as a whole. Other features and benefits of the usage rights language will become apparent in the description of distribution and use scenarios provided below.

[0096] The basic contents of a right are illustrated in FIG. 14. Referring to FIG. 14, a right 1450 has a transactional component 1451 and a specifications component 1452. A right 1450 has a label (e.g. COPY or PRINT) which indicate the manners of use that are embodied by the right. The transactional component 1451 corresponds to a particular way in which a digital work may be used or distributed. The transactional component 1451 is typically embodied in software instructions in a repository which implement the manners of use for the right. The

NVA208575.1

specifications components 1452 are used to specify conditions which must be satisfied prior to the right being exercised or to designate various transaction related parameters. In the currently preferred embodiment, these specifications include copy count 1453, Fees and Incentives 1454, Time 1455, Access and Security 1456 and Control 1457. Each of these specifications will be described in greater detail below with respect to the language grammar elements.

[0097] The usage rights language is based on the grammar described below. A grammar is a convenient means for defining valid sequence of symbols for a language. In describing the grammar the notation "[a.vertline.b.vertline.c]" is used to indicate distinct choices among alternatives. In this example, a sentence can have either an "a", "b" or "c". It must include exactly one of them. The braces { } are used to indicate optional items. Note that brackets, bars and braces are used to describe the language of usage rights sentences but do not appear in actual sentences in the language.

[0098] In contrast, parentheses are part of the usage rights language. Parentheses are used to group items together in lists. The notation (x*) is used to indicate a variable length list, that is, a list containing one or more items of type x. The notation (x)* is used to indicate a variable number of lists containing x.

[0099] Keywords in the grammar are words followed by colons. Keywords are a common and very special case in the language. They are often used to indicate a single value, typically an identifier. In many cases, the keyword and the parameter are entirely optional. When a keyword is given, it often takes a single identifier as its value. In some cases, the keyword takes a list of identifiers. In the usage rights language, time is specified in an hours:minutes:seconds (or hh:mm:ss) representation. Time zone indicators, e.g. PDT for Pacific Daylight Time, may also be specified. Dates are represented as year/month/day (or YYYY/MMM/DD). Note that these time and date representations may specify moments in time or units of time Money units are specified in terms of dollars.

Finally, in the usage rights language, various "things" will need to interact with each other. For example, an instance of a usage right may specify a bank account, a digital ticket, etc. Such things need to be identified and are specified herein using the suffix "-ID."

[0100] The Usage Rights Grammar of the preferred embodiment is listed in its entirety in FIG. 15 and is described below.

[0101] Grammar element 1501 "Digital Work Rights:=(Rights*)" define the digital work rights as a set of rights. The set of rights associated with a digital work define how that digital work may be transferred, used, performed or played. A set of rights will attach to the entire digital work and in the case of compound digital works, each of the components of the digital work. The usage rights of components of a digital may be different.

[0102] Grammar element 1502 "Right:=(Right-Code {Copy-Count} {Control-Spec} {Time-Spec} {Access-Spec} {Fee-Spec})" enumerates the content of a right. Each usage right must specify a right code. Each right may also optionally specify conditions which must be satisfied before the right can be exercised. These conditions are copy count, control, time, access and fee conditions. In the currently preferred embodiment, for the optional elements, the following defaults apply: copy count equals 1, no time limit on the use of the right, no access tests or a security level required to use the right and no fee is required. These conditions will each be described in greater detail below.

[0103] It is important to note that a digital work may have multiple versions of a right, each having the same right code. The multiple version would provide alternative conditions and fees for accessing the digital work.

[0104] Grammar element 1503 "Right-Code: =Render-Code.vertline.Transport-Code.vertline.File-Management-Code.vertline.Derivative-Works-Code Configuration-Code" distinguishes each of the specific rights into a particular right type (although each right is identified by distinct right codes). In this way, the grammar provides a catalog of possible rights that can be associated with parts of digital works. In the following, rights are divided into categories for convenience in describing them.

[0105] Grammar element 1504 "Render-Code:=[Play: {Player: Player-ID}.vertline.Print:

NVA208575.1

{Printer: Printer-ID}]" lists a category of rights all involving the making of ephemeral, transitory, or non-digital copies of the digital work. After use the copies are erased.

[0106] Play A process of rendering or performing a digital work on some processor. This includes such things as playing digital movies, playing digital music, playing a video game, running a computer program, or displaying a document on a display.

[0107] Print To render the work in a medium that is not further protected by usage rights, such as printing on paper.

[0108] Grammar element 1505 "Transport-Code:=[Copy.vertline.Transfer.vertline.Loan {Remaining-Rights: Next-Set-of-Rights}]{(Next-Copy-Rights: Next-Set of Rights)}" lists a category of rights involving the making of persistent, usable copies of the digital work on other repositories. The optional Next-Copy-Rights determine the rights on the work after it is transported. If this is not specified, then the rights on the transported copy are the same as on the original. The optional Remaining-Rights specify the rights that remain with a digital work when it is loaned out. If this is not specified, then the default is that no rights can be exercised when it is loaned out.

[0109] Copy Make a new copy of a work

[0110] Transfer Moving a work from one repository to another.

[0111] Loan Temporarily loaning a copy to another repository for a specified period of time.

[0112] Grammar element 1506 "File-Management-Code:=Backup {Back-Up-Copy-Rights: Next-Set-of Rights}.vertline.Restore.vertline.Delete.vertline.Folder.vertline.Director y {Name:Hide-Local.vertline.Hide-Remote} {Parts: Hide-Local.vertline.Hide-Remote}" lists a category of rights involving operations for file management, such as the making of backup copies to protect the copy owner against catastrophic equipment failure.

NVA208575.1

[0113] Many software licenses and also copyright law give a copy owner the right to make backup copies to protect against catastrophic failure of equipment. However, the making of uncontrolled backup copies is inherently at odds with the ability to control usage, since an uncontrolled backup copy can be kept and then restored even after the authorized copy was sold.

[0114] The File management rights enable the making and restoring of backup copies in a way that respects usage rights, honoring the requirements of both the copy owner and the rights grantor and revenue owner. Backup copies of work descriptions (including usage rights and fee data) can be sent under appropriate protocol and usage rights control to other document repositories of sufficiently high security. Further rights permit organization of digital works into folders which themselves are treated as digital works and whose contents may be "hidden" from a party seeking to determine the contents of a repository.

[0115] Backup To make a backup copy of a digital work as protection against media failure.

[0116] Restore To restore a backup copy of a digital work.

[0117] Delete To delete or erase a copy of a digital work.

[0118] Folder To create and name folders, and to move files and folders between folders.

[0119] Directory To hide a folder or it's contents.

[0120] Grammar element 1507 "Derivative-Works-Code:
[Extract.vertline.Embed.vertline.Edit {Process: Process-ID}] {Next-Copy-Rights: Next-Set-of Rights}" lists a category of rights involving the use of a digital work to create new works.

[0121] Extract To remove a portion of a work, for the purposes of creating a new work.

- [0122] Embed To include a work in an existing work.
- [0123] Edit To alter a digital work by copying, selecting and modifying portions of an existing digital work.
- [0124] Grammar element 1508 "Configuration-Code:=Install.vertline.Uninstall" lists a category of rights for installing and uninstalling software on a repository (typically a rendering repository.) This would typically occur for the installation of a new type of player within the rendering repository.
- [0125] Install: To install new software on a repository.
- [0126] Uninstall: To remove existing software from a repository.
- [0127] Grammar element 1509 "Next-Set-of-Rights:={{(Add: Set-Of-Rights)} {(Delete: Set-Of-Rights)} {(Replace: Set-Of-Rights)} {(Keep: Set-Of-Rights)}}" defines how rights are carried forward for a copy of a digital work. If the Next-Copy-Rights is not specified, the rights for the next copy are the same as those of the current copy. Otherwise, the set of rights for the next copy can be specified. Versions of rights after Add: are added to the current set of rights. Rights after Delete: are deleted from the current set of rights. If only right codes are listed after Delete:, then all versions of rights with those codes are deleted. Versions of rights after Replace: subsume all versions of rights of the same type in the current set of rights.
- [0128] If Remaining-Rights is not specified, then there are no rights for the original after all Loan copies are loaned out. If Remaining-Rights is specified, then the Keep: token can be used to simplify the expression of what rights to keep behind. A list of right codes following keep means that all of the versions of those listed rights are kept in the remaining copy. This specification can be overridden by subsequent Delete: or Replace: specifications.
- [0129] Copy Count Specification. For various transactions, it may be desirable to provide some limit as to the number of "copies" of the work which may be exercised

NVA208575.1

simultaneously for the right. For example, it may be desirable to limit the number of copies of a digital work that may be loaned out at a time or viewed at a time.

[0130] Grammar element 1510 "Copy-Count:=(Copies: positive-integer.vertline.0.vertline.unlimited)" provides a condition which defines the number of "copies" of a work subject to the right. A copy count can be 0, a fixed number, or unlimited. The copy-count is associated with each right, as opposed to there being just a single copy-count for the digital work. The Copy-Count for a right is decremented each time that a right is exercised. When the Copy-Count equals zero, the right can no longer be exercised. If the Copy-Count is not specified, the default is one.

[0131] Rights and fees depend in general on rights granted by the creator as well as further restrictions imposed by later distributors. Control specifications deal with interactions between the creators and their distributors governing the imposition of further restrictions and fees. For example, a distributor of a digital work may not want an end consumer of a digital work to add fees or otherwise profit by commercially exploiting the purchased digital work.

[0132] Grammar element 1511 "Control-Spec:=(Control: {Restrictable.vertline.Unrestrictable} {Unchargeable.vertline.Chargeable})" provides a condition to specify the effect of usage rights and fees of parents on the exercise of the right. A digital work is restrictable if higher level d-blocks can impose further restrictions (time specifications and access specifications) on the right. It is unrestrictable if no further restrictions can be imposed. The default setting is restrictable. A right is unchargeable if no more fees can be imposed on the use of the right. It is chargeable if more fees can be imposed. The default is chargeable.

[0133] It is often desirable to assign a start date or specify some duration as to when a right may be exercised. Grammar element 1512 "Time-Spec: =({Fixed-Interval.vertline.Sliding-Interval.vertline.Meter-Time} Until: Expiration-Date)" provides for specification of time conditions on the exercise of a right. Rights may be granted for a specified time. Different kinds of time specifications are appropriate for different kinds of rights. Some rights may be exercised

NVA208575.1

during a fixed and predetermined duration. Some rights may be exercised for an interval that starts the first time that the right is invoked by some transaction. Some rights may be exercised or are charged according to some kind of metered time, which may be split into separate intervals. For example, a right to view a picture for an hour might be split into six ten minute viewings or four fifteen minute viewings or twenty three minute viewings.

[0134] The terms "time" and "date" are used synonymously to refer to a moment in time. There are several kinds of time specifications. Each specification represents some limitation on the times over which the usage right applies. The Expiration-Date specifies the moment at which the usage right ends. For example, if the Expiration-Date is "Jan. 1, 1995," then the right ends at the first moment of 1995. If the Expiration-Date is specified as *forever*, then the rights are interpreted as continuing without end. If only an expiration date is given, then the right can be exercised as often as desired until the expiration date.

[0135] Grammar element 1513 "Fixed-Interval:=From: Start-Time" is used to define a predetermined interval that runs from the start time to the expiration date.

[0136] Grammar element 1514 "Sliding-Interval:=Interval: Use-Duration" is used to define an indeterminate (or "open") start time. It sets limits on a continuous period of time over which the contents are accessible. The period starts on the first access and ends after the duration has passed or the expiration date is reached, whichever comes first. For example, if the right gives 10 hours of continuous access, the use-duration would begin when the first access was made and end 10 hours later.

[0137] Grammar element 1515 "Meter-Time:=Time-Remaining: Remaining-Use" is used to define a "meter time," that is, a measure of the time that the right is actually exercised. It differs from the Sliding-Interval specification in that the time that the digital work is in use need not be continuous. For example, if the rights guarantee three days of access, those days could be spread out over a month. With this specification, the rights can be exercised until the meter time is exhausted or the expiration date is reached, whichever comes first.

Remaining-Use:=Time-Unit

Start-Time:=Time-Unit

Use-Duration:=Time-Unit

[0138] All of the time specifications include time-unit specifications in their ultimate instantiation.

[0139] The present invention provides for various security mechanisms to be introduced into a distribution or use scheme. Grammar element 1516 "Access-Spec:=({SC: Security-Class} {Authorization: Authorization-ID*} {Other-Authorization: Authorization-ID*} {Ticket: Ticket-ID})" provides a means for restricting access and transmission. Access specifications can specify a required security class for a repository to exercise a right or a required authorization test that must be satisfied.

[0140] The keyword "SC:" is used to specify a minimum security level for the repositories involved in the access. If "SC:" is not specified, the lowest security level is acceptable.

[0141] The optional "Authorization:" keyword is used to specify required authorizations on the same repository as the work. The optional "Other-Authorization:" keyword is used to specify required authorizations on the other repository in the transaction.

[0142] The optional "Ticket:" keyword specifies the identity of a ticket required for the transaction. A transaction involving digital tickets must locate an appropriate digital ticket agent who can "punch" or otherwise validate the ticket before the transaction can proceed. Tickets are described in greater detail below.

[0143] In a transaction involving a repository and a document server, some usage rights may require that the repository have a particular authorization, that the server have some

NVA208575.1

authorization, or that both repositories have (possibly different) authorizations. Authorizations themselves are digital works (hereinafter referred to as an authorization object) that can be moved between repositories in the same manner as other digital works. Their copying and transferring is subject to the same rights and fees as other digital works. A repository is said to have an authorization if that authorization object is contained within the repository.

[0144] In some cases, an authorization may be required from a source other than the document server and repository. An authorization object referenced by an Authorization-ID can contain digital address information to be used to set up a communications link between a repository and the authorization source. These are analogous to phone numbers. For such access tests, the communication would need to be established and authorization obtained before the right could be exercised.

[0145] For one-time usage rights, a variant on this scheme is to have a digital ticket. A ticket is presented to a digital ticket agent, whose type is specified on the ticket. In the simplest case, a certified generic ticket agent, available on all repositories, is available to "punch" the ticket. In other cases, the ticket may contain addressing information for locating a "special" ticket agent. Once a ticket has been punched, it cannot be used again for the same kind of transaction (unless it is unpunched or refreshed in the manner described below.) Punching includes marking the ticket with a timestamp of the date and time it was used. Tickets are digital works and can be copied or transferred between repositories according to their usage rights. In the currently preferred embodiment, a "punched" ticket becomes "unpunched" or "refreshed" when it is copied or extracted. The Copy and Extract operations save the date and time as a property of the digital ticket. When a ticket agent is given a ticket, it can simply check whether the digital copy was made after the last time that it was punched. Of course, the digital ticket must have the copy or extract usage rights associated therewith.

[0146] The capability to unpunch a ticket is important in the following cases:

[0147] A digital work is circulated at low cost with a limitation that it can be used only once.

NVA208575.1

[0148] A digital work is circulated with a ticket that can be used once to give discounts on purchases of other works.

[0149] A digital work is circulated with a ticket (included in the purchase price and possibly embedded in the work) that can be used for a future upgrade.

[0150] In each of these cases, if a paid copy is made of the digital work (including the ticket) the new owner would expect to get a fresh (unpunched) ticket, whether the copy seller has used the work or not. In contrast, loaning a work or simply transferring it to another repository should not revitalize the ticket.

[0151] The billing for use of a digital work is often fundamental to a commercial distribution system. Grammar Element 1517 "Fee-Spec:={Scheduled-Discount} Regular-Fee-Spec.vertline.Scheduled-Fee-Spec.vertline.Markup-Spec" provides a range of options for billing for the use of digital works.

[0152] A key feature of this approach is the development of low-overhead billing for transactions in potentially small amounts. Thus, it becomes feasible to collect fees of only a few cents each for thousands of transactions.

[0153] The grammar differentiates between uses where the charge is per use from those where it is metered by the time unit. Transactions can support fees that the user pays for using a digital work as well as incentives paid by the right grantor to users to induce them to use or distribute the digital work.

[0154] The optional scheduled discount refers to the rest of the fee specification--discounting it by a percentage over time. If it is not specified, then there is no scheduled discount. Regular fee specifications are constant over time. Scheduled fee specifications give a schedule of dates over which the fee specifications change. Markup specifications are used in d-blocks for adding a percentage to the fees already being charged.

NVA208575.1

[0155] Grammar Element 1518 "Scheduled-Discount:=(Scheduled-Discount: (Time-Spec Percentage)*)" A Scheduled-Discount is a essentially a scheduled modifier of any other fee specification for this version of the right of the digital work. (It does not refer to children or parent digital works or to other versions of rights.). It is a list of pairs of times and percentages. The most recent time in the list that has not yet passed at the time of the transaction is the one in effect. The percentage gives the discount percentage. For example, the number 10 refers to a 10% discount.

[0156] Grammar Element 1519 "Regular-Fee-Spec:={Fee:.vertline.Incentive:} [Per-Use-Spec.vertline.Metered-Rate-Spec.vertline.Best-Price-Spec.vertline .Call-For-Price-Spec] {Min: Money-Unit Per: Time-Spec} {Max: Money-Unit Per: Time-Spec} To: Account-ID)" provides for several kinds of fee specifications.

[0157] Fees are paid by the copy-owner/user to the revenue-owner if Fee: is specified. Incentives are paid by the revenue-owner to the user if Incentive: is specified. If the Min: specification is given, then there is a minimum fee to be charged per time-spec unit for its use. If the Max: specification is given, then there is a maximum fee to be charged per time-spec for its use. When Fee: is specified, Account-ID identifies the account to which the fee is to be paid. When Incentive: is specified, Account-ID identifies the account from which the fee is to be paid.

[0158] Grammar element 1520 "Per-Use-Spec:=Per-Use: Money-unit" defines a simple fee to be paid every time the right is exercised, regardless of how much time the transaction takes.

[0159] Grammar element 1521 "Metered-Rate-Spec:=Metered: Money-Unit Per: Time-Spec" defines a metered-rate fee paid according to how long the right is exercised. Thus, the time it takes to complete the transaction determines the fee.

[0160] Grammar element 1522 "Best-Price-Spec:=Best-Price: Money-unit Max: Money-unit" is used to specify a best-price that is determined when the account is settled. This

NVA208575.1

specification is to accommodate special deals, rebates, and pricing that depends on information that is not available to the repository. All fee specifications can be combined with tickets or authorizations that could indicate that the consumer is a wholesaler or that he is a preferred customer, or that the seller be authorized in some way. The amount of money in the Max: field is the maximum amount that the use will cost. This is the amount that is tentatively debited from the credit server. However, when the transaction is ultimately reconciled, any excess amount will be returned to the consumer in a separate transaction.

[0161] Grammar element 1523 "Call-For-Price-Spec:=Call-For-Price " is similar to a "Best-Price-Spec" in that it is intended to accommodate cases where prices are dynamic. A Call-For-Price Spec requires a communication with a dealer to determine the price. This option cannot be exercised if the repository cannot communicate with a dealer at the time that the right is exercised. It is based on a secure transaction whereby the dealer names a price to exercise the right and passes along a deal certificate which is referenced or included in the billing process.

[0162] Grammar element 1524 "Scheduled-Fee-Spec:=(Schedule: (Time-Spec Regular-Fee-Spec)*)" is used to provide a schedule of dates over which the fee specifications change. The fee specification with the most recent date not in the future is the one that is in effect. This is similar to but more general than the scheduled discount. It is more general, because it provides a means to vary the fee agreement for each time period.

[0163] Grammar element 1525 "Markup-Spec:=Markup: percentage To: Account-ID" is provided for adding a percentage to the fees already being charged. For example, a 5% markup means that a fee of 5% of cumulative fee so far will be allocated to the distributor. A markup specification can be applied to all of the other kinds of fee specifications. It is typically used in a shell provided by a distributor. It refers to fees associated with d-blocks that are parts of the current d-block. This might be a convenient specification for use in taxes, or in distributor overhead.

[0164] Usage rights can be grouped in predetermined sets that are commonly opposite. The following are examples of sets of usage rights

NVA208575.1

[0165] ((Play) (Transfer(SC: 3)) (Delete) This work can be played without requirements for fee or authorization on any rendering system. It can be transferred to any other repository of security level 3 or greater. It can be deleted.

[0166] ((Play) (Transfer (SC: 3)) (Delete) (Backup) (Restore (Fee: Per-Use: \$5 To: Account-ID-678)))

[0167] Same as the previous example plus rights for backup and restore. The work can be backed up without fee. It can be restored for a \$5 fee payable to the account described by Account-ID-678.

[0168] ((Play) (Transfer (SC: 3)) (Copy (SC:3)(Fee: Per-Use: \$5 To: Account-ID-678)) (Delete (Incentive: Per-Use: \$2.50 To: Account-ID-678)))

[0169] This work can be played, transferred, copied, or deleted. Copy or transfer operations can take place only with repositories of security level three or greater. The fee to make a copy is \$5 payable to Account-ID-678. If a copy is deleted, then an incentive of \$2.50 is paid to the former copy owner.

[0170] ((Play) (Transfer (SC: 3)) Copy (SC: 3) (Fee: Per-Use: \$10 To: Account-ID-678)) Delete) (Backup) (Restore (SC: 3) (Fee: Per-Use: \$5 To: Account-ID-678)))

[0171] Same as the previous example plus fees for copying. The work can be copied digitally for a fee of \$10 payable to Account-ID-678. The repository on which the work is copied or restored must be at security level 3 or greater.

[0172] ((Play) (Transfer (SC: 3)) (Copy Authorization: License-123-ID (SC: 3))) The digital work can be played, transferred, or copied. Copies or transfers must be on repositories of security level 3 or greater. Copying requires the license License-123-ID issued to the copying

repository. None of the rights require fees.

[0173] ((Play) (Print Printer: Printer-567-ID (Fee: Per-Use: \$1 To: Account-ID-678)))

This work can be played for free. It can be printed on any printer with the identifier Printer-567-ID for a fee of \$1 payable to the account described by Account-ID-678.

[0174] ((Play Player: Player-876-ID) (From: 94/02/14 Until: 95/02/15) (Fee: Metered:

\$0.01 Per: 0:1:0 Min: \$0.25 Per: 0/1/0 To: Account-ID-567)) This work can be played on any player holding the ID Player-876-ID. The time of this right is from Feb. 14, 1994 until Feb. 15, 1995. The fee for use is one cent per minute with a minimum of 25 cents in any day that it is used, payable to the account described by Account-ID-567.

[0175] ((Play) (Transfer) (Delete) (Loan 2(Delete: Transfer Loan))) This work can be played, transferred, deleted, or loaned. Up to two copies can be loaned out at a time. The loaned copy has the same rights except that it cannot be transferred. When both copies are loaned out, no rights can be exercised on the original on the repository.

[0176] ((Play) (Transfer) (Delete) (Backup) (Restore (SC:3)) (Loan 2 Remaining-Copy-Rights: (Delete: Play Transfer) Next-Set-of-Rights: (Delete: Transfer Loan))) Similar to previous example. Rights to Backup and Restore the work are added, where restoration requires a repository of at least security level three. When all copies of the work are loaned out, the remaining copy cannot be played or transferred.

[0177] ((Play) (Transfer) (Copy) (Print) (Backup) (Restore (SC:3)) (Loan 1 Remaining-Copy-Rights: (Add: Play Print Backup) Next-Set-of-Rights: (Delete: Transfer Loan) (Fee: Metered: \$10 Per: 1:0:0 To: Account-ID-567)) (Loan 1 Remaining-Copy-Rights: Add: ((Play Player: Player-876-ID) 2 (From: 94/02/14 Until: 95/02/15) (Fee: Metered: \$0.01 Per: 0:1:0 Min: \$0.25 Per: 0/1/0 To: Account-ID-567)))))

[0178] The original work has rights to Play, Transfer, Copy, Print, Backup, Restore, and Loan. There are two versions of the Loan right. The first version of the loan right costs \$10 per

NVA208575.1

day but allows the original copy owner to exercise free use of the Play, Print and Backup rights. The second version of the Loan right is free. None of the original rights are applicable. However a right to Play the work at the specified metered rate is added.

[0179] ((Play Player: Player-Small-Screen-123-ID) (Embed (Fee: Per-Use \$.01 To: Account-678-ID)) (Copy (Fee: Per-Use \$1.00 To: Account-678-ID)))

The digital work can be played on any player with the identifier Player-Small-Screen-123-ID. It can be embedded in a larger work. The embedding requires a modest one cent registration fee to Account-678-ID. Digital copies can be made for \$1.00.

[0180] When a user requests access to a digital work, the repository will initiate various transactions. The combination of transactions invoked will depend on the specifications assigned for a usage right. There are three basic types of transactions, Session Initiation Transactions, Financial Transactions and Usage Transactions. Generally, session initiation transactions are initiated first to establish a valid session. When a valid session is established, transactions corresponding to the various usage rights are invoked. Finally, request specific transactions are performed.

[0181] Transactions occur between two repositories (one acting as a server), between a repository and a document playback platform (e.g. for executing or viewing), between a repository and a credit server or between a repository and an authorization server. When transactions occur between more than one repository, it is assumed that there is a reliable communication channel between the repositories. For example, this could be a TCP/IP channel or any other commercially available channel that has built-in capabilities for detecting and correcting transmission errors. However, it is not assumed that the communication channel is secure. Provisions for security and privacy are part of the requirements for specifying and implementing repositories and thus form the need for various transactions.

[0182] Transactions require that there be some communication between repositories. Communication between repositories occurs in units termed as messages. Because the communication line is assumed to be unsecure, all communications with repositories that are

NVA208575.1

above the lowest security class are encrypted utilizing a public key encryption technique. Public key encryption is a well known technique in the encryption arts. The term key refers to a numeric code that is used with encryption and decryption algorithms. Keys come in pairs, where "writing keys" are used to encrypt data and "checking keys" are used to decrypt data. Both writing and checking keys may be public or private. Public keys are those that are distributed to others. Private keys are maintained in confidence.

[0183] Key management and security is instrumental in the success of a public key encryption system. In the currently preferred embodiment, one or more master repositories maintain the keys and create the identification certificates used by the repositories.

[0184] When a sending repository transmits a message to a receiving repository, the sending repository encrypts all of its data using the public writing key of the receiving repository. The sending repository includes its name, the name of the receiving repository, a session identifier such as a nonce (described below), and a message counter in each message.

[0185] In this way, the communication can only be read (to a high probability) by the receiving repository, which holds the private checking key for decryption. The auxiliary data is used to guard against various replay attacks to security. If messages ever arrive with the wrong counter or an old nonce, the repositories can assume that someone is interfering with communication and the transaction terminated.

[0186] The respective public keys for the repositories to be used for encryption are obtained in the registration transaction described below.

[0187] A usage transaction is carried out in a session between repositories. For usage transactions involving more than one repository, or for financial transactions between a repository and a credit server, a registration transaction is performed. A second transaction termed a login transaction, may also be needed to initiate the session. The goal of the registration transaction is to establish a secure channel between two repositories who know each other's identities. As it is assumed that the communication channel between the repositories is reliable

NVA208575.1

but not secure, there is a risk that a non-repository may mimic the protocol in order to gain illegitimate access to a repository.

[0188] The registration transaction between two repositories is described with respect to FIGS. 16 and 17. The steps described are from the perspective of a "repository-1" registering its identity with a "repository-2". The registration must be symmetrical so the same set of steps will be repeated for repository-2 registering its identity with repository-1. Referring to FIG. 16, repository-1 first generates an encrypted registration identifier, step 1601 and then generates a registration message, step 1602. A registration message is comprised of an identifier of a master repository, the identification certificate for the repository-1 and an encrypted random registration identifier. The identification certificate is encrypted by the master repository in its private key and attests to the fact that the repository (here repository-1) is a bona fide repository. The identification certificate also contains a public key for the repository, the repository security level and a timestamp (indicating a time after which the certificate is no longer valid.) The registration identifier is a number generated by the repository for this registration. The registration identifier is unique to the session and is encrypted in repository-1's private key. The registration identifier is used to improve security of authentication by detecting certain kinds of communications based attacks. Repository-1 then transmits the registration message to repository-2, step 1603.

[0189] Upon receiving the registration message, repository-2 determines if it has the needed public key for the master repository, step 1604. If repository-2 does not have the needed public key to decrypt the identification certificate, the registration transaction terminates in an error, step 1618.

[0190] Assuming that repository-2 has the proper public key the identification certificate is decrypted, step 1605. Repository-2 saves the encrypted registration identifier, step 1606, and extracts the repository identifier, step 1607. The extracted repository identifier is checked against a "hotlist" of compromised document repositories, step 1608. In the currently preferred embodiment, each repository will contain "hotlists" of compromised repositories. If the repository is on the "hotlist", the registration transaction terminates in an error per step 1618. Repositories can be removed from the hotlist when their certificates expire, so that the list does

NVA208575.1

not need to grow without bound. Also, by keeping a short list of hotlist certificates that it has previously received, a repository can avoid the work of actually going through the list. These lists would be encrypted by a master repository. A minor variation on the approach to improve efficiency would have the repositories first exchange lists of names of hotlist certificates, ultimately exchanging only those lists that they had not previously received. The "hotlists" are maintained and distributed by Master repositories.

[0191] Note that rather than terminating in error, the transaction could request that another registration message be sent based on an identification certificate created by another master repository. This may be repeated until a satisfactory identification certificate is found, or it is determined that trust cannot be established.

[0192] Assuming that the repository is not on the hotlist, the repository identification needs to be verified. In other words, repository-2 needs to validate that the repository on the other end is really repository-1. This is termed performance testing and is performed in order to avoid invalid access to the repository via a counterfeit repository replaying a recording of a prior session initiation between repository-1 and repository-2. Performance testing is initiated by repository-2 generating a performance message, step 1609. The performance message consists of a nonce, the names of the respective repositories, the time and the registration identifier received from repository-1. A nonce is a generated message based on some random and variable information (e.g. the time or the temperature.) The nonce is used to check whether repository-1 can actually exhibit correct encrypting of a message using the private keys it claims to have, on a message that it has never seen before. The performance message is encrypted using the public key specified in the registration message of repository-1. The performance message is transmitted to repository-1, step 1610, where it is decrypted by repository-1 using its private key, step 1611. Repository-1 then checks to make sure that the names of the two repositories are correct, step 1612, that the time is accurate, step 1613 and that the registration identifier corresponds to the one it sent, step 1614. If any of these tests fails, the transaction is terminated per step 1616. Assuming that the tests are passed, repository-1 transmits the nonce to repository-2 in the clear, step 1615. Repository-2 then compares the received nonce to the original nonce, step 1617. If they are not identical, the registration transaction terminates in an error per step

NVA208575.1

1618. If they are the same, the registration transaction has successfully completed.

[0193] At this point, assuming that the transaction has not terminated, the repositories exchange messages containing session keys to be used in all communications during the session and synchronize their clocks. FIG. 17 illustrates the session information exchange and clock synchronization steps (again from the perspective of repository-1.) Referring to FIG. 17, repository-1 creates a session key pair, step 1701. A first key is kept private and is used by repository-1 to encrypt messages. The second key is a public key used by repository-2 to decrypt messages. The second key is encrypted using the public key of repository-2, step 1702 and is sent to repository-2, step 1703. Upon receipt, repository-2 decrypts the second key, step 1704. The second key is used to decrypt messages in subsequent communications. When each repository has completed this step, they are both convinced that the other repository is bona fide and that they are communicating with the original. Each repository has given the other a key to be used in decrypting further communications during the session. Since that key is itself transmitted in the public key of the receiving repository only it will be able to decrypt the key which is used to decrypt subsequent messages.

[0194] After the session information is exchanged, the repositories must synchronize their clocks. Clock synchronization is used by the repositories to establish an agreed upon time base for the financial records of their mutual transactions. Referring back to FIG. 17, repository-2 initiates clock synchronization by generating a time stamp exchange message, step 1705, and transmits it to repository-1, step 1706. Upon receipt, repository-1 generates its own time stamp message, step 1707 and transmits it back to repository-2, step 1708. Repository-2 notes the current time, step 1709 and stores the time received from repository-1, step 1710. The current time is compared to the time received from repository-1, step 1711. The difference is then checked to see if it exceeds a predetermined tolerance (e.g. one minute), step 1712. If it does, repository-2 terminates the transaction as this may indicate tampering with the repository, step 1713. If not repository-2 computes an adjusted time delta, step 1714. The adjusted time delta is the difference between the clock time of repository-2 and the average of the times from repository-1 and repository-2.

[0195] To achieve greater accuracy, repository-2 can request the time again up to a fixed

NVA208575.1

number of times (e.g. five times), repeat the clock synchronization steps, and average the results.

[0196] A second session initiation transaction is a Login transaction. The Login transaction is used to check the authenticity of a user requesting a transaction. A Login transaction is particularly prudent for the authorization of financial transactions that will be charged to a credit server. The Login transaction involves an interaction between the user at a user interface and the credit server associated with a repository. The information exchanged here is a login string supplied by the repository/credit server to identify itself to the user, and a Personal Identification Number (PIN) provided by the user to identify himself to the credit server. In the event that the user is accessing a credit server on a repository different from the one on which the user interface resides, exchange of the information would be encrypted using the public and private keys of the respective repositories.

[0197] Billing Transactions are concerned with monetary transaction with a credit server. Billing transactions are carried out when all other conditions are satisfied and a usage fee is required for granting the request. For the most part, billing transactions are well understood in the state of the art. These transactions are between a repository and a credit server, or between a credit server and a billing clearinghouse. Briefly, the required transactions include the following:

[0198] Registration and LOGIN transactions by which the repository and user establish their bona fides to a credit server. These transactions would be entirely internal in cases where the repository and credit server are implemented as a single system.

[0199] Registration and LOGIN transactions, by which a credit server establishes its bona fides to a billing clearinghouse.

[0200] An Assign-fee transaction to assign a charge. The information in this transaction would include a transaction identifier, the identities of the repositories in the transaction, and a list of charges from the parts of the digital work. If there has been any unusual event in the transaction such as an interruption of communications, that information is included as well.

[0201] An Begin-charges transaction to assign a charge. This transaction is much the

NVA208575.1

same as an assign-fee transaction except that it is used for metered use. It includes the same information as the assign-fee transaction as well as the usage fee information. The credit-server is then responsible for running a clock.

[0202] An End-charges transaction to end a charge for metered use. (In a variation on this approach, the repositories would exchange periodic charge information for each block of time.)

[0203] A report-charges transaction between a personal credit server and a billing clearinghouse. This transaction is invoked at least once per billing period. It is used to pass along information about charges. On debit and credit cards, this transaction would also be used to update balance information and credit limits as needed.

[0204] All billing transactions are given a transaction ID and are reported to the credit servers by both the server and the client. This reduces possible loss of billing information if one of the parties to a transaction loses a banking card and provides a check against tampering with the system.

[0205] After the session initiation transactions have been completed, the usage request may then be processed. To simplify the description of the steps carried out in processing a usage request, the term requester is used to refer to a repository in the requester mode which is initiating a request, and the term server is used to refer to a repository in the server mode and which contains the desired digital work. In many cases such as requests to print or view a work, the requester and server may be the same device and the transactions described in the following would be entirely internal. In such instances, certain transaction steps, such as the registration transaction, need not be performed.

[0206] There are some common steps that are part of the semantics of all of the usage rights transactions. These steps are referred to as the common transaction steps. There are two sets--the "opening" steps and the "closing" steps. For simplicity, these are listed here rather than repeating them in the descriptions of all of the usage rights transactions.

NVA208575.1

[0207] Transactions can refer to a part of a digital work, a complete digital work, or a digital work containing other digital works. Although not described in detail herein, a transaction may even refer to a folder comprised of a plurality of digital works. The term "work" is used to refer to what ever portion or set of digital works is being accessed.

[0208] Many of the steps here involve determining if certain conditions are satisfied. Recall that each usage right may have one or more conditions which must be satisfied before the right can be exercised. Digital works have parts and parts have parts. Different parts can have different rights and fees. Thus, it is necessary to verify that the requirements are met for ALL of the parts that are involved in a transaction For brevity, when reference is made to checking whether the rights exist and conditions for exercising are satisfied, it is meant that all such checking takes place for each of the relevant parts of the work.

[0209] FIG. 18 illustrates the initial common opening and closing steps for a transaction. At this point it is assumed that registration has occurred and that a "trusted" session is in place. General tests are tests on usage rights associated with the folder containing the work or some containing folder higher in the file system hierarchy. These tests correspond to requirements imposed on the work as a consequence of its being on the particular repository, as opposed to being attached to the work itself. Referring to FIG. 18, prior to initiating a usage transaction, the requester performs any general tests that are required before the right associated with the transaction can be exercised, step, 1801. For example, install, uninstall and delete rights may be implemented to require that a requester have an authorization certificate before the right can be exercised. Another example is the requirement that a digital ticket be present and punched before a digital work may be copied to a requester. If any of the general tests fail, the transaction is not initiated, step, 1802. Assuming that such required tests are passed, upon receiving the usage request, the server generates a transaction identifier that is used in records or reports of the transaction, step 1803. The server then checks whether the digital work has been granted the right corresponding to the requested transaction, step 1804. If the digital work has not been granted the right corresponding to the request, the transaction terminates, step 1805. If the digital work has been granted the requested right, the server then determines if the various conditions

NVA208575.1

for exercising the right are satisfied. Time based conditions are examined, step 1806. These conditions are checked by examining the time specification for the version of the right. If any of the conditions are not satisfied, the transaction terminates per step 1805.

[0210] Assuming that the time based conditions are satisfied, the server checks security and access conditions, step 1807. Such security and access conditions are satisfied if: 1) the requester is at the specified security class, or a higher security class, 2) the server satisfies any specified authorization test and 3) the requester satisfies any specified authorization tests and has any required digital tickets. If any of the conditions are not satisfied, the transaction terminates per step 1805.

[0211] Assuming that the security and access conditions are all satisfied, the server checks the copy count condition, step 1808. If the copy count equals zero, then the transaction cannot be completed and the transaction terminates per step 1805.

[0212] Assuming that the copy count does not equal zero, the server checks if the copies in use for the requested right is greater than or equal to any copy count for the requested right (or relevant parts), step 1809. If the copies in use is greater than or equal to the copy count, this indicates that usage rights for the version of the transaction have been exhausted. Accordingly, the server terminates the transaction, step 1805. If the copy count is less than the copies in use for the transaction the transaction can continue, and the copies in use would be incremented by the number of digital works requested in the transaction, step 1810.

[0213] The server then checks if the digital work has a "Loan" access right, step 1811. The "Loan" access right is a special case since remaining rights may be present even though all copies are loaned out. If the digital work has the "Loan" access right, a check is made to see if all copies have been loaned out, step 1812. The number of copies that could be loaned is the sum of the Copy-Counts for all of the versions of the loan right of the digital work. For a composite work, the relevant figure is the minimal such sum of each of the components of the composite work. If all copies have been loaned out, the remaining rights are determined, step 1813. The remaining-rights is determined from the remaining rights specifications from the versions of the

NVA208575.1

Loan right. If there is only one version of the Loan right, then the determination is simple. The remaining rights are the ones specified in that version of the Loan right, or none if Remaining-Rights: is not specified. If there are multiple versions of the Loan right and all copies of all of the versions are loaned out, then the remaining rights is taken as the minimum set (intersection) of remaining rights across all of the versions of the loan right. The server then determines if the requested right is in the set of remaining rights, step 1814. If the requested right is not in the set of remaining rights, the server terminates the transaction, step 1805.

[0214] If Loan is not a usage right for the digital work or if all copies have not been loaned out or the requested right is in the set of remaining rights, fee conditions for the right are then checked, step 1815. This will initiate various financial transactions between the repository and associated credit server. Further, any metering of usage of a digital work will commence. If any financial transaction fails, the transaction terminates per step 1805.

[0215] It should be noted that the order in which the conditions are checked need not follow the order of steps 1806-1815.

[0216] At this point, right specific steps are now performed and are represented here as step 1816. The right specific steps are described in greater detail below.

[0217] The common closing transaction steps are now performed. Each of the closing transaction steps are performed by the server after a successful completion of a transaction. Referring back to FIG. 18, the copies in use value for the requested right is decremented by the number of copies involved in the transaction, step 1817. Next, if the right had a metered usage fee specification, the server subtracts the elapsed time from the Remaining-Use-Time associated with the right for every part involved in the transaction, step 1818. Finally, if there are fee specifications associated with the right, the server initiates End-Charge financial transaction to confirm billing, step 1819.

[0218] An important area to consider is the transmission of the digital work from the server to the requester. The transmission protocol described herein refers to events occurring

NVA208575.1

after a valid session has been created. The transmission protocol must handle the case of disruption in the communications between the repositories. It is assumed that interference such as injecting noise on the communication channel can be detected by the integrity checks (e.g., parity, checksum, etc.) that are built into the transport protocol and are not discussed in detail herein.

[0219] The underlying goal in the transmission protocol is to preclude certain failure modes, such as malicious or accidental interference on the communications channel. Suppose, for example, that a user pulls a card with the credit server at a specific time near the end of a transaction. There should not be a vulnerable time at which "pulling the card" causes the repositories to fail to correctly account for the number of copies of the work that have been created. Restated, there should be no time at which a party can break a connection as a means to avoid payment after using a digital work.

[0220] If a transaction is interrupted (and fails), both repositories restore the digital works and accounts to their state prior to the failure, modulo records of the failure itself.

[0221] FIG. 19 is a state diagram showing steps in the process of transmitting information during a transaction. Each box represents a state of a repository in either the server mode (above the central dotted line 1901) or in the requester mode (below the dotted line 1901). Solid arrows stand for transitions between states. Dashed arrows stand for message communications between the repositories. A dashed message arrow pointing to a solid transition arrow is interpreted as meaning that the transition takes place when the message is received. Unlabeled transition arrows take place unconditionally. Other labels on state transition arrows describe conditions that trigger the transition.

[0222] Referring now to FIG. 19, the server is initially in a state 1902 where a new transaction is initiated via start message 1903. This message includes transaction information including a transaction identifier and a count of the blocks of data to be transferred. The requester, initially in a wait state 1904 then enters a data wait state 1905.

[0223] The server enters a data transmit state 1906 and transmits a block of data 1907 and then enters a wait for acknowledgement state 1908. As the data is received, the requester enters a data receive state 1909 and when the data blocks is completely received it enters an acknowledgement state 1910 and transmits an Acknowledgement message 1911 to the server.

[0224] If there are more blocks to send, the server waits until receiving an Acknowledgement message from the requester. When an Acknowledgement message is received it sends the next block to the requester and again waits for acknowledgement. The requester also repeats the same cycle of states.

[0225] If the server detects a communications failure before sending the last block, it enters a cancellation state 1912 wherein the transaction is cancelled. Similarly, if the requester detects a communications failure before receiving the last block it enters a cancellation state 1913.

[0226] If there are no more blocks to send, the server commits to the transaction and waits for the final Acknowledgement in state 1914. If there is a communications failure before the server receives the final Acknowledgement message, it still commits to the transaction but includes a report about the event to its credit server in state 1915. This report serves two purposes. It will help legitimize any claims by a user of having been billed for receiving digital works that were not completely received. Also it helps to identify repositories and communications lines that have suspicious patterns of use and interruption. The server then enters its completion state 1916.

[0227] On the requester side, when there are no more blocks to receive, the requester commits to the transaction in state 1917. If the requester detects a communications failure at this state, it reports the failure to its credit server in state 1918, but still commits to the transaction. When it has committed, it sends an acknowledgement message to the server. The server then enters its completion state 1919.

[0228] The key property is that both the server and the requester cancel a transaction if it is interrupted before all of the data blocks are delivered, and commits to it if all of the data

NVA208575.1

blocks have been delivered.

[0229] There is a possibility that the server will have sent all of the data blocks (and committed) but the requester will not have received all of them and will cancel the transaction. In this case, both repositories will presumably detect a communications failure and report it to their credit server. This case will probably be rare since it depends on very precise timing of the communications failure. The only consequence will be that the user at the requester repository may want to request a refund from the credit services--and the case for that refund will be documented by reports by both repositories.

[0230] To prevent loss of data, the server should not delete any transferred digital work until receiving the final acknowledgement from the requester. But it also should not use the file. A well known way to deal with this situation is called "two-phase commit" or 2PC. Two-phase commit works as follows. The first phase works the same as the method described above. The server sends all of the data to the requester. Both repositories mark the transaction (and appropriate files) as uncommitted. The server sends a ready-to-commit message to the requester. The requester sends back an acknowledgement. The server then commits and sends the requester a commit message. When the requester receives the commit message, it commits the file.

[0231] If there is a communication failure or other crash, the requester must check back with the server to determine the status of the transaction. The server has the last word on this. The requester may have received all of the data, but if it did not get the final message, it has not committed. The server can go ahead and delete files (except for transaction records) once it commits, since the files are known to have been fully transmitted before starting the 2PC cycle.

[0232] There are variations known in the art which can be used to achieve the same effect. For example, the server could use an additional level of encryption when transmitting a work to a client. Only after the client sends a message acknowledging receipt does it send the key. The client then agrees to pay for the digital work. The point of this variation is that it provides a clear audit trail that the client received the work. For trusted systems, however, this

NVA208575.1

variation adds a level of encryption for no real gain in accountability.

[0233] The transaction for specific usage rights are now discussed. A Copy transaction is a request to make one or more independent copies of the work with the same or lesser usage rights. Copy differs from the extraction right discussed later in that it refers to entire digital works or entire folders containing digital works. A copy operation cannot be used to remove a portion of a digital work.

[0234] The requester sends the server a message to initiate the Copy Transaction. This message indicates the work to be copied, the version of the copy right to be used for the transaction, the destination address information (location in a folder) for placing the work, the file data for the work (including its size), and the number of copies requested.

[0235] The repositories perform the common opening transaction steps. The server transmits the requested contents and data to the client according to the transmission protocol. If a Next-Set-Of-Rights has been provided in the version of the right, those rights are transmitted as the rights for the work. Otherwise, the rights of the original are transmitted. In any event, the Copy-Count field for the copy of the digital work being sent right is set to the number-of-copies requested.

[0236] The requester records the work contents, data, and usage rights and stores the work. It records the date and time that the copy was made in the properties of the digital work.

[0237] The repositories perform the common closing transaction steps. A Transfer transaction is a request to move copies of the work with the same or lesser usage rights to another repository. In contrast with a copy transaction, this results in removing the work copies from the server.

[0238] The requester sends the server a message to initiate the Transfer Transaction. This message indicates the work to be transferred, the version of the transfer right to be used in the

transaction, the destination address information for placing the work, the file data for the work, and the number of copies involved.

[0239] The repositories perform the common opening transaction steps. The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, the rights of the original are transmitted. In either case, the Copy-Count field for the transmitted rights are set to the number-of-copies requested.

[0240] The requester records the work contents, data, and usage rights and stores the work.

[0241] The server decrements its copy count by the number of copies involved in the transaction.

[0242] The repositories perform the common closing transaction steps.

[0243] If the number of copies remaining in the server is now zero, it erases the digital work from its memory.

[0244] A loan transaction is a mechanism for loaning copies of a digital work. The maximum duration of the loan is determined by an internal parameter of the digital work. Works are automatically returned after a predetermined time period.

[0245] The requester sends the server a message to initiate the Transfer Transaction. This message indicates the work to be loaned, the version of the loan right to be used in the transaction, the destination address information for placing the work, the number of copies involved, the file data for the work, and the period of the loan.

[0246] The server checks the validity of the requested loan period, and ends with an error if the period is not valid. Loans for a loaned copy cannot extend beyond the period of the original

NVA208575.1

loan to the server.

[0247] The repositories perform the common opening transaction steps. The server transmits the requested contents and data to the requester. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, the rights of the original are transmitted, as modified to reflect the loan period.

[0248] The requester records the digital work contents, data, usage rights, and loan period and stores the work.

[0249] The server updates the usage rights information in the digital work to reflect the number of copies loaned out..

[0250] The repositories perform the common closing transaction steps. The server updates the usage rights data for the digital work. This may preclude use of the work until it is returned from the loan. The user on the requester platform can now use the transferred copies of the digital work. A user accessing the original repository cannot use the digital work , unless there are copies remaining. What happens next depends on the order of events in time.

[0251] Case 1. If the time of the loan period is not yet exhausted and the requester sends the repository a Return message.

[0252] The return message includes the requester identification, and the transaction ID. The server decrements the copies-in-use field by the number of copies that were returned. (If the number of digital works returned is greater than the number actually borrowed, this is treated as an error.) This step may now make the work available at the server for other users.

[0253] The requester deactivates its copies and removes the contents from its memory.

[0254] Case 2. If the time of the loan period is exhausted and the requester has not yet sent a Return message. The server decrements the copies-in-use field by the number digital

NVA208575.1

works that were borrowed.

[0255] The requester automatically deactivates its copies of the digital work. It terminates all current uses and erases the digital work copies from memory. One question is why a requester would ever return a work earlier than the period of the loan, since it would be returned automatically anyway. One reason for early return is that there may be a metered fee which determines the cost of the loan. Returning early may reduce that fee.

[0256] A play transaction is a request to use the contents of a work. Typically, to "play" a work is to send the digital work through some kind of transducer, such as a speaker or a display device. The request implies the intention that the contents will not be communicated digitally to any other system. For example, they will not be sent to a printer, recorded on any digital medium, retained after the transaction or sent to another repository.

[0257] This term "play" is natural for examples like playing music, playing a movie, or playing a video game. The general form of play means that a "player" is used to use the digital work. However, the term play covers all media and kinds of recordings. Thus one would "play" a digital work, meaning, to render it for reading, or play a computer program, meaning to execute it. For a digital ticket the player would be a digital ticket agent.

[0258] The requester sends the server a message to initiate the play transaction. This message indicates the work to be played, the version of the play right to be used in the transaction, the identity of the player being used, and the file data for the work.

[0259] The server checks the validity of the player identification and the compatibility of the player identification with the player specification in the right. It ends with an error if these are not satisfactory.

[0260] The repositories perform the common opening transaction steps. The server and requester read and write the blocks of data as requested by the player according to the transmission protocol. The requester plays the work contents, using the player.

NVA208575.1

[0261] When the player is finished, the player and the requester remove the contents from their memory.

[0262] The repositories perform the common closing transaction steps. A Print transaction is a request to obtain the contents of a work for the purpose of rendering them on a "printer." We use the term "printer" to include the common case of writing with ink on paper. However, the key aspect of "printing" in our use of the term is that it makes a copy of the digital work in a place outside of the protection of usage rights. As with all rights, this may require particular authorization certificates.

[0263] Once a digital work is printed, the publisher and user are bound by whatever copyright laws are in effect. However, printing moves the contents outside the control of repositories. For example, absent any other enforcement mechanisms, once a digital work is printed on paper, it can be copied on ordinary photocopying machines without intervention by a repository to collect usage fees. If the printer to a digital disk is permitted, then that digital copy is outside of the control of usage rights. Both the creator and the user know this, although the creator does not necessarily give tacit consent to such copying, which may violate copyright laws.

[0264] The requester sends the server a message to initiate a Print transaction. This message indicates the work to be played, the identity of the printer being used, the file data for the work, and the number of copies in the request.

[0265] The server checks the validity of the printer identification and the compatibility of the printer identification with the printer specification in the right. It ends with an error if these are not satisfactory.

[0266] The repositories perform the common opening transaction steps.

[0267] The server transmits blocks of data according to the transmission protocol.

[0268] The requester prints the work contents, using the printer.

[0269] When the printer is finished, the printer and the requester remove the contents from their memory.

[0270] The repositories perform the common closing transaction steps. A Backup transaction is a request to make a backup copy of a digital work, as a protection against media failure. In the context of repositories, secure backup copies differ from other copies in three ways: (1) they are made under the control of a Backup transaction rather than a Copy transaction, (2) they do not count as regular copies, and (3) they are not usable as regular copies. Generally, backup copies are encrypted.

[0271] Although backup copies may be transferred or copied, depending on their assigned rights, the only way to make them useful for playing, printing or embedding is to restore them.

[0272] The output of a Backup operation is both an encrypted data file that contains the contents and description of a work, and a restoration file with an encryption key for restoring the encrypted contents. In many cases, the encrypted data file would have rights for "printing" it to a disk outside of the protection system, relying just on its encryption for security. Such files could be stored anywhere that was physically safe and convenient. The restoration file would be held in the repository. This file is necessary for the restoration of a backup copy. It may have rights for transfer between repositories.

[0273] The requester sends the server a message to initiate a backup transaction. This message indicates the work to be backed up, the version of the backup right to be used in the transaction, the destination address information for placing the backup copy, the file data for the work.

[0274] The repositories perform the common opening transaction steps. The server transmits the requested contents and data to the requester. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, a set of default rights for backup files of the original are transmitted by the server.

[0275] The requester records the work contents, data, and usage rights. It then creates a one-time key and encrypts the contents file. It saves the key information in a restoration file.

[0276] The repositories perform the common closing transaction steps. In some cases, it is convenient to be able to archive the large, encrypted contents file to secure offline storage, such as a magneto-optical storage system or magnetic tape. This creation of a non-repository archive file is as secure as the encryption process. Such non-repository archive storage is considered a form of "printing" and is controlled by a print right with a specified "archive-printer." An archive-printer device is programmed to save the encrypted contents file (but not the description file) offline in such a way that it can be retrieved.

[0277] The Restore Transaction A Restore transaction is a request to convert an encrypted backup copy of a digital work into a usable copy. A restore operation is intended to be used to compensate for catastrophic media failure. Like all usage rights, restoration rights can include fees and access tests including authorization checks.

[0278] The requester sends the server a message to initiate a Restore transaction. This message indicates the work to be restored, the version of the restore right for the transaction, the destination address information for placing the work, and the file data for the work.

[0279] The server verifies that the contents file is available (i.e. a digital work corresponding to the request has been backed-up.) If it is not, it ends the transaction with an error.

[0280] The repositories perform the common opening transaction steps.

[0281] The server retrieves the key from the restoration file. It decrypts the work contents, data, and usage rights.

[0282] The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the work. Otherwise, a set of default rights for backup files of the original are transmitted by the server.

[0283] The requester stores the digital work.

[0284] The repositories perform the common closing transaction steps. A Delete transaction deletes a digital work or a number of copies of a digital work from a repository. Practically all digital works would have delete rights.

[0285] The requester sends the server a message to initiate a delete transaction. This message indicates the work to be deleted, the version of the delete right for the transaction.

[0286] The repositories perform the common opening transaction steps.

[0287] The server deletes the file, erasing it from the file system.

[0288] The repositories perform the common closing transaction steps. A Directory transaction is a request for information about folders, digital works, and their parts. This amounts to roughly the same idea as protection codes in a conventional file system like TENEX, except that it is generalized to the full power of the access specifications of the usage rights language.

[0289] The Directory transaction has the important role of passing along descriptions of the rights and fees associated with a digital work. When a user wants to exercise a right, the user interface of his repository implicitly makes a directory request to determine the versions of the right that are available. Typically these are presented to the user--such as with different choices of billing for exercising a right. Thus, many directory transactions are invisible to the user and

NVA208575.1

are exercised as part of the normal process of exercising all rights.

[0290] The requester sends the server a message to initiate a Directory transaction. This message indicates the file or folder that is the root of the directory request and the version of the directory right used for the transaction.

[0291] The server verifies that the information is accessible to the requester. In particular, it does not return the names of any files that have a HIDE-NAME status in their directory specifications, and it does not return the parts of any folders or files that have HIDE-PARTS in their specification. If the information is not accessible, the server ends the transaction with an error.

[0292] The repositories perform the common opening transaction steps.

[0293] The server sends the requested data to the requester according to the transmission protocol. The requester records the data.

[0294] The repositories perform the common closing transaction steps. A Folder transaction is a request to create or rename a folder, or to move a work between folders. Together with Directory rights, Folder rights control the degree to which organization of a repository can be accessed or modified from another repository.

[0295] The requester sends the server a message to initiate a Folder transaction. This message indicates the folder that is the root of the folder request, the version of the folder right for the transaction, an operation, and data. The operation can be one of create, rename, and move file. The data are the specifications required for the operation, such as a specification of a folder or digital work and a name.

[0296] The repositories perform the common opening transaction steps. The server performs the requested operation--creating a folder, renaming a folder, or moving a work between folders.

NVA208575.1

[0297] The repositories perform the common closing transaction steps. A extract transaction is a request to copy a part of a digital work and to create a new work containing it. The extraction operation differs from copying in that it can be used to separate a part of a digital work from d-blocks or shells that place additional restrictions or fees on it. The extraction operation differs from the edit operation in that it does not change the contents of a work, only its embedding in d-blocks. Extraction creates a new digital work.

[0298] The requester sends the server a message to initiate an Extract transaction. This message indicates the part of the work to be extracted, the version of the extract right to be used in the transaction, the destination address information for placing the part as a new work, the file data for the work, and the number of copies involved.

[0299] The repositories perform the common opening transaction steps. The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the new work. Otherwise, the rights of the original are transmitted. The Copy-Count field for this right is set to the number-of-copies requested.

[0300] The requester records the contents, data, and usage rights and stores the work. It records the date and time that new work was made in the properties of the work.

[0301] The repositories perform the common closing transaction steps. An embed transaction is a request to make a digital work become a part of another digital work or to add a shell d-block to enable the adding of fees by a distributor of the work.

[0302] The requester sends the server a message to initiate an Embed transaction. This message indicates the work to be embedded, the version of the embed right to be used in the transaction; the destination address information for placing the part as a work, the file data for the work, and the number of copies involved.

[0303] The server checks the control specifications for all of the rights in the part and the

NVA208575.1

destination. If they are incompatible, the server ends the transaction with an error.

[0304] The repositories perform the common opening transaction steps. The server transmits the requested contents and data to the requester according to the transmission protocol. If a Next-Set-Of-Rights has been provided, those rights are transmitted as the rights for the new work. Otherwise, the rights of the original are transmitted. The Copy-Count field for this right is set to the number-of-copies requested.

[0305] The requester records the contents, data, and usage rights and embeds the work in the destination file.

[0306] The repositories perform the common closing transaction steps. An Edit transaction is a request to revise a digital work by copying, selecting and modifying portions of an existing digital work. This operation can actually change the contents of a digital work. The kinds of changes that are permitted depend on the process being used. Like the extraction operation, edit operates on portions of a digital work. In contrast with the extract operation, edit does not effect the rights or location of the work. It only changes the contents. The kinds of changes permitted are determined by the type specification of the processor specified in the rights. In the currently preferred embodiment, an edit transaction changes the work itself and does not make a new work. However, it would be a reasonable variation to cause a new copy of the work to be made.

[0307] The requester sends the server a message to initiate an Edit transaction. This message indicates the work to be edited, the version of the edit right to be used in the transaction, the file data for the work (including its size), the process-ID for the process, and the number of copies involved.

[0308] The server checks the compatibility of the process-ID to be used by the requester against any process-ID specification in the right. If they are incompatible, it ends the transaction with an error.

[0309] The repositories perform the common opening transaction steps. The requester uses the process to change the contents of the digital work as desired. (For example, it can select and duplicate parts of it; combine it with other information; or compute functions based on the information. This can amount to editing text, music, or pictures or taking whatever other steps are useful in creating a derivative work.)

[0310] The repositories perform the common closing transaction steps. The edit transaction is used to cover a wide range of kinds of works. The category describes a process that takes as its input any portion of a digital work and then modifies the input in some way. For example, for text, a process for editing the text would require edit rights. A process for "summarizing" or counting words in the text would also be considered editing. For a music file, processing could involve changing the pitch or tempo, or adding reverberations, or any other audio effect. For digital video works, anything which alters the image would require edit rights. Examples would be colorizing, scaling, extracting still photos, selecting and combining frames into story boards, sharpening with signal processing, and so on.

[0311] Some creators may want to protect the authenticity of their works by limiting the kinds of processes that can be performed on them. If there are no edit rights, then no processing is allowed at all. A processor identifier can be included to specify what kind of process is allowed. If no process identifier is specified, then arbitrary processors can be used. For an example of a specific process, a photographer may want to allow use of his photograph but may not want it to be colorized. A musician may want to allow extraction of portions of his work but not changing of the tonality.

[0312] There are many ways that authorization transactions can be defined. In the following, our preferred way is to simply define them in terms of other transactions that we already need for repositories. Thus, it is convenient sometimes to speak of "authorization transactions," but they are actually made up of other transactions that repositories already have.

[0313] A usage right can specify an authorization-ID, which identifies an authorization object (a digital work in a file of a standard format) that the repository must have and which it

NVA208575.1

must process. The authorization is given to the generic authorization (or ticket) server of the repository which begins to interpret the authorization.

[0314] As described earlier, the authorization contains a server identifier, which may just be the generic authorization server or it may be another server. When a remote authorization server is required, it must contain a digital address. It may also contain a digital certificate.

[0315] If a remote authorization server is required, then the authorization process first performs the following steps:

[0316] The generic authorization server attempts to set up the communications channel. (If the channel cannot be set up, then authorization fails with an error.)

[0317] When the channel is set up, it performs a registration process with the remote repository. (If registration fails, then the authorization fails with an error.)

[0318] When registration is complete, the generic authorization server invokes a "Play" transaction with the remote repository, supplying the authorization document as the digital work to be played, and the remote authorization server (a program) as the "player." (If the player cannot be found or has some other error, then the authorization fails with an error.)

[0319] The authorization server then "plays" the authorization. This involves decrypting it using either the public key of the master repository that issued the certificate or the session key from the repository that transmitted it. The authorization server then performs various tests. These tests vary according to the authorization server. They include such steps as checking issue and validity dates of the authorization and checking any hot-lists of known invalid authorizations. The authorization server may require carrying out any other transactions on the repository as well, such as checking directories, getting some person to supply a password, or playing some other digital work. It may also invoke some special process for checking information about locations or recent events. The "script" for such steps is contained within the authorization server.

NVA208575.1

[0320] If all of the required steps are completed satisfactorily, the authorization server completes the transaction normally, signaling that authorization is granted.

[0321] An Install transaction is a request to install a digital work as runnable software on a repository. In a typical case, the requester repository is a rendering repository and the software would be a new kind or new version of a player. Also in a typical case, the software would be copied to file system of the requester repository before it is installed.

[0322] The requester sends the server an Install message. This message indicates the work to be installed, the version of the Install right being invoked, and the file data for the work (including its size).

[0323] The repositories perform the common opening transaction steps. The requester extracts a copy of the digital certificate for the software. If the certificate cannot be found or the master repository for the certificate is not known to the requester, the transaction ends with an error.

[0324] The requester decrypts the digital certificate using the public key of the master repository, recording the identity of the supplier and creator, a key for decrypting the software, the compatibility information, and a tamper-checking code. (This step certifies the software.)

[0325] The requester decrypts the software using the key from the certificate and computes a check code on it using a 1-way hash function. If the check-code does not match the tamper-checking code from the certificate, the installation transaction ends with an error. (This step assures that the contents of the software, including the various scripts, have not been tampered with.)

[0326] The requester retrieves the instructions in the compatibility-checking script and follows them. If the software is not compatible with the repository, the installation transaction ends with an error. (This step checks platform compatibility.)

[0327] The requester retrieves the instructions in the installation script and follows them. If there is an error in this process (such as insufficient resources), then the transaction ends with an error. Note that the installation process puts the runnable software in a place in the repository where it is no longer accessible as a work for exercising any usage rights other than the execution of the software as part of repository operations in carrying out other transactions.

[0328] The repositories perform the common closing transaction steps. An Uninstall transaction is a request to remove software from a repository. Since uncontrolled or incorrect removal of software from a repository could compromise its behavioral integrity, this step is controlled.

[0329] The requester sends the server an Uninstall message. This message indicates the work to be uninstalled, the version of the Uninstall right being invoked, and the file data for the work (including its size).

[0330] The repositories perform the common opening transaction steps. The requester extracts a copy of the digital certificate for the software. If the certificate cannot be found or the master repository for the certificate is not known to the requester, the transaction ends with an error.

[0331] The requester checks whether the software is installed. If the software is not installed, the transaction ends with an error.

[0332] The requester decrypts the digital certificate using the public key of the master repository, recording the identity of the supplier and creator, a key for decrypting the software, the compatibility information, and a tamper-checking code. (This step authenticates the certification of the software, including the script for uninstalling it.)

[0333] The requester decrypts the software using the key from the certificate and computes a check code on it using a 1-way hash function. If the check-code does not match the tamper-checking code from the certificate, the installation transaction ends with an error. (This

NVA208575.1

step assures that the contents of the software, including the various scripts, have not been tampered with.)

[0334] The requester retrieves the instructions in the uninstallation script and follows them. If there is an error in this process (such as insufficient resources), then the transaction ends with an error.

[0335] The repositories perform the common closing transaction steps. To appreciate the robustness and flexibility of the present invention, various distribution and use scenarios for digital works are illustrated below. These scenarios are meant to be exemplary rather than exhaustive.

[0336] In this scenario, a creator distributes copies of his works to various consumers. Each consumer is a potential distributor of the work. If the consumer copies the digital work (usually for a third party), a fee is collected and automatically paid to the creator.

[0337] This scenario is a new twist for digital works. It depends on the idea that "manufacturing" is just copying and is essentially free. It also assumes that the consumers as distributors do not require a fee for their time and effort in distributing the work.

[0338] This scenario is performed as follows: A creator creates a digital work. He grants a Copy right with fees paid back to himself. If he does not grant an Embed right, then consumers cannot use the mechanism to act as distributors to cause fees to be paid to themselves on future copies. Of course, they could negotiate side deals or trades to transfer money on their own, outside of the system.

[0339] In another scenario, every time a copy of a digital work is sold a fee is paid to the creator and also to the immediate distributor.

[0340] This scenario does not give special status to any particular distributor. Anyone who sells a document has the right to add a fee to the sale price. The fee for sale could be

NVA208575.1

established by the consumer. It could also be a fixed nominal amount that is contributed to the account of some charity.

[0341] This scenario is performed as follows: A creator creates a digital work. He grants a Copy right with fees to be paid back to himself. He grants an Embed right, so that anyone can add shells to have fees paid to themselves.

[0342] A distributor embeds the work in a shell, with fees specified to be paid back to himself. If the distributor is content to receive fees only for copies that he sells himself, he grants an Extract right on the shell.

[0343] When a consumer buys a copy from the distributor, fees are paid both to the distributor and to the creator. If he chooses, the consumer can extract the work from the distributor's shell. He cannot extract it from the creator's shell. He can add his own shell with fees to be paid to himself.

[0344] In this scenario, a creator wants to protect the reputation and value of his work by making certain requirements on its distributors. He issues licenses to distributors that satisfy the requirements, and in turn, promises to reward their efforts by assuring that the work will not be distributed over competing channels. The distributors incur expenses for selecting the digital work, explaining it to buyers, promoting its sale, and possibly for the license itself. The distributor obtains the right to enclose the digital work in a shell, whose function is to permit the attachment of usage fees to be paid to the distributor in addition to the fees to be paid to the creator.

[0345] This differs from the previous scenario in that it precludes the typical copy owner from functioning as a distributor, since the consumer lacks a license to copy the document. Thus, a consumer cannot make copies, even for free. All copies must come initially from authorized distributors. This version makes it possible to hold distributors accountable in some way for the sales and support of the work, by controlling the distribution of certificates that enable distributors to legitimately charge fees and copy owners to make copies. Since licenses are

NVA208575.1

themselves digital works, the same mechanisms give the creators control over distributors by charging for licenses and putting time limits on their validity.

[0346] This scenario is performed as follows: A creator purchases a digital distribution license that he will hand out to his distributors. He puts access requirements (such as a personal license) on the Copy and Transfer rights on the distribution license so that only he can copy or transfer it.

[0347] The creator also creates a digital work. He grants an Embed right and a Copy right, both of which require the distribution license to be exercised. He grants a Play right so that the work can be played by anyone. He may optionally add a Transfer or Loan right, so that end consumers can do some non-commercial exchange of the work among friends.

[0348] A distributor obtains the distribution license and a number of copies of the work. He makes copies for his customers, using his distribution license.

[0349] A customer buys and uses the work. He cannot make new copies because he lacks a distribution license.

[0350] This is a variation on the previous scenarios. A distributor can sell to anyone and anyone can sell additional copies, resulting in fees being paid back to the creator. However, only licensed distributors can add fees to be paid to themselves.

[0351] This scenario gives distributors the right to add fees to cover their own advertising and promotional costs, without making them be the sole suppliers. Their customers can also make copies, thus broadening the channel without diminishing their revenues. This is because distributors collect fees from copies of any copies that they originally sold. Only distributors can add fees.

[0352] This scenario is performed similarly to the previous ones. There are two key differences. (1) The creator only grants Embed rights for people who have a Distribution license.

NVA208575.1

This is done by putting a requirement for a distributor's license on the Embed right. Consequently, non-distributors cannot add their own fees. (2) The Distributor does not grant Extract rights, so that consumers cannot avoid paying fees to the Distributor if they make subsequent copies. Consequently, all subsequent copies result in fees paid to the Distributor and the Creator.

[0353] In this scenario, a distributor gets a fee for any copy he sells directly. However, if one of his customers sells further copies, he gets no further fee for those copies.

[0354] This scenario pays a distributor only for use of copies that he actually sold. This scenario is performed similarly to the previous ones. The key feature is that the distributor creates a shell which specifies fees to be paid to him. He puts Extract rights on the shell. When a consumer buys the work, he can extract away the distributor's shell. Copies made after that will not require fees to be paid to the distributor.

[0355] In another scenario, distributors sell to other distributors and fees are collected at each level. Every copy sold by any distributor--even several d-blocks down in the chain--results in a fee being paid back to all of the previous distributors.

[0356] This scenario is like a chain letter or value chain. Every contributor or distributor along the way obtains fees, and is thereby encouraged to promote the sale of copies of the digital work.

[0357] This scenario is performed similarly to the previous ones. The key feature is that the distributor creates a shell which specifies fees to be paid to him. He does not grant Extract rights on the shell. Consequently, all future copies that are made will result in fees paid to him.

[0358] In this scenario, distributors make money according to a distribution tree. The fee that they make depends on various parameters, such as time since their sale or the number of subsequent distributors.

[0359] This is a generalized version of the Distribution Tree scenario, in that it tries to vary the fee to account for the significance of the role of the distributor.

[0360] This scenario is similar to the previous one. The difference is that the fee specification on the distributor's shell has provisions for changes in prices. For example, there could be a fee schedule so that copies made after the passage of time will require lower fees to be paid to the distributor. Alternatively, the distributor could employ a "best-price" billing option, using any algorithm he chooses to determine the fee up to the maximum specified in the shell.

[0361] In this scenario, a first creator creates a work. It is distributed by a first distributor and purchased by a second creator. The second creator extracts a portion of the work and embeds in it a new work distributed by a second distributor. A consumer buys the new work from the second distributor. The first creator receives fees from every transaction; the first distributor receives fees only for his sale; the second creator and second distributor receive fees for the final sale.

[0362] This scenario shows how that flexible automatic arrangements can be set up to create automatic charging systems that mirror current practice. This scenario is analogous to when an author pays a fee to reuse a figure in some paper. In the most common case, a fee is paid to the creator or publisher, but not to the bookstore that sold the book.

[0363] The mechanisms for derived works are the same as those for distribution. In this scenario, several first creators create works. A second creator makes a selection of these, publishing a collection made up of the parts together with some new interstitial material. (For example, the digital work could be a selection of music or a selection of readings.) The second creator wants to continue to allow some of the selected works to be extractable, but not the interstitial material.

[0364] This scenario deals with fine grained control of the rights and fees for reuse.

[0365] This scenario is performed as follows: The first creators create their original

NVA208575.1

works. If they grant extraction and embedding rights, then the second creator can include them in a larger collected work. The second creator creates the interstitial material. He does grant an Extract right on the interstitial material. He grants Extract rights on a subset of the reused material. A consumer of the collection can only extract portions that have that right. Fees are automatically collected for all parts of the collection.

[0366] Commercial libraries buy works with the right to loan. They limit the loan period and charge their own fees for use. This scenario deals with fees for loaning rather than fees for making copies. The fees are collected by the same automatic mechanisms.

[0367] The mechanisms are the same as previous scenarios except that the fees are associated with the Loan usage right rather than the Copy usage right.

[0368] A creator believes that if people try his work that they will want to buy it or use it. Consumers of his work can copy the work for free, and play (or execute) a limited version of the work for free, and can play or use the full featured version for a fee.

[0369] This scenario is performed as follows: The creator creates a digital work and grants various rights and fees. The creator grants Copy and Embed rights without a fee, in order to ensure widespread distribution of the work. Another of the rights is a limited play right with little or no fee attached. For example, this right may be for playing only a portion of the work. The play right can have various restrictions on its use. It could have a ticket that limits the number of times it is used. It could have internal restrictions that limit its functionality. It could have time restrictions that invalidate the right after a period of time or a period of use. Different fees could be associated with other versions of the Play right.

[0370] A consumer buys a digital work together with an agreement that he can upgrade to a new version at a later date for a modest fee, much less than the usual purchase price. When the new version becomes available, he goes to a qualified vendor to make the transaction.

[0371] This scenario deals with a common situation in computer software. It shows how

NVA208575.1

a purchase may include future "rights." Two important features of the scenario are that the transaction must take place at a qualified vendor, and that the transaction can be done only once per copy of the digital work purchased.

[0372] This scenario is performed as follows: The creator creates a digital work, an upgrade ticket, and a distribution license. The upgrade ticket uses the a generic ticket agent that comes with repositories. As usual, the distribution license does not have Copy or Transfer rights. He distributes a bundled copies of the work and the ticket to his distributors as well as distribution licenses.

[0373] The distributor sells the old bundled work and ticket to customers. The customer extracts the work and the ticket. He uses the work according to the agreements until the new version becomes available.

[0374] When the new work is ready, the creator gives it to distributors. The new work has a free right to copy from a distributor if a ticket is available.

[0375] The consumer goes to distributors and arranges to copy the work. The transaction offers the ticket. The distributor's repository punches the ticket and copies the new version to the consumers repository.

[0376] The consumer can now use the new version of the work. A consumer buys a digital work together with an agreement that he can upgrade to a new version at a later date for a modest fee, much less than the usual purchase price. When the new version becomes available, he goes to anyone who has the upgraded version and makes the transaction.

[0377] This scenario is like the previous one in that the transaction can only be done once per copy of the digital work purchased, but the transaction can be accomplished without the need to connect to a licensed vendor.

[0378] This scenario is similar to the previous one except that the Copy right on the new

NVA208575.1

work does not require a distribution license. The consumer can upgrade from any repository having the new version. He cannot upgrade more than once because the ticket cannot work after it has been punched. If desired, the repository can record the upgrade transaction by posting a zero cost bill to alert the creator that the upgrade has taken place.

[0379] A consumer buys a digital work and wants to make a few ephemeral copies. For example, he may want to print out a paper copy of part of a digital newspaper, or he may want to make a (first generation) analog cassette tape for playing in his car. He buys the digital work together with a ticket required for printing rights.

[0380] This scenario is like the common practice of people making cassette tapes to play in their car. If a publisher permits the making of cassette tapes, there is nothing to prevent a consumer from further copying the tapes. However, since the tapes are "analog copies," there is a noticeable quality loss with subsequent generations. The new contribution of the present invention is the use of tickets in the access controls for the making of the analog copies.

[0381] This scenario is performed as follows: The creator sells a work together with limited printing rights. The printing rights specify the kind of printer (e.g., a kind of cassette recorder or a kind of desktop paper printer) and also the kind of ticket required. The creator either bundles a limited number of tickets or sells them separately. If the tickets use the generic ticket agent, the consumer with the tickets can exercise the right at his convenience.

[0382] Professors in a business school want to put together course books of readings selected from scenario studies from various sources. The bookstore wants to be able to print the books from digital masters, without negotiating for and waiting for approval of printing of each of the scenarios. The copyright holders of the scenarios want to be sure that they are paid for every copy of their work that is printed.

[0383] On many college campuses, the hassle of obtaining copy clearances in a timely way has greatly reduced the viability of preparing course books. Print shops have become much more cautious about copying works in the absence of documented permission.

[0384] Demand Publishing is performed as follows: the creator sells a work together with printing rights for a fee. There can be rights to copy (distribute) the work between bookstore repositories, with or without fee. The printing rights specify the kind of printer. Whenever a bookstore prints one of the works (either standalone or embedded in a collection), the fee is credited to the creator automatically. To discourage unauthorized copying of the print outs, it would be possible for the printer to print tracer messages discretely on the pages identifying the printing transaction, the copy number, and any other identifying information. The tracer information could be secretly embedded in the text itself (encoded in the grey scale) or hidden in some other way.

[0385] A consumer does not know what music to purchase until he decides whether he likes it. He would like to be able to take it home and listen to it, and then decide whether to purchase. Furthermore, he would like the flexibility of paying less if he listens to it very infrequently.

[0386] This scenario just uses the capability of the approach to have multiple versions of a right on a digital work. Each version of the right has its own billing scheme. In this scenario, the creator of the work can offer the Copy right without fee, and defer billing to the exercise of the Play right. One version of the play right would allow a limited performance without fee--a right to "demo". Another version of the right could have a metered rate, of say \$0.25 per hour of play. Another version could have a fee of \$15.00 for the first play, but no fee for further playing. When the consumer exercises a play right, he specifies which version of the right is being selected and is billed accordingly.

[0387] A designer of type fonts invests several months in the design of special fonts. The most common way of obtaining revenue for this work is to sell copies of the fonts to publishers for unlimited use over unlimited periods of time. A font designer would like to charge a rate that reflects the amount that the font is used.

[0388] This scenario is performed as follows: the font designer creates a font as a digital

NVA208575.1

work. He creates versions of the Play right that bill either for metered use or "per-use". Each version of the play right would require that the player (a print layout program) be of an approved category. The font designer assigns appropriate fees to exercise the Copy right. When a publisher client wants to use a font, he includes it as input to a layout program, and is billed automatically for its use. In this way, a publisher who makes little use of a font pays less than one who uses it a lot.

[0389] Online information retrieval services typically charge for access in a way that most clients find unpredictable and uncorrelated to value or information use. The fee depends on which databases are open, dial-up connect time, how long the searches require, and which articles are printed out. There are no provisions for extracting articles or photographs, no method for paying to reuse information in new works, no distinction between having the terminal sit idly versus actively searching for data, no distinction between reading articles on the screen and doing nothing, and higher rates per search when the centralized facility is busy and slow servicing other clients. Articles can not be offloaded to the client's machine for off-site search and printing. To offer such billing or the expanded services, the service company would need a secure way to account for and bill for how information is used.

[0390] This scenario is performed as follows: The information service bundles its database as files in a repository. The information services company assigns different fees for different rights on the information files. For example, there could be a fee for copying a search database or a source file and a different fee for printing. These fees would be in addition to fees assigned by the original creator for the services. The fees for using information would be different for using them on the information service company's computers or the client's computers. This billing distinction would be controlled by having different versions of the rights, where the version for use on the service company's computer requires a digital certificate held locally. Fees for copying or printing files would be handled in the usual way, by assigning fees to exercising those rights. The distinction between searching and viewing information would be made by having different "players" for the different functions. This distinction would be maintained on the client's computers as well as the service computers. Articles could be extracted for reuse under the control of Extract and Embed rights. Thus, if a client extracts part of an

NVA208575.1

article or photograph, and then sells copies of a new digital work incorporating it, fees could automatically be collected both by the information service and earlier creators and distributors of the digital work. In this way, the information retrieval service could both offer a wider selection of services and billing that more accurately reflects the client's use of the information.

[0391] In the simplest scenario, when a user wants to print a digital document he issues a print command to the user interface. If the document has the appropriate rights and the conditions are satisfied, the user agrees to the fee and the document is printed. In other cases, the printer may be on a remote repository and it is convenient to spool the printing to a later time. This leads to several issues. The user requesting the printing wants to be sure that he is not billed for the printing until the document is actually printed. Restated, if he is billed at the time the print job is spooled but the job is canceled before printing is done, he does not want to pay. Another issue is that when spooling is permitted, there are now two times at which rights, conditions and fees could be checked: the time at which a print job is spooled and the time at which a print is made. As with all usage rights, it is possible to have rights that expire and to have rights whose fee depends on various conditions. What is needed is a means to check rights and conditions at the time that printing is actually done.

[0392] This scenario is performed as follows: A printing repository is a repository with the usual repository characteristics plus the hardware and software to enable printing. Suppose that a user logs into a home repository and wants to spool print jobs for a digital work at a remote printing repository. The user interface for this could treat this as a request to "spool" prints. Underneath this "spooling" request, however, are standard rights and requests. To support such requests, the creator of the work provides a Copy right, which can be used to copy the work to a printing repository. In the default case, this Copy right would have no fees associated for making the copy. However, the Next-Set-Of-Rights for the copy would only include the Print rights, with the usual fees for each variation of printing. This version of the Copy right could be called the "print spooling" version of the Copy right. The user's "spool request" is implemented as a Copy transaction to put a copy of the work on the printing repository, followed by Print transactions to create the prints of the work. In this way, the user is only billed for printing that is actually done. Furthermore, the rights, conditions and fees for printing the work are determined when the work

NVA208575.1

is about to be printed.

[0393] Thus, a system for enforcing the usage rights of digital works is disclosed. While the embodiments disclosed herein are preferred, it will be appreciated from this teaching that various alternative, modifications, variations or improvements therein may be made by those skilled in the art, which are intended to be encompassed by the appended claims which define the invention.

GLOSSARY

[0394] Authorization Repository - A special type of repository which provides an authorization service. An authorization may be specified by a usage right. The authorization must be obtained before manner of use specified by the right may be exercised.

[0395] Billing Clearinghouse - A financial institution or the like whose purpose is to reconcile billing information received from credit servers. The billing clearinghouse may generate bills to users or alternatively, credit and debit accounts involved in the commercial transactions.

[0396] Billing Transactions - The protocol used by which a repository reports billing information to a credit server.

[0397] Clearinghouse Transactions - The protocol used between a credit server and a clearinghouse.

[0398] Composite Digital Work - A digital work comprised of distinguishable parts. Each of the distinguishable parts is itself a digital work which may have associated usage rights. Content - The digital information that is to be rendered of digital work.

[0399] Copy Owner - A term which refers to the party who owns a digital work stored in a repository. In the typical case, this party has purchased various rights to the document for

NVA208575.1

printing, viewing, transferring, or other specific uses.

[0400] Creator - A term which refers to a party who produces a digital work.

[0401] Credit Server - A device which collects and reports billing information for a repository. In many implementations, this could be built as part of a repository. It requires a means for periodically communicating with a billing clearinghouse.

[0402] Description Tree - A structure which describes the location of content and the usage rights and usage fees for a digital work. A description tree is comprised of description blocks. Each description block corresponds to a digital work or to an interest (typically a revenue bearing interest) in a digital work.

[0403] Digital Work (Work) - Digital content with any associated usage rights. Such digital content may represent music, a magazine or book, or a multimedia composition.
Distributor - A term which refers to a party who legitimately obtains a copy of a digital work and offers it for sale.

[0404] Identification (Digital) Certificate - A signed digital message that attests to the identity of the possessor. Typically, digital certificates are encrypted in the private key of a well-known master repository.

[0405] Master Repository - A special type of repository which issues identification certificates and distributes lists of repositories whose integrity have been compromised and which should be denied access to digital works (referred to as repository "hotlists".)

[0406] Public Key Encryption - An encryption technique used for secure transmission of messages on a communication channel. Key pairs are used for the encryption and decryption of messages. Typically one key is referred to as the public key and the other is the private key. The keys are inverses of each other from the perspective of encryption. Restated, a digital work that is encrypted by one key in the pair can be decrypted only by the other.

[0407] Registration Transactions - The protocol used between repositories to establish a trusted session.

[0408] Rendering Repository - A special type of repository which is typically coupled to a rendering system. The rendering repository will typically be embodied within the secure boundaries of a rendering system.

[0409] Rendering System - The combination of a rendering repository and a rendering device. Examples of a rendering systems include printing systems, display systems, general purpose computer systems, video systems or audio systems.

[0410] repository - Conceptually a set of functional specifications defining core functionality in the support of usage rights. A repository is a trusted system in that it maintains physical, communications and behavioral integrity.

[0411] requester Mode - A mode of a repository where it is requesting access to a digital work.

[0412] venue Owners - A term which refers to the parties that maintain an interest in collecting fees for document use or who stand to lose revenue if illegitimate copies of the digital work are made.

[0413] Server Mode - A mode of a repository where it is processing an incoming request to access a digital work.

[0414] null Description Block - A special type of description block designating an interest in a digital work, but which does not add content. This will typically be added by a distributor of a digital work to add their fees.

Transactions - A term used to refer to the protocols by which repositories communicate.

[0415] Usage Fees - A fee charged to a requester for access to a digital work. Usage fees are specified within the usage rights language.

[0416] Usage Rights - An indication of the manner of use by which a digital work may be used or distributed, as well as any conditions on which manner of use is premised.

[0417] Usage Transactions - A set of protocols by which repositories communicate in the exercise of a usage rights. Each usage right has it's own transaction steps.

FOR THE TESTER

What is claimed is:

1. A digital work adapted to be distributed within a system for controlling at least one of the distribution and use of digital works, said digital work comprising:

digital content representing a portion of a digital work suitable for being rendered by a rendering device; and
usage rights associated with said digital content, said usage rights specifying a manner of use indicating one or more stated purposes for which the digital work can be at least one of used and distributed by an authorized party.

2. The digital work as recited in claim 1, wherein said usage rights further specify status information indicating the status of the digital work.

3. The digital work as recited in claim 1, wherein said usage rights further specify a usage fee associated with exercise of the manner of use, said usage fee comprising a fee type and fee parameters.

4. The digital work as recited in claim 3 wherein said fee type is a metered use fee and said fee parameters comprise a fee unit and a time unit.

5. The digital work as recited in claim 3 wherein said fee type is a per use fee and said fee parameter comprises a fee unit.

6. The digital work as recited in claim 3 wherein said fee type is a best price fee and said fee parameters comprise an identifier to identify said best price.

7. The digital work as recited in claim 3 wherein said fee type is a scheduled fee and said fee parameters comprise time units and fee units.

8. The digital work as recited in claim 3 wherein said fee type is a mark-up fee and said fee parameters comprise a mark-up percentage.

NVA208575.1

9. The digital work as recited in claim 1 wherein said digital content and said usage rights are stored on the same physical device.

10. The digital work as recited in claim 1, wherein said digital content and said usage rights are stored on different physical devices.

11. A method for creating a digital work to be used in a system for reporting fees associated with at least one of distribution and use of the digital work, said method comprising:

a) obtaining digital content representing a portion of a digital work suitable for being rendered by a rendering device;

b) associating usage rights with the digital content, the usage rights specifying a manner of use indicating one or more stated purposes for which the digital work can be at least one of distributed; and

c) storing the digital content and the usage rights as the digital work.

12. A method as recited in claim 11, wherein said storing step comprises storing the digital content and the usage rights on the same physical device.

13. A method as recited in claim 11, wherein said storing step comprises storing the digital content and the usage rights on different physical devices.

14. A method as recited in claim 11, wherein said usage rights further specify a usage fee associated with exercise of the manner of use.

15. The method as recited in claim 14, wherein said usage fee comprises usage fee information including a fee type and fee parameters.

16. The method as recited in claim 15 wherein said fee type is a metered use fee and said fee parameters comprise a fee unit and a time unit.

17. The method as recited in claim 15 wherein said fee type is a per use fee and said fee

NVA208575.1

parameters comprise a fee unit.

18. The method as recited in claim 15 wherein said fee type is a best price fee and said fee parameters comprise an identifier to identify the best price.

19. The method as recited in claim 15 wherein said fee type is a scheduled fee and said fee parameters comprise time units and fee units.

20. The method as recited in claim 17 wherein said fee type is a mark-up fee and said fee parameters comprise a mark-up percentage.

21. A digital work adapted to be distributed within a system for controlling at least one of the distribution and use of digital works, said system having a mechanism for reporting status information relating to digital works, said digital work comprising:

digital content representing a portion of a digital work suitable for being rendered by a rendering device; and

usage rights associated with said digital content, said usage rights specifying a manner of use indicating one or more stated purposes for which the digital work can be at least one of used and distributed, said usage rights also including state information indicating the extent of the manner of use exercised by an authorized user.

22. A digital work as recited in claim 21, wherein the state information comprises information indicating the number of copies of the digital work that are in use.

23. A digital work as recited in claim 21, wherein the state information comprises information indicating the number of time units for which the digital work can be used.

24. A digital work as recited in claim 21, wherein the state information comprises information indicating the revenue owner of the digital work.

25. A digital work as recited in claim 21, wherein the state information comprises information indicating the publication date of the digital work.
26. A digital work as recited in claim 21, wherein the state information comprises information indicating the history of at least one of the use and distribution of the digital work.
27. A method for creating a digital work to be used in a system for reporting fees associated with at least one of distribution and use of the digital work, said method comprising:
- a) obtaining digital content representing a portion of a digital work suitable for being rendered by a rendering device;
 - b) associating usage rights with the digital content, the usage rights specifying a manner of use indicating one or more stated purposes for which the digital work can be at least one of distributed, said usage rights also including state information indicating the extent of the manner of use exercised by an authorized user; and
 - c) storing the digital content and the usage rights as the digital work.
28. A method as recited in claim 27, wherein said storing step comprises storing the digital content and the usage rights on the same physical device.
29. A method as recited in claim 27, wherein said storing step comprises storing the digital content and the usage rights on different physical devices.
30. A method as recited in claim 27, wherein the state information comprises information indicating the number of copies of the digital work that are in use.
31. A method as recited in claim 27, wherein the state information comprises information indicating the number of time units for which the digital work can be used.
32. A method as recited in claim 27, wherein the state information comprises information indicating the revenue owner of the digital work.

33. A method as recited in claim 27, wherein the state information comprises information indicating the publication date of the digital work.

34. A method as recited in claim 27, wherein the state information comprises information indicating the history of at least one of the use and distribution of the digital work.

1045951-124704
TOTAL TESTS

ABSTRACT

[0418] Digital work adapted to be distributed within a system for controlling at least one of the distribution and use of digital works. The digital work includes: digital content representing a portion of a digital work suitable for being rendered by a rendering device and usage rights associated with the digital content. The usage rights specify a manner of use indicating one or more stated purposes for which the digital work can be at least one of used and distributed by an authorized party.

TOPT TEST



UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov



Bib Data Sheet

CONFIRMATION NO. 5128

SERIAL NUMBER 10/015,951	FILING DATE 12/17/2001 RULE	CLASS 705	GROUP ART UNIT 2161	ATTORNEY DOCKET NO. 111325-88
APPLICANTS Mark J. Stefik, Portola Valley, CA; Peter L.T. Pirolli, San Francisco, CA;				
** CONTINUING DATA ***** THIS APPLICATION IS A CON OF 09/778,001 02/07/2001 WHICH IS A DIV OF 08/967,084 11/10/1997 PAT 6,236,971 WHICH IS A CON OF 08/344,760 11/23/1994 ABN				
** FOREIGN APPLICATIONS *****				
IF REQUIRED, FOREIGN FILING LICENSE GRANTED ** 01/10/2002				
Foreign Priority claimed ☐ yes ☐ no 35 USC 119 (a-d) conditions ☐ yes ☐ no ☐ Met after met Allowance Verified and Acknowledged _____ Examiner's Signature Initials		STATE OR COUNTRY CA	SHEETS DRAWING 13	TOTAL CLAIMS 34
				INDEPENDENT CLAIMS 4
ADDRESS 22204				
TITLE Digital works having usage rights and method for creating the same				
FILING FEE RECEIVED 1206	FEES: Authority has been given in Paper No. _____ to charge/credit DEPOSIT ACCOUNT No. _____ for following:		☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other _____ ☐ Credit	

PATENT APPLICATION FEE DETERMINATION RECORD

Effective October 1, 2001

Application or Docket Number

10.015951

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	34	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	34 minus 20 = *	14
INDEPENDENT CLAIMS	4 minus 3 = *	1
MULTIPLE DEPENDENT CLAIM PRESENT ☐		

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE ☐

OR

OTHER THAN SMALL ENTITY

RATE	FEE
BASIC FEE	370.00
X\$ 9=	
X42=	
+140=	
TOTAL	

OR

OR

OR

OR

OR

RATE	FEE
BASIC FEE	740.00
X\$18=	
X84=	
+280=	
TOTAL	

CLAIMS AS AMENDED - PART II

	(Column 1)	(Column 2)	(Column 3)
AMENDMENT A	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total	*	Minus	**
Independent	*	Minus	***
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐			

SMALL ENTITY

OR

OTHER THAN SMALL ENTITY

RATE	ADDITIONAL FEE
X\$ 9=	
X42=	
+140=	
TOTAL ADDIT. FEE	

OR

OR

OR

OR

RATE	ADDITIONAL FEE
X\$18=	
X84=	
+280=	
TOTAL ADDIT. FEE	

	(Column 1)	(Column 2)	(Column 3)
AMENDMENT B	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total	*	Minus	**
Independent	*	Minus	***
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐			

RATE	ADDITIONAL FEE
X\$ 9=	
X42=	
+140=	
TOTAL ADDIT. FEE	

OR

OR

OR

OR

RATE	ADDITIONAL FEE
X\$18=	
X84=	
+280=	
TOTAL ADDIT. FEE	

	(Column 1)	(Column 2)	(Column 3)
AMENDMENT C	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total	*	Minus	**
Independent	*	Minus	***
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐			

RATE	ADDITIONAL FEE
X\$ 9=	
X42=	
+140=	
TOTAL ADDIT. FEE	

OR

OR

OR

OR

RATE	ADDITIONAL FEE
X\$18=	
X84=	
+280=	
TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."

***If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."

The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.



UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

APPLICATION NUMBER	FILING/RECEIPT DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NUMBER
10/015,951	12/17/2001	Mark J. Stefik	111325-88

CONFIRMATION NO. 5128

FORMALITIES LETTER



OC000000007296565

22204
NIXON PEABODY, LLP
8180 GREENSBORO DRIVE
SUITE 800
MCLEAN, VA 22102

Date Mailed: 01/11/2002

NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL APPLICATION

FILED UNDER 37 CFR 1.53(b)

Filing Date Granted

An application number and filing date have been accorded to this application. The item(s) indicated below, however, are missing. Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items and pay any fees required below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- The statutory basic filing fee is missing.
Applicant must submit \$ 740 to complete the basic filing fee for a non-small entity. If appropriate, applicant may make a written assertion of entitlement to small entity status and pay the small entity filing fee (37 CFR 1.27).
- Total additional claim fee(s) for this application is \$336.
 - \$252 for 14 total claims over 20.
 - \$84 for 1 independent claims over 3 .
- The oath or declaration is missing.
A properly signed oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date, is required.
- To avoid abandonment, a late filing fee or oath or declaration surcharge as set forth in 37 CFR 1.16(l) of \$130 for a non-small entity, must be submitted with the missing items identified in this letter.
- **The balance due by applicant is \$ 1206.**

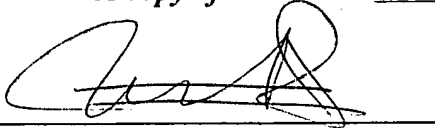
The application is informal since it does not comply with the regulations for the reason(s) indicated below.

The required item(s) identified below must be timely submitted to avoid abandonment:

- Substitute drawings in compliance with 37 CFR 1.84 because:
 - drawing sheets do not have the appropriate margin(s) (see 37 CFR 1.84(g)). Each sheet must include a top margin of at least 2.5 cm. (1 inch), a left side margin of at least 2.5 cm. (1 inch), a right side margin of at least 1.5 cm. (5/8 inch), and a bottom margin of at least 1.0 cm. (3/8 inch);

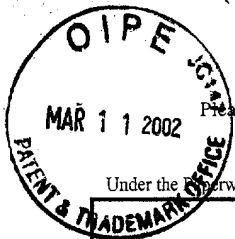
.....

A copy of this notice MUST be returned with the reply.

A handwritten signature in black ink, appearing to be "J. A. [unclear]", is written over a horizontal line.

Customer Service Center
Initial Patent Examination Division (703) 308-1202

PART 3 - OFFICE COPY



Please type a plus sign (+) inside this box → [+]

PTO/SB/21 (08-00)

Approved for use through 10/31/2002. OMB 0651-0031

U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM (to be used for all correspondence after initial filing)		Application Number	10/015,951
		Filing Date	December 17, 2001
		First Named Inventor	Mark J. Stefik et al.
		Group Art Unit	2161
		Examiner Name	Not yet assigned
Total Number of Pages in This Submission		Attorney Docket Number	111325-88

ENCLOSURES (check all that apply)

☒ Fee Transmittal Form ☒ Fee Attached ☐ Amendment / Reply ☐ After Final ☐ Affidavits/declaration(s) ☐ Extension of Time Request ☐ Express Abandonment Request ☐ Information Disclosure Statement ☐ Certified Copy of Priority Document(s) ☒ Response to Missing Parts/ Incomplete Application ☒ Response to Missing Parts under 37 CFR 1.52 or 1.53	☒ Assignment Papers (for an Application) ☒ Drawing(s) ☒ Declaration and Power of Attorney ☐ Licensing-related Papers ☐ Petition ☐ Petition to Convert to a Provisional Application ☐ Power of Attorney, Revocation Change of Correspondence Address ☐ Terminal Disclaimer ☐ Request for Refund ☐ CD, Number of CD(s) _____	☐ After Allowance Communication to Group ☒ Other: Application Data Sheet
Remarks ☒ The Commissioner is hereby authorized to charge any additional fees required or credit any overpayments to Deposit Account No. 19-2380 for the above identified docket number.		

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	<u>Marc S. Kaufman</u> Nixon Peabody LLP 8180 Greensboro Drive Suite 800 McLean, VA 22102
Signature	
Date	March 11, 2002

CERTIFICATE OF MAILING

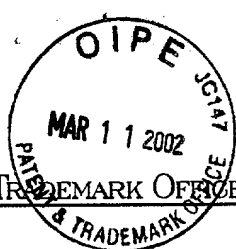
I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, Washington, DC 20231 on this date: 		
Type or printed name		
Signature		Date _____

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, Washington, DC 20231.

NVA218034.1



UNITED STATES PATENT AND TRADEMARK OFFICE



COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

APPLICATION NUMBER	FILING/RECEIPT DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NUMBER
10/015,951	12/17/2001	Mark J. Stefik	111325-88

CONFIRMATION NO. 5128

22204
NIXON PEABODY, LLP
8180 GREENSBORO DRIVE
SUITE 800
MCLEAN, VA 22102

FORMALITIES LETTER



OC000000007296565

Date Mailed: 01/11/2002

NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL APPLICATION

03/15/2002 BABRAHA1 00000053 10015951

FILED UNDER 37 CFR 1.53(b)

01 FC:101	740.00 OP
02 FC:105	130.00 OP
03 FC:103	252.00 OP
04 FC:102	84.00 OP

Filing Date Granted

An application number and filing date have been accorded to this application. The item(s) indicated below, however, are missing. Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items and pay any fees required below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- The statutory basic filing fee is missing.
Applicant must submit \$ 740 to complete the basic filing fee for a non-small entity. If appropriate, applicant may make a written assertion of entitlement to small entity status and pay the small entity filing fee (37 CFR 1.27).
- Total additional claim fee(s) for this application is \$336.
 - \$252 for 14 total claims over 20.
 - \$84 for 1 independent claims over 3 .
- The oath or declaration is missing.
A properly signed oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date, is required.
- To avoid abandonment, a late filing fee or oath or declaration surcharge as set forth in 37 CFR 1.16(l) of \$130 for a non-small entity, must be submitted with the missing items identified in this letter.
- The balance due by applicant is \$ 1206.**

The application is informal since it does not comply with the regulations for the reason(s) indicated below.

The required item(s) identified below must be timely submitted to avoid abandonment:

- Substitute drawings in compliance with 37 CFR 1.84 because:
 - drawing sheets do not have the appropriate margin(s) (see 37 CFR 1.84(g)). Each sheet must include a top margin of at least 2.5 cm. (1 inch), a left side margin of at least 2.5 cm. (1 inch), a right side margin of at least 1.5 cm. (5/8 inch), and a bottom margin of at least 1.0 cm. (3/8 inch);

A copy of this notice MUST be returned with the reply.

er

Customer Service Center

Initial Patent Examination Division (703) 308-1202

PART 2 - COPY TO BE RETURNED WITH RESPONSE

Figure 1 is a schematic representation of the experimental design. It shows a vertical timeline of events for two groups: 'Control' and 'Experimental'. The timeline starts with 'Baseline' and ends with 'Post-test'. The 'Control' group receives 'Baseline', 'Training', and 'Post-test'. The 'Experimental' group receives 'Baseline', 'Training', 'Post-test', and 'Follow-up'. The 'Training' phase is marked with a box labeled 'Training' and a box labeled 'Post-test'.



Docket No. 111325-88

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Re Application of)
Mark J. Stefik et al.) Group Art Unit: 2161
Serial No.: 10/015,951) Examiner: Not yet assigned
Filed: December 17, 2001)
For: Digital Works Having Usage Rights And)
Method For Creating The Same)

SUBMISSION OF FORMAL DRAWINGS

Commissioner for Patents
Washington, D. C. 20231

March 11, 2002

Sir:

Submitted herewith are thirteen (13) sheets of formal drawings (Figs. 1-19) in connection with the above-identified application.

Respectfully submitted,

NIXON PEABODY LLP

Marc S. Kaufman
Registration No. 35,212

MSK/kbd

NIXON PEABODY LLP
8180 Greensboro Drive, Suite 800
McLean, Virginia 22102
Telephone (703) 790-9110

APPLICATION DATA SHEET

Electronic Version 0.0.11

Stylesheet Version: 1.0

Publication Filing Type:

Application Type:

new-utility

utility

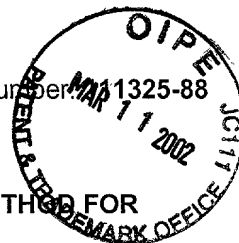
Title of Invention:

DIGITAL WORKS HAVING USAGE RIGHTS AND METHOD FOR
CREATING THE SAME

Suggested Representative Figure:

Fig.1

Attorney Docket Number



Legal Representative:

Attorney or Agent:

Marc S. KAUFMAN

Registration Number:

35212

Customer Number Correspondence Address:

22204

22204

Continuity Data:

This application is a continuation of 09/778,001 A1 2001-02-07 US Pending
which is a division of 08/967,084 A1 1997-11-10 US 6,236,971 2001-05-22 A1 US
which is a continuation of 08/344,760 A1 1994-11-23 US Abandoned

Assignee (Publish): CONTENTGUARD HOLDINGS, INC.

INVENTOR(s):

Primary Citizenship: US
Given Name: Mark
Middle Name: J.
Family Name: STEFIK
Residence City: Portola Valley
Residence State: CA
Residence Country: US
Address: 10 Portola Green Circle
Portola Valley California, 94028 US

Primary Citizenship: Canada
Given Name: Peter
Middle Name: L.T.
Family Name: PIROLI
Residence City: San Francisco
Residence State: CA
Residence Country: US
Address: 2958 Sloat Blvd.
San Francisco California, 94116 US

Figure 1

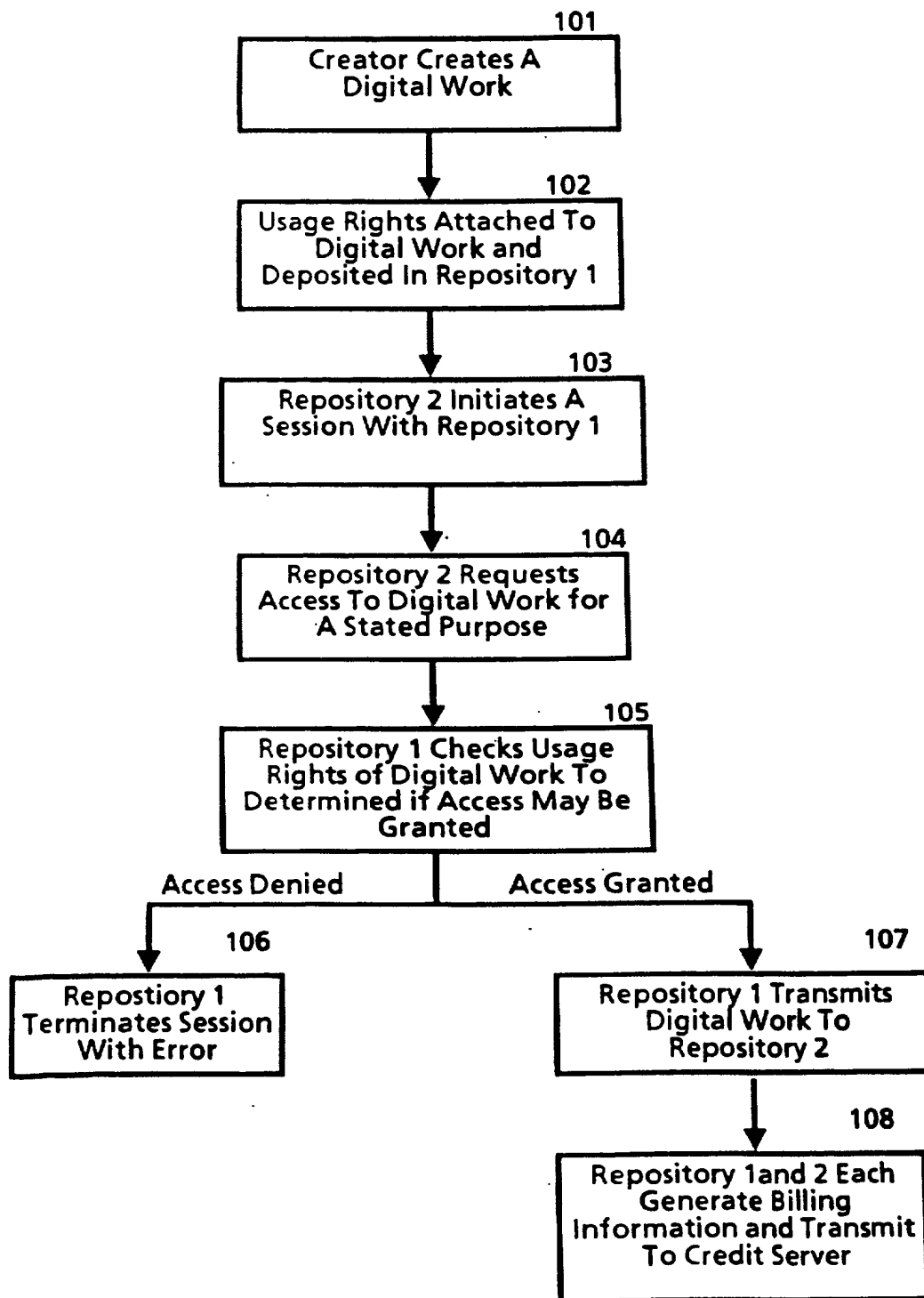


Figure 2

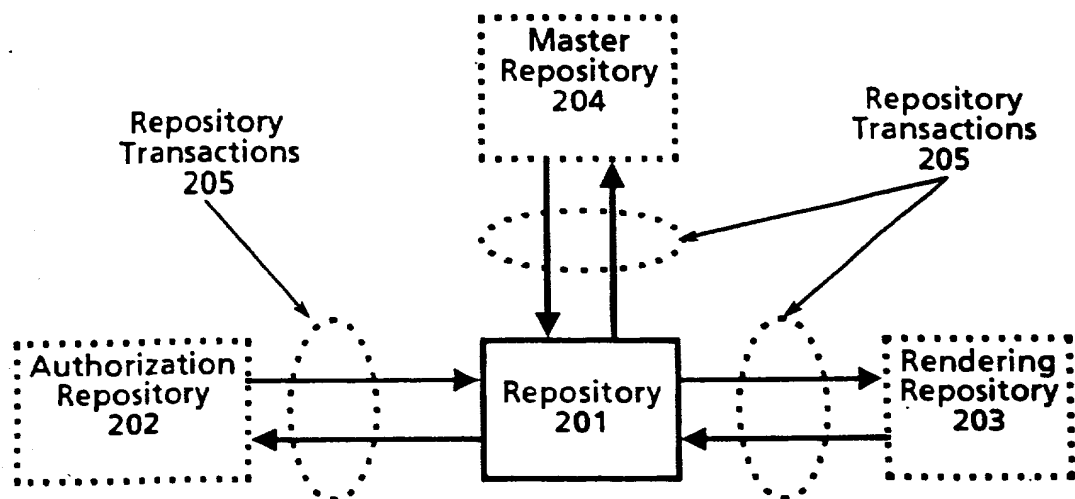


Figure 3

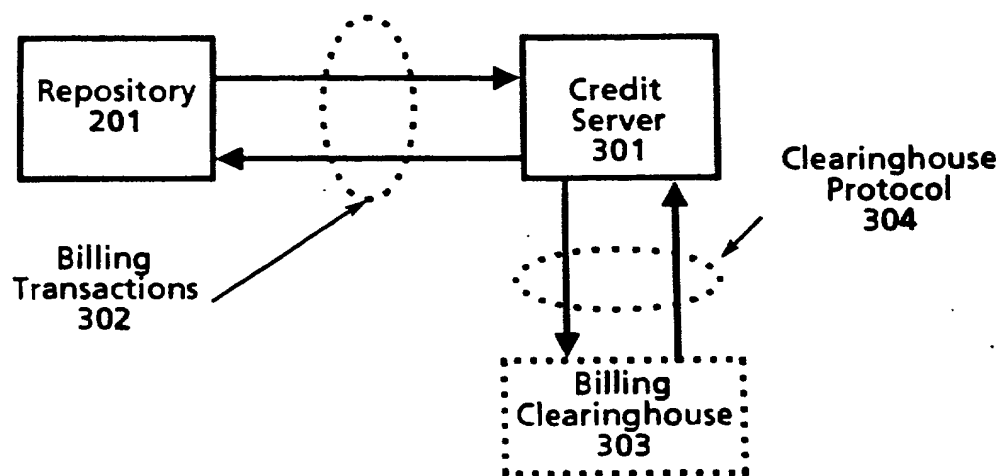


Figure 4a

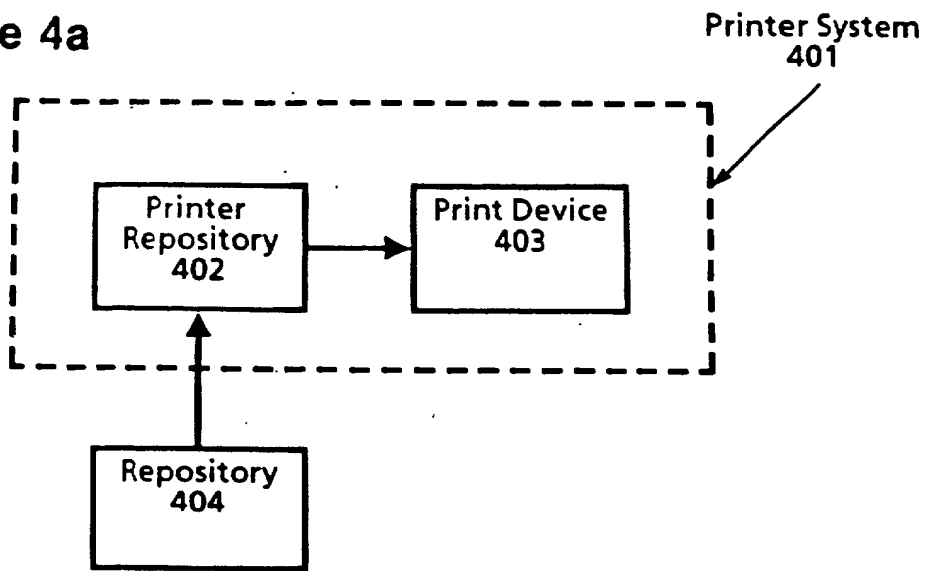
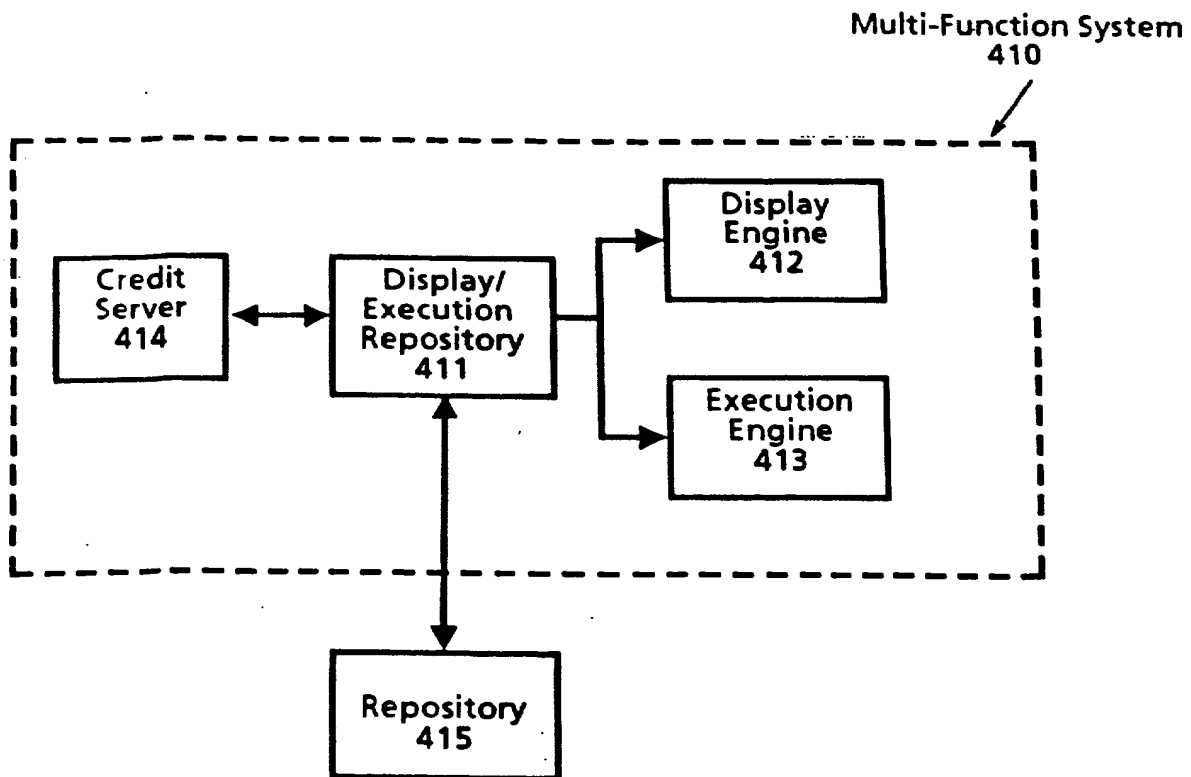


Figure 4b



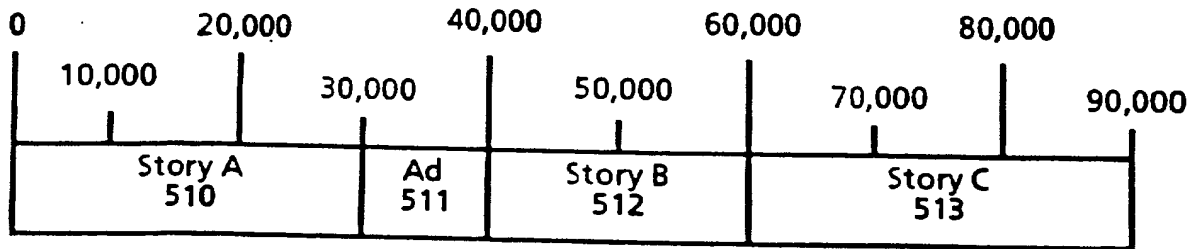


Figure 5

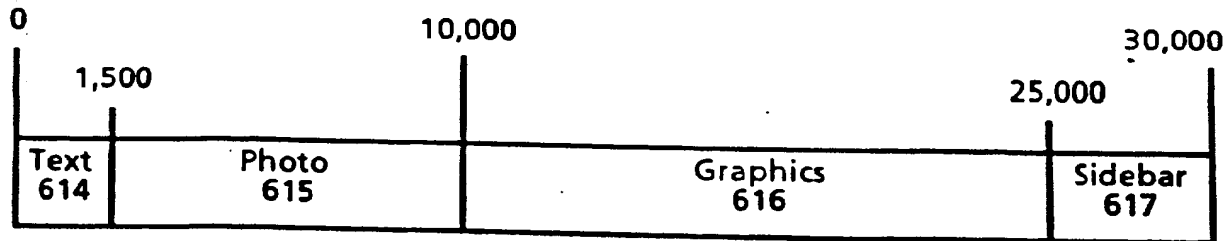


Figure 6

Figure 7

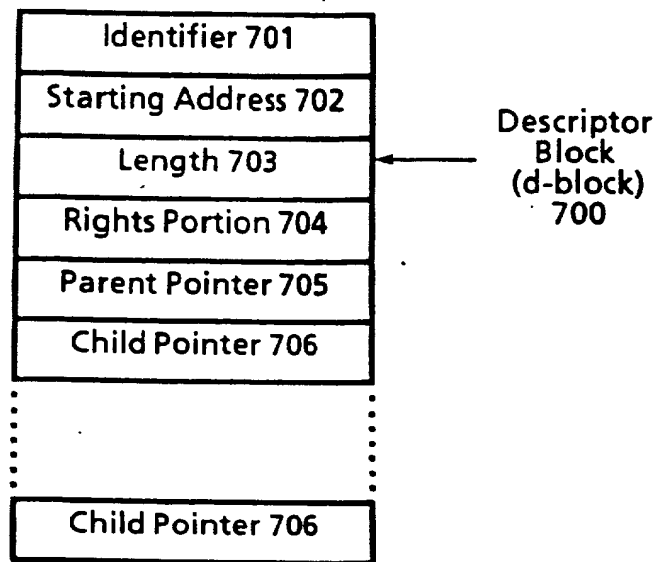


Figure 8

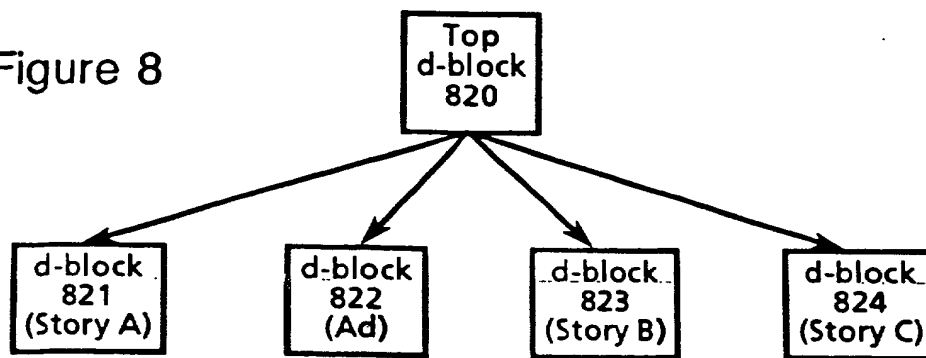


Figure 9

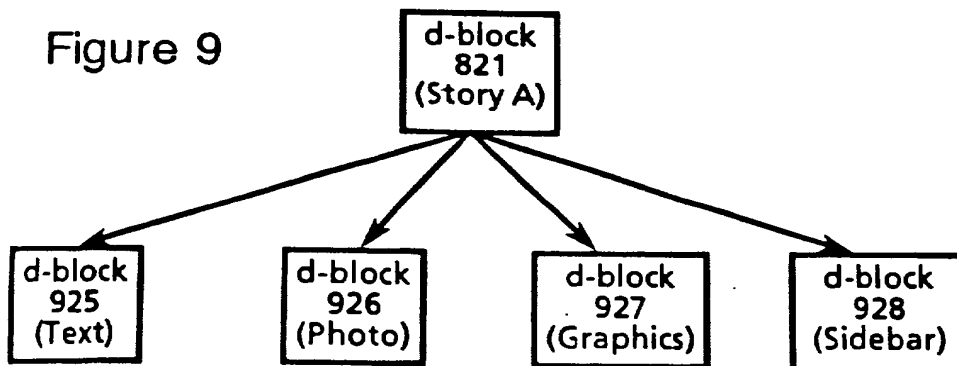


Figure 10

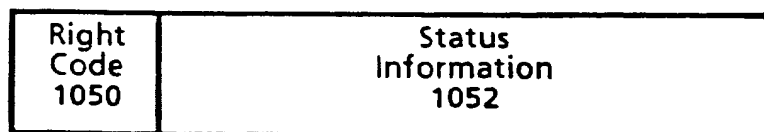


Figure 14

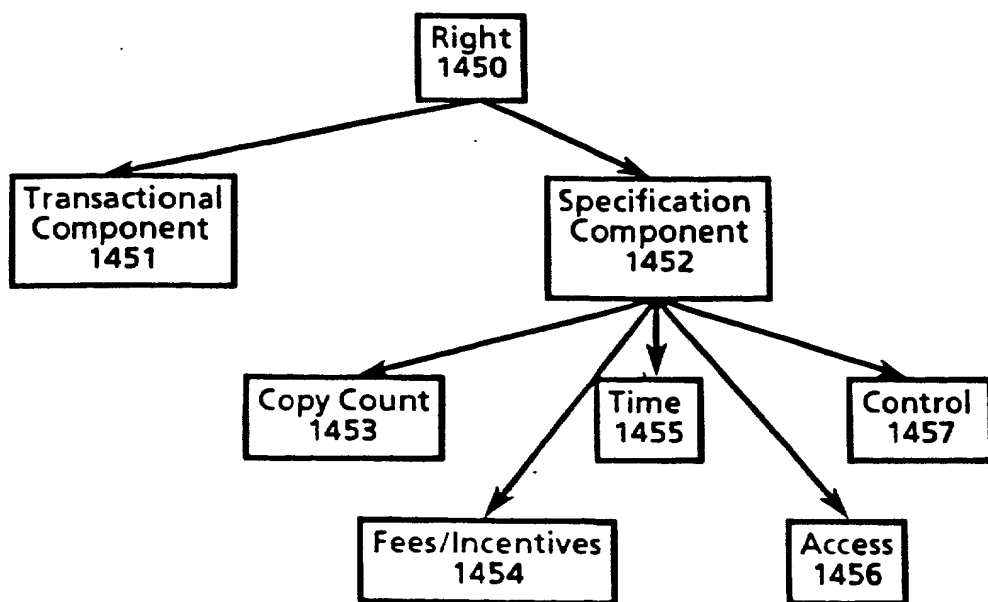


Figure 11

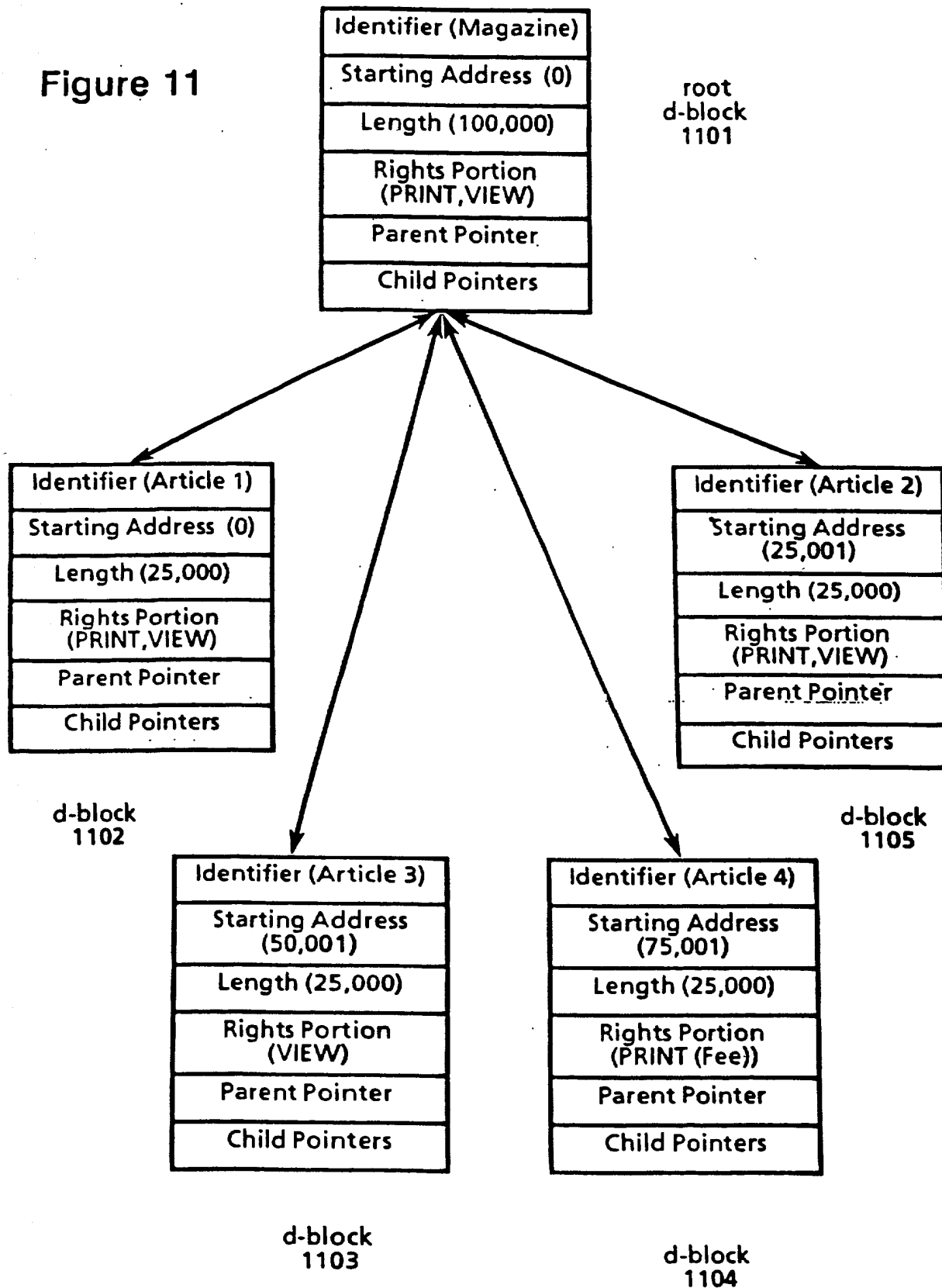


Figure 12

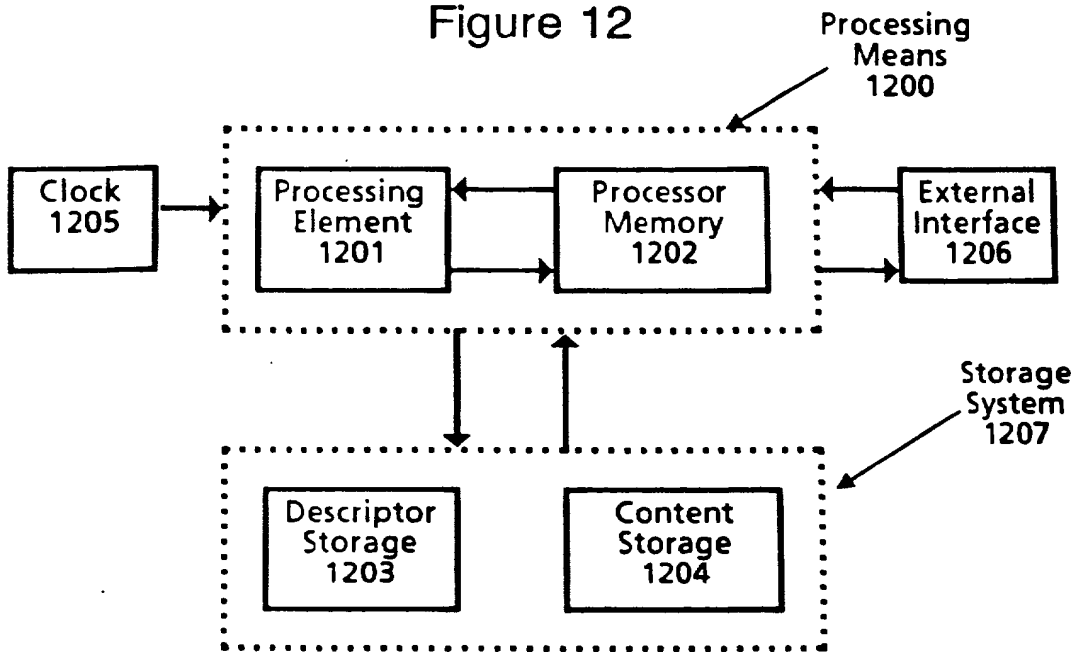


Figure 13

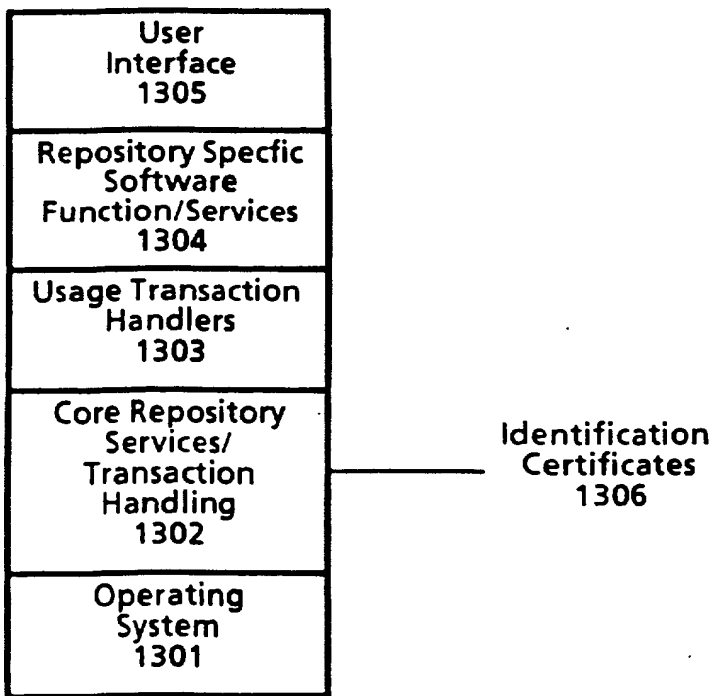


FIGURE 15

1501 ~ Digital Work Rights := (Rights*)

1502 ~ Right := (Right-Code {Copy-Count} {Control-Spec} {Time-Spec} {Access-Spec} {Fee-Spec})

1503 ~ Right-Code := Render-Code | Transport-Code | File-Management-Code | Derivative-Works-Code | Configuration-Code

1504 ~ Render-Code := [Play : {Player: Player-ID} | Print : {Printer: Printer-ID}]

1505 ~ Transport-Code := [Copy | Transfer | Loan {Remaining-Rights: Next-Set-of-Rights}] { (Next-Copy-Rights: Next-Set-of-Rights) }

1506 ~ File-Management-Code := Backup {Back-Up-Copy-Rights: Next-Set-of-Rights} | Restore | Delete | Folder | Directory {Name: Hide-Local | Hide-Remote} {Parts: Hide-Local | Hide-Remote}

1507 ~ Derivative-Works-Code := [Extract | Embed | Edit {Process: Process-ID}] { (Next-Copy-Rights: Next-Set-of-Rights) }

1508 ~ Configuration-Code := Install | Uninstall

1509 ~ Next-Set-of-Rights := { (Add: Set-Of-Rights) } { (Delete: Set-Of-Rights) } { (Replace: Set-Of-Rights) } { (Keep: Set-Of-Rights) }

1510 ~ Copy-Count := (Copies: positive-integer | 0 | Unlimited)

1511 ~ Control-Spec := (Control: {Restrictable | Unrestrictable} {Unchargeable | Chargeable})

1512 ~ Time-Spec := { (Fixed-Interval | Sliding-Interval | Meter-Time) Until: Expiration-Date }

1513 ~ Fixed-Interval := From: Start-Time

1514 ~ Sliding-Interval := Interval: Use-Duration

1515 ~ Meter-Time := Time-Remaining: Remaining-Use

1516 ~ Access-Spec := { (SC: Security-Class) {Authorization: Authorization-ID*} {Other-Authorization: Authorization-ID*} {Ticket: Ticket-ID} }

1517 ~ Fee-Spec := {Scheduled-Discount} Regular-Fee-Spec | Scheduled-Fee-Spec | Markup-Spec

1518 ~ Scheduled-Discount := Scheduled-Discount: (Scheduled-Discount: (Time-Spec Percentage)*)

1519 ~ Regular-Fee-Spec := { (Fee: | Incentive:) [Per-Use-Spec | Metered-Rate-Spec | Best-Price-Spec | Call-For-Price-Spec] {Min: Money-Unit Per: Time-Spec} {Max: Money-Unit Per: Time-Spec} To: Account-ID }

1520 ~ Per-Use-Spec := Per-Use: Money-unit

1521 ~ Metered-Rate-Spec := Metered: Money-Unit Per: Time-Spec

1522 ~ Best-Price-Spec := Best-Price: Money-unit Max: Money-unit

1523 ~ Call-For-Price-Spec := Call-For-Price

1524 ~ Scheduled-Fee-Spec := (Schedule: (Time-Spec Regular-Fee-Spec)*)

1525 ~ Markup-Spec := Markup: percentage To: Account-ID

Figure 16

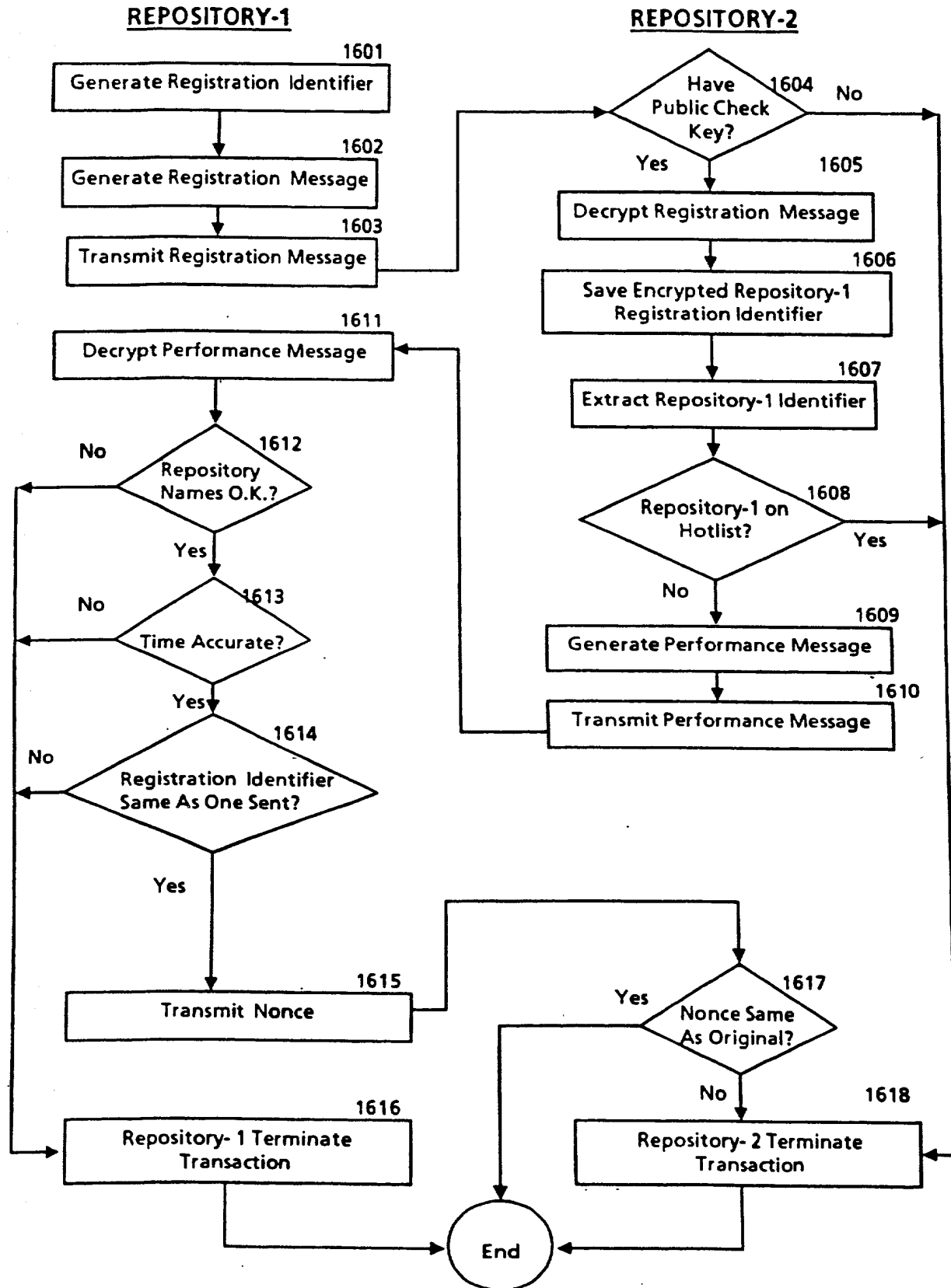


Figure 17

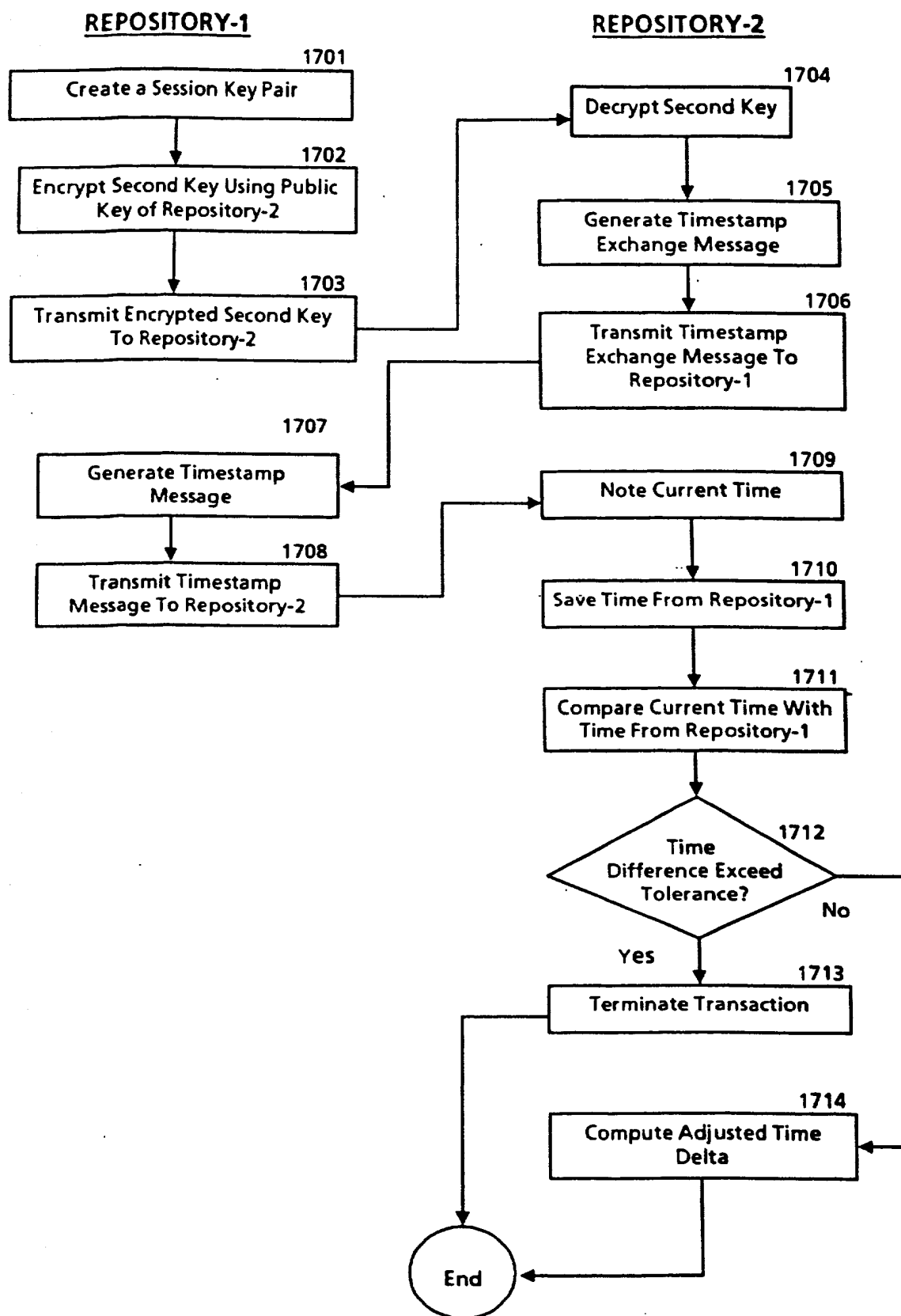


Figure 18

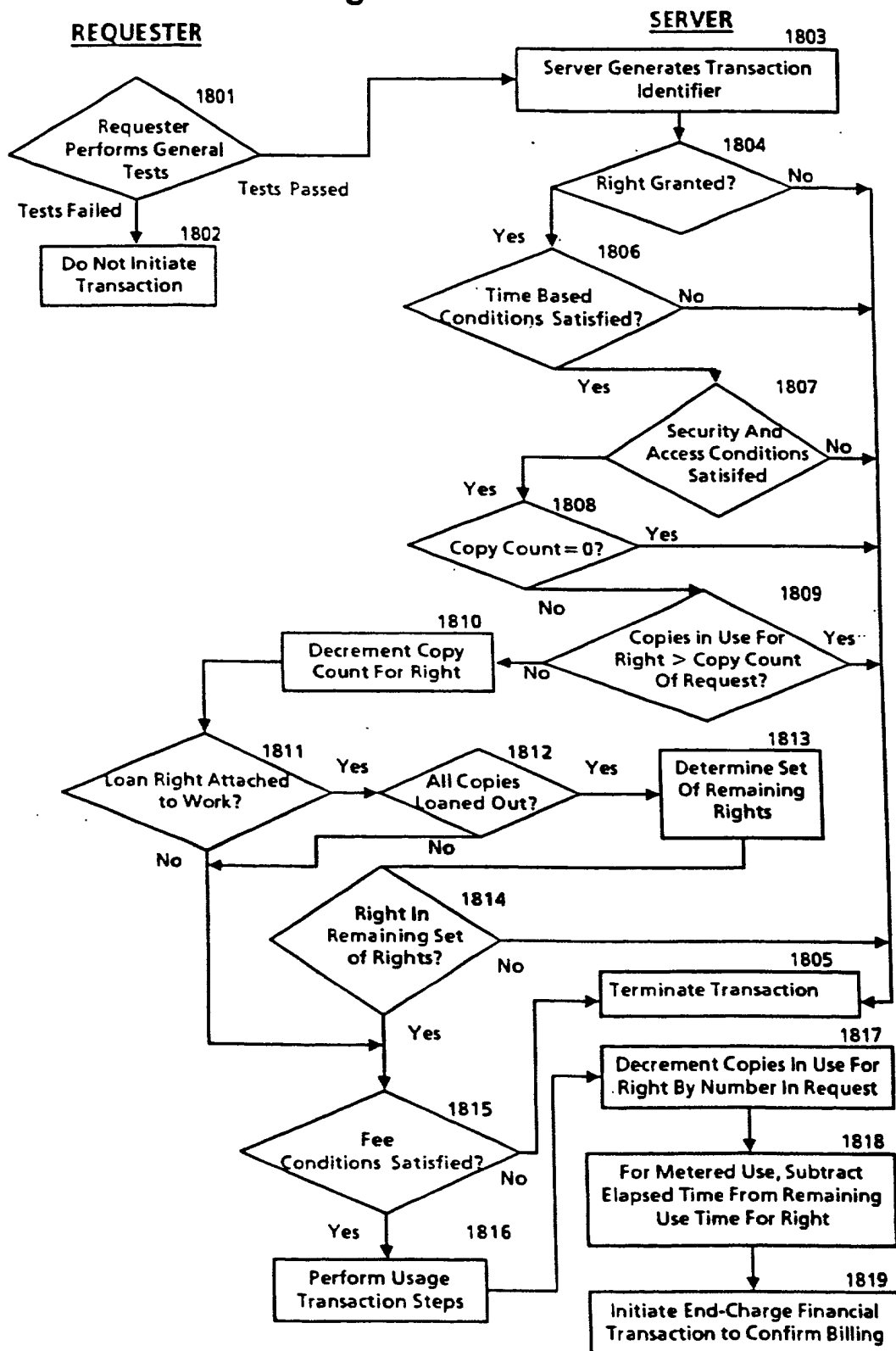
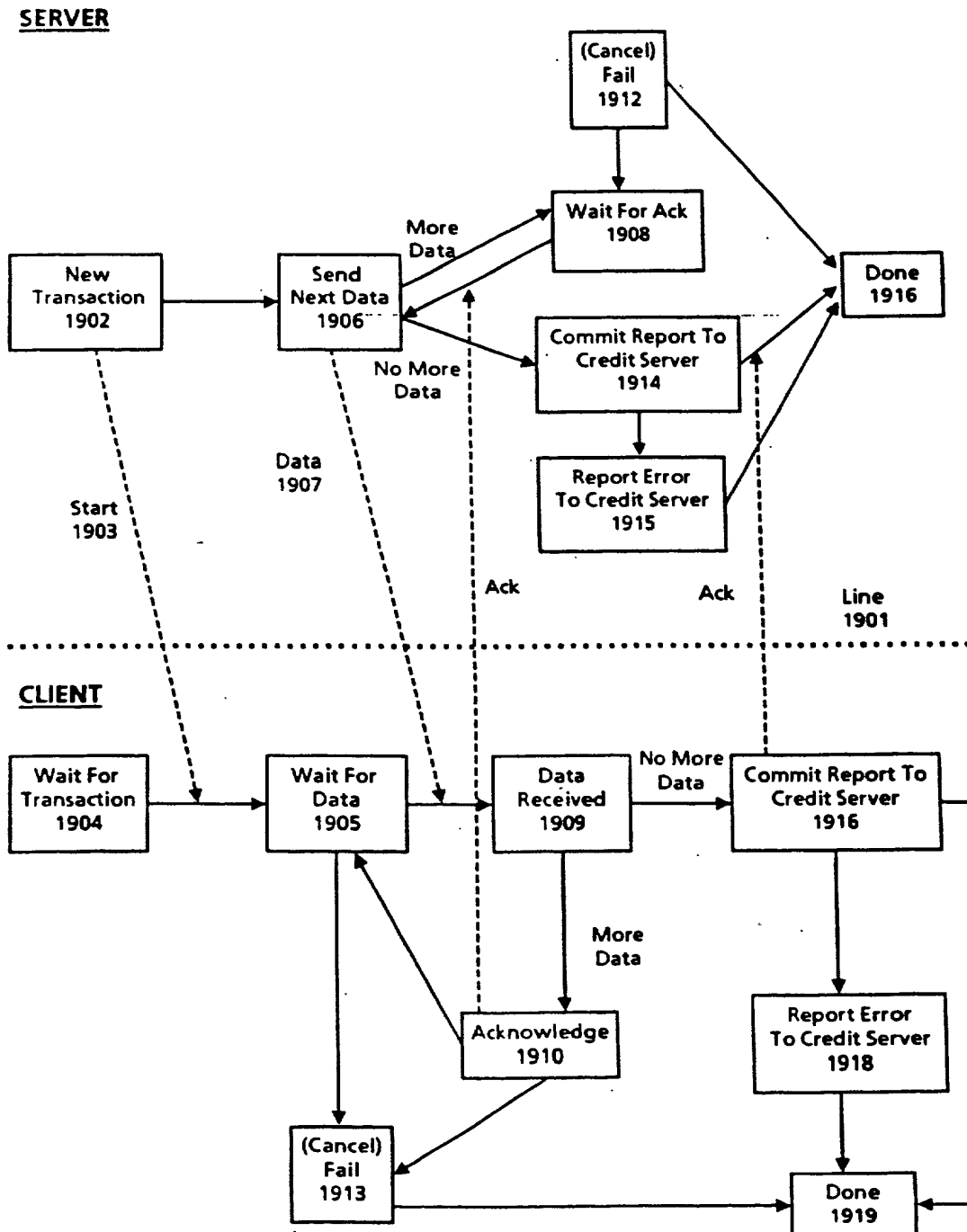


Figure 19



Please type a plus sign (+) inside this box → [+]

PTO/SB/01 (12-97)

Approved for use through 9/30/00. OMB 0651-0032

Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OBM control number.

**DECLARATION FOR UTILITY OR
DESIGN PATENT APPLICATION
(37 CFR 1.63)**

Attorney Docket Number	111325-88
First Named Inventor	Mark J. Stefik et al.
COMPLETE IF KNOWN	
Application Number	10/015,951
Filing Date	December 17, 2001
Group Art Unit	2161
Examiner Name	Not yet assigned

☐ Declaration Submitted With Initial Filing OR ☒ Declaration Submitted after Initial Filing (surcharge (37 CFR 1.16(d)) required)

As a below named inventor, I hereby declare that:

My residence, post office address, and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

Digital Works Having Usage Rights And Method For Creating The Same

(Title of the Invention)

the specification of which

☐ is attached hereto
OR

☒ was filed on (MM/DD/YYYY) **12/17/2001** As United States Application Number or PCT International Application Number **10/015,951**
And was amended on (MM/DD/YYYY) _____ (If applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment specifically referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR 1.56.

I hereby claim foreign priority benefits under 35 U.S.C. 119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or of any PCT international application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application Number(s)	Country	Foreign Filing Date (MM/DD/YYYY)	Priority Not Claimed	Certified Copy Attached	
				YES	No
			☐ ☐ ☐ ☐	☐ ☐ ☐ ☐	☐ ☐ ☐ ☐

☐ Additional foreign application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto:

I hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional application(s) listed below.

Application Number(s)	Filing Date (MM/DD/YYYY)	☐ Additional provisional application Numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.

[Page 1 of 3]

Burden Hour Statement: This form is estimated to take 0.4 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissions for Patents, Washington, DC 20231.

Please type a plus sign (+) inside this box → [+]

PTO/SB/01 (12-97)

Approved for use through 9/30/00. OMB 0651-0032

Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OBM control number.

DECLARATION – Utility or Design Patent Application

I hereby claim the benefit under 35 U.S.C. 120 of any United States application(s), or 365© of any PT international application designating the United States of America, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of 35 U.S.C. 112, I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application.

U.S. Parent Application or PCT Parent Number	Parent Filing Date (MM/DD/YYYY)	Parent Patent Number (if applicable)
This Application is a Continuation of USSN 09/778,001 which is a Divisional Application of USSN 08/967,084 which is a Continuation of USSN 08/344,760	02/07/2001 11/10/1997 11/23/1994	6,236,971 now abandoned

[] Additional U.S. or PCT international application are listed on a supplemental priority date sheet PTO/SB/02B attached hereto.

As a named inventor, I hereby appoint the following registered practitioner(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith: [X] Customer Number 22204

OR

[X] Registered practitioner(s) name/registration number listed below.

Name	Registration Number	Name	Registration Number
Stuart J. Friedman	24,312	Eric J. Robinson	38,285
Charles M. Leedom, Jr.	26,477	Daniel S. Song	43,143
David S. Safran	27,997	Marc S. Kaufman	35,212
Thomas W. Cole	28,290	Corinne R. Gorski	34,339
Donald R. Studebaker	32,815	Jason H. Vick	45,285
Jeffrey L. Costellia	35,483	Luan C. Do	38,434
Tim L. Brackett, Jr.	36,092		

Direct all correspondence to: [X] Customer Number 22204

Name: Marc S. Kaufman

Firm: NIXON PEABODY LLP

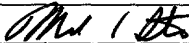
Address: 8180 Greensboro Drive, Suite 800

City: McLean State: VA ZIP: 22102

Country: United States Telephone: (703) 790-9110 FAX: (703) 883-0370

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Name of Sole or First Inventor: [] A petition has been filed for this unsigned inventor.

Given Name (first and middle [if any])		Family Name or Surname	
Mark J.		STEFIK	
Inventor's Signature: 		Date: 5 Feb '02	
Mailing Address (Street or P.O. Box): 55 Big Tree Way 10 Portola Green Circle			
City: Woodside Portola Valley		State: California ZIP: 94062 94025 Country: US	
Residence: City:		State: Country:	
Citizenship: USA			

[] Additional inventors are being named on the _____ Supplemental Additional Inventor(s) sheet(s) PTO/SB/02A attached hereto.

Please type a plus sign (+) inside this box → [+]

PTO/SB/02A (10-00)

Approved for use through 10/31/2002. OMB 0651-0032

Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OBM control number.

DECLARATION	ADDITIONAL INVENTOR(S) Supplemental Sheet Page ____ of ____

Name of Additional Joint Inventor, if any:		☐ A petition has been filed for this unsigned inventor	
Given Name Peter L.T.		Family Name Or Surname PIROLI	
Inventor's Signature <i>P.L. Pirolli</i> (PP)		Date 2/5/02	
Mailing Address (Street or P.O. Box): 44 Wildwood Place 2958 Sloat Blvd			
City: El Cerrito San Francisco (PP)		State: California ZIP: 94530 94116 (PP) Country: US	
Residence: City:		State: Country:	
Citizenship: Canadian			
Name of Additional Joint Inventor, if any:		☐ A petition has been filed for this unsigned inventor	
Given Name		Family Name Or Surname	
Inventor's Signature		Date	
Mailing Address (Street or P.O. Box):			
City:		State: ZIP: Country:	
Residence: City:		State: Country:	
Citizenship:			
Name of Additional Joint Inventor, if any:		☐ A petition has been filed for this unsigned inventor	
Given Name		Family Name Or Surname	
Inventor's Signature		Date	
Mailing Address (Street or P.O. Box):			
City:		State: ZIP: Country:	
Residence: City:		State: Country:	
Citizenship:			

Burden Hour Statement: This form is estimated to take 21 minutes to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, Washington, DC 20231.

[Page 3 of 3]

Figure 1

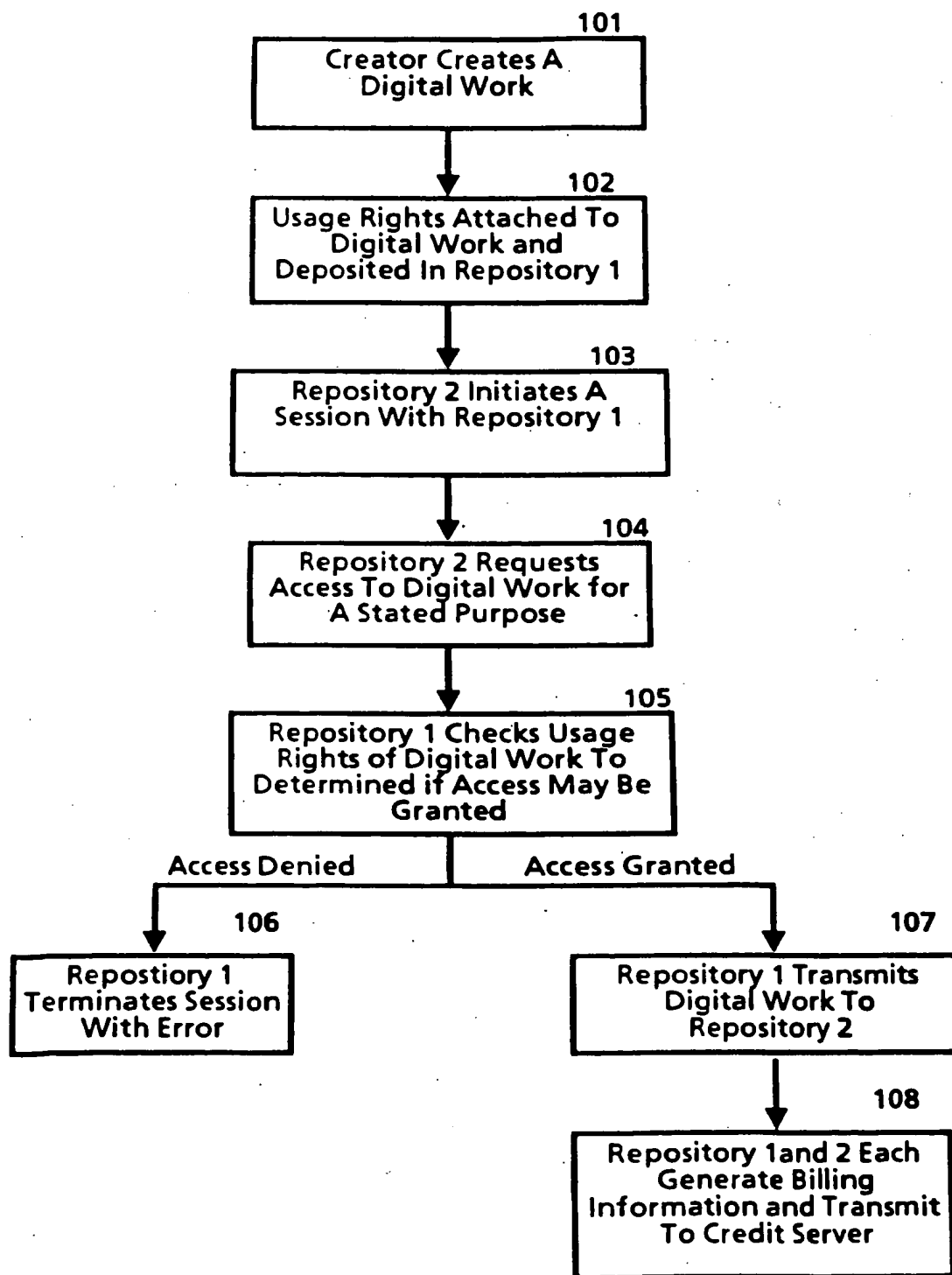


Figure 2

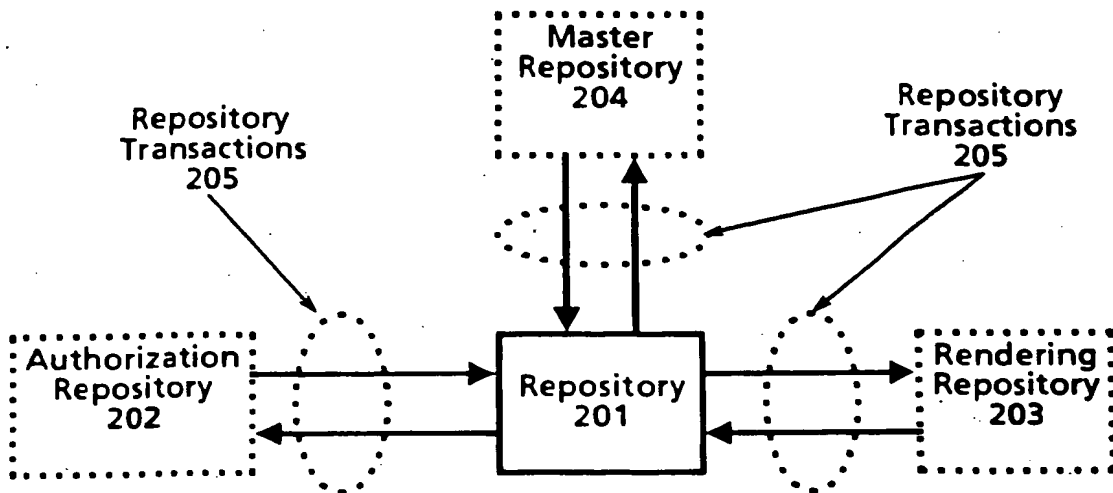


Figure 3

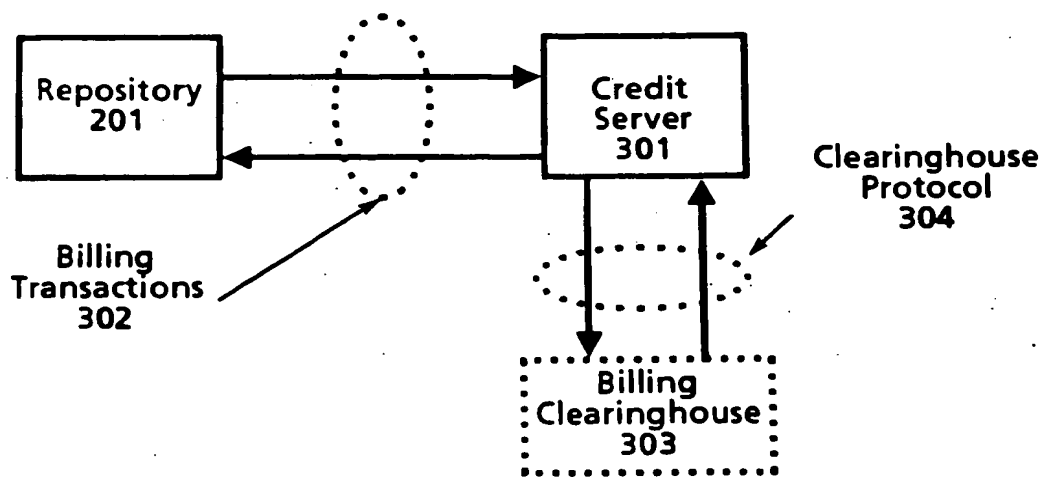


Figure 4a

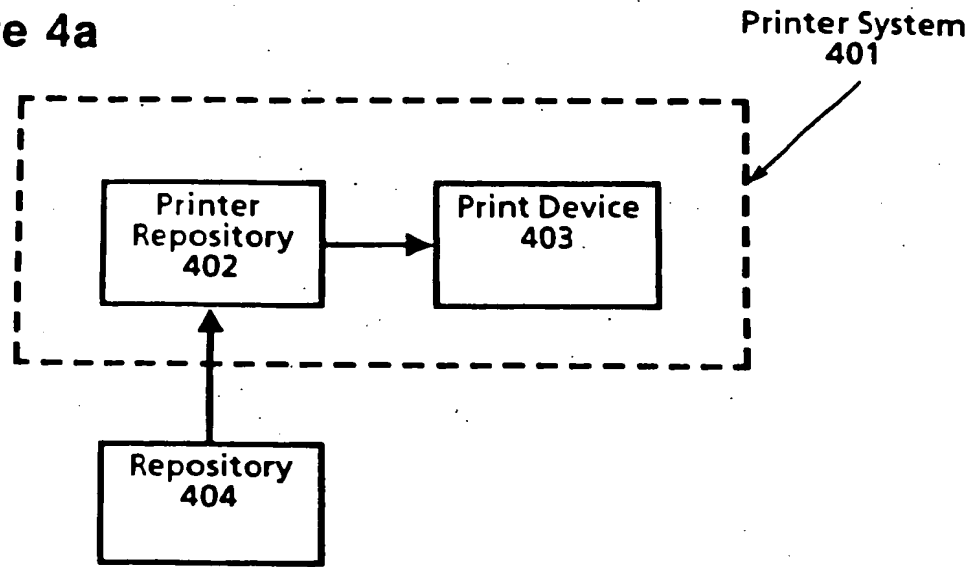
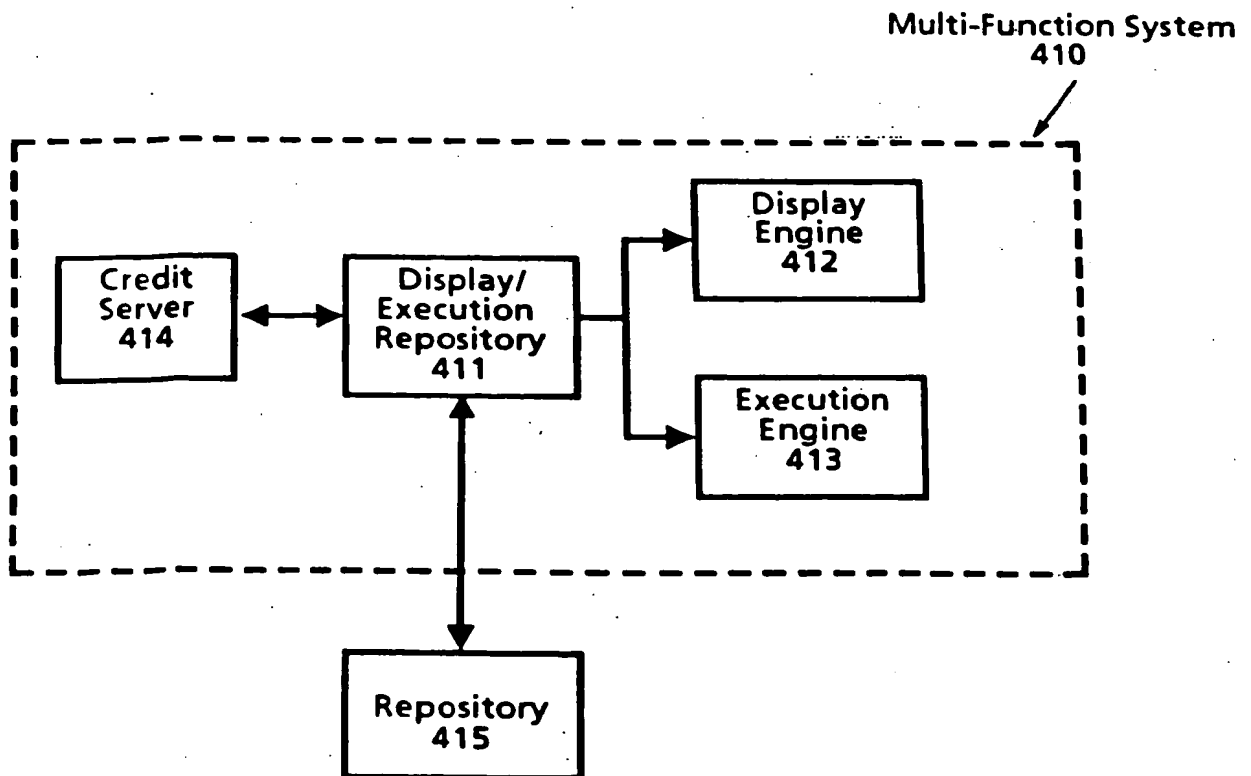


Figure 4b



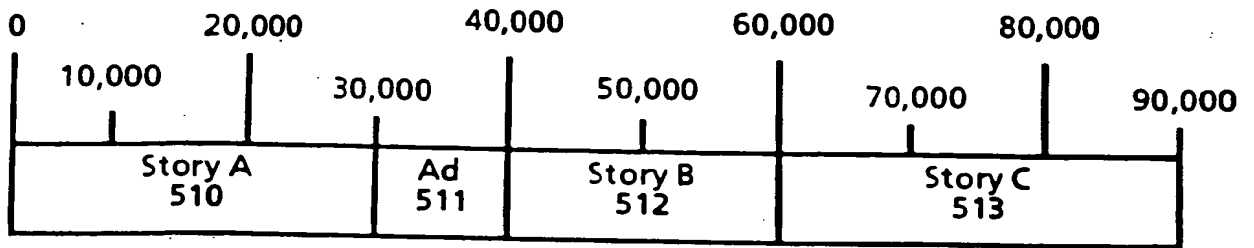


Figure 5

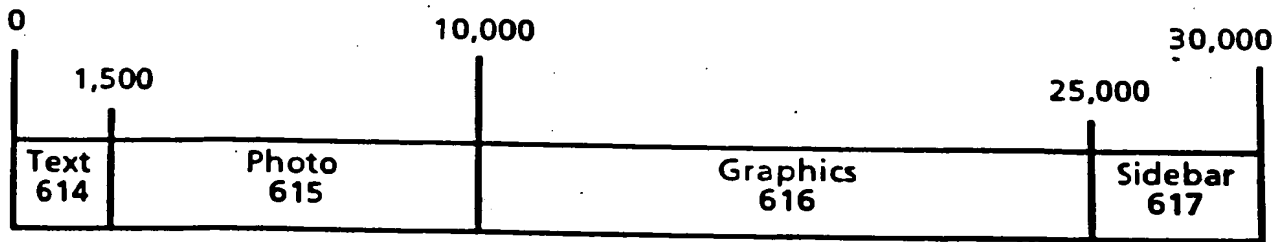


Figure 6

Figure 7

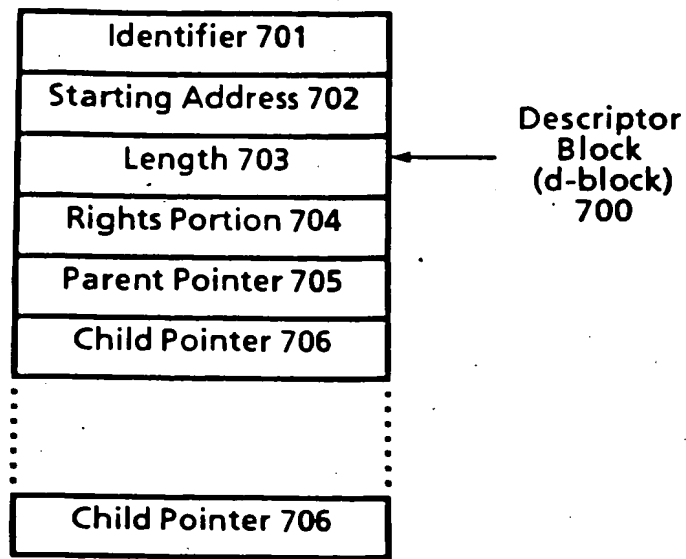


Figure 8

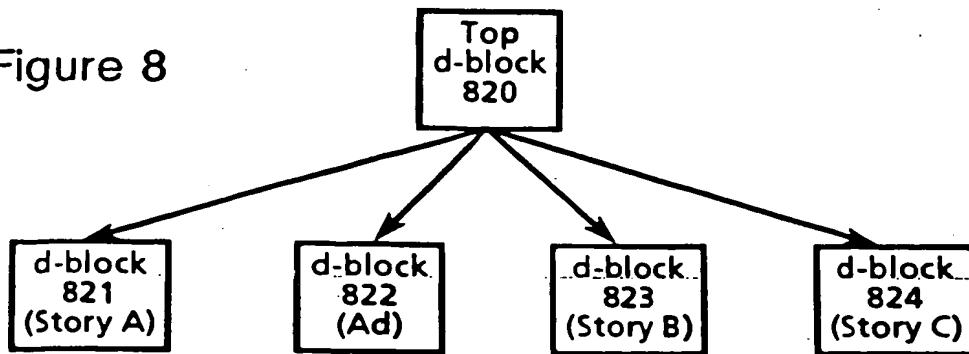


Figure 9

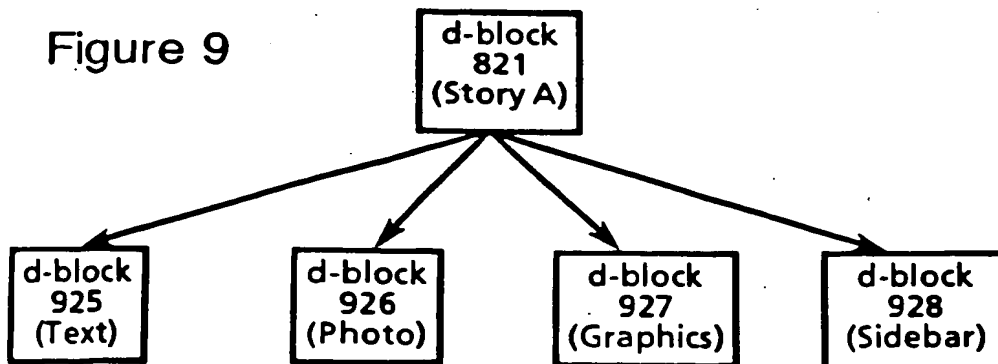


Figure 10



Figure 14

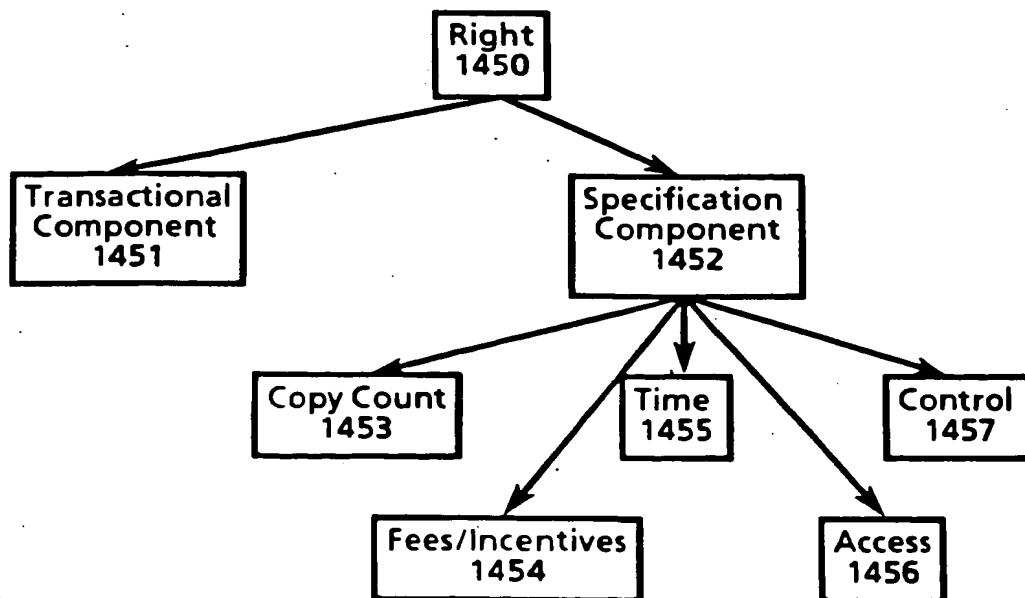


Figure 11

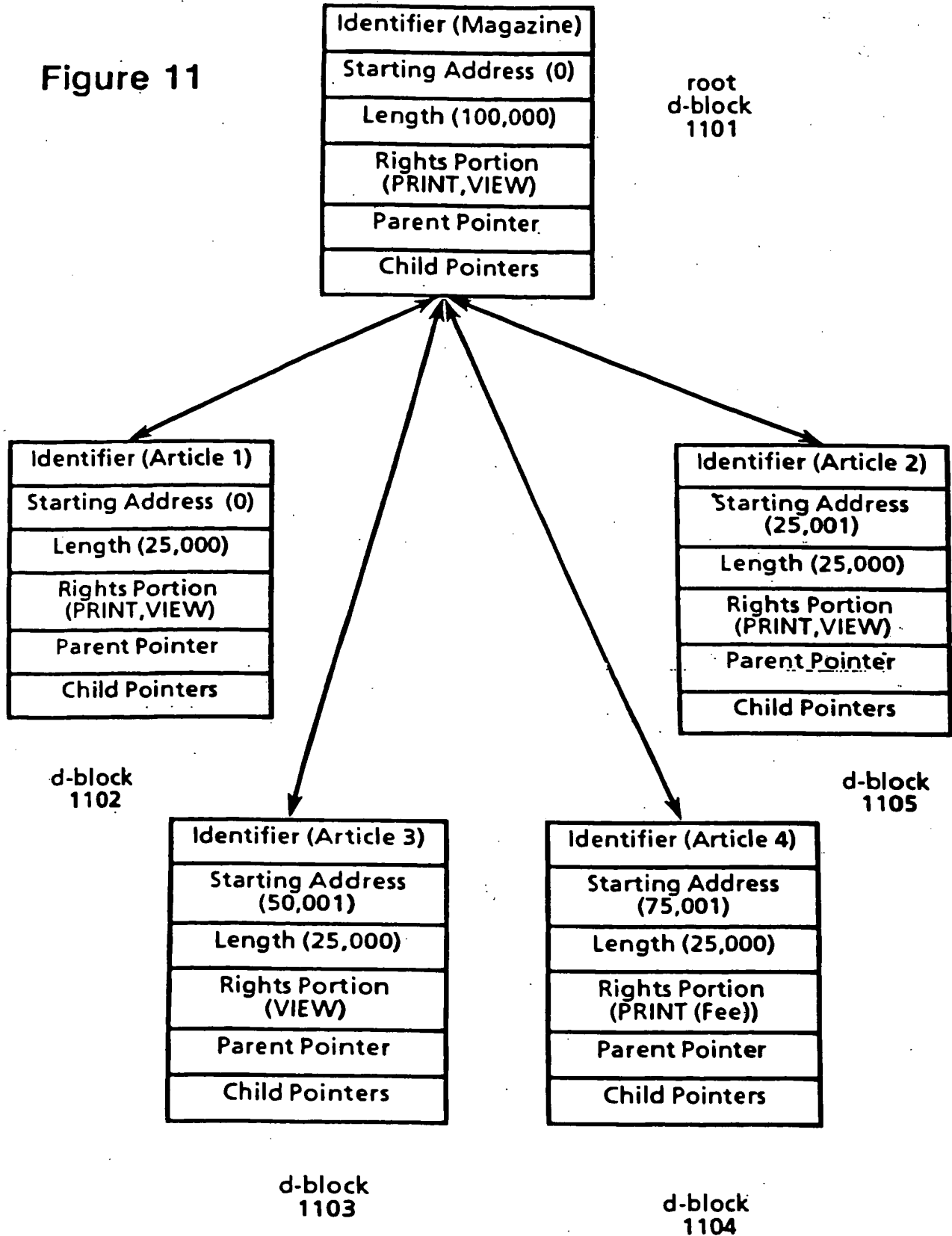


Figure 12

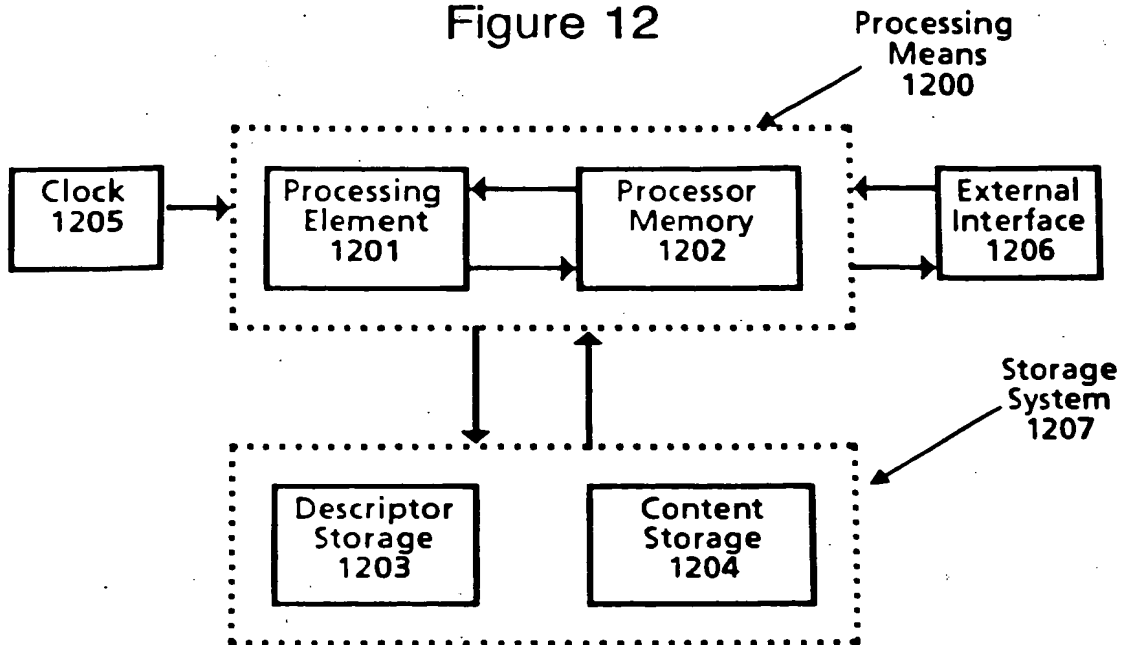


Figure 13

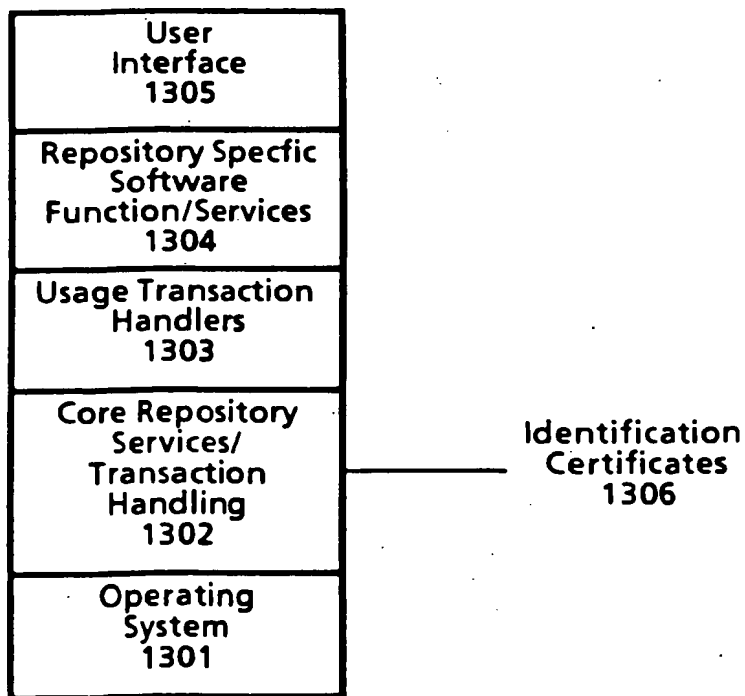


FIGURE 15

1501 ~ Digital Work Rights := (Rights*)

1502 ~ Right := (Right-Code {Copy-Count} {Control-Spec} {Time-Spec} {Access-Spec} {Fee-Spec})

1503 ~ Right-Code := Render-Code | Transport-Code | File-Management-Code | Derivative-Works-Code | Configuration-Code

1504 ~ Render-Code := [Play : {Player: Player-ID} | Print : {Printer: Printer-ID}]

1505 ~ Transport-Code := [Copy | Transfer | Loan {Remaining-Rights: Next-Set-of-Rights}] { (Next-Copy-Rights: Next-Set-of-Rights) }

1506 ~ File-Management-Code := Backup {Back-Up-Copy-Rights: Next-Set-of-Rights} | Restore | Delete | Folder | Directory {Name: Hide-Local | Hide-Remote} {Parts: Hide-Local | Hide-Remote}

1507 ~ Derivative-Works-Code := [Extract | Embed | Edit {Process: Process-ID}] {Next-Copy-Rights: Next-Set-of-Rights}

1508 ~ Configuration-Code := Install | Uninstall

1509 ~ Next-Set-of-Rights := { (Add: Set-Of-Rights) } { (Delete: Set-Of-Rights) } { (Replace: Set-Of-Rights) } { (Keep: Set-Of-Rights) }

1510 ~ Copy-Count := (Copies: positive-integer | 0 | Unlimited)

1511 ~ Control-Spec := (Control: {Restrictable | Unrestrictable} {Unchargeable | Chargeable})

1512 ~ Time-Spec := { (Fixed-Interval | Sliding-Interval | Meter-Time) Until: Expiration-Date }

1513 ~ Fixed-Interval := From: Start-Time

1514 ~ Sliding-Interval := Interval: Use-Duration

1515 ~ Meter-Time := Time-Remaining: Remaining-Use

1516 ~ Access-Spec := { (SC: Security-Class) {Authorization: Authorization-ID*} {Other-Authorization: Authorization-ID*} {Ticket: Ticket-ID} }

1517 ~ Fee-Spec := {Scheduled-Discount} Regular-Fee-Spec | Scheduled-Fee-Spec | Markup-Spec

1518 ~ Scheduled-Discount := Scheduled-Discount: (Scheduled-Discount: (Time-Spec Percentage)*)

1519 ~ Regular-Fee-Spec := { (Fee: | Incentive:) } [Per-Use-Spec | Metered-Rate-Spec | Best-Price-Spec | Call-For-Price-Spec] {Min: Money-Unit Per: Time-Spec} {Max: Money-Unit Per: Time-Spec} To: Account-ID

1520 ~ Per-Use-Spec := Per-Use: Money-unit

1521 ~ Metered-Rate-Spec := Metered: Money-Unit Per: Time-Spec

1522 ~ Best-Price-Spec := Best-Price: Money-unit Max: Money-unit

1523 ~ Call-For-Price-Spec := Call-For-Price

1524 ~ Scheduled-Fee-Spec := (Schedule: (Time-Spec Regular-Fee-Spec)*)

1525 ~ Markup-Spec := Markup: percentage To: Account-ID

Figure 16

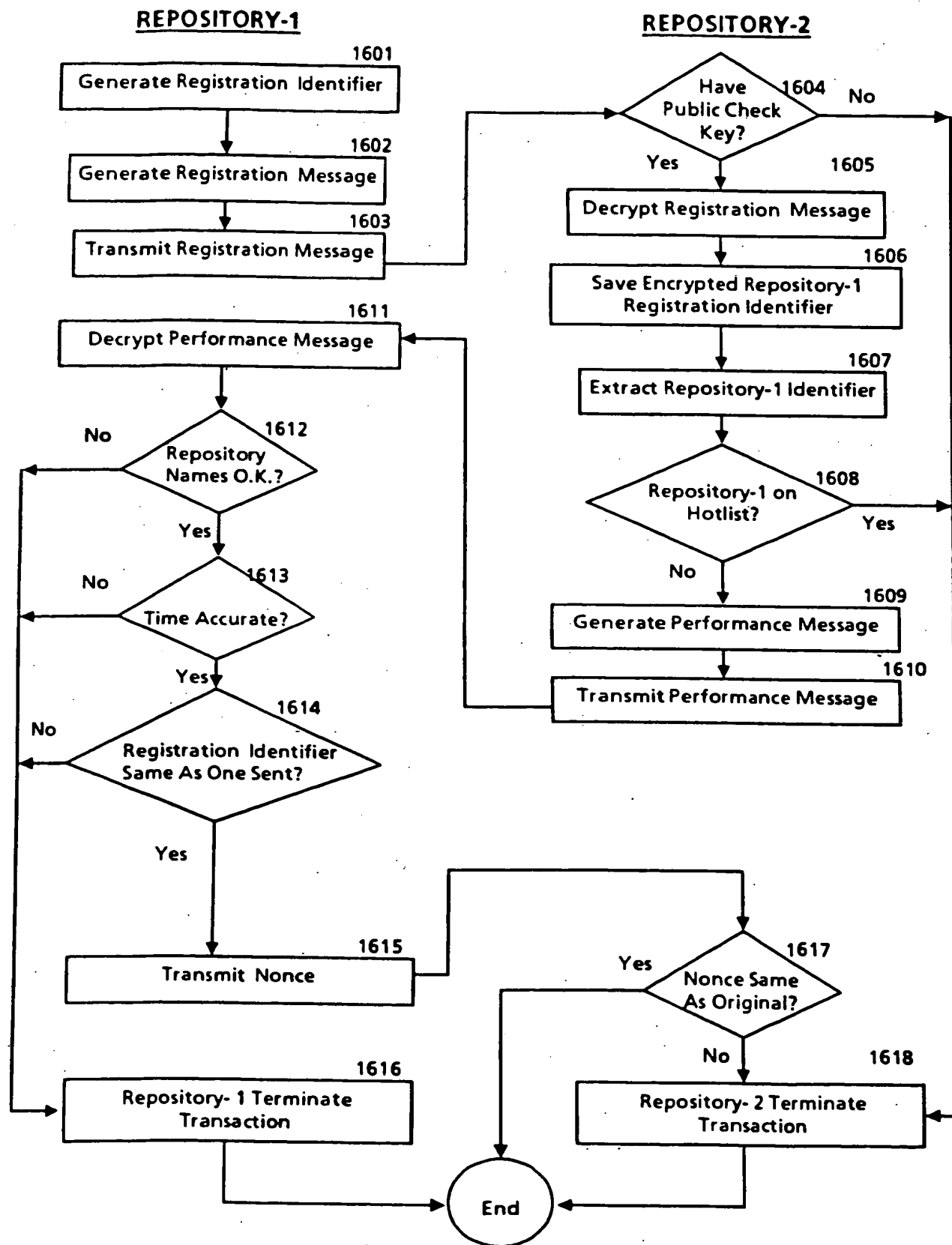


Figure 17

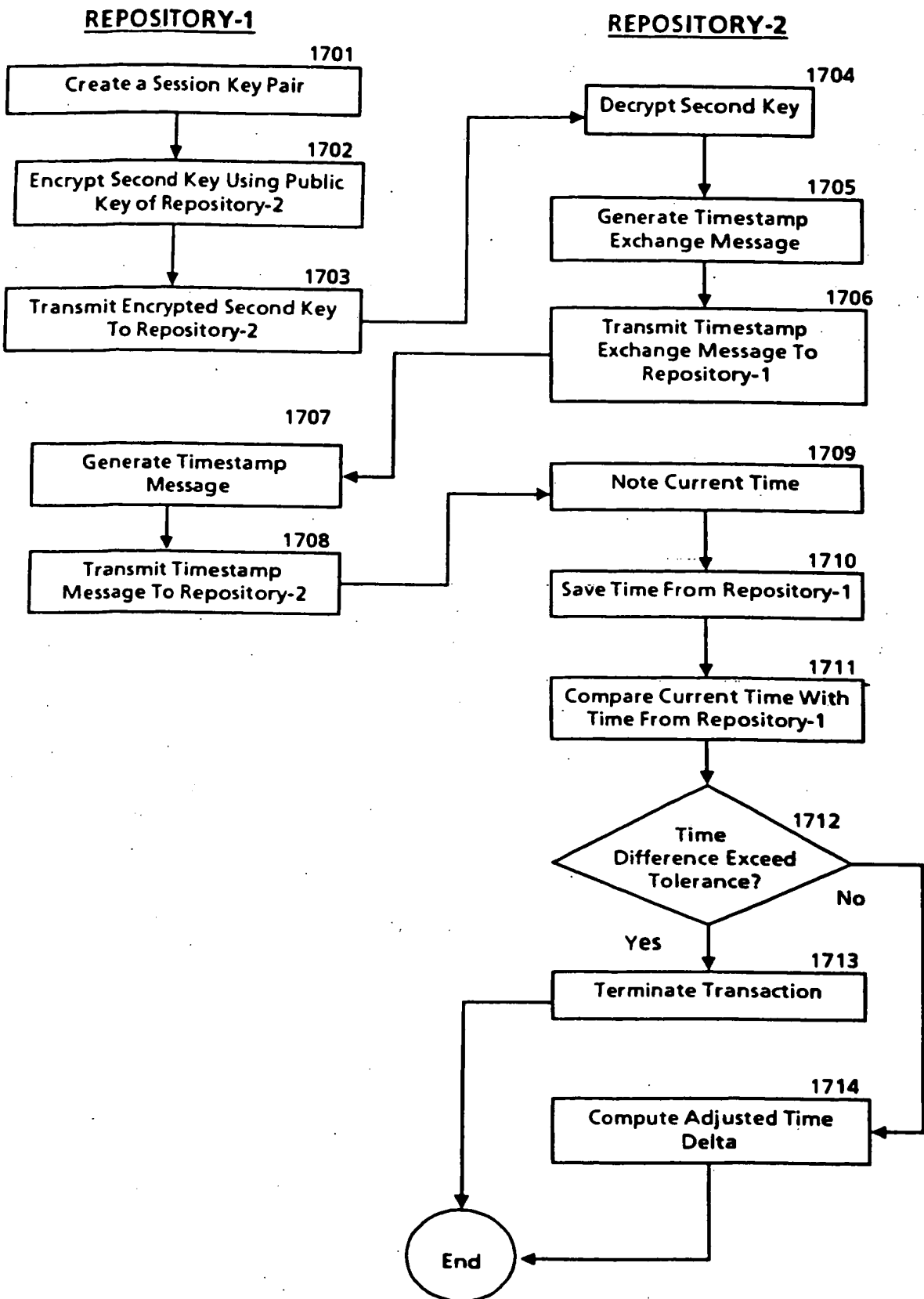


Figure 18

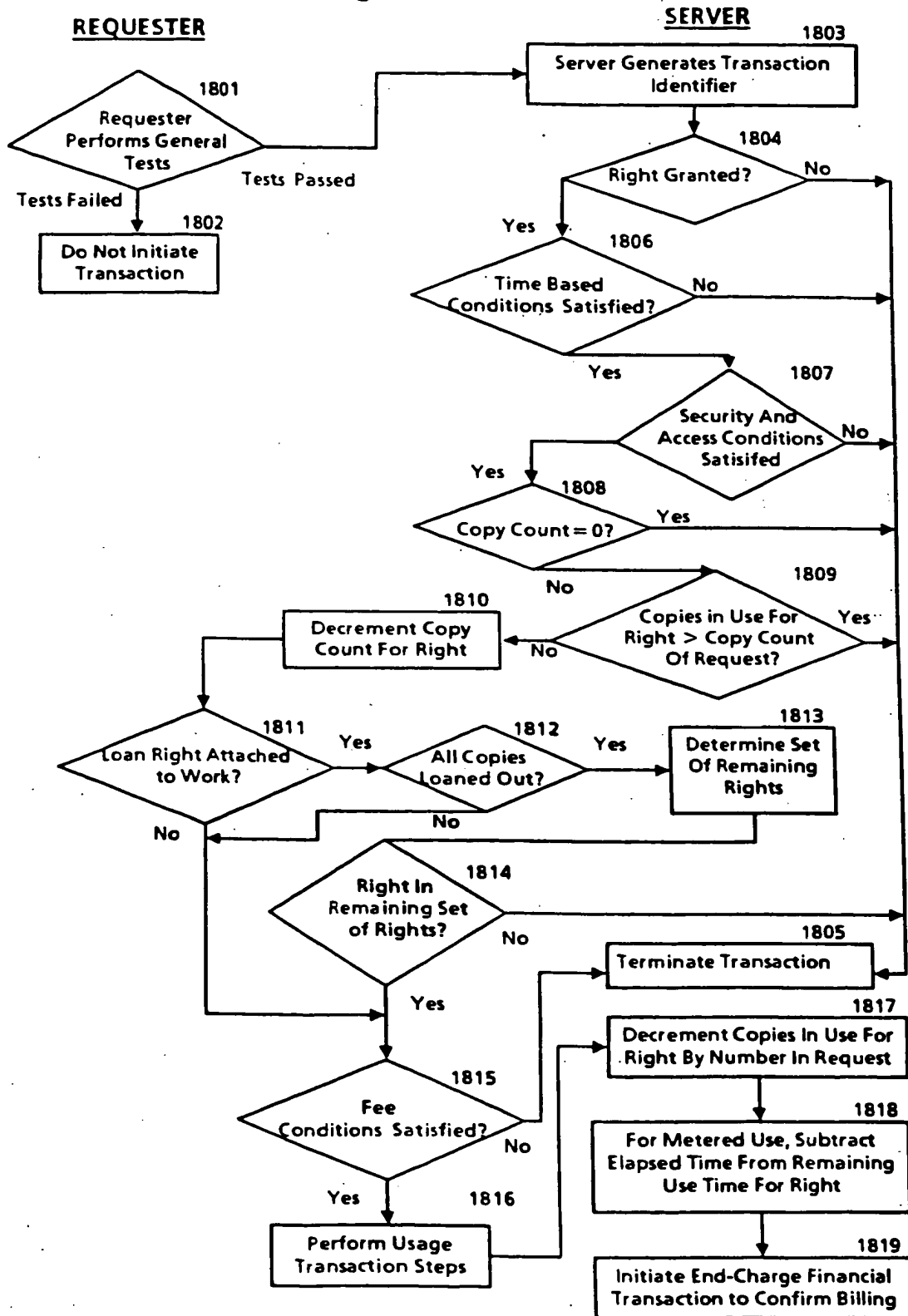
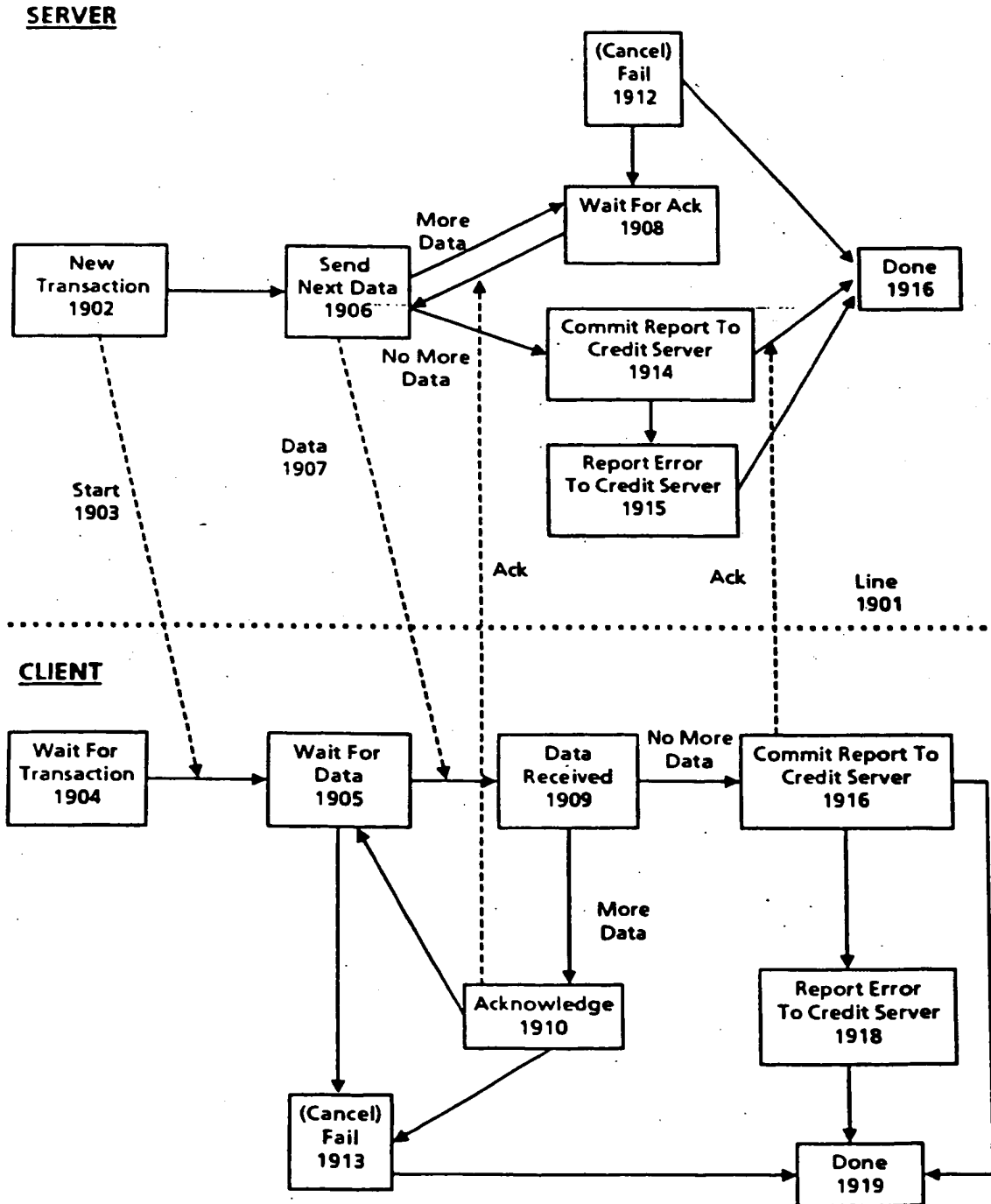


Figure 19



Please type a plus sign (+) inside this box → [+]

PTO/SB/01 (12-97)
Approved for use through 9/30/00. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OBM control number.

**DECLARATION FOR UTILITY OR
DESIGN PATENT APPLICATION
(37 CFR 1.63)**

Attorney Docket Number	111325-88
First Named Inventor	Mark J. Stefik et al.
COMPLETE IF KNOWN	
Application Number	10/015,951
Filing Date	December 17, 2001
Group Art Unit	2161
Examiner Name	Not yet assigned

☐ Declaration Submitted With Initial Filing ☒ Declaration Submitted after Initial Filing (surcharge (37 CFR 1.16(d)) required)

As a below named inventor, I hereby declare that:

My residence, post office address, and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

Digital Works Having Usage Rights And Method For Creating The Same
(Title of the Invention)

the specification of which

☐ is attached hereto
OR

☒ was filed on (MM/DD/YYYY) 12/17/2001 As United States Application Number or PCT International Application Number 10/015,951
And was amended on (MM/DD/YYYY) _____ (If applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment specifically referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR 1.56.

I hereby claim foreign priority benefits under 35 U.S.C. 119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or of any PCT international application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application Number(s)	Country	Foreign Filing Date (MM/DD/YYYY)	Priority Not Claimed	Certified Copy Attached	
				YES	No
			☐ ☐ ☐ ☐	☐ ☐ ☐ ☐	☐ ☐ ☐ ☐

☐ Additional foreign application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto:

I hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional application(s) listed below.

Application Number(s)	Filing Date (MM/DD/YYYY)	☐ Additional provisional application Numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.

[Page 1 of 3]

Burden Hour Statement: This form is estimated to take 0.4 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissions for Patents, Washington, DC 20231.

Please type a plus sign (+) inside this box → [+]

PTO/SB/01 (12-97)
Approved for use through 9/30/00. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OBM control number.

DECLARATION – Utility or Design Patent Application

I hereby claim the benefit under 35 U.S.C. 120 of any United States application(s), or 365© of any PT international application designating the United States of America, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of 35 U.S.C. 112, I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application.

U.S. Parent Application or PCT Parent Number	Parent Filing Date (MM/DD/YYYY)	Parent Patent Number (if applicable)
This Application is a Continuation of USSN 09/778,001 which is a Divisional Application of USSN 08/967,084 which is a Continuation of USSN 08/344,760	02/07/2001 11/10/1997 11/23/1994	 6,236,971 now abandoned

[] Additional U.S. or PCT international application are listed on a supplemental priority date sheet PTO/SB/02B attached hereto.

As a named inventor, I hereby appoint the following registered practitioner(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith: [X] Customer Number 22204

OR

[X] Registered practitioner(s) name/registration number listed below.

Name	Registration Number	Name	Registration Number
Stuart J. Friedman	24,312	Eric J. Robinson	38,285
Charles M. Leedom, Jr.	26,477	Daniel S. Song	43,143
David S. Safran	27,997	Marc S. Kaufman	35,212
Thomas W. Cole	28,290	Corinne R. Gorski	34,339
Donald R. Studebaker	32,815	Jason H. Vick	45,285
Jeffrey L. Costellia	35,483	Luan C. Do	38,434
Tim L. Brackett, Jr.	36,092		

Direct all correspondence to: [X] Customer Number 22204

Name: Marc S. Kaufman

Firm: NIXON PEABODY LLP

Address: 8180 Greensboro Drive, Suite 800

City: McLean

State: VA

ZIP: 22102

Country: United States

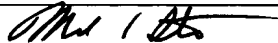
Telephone: (703) 790-9110

FAX: (703) 883-0370

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Name of Sole or First Inventor:

[] A petition has been filed for this unsigned inventor.

Given Name (first and middle [if any])		Family Name or Surname	
Mark J.		STEFIK	
Inventor's Signature: 		Date: 5 Feb '02	
Mailing Address (Street or P.O. Box): 55 Big Tree Way 10 Portola Green Circle			
City: Woodside	Portola Valley	State: California	ZIP: 94062 94025 Country: US
Residence: City:	State:	Country:	
Citizenship: USA			
[] Additional inventors are being named on the _____ Supplemental Additional Inventor(s) sheet(s) PTO/SB/02A attached hereto.			

Please type a plus sign (+) inside this box → [+]

PTO/SB/02A (10-00)
Approved for use through 10/31/2002. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OBM control number.

DECLARATION	ADDITIONAL INVENTOR(S) Supplemental Sheet Page ____ of ____

Name of Additional Joint Inventor, if any:		☐ A petition has been filed for this unsigned inventor	
Given Name Peter L.T.		Family Name Or Surname PIROLI	
Inventor's Signature <i>Peter L.T. Piroli</i>		Date <i>2/5/02</i>	
Mailing Address (Street or P.O. Box): <i>44 Wildwood Place 2958 Sloat Blvd</i>			
City: <i>El Cerrito San Francisco</i>		State: <i>California</i>	ZIP: <i>94530</i> Country: <i>US</i>
Residence: City:		State:	Country:
Citizenship: <i>Canadian</i>			
Name of Additional Joint Inventor, if any:		☐ A petition has been filed for this unsigned inventor	
Given Name		Family Name Or Surname	
Inventor's Signature		Date	
Mailing Address (Street or P.O. Box):			
City:		State:	ZIP: Country:
Residence: City:		State:	Country:
Citizenship:			
Name of Additional Joint Inventor, if any:		☐ A petition has been filed for this unsigned inventor	
Given Name		Family Name Or Surname	
Inventor's Signature		Date	
Mailing Address (Street or P.O. Box):			
City:		State:	ZIP: Country:
Residence: City:		State:	Country:
Citizenship:			

Burden Hour Statement: This form is estimated to take 21 minutes to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, Washington, DC 20231.

[Page 3 of 3]

APPLICATION DATA SHEET

Electronic Version 0.0.11

Stylesheet Version: 1.0

Publication Filing Type:

Application Type:

Title of Invention:

Suggested Representative Figure:

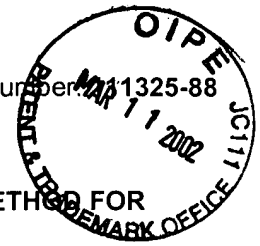
new-utility

utility

DIGITAL WORKS HAVING USAGE RIGHTS AND METHOD FOR
CREATING THE SAME

Fig.1

Attorney Docket Number: 11325-88

**Legal Representative:**

Attorney or Agent:

Marc S. KAUFMAN

Registration Number:

35212

Customer Number Correspondence Address:

22204

22204

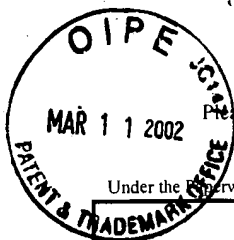
Continuity Data:

This application is a continuation of 09/778,001 A1 2001-02-07 US Pending
which is a division of 08/967,084 A1 1997-11-10 US 6,236,971 2001-05-22 A1 US
which is a continuation of 08/344,760 A1 1994-11-23 US Abandoned

Assignee (Publish): CONTENTGUARD HOLDINGS, INC.**INVENTOR(s):**

Primary Citizenship: US
Given Name: Mark
Middle Name: J.
Family Name: STEFIK
Residence City: Portola Valley
Residence State: CA
Residence Country: US
Address: 10 Portola Green Circle
Portola Valley California, 94028 US

Primary Citizenship: Canada
Given Name: Peter
Middle Name: L.T.
Family Name: PIROLLI
Residence City: San Francisco
Residence State: CA
Residence Country: US
Address: 2958 Sloat Blvd.
San Francisco California, 94116 US



Please type a plus sign (+) inside this box → [+]

PTO/SB/21 (08-00)

Approved for use through 10/31/2002. OMB 0651-0031

U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

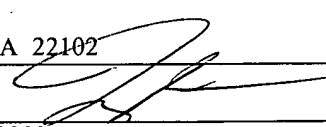
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM (to be used for all correspondence after initial filing)	Application Number	10/015,951	
	Filing Date	December 17, 2001	
	First Named Inventor	Mark J. Stefik et al.	
	Group Art Unit	2161	
	Examiner Name	Not yet assigned	
Total Number of Pages in This Submission		Attorney Docket Number	111325-88

ENCLOSURES (check all that apply)

☒ Fee Transmittal Form ☒ Fee Attached ☐ Amendment / Reply ☐ After Final ☐ Affidavits/declaration(s) ☐ Extension of Time Request ☐ Express Abandonment Request ☐ Information Disclosure Statement ☐ Certified Copy of Priority Document(s) ☒ Response to Missing Parts/Incomplete Application ☒ Response to Missing Parts under 37 CFR 1.52 or 1.53	☒ Assignment Papers (for an Application) ☒ Drawing(s) ☒ Declaration and Power of Attorney ☐ Licensing-related Papers ☐ Petition ☐ Petition to Convert to a Provisional Application ☐ Power of Attorney, Revocation Change of Correspondence Address ☐ Terminal Disclaimer ☐ Request for Refund ☐ CD, Number of CD(s) _____	☐ After Allowance Communication to Group ☒ Other: Application Data Sheet
Remarks ☒ The Commissioner is hereby authorized to charge any additional fees required or credit any overpayments to Deposit Account No. 19-2380 for the above identified docket number.		

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	<u>Marc S. Kaufman</u> Nixon Peabody LLP 8180 Greensboro Drive Suite 800 McLean, VA 22102
Signature	
Date	March 11, 2002

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, Washington, DC 20231 on this date: 	
Type or printed name	
Signature	Date _____

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, Washington, DC 20231.

**FEE TRANSMITTAL
FOR FY 2002**

Patent fees are subject to annual revision.

Application Number		10/015,951
Filing Date		December 17, 2001
First Named Inventor		Mark J. Stefik et al.
Examiner Name		Not yet assigned
Group Art Unit		2161
Attorney Docket No.		111325-88

TOTAL AMOUNT OF PAYMENT **\$1,246.00****METHOD OF PAYMENT**

- 1.
- ☒
- The Commissioner is hereby authorized to charge indicated fees and credit any overpayments to:

Deposit
Account
Number

19-2380

Deposit
Account
Name

Nixon Peabody LLP

- ☒
- Charge Any Additional Fee Required
-
- Under 37 CFR 1.16 and 1.17

☐ Applicant claims small entity status.
See 37 CFR 1.27

- 2.
- ☒
- Payment Enclosed:

☒ Check ☐ Credit Card ☐ Money
Order ☐ Other**FEE CALCULATION****1. BASIC FILING FEE**

Large Entity Fee Code	Large Entity Fee (\$)	Small Entity Fee Code	Small Entity Fee (\$)	Fee Description	Fee Paid
101	740	201	370	Utility filing fee	740.00
106	330	206	165	Design filing fee	
107	510	207	255	Plant filing fee	
108	740	208	370	Reissue filing fee	
114	160	214	80	Provisional filing fee	

SUBTOTAL (1) **\$740.00****2. EXTRA CLAIM FEES**

Total Claims 34 -20** = 14 X 18.00 = 252.00

*Independent Claims 4 -3** = 1 X 84.00 = 84.00

Multiple Dependent =

Large Entity Fee Code	Large Entity Fee (\$)	Small Entity Fee Code	Small Entity Fee (\$)	Fee Description	Fee Paid
103	18	203	9	Claims in excess of 20	
102	84	202	42	Independent claims in excess of 3	
104	280	204	140	Multiple dependent claim, if not paid	
109	84	209	42	** Reissue independent claims over original patent	
110	18	210	9	** Reissue claims in excess of 20 and over original patent	

SUBTOTAL (2) **\$336.00**

**or number previously paid, if greater; For Reissues, see above

FEE CALCULATION (continued)**3. ADDITIONAL FEES**

Fee Code	Large Entity Fee (\$)	Small Entity Fee Code	Small Entity Fee (\$)	Fee Description	Fee Paid
105	130	205	65	Surcharge - late filing fee or oath	130.00
127	50	227	25	Surcharge - late provisional filing fee or cover sheet	
139	130	139	130	Non-English transaction	
147	2,520	147	2,520	For filing a request for <i>ex parte</i> reexamination	
112	920*	112	920*	Requesting publication of SIR prior to Examiner action	
113	1,840*	113	1,840*	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for reply within first month	
116	400	216	200	Extension for reply within second month	
117	920	217	460	Extension for reply within third month	
118	1,440	218	720	Extension for reply within fourth month	
128	1,960	228	980	Extension for reply within fifth month	
119	320	219	160	Notice of Appeal	
120	320	220	160	Filing a brief in support of an appeal	
121	280	221	140	Request for oral hearing	
138	1,510	138	1,510	Petition to institute a public use proceeding	
140	110	240	55	Petition to revive - unavoidable	
141	1,280	241	640	Petition to revive - unintentional	
142	1,280	242	640	Utility issue fee (or reissue)	
143	460	243	230	Design issue fee	
144	620	244	310	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	50	123	50	Processing fee under 37 CR 1.17(q)	
126	180	126	180	Submission of Information Disclosure Stmt	
581	40	581	40	Recording each patent assignment per property (times number of properties)	40.00
146	740	246	370	Filing a submission after final rejection (37 CFR § 1.129(a))	
149	740	249	370	For each additional invention to be examined (37 CFR § 1.29(b))	
179	740	279	370	Request for Continued Examination (RCE)	
169	900	169	900	Request for expedited examination of a design application	

Other fee (specify) _____

* Reduced by Basic Filing Fee Paid

SUBTOTAL (3) **\$170.00****CERTIFICATE OF MAILING**

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Commissioner for Patents, Washington, DC 20231, on _____

Name: _____

SUBMITTED BY

Name (Print/Type)

Marc S. Kaufman

Signature

Registration No.
(Attorney/Agent)

35,212

Complete (if applicable)

Telephone

703-790-9110

Date

March 11, 2002



Docket No. 111325-88

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

re Application of)
Mark J. Stefik et al.) Group Art Unit: 2161
Serial No.: 10/015,951) Examiner: Not yet assigned
Filed: December 17, 2001)
For: Digital Works Having Usage Rights And)
Method For Creating The Same)

SUBMISSION OF FORMAL DRAWINGS

Commissioner for Patents
Washington, D. C. 20231

March 11, 2002

Sir:

Submitted herewith are thirteen (13) sheets of formal drawings (Figs. 1-19) in connection with the above-identified application.

Respectfully submitted,

NIXON PEABODY LLP

Marc S. Kaufman
Registration No. 35,212

MSK/kbd

NIXON PEABODY LLP
8180 Greensboro Drive, Suite 800
McLean, Virginia 22102
Telephone (703) 790-9110



UNITED STATES PATENT AND TRADEMARK OFFICE



COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

APPLICATION NUMBER	FILING/RECEIPT DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NUMBER
10/015,951	12/17/2001	Mark J. Stefik	111325-88

CONFIRMATION NO. 5128

FORMALITIES LETTER



OC000000007296565

22204
NIXON PEABODY, LLP
8180 GREENSBORO DRIVE
SUITE 800
MCLEAN, VA 22102

Date Mailed: 01/11/2002

NOTICE TO FILE MISSING PARTS OF NONPROVISIONAL APPLICATION

03/15/2002 BABRAWA1 00000053 10015951

FILED UNDER 37 CFR 1.53(b)

01 FC:101	740.00 OP
02 FC:105	130.00 OP
03 FC:103	252.00 OP
04 FC:102	84.00 OP

Filing Date Granted

An application number and filing date have been accorded to this application. The item(s) indicated below, however, are missing. Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items and pay any fees required below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

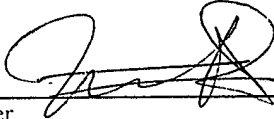
- The statutory basic filing fee is missing.
Applicant must submit \$ 740 to complete the basic filing fee for a non-small entity. If appropriate, applicant may make a written assertion of entitlement to small entity status and pay the small entity filing fee (37 CFR 1.27).
- Total additional claim fee(s) for this application is \$336.
 - \$252 for 14 total claims over 20.
 - \$84 for 1 independent claims over 3 .
- The oath or declaration is missing.
A properly signed oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date, is required.
- To avoid abandonment, a late filing fee or oath or declaration surcharge as set forth in 37 CFR 1.16(l) of \$130 for a non-small entity, must be submitted with the missing items identified in this letter.
- The balance due by applicant is \$ 1206.**

The application is informal since it does not comply with the regulations for the reason(s) indicated below.

The required item(s) identified below must be timely submitted to avoid abandonment:

- Substitute drawings in compliance with 37 CFR 1.84 because:
 - drawing sheets do not have the appropriate margin(s) (see 37 CFR 1.84(g)). Each sheet must include a top margin of at least 2.5 cm. (1 inch), a left side margin of at least 2.5 cm. (1 inch), a right side margin of at least 1.5 cm. (5/8 inch), and a bottom margin of at least 1.0 cm. (3/8 inch);

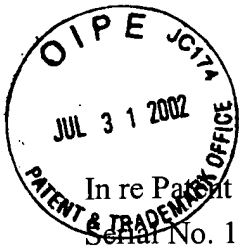
A copy of this notice MUST be returned with the reply.

A handwritten signature in black ink, appearing to be "J. R.", is written over a horizontal line.

Customer Service Center

Initial Patent Examination Division (703) 308-1202

PART 2 - COPY TO BE RETURNED WITH RESPONSE



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of: Mark J. Stefik et al.

Serial No. 10/015,951

Filed: 12/17/2001

For: DIGITAL WORKS HAVING USAGE RIGHT AND
METHOD FOR CREATING THE SAME

) Examiner: Unassigned

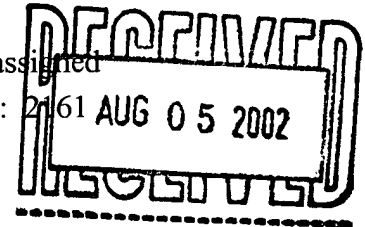
) Group Art Unit: 2161 AUG 05 2002

)

)

)

)



RECEIVED

AUG 02 2002

GROUP 3600

INFORMATION DISCLOSURE STATEMENT

Commissioner of Patents and Trademarks
Washington, D.C. 20231

Sir:

In accordance with the duty of disclosure as set forth in 37 C.F.R. §1.56, Applicants hereby submit the following information in conformance with 37 C.F.R. §§ 1.97 and 1.98. Pursuant to 37 C.F.R. § 1.98, a copy of each of the documents cited is enclosed.

The documents are being submitted within three (3) months of the filing of this application or entry into the national stage of this application, or before the first Office Action on the merits, whichever is later, therefore no fee or certification is required under 37 C.F.R. § 1.97(b).

It is requested that the accompanying information disclosure statement be considered and made of record in the above-captioned application. To assist the Examiner, the documents are listed on the attached form PTO-1449. It is respectfully requested that an Examiner initialed copy of this form be returned to the undersigned.

The Commissioner is hereby authorized to charge any fees connected with this filing which may be required now, or credit any overpayment to Deposit Account No. 19-2380.

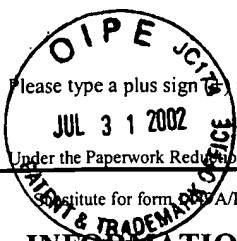
Respectfully submitted,

By: Marc S. Kaufman, Esq.

Registration No. 35,212

NIXON PEABODY LLP
8180 Greensboro Drive, Suite 800
McLean, Virginia 22102
Telephone: (703) 770-9300

NVA234480.1



Please type a plus sign (+) inside this box → [+]

PTO/SB/08A (08-00)

Approved for use through 10/31/2002. OMB 0651-0031

U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Complete if Known	
				Application Number	10/015,951
				Filing Date	12/17/2001
				First Named Inventor	Mark J. Stefik et al
				Group Art Unit	2161
				Examiner Name	Unassigned
Sheet	1	of	1	Attorney Docket Number	111325/000088

RECEIVED
AUG 02 2002
GROUP 3600

U.S. PATENT DOCUMENTS						
Examiner Initials [*]	Cite No. ¹	U.S. Patent Document		Name of Patentee or Applicant of Cited Document	Date of Publication of Cited Document MM-DD-YYYY	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Number	Kind Code ² (if known)			
		6,253,193		Ginter et al	06/26/2001	
		4,713,753		Beobert et al	12/15/1987	
		5,052,040		Preston et al	09/24/1991	
		5,301,231		Abraham et al	04/05/1994	
		5,502,766		Boebert et al	03/26/1996	
		5,649,013		Stuckey et al	07/15/1997	
		5,737,416		Cooper et al	04/07/1998	
		5,737,413		Akiyama et al	04/07/1998	
		5,757,907		Cooper et al	05/26/1998	
		6,301,660		Benson	10/09/2001	
		6,327,652		England et al	12/04/2001	
		6,330,670		England et al	12/11/2001	

FOREIGN PATENT DOCUMENTS								
Examiner Initials ⁵	Cite No. ¹	Foreign Patent Document			Name of Patentee or Applicant of Cited Document	Date of Publication of Cited Document MM-DD-YYYY	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T ⁶
		Office ³	Number ⁴	Kind Code ² (if known)				

OTHER PRIOR ART - NON PATENT LITERATURE DOCUMENTS			
Examiner Initials ⁵	Cite No. ¹	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T ²

Examiner Signature	Date Considered

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

¹ Unique citation designation number. ² See attached Kinds of U.S. Patent Documents. ³ Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). ⁴ For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. ⁵ Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. ⁶ Applicant is to place a check mark here if English language Translation is attached.

¹ Unique citation designation number. ² Applicant is to place a check mark here if English language Translation is attached.

Burden Hour Statement: This form is estimated to take 2.0 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, Washington, DC 20231.

NVA234468.1



Application No. 10/015,951
Docket No. 111325/000088

2167
8-16-02

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of: Mark J. Stefik et al.) Examiner: Unassigned
Serial No. 10/015,951) Group Art Unit: 2161
Filed: 12/17/2001)
For: DIGITAL WORKS HAVING USAGE RIGHT AND)
METHOD FOR CREATING THE SAME)

RECEIVED
AUG 13 2002
GROUP 3600

INFORMATION DISCLOSURE STATEMENT

Commissioner of Patents
Washington, D.C. 20231

RECEIVED
AUG 10 2002

Sir:

In accordance with the duty of disclosure as set forth in 37 C.F.R. §1.56, Applicants hereby submit the following information in conformance with 37 C.F.R. §§ 1.97 and 1.98. Pursuant to 37 C.F.R. § 1.98, a copy of each of the documents cited is enclosed.

The documents are being submitted within three (3) months of the filing of this application or entry into the national stage of this application, or before the first Office Action on the merits, whichever is later, therefore no fee or certification is required under 37 C.F.R. § 1.97(b).

The submitted documents are patents issued to a company known to be developing related technology.

It is requested that the accompanying information disclosure statement be considered and made of record in the above-captioned application. To assist the Examiner, the documents are listed on the attached form PTO-1449. It is respectfully requested that an Examiner initialed copy of this form be returned to the undersigned.

The Commissioner is hereby authorized to charge any fees connected with this filing which may be required now, or credit any overpayment to Deposit Account No. 19-2380.

Respectfully submitted,

By: Marc S. Kaufman, Esq.
Registration No. 35,212

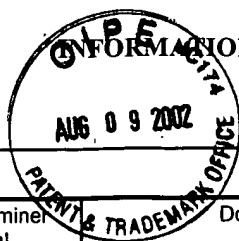
NIXON PEABODY LLP
8180 Greensboro Drive, Suite 800
McLean, Virginia 22102
Telephone: (703) 770-9300

NVA235901.1

Form PTO-1449
(Rev. 8-83)U.S. Department of Commerce
Patent and Trademark Office

Atty Docket 111325-88

Serial No. 10/015,951



INFORMATION DISCLOSURE STATEMENT

Applicants: Stefik et al.

Filing Date: December 17, 2001

Group Art Unit:

U.S. PATENT DOCUMENTS

Examiner Initial	Document Number	Date	Name	Class	Subclass	Filing Date (if appropriate)
	4,827,508	05/02/1989	Shear			
	4,977,594	12/11/1990	Shear			
	5,050,213	09/17/1991	Shear			
	5,410,598	04/25/1995	Shear			
	5,892,900	04/06/1999	Ginter et al.			
	5,910,987	06/08/1999	Ginter et al.			
	5,915,019	06/22/1999	Ginter et al.			
	5,917,912	06/29/1999	Ginter et al.			
	5,920,861	07/06/1999	Hall et al.			
	5,940,504	08/17/1999	Griswold			
	5,943,422	08/24/1999	Van Wie et al.			
	5,949,876	09/07/1999	Ginter et al.			
	5,982,891	11/09/1999	Ginter et al.			
	5,999,949	12/07/1999	Crandall			
	6,112,181	08/29/2000	Shear et al.			
	6,138,119	10/24/2000	Hall et al.			
	6,157,721	12/05/2000	Shear et al.			
	6,185,683	02/06/2001	Ginter et al.			
	6,237,786	05/29/2001	Ginter et al.			
	6,240,185	05/29/2001	Van Wie et al.			
	6,253,193	06/26/2001	Ginter et al.			
	6,292,569	09/18/2001	Shear et al.			
	6,363,488	05/26/2002	Ginter et al.			
	6,389,402	05/14/2002	Ginter et al.			

RECEIVED
AUG 13 2002
GROUP 3600

Examiner

Date Considered

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

08/09/2002

Petitioner Apple Inc. - Exhibit 1012, p. 259



Application No. 10/015,951
Docket No. 111325/000088

76
NY
9-23-02

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of: Mark J. Stefik et al.) Examiner: Unassigned
Serial No. 10/015,951) Group Art Unit: 2161
Filed: 12/17/2001)
For: DIGITAL WORKS HAVING USAGE RIGHT AND)
METHOD FOR CREATING THE SAME)
)

RECEIVED

SEP 04 2002

GROUP 3600

INFORMATION DISCLOSURE STATEMENT

Commissioner of Patents
Washington, D.C. 20231

Sir:

In accordance with the duty of disclosure as set forth in 37 C.F.R. §1.56, Applicants hereby submit the following information in conformance with 37 C.F.R. §§ 1.97 and 1.98. Pursuant to 37 C.F.R. § 1.98, a copy of each of the documents cited is enclosed.

The documents are being submitted within three (3) months of the filing of this application or entry into the national stage of this application, or before the first Office Action on the merits, whichever is later, therefore no fee or certification is required under 37 C.F.R. § 1.97(b).

It is requested that the accompanying information disclosure statement be considered and made of record in the above-captioned application. To assist the Examiner, the documents are listed on the attached form PTO-1449. It is respectfully requested that an Examiner initialed copy of this form be returned to the undersigned.

The Commissioner is hereby authorized to charge any fees connected with this filing which may be required now, or credit any overpayment to Deposit Account No. 19-2380.

Respectfully submitted,

By: Marc S. Kaufman
Registration No. 35,212

NIXON PEABODY LLP
8180 Greensboro Drive, Suite 800
McLean, Virginia 22102
Telephone: (703) 770-9300

Form PTO-1449
(Rev. 8-83)U.S. Department of Commerce
Patent and Trademark Office

Atty Docket 111325-88

Serial No. 10/015,951

Applicants: Stefik et al.

Filing Date: December 17, 2001

Group Art Unit:



INFORMATION DISCLOSURE STATEMENT

U.S. PATENT DOCUMENTS

Examiner Initial	Document Number	Date	Name	Class	Subclass	Filing Date (if appropriate)
	3,263,158	07/01/1966	Janis			
	3,609,697	09/28/1971	Blevins et al.			
	3,790,700	02/05/1974	Callais et al.			
	3,798,605	03/19/1974	Feistel			
	4,159,468	06/26/1979	Barnes et al.			
	4,220,991	09/02/1980	Hamano et al.			
	4,278,837	07/14/1981	Best			
	4,323,921	04/06/1982	Guillou			
	4,442,486	04/10/1984	Mayer			
	4,529,870	07/16/1985	Chaum			
	4,558,176	12/10/1985	Arnold et al.			
	4,593,376	06/03/1986	Volk			
	4,614,861	09/30/1986	Pavlov et al.			
	4,644,493	02/17/1987	Chandra et al.			
	4,658,093	04/14/1987	Hellman			
	4,817,140	03/28/1989	Chandra et al.			
	4,868,376	09/19/1989	Lessin et al.			
	4,891,838	01/02/1990	Faber			
	4,924,378	05/08/1990	Hershey et al.			
	4,932,054	06/05/1990	Chou et al.			
	4,937,863	06/26/1990	Robert et al.			
	4,949,187	08/14/1990	Cohen			
	4,953,209	08/28/1990	Ryder, Sr. et al.			
	4,961,142	10/02/1990	Elliott et al.			
	4,975,647	12/04/1990	Downer et al.			
	4,999,806	03/12/1991	Chernow et al.			
	5,010,571	04/23/1991	Katznelson			
	5,014,234	05/07/1991	Edwards, Jr.			
	5,023,907	06/11/1991	Johnson et al.			
	5,047,928	09/10/1991	Wiedemer			
	5,052,040	09/24/1991	Preston et al.			

RECEIVED

SEP 04 2002

GROUP 3600

Examiner

Date Considered

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

08/27/2002

Petitioner Apple Inc. - Exhibit 1012, p. 261

Form PTO-1449
(Rev. 8-83)U.S. Department of Commerce
Patent and Trademark Office

Atty Docket 111325-88

Serial No. 10/015,951

INFORMATION DISCLOSURE STATEMENT

Applicants: Stefik et al.

Filing Date: December 17, 2001

Group Art Unit:

U.S. PATENT DOCUMENTS

Examiner Initial	Document Number	Date	Name	Class	Subclass	Filing Date (if appropriate)
	5,058,164	10/15/1991	Elmer et al.			
	5,103,476	04/07/1992	Waite et al.			
	5,113,519	05/12/1992	Johnson et al.			
	5,136,643	08/04/1992	Fischer			
	5,138,712	08/11/1992	Corbin			
	5,146,499	09/08/1992	Geffrotin			
	5,148,481	09/15/1992	Abraham et al.			
	5,159,182	10/27/1992	Eisele			
	5,183,404	02/02/1993	Aldous et al.			
	5,191,193	03/02/1993	Le Roux			
	5,204,897	04/20/1993	Wyman			
	5,222,134	06/22/1993	Waite et al.			
	5,235,642	08/10/1993	Wobber et al.			
	5,247,575	09/21/1993	Sprague et al.			
	5,255,106	10/19/1993	Castro			
	5,260,999	11/09/1993	Wyman			
	5,263,157	11/16/1993	Janis			
	5,263,158	11/16/1993	Janis			
	5,276,444	01/04/1994	McNair			
	5,276,735	01/04/1994	Boebert et al.			
	5,291,596	03/01/1994	Mita			
	5,311,591	05/10/1994	Fischer			
	5,319,705	06/07/1994	Halter et al.			
	5,337,357	08/09/1994	Chou et al.			
	5,339,091	08/16/1994	Yamazaki et al.			
	5,341,429	08/23/1994	Stringer et al.			
	5,347,579	09/13/1994	Blandford			
	5,381,526	01/10/1995	Ellson			
	5,394,469	02/28/1995	Nagel et al.			
	5,412,717	05/02/1995	Fischer			
	5,428,606	06/27/1995	Moskowitz			

RECEIVED

SEP 04 2002

GROUP 3600

Examiner

Date Considered

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

08/27/2002

Petitioner Apple Inc. - Exhibit 1012, p. 262

Form PTO-1449
(Rev. 8-83)U.S. Department of Commerce
Patent and Trademark Office

Atty Docket 111325-88

Serial No. 10/015,951

 INFORMATION DISCLOSURE STATEMENT

Applicants: Stefik et al.

Filing Date: December 17, 2001

Group Art Unit:

U.S. PATENT DOCUMENTS

Examiner Initial	Document Number	Date	Name	Class	Subclass	Filing Date (if appropriate)
	5,432,849	07/11/1995	Johnson et al.			
	5,438,508	08/01/1995	Wyman			
	5,444,779	08/22/1995	Daniele			
	5,453,601	09/26/1995	Rosen			
	5,455,953	10/03/1995	Russell			
	5,457,746	10/10/1995	Dolphin			
	5,473,687	12/05/1995	Lipscomb et al.			
	5,473,692	12/05/1995	Davis			
	5,499,298	03/12/1996	Narasimhalu et al.			
	5,504,814	04/02/1996	Miyahara			
	5,504,818	04/02/1996	Okano			
	5,504,837	04/02/1996	Griffeth et al.			
	5,509,070	04/16/1996	Schull			
	5,530,235	06/25/1996	Stefik et al.			
	5,532,920	07/02/1996	Hartrick et al.			
	5,534,975	07/09/1996	Stefik et al.			
	5,539,735	07/23/1996	Moskowitz			
	5,563,946	10/08/1996	Cooper et al.			
	5,568,552	10/22/1996	Davis			
	5,621,797	04/15/1997	Rosen			
	5,629,980	05/13/1997	Stefik et al.			
	5,633,932	05/27/1997	Davis et al.			
	5,634,012	05/27/1997	Stefik et al.			
	5,638,443	06/10/1997	Stefik et al.			
	5,655,077	08/05/1997	Jones et al.			
	5,708,717	01/13/1998	Alasia			
	5,734,823	03/31/1998	Saigh et al.			
	5,734,891	03/31/1998	Saigh			
	5,745,569	04/28/1998	Moskowitz et al.			
	5,748,783	05/05/1998	Rhoads			
	5,761,686	06/02/1998	Bloomberg			

RECEIVED

SEP 04 2002

GROUP 3600

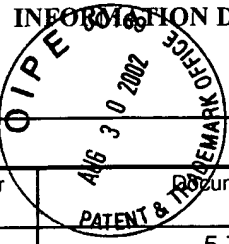
Examiner

Date Considered

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

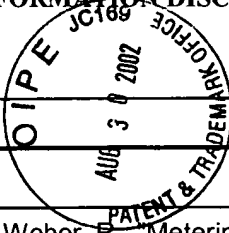
08/27/2002

Petitioner Apple Inc. - Exhibit 1012, p. 263

Form PTO-1449 (Rev. 8-83)		U.S. Department of Commerce Patent and Trademark Office		Atty Docket 111325-88		Serial No. 10/015,951	
INFORMATION DISCLOSURE STATEMENT 				Applicants: Stefik et al.			
				Filing Date: December 17, 2001		Group Art Unit:	
U.S. PATENT DOCUMENTS							
Examiner Initial	Document Number	Date	Name	Class	Subclass	Filing Date (if appropriate)	
	5,765,152	06/09/1998	Erickson				
	5,768,426	06/16/1998	Rhoads				
	5,825,892	10/20/1998	Braudaway et al.				
	6,047,067	04/04/2000	Rosen				
	6,115,471	09/05/2000	Oki et al.				
	6,233,684	05/15/2001	Stefik et al.				
	6,266,618	05/01/2001	Downs et al.				
	6,345,256	02/05/2002	Milsted et al.				
FOREIGN PATENT DOCUMENTS							
Examiner Initial	Document Number	Date	Country	Class	Subclass	Translation Yes No	
	0 084 441	07/27/1983	EP			Full Eng	
	0 180 460	05/07/1986	EP			Full Eng	
	0 332 707	09/01/1989	EP			Full Eng	
	0 651 554	05/03/1995	EP			Full Eng	
	0 668 695	08/23/1995	EP			Full Eng	
	0 725 376	08/07/1996	EP			Full Eng	
	2 136 175	09/12/1984	GB			Full Eng	
	2 236 604	04/10/1991	GB			Full Eng	
	WO 01/63528	08/30/2001	PCT			Full Eng	
	WO 92/20022	11/12/1992	PCT			Full Eng	
	WO 93/01550	01/21/1993	PCT			Full Eng	
	WO 99/49615	09/30/1999	PCT			Full Eng	
OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)							
Examiner Initial							
	"National Semiconductor and EPR Partner for Information Metering/Data Security Cards" March 4, 1994, Press Release from Electronic Publishing Resources, Inc.						
	Weber, R., "Digital Rights Management Technology" October 1995						
	Flasche, U. et al., "Decentralized Processing of Documents", pp. 119-131, 1986, Comput. & Graphics, Vol. 10, No. 2						
	Mori, R. et al., "Superdistribution: The Concept and the Architecture", pp. 1133-1146, 1990, The Transactions of						
Examiner				Date Considered			
*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.							

08/27/2002

Petitioner Apple Inc. - Exhibit 1012, p. 264

Form PTO-1449 (Rev. 8-83)		U.S. Department of Commerce Patent and Trademark Office		Atty Docket 111325-88	Serial No. 10/015,951
INFORMATION DISCLOSURE STATEMENT 				Applicants: Stefik et al.	
				Filing Date: December 17, 2001	Group Art Unit:
OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)					
Examiner Initial					
	Weber, R., "Metering Technologies for Digital Intellectual Property", pp. 1-29, Oct. 1994, A Report to the International Federation of Reproduction Rights Organizations				
	Clark, P.C. et al., "Bits: A Smartcard protected Operating System", pp. 66-70 and 94, November 1994, Communications of the ACM, Vol. 37, No. 11				
	Ross, P.E., "Data Guard", pp. 101, June 6, 1994, Forbes				
	Saigh, W.K., "Knowledge is Sacred", 1992, Video Pocket/Page Reader Systems, Ltd.				
	Kahn, R.E., "Deposit, Registration and Recordation in an Electronic Copyright Management System", pp. 1-19, August 1992, Corporation for National Research Initiatives, Virginia				
	Hilts, P. et al., "Books While U Wait", pp. 48-50, January 3, 1994, Publishers Weekly				
	Strattner, A, "Cash Register on a Chip may Revolutionize Software Pricing and Distribution; Wave Systems Corp.", pp. 62, April 1994, Computer Shopper, Vol. 14, No. 4, ISSN 0886-0556				
	O'Conner, M., "New Distribution Option for Electronic Publishers; iOpener Data Encryption and Metering System for CD-ROM use; Column", pp. 134, March 1994, CD-ROM Professional, Vol. 7, No. 2, ISSN: 1409-0833				
	Willett, S., "Metered PCs: Is Your System Watching You? Wave System beta tests new technology", pp. 84, May 2, 1994, InfoWorld				
	Linn, R., "Copyright and Information Services in the Context of the National Research and Education Network ", pp. 9-20, January 1994, IMA Intellectual Property Project Proceedings, Vol. 1, Issue 1				
	Perrit, Jr., H., "Permission Headers and Contract Law", pp. 27-48, January 1994, IMA Intellectual Property Project Proceedings, Vol. 1, Issue 1				
	Upthegrove, L., "Intellectual Property Header Descriptors: A Dynamic Approach", pp. 63-66, January 1994, IMA Intellectual Property Project Proceedings, Vol. 1, Issue 1				
	Sirbu, M., "Internet Billing Service Design and prototype Implementation", pp. 67-80, January 1994, IMA Intellectual Property Project Proceedings, Vol. 1, Issue 1				
	Simmel, S. et al., "Metering and Licensing of Resources: Kala's General Purpose Approach", pp. 81-110, January 1994, IMA Intellectual Property Project Proceedings, Vol. 1, Issue 1				
	Kahn, R., "Deposit, Registration and Recordation in an Electronic Copyright Management System", pp. 111-120, January 1994, IMA Intellectual Property Project Proceedings, Vol. 1, Issue 1				
	Tygar, J. et al., "Dyad: A System for Using Physically Secure Coprocessors", pp. 121-152, January 1994, IMA Intellectual Property Project Proceedings, Vol. 1, Issue 1				
	Griswold, G., "A Method for Protecting Copyright on Networks", pp. 169-178, January 1994, IMA Intellectual Property Project Proceedings, Vol. 1, Issue 1				
	Nelson, T., "A Publishing and Royalty Model for Networked Documents", pp. 257-259, January 1994, IMA Intellectual Property Project Proceedings, Vol. 1, Issue 1				
	Robinson, E., "Redefining Mobile Computing", pp. 238-240, 247-248 and 252, July 1993, PC Computing				
	Abadi, M. et al., "Authentication and Delegation with Smart-cards", 1990, Research Report DEC Systems Research Center				
	Mark Stefik, "Letting Loose the Light: Igniting Commerce in Electronic Publication", pp. 219-253, 1996, Internet Dreams: Archetypes, Myths, and Metaphors, IDSN 0-262-19373-6				
	Mark Stefik, "Letting Loose the Light: Igniting Commerce in Electronic Publication", pp. 2-35, February 8, 1995, Internet Dreams: Archetypes, Myths and Metaphors,				
Examiner	Date Considered				
*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.					

08/27/2002

Form PTO-1449
(Rev. 8-83)U.S. Department of Commerce
Patent and Trademark Office

Atty Docket 111325-88

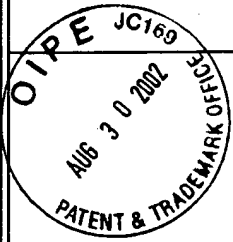
Serial No. 10/015,951

INFORMATION DISCLOSURE STATEMENT

Applicants: Stefik et al.

Filing Date: December 17, 2001

Group Art Unit:

**RECEIVED**
SEP 04 2002
GROUP 3600

Examiner

Date Considered

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

08/27/2002

Requested Patent: GB2136175A

Title: FILE ACCESS SECURITY METHOD AND MEANS ;

Abstracted Patent: US4588991 ;

Publication Date: 1986-05-13 ;

Inventor(s): ATALLA MARTIN M (US) ;

Applicant(s): ATALLA CORP (US) ;

Application Number: US19830472609 19830307 ;

Priority Number(s): US19830472609 19830307 ;

IPC Classification: ;

Equivalents:

DE3407642, FR2542471, JP1434850C, JP59169000, JP62042304B, ZA8401400 ;

ABSTRACT:

An improved file access security technique and associated apparatus accesses data which is stored in encrypted form under one encryption key and re-stores the data re-encrypted under another encryption key, and produces a record of each access and data re-encryption both as the control source of encryption keys for access and re-entry of encrypted data and as a secured audit record of users that had access to each file.

(12) UK Patent Application (19) GB (11) 2 136 175 A

(43) Application published 12 Sep 1984

(21) Application No 8405950

(22) Date of filing 7 Mar 1984

(30) Priority data

(31) 472609 (32) 7 Mar 1983 (33) US

(71) Applicant
Atalla Corporation, (USA—California),
2363 Bering Drive, San Jose, California 95131,
United States of America

(72) Inventor
Martin M. Atalla

(74) Agent and/or address for service
Peter A. Oliver, 8 Coombe Close, Frimley, Surrey,
GU16 5DZ

(51) INT CL³
H03K 13/24 G06F 15/00

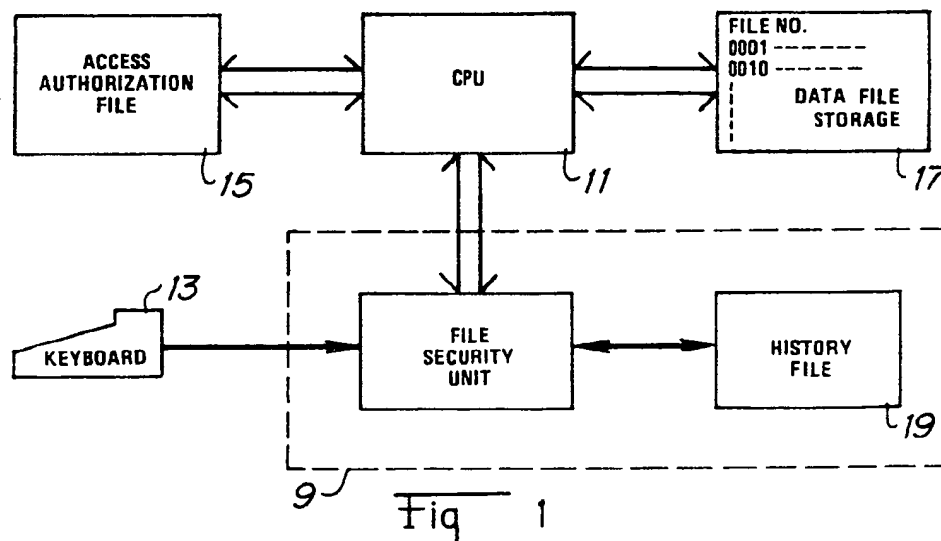
(52) Domestic classification
G4A AP

(56) Documents cited
None

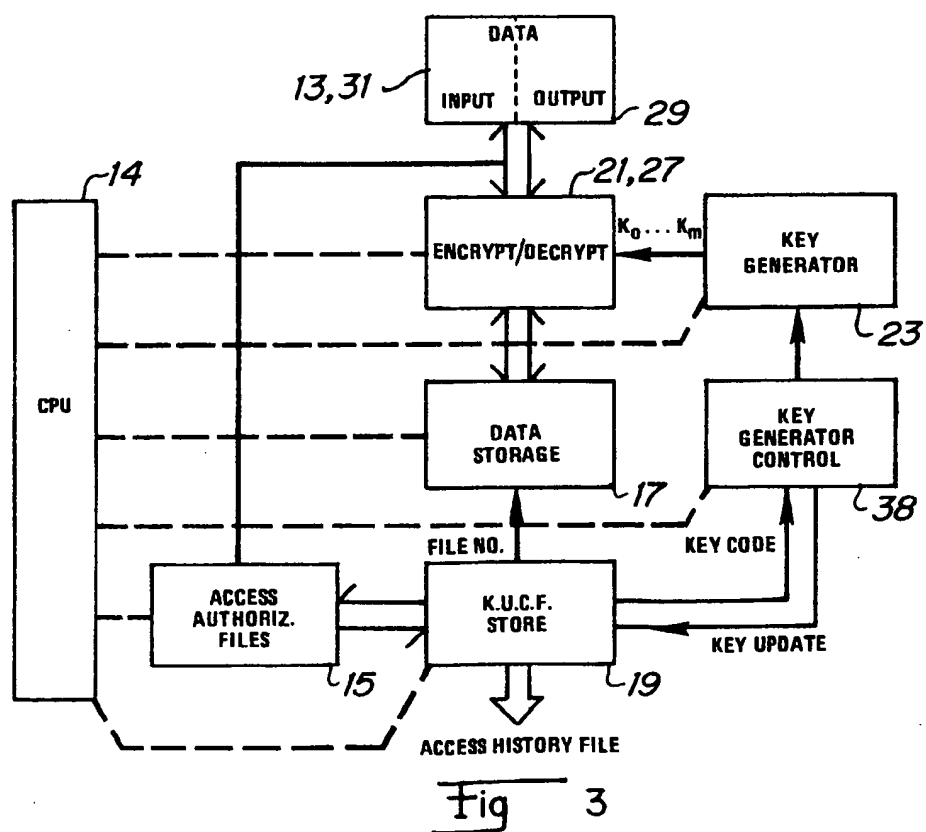
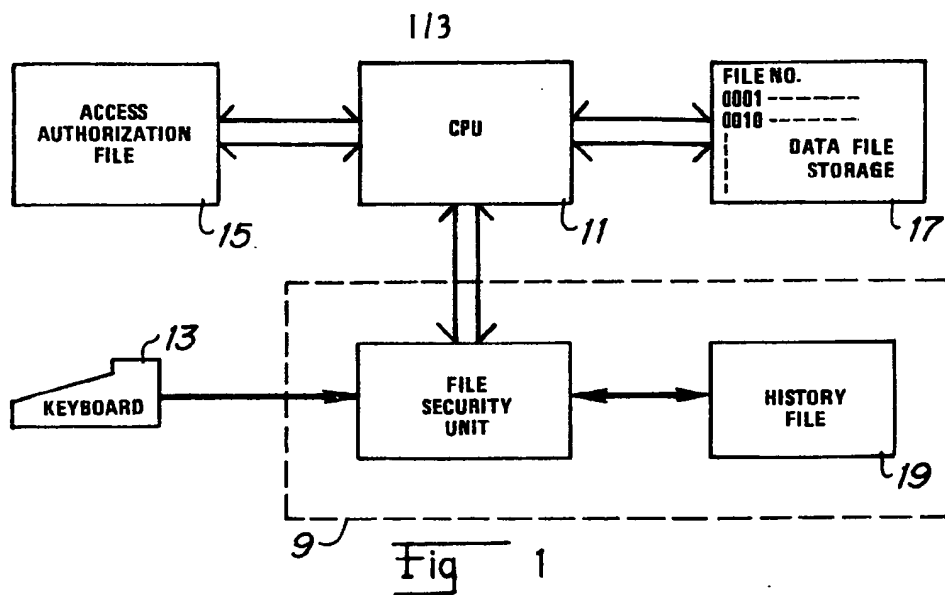
(58) Field of search
G4A

(54) File Access Security Method and Means

(57) An improved file access security technique and associated apparatus 9 accesses data which is stored at 17 in encrypted form under one encryption key and re-stores the data re-encrypted under another encryption key, and produces at 19 a record of each access and data re-encryption both as the control source of encryption keys for access and re-entry of encrypted data and as a secured audit record of users that had access to each file.



GB 2 136 175 A



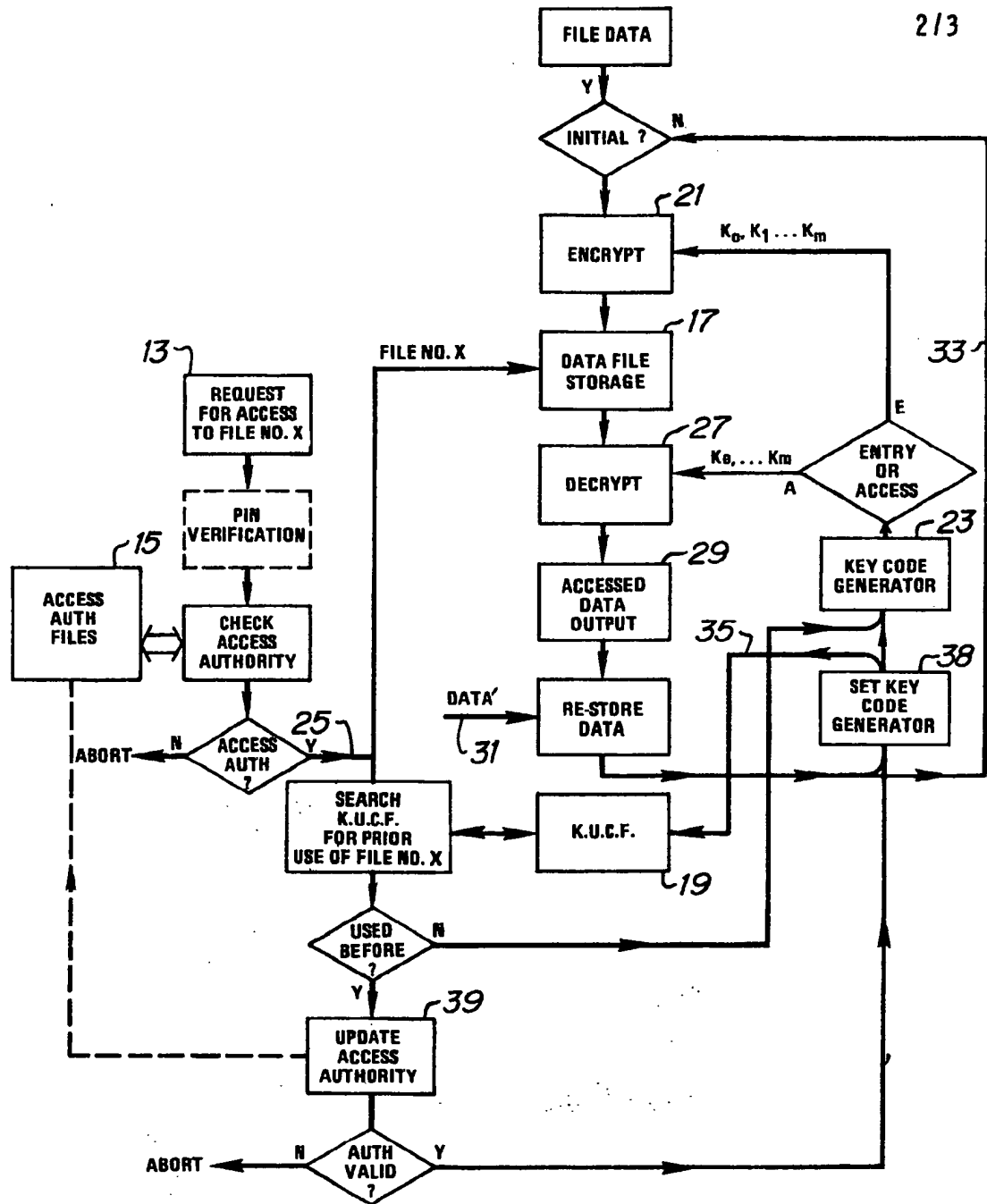


Fig. 2

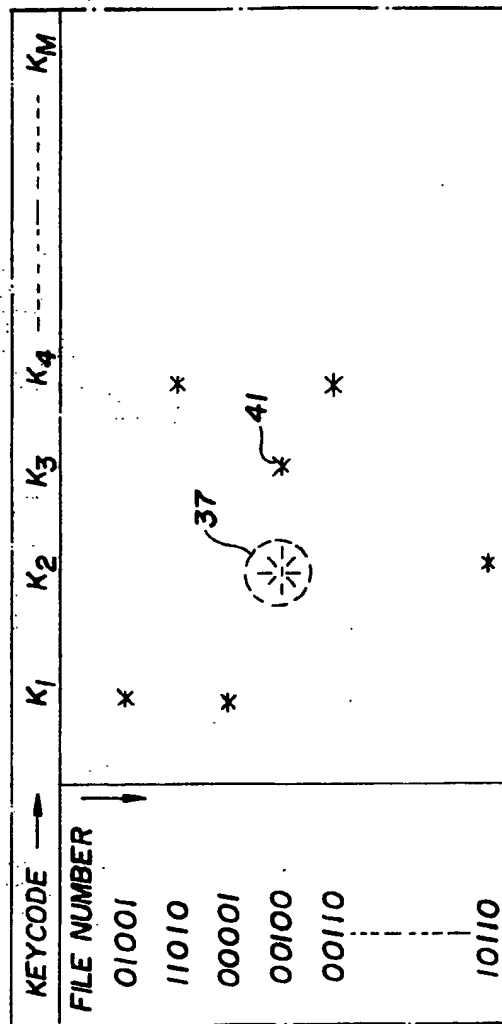


FIG. 4

SPECIFICATION

File Access Security Method and Apparatus

This invention is concerned with a method of and apparatus for securing data files in storage.

- 5 Many known computer-controlled operations on secured data files require verification of the identity of an individual seeking to access a file before the data (usually in encrypted form) can be accessed (see, for example, U.S. Patents
10 3,938,091, 3,587,051, 3,611,293 and 4,198,619). In addition, many known record-securing schemes including those associated with credit cards, require verification of both the authority of the using individual and the
15 authenticity of the data in the record, to protect against unauthorized users and against counterfeit or duplicate records. Schemes of this type are disclosed in U.S. Patents 4,304,990, 4,328,414 and 4,357,429.
- 20 One disadvantage associated with computer-controlled security schemes of these types is that there is typically no indication left on file of who gained access to a secured record.

- 25 The present invention provides a method of securing data files in storage against unauthorized access, the method comprising the steps of encrypting file data as a selected logical combination thereof with an initial one of a plurality of encryption key codes to produce file
30 data in encrypted form for storage at selected file address locations, establishing a record of accesses to each selected file address location and the one of the plurality of encryption key codes with which the file data at the address
35 location is encrypted, processing a request for access to file data at a selected file address location by determining from the record the number of prior accesses thereof and the encryption key code associated therewith,
40 decrypting file data at the selected file address location using said associated encryption key code, re-encrypting file data for said selected file address location using a new one of said plurality of encryption key codes in said selected logical
45 combination, storing the newly re-encrypted file data at the accessed file address location, and modifying the record to indicate an additional access to the selected file address location and the new encryption key code associated
50 therewith.

- In performing a method as set forth in the last preceding paragraph, it is preferred that in carrying out the step of decrypting, file data at a selected file address location is decrypted using
55 said initial encryption key code in response to determination from the record that said selected file address location was not previously accessed.

- A method as set forth in either one of the last two immediately preceding paragraphs may
60 further comprise the additional steps of establishing a file of user access authorizations, and prior to accessing a selected file address location, determining the authorization status of a

- user to gain access to the selected file address location.

- A method as set forth in the last preceding paragraph may further comprise the additional step of selectively altering the access
70 authorization of a user to gain subsequent access to the selected file address location in response to re-encryption of the file data for storage at the selected file address location.

- A method as set forth in any one of the last four immediately preceding paragraphs may
75 further comprise the steps of reinitializing all the file data by decrypting the file data at each selected file address location using the encryption key code thereof determined from the record, and re-encrypting the file data at each such file
80 address location using a new initial one of a plurality of key codes.

- In performing a method as set forth in the last preceding paragraph, it is preferred that in carrying out the reinitialization step, the file data
85 at any file address location which is not indicated in the record to have been accessed previously is decrypted using the initial encryption key code.

- The present invention also provides apparatus for securing data files in storage against
90 unauthorized access, comprising storage means for storing file data in encrypted form at selectable file address locations, encryption means for supplying encrypted file data to a selected file address location as the logical
95 encoding combination of file data and an encryption key signal applied thereto, generator means for applying selected encryption key signals to the encryption means, record means for producing indication of selected file address
100 locations and key code signals associated with encryption of file data stored therein, circuit means responsive to identification of a selected file address location for determining from said record means the encryption key signal
105 associated therewith for setting the generator means to supply the associated encryption key signal, decryption means disposed to receive encryption key signals from the generator means and encrypted file data from the storage means
110 and operable in accordance with said logical encoding combination to decrypt the file data at said selected file address location, and means operable upon the decrypted file data for altering the generator means to supply a new encryption
115 key signal for re-storing the file data at the selected file address location newly encrypted with a new encryption key signal, said means altering the record means to produce an indication of the new encryption key signal
120 associated with file data in the selected file address location.

- In apparatus as set forth in the last preceding paragraph, it is preferred that said circuit means is responsive to the indication in said record means
125 that a selected file address location was not previously accessed for setting said generator means to supply the initial encryption key signal to the decryption means.

Apparatus as set forth in either one of the last two immediately preceding paragraphs may further comprise access record means for storing data representative of the authorization of users to selectively access file data in said storage means, and means disposed to receive identification data from a user, and coupled to said circuit means for inhibiting the generator means from supplying an encryption key signal to said decryption means for an unauthorized, identified user.

Apparatus as set forth in the last preceding paragraph may further comprise means responsive to re-storing of file data at the selected file address location newly encrypted with a new encryption key signal for altering the identified user's authorization in said access record means to access said selected file address location.

Apparatus as set forth in the last preceding paragraph but three may further comprise initializing means coupled to said generator means, said encryption means and decryption means and to said record means for setting the generator means to selectively decrypt file data in each file address location using the encryption key signals from said generator means established from the record means for each such file address location, and for re-encrypting the decrypted file data for each file address location using a new initial encryption key signal for re-storage at the respective file address location.

In apparatus as set forth in last preceding paragraph, it is preferred that said initializing means responds to indication from said record means of no previous access to a selected file address locations for decrypting file data therein in using an initial encryption key signal and for re-encrypting the decrypted file data using a new initial encryption key signal to re-store the newly encrypted file data at the respective file address location.

The present invention further provides a file access record produced by a process comprising the steps of storing at selected file address locations file data that is encrypted as the logical combination of file data and selected ones of a plurality of encryption key signals; decrypting file data at a selected file address location using the encryption key signal associated therewith in accordance with said logical combination, re-encrypting the decrypted file data as a logical combination thereof and a new encryption key signal for re-storing at the corresponding file address location, and producing said file access record as the compilation at least of the number of times each selected file address location was decrypted and information indicative of the encryption key signals with which the file data at each selected file address location was reencrypted and re-stored therein.

In accordance with the preferred embodiment of the present invention, a dynamic record of encryption control keys used to gain access initially and at all subsequent occasions to secured encrypted files is generated both as an

active element of the accessing scheme and as a secured, historic record for audit purposes of all accesses to encrypted files. In addition, substitutions of outdated files are prevented once a file is accessed, even merely for display without alteration, so that a file once accessed, and therefore with its security compromised, can be resecured against duplication, substitution, and re-use. Schemes of this type are particularly useful in banking and funds-transfer operations where proper access initially to an account file, for example, to effect a withdrawal of funds, must thereafter be carefully controlled to avoid such disastrous practices as multiple replication of the same operation coupled with substitution of the original balance back into the file. Further, the historic record of accesses to files produced by the present invention constitutes an audit record in encrypted form of such accesses.

There now follows a detailed description which is to be read with reference to the accompanying drawings of a method and apparatus according to the present invention; it is to be clearly understood that this method and apparatus have been selected for description to illustrate the invention by way of example and not by way of limitation.

In the accompanying drawings:

Figure 1 is a pictorial block diagram showing one application of the apparatus of the present invention;

Figure 2 is a flow chart illustrating the operation of the apparatus of Figure 1;

Figure 3 is a block diagram of the illustrated embodiment of the present invention; and

Figure 4 is a chart illustrating the formation and operation of the key usage control file according to the present invention.

Referring now to Figure 1, there is shown a pictorial block diagram of the present invention illustrating the addition of an access-securing module 9 to a typical computer system comprising a central-processing unit 11, keyboard controller 13, and memory means 15, 17 for storing files. The memory means 15, 17 may use any conventional form of storage technology such as semiconductor memory, magnetic memory in core, crystal, disc, drum, or tape form, and any combinations thereof, to provide means 17 for storing the data to which access is to be controlled, and to provide means 15 for storing access authorization information about individuals and entities that may access the stored data means 17. The keyboard controller 13 provides manual-entry access to the computer system in conventional manner and is representative of other computer-accessing schemes such as by another computer system, and the like.

In accordance with the present invention, such a typical computer system is modified to include the access-securing module 9 which operates with the computer system to progressively re-encrypt the data in storage in memory means 17 each time a file is accessed, and optionally to

update the access authorization information in storage in the memory means 15 in response to authorizations granted, and to generate historic files in encrypted form of the encryption keys used to decrypt and re-encrypt each file accessed from the memory means 17. In addition, the module 9 operates in a controlled reinitialization mode to restore all files in the memory means 17 to a new, standard encryption key after numerous accesses of files in the memory means 17 have been authorized. The number of accesses before requiring reinitialization is determined by the memory capacity in the module 9.

Referring now to Figures 2 and 3 in addition to Figure 1, there are shown a flow chart and a block diagram, respectively, illustrating the operation of the system of Figure 1 under control of a central processing unit 11. In operation, a person or entity, R, requesting access to a particular file may enter personal identification numbers, information about the particular file, and the like, via the keyboard controller 13. Optionally, a personal-identity verification routine may be performed in conventional manner (as disclosed, for example, in U.S. Patent 3,938,091 or 4,198,619) and the access-authorization files in the memory means 15 may be searched for authorization to access the requested file. All such files in memory means 17 are initially encrypted with an initial key code, K_0 , in a conventional manner (for example, using the Data Encryption Standard module available from the National Bureau of Standards) by encrypting the file data in an encryption module 21 with key code, K_0 , from a key code generator 23.

With authorization established 25, the particular file #X may be accessed, but decrypting the file #X requires the correct key code. For this purpose, a key-usage control file 19, later described herein in detail, is searched to determine if the file #X was previously accessed. The conditions of prior access, namely, that it was, or it was not previously accessed, are possible. If it was not, then file #X will not appear in the key-usage control file, an indication that it appears in storage provided by the memory means 17 encrypted with the initial key code, K_0 . The key code generator 23 is capable of generating a sequence of different key codes $K_0, K_1, K_2, K_3 \dots K_n$ and is set to supply key code K_0 to a decryption module 27 (which, of course, may be the same type of DES module, or may be the same module, as the encryption module 21). The requested file #X may therefore be decrypted in conventional manner using the key code K_0 to provide accessed data 29 in clear text. The data is then returned to storage, either without or with new data modifications 31 that reflect a data-oriented transaction such as sale, deposit, withdrawal, or the like, and is re-stored in encrypted form using a new key code K_1 . This is accomplished by resetting 38 the key code generator 23 to supply the key code K_1 to the encryption module 21 and returning the data 33 with or without modifications for encryption in the

module 21 with the key code K_1 . In addition, the key-usage control file 19 is updated to reflect that the file #X was accessed and now resides in storage newly-encrypted with the new key code K_1 in the sequence. Further, the access-authorization in the memory means files 15 may be updated optionally to inhibit further access to file #X by user R, for example, to inhibit R's further access until a "new date", or until accessed by another user, or the like. Subsequent access to file #X by user R, if continuously authorized, or by any other user must be via decryption with the key code K_1 .

If file #X was previously accessed, then the key-usage control file 19 will contain the entry of file #X having been previously accessed and returned to storage encrypted with a new key code $K_1, K_2 \dots K_n$, depending upon the number of previous accesses to file #X. Thus, with reference to the chart of Figure 4 which illustrates the typical entries in the key-usage control file 19, if file #X is file #00100, then the previous accesses to this file resulted in its being re-stored encrypted with key code K_2 (at entry 37). The search of the key-usage control file 19 thus indicates that file #00100 was previously accessed twice and now requires decryption with key code K_2 . If authorization of the requesting user is still valid 39, then the key code generator 23 is set to supply the key code K_2 to the decryption module 27 in order to furnish the data in this file in clear text 29. Re-storing the data from this file in modified or unmodified form is accomplished by resetting 38 the key code generator 23 to supply the key code K_3 (entry 41 in Figure 4) to the encryption module 21 for encryption therein of the returned data with the new key code K_3 . All retrievals of data in storage in the memory means 17 may be by destructive read of information in the addressed file so that data for restoring therein may be written in the newly-encrypted form. After numerous accesses to files in storage in the memory means 17, the key-usage control file 19 will typically include entries as illustrated in Figure 4. Such a file optionally may also include codes to identify the particular users who gained access to each file. The file 19 thus provides an audit record of the accesses to the files in the memory means 17. In addition, the key-usage control file 19 is in encrypted form since it neither reveals the data in storage in the memory means 17 nor the actual key codes $K_1 \dots K_n$ (only generated by the generator 23) required to decrypt the data in storage. Further, the key codes $K_0 \dots K_n$ which serve as file-protect codes can be generated internally in conventional manner, for example, by a random-number generator 23 and therefore need not be known to anyone.

After numerous accesses to the data in storage 17 which approaches the limit of the sequence of key codes for any particular file, or on a periodic basis, the entire collection of files in storage 17 may be re-encrypted with a new initial key code K_0' of a sequence of new key codes $K_0', K_1' \dots K_n'$

using the apparatus illustrated in Figure 3 under control of the central processing unit 14. However, since the files in storage 17 are encrypted with different key codes, the key-usage control file 19 must be consulted to determine which key code to use to decrypt the data in each file for re-encryption with a new initial key code K_0' . After completion of this reinitialization mode of operation, the key-usage control file 19 for the sequence of key codes $K \dots K_n$ may be retired to serve as an historic record of access to the data in storage 17 without compromising the security of the system or of the data in storage 17 under new encryption codes.

15 CLAIMS

1. A method of securing data files in storage against unauthorized access, the method comprising the steps of:

- 20 encrypting file data as a selected logical combination thereof with an initial one of a plurality of encryption key codes to produce file data in encrypted form for storage at selected file address locations;
- 25 establishing a record of accesses to each selected file address location and the one of the plurality of encryption key codes with which the file data at the address location is encrypted;
- 30 processing a request for access to file data at a selected file address location by determining from the record the number of prior accesses thereof and the encryption key code associated therewith;
- 35 decrypting file data at the selected file address location using said associated encryption key code;
- re-encrypting file data for said selected file address location using a new one of said plurality of encryption key codes in said selected logical combination;
- 40 storing the newly re-encrypted file data at the accessed file address location; and
- modifying the record to indicate an additional access to the selected file address location and the new encryption key code associated therewith.

2. A method according to claim 1 wherein, in carrying out the step of decrypting, file data at a selected file address location is decrypted using said initial encryption key code in response to determination from the record that said selected file address location was not previously accessed.

3. A method according to either one of claims 1 and 2 comprising the additional steps of establishing a file of user access authorizations; and prior to accessing a selected file address location

determining the authorization status of a user to gain access to the selected file address location.

4. A method according to claim 3 comprising the additional step of selectively altering the access authorization of a user to gain subsequent access to the selected file address location in

response to re-encryption of the file data for storage at the selected file address location.

5. A method according to any one of the preceding claims and further comprising the steps of:

- 70 reinitializing all the file data by decrypting the file data at each selected file address location using the encryption key code therefor determined from the record; and
- 75 re-encrypting the file data at each such file address location using a new initial one of a plurality of key codes.

6. A method according to claim 5, wherein, in carrying out the reinitialization step the file data at any file address location which is not indicated in the record to have been accessed previously is decrypted using the initial encryption key code.

7. A method of securing data files in storage against unauthorized access substantially as hereinbefore described with reference to the accompanying drawings.

8. Apparatus for securing data files in storage against unauthorized access, comprising:

- storage means for storing file data in encrypted form at selectable file address locations;
- 90 encryption means for supplying encrypted file data to a selected file address location as the logical encoding combination of file data and an encryption key signal applied thereto;
- generator means for applying selected encryption key signals to the encryption means;
- 95 record means for producing indication of selected file address locations and key code signals associated with encryption of file data stored therein;
- 100 circuit means responsive to identification of a selected file address location for determining from said record means the encryption key signal associated therewith for setting the generator means to supply the associated encryption key signal;
- 105 decryption means disposed to receive encryption key signals from the generator means and encrypted file data from the storage means and operable in accordance with said logical encoding combination to decrypt the file data at said selected file address location; and
- 110 means operable upon the decrypted file data for altering the generator means to supply a new encryption key signal for re-storing the file data at the selected file address location newly encrypted with a new encryption key signal, said means altering the record means to produce an indication of the new encryption key signal associated with file data in the selected file address location.

9. Apparatus according to claim 8 wherein said circuit means is responsive to the indication in said record means that a selected file address location was not previously accessed for setting said generator means to supply the initial encryption key signal to the decryption means.

10. Apparatus according to either one of claims 7, 8 and 9 and further comprising:
 access record means for storing data
 representative of the authorization of users
 to selectively access file data in said storage
 means; and
 means disposed to receive identification data
 from a user, and coupled to said circuit
 means for inhibiting the generator means
 from supplying an encryption key signal to
 said decryption means for an unauthorized,
 identified user.
11. Apparatus according to claim 10 comprising
 means responsive to re-storing of file data at the
 selected file address location newly encrypted with a
 new encryption key signal for altering the identified
 user's authorization in said access record means to
 access said selected file address location.
12. Apparatus according to claim 8 comprising
 initializing means coupled to said generator
 means, said encryption means and decryption
 means and to said record means for setting the
 generator means to selectively decrypt file data in
 each file address location using the encryption
 key signals from said generator means
 established from the record means for each such
 file address location, and for re-encrypting the
 decrypted file data for each file address location
 using a new initial encryption key signal for
 restorage at the respective file address location.
13. Apparatus according to claim 12 wherein
 said initializing means responds to indication from

- said record means of no previous access to a
 selected file address location for decrypting file
 data therein in using an initial encryption key
 signal and for re-encrypting the decrypted file
 data using a new initial encryption key signal to
 re-store the newly encrypted file data at the
 respective file address location.
14. Apparatus for securing data files in storage
 against unauthorized access substantially as
 hereinbefore described with reference to the
 accompanying drawings.
15. A file access record produced by a process
 comprising the steps of:
 storing at selected file address locations file
 data that is encrypted as the logical
 combination of file data and selected ones of
 a plurality of encryption key signals;
 decrypting file data at a selected file address
 location using the encryption key signal
 associated therewith in accordance with said
 logical combination;
 re-encrypting the decrypted file data as a
 logical combination thereof and a new
 encryption key signal for restoring at the
 corresponding file address location; and
 producing said file access record as the
 compilation at least of the number of times
 each selected file address location was
 decrypted and information indicative of the
 encryption key signals with which the file
 data at each selected file address location
 was re-encrypted and re-stored therein.

(21) Application No 9009655.3

(22) Date of filing 30.04.1990

(30) Priority data

(31) 415984

(32) 02.10.1989

(33) US

(71) Applicant

Sun Microsystems Inc

(Incorporated in the USA - Delaware)

2550 Garcia Avenue, Mountain View, California 94043,
United States of America

(72) Inventor

John Richard Corbin

(74) Agent and/or Address for Service

Potts Kerr and Co

15 Hamilton Square, Birkenhead, Merseyside, L41 6BR,
United Kingdom

(51) INT CL^a

G06F 1/00

(52) UK CL (Edition K)

G4A AAP

(56) Documents cited

EP 0002390 A1 WO 88/02202 A1

(58) Field of search

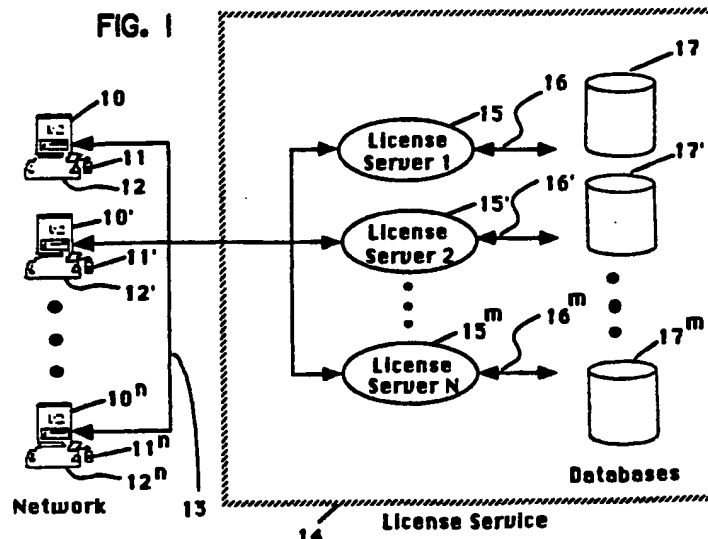
UK CL (Edition K) G4A AAP

INT CL^a G06F 1/00 12/14

Online database: WPI

(54) Protecting against the unauthorised use of software in a computer network

(57) The present invention provides to a software application the verification and licence check out functions which are normally performed by a licence server. The encrypted licence information is contained in a licence token, and is stored in a database 17 controlled by the licence server 15. In contrast to the prior art where the server either grants or denies the request after verifying the user's credentials, the server in the preferred embodiment of the present invention finds the correct licence token for the software application and transmits the token to a licencing library. A licence access module attached to the application decodes the token. Routines in the licencing library coupled to the software application verify the licence information before issuing the licence and updating the token. The access module then encodes the updated token before returning it to the server. Because the verification and issuing function of a token are performed by a software application, the application rather than the server becomes the point of attack by unauthorised users. Reverse engineering the access module is less rewarding than attacking the server because the module reveals the contents of a small fraction of a database of licences.



At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

GB 2 236 604 A

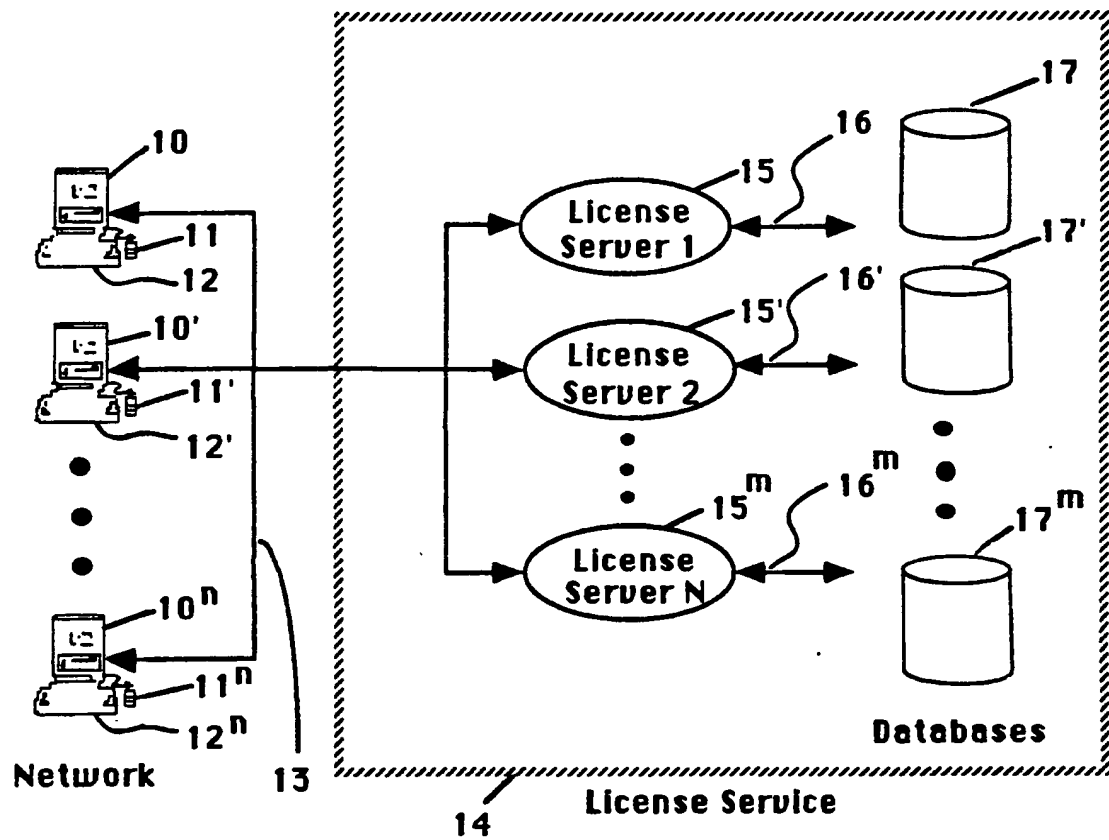


FIG. 1

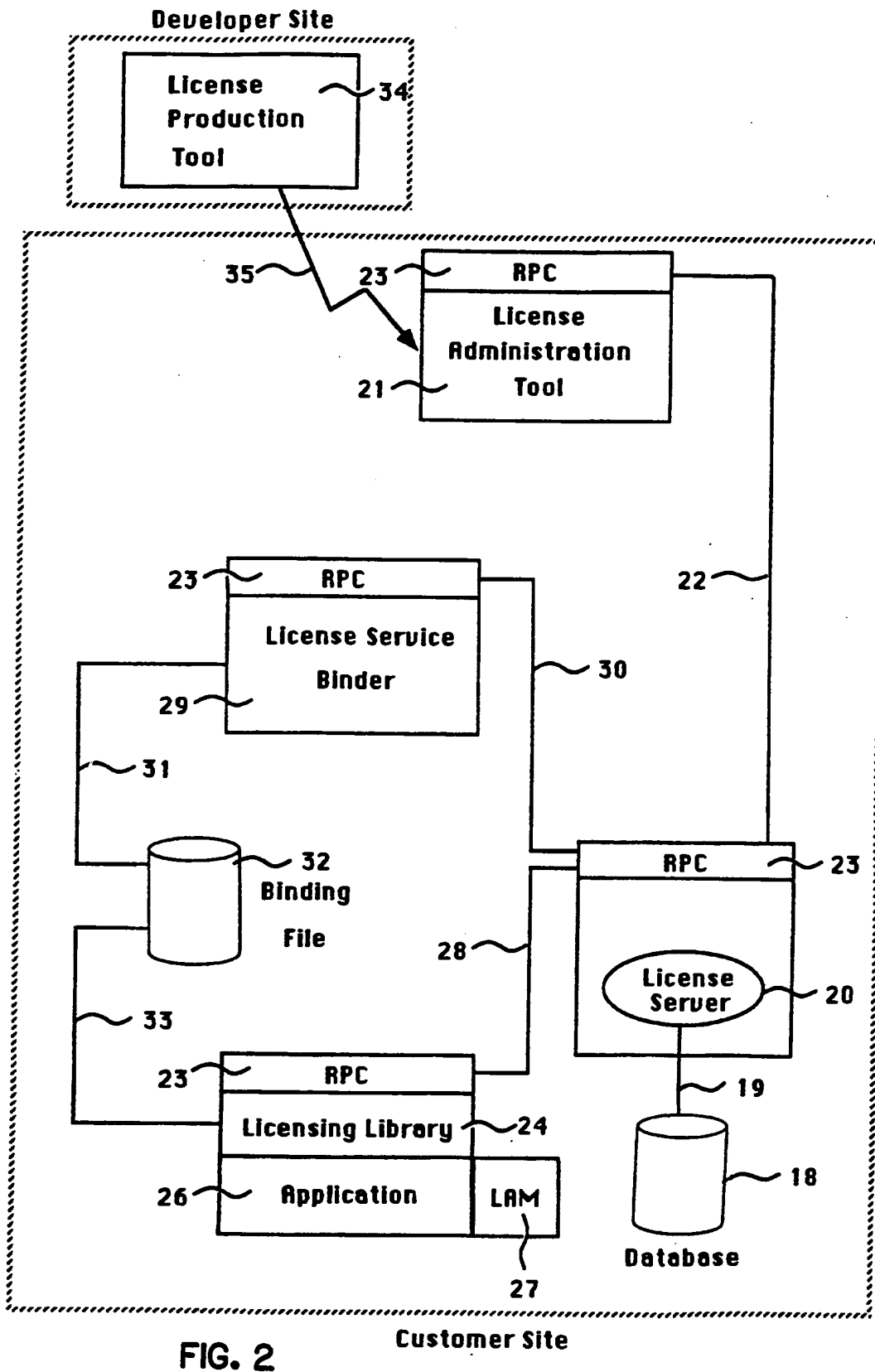


FIG. 2

Customer Site

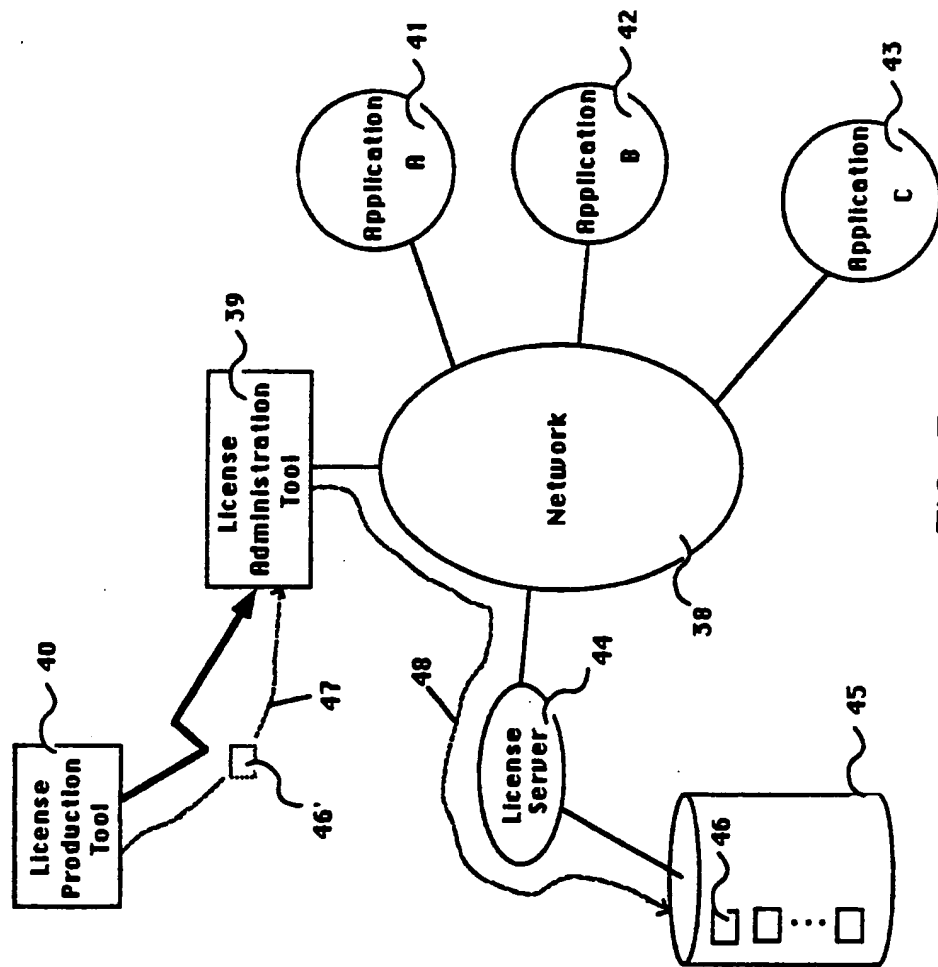
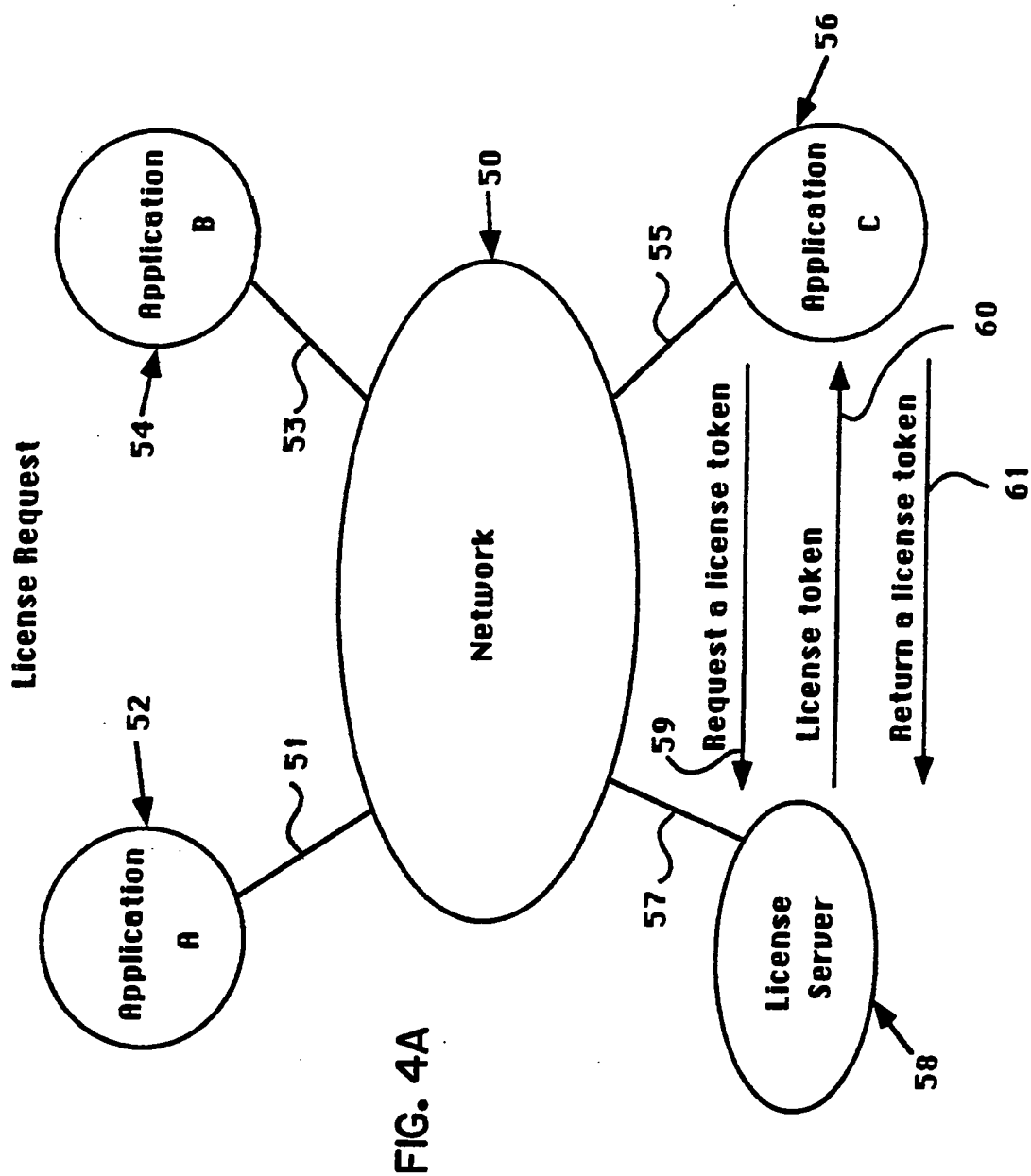


FIG. 3



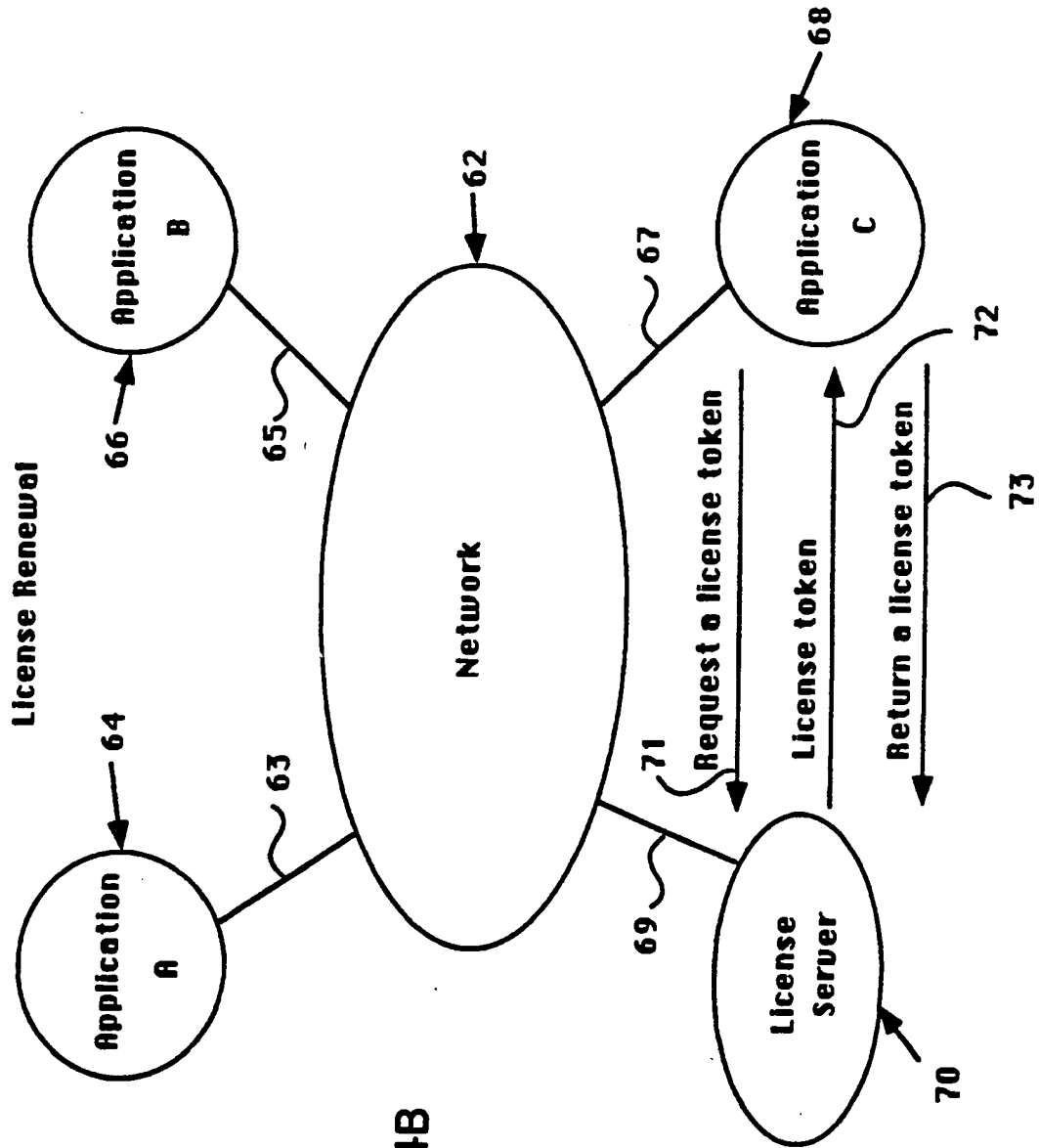


FIG. 4B

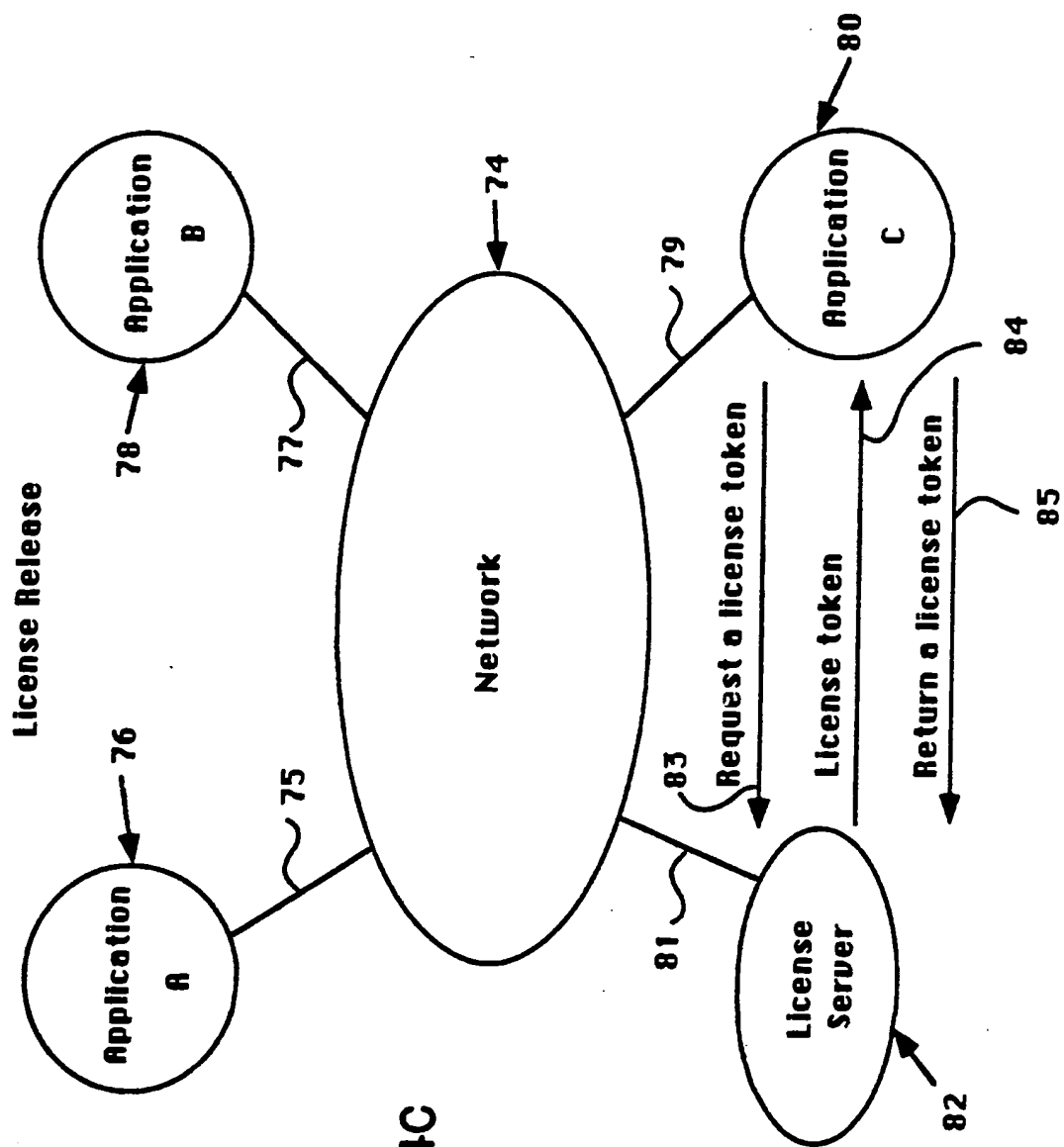


FIG. 4C

METHOD FOR PROTECTING AGAINST THE UNAUTHORIZED USE
OF SOFTWARE IN A COMPUTER NETWORK ENVIRONMENT

BACKGROUND OF THE INVENTION

1. FIELD OF THE INVENTION

The present invention relates to a method for protecting against
5 the unauthorized use of a software application in a computer network
environment.

2. ART BACKGROUND

A computer network is typically an interconnection of machines or
10 agents over links or cables. The open access characteristics of a computer
network presents opportunities for the unauthorized copying of software, thus
eroding the licensing revenue potential of software developers. Traditionally,
either the entire network must be licensed (commonly referred to as a site
license), or each node where the software is run must be licensed (commonly
15 referred to as a node license). A node refers to a single machine, agent or
system in a computer network. A license is an authorization given by a
software developer to a customer to use a software application in a specific
manner.

20 A site license lets all users at a designated location or network
use the software application, regardless of their position on the network. This
flat-fee approach is an overkill for a low usage software application. A node
license not only ties a software application to a particular machine in a
network, but also is not cost effective for the infrequent use of a software
25 application. See, for example, U.S. Patent No. 4,688,169. Furthermore, if new
users of licensed nodes wish to use the software application, they are often
required to purchase additional licenses.

An alternative to a site license or a node license is the concept of
30 a concurrent usage license. A concurrent usage license restricts the number
of users allowed to use a software application at any given time, regardless of
their location on the network. Just as renters check out available copies of a

movie video from a video rental store, users on a network check out a software application from an agent on a first-come-first-serve basis. Thus, a concurrent usage license charges a fee for the use of a software application proportional to its actual use.

5

Methods to license a software application for concurrent use in a network environment are currently offered by Highland Software, Inc. and Apollo Computer, Inc. See, M. Olson and P. Levine, "Concurrent Access Licensing", *Unix Review*, September 1988, Vol. 6, No. 9. In general, the

10 license for a software application is stored in a database controlled by a license server. A license server is a program that not only stores the license, but also verifies the user's credentials before checking out the license to the authenticated user. To protect against the unauthorized use, these methods to license concurrent usage rely on secured communications such as

15 public/private key encryption. Under public/private key encryption, each user of the system has two keys, one of which is generally known to the public, and the other which is private. The private transformation using the private key is related to the public one using the public key but the private key cannot be computationally determined from the public key. See Denning, D.,

20 *Cryptography and Data Security*, Addison-Wesley, 1982. The encryption key is hidden in the license server to encrypt the database of licenses. Well designed public/private key encryption schemes are difficult to crack, especially if the license server is located in a trusted environment. A trusted environment is one whose access is limited to users having the proper

25 credentials. However, a license server is more likely to be located at a customer's site and hence in an hostile environment. It follows that the license server is vulnerable to sophisticated intruders. Once the private key is decrypted, all sensitive information on the license server such as licenses are compromised.

30

It is therefore an object of the present invention to provide a more secure method to protect against the unauthorized use of software in a concurrent use licensing environment.

SUMMARY OF THE INVENTION

The present invention provides to the software application the verification and license check out functions which are normally performed by a license server. The preferred embodiment of the present invention comprises a computer network including a plurality of agents running at least one license server and at least one software application. The license server controls a database of an agent containing the license information for the software application. The license information is contained in a license token, and is stored in the database controlled by the license server. The license token is a special bit pattern or packet which is encrypted by the software vendor of the application software. The software application communicates with the license server through a licensing library. The licensing library is a collection of library routines that the software application invokes to request or renew a license from the license server. Before a software application obtains a license, the license token must be decoded by a license access module. The license access module, which is linked with the software application and the licensing library is a program that decodes the license token from a vendor specific format to a licensing library format.

20

When an user wishes to run a software application, the licensing library invokes a call to request a license token from the license server. In contrast to the prior art where the license server either grants or denies the request after verifying the user's credentials, the license server in the preferred embodiment of the present invention finds the correct license token for the software application and transmits the license token to the licensing library. The license access module attached to the licensing library decodes the licensing token. Routines in the licensing library coupled to the software application verify the license information before checking out the license and updating the license token. The license access module encodes the updated license token before returning it to the license server.

30

Because the verification and check out function of a license token are performed by a software application, the software application rather than the license server becomes the point of attack by unauthorized users. Reverse engineering the license access module is less rewarding than attacking the

5 license server because the license access module reveals the contents of a fraction of a database of licenses. By the time most attackers crack the license access module, the software vendors would most likely introduce newer versions of the software application and new license access modules for them. Thus the present invention provides a more secure method for protecting

10 against the unauthorized use of a software application in a computer network environment without modifying the underlying computer network.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a network environment employing the present invention.

5

Figure 2 describes the architecture of a network licensing scheme employing the preferred embodiment of the present invention.

Figure 3 describes the installation of a license token in the preferred
10 embodiment of the present invention.

Figure 4a illustrates the use of a license token to request a license from a license server in the preferred embodiment of the present invention.

Figure 4b illustrates the use of a license token to renew a license from
15 a license server in the preferred embodiment of the present invention.

Figure 4c illustrates the use of a license token to release a license from a license server in the preferred embodiment of the present invention.

20

NOTATION AND NOMENCLATURE

The detailed description that follows is presented largely in terms of algorithms and symbolic representations of operations on data bits and data structures within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art.

10 An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. These steps are those requiring physical manipulation of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It proves
15 convenient at times, principally for reasons of common usage, to refer to these signals as bit patterns, values, elements, symbols, characters, data packages, or the like. It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities.

20

Further, the manipulations performed are often referred to in terms, such as adding or comparing, that are commonly associated with mental operations performed by a human operator. No such capability of a human operator is necessary, or desirable in most cases, in any of the operations described
25 herein that form part of the present invention; the operations are machine operations. Useful machines for performing the operations of the present invention include general purpose digital computers or other similar devices. In all cases there should be borne in mind the distinction between the method of operations in operating a computer and the method of computation itself. The
30 present invention relates to method steps for operating a computer in processing electrical or other (e.g. mechanical, chemical) physical signals to generate other desired physical signals.

The present invention also relates to an apparatus for performing these operations. This apparatus may be specially constructed for the required purposes, or it may comprise a general purpose computer as selectively
5 activated or reconfigured by a computer program stored in the computer. The algorithms presented herein are not inherently related to any particular computer or other apparatus. In particular, various general purpose machines may be used with programs written in accordance with the teachings herein, or it may prove more convenient to construct a more specialized apparatus to
10 perform the required method steps. The required structure for a variety of these machines will appear from the description given below.

DETAILED DESCRIPTION OF THE INVENTION

The following detailed description is divided into several sections. The first of these sections describes a general network environment for accessing a database of licensed software programs. Subsequent sections discuss the details of a method for protecting against the unauthorized use of a software application.

I. General Network Environment

Referring to Figure 1, computer network environment comprises a plurality of data processing devices identified generally by numerals 10 through 10ⁿ (illustrated as 10, 10' and 10ⁿ). These data processing devices may include terminals, personal computers, workstations, minicomputer, mainframes and even supercomputers. For the purposes of this Specification, all data processing devices which are coupled to the present invention's network are collectively referred to as "agents". It should be understood that the agents may be manufactured by different vendors and may also use different operating systems such as MS-DOS, UNIX, OS/2, MAC OS and others. Particular examples of suitable agents include machines manufactured by Sun Microsystems, Inc., Mountain View, Calif. Each of the agents has an input device such as a keyboard 11, 11' and 11ⁿ or a mouse 12, 12' and 12ⁿ. As shown, agents 10 through 10ⁿ (illustrated as 10, 10' and 10ⁿ) are interconnected for data transfer to one another by a common cable 13. It will be appreciated by one skilled in the art that the common cable 13 may comprise any shared media, such as coaxial cable, fiber optics, radio channel and the like. Furthermore, the network resulting from the interconnection of the cable 13 and agents 10 through 10ⁿ (illustrated as 10, 10' and 10ⁿ) may assume a variety of topologies, such as ring, star, bus, and may also include a collection of smaller networks linked by gateways or bridges.

Referring again to **Figure 1** is a license service **14**. The license service **14** is a resource shared by every agent connected to the network. In the preferred embodiment of the present invention, the license service **14** comprises license servers **15** through **15^m** (illustrated as **15**, **15'** and **15^m**) and databases **17** through **17^m** (illustrated as **17**, **17'** and **17^m**), where **m** is less than or equal to **n**. A license server is a program that runs on an agent with a memory storage capability. Each license server **15** (illustrated as **15**, **15'** and **15^m**) communicates with a database **17** stored in memory on the agent over an interface **16** (illustrated as **16**, **16'** and **16^m**). As will be described in detail below, the database **17** stores licensing information for various software applications which are purchased and authorized to run in the computer network environment. The license server is not limited to run on a specific agent, but can operate on any agent including the agent on which the user is to operate the application. Thus, any agent connected to the network may function as a license server as well as a device on which a user may operate application software. As will be described below, the license server does not perform verification of licenses of application software; rather the license server is passive and provides storing, locking, logging, and crash recovering function for the application software.

20

Figure 2 illustrates the architecture of a network licensing scheme of the present invention. The architecture comprises a database **18**, database interface **19**, license server **20**, licensing library **24**, License access module **27**, license administration tool **21**, license service binder **29**, and license production tool **34**.

25

The database **18** stores licensing information and application usage data. Preferably the database **18** comprises a plurality of records which contain the following information:

	<u>Database Element</u>	<u>Description</u>
	Unique Key Table	Keys for all other tables
	Vendor Table	Vendor's ID and name
	Product Table	Product number and name
5	Version Table	Version number and date
	License Table	License #, exp date, total units
	License Token Table	Stores encoded license token
	Unit Group Table	A group's allocation of license
	Group List Table	Name of the group
10	Allowed Users Table	Credentials of allowed users
	Current License Use Table	Applications using a license
	Lock Table	Locked records in database
	Authorized administrator Table	Login names of administrators
	License Operation Log Table	Administrator's log information
15	License Usage Log Table	Request handle plus Client Log
	License Queue Log Table	License wait queue
	Application Message Log Table	Application specific messages

20

A database interface 19 provides communication between the license server 20 and the database 18 in order to prevent concurrent access to the same database record by multiple users which can cause the data in the record to become corrupted. Thus, only the owner of the lock can read from
25 and write to the locked record during the usage of the application.

The license server 20 operates on an agent and interfaces the database 18 to license administration tool 21, licensing library 24 and license service binder 29. The license server 20 communicates with the license
30 administration tool 21, licensing library 24 and license service binder 29 via an interface 23. Preferably the interface 23 is a remote procedure call

mechanism which permits a process operating on one device or agent connected to the network to request a resource or service from a remote device or agent connected to the network. See A. Birrell and B. Nelson, "Implementing Remote Procedure Calls," *ACM Transaction on Computer Systems*, February 5 1984, Vol. 2, No. 1.

Multiple license servers may reside on multiple agents. Preferably the license server 20 operates in a background mode of the agent such that its operation is transparent to a user of that agent. More particularly, as will be 10 described below, the license server 20 provides the following functions: (1) servicing the requests from the licensing library 24 for license token; (2) maintaining a wait queue for requests to the database 18 when no licensing units are available; (3) generating locks for exclusive access to database 18; and (4) providing access to information in the database 18.

15 The licensing library 24 is a set of library routines which enable the application 26 to request licensing service from the license server 20. Upon receiving the request for service from the licensing library 24, the license server 20 retrieves a license token from the database 18 and transmits it to the 20 licensing library 24. The licensing library 24 is linked with the application 26 and communicates with the license server 20 over a path 28 with, preferably, a remote procedure call mechanism 23. Among the major library calls in the licensing library 24 is the application's request for a license from the license server 20. Other important library calls include the request to renew and to 25 release a license. The use of the license token to accomplish the request for the various licensing service will be described in detail below.

The license access module (LAM) 27 is prepared by the software vendor 24 to decode the license token. Once decoded, the application 26 via 30 routines in the licensing library verifies the licensing information in the license token and determines whether a license may be checked out. The LAM 27

also encodes the license token before the application returns it to the database 18 via license server 20. The license access module 27 is described in further detail below.

5 The license administration tool 21 is utilized by the network administrator to perform administrative functions relevant to the concurrent usage of a software application. The license administration tool 21 may run on any agent connected to the computer network. The license administration tool 21 is primarily used to install the license token into the database 18 through the
10 license server 20. The functionality of the license administration tool 21 includes: (1) starting or terminating a license server, (2) accessing a database controlled by a license server; and (3) generating and printing reports on license usage.

15 The application 26 may not access the database 18 directly; rather, the request for a license is made through the licensing library 24 to the license server 20 over a path 28. Most network licensing schemes employ secured communication between the licensing library 24 and the license server 20. In contrast, the present invention uses the license access module (LAM) 27 the
20 license library 24 and a plurality of license tokens to protect against the unauthorized use of software application in a computer network.

Referring once again to Figure 2, a license service binder 29 is shown coupled to the license server 20 over a path 30. The license service binder
25 29 is invoked by means known in the art, such as a network service program. The license service binder 29 locates all agents that are designated as servers on the network, and keeps track of which server is servicing which application. The license service binder 29 contacts each server on its table of available servers and requests a list of products it serves. Finally the license service
30 binder 29 writes the contents of the table of available license servers and the list of products into a binding file 32 over a path 31. In Figure 2, the binding file 32 is coupled to the licensing library 24 over a path 33. The application 26

queries the binding file 32 to see which license server can service its request for a license.

5 A license production tool 34 is used by the software vendor to create a license token for transmittal to the network administrator. Receiving the license token, the network administrator installs it with the license administration tool 21 into the database 18 through license server 20.

II. License Token

10 Referring to Figure 3, the creation of a license token in a computer network employing the preferred embodiment of the present invention will be described. A computer network 38 is shown coupled with a license administration tool 39 and a single license server 44. The license server 44 communicates with a database 45. Applications 41, 42, and 43 are shown
15 requesting licensing service from the license server 44. When a customer purchases a license for an application, such as a CAD/CAM program for its research and development department, the software vendor creates a license token with a license production tool, and delivers the license token to the customer's network administrator. A license token is a special bit pattern or
20 packet representing a license to use a software application. The network administrator installs the license token 46 into the database of the license server using the license administration tool 39. Unlike the token used in a token ring which is passed from agent to agent, a license token in the preferred embodiment of the present invention is passed only between a license server
25 and a licensing library for a predetermined amount of time. The predetermined amount of time corresponds to the time the license token is checked out of the license server. Currently, the license token is checked out to an application for no more than ten seconds, and the license token is returned as quickly as possible to the issuing license server. The license token 46 contains
30 information encrypted in the vendor's format such as vendor identification, product and version numbers as well as the number of license units purchased

for the license token. A license unit corresponds to the license weighting for an agent connected to the computer network. For example, powerful workstations could require more license units to use a software application than an average personal computer.

5

The software vendor produces a license token using a license production tool 40. A path 47 illustrates how a license token 46' makes its way to a license administration tool 39 at the customer's site. There, the system administrator installs the license token 46' as license token 46 into the license database 45 of the license server 44. A path 48 indicates the transfer of the license token 46' from the license administration tool 39 to the license server 44 and into the database 45 as license token 46. The license server 44 is now ready to entertain requests from applications 41, 42, and 43 for a license to use the application corresponding to token 46 as well as other applications represented in its database 45.

It should be understood that each network may have a plurality of license servers and each license server may have in its database a plurality of license tokens for a variety of software applications. Referring again to Figure 3, If application A 41 requests and checks out the license token 46 for less than ten seconds, applications B and C 42, 43 would be unable to check out the license token 46 if their requests were made during the same time application 41 is checking out a license from the license token 46 because of the locking mechanism provided by database interface 19. Thus, to achieve concurrent license usage in network 38, it is preferred that the network administrator installs more than one license server. To minimize the task of recovering from license server crashes, it is also preferred that the system administrator spreads the license units for any one application among a plurality of strategically located license servers. For instance, if a network has four license servers, the network administrator may want to allocate the twenty license units for a particular popular application among four license tokens with

five license units for each license token. In the event one license server crashes or the license token is checked out, the other three license servers may provide licensing service to other applications.

5 **Figure 4a illustrates the use of a license token to request for a license.**
As shown, a network 50 is provided, and is coupled to Applications 52, 54
and 56 respectively. Application 56 succeeded in requesting a license token
from the license server 58 in step 59. The license token is transmitted to
application 56 in step 60. When done, Application 56 returns the license
10 token to the license server 58 in step 61. Aside from the license request
function performed with the license token as shown in Figure 4a, the license
token is also used in other critical stages of the licensing process. For
example, an user may wish to run an application beyond the initial allocated
time. As shown in Figure 4b, Application 68 makes a license renewal
15 request 71 from the license server 70 with license token 72. Similarly, in
Figure 4c the user makes a license release request 83 when the application
no longer needs the license units. As such, the user updates the license token
84 by returning the updated license token to the license server 82 in step 85.

20

III. License Access Module

In Figure 2, a license access module (LAM) 27 is linked with the
application 26 and the licensing library 24 to form the executable code that
software vendors ship to the customers. The license access module 27
decodes and encodes the encrypted license token as it is being passed
25 between the license server and the licensing library 24. Thus the level of
security of an application from unauthorized use depends heavily upon how
secure the license access module is.

Conventional network licensing schemes use public/private key
30 encryption to encode sensitive information. Such a scheme is effective if the
license server is in a trusted environment. However, the customer has the

same access to any agent in a network, including the license server. The security of the licensing scheme can be compromised by a user who decrypts the license server's private key. Once the unauthorized user determines the server's private key, he can decrypt all sensitive information on the license
5 server. Should all license servers use the same key, as is frequently done, then all the security of the applications served by all the license servers will be compromised.

The license access module 27 first translates a license token from a
10 vendor specific format to a format usable by the licensing library 24. The license access module accomplishes the translation in two modules. One module translates or decodes a license token from a vendor specific format to a licensing library format. The second module translates or encodes the updated license token from the licensing library format to the vendor specific format.
15 The second module is invoked anytime the licensing library updates the information in a license token.

Upon receiving the license token in the licensing library format, the licensing library invokes routines which verify the correctness of the license by
20 reviewing the following license information stored in the token: (1) flag, (2) maintenance contract date, (3) host name and domain, (4) product name, (5) host id number, (6) license serial number, and (7) expiration date of license. This is compared to the information maintained by the application. If the information matches, the license is verified. After completing the verification
25 process, a routine in the licensing library is initiated which checks out the license by decrementing the license units in license token by the number of licensing units being checked out.

The decoding and encoding routines allow software vendors to
30 implement their own security mechanism to protect their licenses from unauthorized use even though they reside at the customer's site.

Below is an example of a sample application using the licensing library and the license access module written in C language:

```

5  #define LIC_RENEWAL_TIME (60)           /set renewal time for this session/
   #define EST_LIC_RENEWAL_TIME (LIC_RENEWAL_TIME x .9)

   NL_vendor_id NL_Vendor_id = 1223;      /set vendor #/
   NL_prod_num NL_Prod_num = "02"         /set product #/
10  NL_version NL_Version = ( 12/20/88, "1.0" ); /set version id #/

   ...
   status = NL_init (vendor_id, NULL, &job_id); /initialize license service/
   if (status != NL_NO_ERROR) /accept job id if no error/
   {
15       fprintf (stderr, "nl_init failed - error =
           %d\n", status ); /error message if error and
                               return/
       return;
   }

20  units = 3;
   code_funcs.encode_p = nl_encode; /pointer to encode function/
   code_funcs.decode_p = nl_decode; /pointer to decode function/
   if (signal (SIGALRM), alarm_intr ) == (void *) -1 /set alarm if no
                                                       error/

25  {
       perror ("Cannot set SIGALRM"); /otherwise, error message/
       return;
   }

   status = NL_request (job_id, NL_Prod_num, /request a license/
30   &NL_Version,
   units, LIC_RENEWAL_TIME, NL_L2_SRCH,
   &code_funcs, NULL,
   &req_handle, NULL, &app_info);

   if (status != NL_NO_ERROR) /no error, license checked
35   { /out from license server/
       fprintf (stderr, "nl_request failed - error =
           %d\n", status); /otherwise, error message/
       return;
   }

40  /*
   * We got a license /license request successful/
   */

   alarm (EST_LIC_RENEWAL_TIME); /set alarm for license renewal
45   ... time/
   ... Application Runs /runs application/
   ...

   status = NL_release (req_handle); /request to release a license/
   if (status != NL_NO_ERROR)
50   {
       fprintf (stderr, "nl_release failed - error = /otherwise, error

```

```

        %d\n", status);
        return;
    }

5    ...
    int
    alarm_intr ()
    {
        status = NL_confirm (req_handle,    /renew licensing unit with
        LIC_RENEWAL_TIME, NULL);           licensing server/

10    /* Verify vendor private information
        */
        }

        If (status!= NL_NO_ERROR)
15    fprintf (stderr, "nl_confirm failed - error =    /otherwise, error
        %d\n", status);                      message/
        {
            puts ("license renewed")         /successful license
        }                                    renewal/

20

```

The sample application given above is accompanied by self-explanatory annotation to the right margin of the codes. Of particular interest are code_func.encode_p and code_func.decode_p. Encode_p and decode_p are pointers to the software vendor's encode and decode routines, respectively. Taking the pointers in the code_func variable, the licensing library can use the pointers to invoke the decoding and encoding routines in the license access module. The three major licensing library routines, request for a license (NL_request), release a license (NL_release) and renew a license (NL_confirm) invoke the decoding and encoding routines. For example of a license access module, see Appendix 1.

In implementing the license access module, the license server becomes merely a repository for license tokens. The licensing library coupled to the application performs the procedure of authenticating the license token prior to granting a license and therefore access to run the application.

Because the level of security of the system is dictated by the license access module, the software vendors are free to make the license access module as simple or as complex as they desire. In particular, they are free to

adopt any of the encryption schemes as part of their encryption routines. If the security mechanism is broken, and the encryption known to others, then the software vendors can easily remedy the situation by releasing a new version of the product with a new license access module.

5

While the present invention has been particularly described with reference to Figures 1-4 as well as Appendix 1, and with emphasis on certain language in implementing a method to protect against the unauthorized use of software application in a computer network environment, it should be

10 understood that they are for illustration only and should not be taken as limitation upon the invention. In addition, it is clear that the method of the present invention has utility in any application run in a computer network environment. It is contemplated that many changes and modifications may be made, by one skilled in the art, without departing from the spirit and scope of
15 the invention disclosed above.

CLAIMS

1. In a computer network environment including a plurality of software applications licensed to run on at least one network of agents, said applications located on said agents wherein use of the application on a particular agent is permitted upon the grant of a license, said license being requested by a user from said agent of said applications, a system for protecting against the unauthorized use of said applications comprising:

license token means for storing licensing information of said applications; license server means connected to said agents for communicating with said applications, said license server means having a database which stores said license token means, said license server means further retrieving said license token means from said database upon a request for a license by said applications, said license server means further transmitting said license token means to said applications;

license access means connected to said agents for decoding and encoding said license token means from said license server means, said license access means being integrated with said applications, said license access means receiving said license token means from said license server means; and

licensing library means connected to said agents for verifying said decoded license token means before access to said license is granted, said licensing library means being integrated with said applications.

2. The system as defined in claim 1, wherein each said license token means containing licensing information for at least one version of each said applications.

3. The system as defined in claim 1, wherein the contents of said license token means is encrypted.

4. The system as defined in claim 1, wherein said license token means is passed between said license server means and said licensing library means for a predetermined time period.

5. The license token means as defined in claim 4, wherein during said predetermined time period, only one said applications may check out one said license token means.

6. The system as defined in claim 1, wherein said license server means receives said request for a license from said applications, said license server searches in said database for a license token means storing the license requested by said application before retrieving said license token means.

7. The system as defined in claim 1, wherein said license access means decodes the contents of said license token means before said licensing library means verifies said license token means.

8. The system as defined in claim 1, wherein said license access means encodes said license token means after said licensing library verifies said license token means and prior to returning said license token means to said license server means.

9. The system as defined in claim 1, wherein said licensing library verifies said license token means by

comparing the licensing information stored in said license token means with the licensing information maintained by said application.

10. The system as defined in claim 1, wherein said licensing library means checks out said license of said application in response to a positive comparison of the license information.

11. The licensing library means as defined in claim 10, wherein said license for said application being checked out after said licensing library verifies said license token means.

12. In a computer network environment including a plurality of software applications licensed to run on at least one network of agents, said applications located on said agents wherein use of the application on a particular agent is permitted upon the grant of a license, said license being requested by a user from said agent of said applications, a system for protecting against the unauthorized use of said applications comprising:

license token means for storing licensing information of said applications;

license server means connected to said agents for communicating with said applications, said license server means having a database which stores said license token means, said license server means further retrieving said license token means from said database upon a request for a license by said applications, said license server means further transmitting said license token means to said applications;

license access means connected to said application and accessible from said agents for decoding and encoding said license token means from said license server means, said license access means being integrated with said applications;

licensing library means connected to said application and accessible from said agents for verifying said decoded license token means before access to said license is granted, said licensing library means being integrated with said applications; and

license binding means connected to said license server means and to said licensing library means for constructing a binding file, said binding file informing said licensing library means which of said license server means may grant a license to said application.

13. The system as defined in claim 12, wherein said licensing library means are located on the same agents as said applications.

14. The system as defined in claim 12, wherein said license sever means are located on the same agents as said licensing library means.

15. The system as defined in claim 12, wherein each said license token means contains licensing information for at least one version of each of said applications.

16. The system as defined in claim 12, wherein the contents of said license means is encrypted.

17. The system as defined in claim 12, wherein said license token means is passed between said license server

means and said licensing library means for a predetermined time period.

18. The license token means as defined in claim 17, wherein, during said predetermined time period, only one of said applications may check out one said license token means.

19. The system as defined in claim 12, wherein said license server means further transmit said license token means to said licensing library means.

20. The system as defined in claim 12, wherein said license access means decodes the contents of said license token means before said licensing library means verifies said license token means.

21. The system as defined in claim 12, wherein said license access means encodes said license token means after said licensing library verifies said license token means and prior to returning said license token means to said license server means.

22. The system as defined in claim 12, wherein said license binding means constructs said binding file by contracting each said license server means to request for a list of applications it serves, said binding file containing said list of applications available from said license server means.

23. In a computer network environment including a plurality of software applications licensed to run on at least one network of agents, said applications located on

said agents wherein use of the application on a particular agent is permitted upon the grant of a license, said license being requested by a user from said agent of said applications, a system for protecting against the unauthorized use of said applications substantially as hereinbefore described with reference to the accompanying drawings.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
30 August 2001 (30.08.2001)

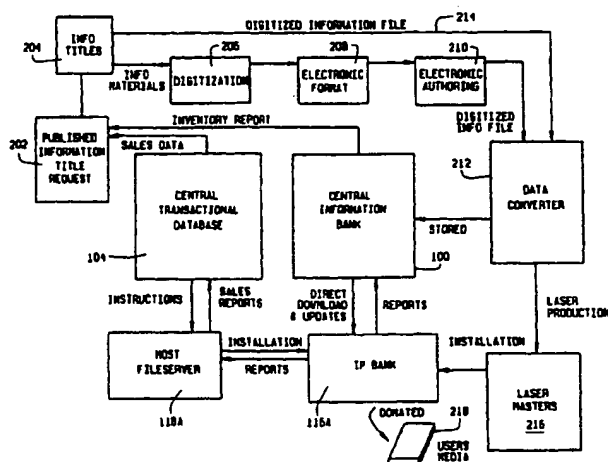
PCT

(10) International Publication Number
WO 01/63528 A1

- (51) International Patent Classification⁷: **G06F 17/60** (74) Agents: **BEULICK, John, S. et al.**; Armstrong Teasdale LLP, Suite 2600, One Metropolitan Square, St. Louis, MO 63102 (US).
- (21) International Application Number: **PCT/US01/05706**
- (22) International Filing Date: 22 February 2001 (22.02.2001)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
09/511,537 23 February 2000 (23.02.2000) US
- (71) Applicant (for all designated States except US): **IPDN CORPORATION** [US/US]; 104 E. Main Street, DuQuoin, IL 62832 (US).
- (81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.
- (84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **SAIGH, Michael, M.** [US/US]; 535 East Main Street, DuQuoin, IL 62832 (US). **BARRETTE, Pierre, Philip** [US/US]; 662 Lake Shore Drive, Murphysboro, IL 62966 (US).
- Published:
— with international search report
— before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments

[Continued on next page]

(54) Title: **METHODS AND DEVICES FOR STORING, DISTRIBUTING, AND ACCESSING INTELLECTUAL PROPERTY IN DIGITAL FORM**



(57) Abstract: In one embodiment, the present invention is an apparatus for facilitating obtaining text of an IP, the apparatus including a storage device (116A) having stored therein text of a plurality of IP, wherein the text includes electronically stored representations of printed text works, movies, films, video presentations, television programming, music, audio works, audio presentations, radio programs, graphic material, art works, plays, operas, novels, writings, photographs, pictures, images (including electronic images of all types including virtual images), advertising copy, or software, or portions and combinations thereof; a processor (118A) connected to the storage device (116A), the storage device (116A) further having stored therein a program for controlling the processor to: receive an IP selection request; receive a user identification associated with the IP selection request; and output encrypted text of the selected IP if the user identification and IP selection are valid utilizing a determined level of IP encryption.



For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHODS AND DEVICES FOR STORING,
DISTRIBUTING, AND ACCESSING
INTELLECTUAL PROPERTY IN DIGITAL FORM

CROSS REFERENCE TO RELATED APPLICATIONS

5 This application is a continuation-in-part of U.S. patent application
serial no. 09/175,559, filed October 20, 1998, which is a continuation-in-part of U.S.
patent application serial no. 09/049,321, filed March 27, 1998, which is a continuation
of U.S. patent application serial no. 08/687,292, filed July 25, 1996 (now U.S. Patent
No. 5,734,823, issued March 31, 1998), which is a continuation of U.S. patent
application serial no. 08/367,056, filed December 30, 1994 (now abandoned), which
10 is a continuation-in-part of U.S. patent application serial no. 08/296,120, filed August
25, 1994 (now abandoned), which is a continuation of U.S. patent application serial
no. 07/787,536, filed November 4, 1991 (now abandoned), all of which are
incorporated herein by reference.

BACKGROUND OF THE INVENTION

15 The present invention relates to methods and apparatus for the secure
consolidated electronic storage and distribution of Intellectual Property in digital form
and more particularly to methods and apparatus for electronically storing and
transmitting information in consolidated, digital form to and from centralized storage
facilities and users.

20 As used herein, "Intellectual Property" ("IP") refers to that, which in
non-electronic form, would be referred to as books, text, pictures, photographs,
videos, movies, films, audio, music, software, computer games, video games, and
other types of expressions of ideas or concepts that can be stored and distributed in
digital form. IP also includes all material that is capable of being protected by
copyright, as well as other materials. In some embodiments, IP's include

compilations and/or portions of one or more of the above types of works as well as the works themselves.

Current networks for dissemination of IP lack a uniform distribution system. There are, for example, several different encryption technologies providing digital security (i.e., prevention of piracy or illegal copying of IP) for different media such as books, video, software, and multi-media. Similarly, there are different network distribution systems competing with each other to provide IP with little or no compatibility between the systems as seen from the standpoint of users, publishers (i.e., content providers), or the IP itself. Each entity has its own web site, data center, and encryption algorithm. Uniformity is lacking because many different publishers, studios, and intellectual property owners have their own information retrieval service. Present methods and systems available for electronically distributing IP are not believed to provide a sufficient level of security against unauthorized use and copying to be acceptable vehicles for the commercial distribution of IP. This fragmentation produces inefficiencies in the distribution of intellectual property. There is currently no operator of an overall distribution network capable of providing publishers assurance of IP protection and easy-to-use, efficient, inexpensive, and universal dissemination of IP. There are currently no standards available that establish a criteria for the development and operation of the infrastructure, networking and systems for the commercial electronic delivery of IP with a level of security that would be desirable to most commercial providers of IP.

Traditionally, each type of IP has had its own unique way of being provided to the consumer. Intellectual Property (hereinafter referred to as IP) have been printed on paper and in books, for example, and music has been "burned" onto CDs. Retrieval methods for obtaining IP via digital readers, MP3 music hardware players and other hardware are currently specialized for the particular forms of IP for which the reader, player or other hardware is adapted. This form of specialization further fragments the IP market into specialized content data centers and reduces efficiency in distribution while failing to provide a level of security against

redistribution and copying that would be desirable to most commercial providers of IP. In addition, a proliferation of web-sites and market access locations make it more difficult and confusing for a consumer to access the available choices in a meaningful manner. Furthermore, a lack of uniformity in methods for identifying particular
5 digitally-formatted IP makes it more difficult to track the flow of use of IP, because multiple tracking methods must be employed to accommodate multiple methods of identification. However, information provided in each case can be reduced, ultimately, to binary information. It would therefore be desirable to provide a distribution network designed for distributing various different IP types and a
10 distribution method that can securely deliver each of these various IP types to the consumer, while safeguarding the transmitted IP types from illegal reproduction.

Present methods for distributing IP rely upon straight-forward transfer of data without inherent security built into the transferred data files. These methods provided adequate protection before the proliferation of public networks and the
15 Internet. However, if one person were to "crack" a security code, any or all of the IP distributed in a file would be available for unencrypted transfer for all to have and use without regard for licensing and ownership. It would therefore be desirable to provide methods and devices for securing transferred data files that avoid making the IP available for unencrypted transfer after a code is cracked. Further, it would be
20 beneficial to provide a distribution system and mechanism that regularly and automatically updates and/or changes encryption and decryption algorithms, keys, and/or formulae applicable to each IP electronically distributed to recapture security features for any IP for which previous encryption and decryption algorithms, keys, and/or formulae have been broken or compromised, to thereby inhibit cracking or
25 compromising of IP encryption and decryption mechanisms.

Furthermore, the present proliferation of web sites and the Internet increases the difficulty of the user in finding and accessing specific IP's. There is currently a lack of uniformity in methods of identifying particular IP in digital form. This lack of uniformity makes it more difficult to track the flow and use of IP because

multiple tracking methods are required to accommodate multiple methods of identification. Lack of uniformity in electronic storage and distribution of IP also make distribution and delivery of IP more expensive and less efficient. Moreover, users of IP are inconvenienced because they must acquire multiple devices or software programs to read many different IP data formats.

It would therefore be desirable to provide methods and apparatus for consolidating all forms of digitally distributable IP.

It would further be desirable to provide methods and apparatus that provide a consistent interface between IP organizations and content owners on the one hand, and consumers of IP on the other.

BRIEF SUMMARY OF THE INVENTION

In one embodiment, the present invention is an apparatus for facilitating obtaining text of an IP, the apparatus including a storage device having stored therein text of a plurality of IP, wherein the text includes electronically stored representations of printed text works, movies, films, video presentations, television programming, music, audio works, audio presentations, radio programs, graphic material, art works, plays, operas, novels, writings, photographs, pictures, images, advertising copy, computer games, video games, or software, or portions and combinations thereof; a processor connected to the storage device, the storage device further having stored therein a program for controlling the processor to: receive an IP selection request; receive a user identification associated with the IP selection request; and output encrypted text of the selected IP if the user identification and IP selection are valid utilizing a determined level of IP encryption.

As is explained below, the terms "text" and "IP" as used herein are to be interpreted broadly. For example, a "IP" that is a musical recording includes "text" that is audio, or more specifically, music. Other types of "IP" may include more than one type of "text."

This embodiment of the present invention consolidates various forms of digitally distributable IP and provides a consistent interface between IP organizations and content owners on the one hand, and consumers of IP on the other.

BRIEF DESCRIPTION OF THE DRAWINGS

5 Fig. 1 illustrates one embodiment of the present information distribution system architecture;

Fig. 2 illustrates information flow in the system architecture shown in Fig. 1;

Fig. 3 is a block diagram illustration of a point of purchase delivery system;

10 Fig. 4 is a more detailed block diagram illustration of the host files server shown in Fig. 3;

Fig. 5 is a perspective view of an IP Bank embodiment;

Fig. 5A is an alternative IP Bank embodiment;

Fig. 5B is yet another alternative IP Bank embodiment;

15 Fig. 6 is a block diagram illustration of the IP Bank circuitry;

Fig. 7 is a block diagram illustration of an end user's storage medium;

Fig. 8 illustrates the information flow for the point of purchase configuration;

20 Fig. 9 is a block diagram illustration of certain elements of a point of rental delivery system;

Fig. 10 is a block diagram illustration of certain elements of an IP Bank subsystem;

Fig. 11 is a block diagram illustration of certain elements of a promotional delivery system; and

Fig. 12 is a flow chart illustrating the encryption process implemented in accordance with the present invention; and

5 Fig. 13 is another IP Bank embodiment.

Fig. 14 is a representation of a network showing, in one embodiment of the present invention, how IP from content suppliers is provided through a central data center to a user device.

10 Fig. 15 is a more detailed block diagram of the user device shown in Fig. 14.

Fig. 16 is a block diagram of the user device shown in Fig. 15 to which an external authentication device is attached.

DETAILED DESCRIPTION OF THE INVENTION

As used herein, the term "network" refers to any electronic interconnection between two or more electronic devices over which data is transferred, including, but not limited to, the Internet, an intranet, a land line or
15 traditional telephone network, a cellular or wireless mobile network, a wireless microwave network, television or radio wave transmissions, a cable network, a wireless connection (for example, infrared or microwave connections), satellite, a localized land network system, induction connection using electric lines, a wireless
20 network using lasers as the transmitting medium, any combination of any of the preceding or any other system for the transmission of data between two or more units. A "secure network" is a network employing security measures against unauthorized access to data being transmitted via the network or data stored within a memory storage area of a device connected to the secure network.

The term "IP Bank," as used herein, refers to an interface between a network and a user. Such interface may be physically located in proximity to a central information storage facility, at a location remote from a central information storage facility (for example, in a physical housing such as a kiosk located at a retail establishment), or in proximity to and directly connected to a user's computer. In one embodiment of the invention, the IP Bank is wholly or partially a virtual device generated from an interaction a software program stored on a user's computer and software located at another IP Bank or a central information storage facility. In embodiments involving a virtual device, some of the hardware necessary for operation of an IP Bank is located proximate to or within a user's computer, or proximate to or within the IP Bank or the central information storage facility, depending upon a location at which user contact is made. For example, in one embodiment, a "IP Bank" comprises memory storage, a processing unit, a keyboard, slots for credit cards, slots for storage media on which downloaded digital data such as electronic versions of printed IP texts are stored, and a printer (for bills and receipts). A virtual "IP Bank" comprises, for example, only a memory and a processor (located at, e.g., a retail establishment, a data storage center, or at a security encryption compression module. Keyboards, slots for credit cards and storage media, etc. are supplied in this embodiment by the user's device and the user's software. Although the term "IP Bank" and "text" may suggest book-type or written material based upon a reader's prior notions of the subject matter to which these terms refer, the terms are not intended to be so limited in this description. Other types of material, such as movies, music, voice, video, graphic images, natural language and other text, audio, computer games, video games computer software material, and any other type of intellectual property capable of being converted to and stored in digital form is intended to be included when the terms "IP" and "text" are used, unless otherwise specified or made clearly obvious from context. (For example, "natural language text" does not refer to video or music.) An IP Bank is a self-service, user interactive information vending device, which, in one embodiment, is a separate, stand-alone device. In another embodiment, an IP bank includes hardware and software located at a central data

storage facility and hardware and software located within or proximate to a user's computer, with both being in electronic communication via a suitable network, thereby providing operation coordinated in a manner such that the components function in the same manner as a stand-alone IP bank. In yet another embodiment, an IP bank is a device located proximate to a user's computer or a computer board located within the user's computer and which is electronically connected to the user's computer via a suitable network connection, and which is also connectable to a central information storage facility using a suitable network. In another embodiment of the present invention, for example, each IP bank contains a high capacity, local memory storage having a customized portfolio of the most demanded information products for a particular site at which the IP Bank is located. In another embodiment, each IP Bank comprises a processor, a monitor, a network connection, and one or more slots configured for insertion of a portable memory storage medium and/or a connection to a user device, for example, a portable digital assistant (PDA). Other information is transferred via a network to an IP Bank for supplemental, secondary, and less demanded purposes. A processing unit within the IP Bank and coupled to the IP Bank local memory and storage controls downloading and dynamic encryption of information.

The term "IP," as used herein, includes all different types of intellectual property that is capable of being electronically stored in digital form. The term "IP" includes traditional printed text works, movies, films, video presentations, television programming, music, audio works or presentations, radio programming, graphic material, art works, plays, operas, novels, writings, photographs, pictures, images, advertising copy, computer games, video games, computer software, any portion of combination of the above, and/or other types of intellectual property.

The term "IP file," as used herein, refers to a digital version of an IP. The description is not dependent upon the format of the stored digitized material, and is equally applicable whether the digitized material is stored using HTML, XML, Adobe PDF (portable document format), SHOCKWAVE® format (Macromedia, Inc.,

San Francisco, CA), or any other format. Where the context so indicates, the term "IP selection request" refers to a request made by a user to obtain a copy of a given IP file. The term "selected IP," as used herein, refers to a particular IP file for which a user has made a request for a copy. The term "text," as used herein, when used in connection with a selected IP or IP file, refers to the content of the digitally stored IP file under consideration. Because the term "IP" is used herein includes not only traditional printed text works, but also other types of intellectual property capable of being electronically stored in digital form, the term "text" as used herein is intended to be interpreted with equal generality so as to include the content of the "IP," whether it be traditional printed text or the content of another type of IP to which the general term "IP" is applied herein. The general applicability of various of the embodiments of the invention disclosed herein will be appreciated if it is understood, for example, that a "IP" that is a television program includes "text" that is television programming. As another example, a "IP" that is a musical recording includes "text" that is audio, or more specifically, music. Extending the analogy further, it is easily understood that one "reads" the "text" of the musical recording "IP" by playing the music via a system that converts the digitally encoded audio into sound. Some types of "IP's" may include more than one type of "text," for example, digital representations of printed text, music, and pictures. All uses herein of the terms "IP" and "text" are intended to be generalized in a similar manner, unless it is explicitly stated that such use is intended to refer only to printed text or natural language text, or the context is such that the term "text" is obviously limited thereto.

The term "user's computer" or "user's device," as used herein, refers to any electronic device performing some or all of the functions traditionally associated with a typical desktop computer, including, but not limited to, a traditional desktop computer (such as an IBM PC, a MacIntosh® (Apple Computer, Inc., Cupertino, CA) or a clone of either), a laptop computer, PDA device, wireless connecting device, Internet connecting device (e.g., WEBTV™ (WebTV Networks, Inc., Palo Alto, CA) or others), digital telephone, video gaming device, ebook, or another electronic device capable of being electronically connected with a network.

The terms "user's computer" and "user's device" do not necessarily refer to different types of units.

As used herein, the term "nonvolatile memory" refers generally to a type of memory that does not depend upon power being continuously applied to retain information. For example, static RAM, segregated hard drive, floppy disk, CD-RW, magnetic optical disk, flash memory, electronically erasable programmable read only memory (EEPROM), and WORM (write once, read many time) memories are types of nonvolatile memory. ROM (read only memory) and field programmable gate arrays are also types of nonvolatile memory that are sometimes mask-programmed. A circuit designer would be able to choose a suitable type of nonvolatile memory for use in a circuit, giving due consideration to voltage, current, data access rate, data transfer rate, physical size of the device desired, and memory density requirements, and to whether the application requires the nonvolatile memory to be field programmable, erasable, and/or reprogrammable.

Also as used herein, the term "encryption algorithm" unless otherwise stated, includes methods for encryption, keys used for encryption, and formulae used for encryption. The term "decryption algorithm," unless otherwise stated, includes methods for decryption, keys used for decryption, and formulae used for decryption. However, keys and/or formulae are sometimes explicitly mentioned along with the algorithms in the explanation to add emphasis.

In one embodiment, a user may select portions or entire contents of one or more IPs. The selected information is then combined and downloaded to the user's storage device, for example, a "cartridge." A "cartridge," "memory module," or "storage medium," as used herein, refers to any device capable of electronic storage of digitized data and which is capable of providing access thereto in a controlled manner. The terms "cartridge," "memory module," and "storage medium" are interchangeable, unless otherwise indicated by context. The terms encompass, for example, devices such as appropriately configured compact disks (CDs), DVDs

(including DVD-R, DVD-RAM and DVD-RW), flash memory cards, and removable storage devices (including floppy disks, memory chips, ZIP™ disks, and other units), where “appropriately configured” means, for example, having instructions or data recorded thereon to control access by an access device to data recorded thereon. Also included are protected segregated portions of a data storage portion of a hard disk, and any other device for the storage of data electronically in digital form and providing controlled access thereto. In one embodiment, a cartridge includes a unique identification number and a predetermined amount of memory for storing the selected information. In another embodiment, a cartridge also includes specialized software (cartridge controlling software, or “CCS”) stored in nonvolatile memory, either on a separate memory chip of the cartridge, or in a segregated protected space within a general memory storage space of the cartridge. The CCS is electronically linked to the cartridge on which it is stored, and controls and regulates reading and use of encrypted and tagged data information files stored within the general memory of the cartridge. As long as any portion of encrypted data stored on the cartridge is being accessed, CCS remains functioning and regulates and controls functions that the user is permitted to perform with regard to encrypted data in question. For example, in most cases, when an encrypted data file is accessed for reading, CCS operates to restrict the ability of the user to copy the data in a decrypted form or to print data onto hard copy. CCS also restricts the user from copying any portion of the decrypted data to any data file other than a temporary file in RAM that is automatically erased when CCS stops operating as an encrypted file that is only readable using CCS operating on authorized equipment. When the user ceases accessing encrypted data files, and CCS ceases operation, just prior to closing down, CCS totally erases all record of any decrypted data and all temporary files in which such data may have been stored. Once CCS has ceased operation, the user is no longer able to access any of the temporary files generated during previous operation, as all record of such files ever existing and the content contained therein will have been permanently excised from the user’s system. Thus, in connection with related application software stored within the user’s device, CCS utilizes one or more of the dynamic encryption and decryption

features, a unique serial or registration number, and various data registration headers to regulate and govern any use the user makes of encrypted files and the data stored within. Without CCS operating, a user is unable to gain access to the encrypted data. In one embodiment of the invention, CCS contains monitoring features that prevent
5 operation of CSS should any attempt be made to alter its operation. In one embodiment, when CCS becomes inoperable for any reason, the user of the device has to bring the affected cartridge to an authorized agent for repair or replacement.

In one exemplary embodiment of CCS, CCS finds the information necessary to decrypt an encrypted text. While the user device in which CCS resides is
10 operating, CCS restricts what the user device is capable of doing. For example, it restricts the user device so that it cannot write information to an external storage device while an internal storage device having an encrypted text is being used. When a cartridge is disengaged from the user device by removal, CCS shuts down the user device, for example, either by shutting down its program or by operating an electronic
15 switch to remove power from the user device. CCS also removes all temporary files related to the program from the user device. In addition, CCS reads a secure real-time clock in the user device to determine whether the user device is presently authorized to access data on a cartridge. CCS also registers when tampering occurs, such as a change in the real-time clock data caused by an obvious backdating attempt.

20 One embodiment of the invention enables a user to obtain updates to any data information file he or she has acquired, by utilizing a dial-up network to dial into an IP Bank or a central data storage facility, or by using the Internet to access an IP Bank or a central data storage facility through an appropriate link via a web site, or the use of a wireless network (e.g., digital satellite, cellular, wireless mobile,
25 microwave, infrared, etc.) to gain access to an IP Bank or the central data storage facility over any network or connection. In one embodiment, a Secure Universal Resource Locator (SURL) is used, including both a secure phone number as well as an Internet-based URL. The same restrictions apply to obtaining an update as to acquire the data information file being updated.

The following sections provide a brief overview of one embodiment and a detailed description of its architecture. Following the detailed architecture description is a detailed description of point of sale delivery embodiment configurations. A detailed description of the various levels of encryption which may
5 be used in the present system is then provided.

A. Brief Overview

In accordance with one embodiment of the present invention, information is distributed from a central information bank to a user's personalized storage medium. Information to be so distributed by the present system is received
10 from outside sources either electronically, over various communication networks (e.g., telephone lines, cable systems, cellular systems, wireless mobile systems or other similar commercial communication networks) or from various storage mediums (e.g., magnetic or electronic disks, cartridges, or tape reels or compact disks, laser disks, tape cassettes, etc.), or in hard copy format. If information is received in a hard
15 copy format, it is initially converted to a standard digital format (e.g., ASCII text, DOS text or other similar standard commercially available text format) by scanning or direct transcription. If information is received in a videotape, NTSC or PAL format video, then the information is digitally encoded in a ".avi" file, a Quicktime file, or other format, including the application of appropriate MPEG-X compression as
20 needed. A content provider, i.e., an outside source, providing information can specify instructions relating to authorized use, access, and cost of the provided information that can be permanently linked to a master data file of the information when it is electronically stored or accessed. In one embodiment, instructions are electronically generated when information is received electronically. These instructions include a
25 description of allowed uses (e.g., copy generation, printing limits and/or authorization, rental options, if any, purchase options, etc.), desired level of security against unauthorized copying or use (from level zero [virtually no security] to a maximum level [most robust security available]) and cost to a user, optionally as a function of user-selected access right. Whenever a request is made to obtain a copy of

the information file (i.e., IP File), the instructions are used in connection with the choices made by the user, within the allowed options, to generate user-specific electronic instructions for limiting use of the requested copy of the IP File. Then the information is digitized, formatted, compressed and initially encrypted to form an electronic master copy which is stored in the central information bank. The master copy is duplicated electronically and dispatched electronically through a communication network, such as a telephone or satellite network, to a point-of-sale delivery system. IP Banks form a part of such a delivery system, and the electronic copies are retained in the IP Banks for downloading into a user's personalized storage medium. Initially, a user selects the information to be downloaded and a tracking entry is made into a transactional database to record the transfer. Prior to and during downloading of the copy on the user's storage medium, the information is dynamically encrypted utilizing a varying level of encryption which is dependent upon a variety of variables, for example, an economic value of the information. A "dynamic" encryption process is utilized so that only the electronic reader associated with the user card used to access the information from the IP Bank and download the information to the user storage cartridge can be utilized to display the information in an understandable text format. In one embodiment, the "dynamic" encryption process creates, encrypts, and transmits data files, in real time, to one or more different user devices or storage media located in one or more different physical locations using a network. For example, in one embodiment, an instructor's lecture notes are electronically encrypted and transmitted in a secure manner to devices and/or storage media of students attending a lecture in real time. The notes are stored on these devices and/or storage media for current or later use.

As explained in greater detail below, in one embodiment, the dynamic encryption process uses a unique serial number (or other identification, for example, an alphanumeric or binary identification) associated with the particular user's storage medium to which the encrypted data is to be downloaded, the unique registration number given the user on registration, and the unique file number associated with the file to be downloaded, to secretly generate the encryption and decryption algorithms

and formulae to uniquely encrypt the copy of the data file as it is downloaded to the user's storage medium and to allow the user's system to decrypt the data for later reading. A copy of the algorithms and formulae are secretly stored at the central data storage facility in a portion of the memory storage having limited access. A copy of the algorithms and formula necessary to decrypt each of the acquired encrypted data files is stored in read-only form on the user's storage medium. Storage space for this read-only copy is designed so that any attempt to access the data stored thereon other than using an unaltered CSS program renders the data stored thereon unreadable. In another embodiment, instead of utilizing a unique registration number permanently tied to the storage medium being used for developing the unique encryption and decryption algorithms and formulae, a unique electronically-generated electronic number is assigned to the storage medium at the time of initial registration with the system, and that assigned number is used in developing the applicable encryption and decryption formulae and algorithms. In this embodiment, the unique electronically-generated number is a number that is randomly generated by the system and checked for uniqueness. In another embodiment, the unique electronically-generated number is generated by a system program that searches features of the user data storage medium to find a unique representation of the device, from which a unique electronic registration number is developed. In other embodiments, other methods, including combinations of these methods, are used to generate unique numbers.

In another embodiment, software programming periodically changes the applicable encryption and decryption algorithms and formulae associated with a particular encrypted IP file to provide enhanced security. In this embodiment, the changes in the applicable encryption and decryption algorithms and formulae occur when a user logs on and connects with the system through an IP Bank connected to the central information storage facility. The changes occur after expiration of a predetermined period of time from a date on which the particular encrypted IP file is initially created, or after expiration of a predetermined period of time from a date on which the applicable encryption and decryption algorithms and formulae are last changed, whichever last occurs. The periodic changing of applicable encryption and

decryption algorithms and formulae makes it more difficult for the encryption to be broken and allows recapturing of IP files into the encryption system if previous encryption has been broker. The frequency of such encryption changes is dependent upon a time value of the IP involved and a length of time that a user of the IP file in question has been authorized to access the particular encrypted IP file.

In one embodiment, security is enhanced through the use of an external authorization device affixed or linked to the reader device and a related identifier. Examples of suitable, currently-available authentication devices include parallel port software locks, iButtons™ (Dallas Semiconductor Corp., Dallas, TX), and "Smart Cards" plugged in via a pc-card.

In another embodiment of the invention, when one or more IP Banks are operated in conjunction with one or more retail establishments, each retail establishment is given a unique identification number that is used to tag each user device, user storage medium and/or encrypted IP file acquired at the retail establishment or through an IP Bank linked to the retail establishment. Using the tag information, particular user devices, storage media and/or IP files, a predetermined portion of the system revenue generated in regard to such tagged user devices, storage media and/or IP files is allocated to and shared with the retail establishment in question. In another embodiment, the tagging is further delineated to allow revenue sharing with manufacturers and distributors of user devices and/or storage media.

IP digital data transmission can occur between two or more than two mobile wireless or cellular devices. The devices can be multipurpose, for example, they may have a combination of PDA, cell phone, visual display, and/or audio capabilities. The devices transmit IP in a secure manner whether the IP is digital audio, video, text, software, or multimedia. The mobile device can also asynchronously obtain, through a wireless access portal (WAP), music, video, text, and other IP through various databases. For example, if a user wishes to access his or her favorite songs, the mobile wireless browser can be used to browse the music

database, access a music title by artist, CD title, or by some other fashion and customize the desired transfer of music IP. An actual order and payment then takes place after the user requests whether they wish to rent (and thus, autoerase) or own a copy of a requested song. A similar mechanism is used, in one embodiment, for
5 ordering natural language texts (e.g., an "ebook"), software games, video and movies. Asynchronous transfer refers to the downloading of IP from one or more than one database using the wireless mobile device. The actual transfer of data takes place and is stored in a customer's SEC (security encryption compression) device. A user could utilize the IP at his or her discretion as opposed to real-time usage with a data stream
10 transfer.

In one embodiment, multiple databases and portals with transparent links can be used and accessed with the mobile wireless device. It is immaterial to the practice of this embodiment of the present invention whether access is through mobile wireless or cable, fiber optics, telephone access to an Internet portal or database. The
15 same end-to-end security as described in this invention for IP is consistent with the invention. The same dynamic encryption and mechanical and/or technical means exists throughout the data transfer distribution cycle. After the transfer occurs, the user (i.e., customer) SEC unit contains data and rules linked to data governed by the "smart" storage device. The smart storage device determines the time of autoerasure,
20 watermarking, level of security from the highest strata to a more open system and other rules directly linked to that data or piece of the IP data. The rules can apply as minutely as a note, word or bit, etc.

Synchronous data transfer refers to a continuous stream of IP data to the mobile cellular device. Here, the same rules for the IP exist in real time and the
25 wireless connection must be maintained throughout its transfer. A combination of synchronous and asynchronous transfer could exist within the database and multipurpose consumer device. A single purpose device, for example, one providing mobile wireless access to an internet music data portal without other functional abilities is also within the scope of the present invention.

In one embodiment, the wireless device can also be linked to other devices synchronously or asynchronously. For example, data transferred on the mobile wireless device after downloading a music IP could be transferred to a car or home stereo system, when music (audio) is played. Various transfer mechanisms can be used for the transfer of music or other IP. One embodiment of the present invention includes the transfer of IP through radio frequency to a car stereo or a home stereo, and the transfer of movies and television programs through radio frequency transfers to television sets and/or computers. In the case of a mobile wireless device such as a car stereo, the mobile wireless device is equipped with radio circuitry that can interface with the car stereo's AM or FM frequency. Transfers to this circuitry can be either synchronous or asynchronous. In a similar manner, the mobile wireless device could access other forms of IP, such as video, movies, software, text, and multimedia, etc., and transfer the IP in an encrypted manner, digitally secured as stipulated by a content owner's rules regarding the IP.

Digital content security can be produced and distributed in real time "on the fly." Professors, musicians, authors, instructors, film producers and any individual or organization or corporation can secure "their" content from redistribution when producing the content in a live setting or to be released on the web or other method of business-to-business or business-to-consumer digital distribution. The scope and quantity of content can be a word, note, picture frame based on DOI (Digital Object Identifier) or a complete or part of an entire work of art or content. Course packs present a classic example of bits and pieces of information gathering. The scope could also mean "on the fly content."

In one embodiment, rules formulated by content providers are matched with technological capabilities. The rules applied toward a certain content will constantly adjust in accordance to market demands, functionality and use of content as well as with technological advancement. The rules relative to a DOI, text, audio, visual, or content form will be tied to the content in a secured manner.

Digital content rules will be assigned with digital security to each and all content in any form (text, music, video, software, multimedia.) Such rules will be established by content owners and providers and will not be alterable or capable of manipulation by digital content users.

5 Functionality is defined as how content is allowed to be used. For example, the content (i.e., "text") can be autoerased, allowed to be printed or transferred to another user (or not), watermarked, and a host of rules applied governing activity, use, and manipulation of the content. The content can include natural language text, music, video, software, or multimedia. The rules regarding
10 digital content and security also includes the tracking, accounting and verification of transactions.

Consolidation of intellectual property for content providers can exist in one master database or several databases or subsystems. Transparent links to outside
15 databases to reserve intellectual property are used in one embodiment of the present invention to provide a customer with greater depth while giving content providers a sense of database control and management. Security, uniformity and integration are maintained throughout the system's network distribution chain. Customers may not know the location from which content is ultimately accessed. This seamless transfer greatly enhances content-driven access and continuity.

20 Since intellectual property content is fragmented into thousands of providers, content continuity and access is important to help mediate content sourcing and availability. A smart browsing and retrieval system that will, in effect, assist a consumer in accessing information by title, subject, author, artist, studio, publisher, etc., is built into the platform to be standardized. Embodiments of the present
25 invention develop systems integration and an active digital distribution database that apply security access rules if needed, and then redistribute the information to business users, household users, and/or organizational users.

Customization, business users, household consumers, etc., can, if content providers allow via rules applied to the content, customize their content. This customization of content can extend to any form of digital intellectual property and polarity. In one embodiment, customers or content providers and integrators can add
5 music to text material, animation, software, pictures, video, or multimedia. Cumulative printing or release play - music, video, etc. - is determined by content owners percentage of total content. A certain percentage of a song could be taken out of security, for example.

Advertising and promotions can be used to supplement supplier's
10 income from IP and/or reduce the cost to customers. Content owners will decide, as will customers, to what extent advertisement will be embedded or linked to content, and thus, how much income will be supplemented and/or cost reduced.

Because of content security and use restrictions, digital content returns are made possible.

15 In one embodiment, autoerasure is provided based on a secured, unalterable (by a user or customer) calendar clock. The clock can be reset when the customer retrieves information via wireless, internet, or other means. Autoerasure rules include the elimination of title (i.e., file) access when time is expired. A customer could then reactivate the affected title access for an additional period of
20 time. If a customer does not choose to renew a rental IP, the content of the IP is totally removed from the system.

B. System Architecture

Figure 1 illustrates one embodiment of the present information
distribution system. The system is shown, for illustration purposes only, as being
25 implemented across the world. Referring to Figures 1 and 2, one of the facilities provided is one or more "central data centers" or "central information banks" 100. Each central data center 100 acts in conjunction with any others present in the

network to store and control delivery of IP. Additional data centers 100 each comprising one or more computers acting in conjunction with each other are provided. A central information bank 100 is a central "library," or storage location, for information. Peripheral information banks 102A-F, coupled to central information bank 100, are libraries, or storage locations, for community oriented information. For example, the information stored in central information bank 100 accessed most often from the San Francisco bay area peripheral information bank 102A may not be accessed often from the peripheral information bank for Rome, Italy 102E. In any event, central information bank 100 is coupled to each peripheral information bank 102A-F to enable sharing of information. As explained in more detail hereinafter with respect to peripheral information bank 102F, each peripheral information bank 102A-F is coupled to one or more point-of-sale sites.

A central transactional database 104 coupled to the central information bank 100 and the peripheral information banks 102A-F, serves a central record keeping function for central information bank 100 and peripheral information banks 102A-F. Central information bank 100 and central transactional database 104 preferably, are commercially available main frame computers, such as an IBM main frame computer. The particular main frame model selected depends on the amount of information to be centrally stored in the network, the extent of record keeping functions to be performed, and the speed at which transfer and processing of information is to occur. Importantly, the present invention is not limited to any one particular computer to serve as the central information bank and/or the central transactional database.

As shown in Figure 1 is an exploded view of the various couplings between central information bank 100 and transactional database 104, peripheral information bank 102F and various point-of-sale delivery sites, particularly, point of purchase sites 108A-C, point of rental sites 110A-D, promotional sites 112A-D, and IP Bank subsystem sites 114A-C. Each point of purchase site 108 includes a point of purchase transactional database, represented by a box, and a user interface,

represented by a circle. As explained above, the user interface is sometimes referred to herein as the "IP Bank." Specifically, point of purchase site 108A contains IP Bank 116A and transactional database 118A, site 108B contains IP Bank 116B and transactional database 118B, and site 108C contains IP Bank 116C and transactional database 118C. Since the central information bank 100 and peripheral information bank 102F, and specifically peripheral information bank memory storage unit 106A, also could serve as IP Banks, such units are illustrated as circles. Further details regarding IP Banks and transactional databases are provided below in Section C.

As illustrated in Figure 1, each point-of-sale delivery system, such as systems 112A, 108A-B, 110A, and 114A-B, may be networked directly to peripheral information bank 102F, or the point-of-sale delivery system, such as systems 108C, 110B-D, 112B, 112D and 114B-C, may be networked to the point of purchase site 108B, which is networked to the peripheral information bank 102F. Point-of-sale delivery system configurations are explained in more detail below in Section C. At the level illustrated in Figure 1, however, it is important to understand that the delivery systems may be integrated into various combinations, such as a promotional point of rental system as shown by 110B and 112B, or a promotional point of purchase system as shown by 108B and 112C or a combination of a promotional, point of purchase, and point of rental systems as shown by 108C, 110D and 112D.

Communication network links between the central information bank 100 central transactional database 104, peripheral information banks 102A-F, and point of sale sites can be made utilizing one or a combination of many commercial available networks such as telephone, satellite or cable networks or any other medium suitable for transmitting information in digitized format. Many well-known protocols could be used in connection with the present system. For example, if the Internet is used as the "backbone" network, the well-known TCP/IP protocol could be used.

Figure 2 illustrates the flow of information in accordance with the embodiment of the system architecture illustrated in Figure 1. For ease of illustration

only, peripheral memory storage unit 106A is consolidated into central information bank 100, and peripheral transactional database 106B is consolidated into central transactional database 104. It should be understood, of course, that communication links between the peripheral information bank 102F and central information bank 100 and central transactional database 104 are provided.

As illustrated by the inputs provided to block 202, a publisher will receive inventory reports from the central information bank 100 and sales data from central transactional database 104. Based on this and other information the publisher can determine whether to place additional information on the network. For ease of reference such information is sometimes referred to herein as "information titles" as shown in block 204. If the information is not present in an electronic format, then the information is digitized 206, disposed in an electronic format 208 and then undergoes electronic authoring 210. The digitized information is then transmitted to a data converter 212 for converting the digitized information into a uniform format. For example, if the central information bank 104 and central transactional database 104 are DOS- or UNIX-based systems, the data converter will convert the information into a DOS or UNIX format, as appropriate. If the information titles are in a digitized format, the information titles are transmitted directly to data converter 212 for direct conversion into the uniform format as illustrated by line 214.

Once the data is in a uniform, digitized format, it undergoes an initial encryption and compression to both reduce the amount of storage space required to store the data and to make the data ready for being transmitted with less risk of unauthorized use while being transmitted through a communications network. The compression is accomplished through the use of one of the commercially available compression protocols. The initial encryption is performed using one of the standard available encryption protocols as discussed below in Section D.

Once in uniform, encrypted and digitized form, the information titles are stored in central information bank 100. An electronic index listing all titles

available and accessible by author, title, subject or ISBN codes and/or ISSN codes is prepared. As new information titles are added, the electronic index is updated to include the new titles. The information titles may then be downloaded to IP Bank 116A. The information titles and corresponding electronic index information may, in
5 addition to or rather than being stored in central information bank 100, be disposed on laser disk masters as illustrated at block 216. Laser disk masters 216 can then be installed directly into IP Bank 116A.

In one embodiment, as desired information data titles are being downloaded from IP Bank 116A to a user's storage medium 218, each file is being
10 dynamically encrypted and encoded with authorization and user identification data to restrict access and user of the information being transferred. The encryption process uses algorithms and formulae generated using a combination of the unique serial number of the storage medium to which the encrypted file is to be downloaded, the unique registration number given the user requesting the downloading at the time of
15 registration and the unique hidden number given the file, a copy of which is to be downloaded. In this manner, the data being downloaded is tied to the user requesting the downloading and the storage medium on which the encrypted data is to be stored. The unique serial number of the storage medium may be a number that is permanently associated with and stored upon the storage medium at the time of manufacture or
20 initial formatting or may be a number that is electronically generated and assigned when the storage medium is initially registered with the system. In both cases the number will be electronically readable and stored on a nonvolatile portion of the memory storage space associated with the user storage medium. In one embodiment, a current time and date are included with the data titles downloaded from IP Bank
25 116A so that a check is made of a clock associated with user storage medium 218 or an associated user device to ensure that no backdating of the clock in end user's device has taken place.

Prior to downloading desired information titles, the user may access an electronic index which contains all the information titles available for downloading

from IP Bank 116A. Through the electronic index, the user obtains the listing for available information titles by author's name, by specific title of the work, by ISBN code or by subject matter. Once compiled, a listing of the available information titles included in the index category selected and the other necessary information to allow the user to purchase or rent any information title contained in the index category listed is displayed on the video screen. Using the video listing, the user selects any title listed thereon and obtains a printout of the relevant information through the printer slot 342. Upon proper access by a user, the information titles may then be downloaded from IP Bank 116A onto a user's storage medium 218.

After downloading of information and corresponding electronic index information from central information bank 100 or installation of laser masters 216 to IP Bank 116A, inventory reports are generated by IP Bank 116A and transmitted to central information bank 100. These inventory reports reflect the information titles presently stored in IP Bank 116A. These reports are then sent to publisher 202. Also, a download completion report is sent from IP Bank 116A to transactional database 118A, sometimes referred to herein as a host fileserver, which in turn generates a status sales report. The sales report is transmitted to central transactional database 104. Transactional database 104 sends the necessary action instruction back to host fileserver 118A and a transaction report to publisher 202 for uses such as accounting and auditing.

The cartridge also contains programming (CCS) that regulates access and use of encrypted data files, decrypted an encrypted file when selected for opening, controls the operation of the user's computer when a decrypted version of an encrypted file is being accessed, and removes all trace of any decrypted version of an encrypted file from the user's hard drive or RAM when being closed. (RAM may include or consist of DRAM, SDRAM, and/or VRAM).

In one embodiment, a content provider can specify instructions relating to authorized use, access and cost of the provided information that can be permanently

linked to a master data file of the information when it is electronically stored or accessed. Instructions are electronically generated when information is received electronically. These instructions include a description of allowed uses (e.g., copy generation, printing limits and/or authorization, rental options, if any, purchase options, etc.), desired level of security against unauthorized copying or use (from
5 level zero [virtually no security] to a maximum level [most robust security available]) and cost to a user, optionally as a function of user-selected access right. Whenever a request is made to obtain a copy of the information file (i.e., IP File), the instructions are used in connection with the choices made by the user, within the allowed options,
10 to generate user-specific electronic instructions for limiting use of the requested copy of the IP File. For example, content providers who wish to allow a user to be able to produce a hard printed copy of a portion of an information file for study purposes, when the information files are being downloaded into the master file, can select that special authorization codes be included with the master data file. Once established,
15 the instruction codes so generated accompany the information file to the storage medium, or cartridge, of the user and regulate the use of the information file by the user. The cartridge retains information relating to such printing and restricts further printing once the limits have been reached. The user determines, within the defined limits, or authorized purposes, the portion of the text (natural language text or other
20 IP) to be produced as a hard copy by using the high lighting features of the reader programming to make a selection. In one embodiment, if no special codes are selected, default codes apply that embody the most restrictive instruction selections. In another embodiment, the content provider can allow the user to select the applicable use and access instructions from a variety of authorized choices with the
25 selection being made at the time of acquisition. The cost of acquisition to be paid by a user will vary based upon the user choices regarding desired use. In one embodiment, rules can be updated or adjusted by a content provider.

In another embodiment of the present invention, when one or more IP Banks are operated in conjunction with one or more retail establishments, each retail
30 establishment is given a unique identification number that is used to tag each user

device, user storage medium and/or encrypted IP file acquired at the retail establishment or through an IP Bank linked to the retail establishment. Using the tagged information, particular user devices, storage media and/or IP files, a predetermined portion of the system revenue generated in regard to such tagged user devices, storage media and/or IP files is allocated to and shared with the retail establishment in question. In another embodiment, the tagging is further delineated to allow revenue sharing with manufacturers and distributors of user devices and/or storage media.

C. Point-of-Sale Delivery System Configurations

The point-of-sale delivery systems, as previously discussed, are classified by function. The functions include one or more of the following: (1) point of purchase delivery system, (2) point of rental delivery system, (3) IP bank subsystem, and (4) promotional delivery system. The configurations for each of these functions are separately discussed in detail below.

Point of Purchase Delivery System

A point of purchase system is illustrated in block form in Figure 3. The point of purchase system is described herein for illustration purposes only as a system from which IP can be purchased. As pointed out above, however, the system is not limited to a particular type of IP and other media capable of being expressed in electronic form such as computer software, music and video could be purchased utilizing the present system. In addition, although the system described here is a point-of-purchase delivery system, other embodiments employ either synchronous or asynchronous IP network protection, or both, making possible real-time IP transmission.

The point of purchase system illustrated in Figure 3 includes an IP Bank 302 coupled to host fileserver 304. Server 304 is coupled to a customer service terminal 306 (of course, there could be more than one terminal) and a cashier's station

308A which is further interconnected to other cashier stations 308B-D. Server 304 also is coupled to an institution network 310, which in turn connects to institution terminals 312A-E. Service terminal 306, cashier stations 304A-D and institution network 310 are connected to server 304 via a computer communication link such as a commercially available computer networking system such as CompuServe, a public network such as the Internet, an intranet, or a virtual private network (VPN). IP Bank 302 and server 304 are connected to central information bank 100 and central transactional database 104 as explained above with reference to Figs. 1 and 2.

Cashier stations 308A-D are in serial, linear networking connections which allows the addition and removal of a number of cashier stations at any time. This configuration accommodates extra cashier stations required during rush seasons or rush hours and the desire to remove cashier stations for better utilization of space after the rush seasons. Customer service terminal 306 has local processing capability that provides customer services such as personal identification initiation, personal identification number changes, processing of complimentary IP, IP refunds, customer information entries and updates. The customer services terminal 306 can also provide the retail outlet with internal administration and the management functions, such as the IP inventory cards management, the IP list management, IP requests, IP reports, financial reports, and e-mail and Bulletin Board management.

Referring now to Figure 4, point of purchase fileserver 304 is shown in more detail. Particularly, server 304 includes one or more central processing units (CPUs) 316, a primary power supply 318, an uninterrupted power supply 320 to assure continuous operation during power failure, and a high density storage 322 that holds all the programs and the databases required for server 304 operation.

Server 304 has four (4) interfaces, i.e., a network interface 324, a maintenance interface 326, a customer service station interface 328 and a cashier station interface 330. CPU 316 transmits instructions to IP Bank 302, creates

transaction databases and reports, and processes orders from cashier stations 308A-D and customer service terminals 306A-D.

From network interface 324, server 304 communicates with central transaction database 104 for electronic filing of transaction reports and communicates with IP Bank 302 to give IP Bank 302 downloading instruction orders and to receive the status reports and the inventory reports from IP Bank 302. Server 304 also is coupled, through network interface 324 to an IP Bank subsystem to receive subsystem reports in order to give instructions and orders whenever necessary, as hereinafter discussed. External network systems such as institutional or corporate network systems with local merchants terminals, community bulletin board services and others can also be coupled to the network interface 324. The network interface 324 also allows two-way connecting with inter-bank networks such as Cirrus, Plus or other similar data transfer network. Coupling to merchants' terminals, promotional system provides local merchants and the local business direct access to update their promotions and coupons. Maintenance interface 326 enables remote or on-site diagnosis and repair of server 304.

Customer service station interface 328 provides for communication between server 304 and customer service terminals 306A-D to handle customer service transactions. Customer service terminals 306A-D are illustrated as being coupled through a data switch 332 to a printer 334. Cashier station interface 330 provides that cashier stations 308A-D can communicate with server 304.

Figure 5 illustrates one embodiment of IP Bank 302. IP Bank 302 includes a high resolution color graphic display 336 which is a touch screen device used to display, for example, instructions, messages, and status reports to the user, indexing information and to receive the user's touch screen input selections. IP Bank 302 also has a keypad 338 that is for the user to input a personal identification as well as other inputs. A magnetic code or other generally accepted card reader 341, shown as an insertion slot, is provided for customers' transactions with a bank card, credit

card or some other form of debit card. A bar code reader 340, shown as an insertion slot, is provided to allow users to insert cards containing ISBN and/or ISSN codes for desired information titles for reading by the IP Bank. ISBN and/or ISSN codes may also be manually inserted, for example, by typing the relevant numbered keys on the keypad 338 or with a joystick type device (not shown). A printer slot 342 also is provided to enable the user to access the output of IP Bank 302, a printer (not shown in Fig. 5), as hereinafter described, to retrieve receipts and transactions reports and ISBN access vouchers. IP Bank 302 also includes a base member 344 with a cut-out portion 346 to enable a user to stand comfortably at keypad 338. Other configurations are possible for users having special physical needs. Importantly, IP Bank 302 also includes a cartridge slot 348 for the user to input a reading cartridge, as explained in detail hereinafter, to obtain a copy of the information selected for downloading.

In another embodiment and as shown in Figure 5A, the physical embodiment of IP Bank 302 may be altered. More specifically and in one embodiment, IP Bank 302 is positioned on a desk 339 using a personal computer 349, for example those available from IBM Corporation or Dell. The user operates IP Bank 302 as described above with reference to Figure 5 except, the user may, for example, sit in a chair (not shown). Such a configuration may be used, for example, in a corporate, dormitory, library, or other similar environment where the user may be accessing information for longer periods of time or in a professional type environment. In other embodiments, readers 340 and 341 and cartridge slot 348 each may be located within computer 349 or in separate devices which are electrically coupled to computer 349.

In yet another embodiment of the invention, and as is shown in Figure 5B, a different physical embodiment of the IP Bank 302 is provided. More specifically, IP Bank 302 is placed in proximity to central information data storage facility 100 and is in electronic communication with central information data storage facility 100 via any suitable means (e.g., cable, wireless, etc.). In this embodiment, IP Bank 302 is contained within a housing 500 for combined central storage 100 and IP

Bank 302. User access to IP Bank 302 is via dial-up modem 502, 512 over the telephone, via the Internet through a web site 504, 506 maintained for that purpose, via satellite linking, via wireless (such as narrowband wireless, broadband wireless, infrared, microwave, radio wave or other similar connection) or via any other network capable of electronically transmitting data in digital form, for example, a cable
5 modem, ADSL-X or broadband wireless network. Once IP Bank 302 is accessed by the user using specialized software contain in a WORM (write once, read many times) memory or other nonvolatile memory on the IP Bank and on the user's storage medium 218, a virtual IP Bank is generated for viewing and access using the user's
10 computer. The virtual IP Bank then performs all of the same functions as IP Bank 302 in another embodiment, except that memory of central information storage facility 100 is used as storage for the information titles and the transaction information, the appropriate hardware connected to the user's computer operates as an access port to IP Bank 302, and specialized software programming stored on the
15 user's storage medium in connection with specialized software programming stored on the WORM (write once, read many times) memory of the IP Bank operates as the software programming to drive the IP Bank.

In yet another embodiment of the invention, IP Bank 302 is a special board configuration located within the user's computer or a separate device
20 electronically linked to the user's computer by a cable, a phone line, a wireless connection (infrared, microwave or otherwise) or any other suitable means. In this embodiment, IP Bank 302 is configured in such a manner that any attempt to read the data contained thereon other than as authorized renders the data totally unreadable. IP Bank 302, in this embodiment, is connected to the central information storage facility
25 100 or another IP Bank 302 using a secure network. In one embodiment, IP Bank 302 also functions as a controller of the user's computer when being used to access encrypted files or the secure network. In this embodiment, IP Bank 302 also contains operating software that is stored in WORM (write once, read many times) memory within IP Bank 302.

Figure 6 is a block diagram description of IP Bank 302 circuitry. Particularly, IP Bank 302 includes a central processing unit (CPU) 350, which is coupled to display 336, keypad 338, magnetic strip reader 341 and bar code reader 340. Although CPU 350 is illustrated as one unit, it is contemplated that CPU 350 could be a parallel processor or distributed processor arrangement. Selection of CPU 350 type depends on the amount of information to be processed, the desired speed of processing and costs. CPU 350 also is coupled to an automatic teller machine (ATM) module 352 to allow transactions with ATM cards. CPU 350 is coupled to a media driver 354 which enables users to insert personalized media for acknowledgment or other functions as hereinafter discussed. IP Bank 302 also includes a primary local storage device 356 provided for the storage of all information masters selected for loading into IP Bank 302 and related index information. A secondary storage device 358 is provided to hold other programs, instructions and transaction related information. A buffer memory 360 is utilized to speed up downloading in order to accommodate high volume users during the peak seasons. A printer 362 is provided to print coupons on demand, receipts and various reports for the users. A power supply 364 provides power to printer 362. CPU 350, secondary storage device 358 and local storage 356. An uninterrupted power supply 366 coupled to primary power supply 364 assures continuous operation even during power down time.

CPU 350 is coupled to a network interface 368 to provide communication to central information bank 100, host fileserver 304 or an IP Bank subsystem, as hereinafter discussed. CPU 350 also is coupled to a wireless communication port 370, which in turn is coupled to an antenna 372. Wireless communication port 370 enables compatibility with an alternative communication medium in the event that such medium is required.

Figure 7 illustrates, in block diagram form, the structure for a user's personalized media storage cartridge 374. As explained hereinafter, a user inserts cartridge 374 into cartridge slot 348 for downloading of the information selected from IP Bank 302. The downloaded information is stored, in an encrypted format, on

cartridge 374 together with relevant basic index information copied from the electronic index contained in IP Bank 202 at the time of initial downloading. Cartridge 374 is compatible with readers to enable the user to view information stored on the cartridge. Cartridge 374 includes reading software 376 that, as explained in
5 more detail hereinafter, performs sequential encryption and decryption of information. Registry correspondence segment 378 also is provided. an IP file registry 380 is created at the time of downloading information onto cartridge 374. Encrypted IP files 382 together with relevant electronic index information are stored on cartridge 374 as well as a non-erasable permanently marked serial number 384. Cartridge 374 also
10 contains a commercial operating system environment 386 and free disk space 388.

In another embodiment of IP Bank 304, slot 348 is a part of a device connected to the user's computer being used to access the encrypted files.

Figure 8 illustrates the user process and component processing which occurs when a user utilizes the point-of-purchase system described above.
15 Particularly, once the user enters the site 390, and if the user a first-time patron 392, the user will complete a user's application form 394. The user will then take the completed application and a picture I.D. to customer service station 306A, where the user will select and input a personal identification number (PIN) and a password 396. The customer service clerk will open an account for the customer 398. The user-
20 selected password is automatically matched with a sequentially created customer account number within the central data banks. Using the keypad accompanying the cashier station, the clerk types in the name, address and social security number for the user. The written application will then be inserted into the printer slot accompanying the cashier station. While loading the customers' information, the central data bank
25 reviews the information to determine if there are any prior problems with the customer or other discrepancies. When the verification process is completed, the designated customer account number for the user is printed on the user's application. Then, the clerk places, into the card slot, a user identification card (for example, a plastic card having dimensions of the standard credit card and containing a magnetic

strip on the back on which can be placed magnetic coded information). The card is then embossed with the user's name and account number and the magnetic strip is encoded with the applicable user codes (account number, card number and designated password information). The written application is then transmitted to the central data storage center for retention. The user now is able to use the issued card to make purchases or to rent the use of information titles. The machine used to emboss and encode the cards is a standard commercially available machine of the type currently being used in connection with the issuance of a bank's card, credit cards or debit cards. Using the password supplied by the user and the application registration number, a unique user identification number is electronically generated. This identification number is then magnetically stored on the magnetic strip on the back of the user's card and within the data files of the machine generating the embossed card. This information is stored in an encrypted file later transmitted to central information storage facility 100 using a suitable secure network. The unique number so generated is used to represent a personal signature (system registration number) of the user with regard to access to the system and encrypted files.

A user may also obtain a personal identification card by accessing the system, through a network, and by supplying the necessary information for the generation of a user identification file. Upon completion of the registration process, the user electronically receives a digital identification file and identification number (or password). At the time of registration, the user is assigned a unique account number that is associated with his or her data files within the system for accounting and tracking purposes. A digital file containing the assigned account number, the user identification number and the applicable selected password information represents the identification file. After the identification file has been stored, the user may obtain a printed copy of the relevant information contained in said file. At the same time as the identification file is being generated, the system programming generates a unique number using a combination of the user account number and the numeric equivalent of the user password. An electronic version of that unique number represents the electronic personal signature (registration number) for the user. A digital copy of the

unique number is stored with the user's identification file using a special encryption method. Only the public (non-hidden) portion of the data is available for printing to hard copy. The digital identification file is available to be used in tagging and encrypting data files and allowing access to the network and encrypted data files. A
5 copy of the registration file, including a copy of the identification file, is stored at central information storage facility 100.

In addition to obtaining a personal identification card, a new enrollee purchases a reader/computer or other acceptable reading device (such as a special computerized interface, or an audio or video playback device). Each such device is
10 assigned a unique serial number and a special code number. In one embodiment, the serial number is contained on a read only memory chip enclosed within the device. All of the many cartridges which accompany each such reading device is encoded in such a manner that information recorded on the cartridge can only be read by the related reading device. In one embodiment of the invention, this is accomplished by a
15 simple program contained within the permanent memory of the device. In another embodiment, if the special code on the device is not the same as the number which the cartridge is seeking, then the cartridge will cause to be displayed in the reading device the words "cartridge cannot be read by this device" and to not allow any further access to any information contained on the cartridge by the particular reading device
20 in question. If the numbers match, further access will be allowed. At the time a reading device is purchased, the clerk enters the serial number for the device into the central data bank through the cashier's station. The central data bank contains a list of the serial numbers of all approved reading devices and the corresponding special code number. The personal identification card for the customer purchasing the reading
25 device is placed by the clerk into an appropriate slot on the cashier's station and the magnetic strip on the identification card is encoded with the applicable serial number for the reading device being purchased. Thereafter, whenever the user desires to obtain additional cartridges for reading by his or her reading device, the user needs only present his personal identification card and the cartridge to be properly coded to
30 the clerk and by inserting the cartridge into the cartridge slot and the identification

card into the card slot and pressing the designated button on the cashiers station, the new cartridge will be correctly encoded to be readable by the user's reading device.

In another embodiment of the invention, the personal identification number given a user at the time of registration, the unique serial number of the user's storage medium being used to store the downloaded files and a hidden unique number
5 given the content being copied for downloading are used to generate unique encryption and decryption algorithms and formulae. A copy of the algorithms and formulae so generated are stored in the central information storage facility. The encryption formula so generated is used to encrypt the data files as they are being
10 downloaded and stored on the user's storage medium. The decryption formula and algorithms are stored in nonvolatile memory associated with the user's storage medium and are available for decryption when access to an encrypted file is requested. The memory unit and the housing thereof, where the decryption formula and algorithms are stored when downloaded to the user's storage medium, are
15 configured so that any attempt to read the information so stored, other than through the use of the CCS software and related operating programming, renders the data unreadable in any manner. WORM memory is used in one embodiment of the memory unit.

The requirement that reading device codes and cartridge codes match,
20 before access will be allowed, means that issued cartridges will not be readily readable by multiple reading devices. Multiple device reading will require special programming and the granting of special allowances, or approved purposes. In one embodiment, by purpose encoding the information, a publisher may expand or limit access to those users having a proper authorization or defined purpose. For example,
25 a publisher may purpose encode a portion of a selected IP to be readable by any user, i.e., any purpose encoded. This type of purpose encoding may be used on, for example, promotional IP or some governmental publications. Other information may be purpose encoded to limit access to a single user to only view the information, i.e., classified material. The requirement reduces the possibility of unauthorized use of

information titles. Special allowances also can include various security code levels that allow publisher studios and content owners the ability to choose between a wide variety of security from completely open to the highest level in which the content of an IP can only be utilized by one individual. Security codes and special allowance
5 rules for content owners could, for example, be determined by the market value of the content. The life cycle of the content can also be considered, so that new edition music, movie release, and new computer games have different security compared to content that has been in the market for an extended period of time. In staging security related to time value, market value, control level, type of intellectual property, or even
10 legality, the highest level of security, in one embodiment, is the use of the highest security encryption algorithm with a hardware smart chip or smart storage device controller chip or processor.

In another embodiment in which IP Bank 302 is located either in proximity to central information storage facility 100 or in proximity to the user's
15 computer, the user's storage medium is encoded with the applicable registration codes and numbers by use of the appropriate slot on either the user's computer or IP Bank 302, were a separate external unit is being used as such. With the appropriate software activated, the user inserts the applicable storage medium into the appropriate slot and using the appropriate software menu selects the appropriate instruction. The
20 software programming then develops the appropriate registration information and encodes it on the storage medium in the appropriate manner. As part of the initiating process, the software programming checks the storage medium to make sure that it already contains the appropriate CCS. A user storage medium that does not already have a unique serial or identification number is given one and, and any user storage
25 medium that does not accept electronically being tagged with a serial number in a appropriate manner is rejected and so marked. Only user storage media containing the proper software programs and registration information are usable to store data information files.

In operation, the customer takes the customer identification card to the IP display area for shopping 400. If the customer previously opened an account, the customer will not have to go through the above described process and can proceed directly to shopping area 400, where the customer will select an IP inventory card matching his IP selection. The IP inventory card has the IP ISBN and/or ISSN numbers, a bar code and information related to the particular information title, author, publisher, and edition date printed thereon. The customer brings the selected IP inventory card to the cashier's station 308A. The cashier magnetically reads the codes on the customer's I.D. card and scans or manually enters the bar codes on IP inventory cards 402. The customer then makes a proper payment, and the customer codes and information title bar codes are transmitted to host fileserver 304. Server 304 searches the existing customer account file to match the identification (i.e., pin and password) and will generate a downloaded IP list file based on the bar codes from the IP inventory cards or as manually loaded. Server 304 downloads the file to IP Bank 302 which electronically generates a portfolio of information titles ready to be downloaded on demand. The user can then proceed to IP Bank 302 at any later time and insert the identification card into the slot 340 of IP Bank 302 and a coded point-of-purchase cartridge 374 into IP Bank cartridge slot 348 to identify himself with a personal identification number, as illustrated in step 404. The user also enters a password 406 into the keypad 338. When IP Bank CPU 350 matches the personal identification number with a downloaded list portfolio, IP Bank CPU 350 starts downloading the requested information from local storage 356, through buffer memory 360, to media driver 354 which copies the information onto cartridge 374. As part of the downloading process, the data is dynamically encrypted to make the data uniquely readable only by authorized reading devices. The dynamic encrypting is described below in Section D. After downloading, the user removes cartridge 374 and then inserts cartridge 374 into his personal reader/computer to access the information acquired. The reader/computers are configured for long-term reading applications. The reading application software is stored on cartridges with the ability

to read the applicable software on the cartridges permanently stored within the memory of the reader/computers or other authorized reading device.

A user may select portions of selected information to combine and download. In one embodiment, the user may select at least one IP and select at least one portion of each selected IP. If more than one portion is selected, where each portion includes up to the entire selected IP, the portions are combined and downloaded to user's cartridge 374. For example, for a specific college course, a student may be required to download specific chapters from ten different IP. After selecting the ten IP and ten specific chapters of the selected IP, the selected information is combined, encrypted using the determined level of protection, and downloaded to the students cartridge 374. Similarly, a user may select individual tracks from music information to combine and download the selected tracks to a single cartridge 374 for playback at a later time.

In another embodiment in which IP Bank 302 comprises hardware and software in the user's computer and hardware and software located other than in proximity to the user's computer (e.g., an IP Bank located proximate to central information storage facility 100 or a stand alone IP Bank located apart from the user's computer and central information storage facility 100), after connecting to the IP Bank unit 100, using the appropriate browser software, the user will access the central storage facility 100 index of available titles and select the desired titles or parts of titles that the user wishes to purchase. From this information (from which a work is created that represents a portion of several different works), IP Bank 302 will cause to be generated a unique identification number for the newly created work. This unique identification number is generated in addition to any hidden code that is generated and serves as an index reference for the work. IP Bank 302 also uses the user's request to generate a shopping list with appropriate price information, which the user may accept as generated or change. When the list is accepted by the user, the menu requests payment information. Generally, at the IP Bank level, payment is via pre-generated credit voucher or via credit or debit cards. After payment approval, IP

Bank 302 causes copies of the requested titles to be generated, encrypted, and downloaded on to the user's storage medium. While connected to the user's computer and storage medium, IP Bank 302 examines the appropriate log files to determine whether there has been any misuse or unauthorized activities. If such a determination is made, the account is flagged for further investigation.

Point of Rental Delivery System

If a user is not interested in obtaining a permanent copy of a particular work but requires a copy for a period of time, e.g., a semester, the user may prefer to visit a point of rental site rather than a point of purchase site. A point of rental system is illustrated in Figure 9. The rental system is identical to the point of purchase system previously described herein (e.g., includes a host fileserver) except with respect to the differences pointed out below. In many instances, a single site or IP Bank may serve as a point of purchase system site and a point of rental system site and a point of delivery system for promotional or commercial information site or any combination thereof. As shown in Figure 9, the point of rental system includes IP Bank user terminal hubs 410A-B coupled to terminals 412A-E, and customer service station 414. User terminals 412A-E allow a customer to do an information title search and index search of the IP Bank memory and to transmit other information between IP Bank 302 and himself. Customer service station 414 combines the function of customer service as well as the cashier's station. For example, at customer service station 414, a credit customers' debit card can be credited and the ATM operation can be overridden, via ATM module 416, if necessary. Information can be printed out from customer service station 414 via printer 418.

Point of rental storage medium 420 is used in the rental system. Point of rental medium 420 is the same as point of purchase medium 374 except that medium 420 includes an automatic erasure mechanism that erases the information downloaded after the expiration of a preset time interval. More specifically, when information is downloaded from IP Bank 302 onto medium 420, IP Bank 302 also

downloads a "time stamp" (using a nonvolatile, time-encoded chip that cannot be manipulated) equal to the time period for which the user has paid to retain a copy of the information. The time stamp could take the form of a value loaded into a memory location on medium 420, which value corresponds to the rental time period. During
5 usage, the actual usage time elapsed is subtracted from the electronically stamped time period. Once the user has consumed all of the usage hours or other time units authorized, the information title will self-destruct, i.e., be deleted from medium 420. This can be achieved simply by calling a stored program which erases the information associated with the memory location where the time value is stored. For example,
10 when the value of the memory location where the authorized usage time is stored is zero, the stored erase program would be called upon to erase the information associated with the "zero" time usage authorization. In one embodiment, an advance alert feature is provided to allow the user time to pay for additional authorized use before the information is erased.

15 Another method for automatic erasure is for each rental or library cartridge to contain a real time clock and independent rechargeable power supply. When the cartridge is initially encoded for use, the real time clock mechanism is activated. As rented information titles are being downloaded, an expiration date is logged into the index information for each title. Any time after the real time clock on
20 the cartridge reaches the designated expiration date, access to the relevant information title is denied. If use of the title is not extended, after the expiration of an additional number of days, use of the cartridge will cause a permanent erasure of the information title from the cartridge memory. With this method, if the real time clock falls to operate, the cartridge will become unreadable without repair. To repair a defective
25 cartridge, the user need only bring the cartridge together with his personal identification card to the nearest service center where the real time clock will either be repaired or the relevant information titles will be loaded onto a replacement cartridge. The user will be credited for any lost time while the cartridge was unreadable. A service center could be located, for example, near each separate point of delivery site.

In one embodiment, GMT clock verification is added to the user device, to provide additional IP security in case the user travels with his device through time zones. An encrypted GMT code is added in another embodiment to compare and verify erasure dates and times with actual GMT.

5 The automatic erasure program could be created as an operating system module or as a separate executable program designed to be "terminate and stay resident" (TSR). A module integral with the operating system is preferred since such a structure ensures that if the operating system is viable, the automatic erasure module is viable.

10 If the user still needs more time with any particular information titles, the user may return to the point of rental site and "re-rent" the information. Alternatively, it is contemplated that the user could renew the rental via a modem coupled to the reader.

15 With respect to the user process for renting information, when a point of rental patron enters a point of rental site, the user will use a valid ATM card, bank card, credit card, or some other debit card and proceed directly to a user terminal 412A-E. Using such a terminal, the user can perform information title searching and download an order entry to IP Bank 302. When the download entry is complete, the user will go to IP Bank 302, insert a rental medium, an identification card, and a
20 credit card, bank card, or debit card into IP Bank 302 for the transaction approval. If the transaction is not cleared or if the ATM system is not working properly, the patron can proceed to the customer service center and have the attendant manually override the ATM process, if appropriate. If the user does not have a valid ATM credit card or debit card, the user will go to the customer service center, pay the service clerk to
25 receive credit on the IP Bank debit card. Then the customer may proceed to the user terminal where the user downloads the order entry.

After the transaction approval is cleared, the patron inserts point of rental medium 420 into IP Bank media driver 354, has his personal identification card

scanned and enters a password. The information is dynamically encrypted and downloaded from IP Bank 302 to medium 420 with an electronic stamp of the number of hours of usage authorized for each information title or an expiration date. After the downloading, the user will apply this medium on the personal reader/computer to
5 access the information on the medium.

Typical examples for the point of rental site are libraries (commercial, education or public access) and IP rental shops. The information downloaded by the user may be free of charge to the users such as in the case of a library, or may incur certain rental fees at a predetermined rate, such as in the case of a rental shop or
10 library charging on a per page use basis. Any given point of rental site may operate as a traditional library in allowing free use to library members for a limited period of time or may operate as a rental shop where fees are collected from users in accordance with the period of use allowed.

IP Bank Subsystem

15 A IP Bank subsystem couples to an IP Bank and host fileserver as described in more detail below. The central element of the subsystem is an IP Bank which is a modified version of the point-of-purchase IP Bank 302. The subsystem is specifically configured for the collective use by members or the staff of a commercial or business entity or a corporation. It delivers and it recalls information titles among
20 authorized users within the business or corporate entity, and provides the capability of limiting the number of copies of a given work that may be distributed to other authorized users. If all of the licensed copies of any information titles have been checked out by the staff of an organization, then no other users may access the same information title within that particular subsystem until one or more of the licensed
25 copies of the particular information is uploaded or recalled to the subsystem or additional copies are purchased. In one embodiment, the information is purpose encoded so as to limit a purpose to which access by a user to the information is allowed. For example, a central corporate library may allow specific users, i.e., R&D

personnel, to selected information, i.e., pending patent applications. All non-R&D personnel users without the proper purpose, or authorization code, are prevented from accessing the information.

5 Instead of purchasing the unlimited use of a limited number of copies, a commercial or business entity may lease the limited use of an unlimited number of copies or the use of a specified portion of a given information title. Under such circumstances, the commercial or business entity would be charged each time the subsystem is accessed from a participating work station for the portion of a specified information titled accessed and for the period of time the access occurs. By
10 restrictions encoded on the interface between a participating work station and the subsystem, while accessing information from the subsystem, the ability of the work station to perform certain operations would be restricted. The restricted operations would be those related to the duplication or transmission of data related to information titles being accessed through the subsystem.

15 More specifically, and referring to Figure 10, IP Bank subsystem 422 contains a high resolution color graphic display 424 coupled to CPU 426 to display the instructions or status of subsystem 422. Subsystem 422 also includes a keyboard 428 with limited access to the system for keying selections for operating certain given functions such as product display. (In one embodiment, a voice recognition system is
20 used for input in place of keyboard 428 or to provide additional input capability.) Subsystem 422 has a media driver 430 for the downloading of information and a local storage 432 which holds a portfolio of information the business entity has ordered for use. A secondary storage 434 is also provided to hold all the software programs that control and perform the functions of subsystem 422. Subsystem 422 further includes
25 a power supply 436 and an uninterrupted power supply 438 to assure continuous operation during power failure downtime. A printer 440 is provided to print various reports.

IP Bank subsystem 422 has a network interface 442 that connects subsystem 422 to IP Bank 302 and host fileserver 304. Network interface 442 also may couple to the corporate or business entity network system 444. With such a structure, the corporate entity may transmit or download its own corporate proprietary information through IP Bank subsystem 422.

Media port extension interface 446 provides access by an adequate number of media drivers to the desired corporate terminals for corporate network stations. Media driver 430 is connected to the terminals or stations by a proprietor driver card. The corporate administration can utilize the dynamic encryption and the dynamic downloading function of IP Bank subsystem 422 to incorporate and accommodate the corporate proprietary information. The corporate proprietary information may be transmitted to IP Bank subsystem 422 using an encryption process and then downloaded selectively to the destination port and to the properly identified authorized personnel. IP Bank subsystem 422 is not only a customized corporate library of copyrighted and proprietary information, but also is a corporate document security device that encrypts and dispatches the corporate documents and the corporate confidential proprietary information in the corporate network system. As part of the network interface connection linking each participating work station to the subsystem and allowing access to encrypted information, a separate unit, e.g., a memory storage unit restricts certain operations which may be performed from the work station so long as the work station has access to encrypted information from the subsystem. The restrictions limit or prevent operations related to the duplication or transmission of data.

Promotional Delivery System

The promotional system is a point of delivery system for promotional and commercial information. It distributes promotional and commercial information in electronic format and users may either view the digitized promotional and commercial information at the site or download the information to their personalized

media for later viewing. User's call access the promotional and commercial information including the dynamic viewing electronically of advertising available discounts, commercials, special promotional events, software demos and product catalogs. Users may even shop electronically by manipulating the promotional and commercial information and placing orders through e-mail from a personal reader/computer or by ordering directly from an interactive promotional IP Bank. The promotional IP Bank has the same structure as IP Bank 302 for the point-of-purchase system.

A promotional system in accordance with the present invention is illustrated in block diagram form in Figure 11. As in the other point-of-sale systems, IP Bank 302 is networked to host fileserver 304. The promotional system further includes a number of promotional units 448A-D which electronically display and promote products. Unit 448A is coupled directly to central transactional database 104 and central information bank 100 while units 448B-D are coupled to host fileserver 304. Unit 448A receives information from merchant terminals 450-A-D and host fileserver 304, receives information via merchant terminals 450E-G. More specifically, host fileserver 304 receives advertising and special offer updates from the local businesses, national or regional advertisers, and corporate sponsors through merchants terminals (MT) 450E-G. The host fileserver 304 is also networked to a central transaction database which, in turn, provides a report to the publishers, advertisers, accounting, auditing firms, merchandise vendors, and others.

The promotional IP Bank allows selective downloading of promotional and commercial information to the user's point of rental medium (see discussion in Section B, System Architecture, for explanation of such downloading) for the user's private review and personal shopping at his convenience. The promotional and the commercial information downloaded will self-destruct (i.e., automatically erase) at the expiration of a pre-determined time interval as explained above with respect to point of rental delivery systems. The promotional IP Bank also provides a user interactive self-service vending feature. The user may order products or information

electronically via the network. Some of the promotional functions are: coupons on demand, virtual shopping, catalog sales, demos, subscription orders, electronic applications of credit cards, calling cards, or other types of services. Some public domain information distributed such as community events, ticket sales, institutional events or even public bulletins could also be distributed with the promotional information as a free or low cost service to the community.

The promotional and the commercial information flow is very similar to the information flow within the point-of-sale delivery system. However, rather than a publisher or copyright information owners, the information sources are local businesses, national or regional advertisers, and appropriate sponsors through advertising agents and other entities.

Information Tracking

In one embodiment, for each exchange, or download, of information, a tracking entry is transmitted, or stored, in an appropriate transactional database, for example central transactional database 104, to record movement, or transfer, of information from a first location to a second location. By reviewing these tracking entries, an information owner may monitor movement of the information and take appropriate action. For example, a tracking entry may be recorded in transactional database 104 each time any information is copied into, removed from, or copied from, IP Bank 302. Based upon the tracking entries, the information owner may charge the receiving or transferring party a fee as defined by the owner of the information. The fee charged may be based on a variety of factors including, but not limited to, an economic value of the information, use or purpose of the information, number of users, and the availability from other sources. The tracking entries may also include additional data so that the information owner may determine who transferred the information, the amount of information transferred, type of transfer, i.e., rental for specified period of time, and the time of transfer. Tracking entries, in one embodiment, are recorded for all transfers, i.e., IP Bank 302 to cartridge 374 and

central information bank 100 to IP Bank 302. Utilizing these entries, an information owner may also determine the type of information that is being transferred, the number of transfers, and the identification of the information receivers. For example, the information may be used to determine whether targeted users are receiving certain promotional materials are being received by targeted users, or to determine responses to different information pricing strategies.

D. Encryption

The above-described point-of-sale delivery systems have the capability of performing dynamic encryption of data as the data is downloaded onto a user's storage medium. Dynamic encryption refers to the process in which the IP Bank works together with the storage medium to perform a proprietary encryption of downloaded data. In one embodiment, different levels of encryption are utilized based on a series of factors or variables. These variables include, but are not limited to, an economic life, a market value, a general availability, a replacement cost, time sensitivity, and potential number of users, of the information. For example, today's TV listings may have a low level, or complexity, of encryption as a result of the low market value, low replacement cost, and general availability from many sources. Conversely, a multi-volume legal treatise may, for example, have a high, or complex, level of encryption as a result of the limited availability, replacement cost, and long economic life of the information. Based on the described factors, a source, i.e., publisher, of information may decide the appropriate level of encryption for each portion of information from the initial transmission to central information bank 100 to a user's cartridge 374. The level of information encryption, in any location, i.e., bank 100, may be higher or lower than another location, i.e., cartridge 374. More specifically, each time information is downloaded, or transmitted, the level of encryption may be independently altered, or determined. For example, the level of encryption at central information bank 100 may be different, i.e., higher or lower, than the level of encryption of an IP downloaded to a user's cartridge 374.

In addition to dynamic encryption, other encryption may be performed as illustrated in Figure 12. Figure 12 illustrates a three level encryption process. For example, prior to transmitting information on the network, the data may be encrypted. This facilitates preventing unauthorized users from accessing the transmitted
5 information on the network. In addition to the pre-transport encryption, the data, may be encrypted prior to being placed in an IP bank. Publishers or other owners of the information may have approval authority over this level of encryption to provide such information owners with satisfaction that the data is adequately protected.

Once the data is stored in the IP bank, dynamic encryption techniques
10 may be used when downloading the data onto storage media. The storage medium (Figure 7) includes a proprietary environment for building, reading, viewing and processing. The medium also has a commercial operating system environment for processing information files. An information file directory registry forms a part of the proprietary application, and a file directory pointer is contained in the operating
15 system application.

The dynamic encryption process, in one form, uses the permanent serial numbers stored in the storage medium, the user's personal identification number, a personal signature code number, and a password to further encrypt the data stored in the IP bank as the data is downloaded to a user's storage medium. The
20 personalized variables and codings are combined with various individualized information file variables to form an individualized data structure for the data downloaded to the user's personalized medium. As a result, those information files are individualized pertaining to the medium, the version of software, the information file itself, and other variables.

25 The dynamic encryption assists in reducing the possibility of the unauthorized use of proprietary or other information by causing all information downloaded through the point-of-sale delivery system to be readable and accessible by a selected number of user readers/computers. Specifically, data storage medium

accessible from one reader/computer will not be accessible using another reader/computer unless such access has been prearranged such as by providing the other reader/computer with an identical user identification number and password.

Examples of well-known encryption algorithms which may used in performing the above described three level encryption include the Z8068 Data
5 CIPHERING PROCESSOR (DCP). The DCP contains the structure to encrypt and decrypt data using National Bureau of Standards encryption algorithms. It may be used in a variety of environments including in dedicated controllers, communication concentrators, terminals and peripheral task processors in general processor systems.
10 DCP provides a high throughput rate using cipher feedback, electronic code IP or cipher block chain operating modes. The provisions of separate ports for key input, clear data and enciphered data enhances security. The host system communicates with the DCP using commands entered in the master port or through auxiliary control lines. Once set up, data can flow through the DCP at high speeds because input,
15 output and ciphering activities can be performed concurrently.

In alternative embodiments, encryption and decryption may be performed in dedicated hardware and/or software functions. For example, each reader and cartridge 374 may include a dedicated encryption integrated circuit (IC) and a dedicated decryption IC to maximize the transfer speed of the information. The
20 level of encryption and decryption may be altered by adding additional functions and by enabling or disabling the additional levels.

With respect to dynamic encryption, the following describes one of many methods of dynamic encryption that could be used. Particularly, each regularly used alpha or numeric symbol is assigned a corresponding number as illustrated in
25 Table 1.

Table 1

symbol	A	B	C	D	E	F	G	H	I	J	K
code	1	2	3	4	5	6	7	8	9	10	11
symbol	L	M	N	O	P	Q	R	S	T	U	V
code	12	13	14	15	16	17	18	19	20	21	22
symbol	W	X	Y	Z	0	1	2	3	4	5	6
code	23	24	25	26	27	28	29	30	31	32	33
symbol	7	8	9	.	,	;	:	+	-	x	
code	34	35	36	37	38	39	40	41	42	43	44

5 The serial number stored on the cartridge would be used to determine how many slots the code should shift to the left at the start the encrypting. For example, if the serial number ended with six, before starting of encryption, the code would be shifted to the left by six places. Table 2 illustrates the code table after the shift.

Table 2

symbol	A	B	C	D	E	F	G	H	I	J	K
code	7	8	9	10	11	12	13	14	15	16	17
symbol	L	M	N	O	P	Q	R	S	T	U	V
code	18	19	20	21	22	23	24	25	26	27	28
symbol	W	X	Y	Z	0	1	2	3	4	5	6
code	29	30	31	32	33	34	35	36	37	38	39
symbol	7	8	9	.	,	;	:	+	-	x	
code	40	41	42	43	44	1	2	3	4	45	6

5 The selected user password then is used to determine after how many symbols the code should again shift to the left. As an example, if the password were ROSE, then using the codes from Table 2, the numeric statement for rose would be 24212511. When the corresponding numbers are added together until reaching, a number between 1 and 10, the number reached in our example is 9 [18.9]. So after every 9th letter, the codes would be shifted another 6 spaces to the left. After the encrypting of 9 letters, the codes would be set as set forth in Table 3.

Table 3

symbol	A	B	C	D	E	F	G	H	I	J	K
code	13	14	15	16	17	18	19	20	21	22	23
symbol	L	M	N	O	P	Q	R	S	T	U	V
code	24	25	26	27	28	29	30	31	32	33	34
symbol	W	X	Y	Z	0	1	2	3	4	5	6
code	35	36	37	38	39	40	41	42	43	44	1
symbol	7	8	9	.	,	;	:	+	-	x	
code	2	3	4	5	6	7	8	9	10	11	12

Because the fact that the encrypting tables are constantly shifting,
under this simple method, the phrase "My brown dog has fleas." would be encrypted
5 as follows:

19 31 6 8 24 21 29 20 6
16 27 19 12 20 13 31 18 24
23 19 37 11

Decoding using only Table 1, the coded phrase would read as follows:

10 S 4 F H X U 2 T F
P 0 S L T M 4 R X
W S . K

Without knowing other information, it would be very difficult to find a pattern that would allow one to decode the symbols.

Knowing the placement of the codes relative to the symbols at the start of the encryption process and the number of symbols between shifts, decoding an encrypted phrase is simply a reversal of the process applying each of the tables in reverse.

There are any number of other methods of dynamic encryption that use different methods to vary encryption codes as one proceeds through the data to be encrypted. The objective, of course, is to make decoding difficult by avoiding obvious patterns associated with conventional language and number usage, so one would simply select a suitably difficult to decode dynamic encryption method.

In another embodiment, the unique dynamic encryption and decryption algorithms and formulae are generated using a unique combination of the personal signature (identification number) of the user, the serial (registration) of the storage medium and the registration number associated with the master data file being copied and encrypted using commercially available encryption programming or services. The electronic copies of the encryption and decryption formulae so generated are then stored at the central information storage facility for later use. With this embodiment of the invention, a unique set of encryption and decryption algorithms or formulae are so created for each separate copy of IP file requested. A copy of the unique decryption algorithm and formula is stored on nonvolatile memory on the cartridge (user storage medium) for later use. The nonvolatile memory on the cartridge (user storage medium) is so encased and configured that any attempt to read data stored thereon in an unauthorized manner will cause the memory to become totally unreadable and unusable. The unique encryption algorithm and formula so created are used to dynamically encrypt the selected IP file as the file is being copied and downloaded to the user cartridge (user storage medium) for storage and later use.

At the same time, a unique electronic header file is created for association with the encrypted copy of the requested IP file. The header file contains the user registration information, the rules and restrictions on use of the encrypted requested IP file and the electronic address for finding the appropriate decryption
5 algorithm and formula and the location of the applicable CCS for decryption and use of the data within the encrypted IP file. After generation, the header file is wrapped within the encrypted IP file as the dynamically encrypted file is being generated, is attached to the electronic copy of the encrypted file as a separate part, or is written as a separate distinct data file that is electronically linked to the encrypted IP file. The
10 manner in which the special header file is attached to the encrypted IP file depends upon the desired level of security and the hardware configuration.

E. Tamper Protection

In one embodiment of the present invention, access to the information is monitored, or recorded, to determine attempted unauthorized access to the
15 information. If an unauthorized access is recorded, or stored, onto a user's medium, for example cartridge 374, the next time that user attempts to download additional information to cartridge 374, an unauthorized access message may be transmitted to notify the appropriate party, for example the cashier. As a result of the unauthorized access message, the cashier may revoke user's cartridge 374, notify the proper
20 authorities, or record an entry into the user's account for future action. More specifically and in one embodiment, the unauthorized access is determined by first reading, or recording, the specific identification data from the information requester, or receiver. If the data provided by the information receiver is determined to not match, i.e., is unequal, predefined values, the unauthorized access message is
25 recorded and information exchange is prevented. The data determination may be completed using known comparison hardware and/or software functions.

Additionally, the unauthorized access message may be generated if a user having an incorrect purpose, or authorization code, attempts to access

5 unauthorized information. For example, in a corporate environment, if a user attempted to access information for which the user did not have the proper authorization code, an unauthorized access message is generated and may be sent to, for example a system administrator or a security official. Different level of unauthorized access messages may also be generated. For example, a high level message may be generated if a user attempts to decrypt the information stored in various locations within the system, for example IP Bank 302 using an unauthorized device. A lower level message may be generated if a remote user has attempted to access data that is one level above that user's authorized level.

10 G. Other Embodiments

In another embodiment, IP Bank 302 may be configured to capture and exchange real-time information. For example, as a professor presents material to students in a classroom, the professor's presentation may be captured and converted into copyrighted text and exchanged with remote users. This conversion may be completed using known voice to text conversion systems using a known computer system. The professor's presentation may be supplemented with previously prepared, or concurrently prepared, written text. The text may be digitized and properly integrated into the text using known methods. Remote users may receive information from the professor's lecture in real-time as the material is presented or may receive the information at a later time. Remote users receive only that information which the remote user is authorized to receive from IP Bank 302 as described above.

In yet another embodiment IP Bank 302 is configured to receive audio, video, and/or computer software code. For example and in one embodiment shown in Figure 13, IP Bank 302 is coupled to a Video Cassette Recorder (VCR) 600, a stereo system 610 including a cassette recorder/player 620, a Compact Disk (CD) or DVD-X player and/or recorder 630, a television 640, and a computer 650. As described above, authorized information is received from IP Bank 302, and in one embodiment is stored to a storage device, for example a memory device 660. The

memory device 660 may be a plurality of memory cells, for example Read Access Ram (RAM), Read Only Memory (ROM), a rotating storage unit, i.e., a hard disk, a magnetic storage medium, i.e., magnetic tape, or other storage media, for example an optical storage medium. After the remote user has selected the appropriate information to receive from IP Bank 302, the information is stored in device 660. Device 660 is configured to transfer the stored information to the selected playback device, i.e., Video Cassette Recorder (VCR) 600, stereo system 610, cassette recorder/player 620, CD or DVD-X player/recorder 630, television 640, or computer 650. For example, in one embodiment, the remote user downloads, or receives, the entire contents of a top ten music album. The contents of the album are stored in device 660. As described above, the information may be permanently stored or may be stored for a fixed period of time or number of uses. After downloading the information to device 660, the remote user may transfer the information to stereo system 610 for listening. In another embodiment, the information may be transferred to, or through, device 660 to one of the other components, i.e., cassette recorder 620, VCR 610, CD or DVD-X recorder 630, or computer 650. To limit unauthorized copying or playback, the information may be playback using only those components, i.e., cassette recorder 620, VCR 610, CD recorder 630, or computer 650, coupled to device 660. For example, the remote user may download a feature movie by saving the movie on a tape using VCR 610. The remote user may then playback the movie as authorized as long as the tape is playback in VCR 610 that is coupled to device 660. Similarly, the remote user may download a software program so that the information is stored in device 660 or in a storage medium in computer 650. Depending upon the authorization code of the software, the program may be configured to execute only from computer 650 when computer 650 is coupled to device 660.

The above-described system facilitates controlled and monitored exchange of information between many types of information owners, distributors, and users. By using the described system, a user may obtain many types of authorized information. The user may, as determined by the information owner, purchase, rent,

or obtain without charge, the authorized information. The information, in one embodiment, is encrypted using various levels, or complexities, of encryption to prevent unauthorized access. The level of encryption depends upon a variety of factors or variables, for example, the economic life of the information. For example, 5 IP Bank 302 may include information representing a reference dictionary and a top ten music album. Information from the reference dictionary and the album may have the same or different levels of encryption. In addition, students from a determined class may access the reference dictionary information without charge as the result of the school purchasing an unlimited use copy of the information, however, those same 10 students would be required to purchase any information downloaded from the album. Additionally, the type of access may differ for different portions of the information. For example, a first track of the album information may be coded so that anyone may download the information without charge, however, the remaining tracks of the album information may be coded to require payment to download.

15 In still another embodiment, and referring to Fig. 14 and Fig. 15, the present invention relates to a storage and retrieval system that is vendor, product, and IP independent. This embodiment provides an object-based system that packages any type of data on a network. The type of data in the package is immaterial to this embodiment. However, accurate, timely, and secure delivery is ensured by the 20 facilities provided by this embodiment of the present invention. Each separate IP is stored in digital form, and a consistent interface is used for its delivery. Various levels of standardized encryption are available. IP so encrypted is distributed and read in a uniform manner.

25 One of the facilities provided is one or more central data centers or central data storage facilities 100. Each central data center 100 acts in conjunction with any others present in the network to store and control delivery of IP from IP content providers 707. In some embodiments, additional data centers 702 are provided, each comprising one or more computers acting in conjunction with each

other. Computers 704 need not be in the same location. Each data center 702 either services one or more local clients 706 or acts as part of central data center 100.

Local public devices, or kiosks 708, are networked to and controlled by the data center. Kiosks 708 are examples of interfaces or access ports by which
5 consumers have access to central data center 100 and by which they acquire desired IP's. Besides kiosks, other examples of access ports are computers (mainframe, desktop, and laptop), personal digital assistants (PDAs), electronic books or ebooks, modems, and other devices designed for accessing electronic networks such as telephone, Lansat, Internet, intranet, and cable networks for electronic transfer of
10 digital data.

A security encryption compression (SEC) module 710 is provided in each consumer product or user device 712 to control access to, and use of IP's. To obtain an IP, a consumer must have a registered SEC module 710 and an authentication password that is recognized by kiosk 708. In addition, the consumer
15 must have a data storage medium 714 to hold a requested IP after downloading. SEC module 710 will securely store the IP for use in an appropriate manner. SEC module 710, in one embodiment, provides its own user interface (for example, a screen and/or speakers, etc.). In another embodiment, SEC module 710 provides one or more external adapters 716A, 716B, to provide a signal for display by one or more other
20 user devices 712A, 712B, 712C, 712D. External devices 712A, 712B, 712C, 712D, communicate with SEC module 710 through, for example, infrared ports, RCA plugs, headphone jacks.

SEC module 710 provides security through an "onion" approach, i.e., one that is made up of multiple protection layers that surround the IP. A first such
25 layer is a hardware layer. SEC 710 comprises, in one embodiment, a chipset having a unique serial number, nonvolatile random access memory (NV-RAM) 718, read-only memory (ROM) 720, and a programmable logic controller/processor 722 with electrically erasable programmable read only memory (EEPROM) 724. The unique

serial number provides part of a public/private encryption key along with a user's access code stored in NV-RAM 718 and the IP being accessed. Each IP is specifically encoded to work only with a specified piece of registered hardware accessed by a specific user's security code. For some IP products and in some
5 embodiments, transfer from one SEC module 710 to another is possible, but only through an authenticating interface such as a kiosk. Hardware protection provided by SEC module 710 is also the first authentication required by the network before it permits IP to be exchanged. If an SEC module 710 is bypassed to obtain direct access to storage medium 714, the physical interruption is archived and transmitted to the
10 network during a subsequent communication, for example, the next communication.

A second layer of security is provided by SEC 710 firmware. This layer is provided by a programmed EEPROM 724 that decrypts stored IP flowing to an output device. Public/private key encryption is dynamically controlled by each IP. In one embodiment, this control is monitored as well as modified each time SEC
15 module 710 communicates with the network.

A third layer of security is in the IP itself. It is encrypted and compressed while downloading to storage medium 714 in an SEC module 710.

Information obtained from a participating publisher 707 in digital form is enveloped with a uniform electronic signature that uniquely marks the version,
20 creating a master copy for distribution. The signed master copy is then encrypted using a custom encryption algorithm and stored in a centralized data storage facility 100 or facilities (depending upon the size of the distribution network) for later retrieval. Each stored work has a title, which is added to a master table of contents or index 726.

25 Central storage facility 100 is electronically accessed by a user via an access port using secure, encrypted application-layer protocols. These protocols are independent of network transport and physical layer protocols, and thus, usable on any transmission network. Utilizing these application-layer protocols, the user

communicates with and receives instructions from central storage facility 100 for acquiring an electronic version of one or more selected IP's. In preparation for downloading the selected IP's, central storage facility 100 further encrypts 728 the IP's selected for distribution in accordance with a level of encryption selected by the publisher of the IP.

Encryption is provided in several different forms. For example, public/private encryption keys are used for data storage. Watermarking is used when sending data to an unsecured device such as a headphone, a television set, or a word processor. For example, a watermark code is injected by a digital-to-analog (D/A) converter into an analog output signal, which, when analyzed, identifies a source of the IP and its purchaser or licensee. The D/A converter code is determined by the serial number and personal ID code in NV-RAM 718 and from coding in the IP itself, thereby uniquely identifying a combination of user, device, and IP. The secure transfer of IP from the network is done, for example, via RSA encoding, which is described in detail in U.S. Patent No. 4,405,829. This secure transfer ensures that each packet is uniquely encrypted as well as monitored for possible hijacking and other data stream tampering.

In this embodiment, compression is provided both to maximize storage capabilities and to provide another layer of security for the IP.

SEC 710, in one embodiment, is an integrated storage device, with at least one of a hard disk, a RAM disk, an NV-RAM card (e.g., a compact flash memory card) or other storage medium as medium 714. One or more interfaces 716A, 716B, 716C, 716D are provided between the integrated storage device 714 and one or more output devices 712A, 712B, 712C, 712D. SEC 710 is designed to permit storage device manufacturers to incorporate the chipset into their storage devices, so that the storage devices can accept IP from the network.

In one embodiment, SEC 710 comprises a storage medium 714, an SEC chipset 718, 720, 722, 724, a USB (universal serial bus) port 730 to interface

with a kiosk 708 for IP transfer, an A/D converter 716A, and a headphone jack 712A. Another embodiment provides both a USB and a USB2 connection. In yet another embodiment, SEC 710 includes a infrared (IR) port and an RCA jack. In still other embodiments, SEC 710 is built into a device such as a personal digital assistant
5 (PDA) or a single-function reader that serves as a device for reading digital IP or ebooks. In yet another embodiment, SEC 710 is built into a user device such as a laptop or desktop computer. For example, half of the hard drive of the computer can be set aside for a traditional operating system and the other half dedicated for storing IP, for example, user video, books, and audio. Still other embodiments include other
10 wireless couplings, iLink, Firewire, and/or IEEE 1394 connections, for example.

Encryption ensures protection for IP by allowing downloaded IP to be read only on a particular SEC 710, and only by a user knowing a particular password. This limitation is provided, for example, utilizing a decryption formula provided by a key generation company 732 such as Verisign, Inc. The key generation company
15 creates a formula that enables a decryption key to reverse the encryption 728 provided the network. This formula is sent during registration of an SEC module 710 and burned into NVRAM 718 or other limited access memory in one embodiment. In another embodiment, the formula is preprogrammed into SEC module 710. In one embodiment, the formula is stored in an encrypted form and decrypted by SEC 710
20 hardware only while it is being used. In one embodiment, the same formula is used by the network to encrypt IP at the central data center and by SEC 710 to decrypt the encrypted IP, using different keys determined by key generation company 732.

In one embodiment of the present invention, modern algorithms and appropriate key lengths are used to protect the IP when an IP file is initially created, during its distribution, and throughout its existence. In addition, system programming
25 regularly and automatically updates and changes encryption and decryption code algorithms, keys, and formulae both to recapture IP for which such protection has been broken or compromised and to inhibit the cracking or compromising of IP protection. The frequency and/or number of times that algorithms, keys, and/or

formulae are changed depends, in one embodiment, on one or more factors such as the length of time a user has rights to an IP, a level of security that is assigned to the IP, and a preassigned schedule that is based upon the level of security assigned to the IP.

5 Employing key generation company 732 independent of the network operator to generate keys is not necessary, but may be preferred by owners of IP who believe that extra security and accountability is provided by such companies. It is not necessary to employ the services of a key generation company, however. Thus, in one embodiment of the present invention, key generation facilities are provided within the network itself.

10 In one embodiment, operation of the network proceeds as follows. Digital IP of various types and from various sources 708 is obtained and stored at central data center 100. Users (or potential users) of the digital IP obtain a SEC module 710 with a unique serial number (USN) permanently "burned" into the electronics of the module. Each user also chooses a personal password (PP) to
15 identify himself or herself for registration. The user-selected PP, in some embodiments, is constrained within limits selected for security and practicality by the network operator. For example, in one embodiment, the PP is constrained to be no less than 6 characters and no more than 12.

20 A user with an SEC module 710 and a selected password then connects SEC module 710 to a kiosk 708 or other access port. SEC 710 transmits the USN and the user-selected PP to the network for registration, where the USN and PP are combined (for example, by concatenation or by a mathematical or logical operation) into a unique code (UC). In at least one embodiment, the UC is sent to a key generation company 732, for example, Verisign, Inc. The network itself also
25 generates a hidden private key (HPK) and sends it to key generation company 732. Key generation company 732 then computes a private encryption key (PEK) from the HPK and the UC and sends the PEK to the network, where it is used for encryption of IP as it is sent to the user of SEC 710. The user then enters the PP into SEC 710,

where it is combined with the USN (as combined by the network) to decrypt the IP for the user.

In one embodiment, servers 734 at the centralized storage facility or facilities 100 determine a most efficient cost effective path to distribute IP, including, but not limited to, books, music, video, computer software and multimedia, and automatically tracks, verifies end user tampering, freezes the end users access, and regulates distribution method requirements. The regulation varies for each account and industry. For example, a retail establishment will use a different set of rules requiring different distribution limitations, retrieval and viewing methods, and security and encryption specifications when distributing IP than an institution or an individual is likely to require. Also, methods of electronic distribution and security requirements vary in accordance with the types of IP that are distributed. For example, digital music transfer and retrieval is performed differently from digitally retrieving, and reading an electronic IP, or transferring, retrieving and viewing video. A portion of the verification programming is devoted to checking and verifying the readability of the IP file after being encrypted and downloaded to the user's storage medium. The verification programming also attempts to correct correctable errors automatically and indicates detectable, but uncorrectable, errors. If uncorrectable errors are found during verification, the downloading and encryption process is repeated for a predetermined number of times or until no uncorrectable errors are detected within the predetermined number of attempts. If uncorrectable errors remain after the predetermined number of attempts is exhausted, the user is directed to a service representative to provide assistance. In one embodiment, CCS is also programmed or otherwise configured to automatically verify the readability of an encrypted IP during access for reading and automatically attempts to correct any errors discovered that do not relate to functioning of the security mechanism. For example, CCS is programmed to detect and correct errors resulting from media deterioration. Such errors can result from passage of time or from repeated access or use.

In one embodiment, the current invention consolidates distribution all forms of IP into one network that is transparent to the end consumer. The network includes transparent links that feeds to various IP web-sites 736 requested information. When an end user orders IP works via a web site 736, central repository 100 storing the requested data links the request and transmits the IP to a user's computer 738 (or other apparatus) with a secure dynamic encryption code that adds another layer of security to the secondary repositories. Participating IP retailers' web sites, kiosks, or other terminals, depending upon the needs of particular retailers, are also transparently linked to the centralized repository. Thus, a retail customer can continue to purchase IP content from a favorite retailer's web site or store, while the retailer links to the central IP data repository "behind the scenes."

Depending upon the most efficient and cost effective distribution method, the one or more centralized repositories 100 are locally based, regionally based or centrally based. Each central repository tracks IP data and credits sales 740, 742 to retailer sites. Key locking is provided so that individual retailers have access to their own sales tracking information at the central depository, irrespective of whether sales transfers occurred via Internet, microwave, infrared, satellite, cable, telephone, or other medium.

One embodiment of the present invention is linked to various web sites 736 in a manner transparent to consumers. When purchases are made through various web sites, stores or organizations, corporations, etc., an earmarked transactional fee is tracked, accounted for, and distributed to that particular entity through a universal tracking system 740, 742. In this embodiment, the central information storage facility is a grouping of separate servers or storage facilities at one or more physical locations that are physically operated by one or more persons and that are linked together via a network arrangement and by appropriate software such that they appear to the user to be a single functioning storage facility. Retail companies are thus provided with income, irrespective of whether their customers receive downloaded IP products via a web site 736 of the retailer, from a high-bandwidth kiosk 708 at a traditional store, or

recorded on media by the retailer on demand at a retail establishment. Using this embodiment of the present invention, retailers can provide a useful, profitable service by distributing IP through in-store kiosks or on media recorded by the retailer on demand to customers lacking high-speed Internet connections. Providing this service
5 may be a decisive advantage to some retailers. The additional flexibility afforded retailers by the present invention may also help retailers avoid market base erosion as Internet bandwidth to the home continues to expand.

In one embodiment, IP's and/or links to IP's (i.e., pointers to another location where the IP is stored, such as on a server of an IP content provider 707) are
10 obtained from content providers 707. Referring to Fig. 16, when IP is stored 800, a format conversion 1001 is performed. Format conversion 1001 includes "system encryption," so that the IP is unreadable on other systems. When a request is received, personalized encryption 1002 is applied, followed by dynamic encryption
15 1003 as the IP is transmitted over a network 804. (Encryptions 1002 and 1003 are configured so that they can be performed at essentially the same time, even though a temporary storage 802 is shown between them.) In the case of an IP link, format conversion 1001 is triggered by a user request. The result is stored in a temporary storage 812 (rather than a permanent storage in central data center 100, as is the case when the IP is stored, rather than linked). Personalized encryption 1002 and dynamic
20 encryption 1003 is then applied before transmission over network 1003. (Temporary storage 814 is not necessarily different from temporary storage 802.) As the user downloads the IP, it is stored in user storage 714, where decryption of encryption algorithms 1003, 1002, and 1001 are performed, in the reverse order to which the encryption 1001, 1002, 1003 were applied. Decryption 1002 is performed from
25 temporary storage 806 into a secure RAM 808, using SEC 710. The content of secure RAM 808 is displayed 810, after encryption 1001 is decrypted. Storage 806 and 808 are erased when the IP is no longer needed. User storage 714 is also erased under some conditions, such as when tampering is detected.

The registration, security, tagging (or watermarking), verification and use analysis programming produces a data trail logging the creation and use of a particular IP file by a user, thereby providing evidence of accountability in the event of an actual or attempted breach of the security features associated with the IP file.

5 The registration, security, tagging verification and use analysis programming creates a permanent linking of a given copy of an IP file to a user requesting its creation and a user device used to create the copy in question. A permanent log of use of the IP file, in digital form, is stored in a separate memory storage area associated with the CCS programming and a copy of the log is periodically transferred to the central

10 information data storage facility for storage and later analysis.

As illustrated in Figure 1, in one embodiment of the invention, security is enhanced through the use of an external authentication device [302A] affixed or connected to the user's device and a related identifier. Examples of suitable, currently available external authentication devices include parallel port software locks,

15 iButtons[™] (Dallas Semiconductor Corp. of Dallas, TX) and "Smart Cards" plugged in via a pc-card. In such embodiment, upon registering a user is physically given an identification code disk, card or button that is encoded with user identification information that is electronically readable when the code disk, card or button is inserted in an appropriate slot associated with the external authentication device. If

20 the externally inserted electronic identification codes match the required codes for access, access to the encrypted files is allowed. If the externally inserted electronic identification codes do not match the required codes for access, access is denied.

In yet other embodiments of the present invention, user device 712 is a cellular or other wireless mobile telephone or wireless communications unit, for example, a cell phone with a wireless browser. With this type of user device, IP text

25 can be downloaded directly to the wireless communications unit via the appropriate cellular or other type of wireless mobile network. In one embodiment, for example, the IP text is written text, pictures, or other visual content that is displayed in a browser window. In another embodiment, the IP text is music that is downloaded

directly to a cellular or other wireless mobile telephone. (One embodiment of a / wireless telephone useful for this invention includes stereo headphones, or has a jack for such headphones.) The IP text undergoes dynamic encryption, as in other embodiments of the invention, and is decrypted, but not stored in the telephone or on
5 other media. When a call is received, an audible or other type of indication is provided on the wireless telephone user device 712, and the music is interrupted when the call is answered. In yet another embodiment, music is selectively streamed to headphones or speakers, saved to a recording medium in the wireless telephone user device 712, or both. Telephone 712 is configured so that the music can be selectively
10 saved via downloading to the recording medium when a call interrupts music being streamed for listening.

It will thus be seen that one or more embodiments of the present invention provide one or more IP consolidation features such as overall "mothership" IP encryption, open web selection and browser access, user screening ID and control,
15 transfer verification, transaction purchase, attachment of secure rules, functionality and user allowances, auditing, and feedback and updating. While the present invention has been described with respect to specific embodiments, many modifications, variations, substitutions, and equivalents will be apparent to those skilled in the art. Accordingly, the invention is to be considered as limited only by the
20 spirit and scope of the appended claims.

CLAIMS:

1. Apparatus for facilitating obtaining text of an IP, comprising:

a storage device having stored therein text of a plurality of IP, wherein the text includes electronically stored representations of at least one member of the group consisting of printed text works, movies, films, video presentations, television programming, music, audio works, audio presentations, radio programs, graphic material, art works, plays, operas, novels, writings, photographs, pictures, images, advertising copy, software, and portions and combinations thereof;

a processor connected to said storage device, said storage device further having stored therein a program for controlling said processor, said processor operative with the program to:

receive an IP selection request;

receive a user identification associated with the IP selection request;

and

output encrypted text of the selected IP if the user identification and IP selection are valid utilizing a determined level of IP encryption.

2. Apparatus in accordance with Claim 1 wherein said processor is further operative with the program to purpose encode the text of the selected IP so as to limit a purpose which access by the user to the text is authorized.

3. Apparatus in accordance with Claim 1 wherein said processor is further operative with the program to encrypt the text of the selected IP utilizing an identifier associated with the user.

4. Apparatus in accordance with Claim 1 configured to electronically associate instructions provided by an IP supplier and relating to allowed

use, access, and pricing of IPs supplied by the supplier with the IPs supplied by the supplier.

5 5. Apparatus in accordance with Claim 4 further configured to receive use and access choices from the user associated with the IP selection request, and to use the instructions provided by the IP supplier and associated with the text of the selected IP to generate an electronic IP instruction file linked with the outputted text of the selected IP to regulate use of and access to the outputted text.

10 6. Apparatus in accordance with Claim 1 further configured to receive a determined level of security including the determined level of encryption from a supplier of the selected IP, the determined level of security being selected from a range of security levels.

 7. Apparatus in accordance with Claim 1 wherein said determined level of IP encryption is based on at least an economic value of the selected IP.

15 8. Apparatus in accordance with Claim 1 wherein said processor is further operative with the program to encrypt the text of the selected IP utilizing a unique identification number associated with at least one of a user device and user data storage medium used to download or store the encrypted text of the selected IP.

20 9. Apparatus in accordance with Claim 1 wherein said processor is further operative with the program to cooperate with a user device being used to download and read a selected IP text, to restrict and regulate operation of the user device to limit reading and copying of the selected IP text.

25 10. Apparatus in accordance with Claim 9 wherein said processor is further operative with the program and the user device to determine whether IP texts downloaded to the user device have been used in an unauthorized manner, and to at least one of restrict or deny access to such IP texts when such unauthorized use is determined to have occurred.

11. Apparatus in accordance with Claim 10 wherein said processor is further operative with the program and the user device to cause the user device to permanently erase an IP text from memory of the user device when such unauthorized use is determined to have occurred.

5 12. Apparatus in accordance with Claim 9 wherein said processor is further operative with the program and the user device to cause the user device to at least one of temporarily restrict or deny access to an IP text in a memory of the user device.

10 13. Apparatus in accordance with Claim 1 further comprising at least one local unit communicatively coupled to said processor, said local unit comprising a memory for storing, in electronic form, information transmitted to said unit from said processor, and a local unit processor for controlling transfer of information stored in said unit to electronic storage media of system users, said local unit configured to encrypt the information when the information is to be transferred to
15 the electronic storage media, said local unit configured to encrypt the information utilizing a determined level of information encryption.

14. Apparatus in accordance with Claim 13 wherein the information stored on a user's storage medium comprises a personal signature code number and serial number.

20 15. Apparatus in accordance with Claim 13 further configured to receive a determined level of security, including the determined level of encryption, from a supplier of the selected IP, the determined level of security being selected from a range of security levels.

25 16. Apparatus in accordance with Claim 15 wherein said determined level of information encryption and security is based upon at least an economic value of the selected IP

17. Apparatus in accordance with Claim 16 wherein the level of information encryption is not equal to the level of the IP encryption.

18. Apparatus in accordance with Claim 13 wherein the memory of the local unit is configured to regulate the use of information stored therein in encrypted or decrypted form so long as an electronic connection exists between the local unit and a user data storage medium, and to permanently erase selected files from the memory of the local unit whenever the electronic connection is no longer active.

19. Apparatus in accordance with Claim 1 wherein said processor is further operative with the program to determine attempted unauthorized access to the output copies of the text of an IP.

20. Apparatus in accordance with Claim 1 further configured to receive information identifying an establishment from which the IP selection request originated, and to tag the encrypted text of the selected IP with an identification of the establishment.

21. Apparatus in accordance with Claim 20 wherein said processor is further operative with the program to utilize the tag to trace IP text to an establishment.

22. Apparatus in accordance with Claim 1 wherein said processor is further operative with the program to determine and store information relating to usage of the encrypted text by the user and which will cause said usage information to be encrypted and stored in a segregated section of the memory associated with the user storage medium or local unit and which will cause said usage information in encrypted form to be transmitted to the central data storage facility when said local unit and/or user data storage unit are later electronically connected to the central data storage facility.

Auditing

23. A method for operating a computer to obtain text of an IP, ~~✕~~
comprising:

inputting into the computer an IP selection request;

5 inputting into the computer a user identification associated with the IP
selection request;

inputting a unique identification number associated with at least one of
a user device or user storage medium being used to at least one of download or store
an encrypted copy of a selected IP corresponding to the IP selection request,

10 generating a unique encryption algorithm and a corresponding
decryption algorithm, using the user identification and at least one member of the
group consisting of: (a) a digital identification number associated with at least one of
the user device or a user data storage medium, and (b) a digital identification
associated with the selected IP;

15 communicating encrypted text of the selected IP to the user device for
storage on the user storage medium using the unique encryption algorithm for
encryption, wherein text of the selected IP comprises electronic representations of at
least one member of the group consisting of: printed text works, movies, films, video
presentations, television programming, music, audio works, audio presentations, radio
programs, graphic material, art work, plays, operas, novels, writings, photographs,
20 pictures, images, advertising copy, software, and portions and combinations thereof;

generating a header associated with the encrypted text of the selected
IP file that contains, in digital form, a user identification, an identification of at least
one of the user device or user data storage medium ^{u n} ~~one~~ _u which the encrypted digital
copy of the selected IP is to be stored, a usage authorization indication, and an
25 electronic address of the corresponding decryption algorithm;

validating the user identification, authorization indication, and IP selection, after the unique encryption algorithm and corresponding decryption algorithm have been generated; and

5 decrypting the digitally encrypted text of the selected IP using the corresponding decryption algorithm, conditioned upon said validation.

24. A method in accordance with Claim 23 wherein inputting to the computer an IP selection comprises the steps of:

selecting at least one of the IP in the memory storage; and

selecting the text of at least a portion of each the selected IP.

10 25. A method in accordance with Claim 24 wherein inputting to the computer an IP selection further comprises the steps of:

determining if more than one portion of an IP is selected; and

if more than one portion of an IP is selected, then combining the selected portions.

15 26. A method in accordance with Claim 23 further comprising the step of generating tracking information corresponding to the selected portions and the selected IP.

20 27. A method in accordance with Claim 23 further comprising the step of purpose-encoding the encrypted text of the selected IP so as to limit a purpose for which access by the user to the text is authorized. UR

28. A method in accordance with Claim 23 further comprising the step of determining a level of IP encryption.

29. A method in accordance with Claim 28 wherein determining a level of IP encryption comprises the step of selecting at least one of a plurality of levels of encryption code.

5 30. A method in accordance with Claim 29 wherein the step of generating the unique encryption algorithm and corresponding decryption algorithm comprises the step of obtaining the unique encryption algorithm and corresponding decryption algorithm from an independent key supplier.

10 31. A method in accordance with Claim 30 further comprising the step of storing copies of the obtained encryption and decryption algorithms in a memory of a central information storage facility for later use.

32. A method in accordance with Claim 30 wherein the at least one of the user device or user storage medium has a write once, read many times (WORM) memory unit contained therein, the WORM memory being configured to become inoperable when unauthorized access thereto is attempted;

15 said method further comprising the step of storing a digital representation of the unique encryption algorithm and of the corresponding decryption algorithms in the WORM memory for later use.

20 33. A method for operating a processor communicatively coupled to a network to obtain text of an IP, the network being coupled to a memory storage having stored therein text of a plurality of IP wherein the text of an IP comprises electronic representations of at least one member of the group consisting of: printed text works, movies, films, video presentations, television programming, music, audio works, audio presentations, radio programs, graphic material, art work, plays, operas, novels, writings, photographs, pictures, images, advertising copy, computer games,
25 video games, software, and portions and combinations thereof; said method comprising:

determining an IP selection request;

inputting into the processor the IP selection request;

inputting into the processor a user identification associated with the IP selection request;

5 inputting a unique identification number associated with at least one of a user device or a user data storage medium in which an encrypted electronic copy of the text of the selected IP is to be stored;

outputting encrypted text of the selected IP if the user identification and IP selection request are valid utilizing a determined level of encryption.

10 34. A method in accordance with Claim 33 wherein determining the IP selection request comprises the steps of:

reviewing the IP in the memory storage;

selecting the text of at least a portion of at least one of the IP in the memory storage.

15 35. A method in accordance with Claim 33 wherein determining the IP selection request comprises the steps of:

selecting at least one IP in the memory storage;

selecting the text of at least a portion of each selected IP in the memory storage;

combining the selected portions of the selected IP.

20 36. A method in accordance with Claim 35 wherein selecting the text of at least a portion of each selected IP in the memory storage comprises the step of selecting the entire IP.

37. A method in accordance with Claim 35 wherein selecting the text of at least a portion of each selected IP in the memory storage comprises the step of selecting at least one word of each selected IP.

5 38. A method in accordance with Claim 35 wherein selecting the text of at least a portion of each selected IP in the memory storage comprises the step of selecting at least one section of each selected IP.

39. A method in accordance with Claim 35 further comprising the step of generating tracking information corresponding to the selected IP in the memory storage.

10 40. A method in accordance with Claim 39 further comprising the step of generating tracking information corresponding to the selected portions of the selected IP.

41. A method in accordance with Claim 39 further comprising the step of generating tracking information relating to usage of selected IP relative to a retail establishment location.

15

42. A method in accordance with Claim 33 further comprising the step of purpose encoding the encrypted text of the selected IP so as to define a purpose of use which access by the user to the text is authorized.

43. A method in accordance with Claim 33 further comprising the step of determining unauthorized access to the encrypted text.

20

44. A method in accordance with Claim 33 wherein the step of outputting encrypted text of the IP comprises the step of generating a unique encryption algorithm and a corresponding decryption algorithm using the user identification and at least one member of the group consisting of: (a) the digital identification number associated with at least one of the user device or the user data storage medium, and (b) a digital identification associated with the selected IP.

25

45. A method in accordance with Claim 44 wherein the step of generating the unique encryption algorithm and corresponding decryption algorithm comprises the step of obtaining the unique encryption algorithm and corresponding decryption algorithm from an independent key supplier.

5 46. A method in accordance with Claim 45 further comprising the step of storing copies of the obtained encryption and decryption algorithms in a memory of a central information storage facility for later use.

47. A method in accordance with Claim 46 wherein the at least one of the user device or user storage medium has a nonvolatile memory contained
10 therein, the nonvolatile memory being configured to become inoperable when unauthorized access thereto is attempted;

said method further comprising the step of storing a digital representation of the unique encryption algorithm and of the corresponding decryption algorithms in the nonvolatile memory for later use.

15 48. A method in accordance with Claim 45 wherein said step of outputting encrypted text of the selected IP comprises the step of using the unique encryption algorithm to encrypt a digital copy of the selected IP as the copy is being made and downloaded to a user's data storage medium.

20 49. A method in accordance with Claim 33 further comprising the step of outputting, in association with the encrypted copy of the selected IP, information from which at least one member of the group consisting of: (a) access authorization codes, (b) usage authorization codes, and (c) a decryption algorithm for decrypting the encrypted text of the IP can be determined.

25 50. Apparatus for facilitating obtaining text of an IP, wherein the text includes electronic representations of at least one member of the group consisting of: printed text works, movies, films, video presentations, television programming, music, audio works, audio presentations, radio programs, graphic material, art work,

plays, operas, novels, writings, photographs, pictures, images, advertising copy, computer games, video games, software, and portions and combinations thereof; said apparatus comprising:

a storage device having stored therein text of a plurality of IP;

5 a processor connected to said storage device, said storage device further having stored therein a program for controlling said processor, said processor operative with the program to:

receive an IP selection request;

receive a user identification associated with the IP selection request;

10 and

dynamically encrypt text of the selected IP as the text is output from the apparatus, using an encryption algorithm.

51. Apparatus in accordance with Claim 50 wherein said apparatus is configured to communicate via a network connection selected from the group
15 consisting of: electronic cable connections, wired connections, commercial telephone connections, commercial cable network connections, cellular and other wireless mobile network connections, infrared connections, microwave connections, radio wave and other wireless connections, television wave connections, local device generated infrared signal connections, laser connections, and connections allowing for
20 transfer of data in digital form from one point to another, and combinations thereof.

52. Apparatus in accordance with Claim 50 wherein said processor is further operative with the program to purpose encode the text of the selected IP so as to limit a purpose which access by the user to the text is authorized.

53. Apparatus in accordance with Claim 50 wherein said processor
25 is further operative with the program to access encode the text of the selected IP so as to regulate access to the encrypted selected IP.

54. Apparatus in accordance with Claim 50 wherein said processor is further operative with the program to receive instructions from an IP supplier relating to permissible use and access to IP texts supplied by the IP supplier, to associate said instructions with corresponding IP texts, to receive usage and access choices from the user in conjunction with the IP request, and to use said instructions associated with said texts to generate an electronic instruction data file with the requested IP, the instruction data file having instructions readable by a user device to regulate use of and access to the selected IP after it is downloaded and stored on a user device or storage medium.

55. Apparatus in accordance with Claim 50 further comprising at least one local unit communicatively coupled to said processor, said local unit comprising a memory for storing, in electronic form, information transmitted to said unit from said processor, and a local unit processor for controlling transfer of information stored in said unit to electronic storage media of system users, said local unit configured to encrypt the information when the information is to be transferred to the electronic storage media, said local unit configured to encrypt the information utilizing a determined level of encryption.

56. Apparatus in accordance with Claim 50 further configured to:

communicate, to a key generation service via a communication network, information relating to at least one member of the group consisting of: (a) a digital identification number associated with at least one of a user device or a user data storage medium, and (b) a digital identification associated with the selected IP; and

receive an encryption algorithm corresponding to the selected level of encryption from the key generation service via the communication network.

57. Apparatus in accordance with Claim 50 wherein said processor is further operative with the program to encrypt the text of the selected IP utilizing a

unique identification associated with at least one of a user device or a user data storage medium being used to download or store the encrypted text.

58. Apparatus in accordance with Claim 50 wherein said processor is further operative with the program to restrict and regulate the operation of a user device and user data storage medium that receives the encrypted text of the IP by restricting ability of the user device to perform at least one operation selected from the list consisting of: (a) printing of the IP, (b) copying of the IP, and (c) permanently storing decrypted text of the IP to another storage medium; and to cause selected information to be permanently erased from memory associated with the user device when a connection between the user device and a user data storage medium on which encrypted data files are stored is broken.

59. Apparatus in accordance with Claim 58 wherein said processor is further operative with the program to periodically change encryption and decryption algorithms and formulae associated with a copy of a selected IP text as a function of at least one of passage of time and use of the selected IP text.

60. Apparatus in accordance with Claim 50 wherein said processor is further operative with the program to receive a user-provided access code and generate a system access code, and to restrict and regulate access to encrypted text of the IP by encrypting text of a selected IP using the user-provided access code and generated system access code so that both are required to decrypt the text of the selected IP.

61. Apparatus in accordance with Claim 50 wherein said processor is further configured to receive, in encrypted form, usage and access information relating to a user device for reading electronic texts of IP when the user device is communicating with the apparatus.

62. A method for distributing intellectual properties (IP's) in digital form, said method comprising the steps of:

*

valid?

obtaining information from a plurality of publishers, the information being selected from the group consisting of IP's, links to IP's, and combinations of IP's and links to IP's;

encrypting IP's obtained from the publishers;

5 storing the encrypted IP's in a storage facility;

indexing the stored, encrypted IP's and links to IP's;

transmitting at least selected portions of the index to customers via a network;

10 receiving selective electronic requests for the IP's from the customers via the network;

transmitting requested IP's to customers in response to their electronic requests, including dynamically encrypting the IP's; and

recording transaction fees for the transmissions.

15 63. A method in accordance with Claim 62 further comprising the step associating an electronic signature personalized to a requesting customer with the IP's transmitted to the requesting customer in response to the requesting customer's request.

64. A method in accordance with Claim 62 further comprising the steps of:

20 maintaining an accounting of IP's transmitted utilizing the electronic signatures;

billing customers for IP's transmitted in response to their requests, and crediting publishers for publisher's IP's transmitted to customers.

65. A method in accordance with Claim 62 wherein the IP's include compilations of portions of IP.

66. A method in accordance with Claim 62 wherein the selective electronic requests for the IP's include requests for information available as IP links,
5 and wherein transmitting requested IP's to customers, including dynamically encrypting the IP's comprises the steps of:

retrieving a copy of an IP corresponding to an IP link;

processing the copy of the IP so as to make it readable on a device of the requesting customer; and

10 dynamically encrypting the processed copy of the IP.

67. A method in accordance with Claim 62 wherein transmitting the requested IP's comprises the step of transmitting instructions to corresponding requesting customers for acquiring the requested IP's.

68. A method in accordance with Claim 67 wherein transmitting
15 instructions for acquiring the requested IP's includes transmitting one or more instruction selected from the set consisting of: how to buy the requested IP's, how to rent the requested IP's, how to pay for the requested IP's, and how to read the requested IP's.

69. A method in accordance with Claim 67 wherein transmitting
20 instructions for acquiring the requested IP's includes transmitting hardware and software requirements for reading the requested IP's.

70. A method in accordance with Claim 62 further comprising the step of encrypting at least one of the requested IP's with a customer-specific algorithm to allow the at least one requested IP to be read on more than one user
25 device having an electronic identification associated with the customer.

71. A method in accordance with Claim 62 wherein transmitting the encrypted IP's comprises further encrypting the selected, encrypted IP's in accordance with a level of encryption selected by a publisher of the IP.

5 72. A method in accordance with Claim 62 wherein said step of receiving selective electronic requests for the IP's from customers via a network comprises receiving selective electronic requests from end users of the IP's via a public network.

73. A method in accordance with Claim 72 wherein the public network comprises the Internet.

10 74. A method in accordance with Claim 62 wherein said step of receiving selective electronic requests for the IP's from customers via a network comprises receiving selective electronic requests transmitted from network terminals located in retail establishments.

15 75. A method in accordance with Claim 74 wherein said step of receiving selective electronic requests transmitted network terminals located in retail establishments comprises receiving electronic requests sent by at least one member of the group consisting of: employees, proprietors, and combinations thereof, of the retail establishments.

20 76. A method in accordance with Claim 75 and further comprising the steps of maintaining an accounting of IP's transmitted utilizing electronic signatures personalized to requesting customers, and crediting IP sales to the retail establishments.

77. A method for uniform storing, encryption, and electronic distribution of digitized intellectual property (IP) comprising:

25 creating an electronic index of a collection of digitally-stored intellectual property;

generating authorization information unique to a user when the user accesses a central digital IP storage facility;

regulating access to digitized IP at the central storage facility to authorized users;

5 transmitting digitized IP from the digital storage facility to authorized users, including dynamically encrypting the digitized IP;

generating a unique registration number for each transmitted digitized IP associating the authorized user downloading the digitized IP and a user storage device receiving the transmitted IP.

10 78. A method in accordance with Claim 77 further comprising the steps of recording and storing, as a data file stored on a central data storage file, information identifying and relating users requesting transmission of a digitized IP with the requested, digitized IP.

15 79. A method in accordance with Claim 78 further comprising the step of providing access to the information identifying and relating the users to the transmitted digitized IP only to selected individuals.

20 80. A method in accordance with Claim 78 further comprising the step of indexing the information identifying and relating the users to the transmitted digitized IP in accordance with one member of the group consisting of: date, user, and digitized IP identification.

81. A method in accordance with Claim 78 wherein dynamically encrypting the transmitted IP comprises generating and attaching a time code tag indicating at least one of: (a) a length of time that access to a transmitted IP will be allowed, and (b) a time period that access to a transmitted IP will be allowed.

25 82. A method in accordance with Claim 78 wherein dynamically encrypting the IP comprises changing assigned encryption and decryption algorithms

for a transmitted IP stored in a user device or on a storage medium, said changing occurring in accordance with a function of at least one of use of the transmitted, stored IP, and time.

5 83. A wireless telephone communications device configured to: ** fixed*
board.
communicate over a wireless network;
receive, decrypt, and play an interruptible, encrypted stream of music; *(valid)*
and

to interrupt the stream of music when a call is received and answered.

10 84. A device in accordance with claim 83 further having storage for
recording the stream of music.

85 A device in accordance with Claim 84 further configured to
selectively record music into storage when the stream of music is interrupted.

86. A device in accordance with Claim 83 configured for stereo
playing of the music stream.

15 87. A method for transmitting IP to a mobile station comprising the
steps of:

obtaining information from a plurality of publishers, the information
being selected from the group consisting of IPs, links to IPs, and combinations of IPs
and links to IPs,

20 receiving selective electronic requests for the IPs from customers via a
mobile, wireless network;

transmitting requested IPs to customers in response to their requests
via the mobile, wireless network, including dynamically encrypting the IPs; and

recording transaction fees for the transmissions.

88. A method in accordance with Claim 87 wherein the IPs
comprise at least one member of the set consisting of movies, films, video
presentations, television programming, music, audio works, audio presentations, and
5 radio programs; and wherein the step of transmitting requested IPs to customers
comprises the step of synchronously transmitting requested IPs to customers.

89. A method in accordance with Claim 87 wherein the IPs
comprise at least one member of the set consisting of movies, films, video
presentations, television programming, music, audio works, audio presentations, and
10 radio programs; and wherein the step of transmitting requested IPs to customers
comprises the step of asynchronously transmitting requested IPs to customers.

90. A method in accordance with Claim 87 wherein the wireless
network is a cellular network.

91. A method in accordance with Claim 87 further comprising the
15 step of transmitting instructions with the requested IPs containing limitations
pertaining to the use of the requested IPs by the customers.

92. A method in accordance with Claim 91 wherein the instructions
include instructions pertaining to the autoerasure of the requested IPs.

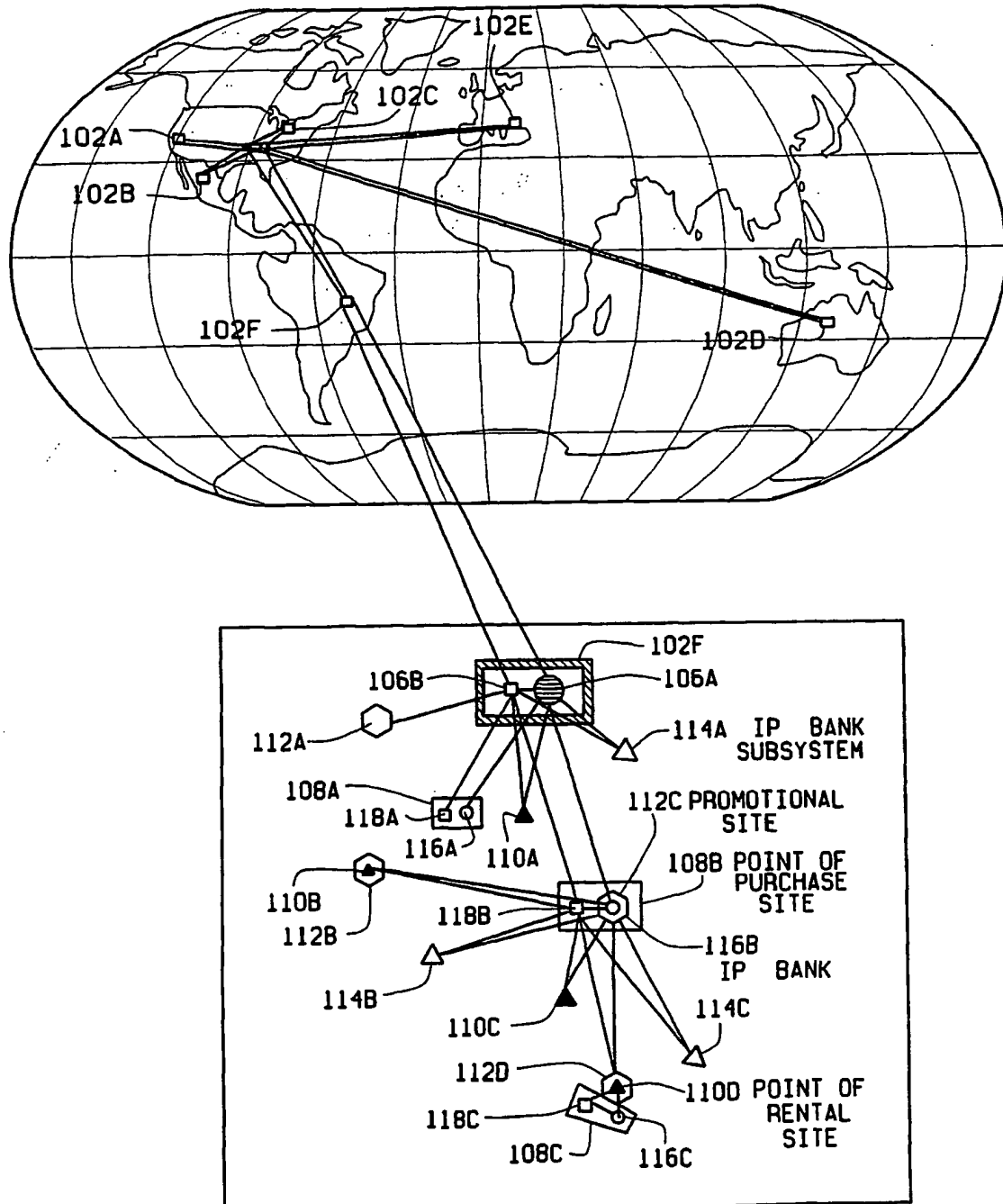
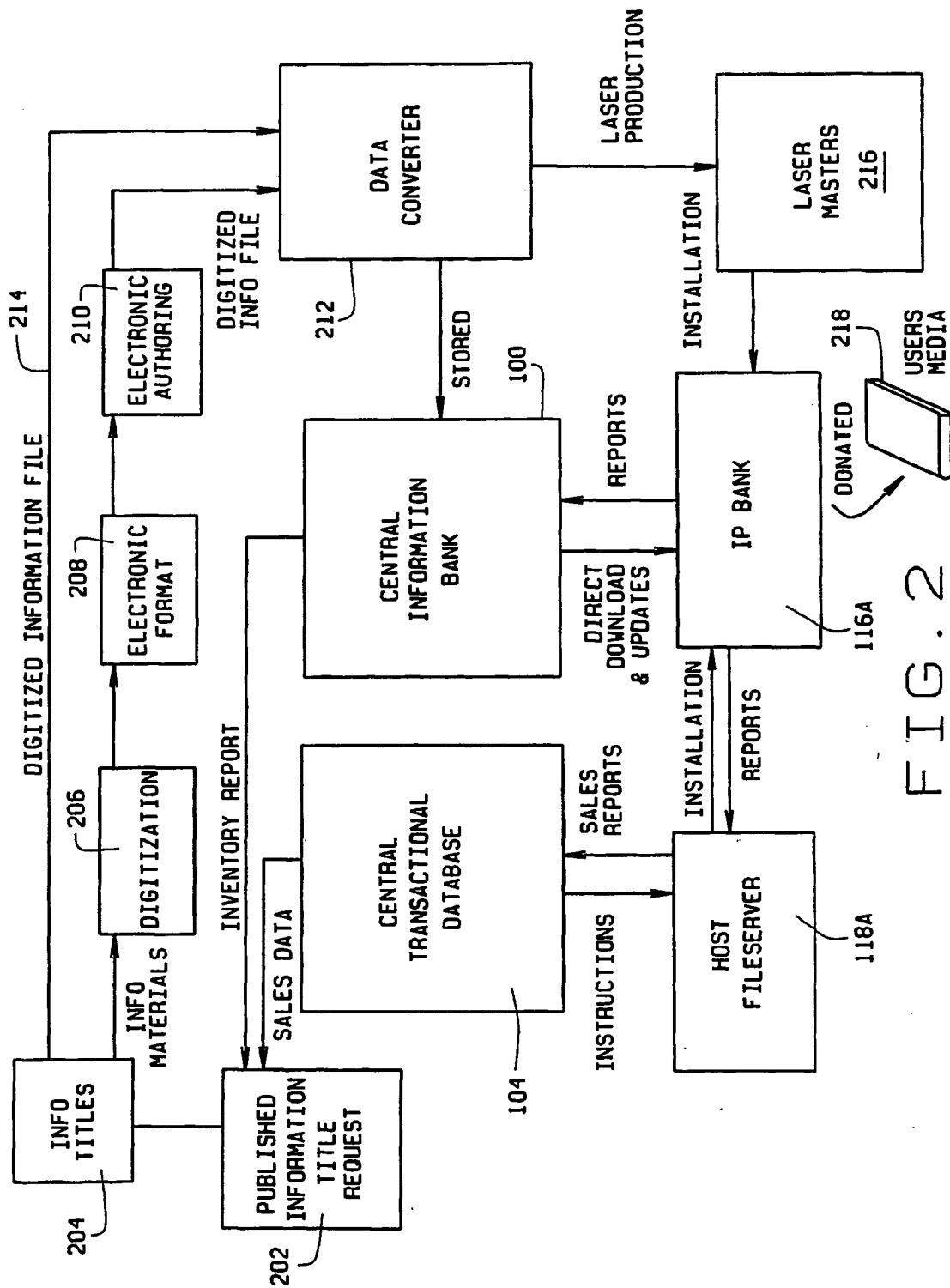
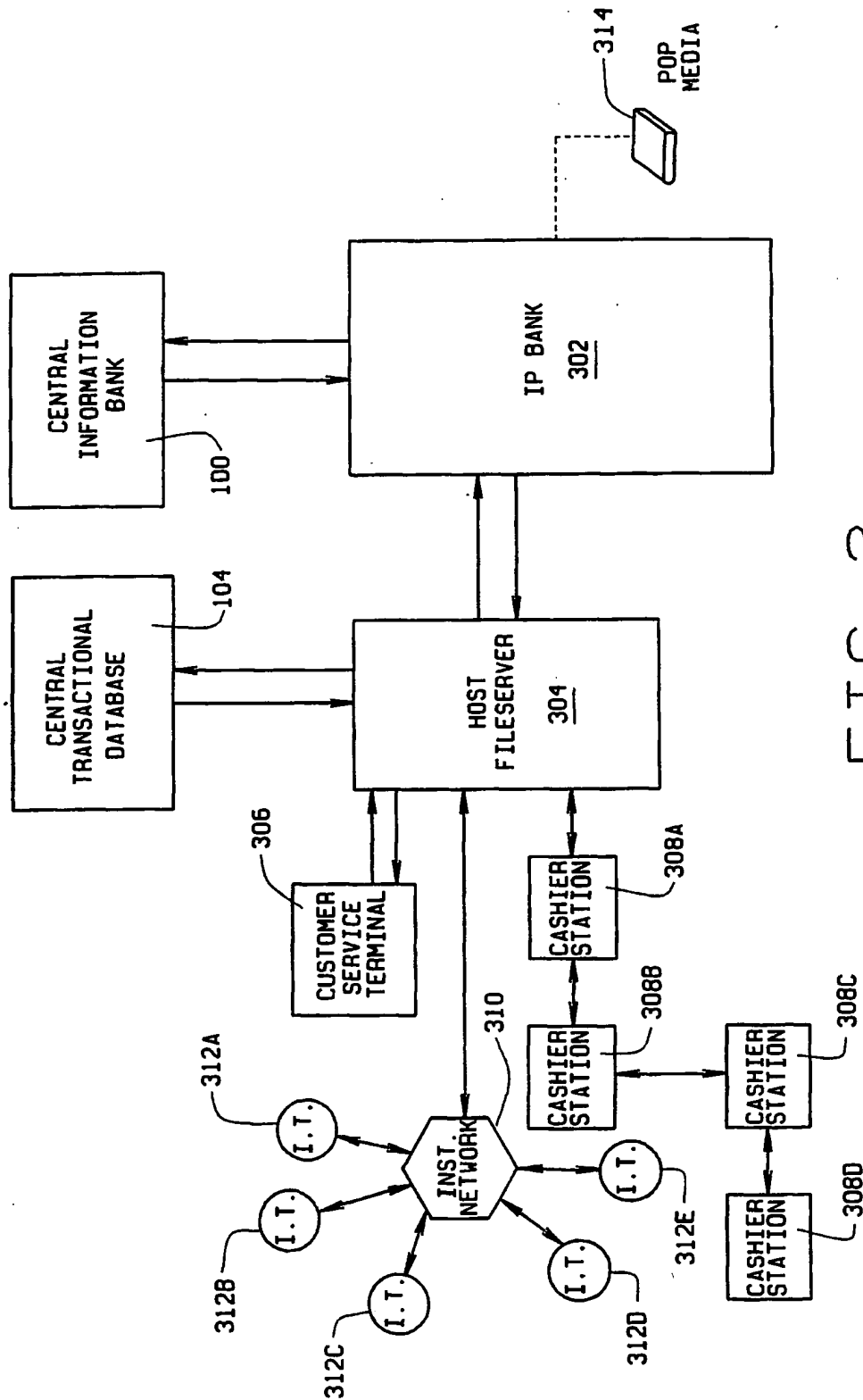


FIG. 1



2.
G
H
F



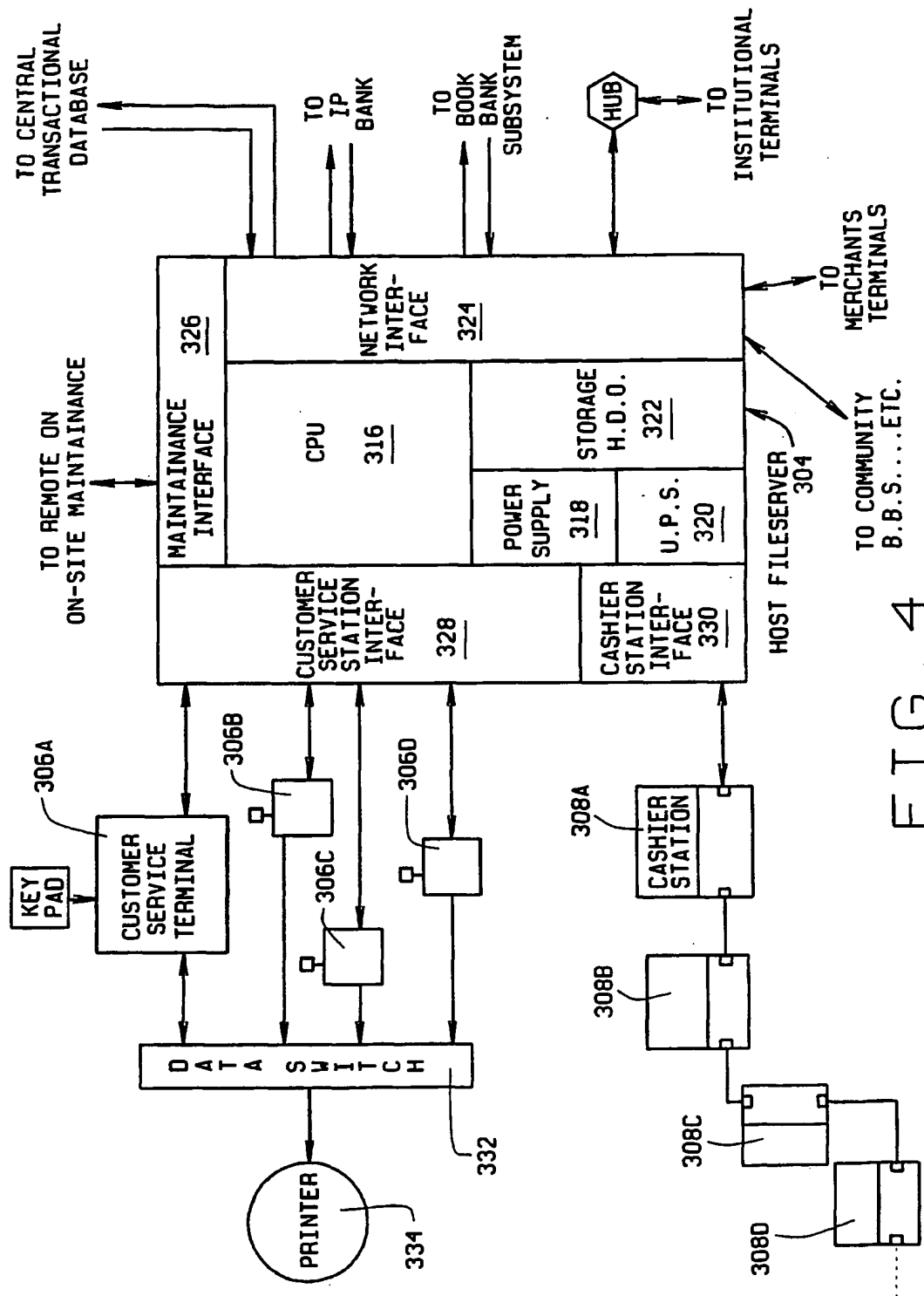
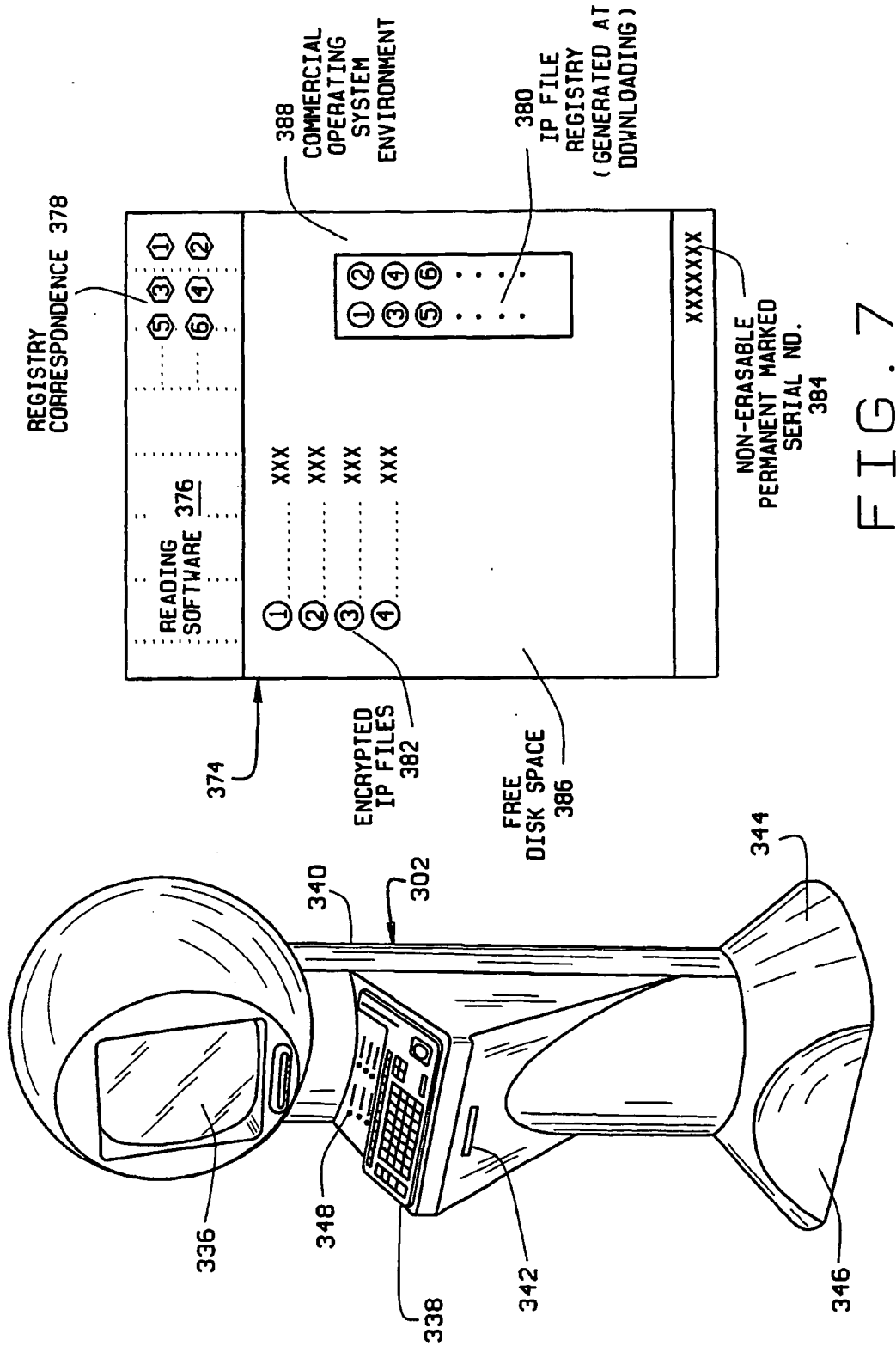


FIG. 4



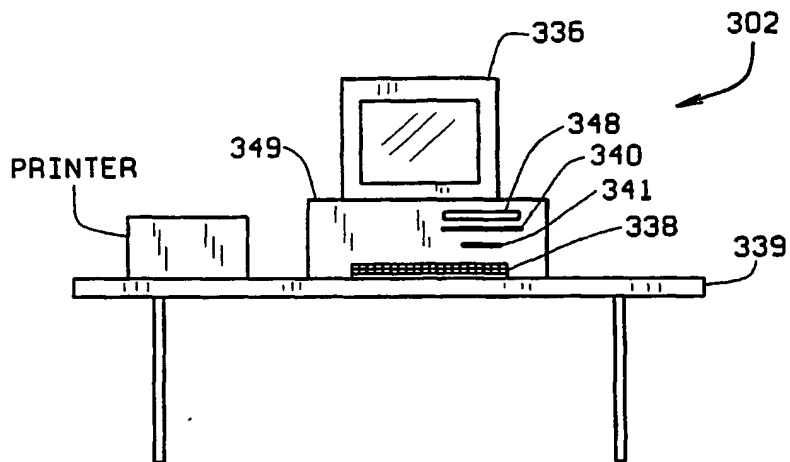


FIG. 5A

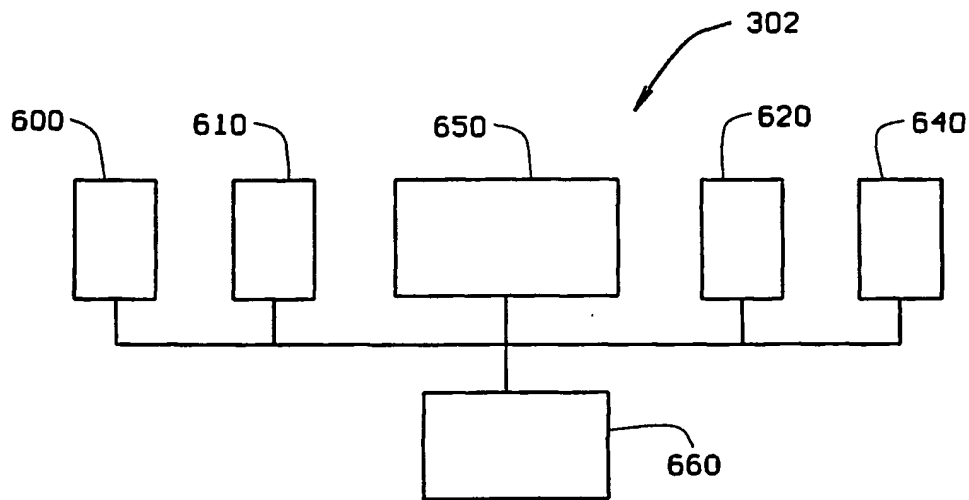


FIG. 13

7/16

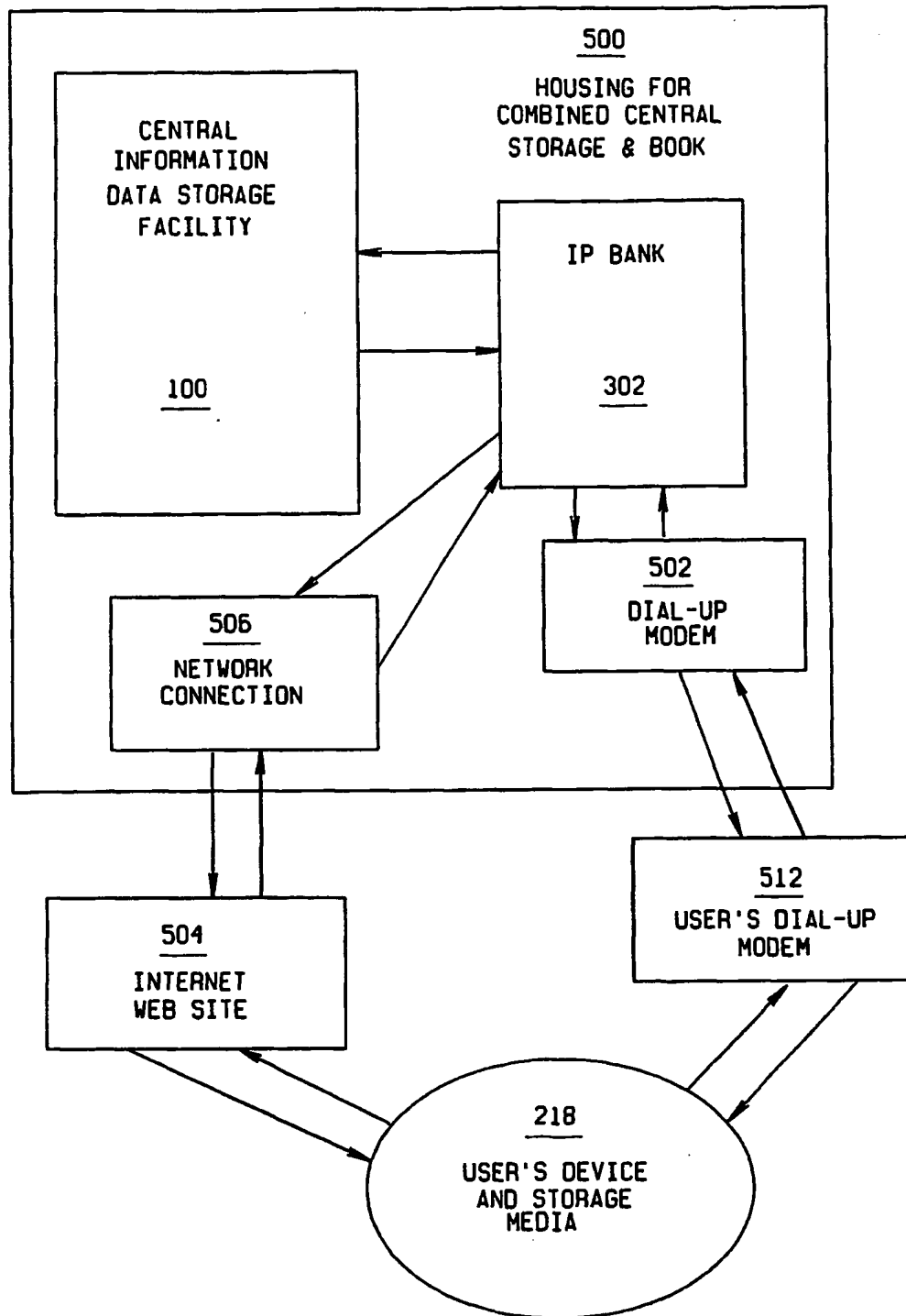


FIG. 5B

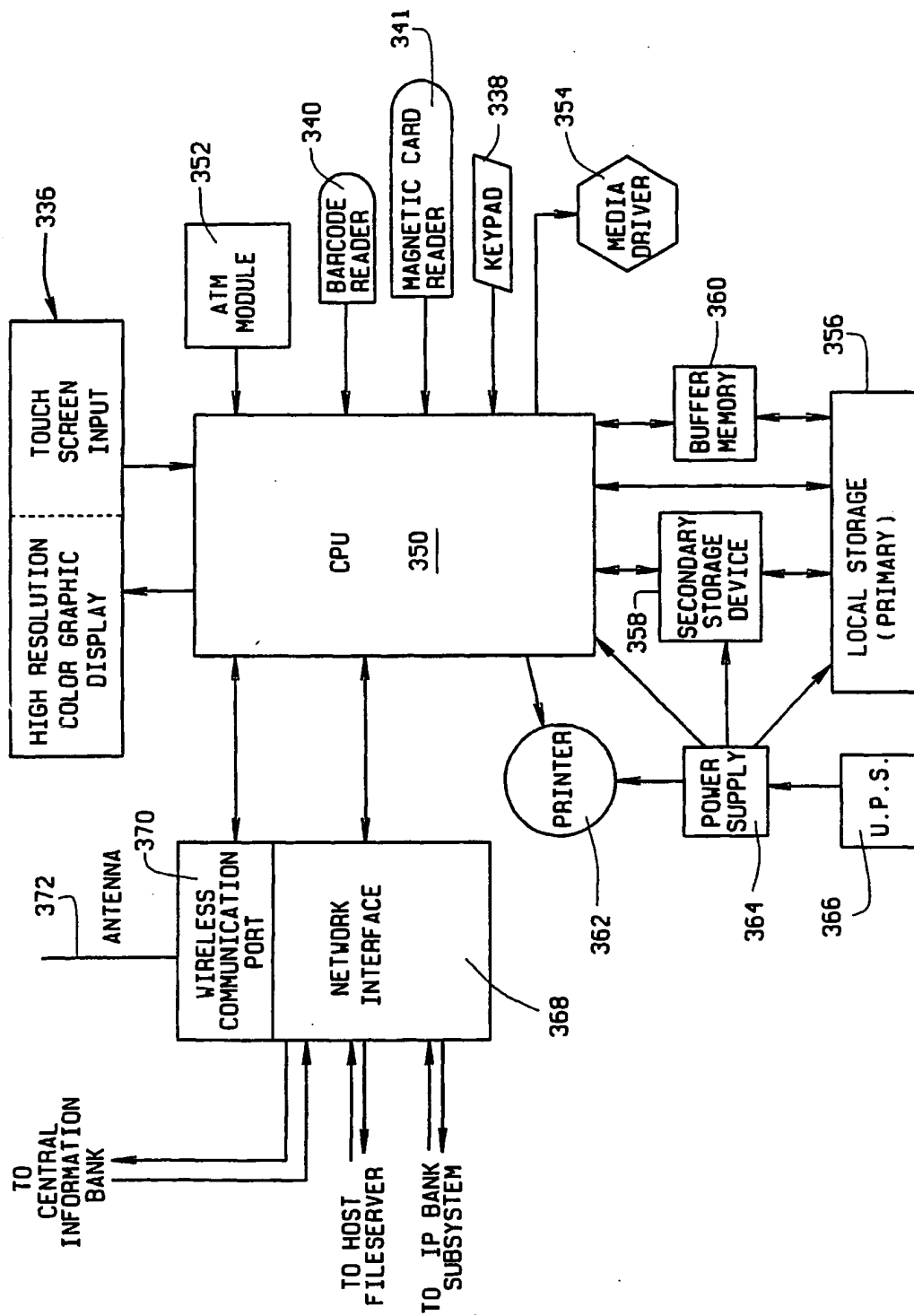


FIG. 6

9/16

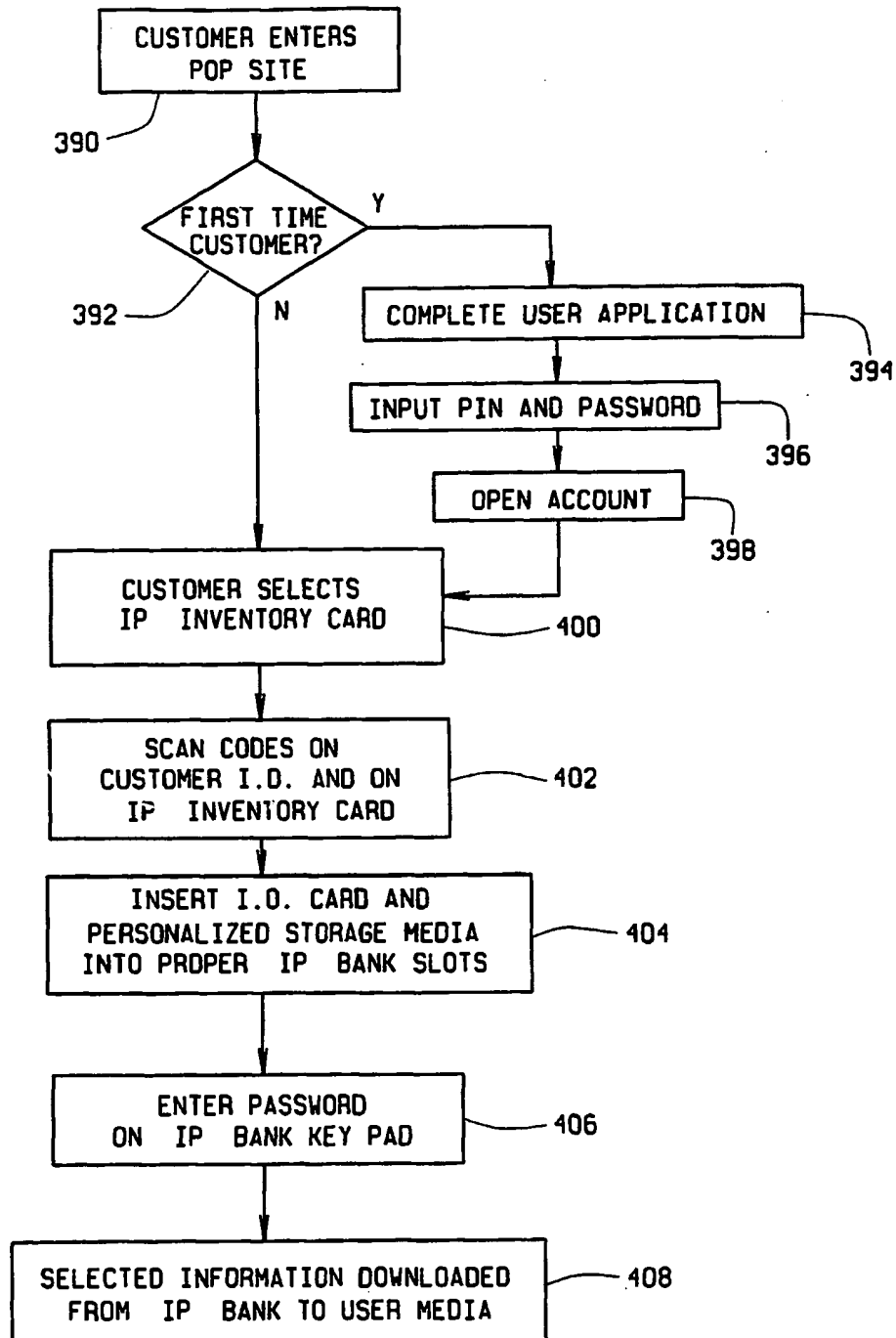


FIG. 8

10/16

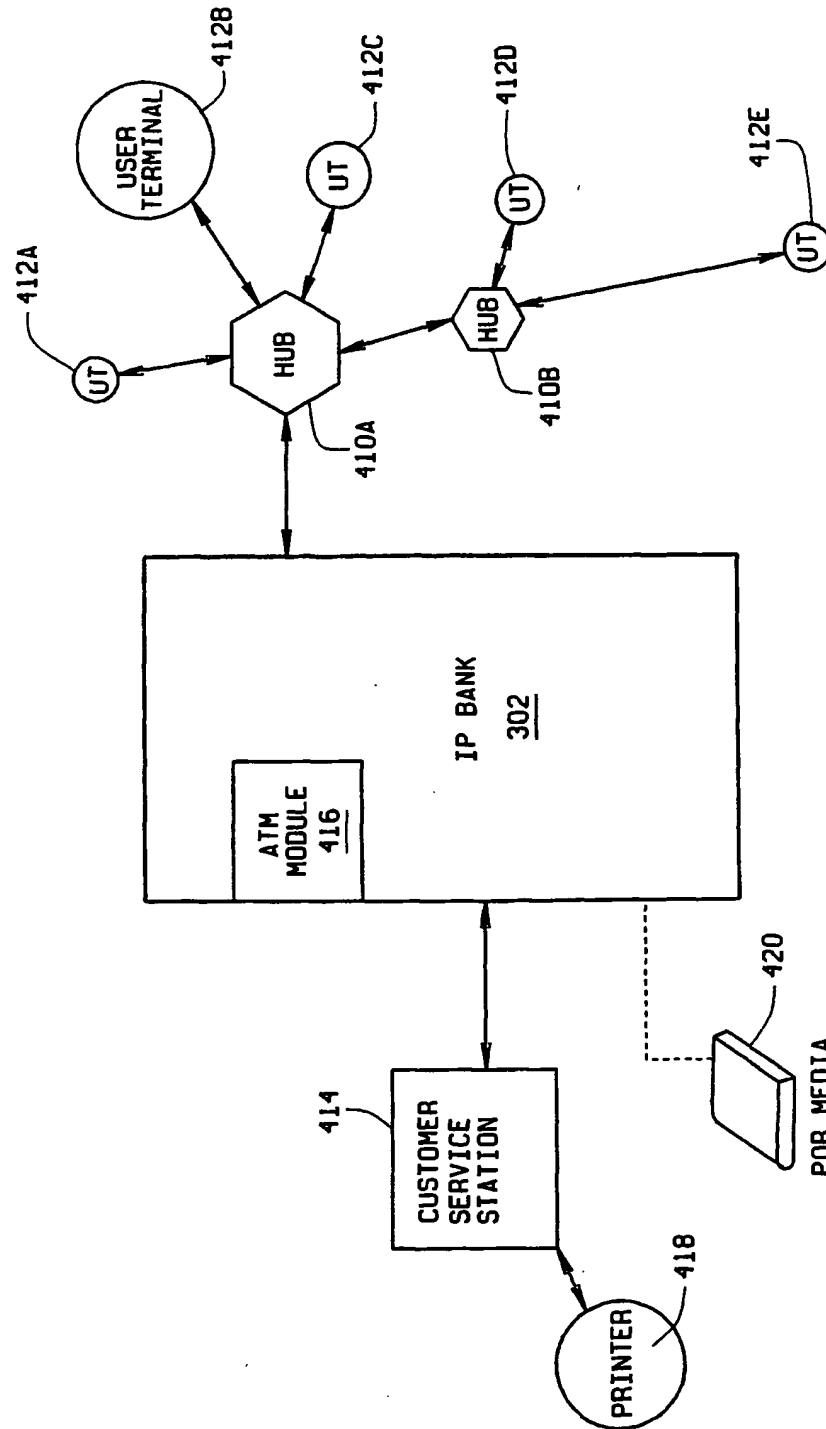


FIG. 9

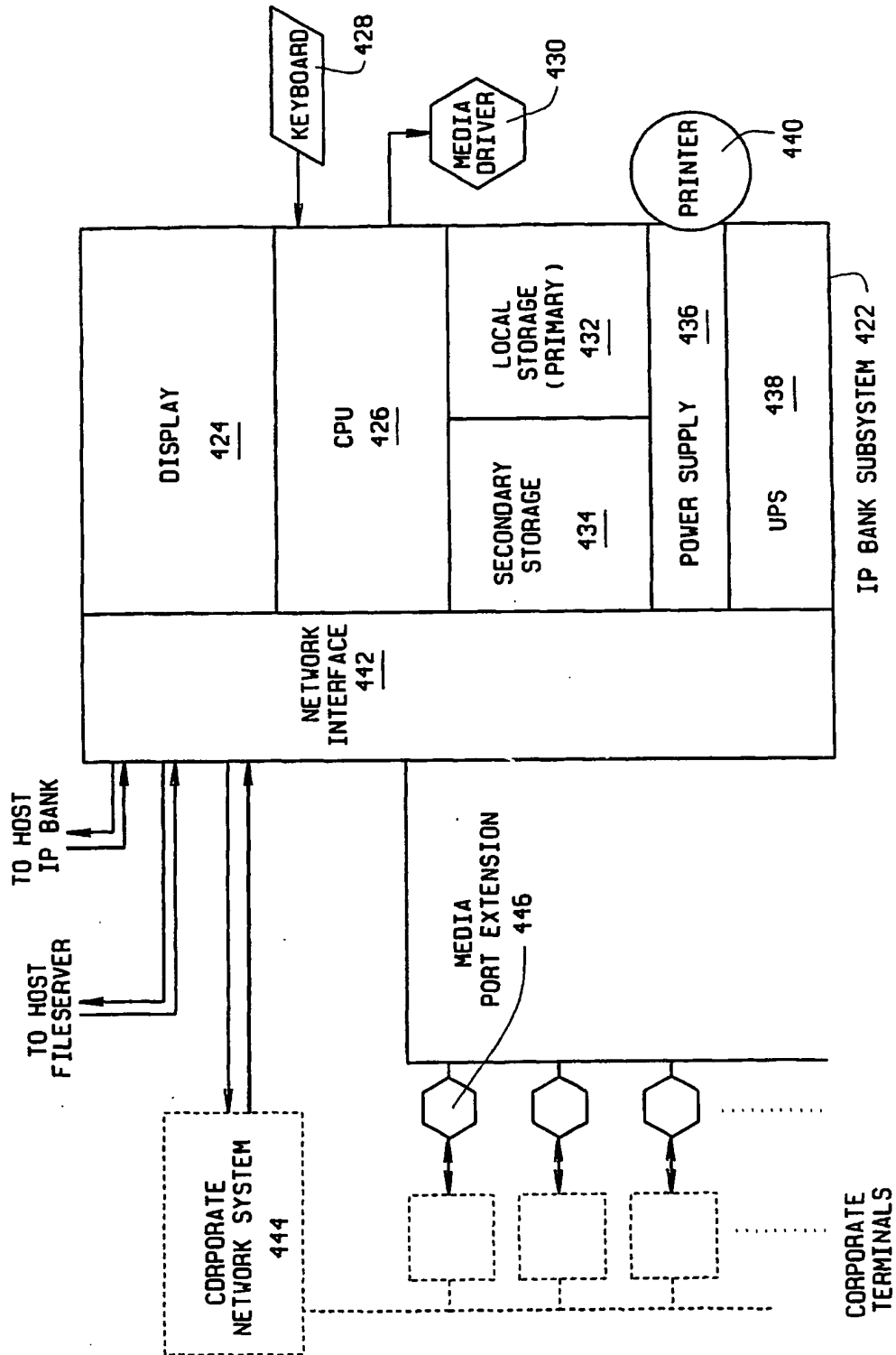
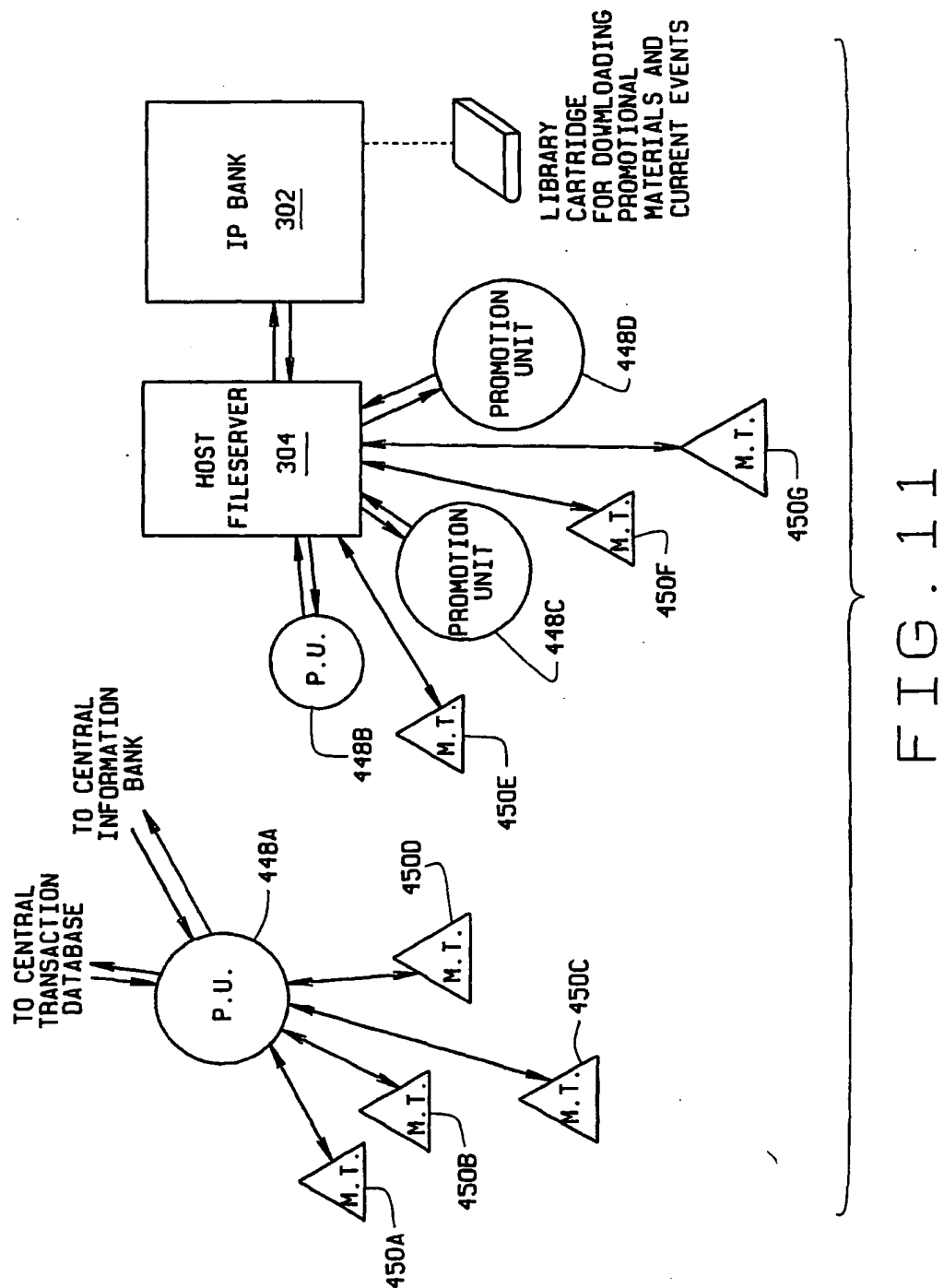


FIG. 10



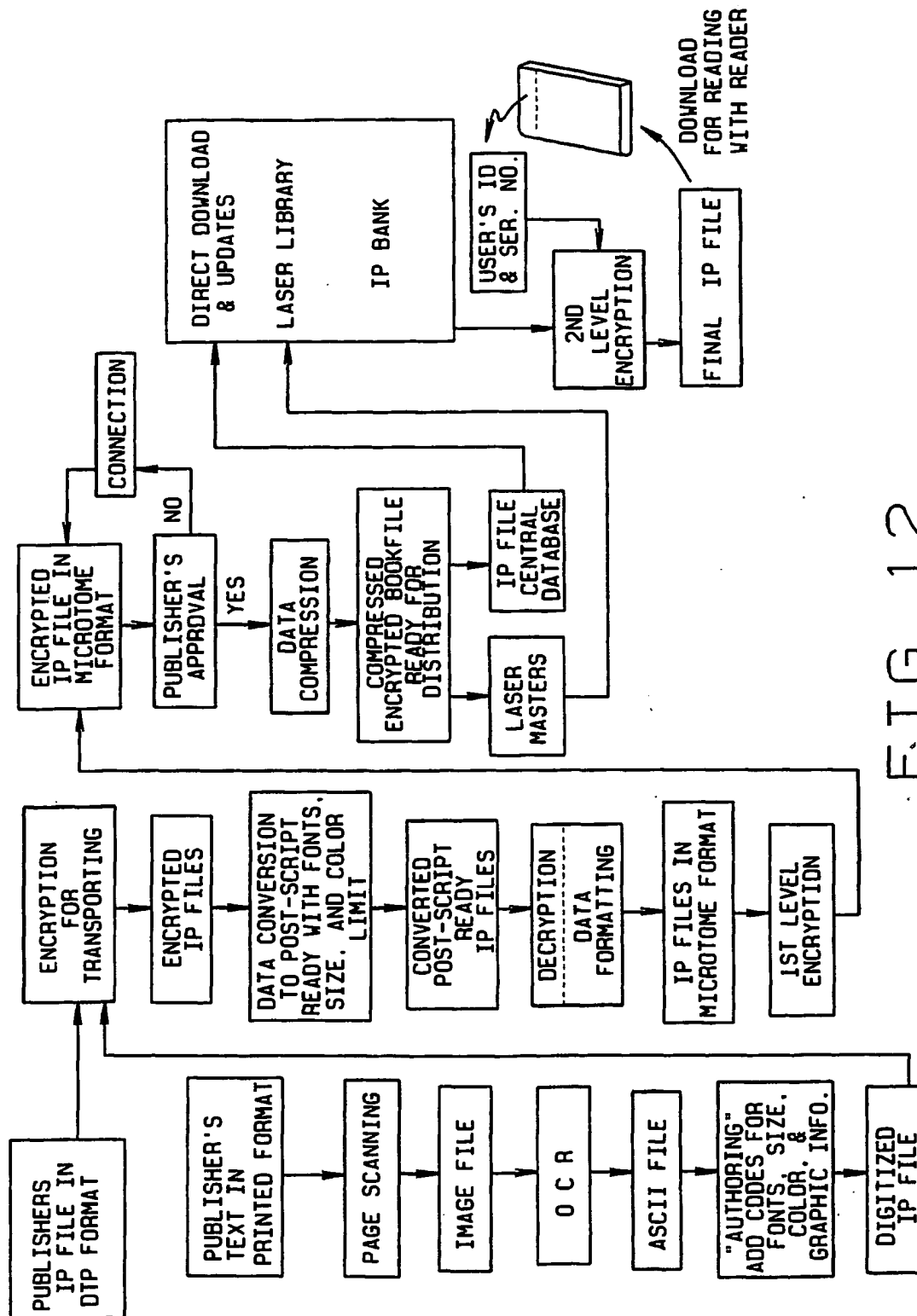


FIG. 12

14/16

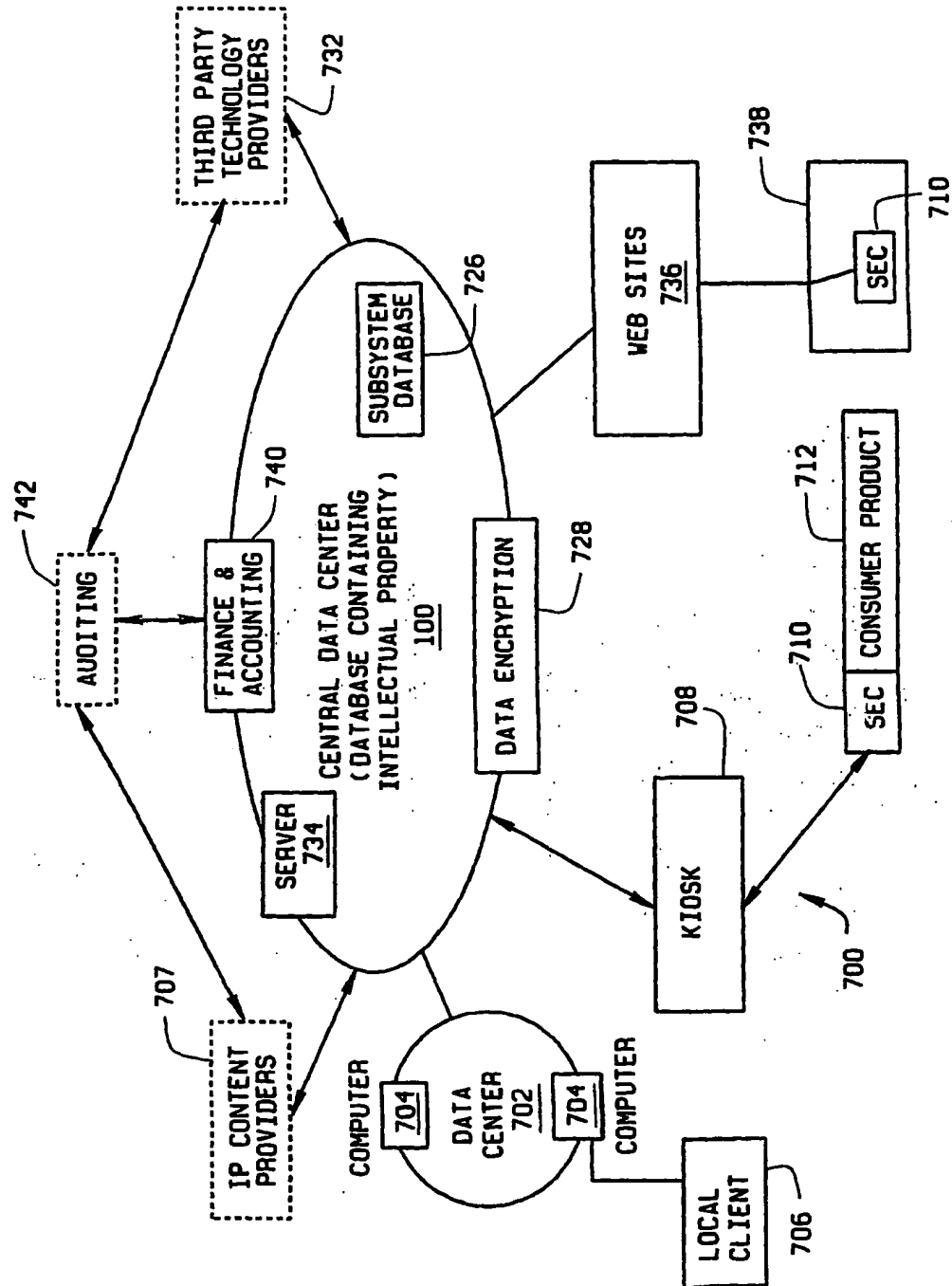


FIG. 14

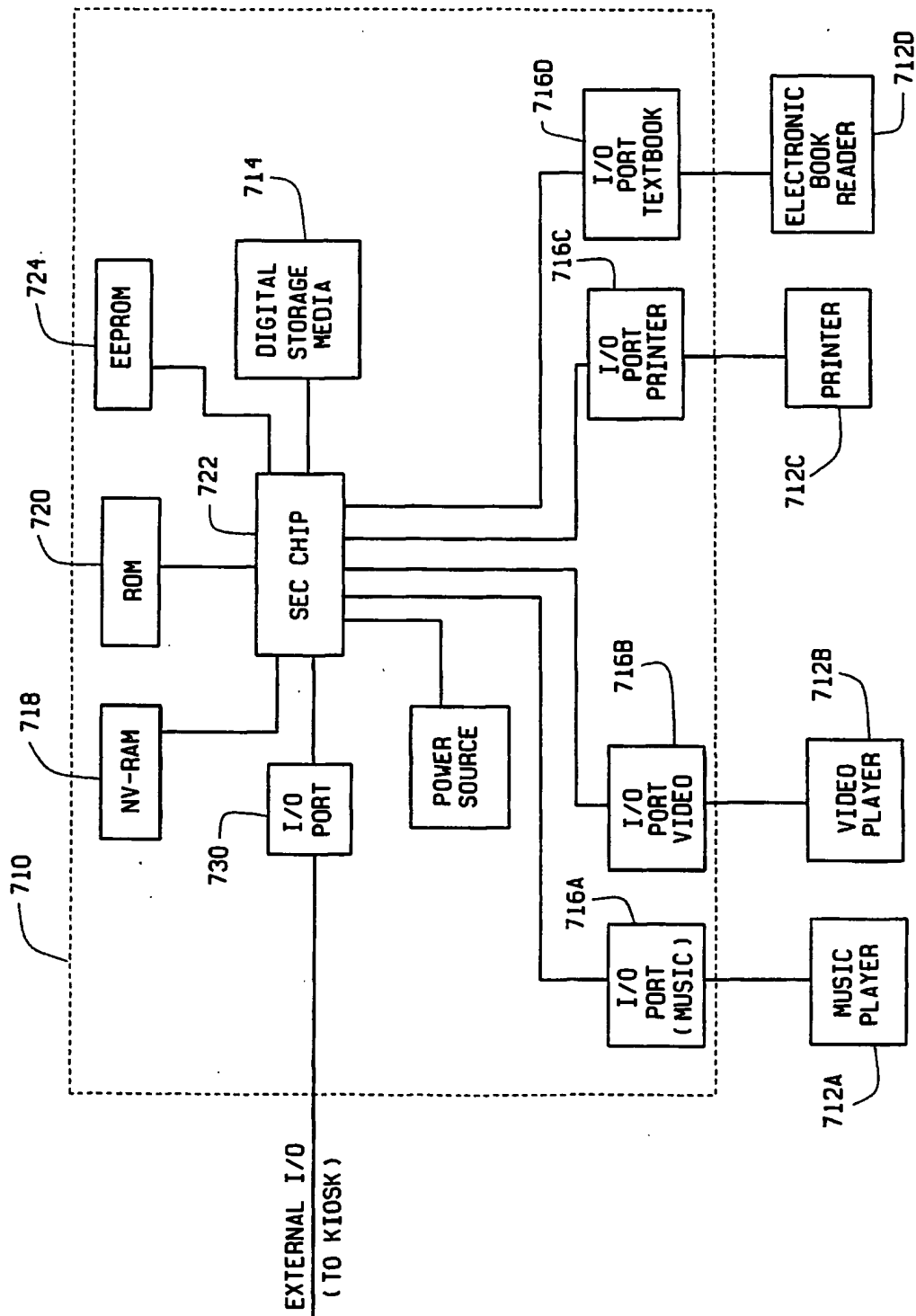
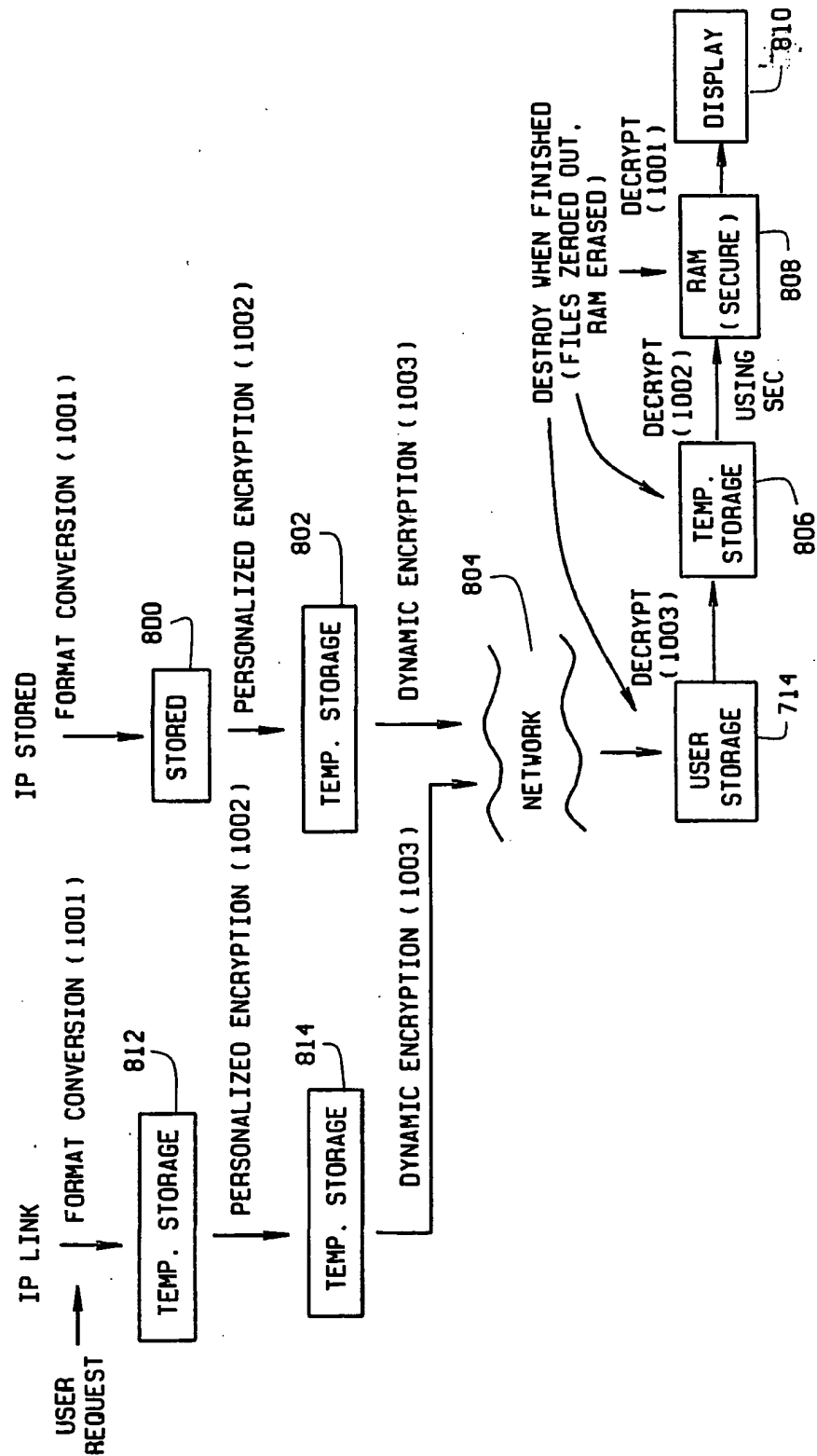


FIG. 15



W.T.G.H.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US01/05706

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G06F 17/60

US CL : 705/50

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 705/50, 51, 52, 53, 54, 57

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,910,987 A (GINTER et al) 08 June 1999 (08.06.1999), See entire document	1-92
A	US 5,715,403 A (STEFIK) 03 February 1998 (03.02.1998), See entire document	1-92
A	US 5,388,196 A (PAJAK et al) 07 February 1995 (07.02.1995), See entire document	1-92
A	US 4,855,725 A (FERNANDEZ) 08 August 1989 (08.08.1989), See entire document	1-92
A	US 4,899,292 A (MONTAGNA et al) 06 February 1990 (06.02.1990), See entire document	1-92

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

Special categories of cited documents:	
* "A" document defining the general state of the art which is not considered to be of particular relevance	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"E" earlier application or patent published on or after the international filing date	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O" document referring to an oral disclosure, use, exhibition or other means	"A" document member of the same patent family
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

24 May 2001 (24.05.2001)

Date of mailing of the international search report

18 JUN 2001

Name and mailing address of the ISA/US

Commissioner of Patents and Trademarks

Box PCT

Washington, D.C. 20231

Facsimile No. (703)305-3230

Authorized officer

James Trammell

Telephone No. (703)305-9700

Form PCT/ISA/210 (second sheet) (July 1998)

Requested Patent: EP0668695A2
Title: METHOD FOR RESTRICTING DATA REPRODUCTION. ;
Abstracted Patent: EP0668695 ;
Publication Date: 1995-08-23 ;
Inventor(s): SUGAHRA TAKAYUKI (JP) ;
Applicant(s): VICTOR COMPANY OF JAPAN (JP) ;
Application Number: EP19950300895 19950214 ;
Priority Number(s): JP19940047762 19940222 ;
IPC Classification: H04N5/913; G11B20/00 ;
Equivalents:
CN1066892B, CN1116803, DE69516326D, DE69516326T, JP2853727B2,
JP7235131, KR201232 ;

ABSTRACT:

In a reproduction apparatus, for reproducing an original signal conveyed as main data by a data medium such as a recording disk or broadcasting system, with medium protection data which are specific to the data medium being conveyed together with the main data, the apparatus includes a section (12) for generating apparatus protection data which are specific to the reproduction apparatus, a section (13) for combining the apparatus protection data with the medium protection data to define a protection level, and a section (14) for applying the protection level to restrict reproduction of the original signal, with stepwise variations in restriction occurring in accordance with changes in protection level. The medium protection data may include information for specifying restricted reproduction of portions of the original signal, such as by producing degraded resolution within specified regions of specified frames of a video signal.



11 Publication number : **0 668 695 A2**

12

EUROPEAN PATENT APPLICATION

21 Application number : **95300895.0**

51 Int. Cl.⁶ : **H04N 5/913, G11B 20/00**

22 Date of filing : **14.02.95**

30 Priority : **22.02.94 JP 47762/94**

43 Date of publication of application :
23.08.95 Bulletin 95/34

84 Designated Contracting States :
DE FR GB

71 Applicant : **VICTOR COMPANY OF JAPAN,
LIMITED**
3-12, Moriya-cho
Kanagawa-ku Yokohama (JP)

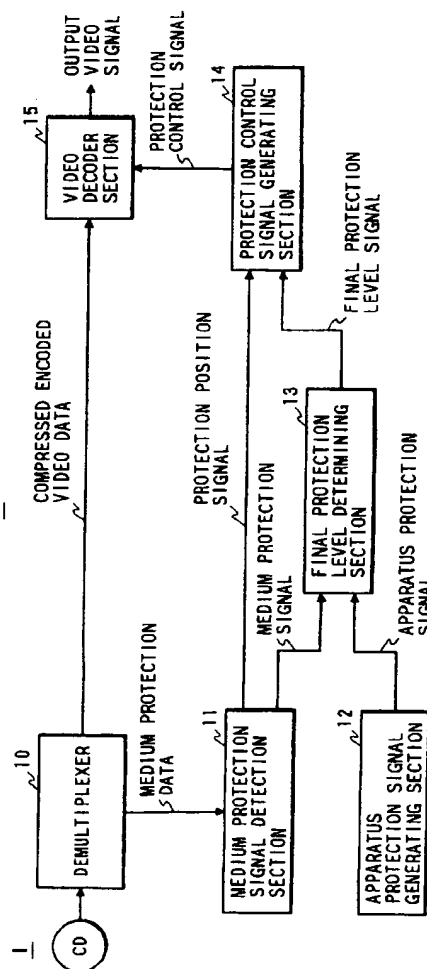
72 Inventor : **Sugahra, Takayuki**
14-7-101, Nakahama,
Isogo-ku
Yokohama (JP)

74 Representative : **Dempster, Benjamin John**
Naftel et al
Withers & Rogers
4 Dyer's Buildings,
Holborn
London EC1N 2JT (GB)

54 Method for restricting data reproduction.

57 In a reproduction apparatus, for reproducing an original signal conveyed as main data by a data medium such as a recording disk or broadcasting system, with medium protection data which are specific to the data medium being conveyed together with the main data, the apparatus includes a section (12) for generating apparatus protection data which are specific to the reproduction apparatus, a section (13) for combining the apparatus protection data with the medium protection data to define a protection level, and a section (14) for applying the protection level to restrict reproduction of of the original signal, with stepwise variations in restriction occurring in accordance with changes in protection level. The medium protection data may include information for specifying restricted reproduction of portions of the original signal, such as by producing degraded resolution within specified regions of specified frames of a video signal.

FIG. 1



EP 0 668 695 A2

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to a data reproduction protection method, and a data reproduction apparatus for implementing such a protection method, whereby reproduction of a signal represented by digital data such as a recorded digital video signal can be selectively restricted.

Description of the Prior Art

In the following, the term "data medium" is to be understood in a very general sense, as applying for example to broadcasting systems which transmit data such as video and/or audio data, in which case the received data may be the object of reproduction protection, and as applying also to any type of recording medium such as recording disks or tapes, etc., in which case playback data derived from the recording medium may be the object of reproduction protection. The reproduction protection may serve to selectively restrict viewing, hearing or copying of the data.

In the prior art, various types of reproduction protection method have been applied in fields such as CATV (cable television) and satellite television broadcasting. One method is to execute scrambling processing of transmitted video and audio data, and to insert a copyright code into the data, for thereby dividing the data into portions which can be freely reproduced and portions for which a fee must be paid in order to reproduce the data. When a program for which payment of a fee is necessary is received by a receiving apparatus, the program can be unscrambled and reproduced only if specified payment conditions are satisfied.

In the case of recorded media, one method of reproduction protection which is applicable to the DAT (digital audio tape recorder) recording system is the SCMS (serial copy management system). With that method, the playback DAT signal from a DAT playback apparatus has a main ID (identification) number which includes a copy inhibit code, whereby a single [copy enable - copy inhibit] sequence is ensured, so that a user can only make a single copy of a pre-recorded digital audio tape.

However with such prior art methods of reproduction protection there are only two control possibilities, i.e. reproduction is made either possible or impossible. It has not been possible hitherto to provide a gradually varying degree of restriction of reproduction of a signal conveyed by a data medium. Thus, such a reproduction protection method can only be used for a single purpose, e.g. for management of payment fees, or for copyright protection. Moreover with such a prior art reproduction protection method, since the data which are to be protected exist only in a transmitting

medium or recording medium prior to being reproduced, it has not been possible to provide a varying degree of limitation of reproduction capability in accordance with some condition of the reproduction apparatus. Thus in some cases, the degree of protection may be excessively severe, or excessively lax, so that it is difficult to achieve an effective degree of protection. For example, certain types of scenes recorded on a video tape may be permitted to be viewed in a certain country, such as the U.S.A., but may not be permissible in other countries. It would be thus advantageous to ensure that when that video tape is played on a reproduction apparatus which is sold to the public in such other countries, reproduction protection is automatically applied such that the aforementioned scenes will not be reproduced, or will not be clearly reproduced. However in the prior art, such a feature has not been possible.

SUMMARY OF THE INVENTION

It is an objective of the present invention to overcome the problems of the prior art set out above, by providing a reproduction protection method and apparatus whereby information specifying a degree of restriction of reproduction of an original signal is conveyed (e.g. by a recording medium or signal transmission medium) together with data expressing the original signal, whereby information specifying a degree of restriction of reproduction of the original signal are generated by a reproduction apparatus which operates on the conveyed data, and whereby information specifying a degree of restriction which is actually applied to reproduction of the original signal is derived based on a combination of the restriction information conveyed by the data medium and the restriction information generated by the reproduction apparatus.

More specifically, the invention provides a reproduction protection method comprising:

attaching medium protection data to main data which are conveyed by a data medium, said main data representing an original signal;

supplying the main data and medium protection data, via the data medium, to a reproduction apparatus;

generating apparatus protection data by the reproduction apparatus;

determining a protection level by combining the medium protection data and the apparatus protection data; and

controlling the reproduction apparatus to utilize the main data to reproduce the original signal in accordance with the protection level.

It is a further objective of the invention to overcome the above problems by providing a reproduction apparatus providing reproduction protection, for operating on main data representing an original signal and medium protection data expressing a medium protec-

tion level, said main data and medium protection being conveyed by a data medium, the apparatus comprising:

means for detecting said medium protection data to obtain a medium protection signal expressing said medium protection level;

means for generating an apparatus protection signal expressing an apparatus protection level which has been assigned to said reproduction apparatus;

means responsive to said medium protection signal and apparatus protection signal for determining a final protection level in accordance with a combination of said medium protection level and apparatus protection level;

means for executing reproduction of said original signal by utilizing said main data, including means for selectively restricting said reproduction in accordance with said final protection level.

With such a method and apparatus for reproduction protection, the protection level can be determined in accordance with the medium protection data, and hence can be determined in accordance with the wishes of the manufacturer of the data medium, or of the copyright owner of the main data. In addition, the protection level which is actually applied (i.e. the final protection level) is also determined in accordance with the apparatus protection data, which can be specified by the manufacturer or the seller of the reproduction apparatus. As a result, when the main data are to be reproduced (for example, during playback of a recording disk or tape), a graduated degree of limitation of reproduction is implemented, with that degree of limitation being determined by the final protection level, i.e. being determined in accordance with a combination of the requirements of the data medium manufacturer or the copyright owner of the main data and the requirements of the manufacturer or seller of the reproduction apparatus. In that way, considerable flexibility can be ensured in selectively restricting reproduction of signals which are conveyed for example by recording disks or tapes or by broadcasting systems.

In particular, the invention enables such reproduction restriction to be applied to specific frames or sequences of frames of a video signal, or to specific regions within each of a sequence of frames.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a general system block diagram of a first embodiment of the invention, which is a CD player providing reproduction protection in accordance with the present invention, for use in describing the basic principles of the invention; Fig. 2 is a matrix diagram showing an example of how final protection levels are determined in a reproduction apparatus according to the present invention;

Fig. 3 shows specific examples of how medium protection levels can be assigned;

Fig. 4 shows specific examples of how apparatus protection levels can be assigned;

Fig. 5 is a general system block diagram of a second embodiment, which is a specific configuration for the apparatus of Fig. 1, wherein protection control is applied to a data decompression section;

Fig. 6 shows examples of relationships between final protection level values and video picture visibility grades, for three different methods of protection control of a video signal;

Fig. 7 is a general system block diagram of a third embodiment, which is a specific configuration for the apparatus of Fig. 1, wherein protection control is applied to a video reproduction control section;

Fig. 8 is a conceptual diagram for describing how time-axis protection and spatial-domain protection control can be applied to a video signal with the present invention;

Fig. 9 is a diagram for describing how video, audio and protection data can be conveyed in a data stream in accordance with the MPEG1 standards;

Fig. 10 is a block diagram of an example of the internal configuration of a video reproduction control section in the embodiment of Fig. 7;

Fig. 11 is a general system block diagram of a fourth embodiment, which is a specific configuration for the apparatus of Fig. 1, wherein protection control is applied to a video reproduction control section and also to an audio reproduction control section;

Fig. 12 shows examples of relationships between final protection level values and audio signal audibility grades, for three different methods of protection control;

Fig. 13 is a block diagram of an example of the internal configuration of an audio reproduction control section in the embodiment of Fig. 11;

Fig. 14 is a matrix diagram for illustrating how variable ranges can be defined for final protection level values, by a fifth embodiment of the invention;

Fig. 15 is a general system block diagram of the fifth embodiment, wherein a final protection level can be modified by operation of a switch provided on the reproduction apparatus; and

Fig. 16 shows relationships between medium protection level values and settings of a modification switch in the embodiment of Fig. 15, for each of respective values of the apparatus protection level.

DESCRIPTION OF PREFERRED EMBODIMENTS

Embodiments of the invention will now be described, in which the data medium is assumed to be a recording medium, specifically a CD (compact disk), having video and audio signal data recorded thereon. In the following, such data representing original signals which are to be reproduced from the data medium will be referred to in general as the main data, to distinguish these from protection data, which are also conveyed by the data medium as described hereinafter. It will further be assumed that the original video and audio signals have been encoded by high-efficiency compression encoding using the MPEG1 algorithm, prior to recording. The MPEG1 algorithm is described for example in "International Standards for Multimedia Encoding", edited by Yasuda, published by the Maruzen company in Japan. Fig. 1 is a diagram for describing the general features of the reproduction protection method and reproduction protection apparatus.

Protection Information Provided on Recording Medium

The concept of medium protection data will first be described, referring to Fig. 1, in which data recorded on a CD (compact disk) 1 are read out from the disk to obtain an input signal for a demultiplexer 10 of a CD player 2, which is shown in block diagram form and which is configured to provide reproduction protection control in accordance with the present invention. The CD 1 has video and audio signals recorded thereon as digital data (referred to in the following as the main data), using MPEG1 compression encoding. Protection data, which are predetermined in accordance with the contents of the main data and will be referred to as the medium protection data, are also recorded on the CD 1. The medium protection data consist of information to be used in selectively restricting reproduction of the main data, as described hereinafter. The medium protection data can for example be recorded within the main code and sub-code header regions of the CD 1, or in the user region of the compressed data. The medium protection data expresses a protection level, referred to as the medium protection level, which can for example take values which successively increase in five steps, from 1 to 5, as shown in Fig. 2 and described hereinafter. In that case, the medium protection data can express the medium protection level by 3 bits. If separate medium protection levels are provided for the recorded video and audio data, then these can be expressed by two sets of 3 bits. The higher the number of the medium protection level, the greater is the degree of protection (i.e. the greater becomes the degree of restriction of reproduction of the recorded video or audio signal). The limitation of reproduction can for example be ef-

fect, in the case of video data, by operating on the playback data obtained from the CD1 such as to produce an output video signal, from the apparatus of Fig. 1 which will result in a mosaic pattern being produced within all or a specific region of a resultant displayed picture, as described hereinafter.

The medium protection level which is expressed by the medium protection data is predetermined in accordance with the wishes of the manufacturer or seller of the data medium (CD 1), or of the owner of the copyright for the main data. Fig. 3 shows two examples of medium protection data, each expressing 5 values of medium protection level. In the first example, limitation of reproduction is based upon the film classification system (i.e. movie ratings system) which is used in the U.S.A. In the case of a film which is rated "free", no restriction on reproduction is imposed by the medium protection data. If the movie is rated "PG" (i.e. parental guidance), then a moderate degree of restriction (protection level 1) is applied, and so on with increasing degrees of restriction for the "R" and "X" ratings. In the second example, limitation of reproduction is based on the rights of the copyright owner, providing successively increasing degrees of restriction of reproduction as the protection level increases from 1 to 5.

The medium protection data is a combination of data for expressing at least one medium protection level as described above, and protection position information which specifies the position (within the encoded main data) at which the protection is to be applied. It is an essential feature of the present invention that the medium protection data can assign the medium protection level in units of frames of the video signal. That is to say, limitation of reproduction of individual frames can be controlled. In addition, limitation of reproduction of one or more specific regions within a specific frame (or sequence of specific frames) can also be predetermined by the medium protection data. In that case, for example, a region formed of a number of fixed-size blocks of pixels can be converted to a blank region, or filled with a mosaic pattern, in the final display picture that is obtained from the output video signal from the apparatus of Fig. 1. That is to say, the protection position information can be used to specify not only a specific frame, but also to specify one or more specific regions within a frame.

Alternatively, the protection position information can specify an identical medium protection level for the entirety of each of a succession of video signal frames, or specify an identical medium protection level for one or more specific regions within each of a succession of frames.

The term "protection position information" is used here, since it may not be necessary to record explicit data constituting the protection position information on the CD1. Instead, that information can be inherent-

ly constituted by the positions at which respective medium protection data are located within the stream of compressed encoded main data that have been recorded on the CD 1. For example, the apparatus may be configured such that if the encoded data for a video signal frame are immediately preceded by a portion of medium protection data, then that indicates that the medium protection data portion is to be applied to that frame. If the medium protection data portion is to be applied to one or more regions within that frame, rather than the entire frame, then the medium protection data portion which immediately precedes the compressed encoded frame in the recorded data can include at least two values for each of these regions, for specifying the respective positions of the regions. In that case, explicit protection position information must be recorded as data on the CD 1, as part of the medium protection data.

Protection Information Generated by Reproduction Apparatus

With the method and apparatus of the present invention, a reproduction apparatus can be configured to produce predetermined protection data which are specific to that reproduction apparatus. For example, the CD 1 in Fig. 1 is played by a reproduction apparatus 2 which is provided with a presettable memory device which will be assumed to be a ROM (read-only memory) which generates apparatus protection data expressing a protection level referred to in the following as an apparatus protection level. The apparatus protection level is specified beforehand by the manufacturer or the seller of the reproduction apparatus. In the same way as for the medium protection data described above, the apparatus protection level can take a plurality of values, corresponding to respectively different degrees of limitation of reproduction of main data which are obtained from a data medium. It will be assumed that the number of apparatus protection level values is 4, i.e. from 1 to 4, so that the apparatus protection data can be consist of two bits. The higher the apparatus protection level number, the greater becomes the degree of protection, i.e. the greater becomes the degree of reproduction limitation.

The contents of the apparatus protection data ROM cannot be rewritten by the user. In the embodiment of Fig. 1, each time that power to the apparatus is switched on, the apparatus protection data are read out from the ROM, and thereafter reproduction is executed in accordance with a combination of the apparatus protection level specified by the apparatus protection data and the medium protection level which is specified by the medium protection data.

Fig. 4 shows three examples of how the apparatus protection level can be assigned. In the first example, the apparatus protection level is preset in ac-

cordance with the country in which the reproduction apparatus is to be used. If the reproduction apparatus is to be used in the U.S.A. for example, then it is possible that the apparatus protection level could be set to a low value such as 1. In the case of a reproduction apparatus which is to be used in other parts of the world, such as Europe, Japan and Taiwan, which have varying degrees of restrictions on video software, the apparatus protection level could be set to higher values, as illustrated. In that way, video scenes which are not permissible in one country can be automatically eliminated (partially or completely), e.g. by insertion of mosaic pattern regions in the resultant display picture, by using the apparatus protection level and medium protection level in combination as described hereinafter.

With the second example in Fig. 4, the apparatus protection level is predetermined in accordance with the type of person who is expected to use the reproduction apparatus. If the reproduction apparatus is to be used only by adults, for example, then the apparatus protection level can be set at a low value such as 1. If the reproduction apparatus will be used by children, the apparatus protection level can be set to a high value such as 4. In that way, video scenes or audio content which are considered unsuitable for children can be partially or completely restricted from being reproduced.

With the third example in Fig. 4, the protection level that is set by the apparatus protection data is predetermined in accordance with the applications for which the reproduction apparatus will be used. For example if the reproduction apparatus is to be sold to the public, then the apparatus protection level can be set to a value such as 1 or 2, whereas if the reproduction apparatus is intended to be used for demonstration purposes in a shop, then the apparatus protection level can be set to a different value, i.e. 3 or 4, and the apparatus of Fig. 1 controlled such that only certain scenes which should be of interest to possible customers of the shop will be displayed. In that way, the apparatus protection level can be set in accordance with the application objectives of the reproduction apparatus.

The overall features of reproduction protection will now be described, referring to Fig. 1. Firstly, data recorded on the CD 1 are read out, as an input signal to the demultiplexer 10. The recorded data on the CD 1 consist of the compressed encoded main data (i.e. compressed encoded video and audio data) which are multiplexed with the medium protection data. The demultiplexer 10 separates the compressed encoded main data from the medium protection data, and supplies the medium protection data to a medium protection signal detection section 11 while supplying the compressed encoded main data to a video decoder section 15. As described hereinabove, the medium protection data may inherently specify protection

position information, or may include explicit protection position information. The medium protection signal detection section 11 serves to detect the protection position information, and generates a corresponding protection position signal, which indicates those portions of the main data to which the medium protection data applies (e.g. specific video signal frames, and/or block regions within specific frames). The protection position signal is supplied to a protection control signal generating section 14. The medium protection signal detection section 11 is further responsive to the medium protection data for generating a corresponding medium protection signal, which expresses the medium protection level and is supplied to a final protection level determining section 13.

The final protection level determining section 13 basically consists of a matrix ROM in this embodiment, i.e. a ROM which stores a pattern of relationships between respective combinations of medium protection levels and apparatus protection levels and resultant final protection levels. The operation of the matrix ROM will be described referring to the matrix diagram of Fig. 2, which shows an example of how the contents of that ROM are read out in response to combinations of medium protection level and apparatus protection level values. There are five possible values (designated as A to E respectively) for the final protection level, successively increasing in degree of reproduction limitation in the sequence A, B, C, D, E. Fig. 2 shows an example of various values of the final protection level which are determined by respective combinations of values of the medium protection level and apparatus protection level, i.e. the values in the range A to E which are located at respective intersections between rows and columns of the matrix in Fig. 2. Thus for example if the medium protection level is 4 and the apparatus protection level is 2, then the final protection level will be B.

It is necessary to clearly distinguish between the ROM of the apparatus protection signal generating section 12 and the matrix ROM of the final protection level determining section 13. The contents of the ROM of the apparatus protection signal generating section 12 can be set in accordance with the requirements for a particular reproduction apparatus, whereas the contents of the matrix ROM of the final protection level determining section 13 will in general be common to a large number of reproduction apparatus units.

The final protection level determining section 13 generates an output signal, referred to as the final protection level signal, which expresses the final protection level that has been determined, and supplies that signal to the protection control signal generating section 14. In response to that signal, and the protection position signal, the protection control signal generating section 14 generates a signal referred to as the protection control signal, which is supplied to con-

trol a section which will be referred to as the video decoder section 15. For simplicity of description, only video signal reproduction will be considered at this stage, and the video decoder section 15 should be understood as a section which converts the compressed encoded video data (main data) from the demultiplexer 10 to a standard (analog) video signal. As the stream of compressed encoded main data flows into the video decoder section 15 from the demultiplexer 10, the protection control signal controls the video decoder section 15 such as to apply reproduction protection in accordance with the final protection level values, at the respective positions within that data flow which are specified by the protection position information. The video decoder section 15 thereby produces an output video signal which will result in video pictures in which reproduction of the main (video) data is limited in accordance with the final protection level values. As will be understood from the above, the final protection level values may change from frame to frame of the video signal, in accordance with changes in the medium protection level.

A first example of limitation of reproduction in accordance with the final protection level will be described, which is implemented by controlling the expansion and decoding of the main data by the video decoder section 15. The example will be described referring to the embodiment of Fig. 5, in which the reproduction apparatus is again a CD playback apparatus, designated by numeral 3. Only the operation with regard to the video data of the main data from the CD 1 will be described. In Fig. 5, a specific configuration for the video decoder section 15 of Fig. 1 is shown, made up of a variable-length decoding section 15a, a dequantizer section 15b, an inverse transform section 15c and a video reproduction control section 15d, with the compressed encoded video data being supplied from the demultiplexer 10 to the variable-length decoding section 15a and with the final output (analog) video signal being produced from the video reproduction control section 15d. In this embodiment, the video data have been recorded on the CD 1 after being subjected to compression by discrete cosine transform processing, and reproduction limitation is controlled by controlling the accuracy of inverse DCT (discrete cosine transform) processing which is effected by the inverse transform section 15c. The protection control signal which is produced from a protection control signal generating section 14a and supplied to control the inverse transform section 15c is derived based on the medium protection data and apparatus protection data, in combination, as described above for section 14 in Fig. 1, i.e. the protection control signal applies control in accordance with the final protection level.

The demultiplexed compressed encoded video data read from the CD 1 are subjected to variable-length decoding in the variable-length decoding sec-

tion 15a, and then to dequantization in the dequantizer section 15b. The resultant compressed data are then subjected to inverse DCT processing in the inverse transform section 15c, with the transform processing being selectively modified in accordance with the protection control signal. The resultant decompressed data are then processed in the video reproduction control section 15d to obtain the final output video signal. The effect of reproduction limitation controlled by the protection control signal acting on the inverse transform section 15c is to selectively produce a degree of blurring or formation of a mosaic pattern within the pictures which are displayed using the final output video signal. Such a degree of blurring will be referred to as the visibility grade. An example of the relationship between the visibility grade and the final protection level values A to E (which are determined by the protection control signal generating section 14a as described hereinabove, and are expressed by the protection control signal) is shown in the leftmost column of Fig. 6. If the protection level is A, i.e. minimum limitation of reproduction of the video data, then the protection control signal is set to a state whereby it does not affect the operation of the inverse transform section 15c. If the protection level is B, then the protection control signal controls the inverse transform section 15c such as to operate on a block size of 8 x 8 picture elements (i.e. the same block size which was utilized in the original DCT processing), using the DC component value for each block, but using only two of the AC transform coefficients, with the values of all of the other transform coefficients being forcibly set to zero. This will result in a substantial lowering of resolution of a display picture that is produced based on the output video signal. If the protection level is C, then the protection control signal controls the inverse transform section 15c such as to operate on a block size of 8 x 8 picture elements of a video signal frame, using only the DC component for each block, (i.e. the DC component is the only transform coefficient used). In this case, since all of the picture elements within an 8 x 8 picture element block will have identical video signal values, this will result in a mosaic pattern being formed in the finally obtained picture. If the protection level is D, then the protection control signal controls the inverse transform section 15c such as to operate on a block size of 16 x 16 picture elements, using only the DC component values. This will again result in a mosaic pattern being formed, in which the blocks of the pattern are of larger size than for the case of protection level C, i.e. a mosaic pattern of macroblocks is formed, thereby further degrading the degree of visibility of the resultant picture. If the protection level is E, then the video data obtained from the inverse transform section 15c are replaced by different video data (produced from a source not shown in the drawing), which are produced from the video reproduction control section 15d as

the final video signal, and which will produce a predetermined picture or pictures. Such a predetermined picture might for example display a warning message concerning copyright protection.

With the MPEG1 algorithm, a block size of 8 x 8 picture elements is used in the DCT processing. The transformed block is expressed by a DC component (i.e. DC coefficient) and a plurality of coefficients (the AC coefficients) which represent signal level values at respectively different successively increasing frequencies. Thus if for example the inverse transform section 15c is controlled such that only the DC coefficient and the two lowest-frequency AC coefficients are used, in the inverse DCT operation for each block, then a specific reduction in resolution of all (or a specific part) of the resultant display picture can be achieved in a very simple manner.

Similarly if only the DC component for a block is used in the inverse DCT processing, with all of the AC coefficients set to zero, then all of the signal level values for the picture elements of a block will be set to an identical value, in the resultant video signal obtained from the inverse DCT operation. Hence, a mosaic pattern can very easily be formed in the resultant picture that is obtained using the final output video signal. Moreover if, for each of respective 16 x 16 element macroblocks (i.e. each consisting of four 8 x 8 picture element blocks), only the DC coefficient for a specific one of the 8 x 8 element blocks is used (for example, the DC coefficient for the upper leftmost one of the 8 x 8 element blocks) in the inverse DCT processing for all of the four 8 x 8 element blocks constituting the 16 x 16 element macroblock, with all of the AC coefficients set to zero, then a mosaic pattern will be formed which is substantially coarser than the mosaic pattern which is formed by using the 8 x 8 element blocks.

It can thus be understood that with the above embodiment of the invention, applied to video data which have been subjected to high-efficiency compression encoding using a data transform operation, stepwise changes in a degree of restriction of reproduction of the video data can easily be accomplished by effecting stepwise changes in a degree of resolution of a finally obtained picture, or in a portion of that picture, and that such stepwise changes in resolution can be easily controlled in accordance with the final protection level which has been established based on the medium protection data and apparatus protection data. In particular, when such control is applied to the inverse DCT processing, it is possible to easily effect stepwise changes in the visibility grade, i.e. in the picture resolution, through use of unit blocks of picture element values which are basic to the transform processing. Such a type of control of the visibility grade, operating within each frame of the video signal, can be considered as applying protection in a (2-dimensional) spatial domain.

With a video data encoding method such as the DCT method, the output digital signal that is produced from the inverse DCT circuit consists of sequential sets of data, each consisting of successive picture element values for the respective picture elements of a unit block (e.g. a block of 8 x 8 picture elements) of a video signal frame. In order to convert such a digital signal into a normal digital video signal, it is necessary to first temporarily store the data produced from the inverse DCT circuit in a video memory (e.g. a frame memory), then to read out the video data in the correct sequence (i.e. as successive picture element values in successive picture scanning line intervals). That operation is the basic function of the video reproduction control section 15d in Fig. 5. Fig. 7 shows another embodiment of the invention, in which control for reproduction protection is applied to the video data which have been produced from the inverse transform section 15c, i.e. in which control by the protection control signal in accordance with the final protection level is applied to a video reproduction control section which is configured such as to respond appropriately to the protection control signal, and is designated as 15d'. Apart from this feature, the configuration and operation of this embodiment is identical to that of Fig. 5 described above. The video reproduction control section 15d' includes a frame memory, into which output data from the inverse transform section 15c are temporarily written, and then read out in the appropriate sequence as described above, to obtain the final output video signal.

In this embodiment, the degree of reproduction limitation is controlled by "thinning out" frames of the video data that are used to form the final output video signal, with the degree of "thinning out" being determined by the final protection level. That is illustrated by the central column in Fig. 6, in which such a type of control is referred to as time domain protection. In the example of Fig. 6, when the final protection level is A, then all of the frames of video data which are successively written into the frame memory of the video reproduction control section 15d' are used to form the final output video signal. If the final protection level is B, then the protection control signal from the protection control signal generating section 14b controls the video reproduction control section 15d' such that only one out of every 15 frames of video data supplied from the inverse transform section 15c is used to form the final output video signal. Specifically, one out of every fifteen frames of video data from the inverse transform section 15c is held stored in the frame memory of the video reproduction control section 15d' for fifteen successive frame periods, and is repetitively read out during that time, to form the final output video signal. Thus a type sample-and-hold operation is performed using the frame memory in the video reproduction control section 15d', whereby the finally obtained picture will change once in every 0.5

seconds. If the final protection level is C, then the video reproduction control section 15d' is controlled such that the contents of the frame memory are updated only once in every 60 frame periods, i.e. the finally obtained picture will change only once in every 2 seconds. If the final protection level is D, then only the video data of certain specific frames (or one specific frame) are written into the frame memory of the video reproduction control section 15d' and read out to obtain the final output video signal. In that way, for example, only a portion of the video data (e.g. a portion which is not subject to copyright protection) will be displayed. If the protection level is E, then the video data for a predetermined picture are written into the frame memory of the video reproduction control section 15d' and repetitively read out, to display only that predetermined picture, which can be for example a warning message concerning copyright protection.

Alternatively, control of the degree of reproduction limitation can be performed by arranging that the protection control signal from the protection control signal generating section 14b acts on the video reproduction control section 15d' such as to vary (in accordance with the final protection level) the number of gradations provided by each video data sample, i.e. to vary the number of amplitude levels that can be expressed by each sample. That can be performed by setting one or more low-order bits of each data sample to a fixed value, e.g. 0. For example if the LSB is always set to 0, then the number of possible gradations is reduced by half, and a corresponding lowering of resolution of the finally obtained display picture is achieved. The bits in each digital data sample which are not fixed in that way will be referred to in the following as the effective bits of the sample. Such gradation control based on the numbers of effective data bits is illustrated by the right-side column in Fig. 6. In that example, if the final protection level is A, then the protection control signal from the protection control signal generating section 14b has no effect on the operation of the video reproduction control section 15d', so that each video data sample used to form the output video signal has the standard number of effective bits, i.e. 8 bits. If the final protection level is B, then the protection control signal controls the video reproduction control section 15d' such that the number of effective bits/sample of the output video signal is reduced to 4 (i.e. by rounding-off the low-order 4 bits to zero). If the final protection level is C, then the protection control signal controls the video reproduction control section 15d' such that the number of effective bits/sample is 2 (i.e. all except the two high-order bits are set to zero), so that the picture resolution is further degraded. Similarly, if the final protection level is D, then the video reproduction control section 15d' is controlled such that the number of effective bits is reduced to 1. If the protection level is E, then the video data for a predetermined picture are written

into the frame memory of the video reproduction control section 15d' and repetitively read out as the final output video signal, to display only that predetermined picture.

It would be equally possible, as indicated by the broken-line connection from the protection control signal generating section 14b to the dequantizer section 15b in Fig. 7, to arrange that the protection control signal from the protection control signal generating section 14b acts on the dequantizer section 15b such as to vary (in accordance with the final protection level) the number of effective bits of each output datum from the dequantizer section 15b. That will provide a similar effect to that described above for the case in which control is effected through the video reproduction control section 15d'.

Another method which may be used to control the degree of reproduction limitation is to apply the protection control signal from the protection control signal generating section 14b such as to control the variable-length decoding section 15a. In that case, the protection control signal is arranged to act on the variable-length decoding section 15a such that, as the protection level is increased from A to E, data having a long code length are set to zero, i.e. are ignored. This will result in a lowering of resolution in the final picture that is obtained from the output video signal.

It should be noted that reproduction limitation control can be executed by a combination of control acting along the time axis and control acting in a spatial domain (i.e. within individual frames). That point is illustrated conceptually in Fig. 8, in which successive vertical lines 20 represent sequential frames of the video signal that is recorded on the CD 1. (For simplicity of description, it will be assumed that the final protection level is identical to the medium protection level). Together with each video signal frame data portion on the CD 1, a medium protection data portion is recorded, which may include position information specifying a region within the frame within which display resolution is to be lowered, to a degree that is in accordance with the final protection level. In this example there are two possible basic display conditions for each frame, i.e. non-display or display (with one or more degraded resolution regions possibly being formed). To achieve this, the medium protection data assigned to each frame includes a 1-bit flag, whose 1 or 0 logic state designates either display or non-display for the frame. If that flag bit indicates that none of the frame is to be displayed, the condition is indicated by a "x" symbol in Fig. 8, while if the flag bit indicates that the frame is to be completely or partially displayed, that condition is indicated by a "o" symbol in Fig. 8. In the example of Fig. 8, a "o" condition is specified for each of the ten consecutive frames designated as F_A , indicating that each frame is to be displayed. In addition, the medium protection data of each of these frames includes position information for

a degraded resolution region. The resultant display picture is designated by numeral 21, containing a degraded resolution region 22, which is rectangular and is shown as a hatched-line region. The degraded resolution region 22 is formed by a plurality of 16 x 16 element macroblocks, and for each of the frames F_A , the corresponding position information in the medium protection data specifies two addresses of macroblocks (designated as 22a and 22b, located at the upper left-side and lower right-side corners of the mosaic region 22) within the frame, to thereby specify the position and size of a rectangular region which is the degraded resolution region 22.

Similarly, a set of four successive frames F_B is each to be displayed, but with a degraded resolution region formed in the final display picture, as indicated by numeral 24. In this case the degraded resolution region is formed of two adjoining rectangular regions, so that it is necessary for the position information in the medium protection data to specify the positions of two pairs of macroblocks within the frame, i.e. the pair of addresses of macroblocks 24a, 24b and the pair of addresses of macroblocks 24c, 24d in Fig. 8.

It can thus be understood that in this case, reproduction protection is applied by a combination of control with respect to the time axis, and control with respect to (2-dimensional) space within each frame. It can be further understood that the invention enables extremely precise control of reproduction limitation, which is determined in accordance with the final protection level.

With data transmission in accordance with the MPEG1 system (i.e. based on the ISO-11172-3 standards), data are transmitted as successive packs of data, which are time-division multiplexed, as illustrated by the data flow 40 shown in Fig. 9. Each pack is made up of a leading portion such as the portion 41, which contains information including a pack start code and a stream identifier which distinguishes the data conveyed by that pack from that of other packs (e.g. to distinguish between video, audio or other data), and a main data portion such as portion 42. In this example each main portion consists of either compressed encoded video data such as portion 42, encoded audio data such as portion 43, or protection data such as portion 44. In this example it will be assumed that protection of the form shown in Fig. 8 is applied to the video data, so an individual protection data portion may be assigned to each of a plurality of video signal frames, i.e. video data frames can be conveyed by respective packs, each preceded by a protection data pack. In that case, each protection data pack in the example of Fig. 9 consists of a 3-bit portion which specifies the medium protection level, a 1-bit frame flag specifying whether or not the frame is to be displayed (as described above for Fig. 8), a portion which specifies the number of macroblock start/end address pairs (to be utilized when at least

one degraded resolution region is to be formed within the frame, e.g. as for each of the frames F_A and F_B in Fig. 8), followed by the pairs of macroblock start/end addresses (e.g. the pair of addresses of macroblocks 24a, 24b, then the pair of addresses of macroblocks 24c, 24d, for each of the frames F_B in the example of Fig. 8).

Fig. 10 shows a specific internal configuration for the video reproduction control section 15d' of the embodiment of Fig. 7. In Fig. 10, the output data from the inverse transform section 15c are supplied via a line 50 to a gradation control section 51, which is controlled by one of two protection control signals that are generated from the protection control signal generating section 14b, and resultant output data from the gradation control section 51 are transferred through a switch 52, which is controlled by the other one of the protection control signals. Data transferred through the switch 52 are written into a frame memory 53, and are subsequently read out from the frame memory 53 in the appropriate sequence to constitute successive frames of the original video signal. The digital video signal thereby produced from the frame memory 53 is supplied to a digital/analog converter 54, to obtain an analog video signal as the final output signal.

So long as the switch 52 is held closed, the contents of the frame memory 53 will be completely updated once in each frame period of the video signal, so that data of a new frame will be sequentially read out from the frame memory 53. However if the switch 52 is held closed during an integral number of frame intervals, then the most recently stored contents of the frame memory 53 will be repetitively read out during each of these frame intervals, i.e. the last frame will be continuously outputted. It will thus be apparent that this circuit can implement the time-axis protection operation described above on successive frames, if the switch 52 is controlled in accordance with the status of the frame bit that is contained in the medium protection data.

The gradation control section 51 operates on each digital video signal sample (in general, each 8-bit datum) that is supplied from the inverse transform section 15c, to set the low-order bits of each sample in accordance with the final protection level. For example referring to the right-side column in Fig. 6, if the final protection level is B, then the gradation control section 51 sets all of the four low-order bits of each sample to a predetermined value, e.g. 0. If the protection level is D, then all of the seven low-order bits of each sample are set to 0.

If reproduction restriction is to be applied within a video signal frame, then the protection control signal generating section 14b responds to the protection position signal such as to apply the above-mentioned protection control signal to the gradation control section 51 during one or more specific time intervals within the corresponding frame interval, with each of

these specific time intervals being determined based on one of the macroblock start/end address pairs which are shown in Fig. 9, described above. In that way, reproduction limitation can be applied within specific regions of a frame, as illustrated in Fig. 8, i.e. the spatial-domain protection operation described above can be applied.

It will be understood that in practical terms, each "start address" will define a time-axis position, within a frame period, of the data sample corresponding to an uppermost left-side pixel of a rectangular region within the frame, while the "end address" similarly defines the position of a data sample corresponding to a lowermost right-side pixel of that region. Such time relationships can be readily established by well-known techniques for operating on a digital video signal, so that detailed description is omitted.

In the above embodiments, only reproduction protection of video data has been described. Fig. 11 shows another embodiment of the invention, in which reproduction protection of both audio and video data is applied. Only the points of difference between this embodiment and previous embodiments will be described. A CD reproduction apparatus 5 of this embodiment differs from that of Fig. 7 by including circuits for decoding and dequantizing an encoded digital audio signal that has been recorded on the CD 1, by an audio decoding section made up of a variable-length decoding section 16a, a dequantizer section 16b, a sub-band combining section 16c and an audio reproduction control section 16d. The compressed encoded audio data are separated from the video and protection data contained in the input data stream, by the demultiplexer 10, and are supplied as input data to the variable-length decoding section 16a, with an output audio signal being produced from the audio reproduction control section 16d. This embodiment further differs from that of Fig. 7 in that the protection control signal generating section 14c of this embodiment produces not only a first protection control signal which acts on either the video reproduction control section 15d' or dequantizer section 15b to apply video signal reproduction protection by varying the number of bits per datum, as described hereinabove for the protection control signal of the embodiment of Fig. 7, but also a second protection control signal which acts on the audio reproduction control section 16d or dequantizer section 16b to apply audio signal reproduction protection, as described in the following. For simplicity of description, it will be assumed that the second protection control signal is produced in accordance with the final protection level that is derived for reproduction protection of the video data, as described hereinabove. However in general, separate medium protection levels and separate apparatus protection levels would be specified for the video and audio data, i.e. to obtain separate final protection levels for video and audio data.

Fig. 12 is a table illustrating three possible methods of applying audio signal reproduction protection with the embodiment of Fig. 11. With each of the three examples shown in Fig. 12, five different audibility grades can be selected for the output audio signal produced from the audio reproduction control section 16d, in accordance with the final protection level, to effect audio signal reproduction protection. Firstly, the method illustrated by the leftmost column in Fig. 12 will be described. In this case, audio signal reproduction protection is applied by selectively restricting the bandwidth of the output audio signal produced from the audio reproduction control section 16d in accordance with the final protection level. If MPEG1 audio signal compression is used, then assuming a sampling frequency of 48 KHz, a bandwidth of 24 KHz is available for the output audio signal. In this example, if the final protection level is A, then no bandwidth restriction is applied, i.e. the audio bandwidth is 24 KHz. If the final protection level is B, then the audio signal bandwidth is restricted to 18 KHz, if the protection level is C the bandwidth is restricted to 12 KHz, if the protection level is D the bandwidth is restricted to 6 KHz, and if the protection level is E then no audio output signal is produced.

Since MPEG1 audio compression utilizes sub-band encoding with 32 bands, such bandwidth restrictions can be effected by causing the second protection control signal to act on the dequantizer section 16b such as to set the inverse quantization values corresponding to certain high-frequency bands to zero. Thus, audio reproduction protection by bandwidth control can be easily implemented.

A second method of audio signal reproduction protection will be described referring to the central column in Fig. 12. In this case, time-axis protection is applied, by "thinning-out" of audio signal sample values that are supplied from the sub-band combining section 16c and used in the audio reproduction control section 16d to obtain the output audio signal. In this example, if the final protection level is A, then all of the audio sample values are used in deriving the output audio signal. If the final protection level is B, then one in every two samples is held for two consecutive sample periods, by a sample-and-hold circuit, i.e. only half of the total samples are used in deriving the output audio signal. If the protection level is C, then only one in every three samples is used in deriving the output audio signal, i.e. one in every three successive samples is held for three consecutive sample periods. If the protection level is D, then only samples which occur during a specified interval are used in producing the output audio signal. For example, this operation could be performed when only a specified part of the recorded audio signal is to be allowed (by the copyright owner) to be reproduced. If the final protection level is E, then no audio output signal is produced.

A third method of audio signal reproduction protection will be described referring to the rightmost column in Fig. 12. In this case, protection control is executed by effecting control of the number of gradation levels provided by the audio data samples, in a similar manner to that described hereinabove for the video data. Generally, a digital audio signal has 16 bits/sample. When the final protection level is A, then all of these 16 bits are utilized, i.e. there is no limitation of audibility. If the final protection level is B, then the number of effective bits/sample is reduced to 12 (i.e. the low-order 4 bits of each 16-bit sample are fixed at 0), causing a lowering of quality of the reproduced audio signal. If the protection level is C, the number of effective bits is further reduced to 8, if the final protection level is D then the number of effective bits/sample is reduced to 4, and if the final protection level is E, then no audio output signal is produced.

It can thus be understood that the invention enables precise limitation of reproduction of a recorded audio signal together with limitation of reproduction of the recorded video signal, in accordance with a combination of medium protection data and apparatus protection data.

Fig. 13 shows an example of the internal configuration of the audio reproduction control section 16d of this embodiment. For the purpose of description, it is assumed that each of the above-mentioned three methods of protection control of the audio reproduction control section 16d is utilized, although in practice only one of these could be utilized. The circuit consists of a digital low-pass filter 61 which receives the output data samples from the sub-band combining section 16c via an input line 60, a sample-and-hold circuit 62 which can be controlled to hold and output each data sample for a specific interval, a gradation control section 63 which effects the aforementioned control of low-order bits of each audio data sample, to thereby control the gradation levels which can be expressed by each sample, a memory 64 for temporarily holding successive data samples, and a digital-to-analog converter 65 for converting the digital audio data to an analog audio signal. The gradation control section 63 is controlled by a protection control signal to provide varying degrees of gradation in accordance with the final protection level that has been determined for the audio data, by setting varying numbers of low-order bits of each 16-bit audio data sample to a fixed value as described above.

As shown, the protection control signal can also be applied to control the sample-and-hold circuit 62, to effect the above-described method of reproduction protection control utilizing sample-and-hold processing of the audio data samples. Similarly, the protection control signal can be applied to control the LPF 61, to achieve reproduction protection control by varying the bandwidth of the audio signal. It will be clear that a simpler circuit configuration can be utilized.

ized that that shown in Fig. 13, if only one of the above three methods of protection control is applied.

In each of the embodiments described above, there is a fixed relationship pattern between combinations of the protection levels which can be expressed by the medium protection data and the protection levels which can be expressed by the apparatus protection data, and the respective final protection levels which are thereby obtained, i.e. the relationship pattern which is stored in the matrix ROM of the final protection level determining section 13, an example of which is shown in Fig. 2. However in some cases there may be a requirement for enabling such a relationship pattern to be selected from a number of different relationship patterns, which have varying degrees of protection severity. Specifically, it may be advantageous to provide the reproduction apparatus with a switch which can be operated by the person who is in charge of the reproduction apparatus, such that the switch can be used to select from a plurality of different relationship patterns, so that the degree of reproduction protection can be flexibly determined by that person. In that way, the person in charge of the reproduction apparatus can ensure that the reproduction protection will be appropriate for the viewing audience. For example, an adult can set the switch such as to ensure that unsuitable scenes cannot be viewed by any children who may use the reproduction apparatus. In that case, assuming for example that the relationship pattern which is the least severe is that shown in Fig. 2, the entire range of possible relationship patterns is illustrated in the table of Fig. 14. Here, for each combination of protection levels obtained from the medium protection data and apparatus protection data, there is a corresponding range of one or more possible values of final protection level, with that range extending from a least severe value to the most severe value (i.e. protection level E). One method of implementing such a capability would be to provide a plurality of matrix ROMs (or to define a plurality of separate matrix regions in a ROM), for storing the respectively different relationship patterns. However an embodiment of the invention will now be described whereby such a capability can be easily implemented by a simple modification of any of the embodiments of the invention that have been previously described.

The embodiment, which is a modification of the embodiment of Fig. 5, is shown in Fig. 15. The embodiment differs from that of Fig. 5 in being provided with a severity modification circuit 30 and a severity modification setting switch 31. The severity modification circuit 30 is connected between the final protection level determining section 13 and the protection control signal generating section 14a, and functions to selectively modify each protection level value which is read out from the matrix ROM of the final protection level determining section 13 (as described hereinabove for the embodiment of Fig. 5), and to

supply a resultant modified final protection level to the protection control signal generating section 14a. The severity modification setting switch 31 can be adjusted by the person who is in charge of the CD player, to select one of five possible switch conditions which will be designated as P_A to P_E, respectively. The severity modification setting switch 31 is coupled to control the severity modification circuit 30 such that the severity modification circuit 30 executes protection level modification in accordance with the specific position at which the switch is set, as described in the following.

The relationships between the five possible values of the medium protection level (determined by the medium protection signal detection section 11 as described hereinabove) and the five positions P_A to P_E of the severity modification setting switch 31, are shown for each of the four possible values of the apparatus protection level (determined by the apparatus protection signal generating section 12), in diagrams (A) to (D) in Fig. 16. Referring first to diagram (A), if the switch position is P_A, then the relationship between the medium protection level values, the apparatus protection level values and the final protection level values is left unchanged from those of the corresponding column in Fig. 2, i.e. this setting of the severity modification setting switch 31 provides the least severe degree of reproduction protection. If the switch position is set to P_B, then the least severe value which can be taken by the modified final protection level becomes level B. That is to say, if a protection level A is established by the final protection level determining section 13, that is changed by the severity modification circuit 30 to a modified final protection level B. If the switch position is P_C, then the least severe value of the modified final protection level is changed to C. If the switch position is P_D, then the least severe value of the modified final protection level is changed to D, and if the switch position is P_E, then the modified final protection level is fixed as E.

The same is true for each of the apparatus protection level values 2, 3 and 4, as illustrated in diagrams (B), (C) and (D) in Fig. 16, which correspond to the second, third and fourth columns in Fig. 2 respectively.

It will be apparent that the severity modification circuit 30 can be easily configured using a logic circuit, which implements a simple algorithm in accordance with the setting of the severity modification setting switch 31, i.e. the algorithm would begin:

[If the severity modification switch 31 is set at P_A, transfer the protection level value established by the final protection level determining section 13 directly to the protection control signal generating section 14a, as the final protection level.

If the severity modification switch 31 is set at P_B, and if the protection level value established by the final protection level determining section 13 is level A,

change that to level B and transfer to the protection control signal generating section 14a as the (modified) final protection level. Otherwise, transfer the protection level value produced from the final protection level determining section 13 unchanged, as the final protection level.

If the severity modification switch 31 is set at P_C , and if the protection level established by the final protection level determining section 13 is level A or level B, change to level C, and transfer to the protection control signal generating section 14a as the final protection level. Otherwise, transfer the protection level established by the final protection level determining section 13 unchanged, as the final protection level], and so on.

Thus with this embodiment, if the severity modification setting switch 31 is set to its least severe position (P_A), then the relationship pattern between combinations of the medium protection level and reproduction apparatus protection level values will be as shown in Fig. 2. If the severity modification setting switch 31 is set to the most severe position (P_E), then the relationship pattern will be such that the final protection level will always be the highest level, i.e. level E. As the severity modification setting switch 31 is successively changed from positions P_A to P_E , the least severe degree of reproduction restriction (within the range of possible degrees of restriction which can be set by the final protection level) is increased to a more severe degree, by successive steps.

For example if the apparatus protection level is 2 and the medium protection level is 4, then the final protection level will be B. However by altering the setting of the severity modification setting switch 31, the user can change the final protection level to a higher value, in the range B to E. Hence with this embodiment, although protection level values can be specified by the manufacturer or copyright owner of the recording medium, and by the manufacturer or seller of the reproduction apparatus, the final degree of protection can be determined by the person who is in charge of the reproduction apparatus. Such a feature is highly useful.

As can be understood from the above description of embodiments, the invention enables a final protection level to be established, for controlling reproduction of recorded or transmitted video or audio signals, with that final protection level being determined based on a combination of protection levels which are respectively separately established by the manufacturer or copyright owner of recorded signals (or broadcaster of transmitted signals, or copyright owner of transmitted signal) and by the manufacturer or seller of the reproduction apparatus. The final protection level can be applied such as to achieve extremely precise protection of reproduction, whereby for example specific frames of a video signal, and/or specific regions within a frame, can be protected by restricting

reproduction, with the degree of restriction being variable in a stepwise manner. The invention can at the same time provide corresponding protection of an audio signal which is being reproduced in conjunction with a video signal.

Although the invention has been described in the above with reference to a CD player apparatus, it will be understood that the invention is not limited in any way to such an apparatus, and is in general applicable to reproduction protection in any type of apparatus which reproduces a recorded or transmitted video and/or audio signal.

Claims

1. A reproduction protection method comprising:
 - attaching medium protection data, which are specific to a data medium, to main data which represent an original signal, and conveying said medium protection data and main data by said data medium;
 - supplying said main data and medium protection data via said data medium to a reproduction apparatus;
 - generating apparatus protection data which are specific to said reproduction apparatus, within said reproduction apparatus;
 - determining a protection level by combining the medium protection data and the apparatus protection data; and
 - controlling said reproduction apparatus to utilize said main data to reproduce said original signal in accordance with said protection level.
2. A reproduction apparatus providing reproduction protection, for operating on main data which are conveyed by a data medium and represent an original signal and on medium protection data which are specific to said data medium and are conveyed by said data medium, the apparatus comprising:
 - means for generating apparatus protection data which are specific to said reproduction apparatus;
 - means for defining a protection level based on said medium protection data and apparatus protection data in combination; and
 - means for executing reproduction of said original signal by utilizing said main data, including means for selectively restricting said reproduction in accordance with said protection level.
3. A reproduction apparatus as claimed in claim 2, wherein the apparatus further includes:
 - means (10,11) for detecting said medium protection data to obtain a medium protection signal expressing said medium protection level;

said generating means comprises means (12) for generating an apparatus protection signal expressing an apparatus protection level which has been assigned to said reproduction apparatus;

said defining means comprises means (13) responsive to said medium protection signal and apparatus protection signal for determining a final protection level in accordance with a combination of said medium protection level and apparatus protection level, and generating a final protection level signal expressing said final protection level.

4. A reproduction apparatus providing reproduction protection, for operating on main data which are conveyed by a data medium and represent an original signal and on medium protection data which are specific to said data medium and are conveyed by said data medium, the apparatus comprising:

means (10,11) for detecting said medium protection data to obtain a medium protection signal expressing said medium protection level;

means (12) for generating an apparatus protection signal expressing an apparatus protection level which has been assigned to said reproduction apparatus;

means (13) responsive to said medium protection signal and apparatus protection signal for determining a final protection level in accordance with a combination of said medium protection level and apparatus protection level, and generating a final protection level signal expressing said final protection level;

means (15) for utilizing said main data to execute reproduction of said original signal, including means (14) responsive to said final protection level signal for selectively restricting said reproduction in accordance with said final protection level.

5. A reproduction apparatus according to claim 2, 3 or 4, wherein said main data are conveyed by said data medium after having been compressed and encoded, wherein said reproduction apparatus includes means (15a, 15b, 15c) for decoding and means for decompressing said main data, and wherein said means for selectively restricting reproduction of the original signal comprise means for controlling at least one of said decoding means and said decompressing means in accordance with said final protection level.
6. A reproduction apparatus according to claim 5,

operation upon each of fixed-size blocks of pixel values within each frame of a video signal, wherein said decompressing means includes means (15c) for applying an inverse transform operation, and wherein said means for selectively restricting reproduction of the original signal serves to selectively eliminate specific ones of a plurality of transform coefficients utilized in executing said inverse transform operation.

7. A reproduction apparatus according to any of claims 2 to 6, wherein said main data are conveyed by said data medium after having been compressed, quantized and encoded, wherein said reproduction apparatus includes means (15b) for dequantizing said main data, and wherein said means for selectively restricting reproduction of the original signal comprise means for controlling said dequantizing means in accordance with said final protection level.

8. A reproduction apparatus according to any of claims 2 to 7, wherein said means for selectively restricting reproduction of the original signal comprise means (52, 62) for periodically omitting selected portions of said main data from use by said reproduction control means in reproducing said original signal.

9. A reproduction apparatus according to claim 8, wherein each of said selected portions comprises at least one digital data sample.

10. A reproduction apparatus according to claim 8 or claim 9, wherein said main data express a digital video signal, and wherein each of said selected portions comprises at least one video signal frame.

11. A reproduction apparatus according to any of claims 2 to 10, wherein said means for selectively restricting reproduction of the original signal comprises gradation control means (51, 63) for selectively varying a number of signal amplitude gradations which can be expressed by each data sample of said main data.

12. A reproduction apparatus according to any of claims 2 to 11, wherein said means for selectively restricting reproduction of the original signal comprises controllable digital low-pass filter means (61) for filtering said main data, to thereby controllably vary a bandwidth of a reproduced original signal obtained from said apparatus.

13. A reproduction apparatus according to any of claims 2 to 12, wherein said medium protection data specify a medium protection level and pro-

tection position information, and wherein said protection position information specifies a portion of said main data which is to be subjected to reproduction restriction in accordance with said final protection level.

5

14. A reproduction apparatus according to claim 13, wherein said main data comprise a digital video signal, and wherein said protection position information specifies at least one region within each of a plurality of frames of said video signal.

10

15. A reproduction apparatus according to any of claims 2 to 14, further comprising protection severity modification means, operable for modifying said final protection level which is determined by said combination of medium protection data and apparatus protection data, to obtain a modified final protection level.

15

20

16. A reproduction apparatus according to claim 15, wherein said severity modification means comprise a manually operable switch (31) provided on said reproduction apparatus, and severity modification circuit means (30) coupled to said switch and coupled to receive a signal representing said final protection level, for modifying said final protection level in accordance with a setting position of said switch and thereby producing an output signal expressing a modified final protection level.

25

30

17. A reproduction apparatus according to claim 16, wherein said final protection level has a value which defines one of a plurality of successively increasing degrees of restriction of reproduction of said original signal, extending between a minimum and a maximum degree of restriction, and wherein said severity modification means is responsive to a change in said setting position of the switch for changing only said minimum degree of restriction.

35

40

18. A data medium for transferring main data representing an original signal to a reproduction apparatus, to reproduce said original signal by said reproduction apparatus, wherein said data medium further transfers medium protection data which are specific to said data medium and wherein said reproduction apparatus generates apparatus protection data which are specific to said reproduction apparatus, said medium protection data and apparatus protection data in combination specifying a degree of restriction on said reproduction of the original signal by said reproduction apparatus.

45

50

55

15

FIG. 1

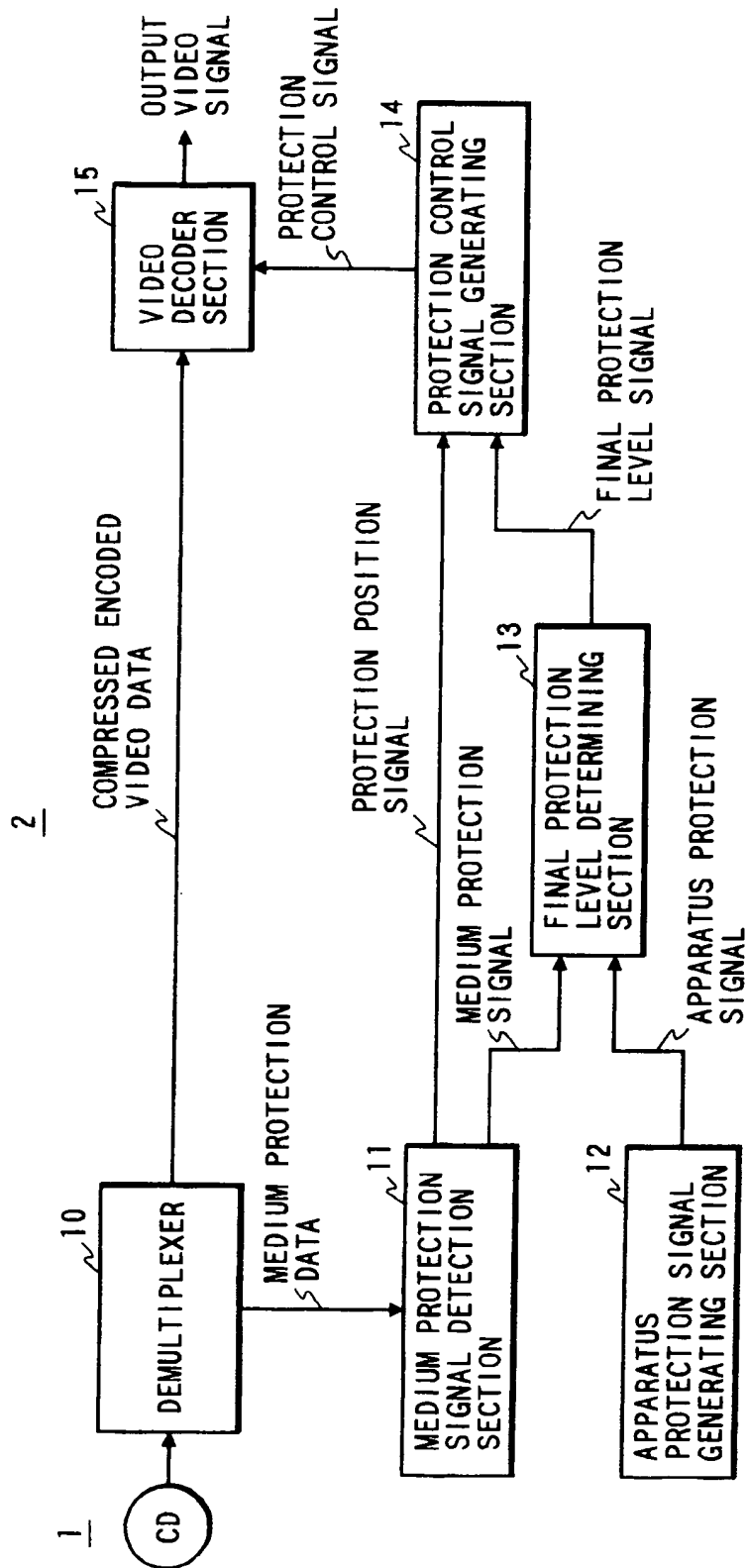


FIG. 2

REPRODUCTION APPARATUS
PROTECTION LEVEL → STRONG

	1	2	3	4
1	A	A	A	A
2	A	A	A	B
3	A	A	B	C
4	A	B ~~~~~	C	D
5	E	E	E	E

MEDIUM PROTECTION
LEVEL ↓
STRONG

FIG. 3

MEDIUM PROTECTION LEVEL →

EXAMPLES OF MEDIUM PROTECTION	1	2	3	4	5
BASED ON U.S. MOVIE RATINGS	FREE	PG	R	X	OTHER
BASED ON COPYRIGHT PROTECTION	FREE	RESTRICTED (WEAK)	RESTRICTED (MODERATE)	RESTRICTED (STRONG)	OTHER

FIG. 4

APPARATUS PROTECTION LEVEL →

EXAMPLES OF PLAYBACK APPARATUS PROTECTION	1	2	3	4
BASED ON COUNTRY	U. S. A.	EUROPE	JAPAN	TAIWAN
BASED ON ADULT/CHILD STATUS	FOR ADULTS	FOR ADULTS	FOR ADULTS	FOR CHILDREN
BASED ON OBJECTIVES	FOR ITEMS TO BE SOLD	FOR ITEMS TO BE SOLD	FOR IN-STORE SALES DEMONSTRATION ITEMS	FOR IN-STORE SALES DEMONSTRATION ITEMS

FIG. 5

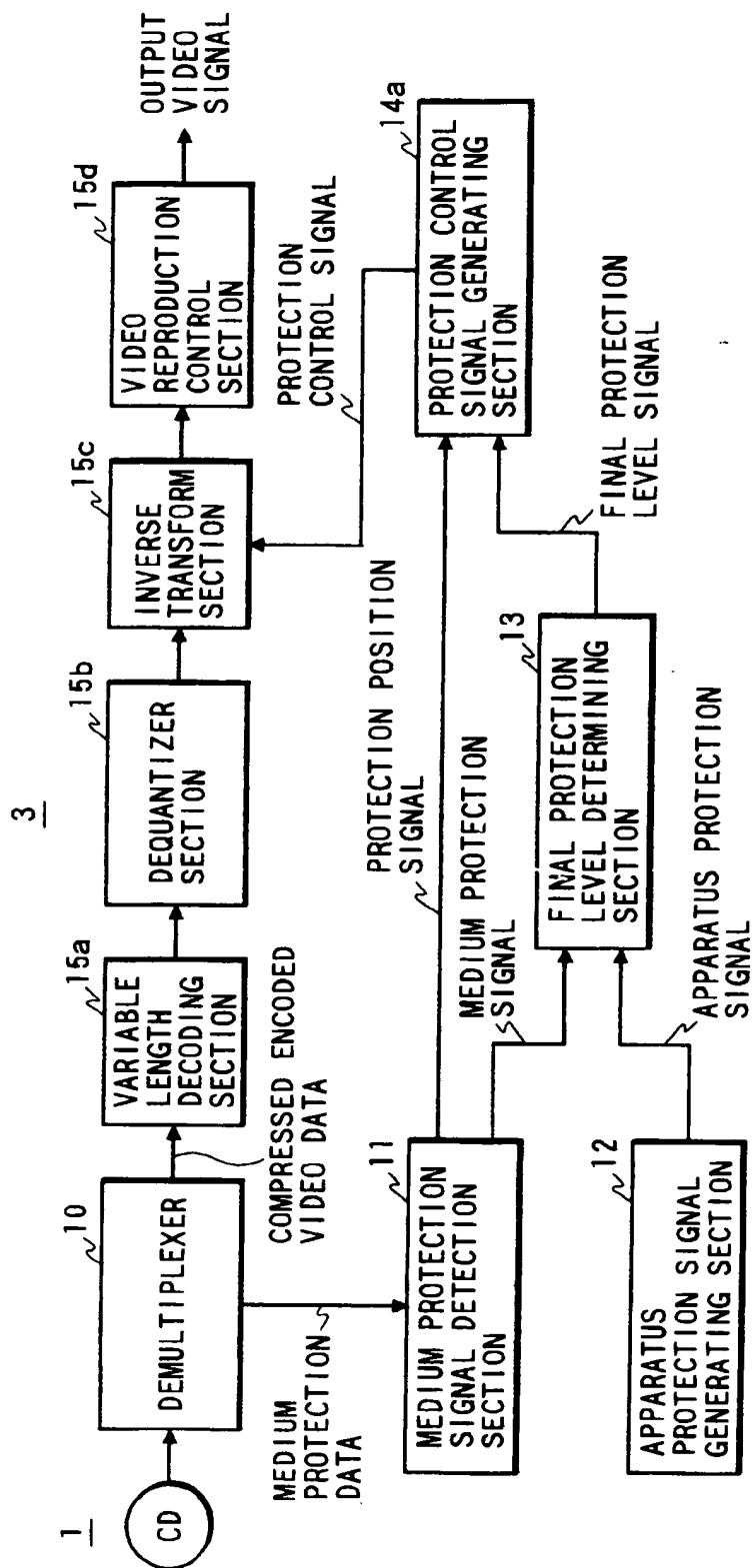


FIG. 6

	EXAMPLE OF VISIBILITY GRADES, USING SPATIAL-DOMAIN PROTECTION CONTROL	EXAMPLE OF VISIBILITY GRADES, USING TIME-AXIS PROTECTION CONTROL	EXAMPLE OF VISIBILITY GRADES, USING PROTECTION BASED ON VIDEO DATA BIT-NUMBER CONTROL
A	ALL PICTURE IS VISIBLE	ALL PICTURE IS VISIBLE	ALL PICTURE IS VISIBLE
B	8x8 TRANSFORM BLOCK SIZE, DC COEFFICIENT AND 2 AC COEFFICIENTS	1 IN 15 FRAMES USED	4 BITS/SAMPLE
C	8x8 TRANSFORM BLOCK SIZE, ONLY DC COEFFICIENT	1 IN 60 FRAMES USED	2 BITS/SAMPLE
D	16x16 TRANSFORM BLOCK SIZE, ONLY DC COEFFICIENT	SPECIFIED FRAME(S) ONLY USED	1 BIT/SAMPLE
E	OTHER PICTURE DISPLAYED	OTHER PICTURE DISPLAYED	OTHER PICTURE DISPLAYED


 FINAL PROTECTION LEVEL

FIG. 7

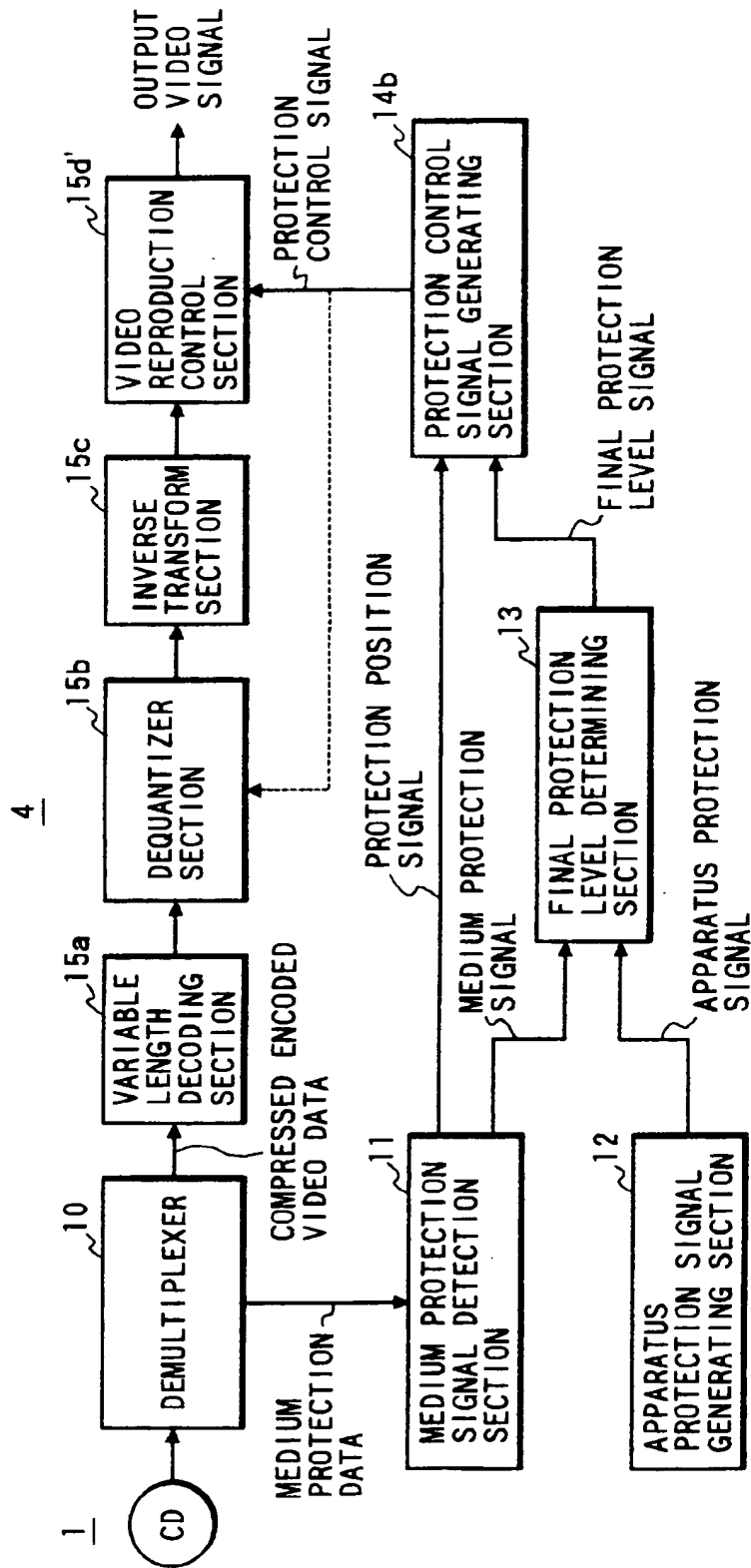


FIG. 8

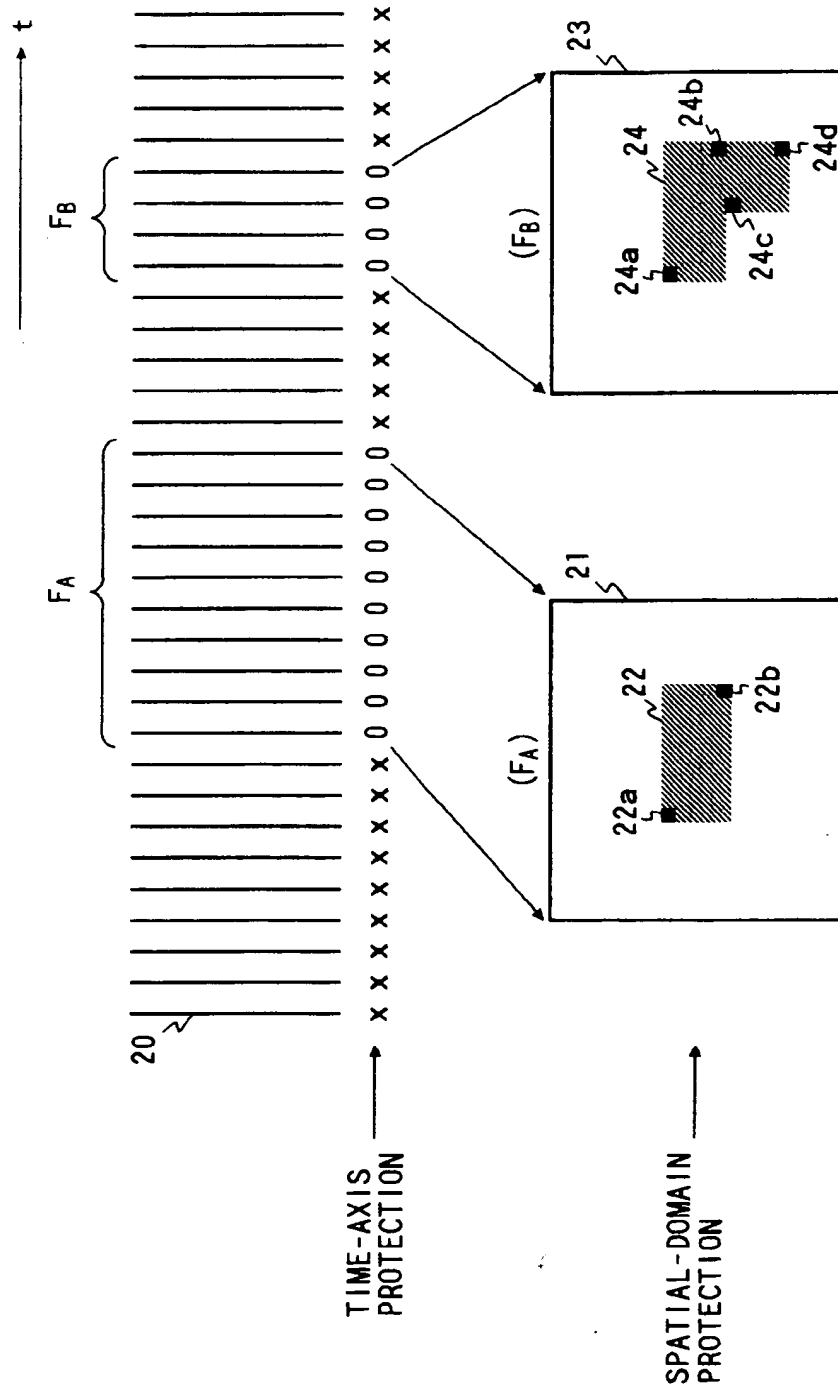


FIG. 9

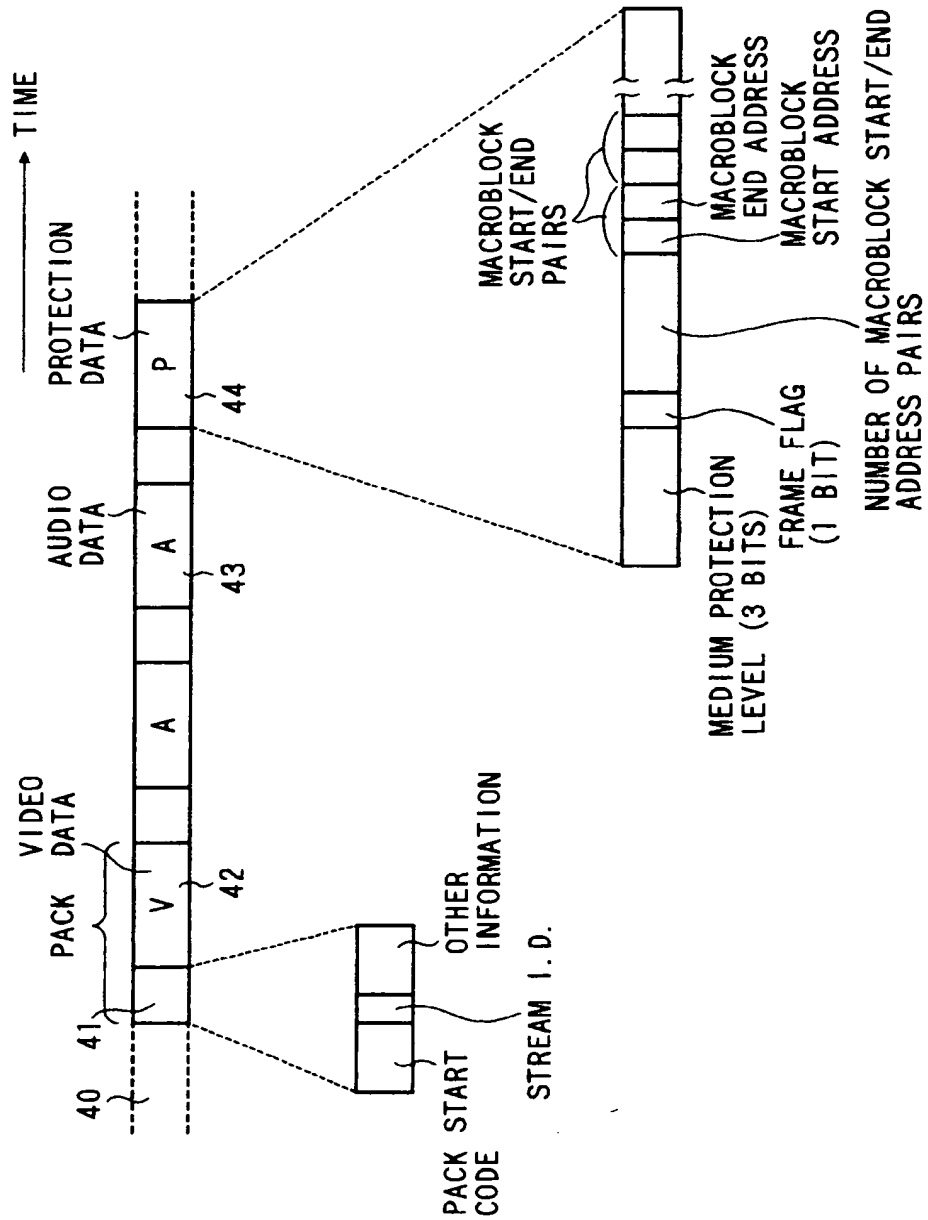


FIG. 10

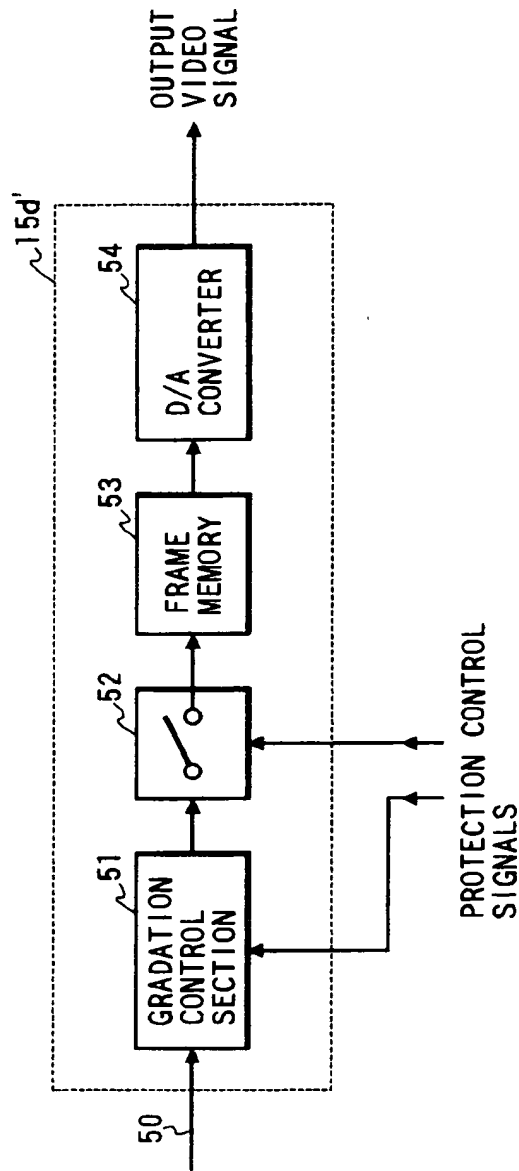


FIG. 11

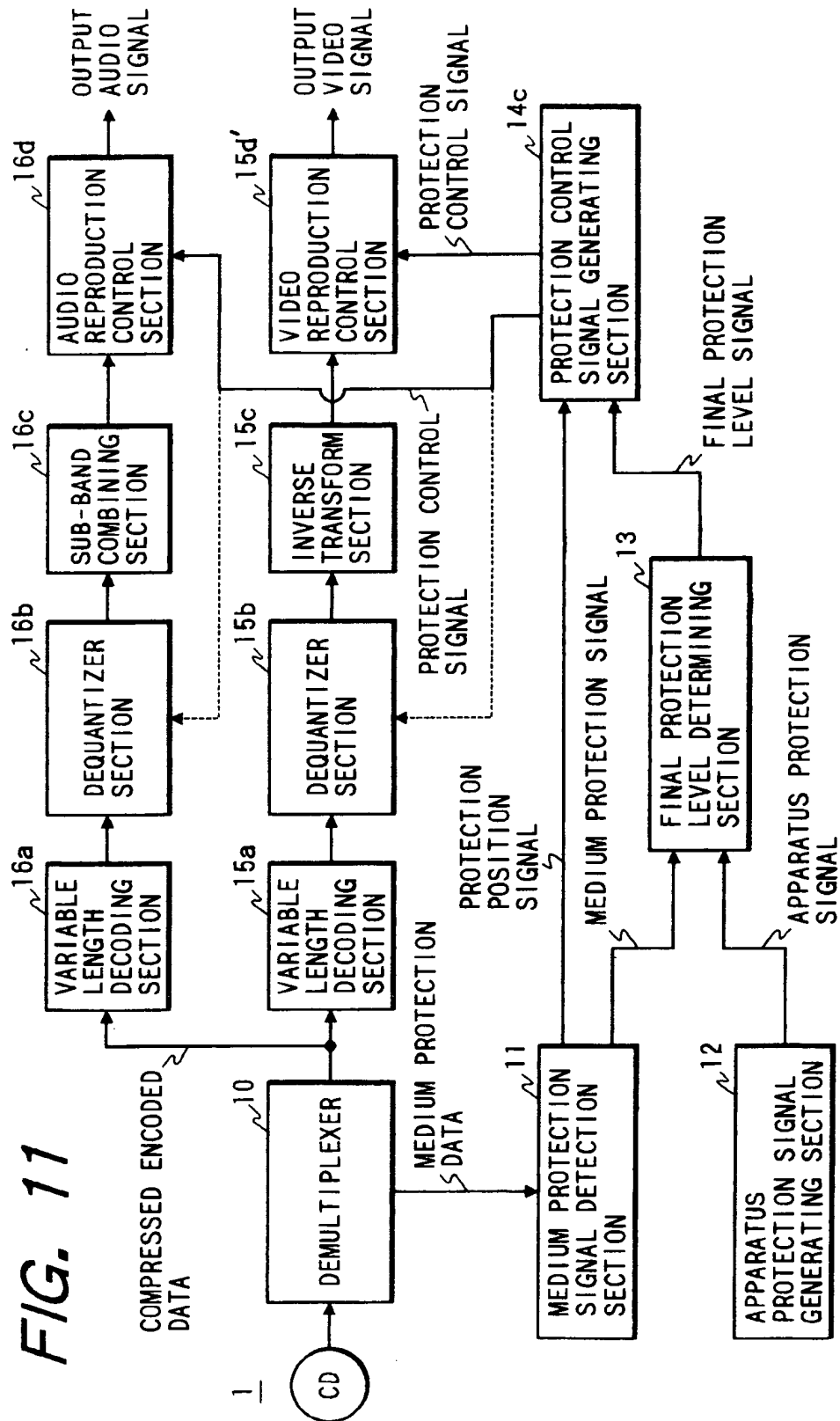


FIG. 12

	EXAMPLE OF AUDIBILITY GRADES, USING BANDWIDTH-RESTRICTION PROTECTION CONTROL	EXAMPLE OF AUDIBILITY GRADES, USING TIME-AXIS PROTECTION CONTROL	EXAMPLE OF AUDIBILITY GRADES, USING PROTECTION BASED ON AUDIO DATA BIT-NUMBER CONTROL
A	ALL OF ORIGINAL AUDIO SIGNAL REPRODUCED	ALL OF ORIGINAL AUDIO SIGNAL REPRODUCED	ALL OF ORIGINAL AUDIO SIGNAL REPRODUCED
B	18 KHz	1 IN EVERY 2 SAMPLES IS USED	12 BITS/SAMPLE
C	12 KHz	1 IN EVERY 3 SAMPLES IS USED	8 BITS/SAMPLE
D	6 KHz	ONLY SPECIFIED SAMPLES ARE USED	4 BITS/SAMPLE
E	NO AUDIO OUTPUT SIGNAL	NO AUDIO OUTPUT SIGNAL	NO AUDIO OUTPUT SIGNAL

→
FINAL PROTECTION LEVEL

FIG. 13

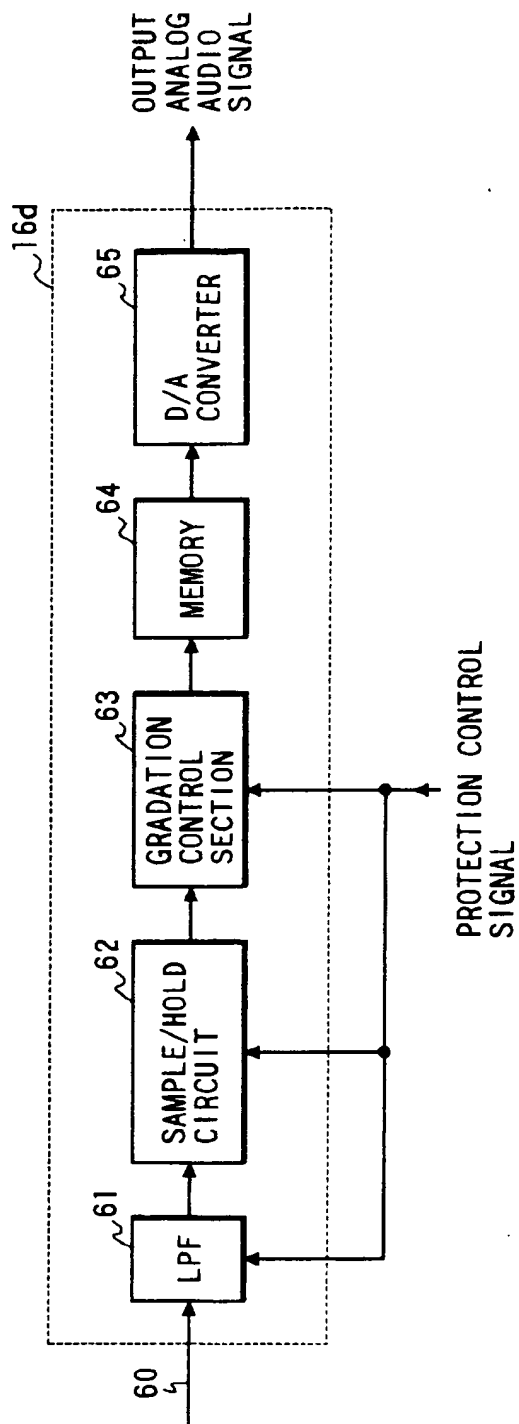


FIG. 14

REPRODUCTION APPARATUS
PROTECTION LEVEL → STRONG

		1	2	3	4
MEDIUM PROTECTION LEVEL ↓ STRONG	1	A~E	A~E	A~E	A~E
	2	A~E	A~E	A~E	B~E
	3	A~E	A~E	B~E	C~E
	4	A~E	B~E ~~~~~	C~E	D~E
	5	E~E	E~E	E~E	E~E

FIG. 15

6

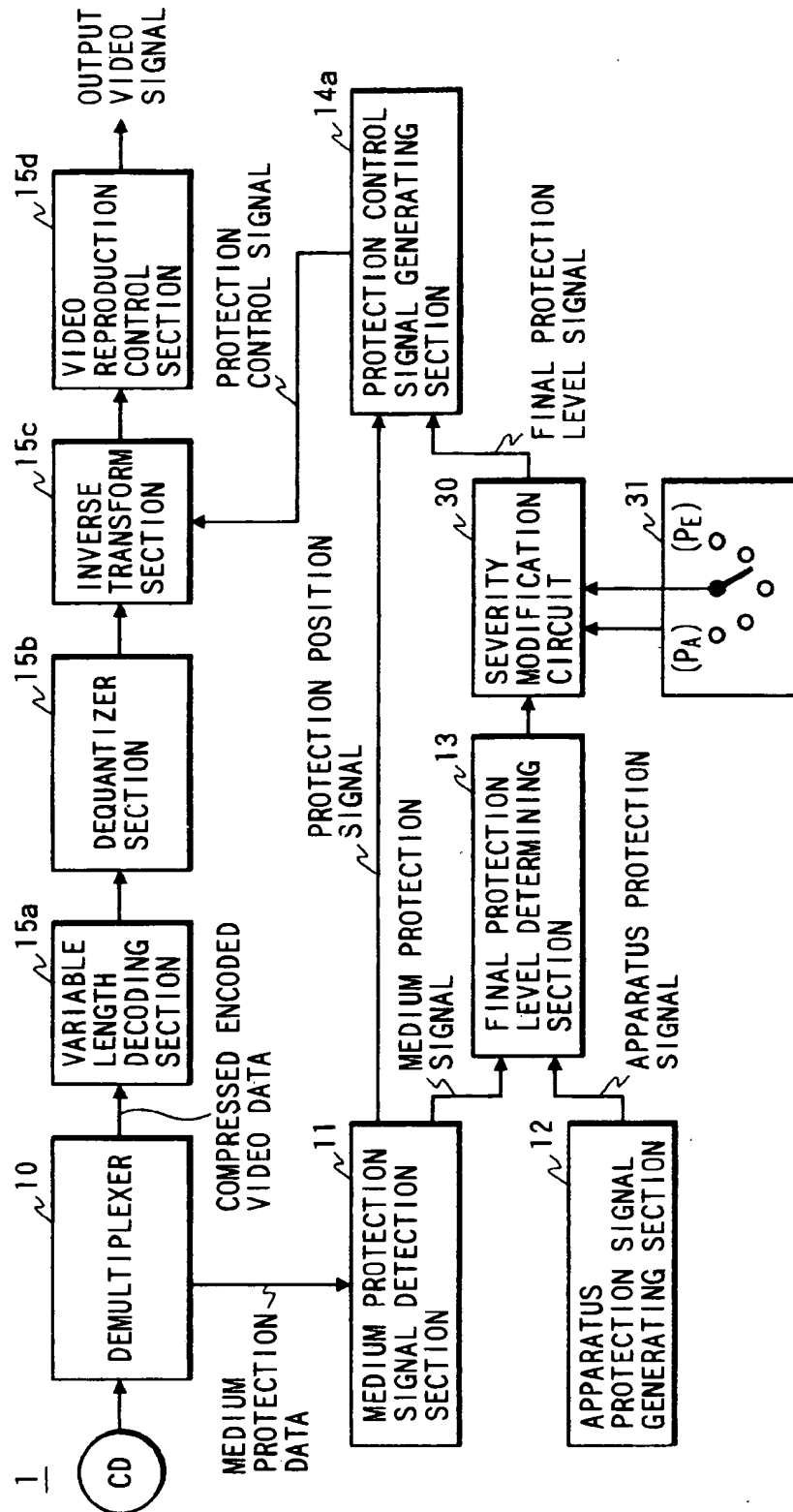


FIG. 16

(WHEN APPARATUS PROTECTION LEVEL=1)		MODIFICATION SWITCH CONDITION					
MEDIUM PROTECTION LEVEL ↓ STRONG		P _A	P _B	P _C	P _D	P _E	
	1	A	B	C	D	E	
	2	A	B	C	D	E	
	3	A	B	C	D	E	
	4	A	B	C	D	E	
	5	E	E	E	E	E	

(A)

(WHEN APPARATUS PROTECTION LEVEL=2)		MODIFICATION SWITCH CONDITION					
MEDIUM PROTECTION LEVEL ↓ STRONG		P _A	P _B	P _C	P _D	P _E	
	1	A	B	C	D	E	
	2	A	B	C	D	E	
	3	A	B	C	D	E	
	4	B	B	C	D	E	
	5	E	E	E	E	E	

(B)

(WHEN APPARATUS PROTECTION LEVEL=3)		MODIFICATION SWITCH CONDITION					
MEDIUM PROTECTION LEVEL ↓ STRONG		P _A	P _B	P _C	P _D	P _E	
	1	A	B	C	D	E	
	2	A	B	C	D	E	
	3	B	B	C	D	E	
	4	C	C	C	D	E	
	5	E	E	E	E	E	

(C)

(WHEN APPARATUS PROTECTION LEVEL=4)		MODIFICATION SWITCH CONDITION					
MEDIUM PROTECTION LEVEL ↓ STRONG		P _A	P _B	P _C	P _D	P _E	
	1	A	B	C	D	E	
	2	B	B	C	D	E	
	3	C	C	C	D	E	
	4	D	D	D	D	E	
	5	E	E	E	E	E	

(D)

Requested Patent: EP0725376A2

Title:

CHARGING METHOD AND CHARGING SYSTEM IN INTERACTIVE ON-LINE
SERVICE ;

Abstracted Patent: EP0725376 ;

Publication Date: 1996-08-07 ;

Inventor(s):

NAKANO HIROAKI (JP); NIIJIMA MAKOTO (JP); SONODA YUMIE (JP); KUMAGAI
YOSHIAKI (JP); NAGAHARA JUNICHI (JP); NASHIDA TATSUSHI (JP) ;

Applicant(s): SONY CORP (JP) ;

Application Number: EP19960300509 19960125 ;

Priority Number(s): JP19950017885 19950206 ;

IPC Classification: G07F19/00; G07F7/00 ;

Equivalents: JP8214281, US5845260 ;

ABSTRACT:

A charging method in interactive on-line service where a server and terminals of users are mutually connected by means of a transmission medium, and a predetermined service is provided from the server to the terminals via the transmission medium, and the fee for such service is collected from each user individually. The charging method comprises the steps of opening, in the server, an imaginary account for a child of the relevant user with a limited maximum amount, and withdrawing from the imaginary account the fee for the service provided to the user's child. The server can provide a predetermined service to the relevant user's child within a range of the limited maximum amount preset in the imaginary account. And when withdrawing the fee from the imaginary account, the server can restrict the service providable to the terminal. Thus, the parent enables his child to receive a desired on-line service, such as on-line shopping or video-on-demand, on the basis of the child's own judgment by setting an upper limit of a service utilisable by the child and still limiting the services providable for the child, hence realising promoted utilisation of the service by children.



(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
07.08.1996 Bulletin 1996/32

(51) Int Cl.⁶: **G07F 19/00, G07F 7/00**

(21) Application number: **96300509.5**

(22) Date of filing: **25.01.1996**

(84) Designated Contracting States:
DE FR GB NL

(30) Priority: **06.02.1995 JP 17885/95**

(71) Applicant: **SONY CORPORATION**
Tokyo 141 (JP)

(72) Inventors:
 • **Nagahara, Junichi, c/o Sony Corp.**
Shinagawa-ku, Tokyo (JP)
 • **Nashida, Tatsushi, c/o Sony Corp.**
Shinagawa-ku, Tokyo (JP)

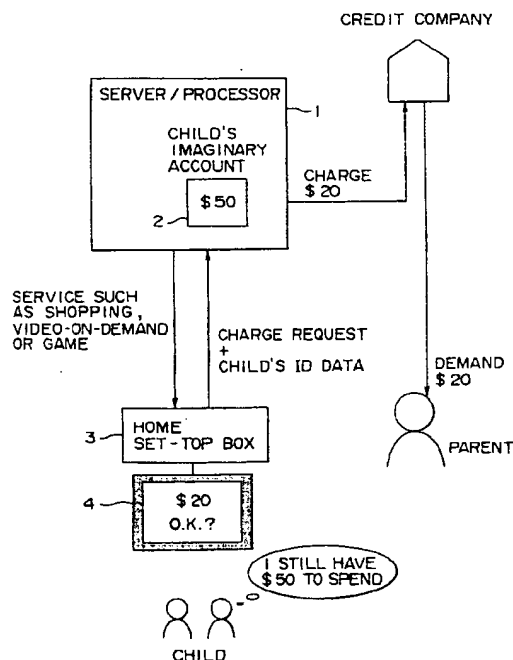
• **Nakano, Hiroaki, c/o Sony Corp.**
Shinagawa-ku, Tokyo (JP)
 • **Niijima, Makoto, c/o Sony Corp.**
Shinagawa-ku, Tokyo (JP)
 • **Sonoda, Yumie, c/o Sony Corp.**
Shinagawa-ku, Tokyo (JP)
 • **Kumagai, Yoshiaki, c/o Sony Corp.**
Shinagawa-ku, Tokyo (JP)

(74) Representative: **Nicholls, Michael John**
J.A. KEMP & CO.
14, South Square
Gray's Inn
London WC1R 5LX (GB)

(54) **Charging method and charging system in interactive on-line service**

(57) A charging method in interactive on-line service where a server and terminals of users are mutually connected by means of a transmission medium, and a pre-determined service is provided from the server to the terminals via the transmission medium, and the fee for such service is collected from each user individually. The charging method comprises the steps of opening, in the server, an imaginary account for a child of the relevant user with a limited maximum amount, and withdrawing from the imaginary account the fee for the service provided to the user's child. The server can provide a predetermined service to the relevant user's child within a range of the limited maximum amount preset in the imaginary account. And when withdrawing the fee from the imaginary account, the server can restrict the service providable to the terminal. Thus, the parent enables his child to receive a desired on-line service, such as on-line shopping or video-on-demand, on the basis of the child's own judgment by setting an upper limit of a service utilisable by the child and still limiting the services providable for the child, hence realising promoted utilisation of the service by children.

FIG. 1



Description

The present invention relates to a charging method and a charging system adapted for use in interactive on-line service or mail-order sale such as on-line shopping or video-on-demand service.

In a conventional charging system where a central server and home television receivers are mutually connected by means of a cable and the server distributes video or data to the television receivers and charges each user individually for the service offered, it is generally customary that the charging process is executed by utilizing the user's credit card. In this case, the user acknowledges the charge by inputting the number of his credit card every time a service is received. Then the service provider company receives the fee from a credit company, and the credit company mails a bill with a detailed account to the actual user to thereby demand payment of the fee.

Thus, in the conventional charging system, any person other than the owner of a credit card is unable to receive the provision of a service. Since none of children under age has a credit card in general, there exists a problem that a child is incapable alone of receiving a service.

In an attempt to solve the above problem, there may be contrived an idea of giving general credit cards to children as well. However, this idea is considered unpractical under the present circumstances where a credit card is usable similarly to cash. Meanwhile in a situation where any child is permitted to use his parent's credit card, it may probably occur that some unnecessary charge is demanded of the parent. Consequently, whenever the child wants to receive a certain service, the parent is obliged to input the number of his credit card each time to approve provision of the service, and therefore it follows that the child is not allowed to receive the desired service unless the parent is always with the child, hence raising an issue that the child's independency is impeded.

On the other hand, as viewed from the service provider company, there exists a problem of probable failure in promoting utilization of the service by children.

It is an object of the present invention to enable a child to receive a desired on-line service on the basis of his own judgment by setting an upper limit of a service utilizable by children and still limiting the services providable for children, hence realizing promoted utilization of the service by children.

According to a first aspect of the present invention, there is provided a charging method in interactive on-line service where a server and terminals of users are mutually connected by means of a transmission medium, and a predetermined service is provided from the server to the terminals via the transmission medium, and the fee for such service is collected from each user individually. This charging method comprises the steps of opening, in the server, an imaginary account for a

child of the relevant user with a limited maximum amount, and withdrawing from the imaginary account the fee for the service provided to the user's child.

The server can provide a predetermined service to the relevant user's child within a range of the limited maximum amount preset in the imaginary account.

And when withdrawing the fee from the imaginary account, the server can restrict the service providable to the terminal.

According to a second aspect of the present invention, there is provided a charging system in interactive on-line service where a server and terminals of users are mutually connected by means of a transmission medium, and a predetermined service is provided from the server to the terminals via the transmission medium, and the fee for such service is collected from each user individually. This charging system comprises a setting means for setting an imaginary account of each user's child with a limited maximum amount, and a charging means for withdrawing from the imaginary account the fee for the service provided to the relevant user's child.

The server can provide a predetermined service to the relevant user's child within a range of the limited maximum amount preset in the imaginary account.

And when withdrawing the fee from the imaginary account, the server can restrict the service providable to the terminal.

In the charging method or the charging system of the present invention where an imaginary account of each user's child is opened in the server, the fee for the service provided to the relevant user's child is withdrawn from the imaginary account. Therefore the charging mode is changeable in accordance with the serviced user.

In one modified charging method or system of the present invention, a child is permitted to receive provision of a predetermined service within the limited maximum amount preset in the imaginary account, so that it is possible to prevent any excessive utilization of the service by the child.

And in another modified charging method or system of the present invention, the service to be provided is restricted when the fee is withdrawn from the imaginary account. Consequently, it becomes possible to restrict provision of any service that is not suited for participation of children.

The invention will be further described by way of non-limitative example, with reference to the accompanying drawings, in which:-

Fig. 1 shows a constitution of an exemplary embodiment representing the charging system of the present invention;

Fig. 2 is a flow chart showing a flow of equivalents in the constitution of Fig. 1;

Fig. 3 is a block diagram showing a detailed consti-

tution of the charging system in Fig. 1;

Fig. 4 is a flow chart showing an example of a charging process executed when a child is serviced with a game program;

Fig. 5 is a flow chart showing an example of a process executed on a closing day of a month in a service provider company; and

Fig. 6 is a flow chart showing an example of a process executed when a parent sets a balance in his child's imaginary account.

Fig. 1 shows a constitution of an exemplary embodiment which represents the charging system of the present invention, and Fig. 2 shows a flow of equivalents in the constitution thereof. First at step S1, a parent sets an imaginary account 2, which is to be used exclusively for charging his child, in a server/processor 1 (setting means, charging means) installed in a network service company. As will be described later with reference to Fig. 2, the server/processor 1 consists of a server (data base) 1a and a management computer 1b. Next at step S2, an adequate amount is preset in the charging imaginary account 2. Suppose now that an amount of \$50 is preset in the imaginary account 2.

Subsequently at step S3, the child operates a display device 4 such as a television receiver (hereinafter referred to as TV) and transmits a charge request for a desired service to be received, such as on-line shopping, video-on-demand or game, to the server and processor 1 via a set-top box 3. And simultaneously the child's identification data is also transmitted with the charge request. As a result, the fee for the desired service relative to the charge request is withdrawn from the child's imaginary account 2 preset in the server/processor 1.

In case the fee for the desired service relative to the child's charge request is \$20 for example, an amount of \$20 is withdrawn from the current balance \$50 in the imaginary account 2, whereby the balance is reduced to \$30. And at the end of each month, the service provider company submits a bill of the total amount (\$20 in this example) during the month to a credit company as a credit for the parent.

Thereafter at step S4, the credit company sends a detailed statement and a bill to the parent at the end of the month to thereby claim payment of the total fee (\$20 in this case) in that month. The present situation regarding utilization of the service can be checked by inquiring at the server/processor 1 without the necessity of waiting for arrival of the detailed statement from the credit company.

Fig. 3 is a block diagram showing the charging system of Fig. 1 in further detail. The set-top box 3 installed in each contractor home transmits or receives data to or from a service provider company via a cable or the

like and supplies the received predetermined data to the display device 4 such as a TV, or transmits to the service provider company a signal which corresponds to a predetermined command obtained by manipulation of an undermentioned remote controller 6.

The display device 4 is capable of displaying characters, patterns or pictures corresponding to predetermined data supplied from the set-top box 3. The remote controller 6 outputs a predetermined command to the set-top box 3 for selecting a desired service such as a game program to be transmitted from the service provider company, or for approving the charge.

In the server (data base) 1a of the service provider company, an imaginary account 2 of the contracted user's child is opened, and there are recorded data of the user inclusive of his credit card number and so forth specified at the time of contract. The management computer 1b transmits or receives predetermined data to or from the set-top box 3 installed in each contractor home for updating the balance of the imaginary account 2 opened in the server 1a, or for reading out some other data therefrom or writing predetermined data therein. Further the computer 1b is capable of transmitting or receiving other data such as charge data to or from the credit company via a telephone line, network or cable.

In a server (data base) 5a of the credit company, various data inclusive of users names, addresses and so forth are recorded. In response to an inquiry or a charge request based on a predetermined credit card number from the service provider company, a management computer 5b retrieves from the server 5a the user's name and address corresponding to the relevant credit card number, and sends a bill to the address. In case the credit company and the contractor home are mutually connected via a telephone line, network or cable, it is possible to execute an on-line demand for payment.

For example, when a desired service such as a movie is selected by manipulating the remote controller 6 in a contractor home and a purchase of the relevant service is commanded, this command is supplied from the set-top box 3 via the cable to the service provider company. At this time, the set-top box 3 supplies also an inherent user ID simultaneously to the service provider company. Then the service provider company recognizes a purchase demand of the desired service by a predetermined user who is identified by the user ID.

In the service provider company, the credit card number stored correspondingly to the user ID, which is supplied from the contractor home, correspondingly in advance at the time of contract is retrieved from the server 1a, and an inquiry and a charge request are executed to the credit company via a network or the like. And a desired service demanded for purchase, i.e., a movie in this example, is transmitted via a cable to the contractor home.

In the credit company, the fee for the service is demanded to the user in response to the charge request from the service provider company, while the fee for the

service is paid to the service provider company. As described above, a demand for payment of the fee to the user is executed by mailing a bill or by on-line transmission of the necessary data in case the credit company and the contractor home are mutually connected via a telephone line or the like.

In the case of an access by a child, it is regarded as a partial charge to his parent and is so processed in the credit company. More specifically, the imaginary account 2 of the child opened in the service provider company is set actually in his parent's account, so that the fee withdrawn from the child's imaginary account 2 is withdrawn actually from his parent's account.

For the purpose of enabling the server/processor 1 to recognize an access by a child, it is possible to devise the system in such a manner that a child needs to input a predetermined code number when making a charge request, or that a remote controller is prepared to be used exclusively by the child and, when a desired service is selected or a charge request is made by manipulating such a remote controller, it is regarded as an access by the child, and then predetermined identification data of the child is transmitted from the set-top box 3 to the server/processor 1. Consequently, the server/processor 1 is rendered capable of recognizing whether the accessing user is a child or not.

Fig. 4 is a flow chart for explaining the detailed operation of the charging system performed when a child is serviced with a game. First at step S11, a child selects a desired game program by manipulating the TV 4 or the remote controller 6, and approves a charge for the fee. Then the identification number of the selected game program and a charge request thereof are transmitted from the set-top box 3 via a predetermined cable to the server/processor 1.

Subsequently the operation proceeds to step S12, where the server/processor 1 recognizes a charge to the service requested by the child, i.e., a game program in this case, and makes a decision as to whether the fee for the requested service is less than or equal to the balance in the child's imaginary account 2, that is, whether the balance in the imaginary account 2 is sufficient or not. And if the result of such a decision signifies that the balance in the child's imaginary account 2 is insufficient for the fee of the game program, the operation proceeds to step S16 where a message indicating insufficiency of the balance is displayed on the screen of the TV 4. Then the operation returns to step S11, and the steps subsequent thereto are repeated.

Meanwhile, if the result of the above decision signifies that the balance in the child's account 2 is sufficient for the fee of the game program, the operation proceeds to step S13, where a check message of confirmation is further displayed on the screen of the TV 4 for example.

Thereafter the operation proceeds to step S14, where the server/processor 1 makes a decision as to whether the child has pushed the check button or not. More specifically, when the child has pushed the check

button provided for confirmation on the TV 4 or the remote controller 6 while watching the check message displayed on the screen of the TV 4 at step S13, then a corresponding signal is transmitted from the set-top box 3 via a cable or the like to the server/processor 1, so that the above decision is made in accordance with this signal.

If the result of the decision at step S14 signifies that the child has not pushed the check button, the operation returns to step S11, and the steps subsequent thereto are repeated. Meanwhile, if the result of the above decision signifies that the child has pushed the check button, the operation proceeds to step S15 where the fee for the service is withdrawn from the imaginary account 2, and the entire process is completed.

In the fee charging process to the child in this embodiment, it is not necessary for the child to input any credit card number differently from a usual case where a parent receives a service. Accordingly, there is no method of ascertaining if the user is really a regular contractor or not. However, the amount is not much since the account is an imaginary one for charging a child and, considering an advance approval of the parent who is an actual owner of the credit card, a simpler means has been selected for convenient utilization of any service by the child. If this selection raises a problem, a modification can be achieved with facility to further enhance the safety by necessitating the child to input a code number.

Fig. 5 is a flow chart for explaining the detailed operation performed in the service provider company. First at step S21, the situation of the service utilization and the balance are referred at the end of each month by the server/processor 1. Next at step S22, a decision is made by the server/processor 1 as to whether any service has been utilized or not.

If the result of the above decision is negative to signify no utilization of any service, the operation proceeds to step S25, where the entire balance in the imaginary account 2 is carried forward to the next month. Then the operation returns to step S21, and the steps subsequent thereto are repeated. Meanwhile, if the result of the decision at step S22 is affirmative to signify utilization of some service, the operation proceeds to step S23.

At step S23, a detailed statement of the service and the fee therefor are sent by the parent's credit card number to the credit company.

Then the operation proceeds to step S24, where the current balance in the imaginary account 2 is carried forward to the next month by the server/processor 1. After that, the operation returns to step S21, and the steps subsequent thereto are repeated.

In this case, only the amount used is billed to the card company, and the current balance is carried forward to the next month. It is usual that the balance tends to decrease as the child is provided with a service and pays the fee for the same. However, in an exemplary future case where the child receives a prize by giving a

correct answer in a quiz program or the like or by winning in an on-line game, the balance may increase due to transfer of the prize money to the imaginary account 2. Since it is impossible in this system to cash the balance in the imaginary account 2, the prize money is consumed also in a form of on-line shopping or the like, so that the service provider company can acquire a merit that the cash never flows out therefrom.

Fig. 6 is a flow chart for explaining the operation performed when a parent sets a balance in a child's imaginary account 2. Initially at step S31, the parent opens the child's imaginary account 2 in the server/processor 1 in the service provider company.

Then the operation proceeds to step S32, where the parent inputs an initial amount to be preset in the imaginary account 2 and further inputs his credit card number. Subsequently the operation proceeds to step S33, where the server/processor 1 inquires of the corresponding card company to make a collation as to whether the inputted credit card number is correct or not. And if the result of such a collation signifies that the inputted credit card number is not correct, the operation returns to step S31 and the steps subsequent thereto are repeated. Meanwhile, if the result of the above collation signifies that the inputted credit card number is correct, the operation proceeds to step S34.

Then at step S34, the balance of the amount inputted at step S32 is set in the imaginary account 2.

The processing flow for opening a new imaginary account 2 is similar to that for practically opening a bank account. This idea is derived from a prepaid card such as a telephone card, and the right of spending a fixed amount of money, which is spendable freely by a child, is purchased by his parent in advance using the parent's card number. The service provider company first inquires of the card company to ascertain if the card number is correct or not, then opens an imaginary account 2 and presets a predetermined amount therein.

Thus, in the service for providing information to each home via a cable, telephone line or network as described above, it becomes possible for a parent to easily manage his child's utilization of the information provision type service by opening a child-charging imaginary account in the server/processor 1 in the service provider company.

In an example where a parent opens his child's imaginary account 2 in the server/processor 1 and, in the same manner as to give a cash allowance to the child per month, presets in the account 2 the amount equivalent to the child's monthly spending money, the parent is rendered capable of managing the child's expense properly so that the child may not spend much amount carelessly for the service, hence realizing prevention of any trouble that results from purchase by the child in mail order or the like, e.g., any trouble relative to purchase of unnecessary commodities or payment therefor.

Meanwhile, as viewed from the service provider

company, participation of children to the service can be promoted to consequently increase expectable charging amounts, since the parent opens the above-described imaginary account 2 and presets a predetermined amount therein, whereby the child is permitted to receive a desired service on the basis of his individual judgment even in the absence of the parent in front of the set-top box 3.

Further, any service can be provided in a state where the independency of each child is respected, so that it becomes possible for the child at home to have quasi-experience of a business transaction extremely similar to a real one.

It is a matter of course that the parent can grasp the contents of the detailed account statement on the basis of the bill sent from the credit company for settlement, and is informed successively of the details by receiving from the service provider company the report of the imaginary account 2 in the server/processor 1, without the necessity of waiting for delivery of the detailed account statement.

And due to the construction where the balance of the imaginary account 2 is displayed on the display device 4 such as a television receiver, the child is enabled to be conscious of paying an equivalent for the provided service, and this brings about an advantage of raising the independency of the child.

Opening the child's imaginary account 2 is quite similar to the case of giving a family credit card to the child and limiting the amount of its credit. However, although a general credit card is usable as cash in nearly entire business transactions, this electronic imaginary account 2 is totally different therefrom in the point that it is available merely in the world of interactive on-line service such as on-line shopping or video-on-demand.

Therefore, in contrast with the case of a credit card where the situation of service utilization is transmitted with a detailed account statement sent per month, it is possible in the imaginary account to successively monitor the situation with a merit of on-line service. For example, a parent manipulates the TV 4 or the remote controller 6 to transmit a command, which is used for monitoring the situation of service utilization in the imaginary account 2, to the server/processor 1 via the set-top box 3. Then, in response to the command from the TV 4 or the remote controller 6, the server/processor 1 inquires into the situation of service utilization in the imaginary account 2 and transmits the result via the set-top box 3 to the TV 4. Consequently, the current situation of service utilization in the imaginary account 2 is displayed on the screen of the TV 4. Since the child's imaginary account is thus opened, an application program for enabling the parent to successively monitor the child's service utilization can be constructed with facility.

In case a child is permitted to carry an ordinary credit card with him, it is impossible for his parent to limit the place where the child may use the card. According to the present invention, however, the parent is capable of

preventing his child's access to any undesirable service by previously restricting some programs which are selectable by the use of the child's imaginary account 2. Consequently, provision of any service or information unsuitable for participation of children can be restricted.

Although in the above embodiment the server/processor 1 and the set-top box 3 are connected mutually via a cable, any other transmission medium such as a telephone line, an optical fiber cable or a network cable may also be employed for the connection.

Further, a monthly withdrawal from the imaginary account in the above embodiment is changeable to be weekly, daily or per service utilization.

Thus, according to the charging method or the charging system of the present invention in interactive on-line service, an imaginary account for a child of the relevant user is opened, and the fee for the service provided to the user's child is withdrawn from the imaginary account. Therefore the charging mode is changeable depending on the person to be provided with the service, whereby the child's utilization of the service is rendered allowable under supervision of the parent. Consequently it becomes possible to promote participation of children to the service to eventually enhance the added value of the service.

In one modification of the charging method or system, the child is permitted to receive provision of desired service within a range of the limited maximum amount preset in the imaginary account, so that any excessive service utilization by the child is rendered preventable to consequently eliminate any excessive charge.

And in another modification, providable service programs are limited at withdrawal of the fee from the imaginary account, so that it becomes possible to restrict provision of any service that is not suited for participation of children, and therefore the content of the service can be changed in accordance with the serviced person.

The invention may be embodied in other specific forms without departing from the spirit or essential characteristics thereof. The present embodiment is therefore to be considered in all respects as illustrative and not restrictive, the scope of the invention being indicated by the appended claims rather than by the foregoing description and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced therein.

Claims

1. A charging method in interactive on-line service where a server and terminals of users are mutually connected by means of a transmission medium, and a predetermined service is provided from said server to the terminals via the transmission medium, and the fee for such service is collected from each user individually, said charging method comprising the steps of:

opening, in said server, an imaginary account for a child of the relevant user with a limited maximum amount; and

withdrawing from said imaginary account the fee for the service provided to the user's child.

2. A charging system in interactive on-line service where a server and terminals of users are mutually connected by means of a transmission medium, and a predetermined service is provided from said server to the terminals via the transmission medium, and the fee for such service is collected from each user individually, said charging system comprising:

a setting means for setting an imaginary account of the relevant user's child with a limited maximum amount; and

a charging means for withdrawing from said imaginary account the fee for the service provided to the relevant user's child.

3. A charging method according to claim 1, or system according to claim 2, wherein said server provides the predetermined service to the relevant user's child within a range of the limited maximum amount preset in said imaginary account.

4. A charging method according to claim 1, or system according to claim 2, or any claim dependent therefrom, wherein, when withdrawing the fee from said imaginary account, said server restricts the service providable to the terminal.

5. The charging method according to claim 1, or system according to claim 2, or any claim dependent therefrom, wherein, in response to an input code number preset for the relevant user's child, said server identifies an access by the child and then withdraws from said imaginary account the fee for the service provided to the relevant user's child.

6. A charging method according to claim 1, or system according to claim 2, or any claim dependent therefrom, wherein, in response to input identification data indicative of the child and added in the terminal by a manipulation of the child's exclusive remote control means, said server identifies an access by the child and then withdraws from said imaginary account the fee for the service provided to the relevant user's child.

7. The charging method according to claim 1, or system according to claim 2, or any claim dependent therefrom, wherein, in response to an input request generated by a manipulation in the terminal for con-

firming the situation of service utilisation in said imaginary account, said server transmits to said terminal the data representing the situation of service utilisation in said imaginary account.

5

10

15

20

25

30

35

40

45

50

55

7

FIG. 1

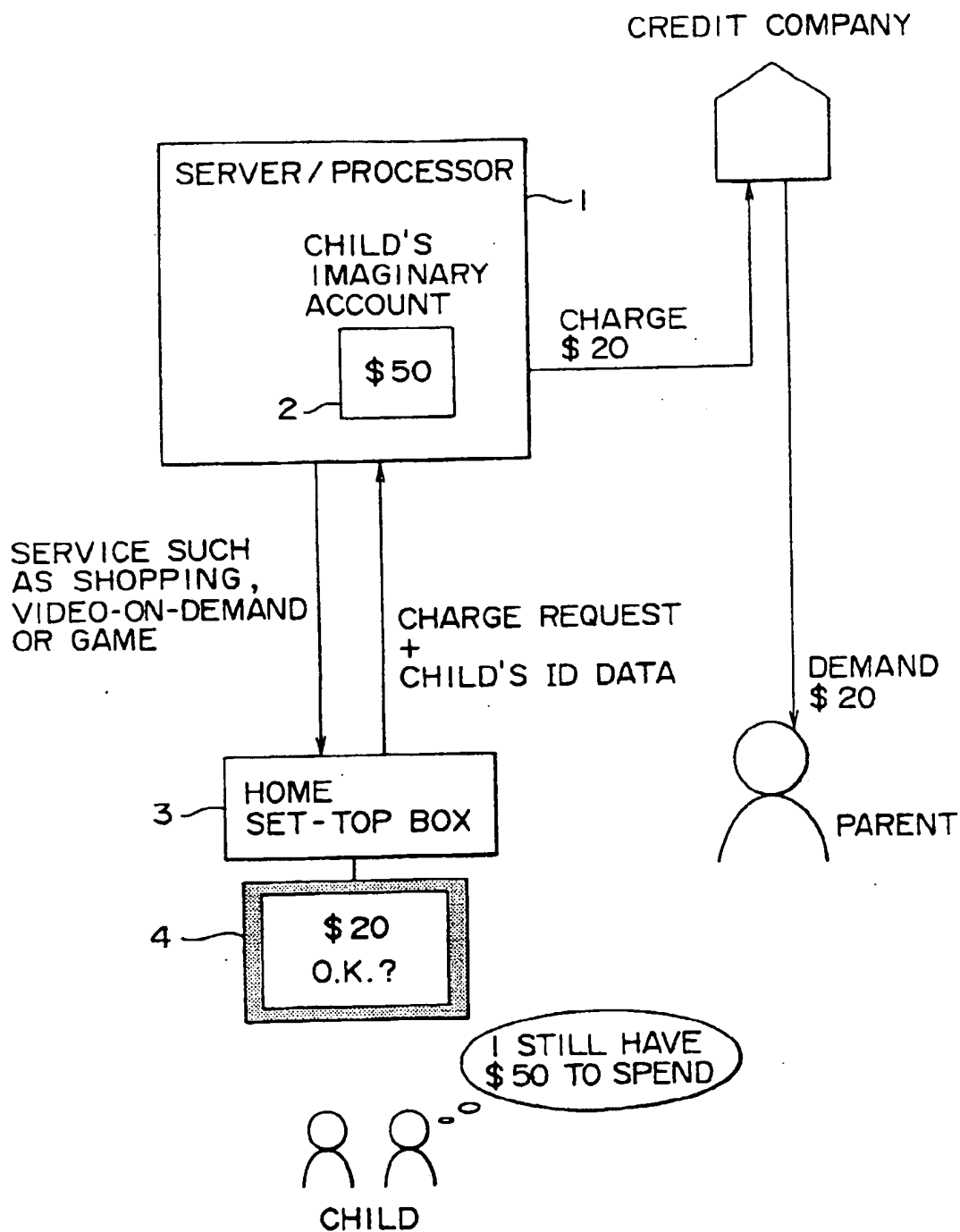


FIG. 2

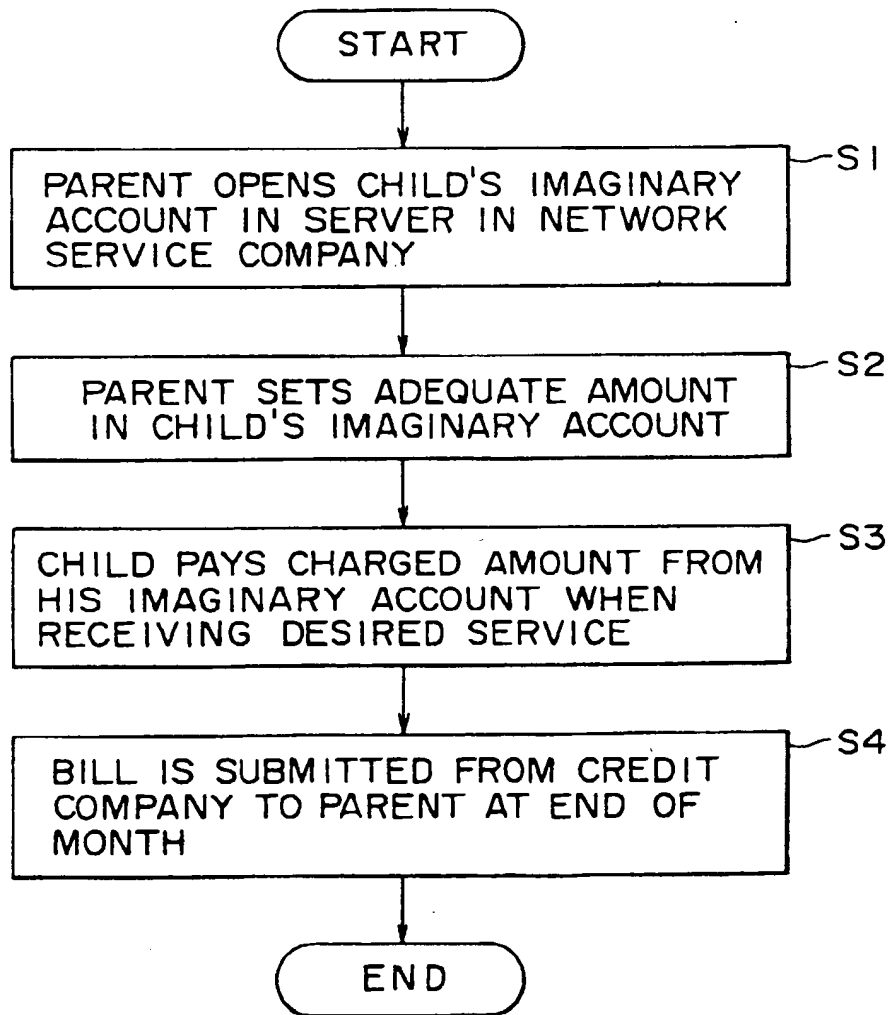


FIG. 3

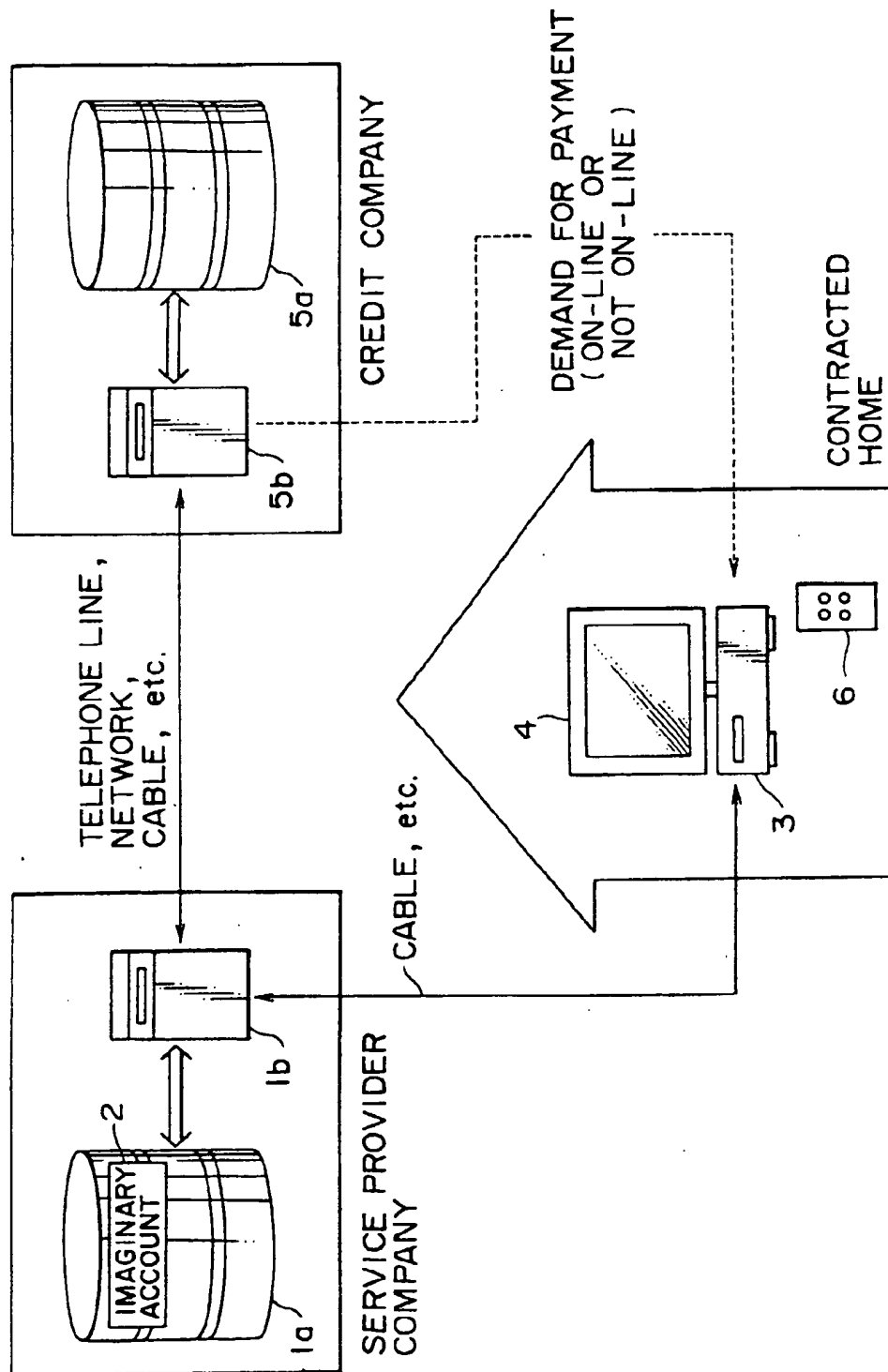


FIG. 4

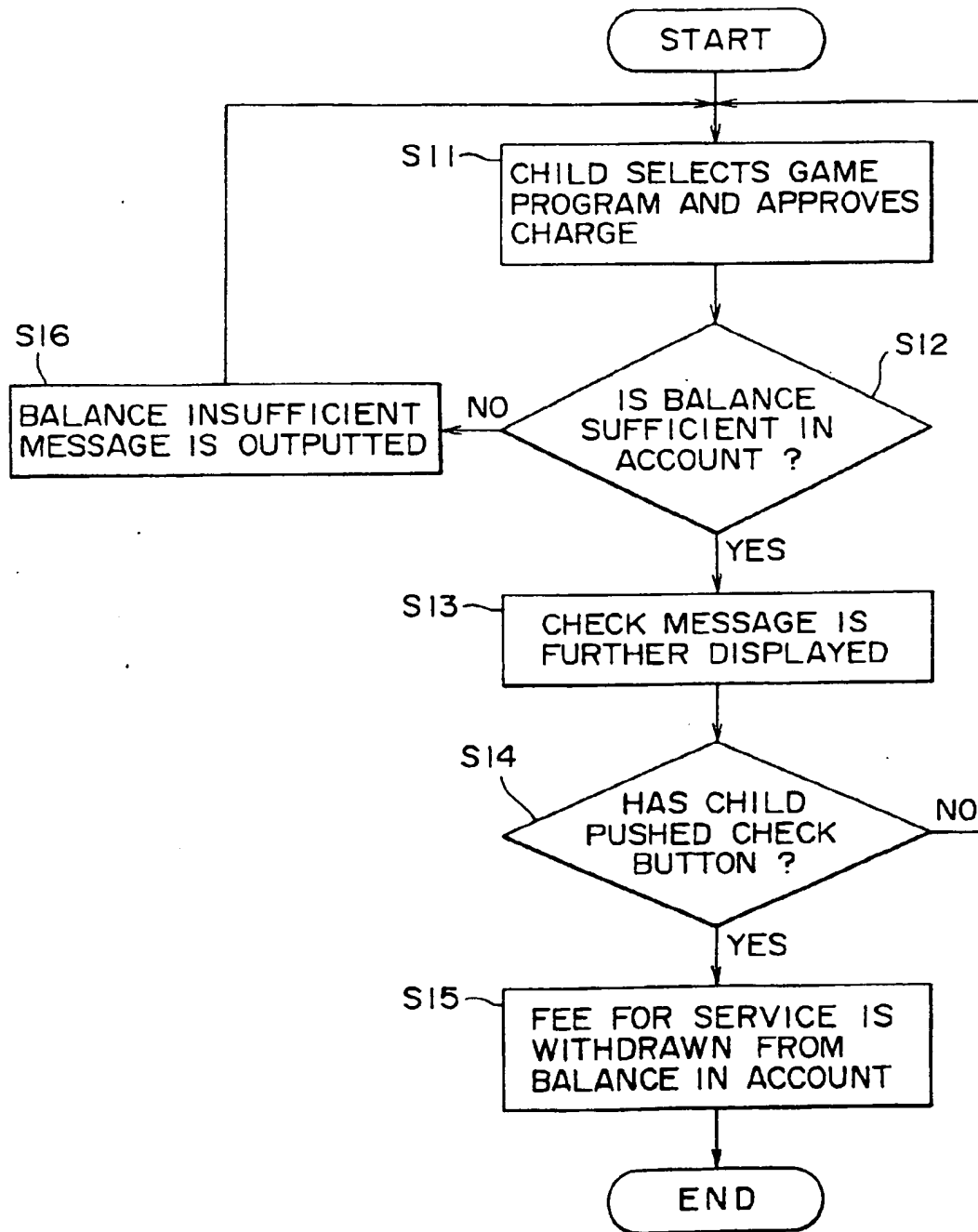


FIG. 5

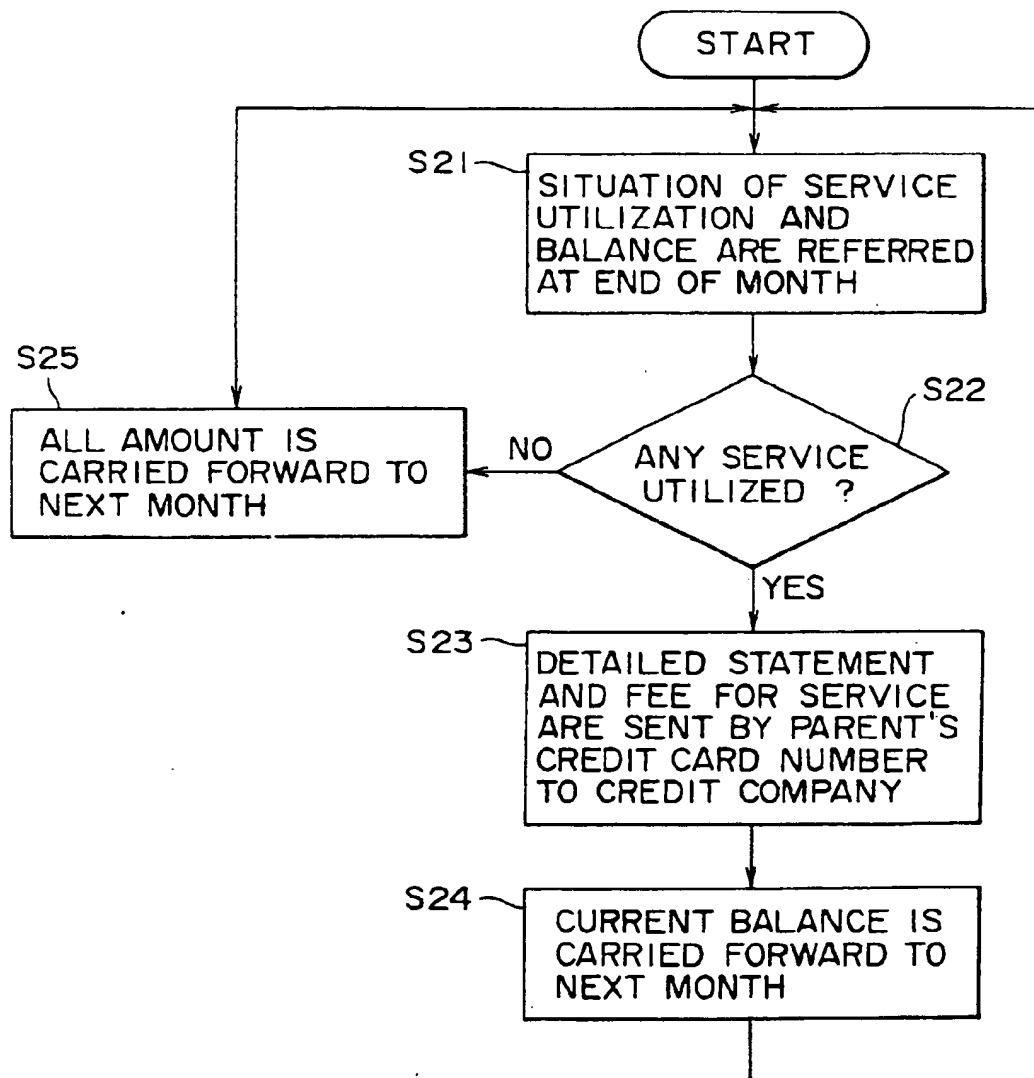
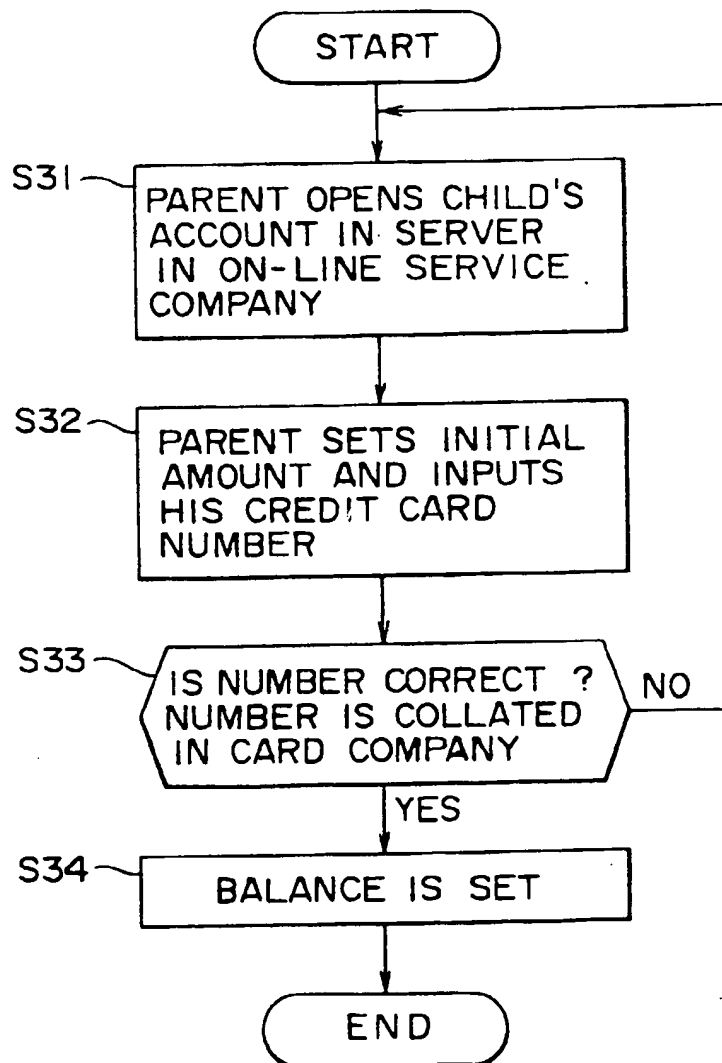
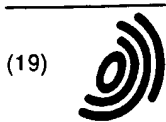


FIG. 6





Europäisches Patentamt
European Patent Office
Office européen des brevets



(11) **EP 0 332 707 B1**

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention
of the grant of the patent:
06.11.1996 Bulletin 1996/45

(21) Application number: **88907376.3**

(22) Date of filing: **22.08.1988**

(51) Int Cl.⁶: **G06F 17/30**

(86) International application number:
PCT/JP88/00832

(87) International publication number:
WO 89/02118 (09.03.1989 Gazette 1989/06)

(54) **DATA PROCESSING APPARATUS AND EDITING APPARATUS USING THE SAME**
DATENVERARBEITUNGSANORDNUNG UND DAMIT AUSGERÜSTETES AUSGABEGERÄT
SYSTEME DE TRAITEMENT DE DONNEES ET DISPOSITIF D'EDITION L'UTILISANT

(84) Designated Contracting States:
DE FR GB

(30) Priority: **28.08.1987 JP 216233/87**
28.08.1987 JP 216234/87
08.09.1987 JP 224276/87

(43) Date of publication of application:
20.09.1989 Bulletin 1989/38

(60) Divisional application: **94119237.9 94119239.5**

(73) Proprietor: **HONDA GIKEN KOGYO KABUSHIKI**
KAISHA
Minato-ku Tokyo 107 (JP)

(72) Inventors:
• **MIYOSHI, Akito**
Iruma-gun Saitama 350-02 (JP)
• **TERAI, Hiromitsu**
Sakado-shi Saitama 350-02 (JP)

(74) Representative: **Lehn, Werner, Dipl.-Ing. et al**
Hoffmann, Eitle & Partner,
Patentanwälte,
Postfach 81 04 20
81904 München (DE)

(56) References cited:
EP-A- 0 110 676 WO-A-86/05294
JP-A- 5 995 645 JP-A-59 214 966

- **I.E.E.E. SOFTWARE**, vol. 4, no. 2, March 1987, New York, USA, pp. 4-14; S. GIBBS et al.: "MUSE: A Multimedia Filing System"
- **I.E.E.E. COMPUTER SOCIETY, OFFICE AUTOMATION SYMPOSIUM**, 29 April 1987, NAT. B. OF STANDARDS, GAITHERSBURG, pages 180-189; D. WOELK et al.: "Multimedia Applications and Database Requirements"
- **COMPUTER & GRAPHICS**, vol. 10, no. 2, 1986, Great Britain, pages 119-131; U. FLASCHE et al.: "Decentralized Processing of Documents"
- **ENCYCLOPEDIA OF COMPUTER SCIENCE AND ENGINEERING** by Anthony Ralston, 1983, page 496 and 786-787
- **SMART TEXTVERARBEITUNG** by Innovative Software, 1987, chapter "Passwort-1" to "Passwort-2" and "Kapitel 1-3" to "Kapitel 1-4".

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

EP 0 332 707 B1

Description

Technical Field

This invention relates to a data processing system. More particularly this invention relates to a data processing system which is capable of easy and efficient retrieval of data.

This invention also relates to an editing apparatus aided by the data processing system. More particularly this invention relates to an editing apparatus possessing protective functions controlled by Security No. and enabling handling manuals such as for automobiles and electric appliances, books in general, magazines, and newspapers (hereinafter referred to collectively as "books") to be edited quickly and easily.

Background Art

In recent years, the practice of performing the work of editing a book by the use of a personal computer or a work station (editing device) has been disseminated. In this case, the work of editing a book is carried out by newly feeding complete data on documents, illustrations, etc. for each of the pages, with the whole information of an edited book memorized as separated in units of page. With the conventional editing apparatus, however, those documents, illustrations, etc. which appear in two or more places cannot be utilized in common but must be repeatedly fed in all over. The conventional editing apparatus, therefore, takes up much time and labor in the production of a book. The production of a book, accordingly, is very expensive.

An editing apparatus may be characterized by attaching an element data key possessing a format common to various kinds of element data such as documents, illustrations, and photographs which jointly form one page of the book, to each of the various kinds of element data thereby enabling the data concerning the elements to be handled as common data, causing data of an edited book to be decomposed into data concerning layout and other data concerning elements for storage, and composing data for one page with the data concerning the layout and the other data concerning desired elements.

When an editing apparatus is so constructed as to operate on the principle just described, since the data for one edited page are enabled to be again decomposed into data concerning the layout and other data concerning the elements and put to storage. By again composing data for one page with the data concerning the layout and those concerning desired elements, therefore, the element data common to the plural pages can be used in common without requiring these data to be introduced newly or repeatedly.

When the editing of a book is enabled to be easily carried out as described above, however, there arises the possibility that an operator not authorized performs

the work of editing a book or revises the contents of an already edited book. Further, since the various kinds of data forming a book are not only retained in the edited book but also memorized as element data, there ensues the possibility that the element data not yet disclosed to the public will be laid open to the public inspection without leave or even will be altered.

From Smart Textverarbeitung, SmartWare from Innovative Software, copyright 1984/1985/1986/1987, pages "passwort-1" to "passwort-2", page "Kapitel 1-3" to "Kapitel 1-4", a word processor is known in which a password can be allocated to a document. This password will always be requested by the program if an attempt is made to access a file through the commands of LOAD, PRINT, or READ.

From the Encyclopedia of Computer Science and Engineering, copyright 1983, editor Anthony Ralston, page 496 and pages 786-787, it is known that private files may be labelled with the name and the system identification number of a user and may even contain in their labels a password that must be matched against one provided by the user at the time of issuing the access request. Furthermore, a log-in procedure is described in which the system will request that the user enter an ID and a non-printing password.

Disclosure of Invention

This invention aims to solve the problem of alteration mentioned above.

This object is achieved by an editing apparatus having the features described in claim 1. An advantageous embodiment is given in the subclaim.

The security of a book and element data can be warranted by checking the security No. of an operator seeking access to the editing apparatus against the memorized security Nos. to determine whether this operator should be allowed to do so.

This description further provides processing means capable of efficient retrieval of data necessary for a new book from an already accumulated data base.

Generally, in the retrieval of data by a data processing apparatus, the wanted data are obtained by assigning titles (items) corresponding to various kinds of data registered in advance and, at the time of actual retrieval, introducing an item corresponding to a specific kind of data (hereinafter referred to as "data item") wanted into the data processing apparatus.

In a typical conventional data retrieval means, the items mentioned above are registered in advance in a hierarchical structure, the concepts of the items are sequentially defined by the use of input means for data items, and a particular data item which conforms to the item finally defined from a major concept is singled out to permit retrieval of the data required.

To be specific, when the retrieval of data is started, the items as major headlines appear on such a display as CRT. When the operator selects from among the dis-

played items one having the contents of a pertinent data item, the second items resulting from further division of the contents of the selected item appear as medial headlines on the display.

When the operator selects from among the newly displayed items one specific item having the contents of the data item, the third items resulting from further division of the contents of the selected item are exhibited as minor headlines.

Thereafter, the processing described above is repeated until the item defined sequentially from the major concept conforms to the data item and the item so conforming is selected and used for the retrieval of necessary data.

Another typical conventional data retrieval means permits direct introduction of a pertinent data item to be used for retrieval of necessary data. To be specific, indexes are allocated one each to all the species of data registered in the data processing apparatus. Thus, the data processing apparatus permits retrieval of necessary data by exhibiting all the items and corresponding indexes on the display and consequently enabling the operator to inject the index corresponding to the necessary data through such input means as a keyboard.

Now, the various conventional techniques will be described further in detail as applied to the retrieval of such information as population, area, etc. of the cities in Japan.

For example, the use of the conventional technique in finding the population of Shinjuku Ward, Tokyo will be explained.

(1) In the typical conventional data retrieval means which defines the major concept sequentially until conformity with a pertinent data item, when the data processing is started, the names of 47 prefectures (including Metropolis and Hokkaido) are exhibited as items of major headlines on the display. At this point the operator selects from the 47 names Tokyo Metropolis as an item embracing Shinjuku Ward being sought.

Subsequently, on the display, the names of 23 wards, cities, towns, and villages forming Tokyo Metropolis are exhibited as a medial headline. At this point, the operator selects from the displayed names Shinjuku Ward as an item being sought.

Then, the information on population, area, and number of households is exhibited as a minor headline on the display. At this point, the population in the items obtained by the sequential definition of the major concept and exhibited as a minor headline conforms to the population of Shinjuku Ward as the desired data item. By singling out the population from the minor headline mentioned above, therefore, the data on the population of Shinjuku Ward are obtained.

(2) In the other typical conventional data retrieval means which has indexes allocated one each to all

the species of data, when the data processing is executed, all the items and corresponding indexes are exhibited on the display and, at the same time, a message announcing that the index corresponding to the pertinent data item should be introduced is exhibited. At this point, the operator is able to get the data on the population of Shinjuku Ward by introducing the index corresponding to the population of Shinjuku Ward. Where the desired item of data happens to be the population of Shibuya Ward, the operator is allowed to take the data on the population of Shibuya Ward by introducing the index corresponding to the population of Shibuya Ward.

The conventional techniques described above, however, have the following problems.

(1) In the conventional data processing apparatus which sequentially define the major concept until conformity with the pertinent data item, the retrieval of data is invariably started with the display of the item of the most significant concept. Where the number of steps to be taken until the major concept sequentially defined conforms to the pertinent data item is large, the efficiency of the operation of the apparatus is diluted because the time and labor spent for these steps is increased. Moreover, since the items to be selected are exhibited on the display in all steps to be involved, an operator of small experience is readily accustomed to the operation of the apparatus but an operator of large experience finds only a small part to manifest the outcome of his learning. An increase in the accumulation of the operator's experience, therefore, cannot be expected to bring about any improvement in the efficiency of the operation of the apparatus.

(2) In the second typical conventional data processing apparatus mentioned above, since indexes are allocated one each to all the species of data, the retrieval of data compels the operator to introduce the pertinent index corresponding to the particular species of data. The search for the pertinent item in the items exhibited on the display is an irksome task. Where the data are so voluminous as to require the items and indexes to be exhibited as split into a plurality of sections the size of the screen of the display, the operator is compelled to scroll the screen of the display. Thus, there arises the possibility that the necessary data are not obtained very easily.

To overcome the problems mentioned above, a data processing apparatus is contemplated which is characterized by comprising data retrieval table memory means for memorizing titles and indexes in a hierarchical structure therein, n'th rank index input means for the introduction of the n'th rank index, data selection means for reading out of the data retrieval table memory means

the (n+1)'th rank data corresponding to the n'th rank index, data existence discrimination means for making a judgment as to the presence or absence of the (n+1)'th rank data in the data retrieval table memory means, and element data reading means for reading pertinent data out of the element data memory means when the absence of the (n+1)'th rank data is confirmed.

As the result, when the n'th rank index is introduced through the index input means, the (n+1)'th rank data are exhibited when the presence of the (n+1)'th rank data in the data retrieval table memory means is confirmed and, to the contrary, the data corresponding to the n'th rank index memorized in the element data memory means are displayed on the display means when the absence of the (n+1)'th rank data mentioned above is confirmed.

The introduction of the n'th rank index, therefore, brings about an operational effect of permitting the start of the retrieval of data from a desired rank.

Brief Description of Drawing

Fig. 1 is a functional block diagram for illustrating a basic arrangement of an embodiment ;

Fig. 2 is a schematic block diagram of the first embodiment;

Fig. 3 is a flowchart for representing a first editing method according to the first embodiment ;

Fig. 4 illustrates a format of the basic layout code;

Fig. 5 represents a data format of one data element when image data or document data is newly entered;

Fig. 6 illustrates a data format of one specification element among machine sort information;

Fig. 7 illustrates a data format of one data element of service data information among machine sort information;

Fig. 8 indicates a data format of one torque information element among machine sort information;

Fig. 9 illustrates a data format of one image information element which is converted during the editing operation;

Fig. 10 represents only element data in one element data of the service data information shown in Fig. 7;

Fig. 11 represents a file for editing a new book;

Fig. 12 illustrates how the converted layout code is produced from the basic layout code;

Fig. 13 is an illustration of one example of a display screen of the basic layout;

Fig. 14 is an illustration of one example of a display screen during the editing operation;

Fig. 15 is an illustration for illustrating how a temporary key of one element data of newly input image information is converted into an element data key;

Fig. 16 is an illustration for illustrating how a temporary key of one element data of newly entered document information is converted into an element data key;

Fig. 17 is an illustration for illustrating a data format of one document information element which is converted during the editing operation;

Fig. 18 is an illustration for showing a data format of one page which is edited by the editing machine;

Fig. 19 is an illustration for representing a data format of one page which is converted by the host computer so as to be registered into the layout D/B 11;

Fig. 20 is a flowchart for representing a second editing method according to the first embodiment ;

Fig. 21 is an illustration for explaining how the data rewriting at the step S104 of Fig. 20 is performed;

Fig. 22 is a schematic diagram for showing a portion of a table for retrieving document data set in the document D/B 12;

Fig. 23 is a schematic diagram for representing a hierarchical structure of a document retrieval table;

Fig. 24 is a flowchart for illustrating a retrieval method according to an embodiment ;

Fig. 25 is a graphic representation for showing relationships of a conversion section, an image section and service data on a CRT;

Fig. 26 is a functional block diagram of the first embodiment;

Fig. 27 is a flow chart illustrating a second procedure of editing.

Fig. 28 is a functional block diagram of an embodiment of the present invention.

Fig. 29 is a flow chart illustrating the first procedure of editing according to the embodiment of Fig. 28.

Fig. 30 is a flow chart illustrating the second procedure of editing according to the embodiment of Fig. 28.

Fig. 31 is a model diagram illustrating the condition of data controlled within the data control No. memory means.

Fig. 32, 33, and 34 are functional diagrams of another embodiment of this invention, respectively.

Fig. 35 is a flow chart illustrating a procedure of data retrieval.

Best Mode for Carrying Out the Invention

Now, the present invention will be described in detail below with reference to the accompanying drawings. Preparatory to the detailed description, the general construction and operation of the editing apparatus embodying the present invention will be described.

In figure 2, a plurality of editing apparatus (work stations) 3-1 to 3-N for inputting/outputting document, photographs, illustrations and the like are connected in an on-line mode to a system controller 2 for controlling these editing apparatus. These editing apparatus receive various information such as document, photographs and illustrations by way of the vector conversion.

A CRT (cathode-ray tube) 6 is connected to the editing apparatus 3-1, a printer 7 is connected to the editing

apparatus 3-2, a scanner 8 is connected to the editing apparatus 3-3, and a personal computer 9 and another CRT 10 are connected to the editing apparatus 3-N. Furthermore, keyboards 5-1 to 5-N are connected to the respective editing apparatus 3-1 to 3-N.

The system controller (a medium-scale relay computer) 2 is furthermore connected to a host computer 1.

A book data base (D/B) 21, an element data base 22 and a layout basic data base 23 are connected to the system controller 2.

The system controller 2 controls these editing apparatus 3-1 to 3-N, and transfers/receives the information in a page unit or an element (document, photograph, illustration) unit constituting one page to and from the respective editing apparatus.

The host computer 1 is a large-scale general purpose computer such as the IBM's computer, IBM 3090 or equivalent. The system controller 2, personal computers 4-1 to 4-N, layout D/B 11, document D/B 12, image D/B 13 and machine sort information D/B 14 are respectively connected to the host computer 1.

It should be noted that the personal computers 4-1 to 4-N may be exclusively used for the editing apparatus, or for the general purpose other than the editing work, which is similar to personal computers 31-1 to 31-N (will be discussed later).

In case that the editing apparatus 3-1 to 3-N possess a function capable of inputting a language (referred to as "a specific language") other than the language input by the personal computers 4-1 to 4-N, a software for inputting and processing the specific language by the personal computers 4-1 to 4-N is provided with, for instance, the system controller 2.

In addition, another software is provided with the system controller 2, by which the document input by the personal computers 4-1 to 4-N is vector-converted, as is similar to the document input by the respective editing apparatus 3-1 to 3-N.

The personal computers 31-1 to 31-N which are employed to carry out the work other than the editing work effected by the editing apparatus, are connected to the host computer 1. In other words, the machines or parts surrounded by a dot line shown in Fig. 2 do not constitute the editing apparatus of this invention.

The host computer 1 constitutes the data of one book from the information which have been stored in the data bases 11 to 14, and transfers this data to the system controller 2. The system controller 2 transfers the information of one page which has been edited in the editing apparatus to the host computer 1.

An operation of the data processing apparatus according to a preferred embodiment of the invention will now be described.

Basically, the editing apparatus employing the constructions as illustrated in Fig. 2 can produce, or edit a new book by way of two editing methods as follows.

(A) When anew book is newly produced at all, all of documents or sentences, illustrations, and photographs etc.

to be written on each page of the new book must be newly input.

This will be referred as "a first editing method".

(B) When a new book is produced by utilizing other books which have previously been edited or produced in the editing apparatus, the necessary portions of other books are utilized for the new book and only the minimum portions required to produce the new book are newly input. This will be referred to as "a second editing method".

Referring now to Figs. 3 to 19, a description is made that a service manual (simply referred to as "a book" hereinafter) for a bike (auto-bicycle) is produced in accordance with the above-defined first editing method.

Fig. 3 is a flowchart for indicating a first editing method.

In Fig. 3, first as a step S1, a basic layout code is input into the editing apparatus to which CRT is connected (for instance, the editing apparatus 3-1) by the keyboard 5-1. The basic layout code is to set a size of a book to be newly produced, namely, a size, the number of character, a column number and a column space in one page of the new book.

Fig. 4 illustrates one example of the format of the basic layout code.

The basic layout code is constructed of a layout code having, for example, a 6-byte code length, layout data, and another data representing the data length of the layout data.

It should be noted that the data representing a data length of certain data is simply referred to as a "LEN" in the following description.

The layout code is constructed of the data for representing which product classification a book to be produced belongs to; the data for indicating which location (country) the new book is designated; the data representative of the sort of the new book, e.g., service manual, or shop manual; and the data representative of the layout of each page of the new book, e.g., a single frame or a double frame.

The layout data is constructed of the data representing a size of each page of the book, i.e., A6 vertical, A6 horizontal and so on; and the data indicative of a line number, a line space, the number of character, a point, style, presence of a ruled line.

The data concerning a concrete layout of a page on a screen of CRT is set in the layout data, and the layout code is merely an identification code of the layout data. The layout code is converted into a document identification number and a page number which correspond to the identification data of the new book, as will be described later in connection with the steps S29 and S30.

When entry of the layout data is accomplished, LEN of the layout data is added in front of the layout data.

As previously described in detail, the basic layout code is produced from the layout code, LEN of layout data, and layout data.

Referring back to Fig. 3, when the basic layout code

is generated, as illustrated in a step S2, the editing apparatus 3-1 transfers the basic layout code to the system controller 2 which registers the basic layout code in the layout basic D/B 23.

Upon completion of registering the basic layout code, in the next steps S3 to S24, entry of such information of illustrations, photographs, document and the like which are to be arranged in each page of the book is carried out.

It should be noted that both illustrations and photographs are referred to as an image in the following description.

In the step S3, a judgement is made whether or not the information to be input corresponds to the image. When the image is entered, as indicated in the step S4, a temporary key having, for instance, a 6-byte length is entered in the editing apparatus 3-3 by the keyboard 5-3. The temporary key corresponds to an identification code of image data to be entered which is determined by an operator himself. "D" indicating that this temporary key relates to the image data is entered into the head byte of the temporary key.

In the next step S5, the image is read out by the scanner 8.

In a step S23, the temporary key, LEN of the readout image data, and the image data are registered into the element D/B 22 as one element data.

If a judgement is made in the previous step S3 that no image is entered, another judgement is made in a step S6 whether or not document or sentence is entered. When the document (sentence) is entered, as indicated in a step S7, the temporary key is input in the editing apparatus 3-1 by the keyboard 5-1 or in any of the personal computer 4-1 to 4-N. As previously explained in the step S4, the temporary key is constructed of, for instance, a 6-byte length, and corresponds to an identification code of the document to be input which is determined by the operator himself. In this case, "T" indicating that the temporary key relates to the document key is input into the head byte of the temporary key.

In the subsequent step S8, the document or sentence is entered in the keyboard 5-1 or any one of personal computers 4-1 to 4-N.

Then, in the step S23, the temporary key entered in the previous step S7, LEN of the document entered in the step S8, and this document are registered as one element data in the element D/B 22.

In Fig. 5, there is shown one element data registered in the step S23.

In the step S6, if a judgement is made that no document is entered, it is decided that said input is related to the machine sort information.

The machine sort information includes the data on the repairing data, tools to be used and the like for a bike. In accordance with the preferred embodiment of this embodiment, the machine sort information includes the specification, service data and torque, which will be described in detail hereinafter.

(A) Specification is, for instance, the repairing data on the bike, and the data representative of the repairing items and details which are entered into a list indicating the data of the tool to be used, and is constructed of alphanumerical data and KANJI characters. In other words, this specification represents a type of an engine oil, a name of a tool to be used, and repairing items such as an inner diameter of a cylinder, and an outer diameter of a piston diameter, and the like.

(B) Service data is numerical data representative of, for instance, maintenance data which is entered into a list of the maintenance data, and constructed of three different data, i.e., a maximum value, a minimum value, and a limit value of usage (or a center value).

That is, as the maintenance data relating to the inner diameter of the cylinder, there are the typical maximum value and minimum value, and the usable maximum value. Also, as the maintenance data concerning the outer diameter of the piston, there are the typical maximum value and minimum value, and the usable minimum value. As to the data of a capacity of an engine oil tank, a capacity of a coolant and the like which has no discrimination in the maximum and minimum values, the same values as these maximum and minimum values are input, by which these data are registered.

(C) Torque is numerical data indicative of fasten torque of fastening screws for the bike (maximum value, minimum value, and center value), a diameter of a screw, the number of the required screws. That is to say, the torque information is constructed of five pieces of numerical data containing the maximum value, minimum value, center value (or the limit value for usage) and other two informations.

When the judgement is made that the input data corresponds to the machine sort information, the machine sort code is entered which represents that the machine sort information to be input is related to which sort of the bike, as indicated in the step S9, by utilizing any one of the personal computers 4-1 to 4-N (e.g., personal computer 4-1) connected to the host computer 1. This machine sort code is constructed of, for instance, 8 bytes, and "K" for representing that the machine sort code is related to the machine sort information is input into the head byte of the machine sort code.

In a step S10, a judgement is made whether or not the machine sort information to be input is the specification.

If a judgement is affirmative in the step S10, then, an information identification code having, for instant, a 4-byte length is input in a step S11. At this time, "A" for indicating that the information identification code is related to the specification is input at the head byte of the identification code.

In a next step S12, an item having, for example, a 3-byte length is entered.

In a subsequent step S13, an image classification or sort having, for instance, a 2-byte length is input. Data representing a language sort of the specification information (for example, the language is Japanese, or English) is entered into this image classification.

In a step S21, specification information is entered by the personal computer 4-1.

In a subsequent step S22, the machine sort code, information identification code, item and image classification which have been input by the previous steps S9, S11, S12, and S13, the data status (which is automatically set in the host computer 1) for indicating the history of the specification information entered in the step S21, and the specification information are registered as one element data in the machine sort D/B 14 by means of the host computer 1.

A data format of the one element data is illustrated in Fig. 6. As is shown in Fig. 6, the data consisting of the machine sort code, information identification code, item, image classification, and data status will be referred to as an element data key in the following description.

If no specification is input in the previous step S10, another judgement is made whether or not the service data is entered in a step S14.

If the service data is entered, the information identification code is input in a step S15, which is similarly done in the preceding step S11. In this case, "B" is input into the head byte of this code.

In the next step S16, the item is input, which is similarly executed in the previous step S12.

In the subsequent step S17, the image section is entered. The function of the image classification information entered at entry of the service data will be discussed later with reference to a step S42.

In the step 21, the service data information is input by means of the personal computer 4-1.

The service data information is constructed of three different data, i.e., the maximum value, minimum value and a limit value of usage (or a central value). When these three data are input in a specific unit in the editing apparatus according to the preferred embodiment, these data are stored and also converted into other units to be stored thereafter.

Fig. 7 illustrates a format of 1-element data (element data key and service data) which is registered in the machine sort information D/B 14 in the step S22.

As illustrated in Fig. 7, the service data is so constructed as to be set in a first data section through a third data section. In the step S21, when the three data are input in the preselected specific unit (for example "mm"), these data are set in the first data section. Although it is not shown in Figs. 3 and 7, just before entry of the service data information, a conversion classification (for example, 2-byte length data) for indicating what other unit three data set in the first data section should be converted into, with the result that the three data entered in the millimeter unit into the first data section are converted

into other units (e.g., an inch unit) than millimeter unit, which are designated by the conversion classification, and thereafter stored in the second data section.

Then the three data in millimeter unit stored in the first data section are converted into still other unit designated by the conversion classification and the resultant converted data are transferred to the third data section.

Referring to Fig. 10, the above-mentioned conditions will now be described more in detail. Fig. 10 illustrates only the service data (element data) among the data shown in Fig. 7.

When the information on the inner diameter of the cylinder of the bike is entered as the service data, the minimum value of 56.003, the maximum value of 56.018, and the limit value of usage of 56.08 (illustrated in Fig. 10 as the service limit) are input in the millimeter unit. These numerical data are set in the first data section.

If the conversion classification set before entry of the service data is designated to convert the millimeter unit into the inch unit, each of the data set in the first data section is converted into 2.2048; 2.2054; and 2.207 inches, respectively and the converted data are set in the second data section.

Also in the third data section, each of the data set in the first data section is converted into still other unit in accordance with the designation of the conversion classification.

In other words, the function of the conversion classification is to automatically convert the service data into a certain unit other than the originally input unit when the service data is entered in the certain specific unit, and thereafter to set these converted data into the second and third data sections.

Next, in the step S22 as shown in Fig. 7, the machine sort code, information identification code, item and image classification which have been entered in the previous steps S9, S15, S16 and S17, the data status (which is automatically fixed in the host computer 1) for representing a history of the service data information entered in the step S21, and the service data information consisting of the first to third data sections are registered as the 1-element data in the machine sort information D/B 14 under the control of the host computer 1.

Then, if no service data is entered the judgement of the step S14 is negative and shows that the torque information is entered.

When the entry of the torque information is judged, the information identification code is entered in the step S18, which is similarly executed in the preceding steps S11 and S15. At this time, "C" is input in the head byte of the information identification code.

In the subsequent step S19, the item is input, which is similarly performed in the steps S12 and S16.

In the step S20, the image classification is entered. The function of the image classification which is entered during entry of the torque information is similar to a func-

tion of an image classification which will be described later relating to a step S42.

In the next step S21, the torque information is input into the personal computer 4-1.

The torque information is constructed of five pieces of data containing the maximum value, minimum value, and center value (or a limit value of usage), and also two pieces of data representative of other information. In the editing apparatus, when these five pieces of data are input in the specific unit similar to the above-mentioned entry method of the service data, these data are stored, and at least three pieces of data other than two pieces of data to represent the other information are firstly converted into other units and secondly stored.

Fig. 8 illustrates a format of 1-element data (element data key and torque) which is registered in the machine sort information D/B 14 in the step S22.

As illustrated in Fig. 8, the torque data is so arranged as to be set in the first to third data sections. In the step S21, when the torque data is entered in some particular unit (for example, "kg"), this torque data is set in the first data section. Although not shown in Figs. 3 and 8, the conversion classification is input, just before entry of the torque information, so as to instruct that five pieces of data set in the first data section should be converted into a certain different unit. As a result, at least three pieces of data input in the "kg" unit into the first data section (i.e., three pieces of data other than two pieces of data representative of the other information among five pieces of data to indicate the torque information) are converted into a different unit (e.g., "lb" unit) designated by the conversion classification except the "kg" unit. Then, the converted data is set in the second data section.

In addition, at least three pieces of data entered into the first data section in the "kg" unit are converted into still other unit which are designated by the conversion classification, and thereafter the converted data are set in the third data section.

In other words, the conversion classification is to automatically convert the torque data into the unit other than the above-described unit when the torque data is input in the certain specific unit, and also to set these converted data into the second and third data sections, in a manner similar to the case of the service data.

Since the data concerning the diameter and the number are not required to be converted by the unit, no unit conversion is needed for these two pieces of data. It may be possible, of course, to perform the similar unit conversion on these two data as same as in other three data. In this case, however, it is necessary to prohibit from using the converted two data.

In the 1-element data as illustrated in Figs. 6 to 8, a byte length of the respective element data key is identical to each other.

In the step S22, as shown in Fig. 8, the machine sort code, information code, item and image classification which have been input in the steps S9, S18, S19

and S20, the data status (which is automatically set in the host computer 1) representative of the history of the torque data which has been input in the step S21, and the torque data information consisting of the first to third data sections are registered as 1-element data by the host computer 1 into the machine sort information D/B 14.

In the steps S22 or S23, when entry of one-image, one-document, or one piece of the machine sort information (these are referred to as "element data" hereinafter) is accomplished, a judgement is made whether or not the entry of the element data is continued in the next step S24. If the entry of the element data is continued, the control process is returned to the previous step S3. If the entry of the element data is not continued, the editing work of the book is commenced from a step S25. The editing work is performed by one page unit.

In the step S25, a new book producing file is read out from the host computer 1 by any one of the personal computers 4-1 to 4-N connected to the host computer 1.

Fig. 11 illustrates an arrangement of the new book producing file. As shown in Fig. 11, the new book producing file is constructed of material supervision or ID No. information, staff-in-charge information, and starting page information.

The material supervision No. is a title of a book to be newly produced or the supervising number. The staff-in-charge information is the data representative of a staff who is qualified to produce the new book. The starting page information indicates what page the book to be produced is commenced from. In other words, the page information indicates the first page of the new book for the starting page. For instance, the second page corresponds to the starting page of a book distributed in Japan, whereas the eleventh page corresponds to the starting page of a book distributed in U.S.A.

The staff-in-charge information is able to be applied to the security supervision of a newly edited book, for example. That is to say, the information relating to a staff who is permitted to produce a new book is previously registered in the host computer 1, and the new book is edited by the staff only when he has been registered in the new book producing file in the host computer 1. Under this security supervision, it can prevent that the book which is not allowed to be edited is mistakenly edited or the previously edited book is unnecessarily and mistakenly revised by the editing apparatus.

In this case, the staff-in-charge information may be preferably encrypted or stored in a magnetic card.

When the new book producing file is read out, predetermined data is input into this file at a step S26. When the predetermined data is entered, as shown in a step S27, the new book producing file is transferred from the host computer 1 to the system controller 2 and then registered in the book D/B 21.

In the next step S28, the layout code (see Fig. 4) is entered by the keyboard 5-1, so that the basic layout code previously registered in the layout basic D/B 23 is

called up to CRT 6.

When the basic layout code is called up in the editing apparatus, both the document supervision No. and the page number to be edited are input. The document supervision number is a title or the supervision or ID number of the new book, which is similar to the document supervision No. of the new book producing file shown in Fig. 11.

After the document supervision No. and the page number to be edited are input in a step S29, the editing apparatus 3-1 converts the layout code among the basic layout code read in the step S28 into the document supervision No. and the page number which have been input in the step S29, as illustrated in a step 530 and Fig. 12.

Upon completion of the code conversion, the editing apparatus 3-1 stores the document supervision No., the number of page, LEN and layout data in the editing apparatus 3-1 in a step S31. These document supervision No., the number of page, LEN and layout data will be referred as "a converted layout code" in the following description.

In a step S32, the basic layout is displayed on CRT 6 by employing the converted layout code. As illustrated in Fig. 13, the basic layout is basically constructed of a contour 101 of a page to be edited and a document entry region 102. The basic layout shown in Fig. 13 is a double frame.

In a subsequent step S33, a judgement is made whether or not the image is displayed on CRT 6.

To display the image, the temporary key (see Fig. 5) is entered by the keyboard 5-1 in a step S34 so as to call up one element data, i.e., temporary key, LEN of the image data to be called up, and the image data from the element D/B 22, and display the image data called up on CRT 6.

In a step S35, by a mouse (not shown) connected to the keyboard 5-1, the image displayed on CRT 6 is moved to a desirable position on the basic layout.

Fig. 14 illustrates an image of the screen of CRT 6 at that moment, which is similar to the illustration of Fig. 13. As shown in Fig. 14, when the image is moved to the region denoted by "P", the upper left coordinates and lower right coordinates of the image are (X1, Y1) and (X2, Y2), respectively.

When the image is transferred in this way, the upper left coordinates (X1, Y1) and lower right coordinates (X2, Y2) on the basic layout are entered into the editing apparatus 3-1.

It should be noted that while the image is moved, the size of the image may be enlarged or reduced in accordance with the capability of editing apparatus. Only the image can be moved over the document input region 102.

In a step S36, as illustrated in Fig. 15, the temporary key among the 1-element data consisting of the temporary key, LEN of image data to be called up, and image data is converted into the element data key.

As illustrated in Fig. 9, the element data key is constructed of the machine sort code, information identification code, item, image classification, and also data status (which is automatically input in the host computer 1) representative of a history of image data, which is similar to the machine sort information including the specification information, service data information, and torque information.

The conversion operation of the temporary key into the element data key in the step S36, is practically performed when an editor has entered the element data key excepting for the data status. "D" is input in the head byte of the information identification code. A byte length of the element data key shown in Fig. 9 is identical to the respective byte lengths shown in Figs. 6 to 8.

In a step S37, both the 1-element data consisting of the element data key, LEN of the image data corresponding to the element data key and the image data, and the coordinates (X1, Y1), (X2, Y2) are registered in the editing apparatus 3-1.

In a step S45, another judgement is made whether or not the edition of one page is accomplished. If not yet accomplished, then the control process is returned to the step S33.

If it is judged in the step S33, that the image is not displayed on CRT 6, another judgement is made whether or not the document or sentence is displayed in a step S38.

When the document or sentence is displayed, in a step S39, the temporary key (see Fig. 5) is entered by the keyboard 5-1, and the one element data consisting of the temporary key, LEN of the document data to be called up, and the document data themselves is called up from the element D/B 22, and the document data are displayed on CRT 6.

By employing the mouse (not shown) connected to the keyboard 5-1, the document displayed on CRT 6 is moved to the desired position on the basic layout in a step S40.

When the document is moved to the position indicated by the symbol "Q" on the basic layout shown in Fig. 14, the upper left coordinates and the lower right coordinates of the document are (X3, Y3) and (X4, Y4), respectively.

After the document has been moved, both the upper left coordinates (X3, Y3) and the lower right coordinates (X4, Y4) of the document on the basic layout are input into the editing apparatus 3-1 (S40).

In a step S41, as illustrated in Fig. 16, the temporary key in the 1-element data consisting of the temporary key, LEN of the called up document, and document data is converted into the element key data. Formats of these element data key, LEN of the called up document data and document data are illustrated in Fig. 17.

As shown in Fig. 17, the element data key corresponding to the document data is constructed of the sentence supervision No. having a length of, for instance, 8 bytes; the detailed item code having a length

of, for example, 6 bytes; the language information having a length of, for example, 2 bytes; and the data status (which is automatically added by the host computer 1) indicative of a history of the sentence or document data.

The sentence supervision No. is the data to specify each document, and in the head byte of which "B" is input. The language information indicates what language the document data has been produced in, for example, Japanese or English. The detailed item code is a classification code of the document or sentence.

It should be noted that after the data status of the 1-element data shown in Fig. 17, appropriate number of blanks are added to make the byte length of the element data key equal to that of the respective element data keys as shown in Figs. 6 to 9. In other words, each byte length of the element data keys corresponding to the document data, image data and machine sort information data is set to be identical with each other.

With employing the detailed item code, the editing apparatus can retrieve the document or sentence. This document retrieval operation will be discussed later in a step S114.

In a step S37, the element data key, LEN of the document data corresponding to the element data key, and document data are registered as the 1-element data together with the coordinates (X3, Y3), (X4, Y4) into the editing apparatus 3-1.

In a step S38, if a judgement is made that no document is displayed on CRT 6, it can be recognized that the machine sort information is to be displayed.

In a step S42, the element data key of the machine sort information to be called up (see Figs. 6 through 8) is input by the keyboard 5-1, whereby the machine sort information having 1-element data, i.e., element data key, specification information, service data information or torque information is called up from the machine sort information D/B 14 to the editing apparatus 3-1, and then the specification information, service data information or torque information is displayed on CRT 6.

In another embodiment, the element data key is entered by the keyboard 5-1, and the machine sort information is directly called up via the host computer 1 and system controller 2 from the machine sort information D/B 14 to the editing apparatus 3-1. However, alternatively, employing any one of the personal computers 4-1 to 4-N, the machine sort information may be called up from the machine sort information D/B 14 to be temporarily registered into the element D/B 22. Thereafter, the element data key may be input by the keyboard 5-1, and the machine sort information may be called up via the system controller 2 from the element D/B 22 to the editing apparatus 3-1.

It should be noted that display conditions or modes of the service data information and torque information among the machine sort information data are determined in accordance with the image classification information in the element data key and the conversion classification (not shown) described with reference to the

step S21.

As to the service data, for instance, as illustrated in Fig. 10, when the data set in the first data section is automatically converted to be set within the second and third data sections in accordance with the conversion classification information previously set, the service data is displayed on CRT 6 in the illustration format of Fig. 25 in accordance with the conversion classification and image classification information.

Fig. 25 is a table for representing one example of a relationship between the conversion classification, image classification and service data, which is displayed on the CRT.

In Fig. 25, if the conversion classification is "10" and the image classification is "10", only the data set in the first data section (see Fig. 10) is displayed on CRT 6 in the millimeter unit.

If the conversion classification is "10" and image classification is "20", the data set in the first data section is displayed in the millimeter unit, and subsequently, the data set in the second data unit is displayed within a parenthesis in the inch unit.

If the conversion classification is "20" and image classification is "10", only the data set in the second data is displayed in the inch unit.

When the conversion classification is "20" and image classification is "20", the data set in the second data is displayed in the inch unit, and subsequently, the data set in the first data unit is displayed within a parenthesis in the millimeter unit.

When the conversion classification is "40", the same data and unit display as in the conversion classification being "10" is performed, and its decimal point is not a period, but a comma.

As is similar to the steps S35 and S40, the data on the machine information is positioned on the basic layout and the coordinates thereof is input in a step S43.

In a subsequent step S44, the element data key and its coordinates information selected from the 1-element data, as illustrated in Figs. 6 to 8, are registered into the editing apparatus 3-1.

Not only the element data key, but also the respective element data together with the coordinates information thereof may be of course registered.

When a judgement is made that the editing work for one page is accomplished in the step S45, the format of one page data is converted into another format shown in Fig. 18 in a step S46. This conversion is executed in the editing apparatus for performing the editing work (the editing apparatus 3-1 in this case).

Fig. 18 illustrates a format of the data having one page information converted in the editing apparatus. In Fig. 18, the one page data converted in the editing apparatus is constructed of the following data:

- (A) The converted layout data which has been converted in the step S30 (see Fig. 12)
- (B) The coordinates entered in the previous steps

S35, S40 or S43, and the element data key of the 1-element data arranged in the region designated by said coordinates. When a plurality of element data (image data, document data, or machine sort information data) are arranged within one page, plural sets of the coordinates and element data key should be stored for one page.

The coordinates and element data key are set subsequent to the converted layout data. A set of coordinates and element data key will be referred to as "a layout for the editing work" in the following description.

(C) The element data key of the element data arranged within one page, LEN of the element data, and the element data. When a plurality of element data are inserted in one page, plural sets of the element data key, LEN, and element data should be arranged for one page.

In other words, the number of the 1-element data constituted of the element data key, LEN and element data is same as that of a pair of the coordinates and element data key contained in the layout for the editing work.

Subsequent to the layout for the editing work, the element data key, LEN and element data are set.

When the element data is the machine sort information, as apparent from Figs. 6 to 8, since no LEN is present in front of the element data, of course, only the element data key and element data are set.

These data arranged behind the layout for the editing work will be referred to an element data group in the following description (see Fig. 18).

Although not shown in Fig. 18, an end code for indicating that one page data is completed is added at the end of the final element data.

When one page data is converted as illustrated in Fig. 18, the resultant converted one page data is registered via the system controller 2 in the book D/B 21 in the step S47.

In the next step S48, one page data which has been converted in the previous step S46 is transferred to the host computer 1.

In a step S49, the above-described one page data is duplicated and the format thereof is converted into that illustrated in Fig. 19 in the host computer 1. Fig. 19 illustrates one page layout information which is converted in the host computer 1 so as to be registered in the layout D/B 11.

The one page data transferred to the host computer 1 is distributed and stored in the respective D/B (i.e., layout D/B 11, document D/B 12, image D/B 13 and machine sort information D/B 14) connected to the host computer 1. The data conversion executed in the step S49 is performed so as to store in the layout D/B 11, only the information relating to the layout among the one page data.

As is obvious from the comparison between Figs. 18 and 19, in the step S49, both LEN and the element data have been removed from the element data group

of the 1-page data which is set subsequent to the layout for the editing work.

In a next step S50, the data converted in the previous step S49 is registered in the D/B 11.

In a step S51, the element data group is copied by employing the 1-page data which has been transferred to the host computer 1 in the previous step S48.

- In a step S52, the data relating to the document data among the element data group is registered in the document D/B 12.

In a step S53, the data relating to the image data among the element data group is registered in the image D/B 13.

In a subsequent step S54, a judgement is made whether or not the edited 1-page data is to be printed out. If no printing operation is to be carried out, the control process is advanced to a step S57. If the printing operation should be performed, the control process is advanced to a step S55.

In a step S55, both the reference supervision No. and page are entered by using of the keyboard 5-2, so that the 1-page data registered in the book D/B 21 is called up into the editing apparatus 3-2.

In a step S56, the 1-page data is printed out by the printer 7.

In a step S57, a judgement is made whether or not the next page is produced. When the next page is produced, the control process is returned to the step S28. Conversely if the next page is not produced, the control process is accomplished.

In accordance with above-mentioned process (first editing method), a new book is newly produced or edited.

Referring now to Figs. 20 to 24, as is similar to the description on the first editing method, a description will be made of the second editing method applied to produce a service manual of a bike or the like.

Fig. 20 is a flowchart illustrating the second editing method

In the respective steps shown in Fig. 20, the same reference numerals as those in Fig. 3 indicate the same or similar processing operations.

According to the second editing method, in a step S25, a new book producing file is first called up which has been registered in the host computer 1, by employing any one of the personal computers 4-1 to 4-N (for instance, the personal computer 4-1) connected to the host computer 1. Since the arrangement of this new book producing file is identical to that shown in Fig. 11, the description thereof is omitted.

After the new book producing file is called up, predetermined data, i.e., the material supervision No. information, staff-in-charge information and starting page information are entered into the new book producing file in a step S26.

Next, as illustrated in a step S27, the new book producing file is transferred by the host computer 1 to the system controller 2 and registered in the book D/B 21.

In a step S101, referring to a book which has been produced in the editing apparatus (referred to as "an original book"), a selection is made that the pages of this original book are utilized to produce a new book. That is to say, the following page selection is carried out, as one example. The 21st to 30th pages of the first original book are available to produce the 1st to 10th pages of the new book, the 46th to 50th pages of the second original book are utilized to produce the subsequent 11th to 15th pages of the new book, and furthermore, the 31st to 40th pages of the first original book are usable to produce the succeeding 16th to 25th pages thereof.

Such a process is performed while an operator actually observes the original books.

In a next step S102, the material supervision No. and the page number of the original book are input by way of the personal computer 4-1, and 1-page data (see Fig. 19) is copied in the host computer 1 from the layout D/B 11.

In a step S103, a new material supervision No. and a new page number of a book to be newly edited are input into the personal computer 4-1.

In a step S104, both the document supervision No. and the page number in the 1-page data copied in the previous step S102 are rewritten by the new material supervision No. and page number of the new book input in the previous step S103.

Fig. 21 illustrates the data rewriting operation executed in the step S104.

The 1-page data of the original book which has been called up and copied in the step S102 is as shown in the upper portion of Fig. 21. In the step S104, as illustrated in the lower portion of Fig. 21, only the material supervision No. and page number among the above-described data are rewritten by those of the new book.

In the step S105, the 1-page data produced in such a way is stored in the host computer 1.

In a step S106, a judgement is made whether or not the rewriting operation of the material supervision No. and the page number is accomplished for the new book. In accordance with the previous example, a judgement is made whether or not all of the 21st to 30th pages of the first original book, the 46th to 50th pages of the second original book, and the 31st to 40th pages of the first original book have been converted into the 1st to 25th pages of the new book.

If one book is not yet rewritten, the control process is returned to a step S102. To the contrary, if the rewriting operation is accomplished, the control process is advanced to a step S107.

In the step S107, the element data or both LEN of the element data and element data corresponding to the element data key, among the 1-page data converted in the step S104 are called up from the respective D/B 12 to 14. These data are added after the respective element data keys of the 1-page data which are arranged in the rear of the layout for the editing work. That is to say, the 1-page data converted in the step S104 is rear-

anged in a complete form as shown in Fig. 18.

In a step S108, a judgement is made whether or not all pages of the new book have been rearranged in a complete form. If all pages of the new complete book are not yet rearranged, the control process is returned to step S107. If all pages are converted, the control process is advanced to a step S109.

In the step S109, the data of the new complete book is transferred from the host computer 1 to the system controller 2.

In a step S110, all pages of one complete book are registered in the book D/B 21 by the system controller 2.

In a subsequent step S111, the new material supervision No. and page number of the new book are entered by the keyboard 5-1 into the editing machine 3-1.

In a step S112, the input page data of the new book is read out from the book D/B 21 to be registered in the editing apparatus 3-1.

Then, in a next step S113, the page called up from the book D/B 21 is displayed on CRT 6.

In a step S130, a judgement is carried out whether or not the editing operation of the 1-page data displayed on CRT 6 is executed. That is to say, a judgement is made whether or not the pages which have been copied from the original books for the new book in the steps S102 to S104 can be merely utilized without any further edition. If no editing work is required, the control process is advanced to a step S46. If the editing work is required, the control process is advanced to a step S114.

In the step S114, a retrieval operation by any one of the personal computers 4-1 to 4-N is performed whether or not the elements of the original book (including the original book selected in the previous step S101) can be utilized for editing the page of the new book displayed on CRT 6. The retrieval method will be discussed later with reference to Figs. 22 to 24.

In a step S115, a judgement is made whether or not the element usable in the new book editing operation is found in the above-described retrieval operation.

If some relevant elements are found, the element data key of the retrieved relevant elements are input in the personal computer (for example, 4-1) in a step S116.

In a step S117, the 1-element data called up by the above retrieval operation is transferred to the system controller 2, and is registered in the element D/B 22 under the control of the system controller.

In a step S118, a judgement is made whether or not the retrieval operation is continued. If yes, then the control step is returned to the step S114.

If the element usable for editing the new book is not found in the step S115, the 1-element data retrieved in the step S114 is cleared in a step S132. It is, of course, not to clear the data stored in the D/B 12 to 14 at this stage.

In the step S119, a judgement is made whether or not the image data, document data or machine information data, namely element is newly input. If the element is not newly input, the control step is returned to the step

S118.

When the element is newly input, the control process is advanced to the step S3. Since the processes effected in the steps S3 to S23 are same as those denoted by the same reference numerals in Fig. 3, the descriptions thereof are omitted.

If the control process defined by the step S22 or S23 is completed, the control step is returned to the step S119.

In the step S118, if the retrieval operation is not continued, another judgement is made whether or not the image is displayed on CRT 6 in the step S33. If the image is displayed, the temporary key or element data key of the image to be displayed is input by the keyboard 5-1 in the step S120, and the predetermined image is read out from the element D/B 22 and then displayed on CRT 6.

In the next step S35, the image is moved to the desired position and the coordinates thereof are input.

In the step S121, a judgement is judged whether or not the temporary key is input in the previous step S120.

When the temporary key is input, it is converted into the element data key in the step S36. The data conversion has been described with reference to Fig. 3, so that no further description is made here.

If a judgement is done that no temporary key was entered in the step S121, or after the process of the step S36 is accomplished, the element data key, LEN of the image data corresponding to the element data key and the image data itself are registered as one element data into the editing apparatus 3-1 together with the coordinates input in the previous step S35.

In the step S33, if a judgement is made that the image is not displayed, another judgement is done whether or not the document or sentence is displayed in the step S38.

When the document is displayed on CRT 6, the temporary key or element data key to be displayed is entered by the keyboard 5-1 in a step S122, and the predetermined document is called up from the element D/B 22 to be displayed on CRT 6.

In a subsequent step S40, the above document is moved to the desired position and the coordinates thereof are entered.

In a step S123, a judgement is made whether or not the temporary key was input in the previous step S122. When the temporary key is input, it is converted into the element data key in the step S41. This key conversion is carried out as same as in the conversion effected in the preceding step S36.

If a judgement is done that no temporary key was input in the step S123, or after the process effected in the step S41 is accomplished, the element data key, and both LEN and the sentence data of the document data corresponding to the element data key, are registered as one element data into the editing apparatus 3-1 together with the coordinates input in the preceding step S35.

If a judgement is made that no sentence was displayed in the preceding step S38, it is judged that the machine sort information is displayed, and the control process is advanced to a step S131.

In the step S131, the element data key of the machine sort information to be displayed is entered by the keyboard 5-1 and the predetermined machine sort information is called up from the element D/B 22 to be displayed on CRT 6.

In the next step S43, the above machine sort information is moved to the desired position and the coordinates thereof are input.

In a step S44, the element data key is registered together with the coordinates entered in the step S43 in the editing apparatus 3-1.

When the process in the step S37 or S44 is completed, a judgement is made whether or not the 1-page data displayed on CRT 6 has been edited. If not yet, the process is returned to the step S33. If completed, then the process is advanced to a step S46.

In a step S46, the edited one page data is converted in the form as illustrated in Fig. 18 in the editing apparatus 3-1.

In the next step S124, the converted one page data is updated in the book D/B 21.

Since the processes defined in the steps S48 through S57 are the same as those denoted by the same reference numerals of Fig. 3, the explanations thereof will be omitted. If in the step S57, the judgement is performed to edit the next page, the control process is returned to the step S111.

It is obvious that although in the explanations with reference to Fig. 2C, the retrieval operation was previously executed before the 1-page data was edited in the steps S33 and thereafter, the necessary data were read out of the D/B 12 through 14 to be registered in D/B 22, these processes may be performed during the editing work.

A detailed description will now be made in that the process of the above-described step S114 is carried out for the document data retrieval operation. The retrieval operation of the step S114 is performed by employing any one of the personal computers 4-1 to 4-N.

Fig. 22 is a schematic diagram showing a portion of a table for the document data retrieval which has been previously set in the document D/B 12. A symbol "0" indicated in the respective tables of Fig. 22 represents "null", or that no data is entered.

In the document D/B 12, as illustrated in Fig. 22, a plurality of document data retrieval tables each consisting of an index (referred to as an "ID" simply) and a title have been input and set. A byte length of the ID is set to be equal to that of the detailed item code (6 bytes in the preferred embodiment) in order to correspond to the detailed item code of the element data key which in turn corresponds to the document or sentence data shown in Fig. 17.

In an a-table of Fig. 22, the different data such as

A, B, C, and so on, are respectively input only in the head byte (first byte) of ID each having a 6-byte length.

In a b-table of Fig. 22, the common data of "A" is input in the first byte of each ID, and the different data of "A, B, C" are respectively input in the second byte thereof.

In a c-table of Fig. 22, the common data of "B" is entered in the first byte of each ID and the different data of "A, B, C" are entered, respectively, in the second byte thereof.

In a d-table of Fig. 22, the common data of "AA" is input in the first and second bytes of each ID, and also the different data of "A, B, C" are entered, respectively in the third byte.

In an e-table of Fig. 22, the common data of "AB" is input in the first and second bytes of each ID, and the different data of "A, B, C" are entered, respectively, in the third byte thereof.

Similarly, in an f-table of Fig. 22, the common data of "AAA" is input in the first to third bytes of each ID, and the different data of "A, B, C" are entered, respectively, in the fourth byte thereof.

In a g-table of Fig. 22, the common data of "AAB" is input in the first to third bytes of each ID, and the different data of "A, B, C" are entered, respectively, in the fourth byte.

Thus, the respective tables for the document data retrieval operation set in the document D/B 12 are formed in a hierarchical structure as illustrated in Fig. 23.

In practice, each of the tables for the document data retrieval operation are formed in the following method. That is to say, in correspondence to the respective ID's in the table (a-table) of the first hierarchy or rank where the data has been input in only the first byte of each ID, each title (auto-bike, automobile, or special-purpose car etc.) representative of the first highest concept of the document data is input. Also, in correspondence to each of ID's in the tables (b and c-tables) of the second hierarchy or order where the data has been input in only the first and second bytes of each ID, another title (explanation, or notice etc.) representative of the second highest concept is input. Similarly, still other title (structure explanation, operating or handling manual etc.) representative of the third rank concept is input in the tables (d and e-tables) of the third hierarchy, and further title (twelfth month maintenance, or sixth month maintenance, etc.) indicative of the fourth rank concept is input in the tables (f and g-tables) of the fourth hierarchy. Also the data same as the ID representative of the lowest concept is previously input in the detailed item code of the element data key shown in Fig. 17. And then the retrieval operation of the document data is performed as illustrated in Fig. 24.

It should be noted that the lowest hierarchy data need not be set in the table of the sixth hierarchy, but may be set in the table higher than the sixth hierarchy.

Fig. 24 is a flowchart for illustrating the retrieval method indicated in the step S114. As previously de-

scribed, the retrieval operation is carried out by any one of the personal computers 4-1 to 4-N.

In a step S150 of Fig. 24, first "n" is set to be 1.

In a step S151, ID of the n-th hierarchy or rank is input by, for instance, the personal computer 4-1.

In a step S152, the (n+1)th hierarchy data containing the n-th hierarchy ID in ID thereof is selected from the table for the document data retrieval operation.

In a step S153, a judgement is made whether or not the (n+1)th hierarchy data is present, namely, the (n+1)th hierarchy data is selected in the step S152.

If yes, then all titles of the (n+1)th hierarchy are displayed in the step S154. Then, after 1 is added to "n" in a step S155, the control process is returned to the step S151.

When a judgement is made that the (n+1)th hierarchy data is not present in the step S153, the control process is transferred to the step S156.

In a step S156, the 1-element data containing the n-th hierarchy ID in the detailed item code thereof (i.e., 1-document or 1-sentence data shown in Fig. 17) is called up via the host computer 1 from the document or sentence D/B 12 and displayed in the personal computer 4-1.

Consequently, when the table for the document data retrieval operation is constructed as in Fig. 22, and "A" is input as ID only to the first byte, the b-table of the second hierarchy or rank where "A" is set in the first byte is selected and displayed, as indicated by an arrow "P".

Then, when "AA" is input in the first and second bytes as ID, the d-table of the third hierarchy where "AA" is set in the first and second bytes is displayed, as denoted by an arrow Q.

Similarly, when "AAA" is entered as ID in the first to third bytes, the f-table of the fourth hierarchy where "AAA" is set in the first to third bytes is displayed, as shown by an arrow R.

According to the above-mentioned method, such a retrieval operation can be done to show that such documents or sentences as a twelfth month maintenance manual are registered with respect to the document of the structure explanation on, for example, the auto-bike.

Furthermore, when "AAAA" is input as ID into the first to fourth bytes for the retrieval operation, the list of the fifth hierarchy (not shown) where "AAAA" is set in the first to fourth bytes is displayed as denoted by an arrow S.

When such a retrieval process is performed for the list of the lowest hierarchy, a judgement can be made whether or not the required document data is registered in the document D/B 12.

When the necessary data is retrieved, the retrieved data can be utilized to edit a new book if the retrieved data is registered in the element D/B 22 as previously described with regards to the step S117.

Although the above explanation was made on the retrieval operation of the document data, the image data or machine sort information may be similarly retrieved.

That is to say, if a plurality of retrieval tables as illustrated in Fig. 22, are set in the image D/B 13 and machine sort information D/B 14, and if ID's of the retrieval tables are input into the items (3-byte length; see Figs. 6 to 9) in the element data key of the image data and machine

sort information, the above-described retrieval operation may be performed.

Although in the preceding description, the machine sort information and image were registered into the respective different D/B (i.e., the machine sort information D/B 14 and image D/B 13), they may be registered in a common D/B since the data of the machine sort information shown in Figs. 6 to 8 has the same format as that of the image data shown in Fig. 9.

Also when the image and document data were input in the previous embodiment, the temporary key was input first and converted into the element data key in the actual editing operation. However, the temporary key may not be input in advance but the element data key may be input at the beginning, which is similar to the entry of the machine sort information data.

According to the embodiment described above, since a new book can be formed by making use of an original book formed by the editing apparatus, the new book is produced efficiently.

Where a service manual or a shop manual covering a machine similar to what is covered in an already edited manual is to be newly produced, since the greater part of the former manual is available for the new production, the new manual can be produced very quickly and inexpensively.

Of course, the editing apparatus of the embodiment can be used for editing books in general, magazines, and newspapers as well as for producing service manuals and shopmanuals.

Incidentally, the second procedure of editing illustrated in Fig. 20 has been depicted as what requires pertinent data to be introduced and registered in the file for the production of a new book at the step S25 to S27. Said procedure, however, does not always require introduction of the material ID No. or the starting page of the file for the production of the new book, because the editing apparatus is so constructed that the data of such particulars are to be introduced at the step S103.

The processing at the step S150 illustrated in fig. 24 may be omitted. The retrieval of data can be started by the introduction of ID of a lower rank than the first rank.

Now, the construction of the first embodiment of this invention will be described. Fig. 32 is a functional block diagram illustrating the construction of this invention. In Fig. 32, the reference numerals which have equals in Fig. 2 denote identical or similar parts.

In Fig. 32, a keyboard 106 and 106A are similar in function to the keyboards 5-1 to 5-N and the keyboards of the personal computers 4-1 to 4-N illustrated in Fig. 2 and a scanner 107 to the scanner 8 illustrated in the same drawing.

Similarly, second element memory means 110 is similar in function to the element D/B 22 of Fig. 2, second layout memory means 111 to the basic layout D/B 23, and page memory means 112 to the book D/B 21 of the same drawing.

In Fig. 32, the keyboard 106 is connected to basic layout code setting means 101, element input means 102, element data key setting means 103, and layout code conversion means 114.

The basic layout code setting means 101 sets the basic layout code (refer to Fig. 4) to be fed in at the step S1 of Fig. 3.

The element data key setting means 103 sets the element data key illustrated in Figs. 6 to 9 and Fig. 17.

The element input means 102 receives convert the document data fed in through the keyboard 106 and the image data fed in through the scanner 107.

The layout code conversion means 114 converts the "layout code" part of the basic layout code introduced through the basic layout code setting means 101 into the material ID No. and page.

The basic layout code setting means 101 is connected to the second layout memory means 111 in the memory means 111. The second layout memory means 111 is connected to the layout code conversion means 114. The element input means 102 and the element data key setting means 103 are connected to the second element memory means 110 in the memory means 113.

The second element memory means 110 is connected to coordinates setting means 104, first page data setting means 105, and second page data setting means 133. The coordinates setting means 104 fulfills the role of feeding the element data key supplies from the second element memory means 110 and the element data corresponding to the element data key to display means 109 and, at the same time, setting the coordinates of the element data to be moved by the of manipulation of a mouse 108, and feeding the information of the coordinates to the display means 109, the first page data setting means 105, and the second page data setting means 133.

The layout code conversion means 114 calls out a desired one of the basic layout codes stores in the second layout memory means 111, converts the basic layout code into such a converted layout code as illustrated in the lower part of Fig. 12, and supplies the converted layout code to the first page data setting means 105 and the display means 109.

The first page data setting means 105 which is actuated when a new book is to be produced wholly newly, serves to prepare such data for one page as illustrated in Fig. 18 by using the element data key and the element data supplied from the second element memory means 110, the converted layout code supplied from the layout code conversion means 114, and the information on the coordinates supplied from the coordinates setting means 104, and supply the produced data for one page to the page memory means 112 in the memory means

113.

The element input means 102 is connected with a broken line to the display means 109, indicating that the data concerning documents, for example, are exhibited on the display means 109 when such documents as letters and sentences are to be introduced through the keyboard 106.

The page memory means 112 is connected to layout conversion means 121 and element conversion means 122.

The layout conversion means 121 prepares such information on layout as illustrated in Fig. 19 by removing the LEN and element data from the group of element data in the one page data (Fig. 18) memorized in the page memory means 112. The layout conversion means 121 is connected to first layout memory means 125 which is in turn connected to information processing means 124. The first layout memory means 125 is a device for memory which corresponds to the layout D/B 11 illustrated in Fig. 2.

The element conversion means 122 copies a group of element data from the data for one page memorized in the page memory means 112 to decompose into units of element. The element conversion means 122 is connected to the first element memory means 123 which is connected to the information processing means 124. The first element memory means 123 is provided with a document D/B 12, an image D/B 13, and a machine sort D/B 14.

The first layout memory means 125 is further connected to page discrimination data alteration means 131. The page discrimination data alteration means 131, when a book is to be produced by utilizing the information on another book already edited by the present editing apparatus and memorized in the first layout memory means 125 and the first element memory means 123, reads necessary part of the layout information memorized in the first layout memory means 125 and converts the material ID No. and page number in the layout information (namely, the page discrimination data for identifying the page of the layout information) into the material ID No. and page number of the new book to be produced. The data necessary for said conversion are introduced through the keyboard 106A.

The page discrimination data alteration means 131 and the first element memory means 123 are connected to element data addition means 132. The element data addition means 132 reads out of the first element memory means 123 the element data corresponding to the element data key being set after the layout on editing in the layout information (Fig. 19) which is supplied from the page discrimination data alteration means 131 and adds the element data to the element data key.

The element data addition means 132 is connected to the page memory means 112 which is further connected to the second page data setting means 133 and the display means 109.

The second page data setting means 133 which is

actuated when a new book similar to an already edited book is to be produced based on the edited book, effects modification of a part of the information for one page supplied from the second element memory means 112, based on the data in the element unit supplied from the second element memory means 110 and the positional coordinates of the element data supplied from the coordinates setting means 104. Thereafter, it feeds back the modified information for one page to the page memory means 112. The second page data setting means 133, when the data in element units are not supplied from the second element memory means 110, causes the information for one page conveyed from the page memory means 112 to be returned in their unmodified form to the page memory means 112.

The first element memory means 123 is further connected to the second element memory means 110.

The basic layout code setting means 101, the element input means 102, the element data key setting means 103, the coordinates setting means 104, the first page data setting means 105, the second page data setting means 133, and the layout code conversion means 114 are severally provided for the editing devices 3-1 to 3-N or the system controller 2 illustrated in Fig. 2. The layout conversion means 121, the element conversion means 122, the page discrimination data alteration means 131, and the element data addition means 132 are provided severally for the host computer 1, the system controller 2, or the personal computers 4-1 to 4-N illustrated in the same drawing.

The embodiment of Fig. 32 has a function to be fulfilled in wholly new production of a new book and a function to be fulfilled in the production of a book similar to an already edited book, based on that edited book.

First, the operation of the editing apparatus for the wholly new production of a book will be described in detail below.

In the first place, information for designating a basic layout for one page is introduced through the keyboard 106. In the basic layout code setting means 101, the information is recomposed in such a basic layout code as illustrated in Fig. 4. The basic layout code is memorized in the second layout memory means 111.

The introduction and memorization of the basic layout may be carried out a plurality of times when necessary.

Then, the element data key corresponding to the element data to be introduced is fed in through the keyboard 106. In the element data key setting means 103, the element data key is set in such a form as illustrated in Figs. 6 to 9 and Fig. 17.

After the introduction of the element data key, information on documents or machine sort is fed in through the keyboard 106 or images are taken in through the scanner 107. The data thus introduced are fed out to the element input means 102. In the element input means 102, an LEN for the element data is attached when necessary to the element data. The LEN-attached element

data are fed out to and memorized in the second element memory means 110 in combination with the element data key. The introduction and memorization of the element data key and element data may be carried out a plurality of times as occasion demands.

Then, the necessary part of the basic layout code memorized within the second layout memory means 111, namely the basic layout code in which the layout data for a page to be edited are set, is read out of the second layout memory means 111 by the introduction of the layout code through the keyboard 106 and then fed out to the layout code conversion means 114.

In the layout code conversion means 114, the "layout code" part of the basic layout code is converted into the material ID No. and page introduced through the keyboard 106. As the result, a converted layout code (as illustrated in the lower part of Fig. 12) is produced. The converted layout code is fed out to the first page data setting means 105 and the display means 109.

From the second element memory means 110, either the element data to be used for editing a page or the LEN and the element data are fed out in conjunction with the element data key corresponding to the element data, to the first page data setting means 105 and the coordinates setting means 104. The coordinates setting means 104 supplies the element data to the display means 109.

When the element data exhibited on the display means 109 are moved by the mouse 108, the coordinates corresponding to the position assumed by the movement are set by the coordinates setting means 104. The coordinates are transferred to the first page data setting means 105.

Optionally, a plurality of sets of such element data are displayed at pertinent positions on the basic layout, to effect the editing of a page.

When the work of editing one page is completed, such data for one page as illustrated in Fig. 18 are produced in the first page data setting means 105, based on the output signals of the layout code conversion means 114, the second element memory means 110, and the coordinates setting means 104. To be more specific, the data for one page are composed of the converted layout code fed out of the layout code conversion means 114, the layout on editing consisting of the positional coordinates fed out of the coordinates setting means 104 and the element data key of the element data corresponding to the positional coordinates, and the group of element data consisting of the element data key forming the layout on editing and the element data corresponding the element data key.

The one page data produced in the first page data setting means 105 are memorized in the page memory means 112. In the page memory means 112, as many pages of edited data as are required for completing the book are memorized. The data are fed out to hard copy forming means (not shown).

The data for one page memorized in the page mem-

ory means 112 are transmitted to the layout conversion means 121. In the layout conversion means 121, such layout information as illustrated in Fig. 19 is produced from the data for one page. The layout information is transmitted to and memorized in the first layout memory means 125.

The data for one page memorized in the page memory means 112 are transmitted to the element conversion means 122. In the element conversion means 122, the group of element data is copied from the data for one page, decomposed into units of element, and transmitted to the first element memory means 123. Of the decomposed group of element data, those concerning the element of document are memorized in the document D/B 12 in the first element memory means 123, those concerning the image element in the image D/B 13, and those concerning the element of machine sort information in the machine sort information D/B 14, respectively.

The element information memorized in the first element memory means 123 and the layout information memorized in the first layout memory means 125 are called out, when necessary, by the information processing means 124 to be utilized for the various information processing and retrieval to be executed by the information processing means 124.

As the result, a new book is completed. The data on the new book are decomposed into those concerning layout and those concerning element and memorized.

Now, the operation of the editing apparatus in the production of a book similar to the book edited as described above, based on the edited book, will be described below.

The operator personally examines the already edited books to find which books and which pages of the books are available for a new book, feeds the material ID Nos. and pages (page identification data) of the books to be utilized, and reads out the data of the selected pages from the first layout memory means 125. The data thus read out are transmitted to the page discrimination data alteration means 131.

When the material ID No. and pages of the book to be produced are fed in through the keyboard 106, the material ID No. and pages of the data so read out are converted in the page discrimination data alteration means 13 into the material ID No. and pages so introduced.

The element data addition means 132 reads out of the first element memory means 123 the "element data" corresponding to the element data key set after the layout of editing in the converted data for one page and suffixes them to the element data key. This step complete the data for one page.

The data for one page are memorized in the page memory means 112, then transmitted to the display means 109 for display thereon, and also transmitted to the page data setting means 133.

The operator examines the data for one page ex-

hibited on the display means 109 to find whether the data are in need of any alteration or not. When the absence of necessity for data alteration is confirmed, the information announcing this fact is fed out from suitable means (not shown) and the data are transmitted to the page memory means 112. In the layout conversion means 121 and the element conversion means 122, the data are decomposed into those concerning the elements and those concerning the layout. In this case, since the data concerning the elements have been already memorized in the first element memory means 123, only the data concerning the layout are fed out to and memorized in the first layout memory means 125.

When the desired "element data" part of the data for one page exhibited on the display means 109 is to be altered, namely when the data for one page exhibited on the display means 109 are to be revised, the element information comprising the element data key and the element data is read out of the first element memory means 123 and registered in the second element memory means 110 by introducing the element data key corresponding to the required element data.

When the element data required for a new book to be edited are not found in the first element memory means 123, the element data key and the element data are newly introduced and registered in the second memory means 110.

The second element memory means 110 is urged by means (not shown) to supply necessary element data and element data key to the coordinates setting means 104 and the second page data setting means 133. The coordinates setting means 104 supplies the element data to the display means 109, sets the coordinates of the element data fixed by the manipulation of the mouse 108, and supplies the coordinates to the second page data setting means 133.

The operations described above are repeated when necessary. When the one page data exhibited on the display means 109 have been altered (edited), the second page data setting means 133 causes the data for one page transmitted from the page memory means 112 to be reset into edited data for one page. To be more specific, the second page data setting means 133 erases from the data for one page transmitted from the page memory means 112, the element data superposed with the element data set newly on the layout by editing, the element data key corresponding to the element data, and the positional coordinates thereof and, at the same time, supplements the data for one page with the element data set newly on the layout by editing, the element data key corresponding to the element data, and the positional coordinates of the element data.

The data for one page altered as described above are supplied from the second page data setting means 133 to the page memory means 112 and memorized therein. The data for one page are supplied in unit of page or book to the layout conversion means 121 and the element conversion means 122. The layout conver-

sion means 121 and the element conversion means 122 decompose the supplied data into those concerning the elements and those concerning the layout to memory, respectively, in the first layout memory means 125 and the first element memory means 123. In this case, those of the data concerning the elements which have been already memorized in the first element memory means 123 are not doubly memorized.

When the operation described above is performed in the production of a book similar to an already edited book, since common element data can be utilized in their unmodified form in the production of the new book, there is obtained a generous saving in the time required for the production of the book.

The embodiment of Fig. 32 has been depicted as so constructed that the basic layout code is preparatorily set in the basic layout code setting means 101, the basic layout code in which the page layout for the page to be edited is set is called out by the use of the layout code, before the editing work is started, and the "layout code" part of the basic layout code is converted into the material ID No. and page in the layout code conversion means 114. This construction is convenient because the layout of a page to be edited (namely the layout data illustrated in Fig. 4) is not required to be set each time a new page is edited.

The embodiment described above is not specifically restricted to those construction. Optionally, the construction may be modified so that the converted layout code illustrated in the lower part of Fig. 12 may be directly set by introducing the material ID No. and page during the introduction of the layout data. In this case, since the converted layout code fixes the layout of only the page to be edited, the converted layout code is not required to be memorized in the second layout memory means 111. The modified construction is not very suitable for the work of editing a book of a large number of pages because the layout code must be set each time a new page is to be edited. It is nevertheless simple as compared with the construction of the embodiment illustrated in Fig. 32.

The embodiment illustrated in Fig. 32 has been depicted as so constructed that, of the data on the elements decomposed by the element conversion means 122, those concerning the image are memorized in the image D/B 13 and those concerning the information on the machine sort in the machine sort information D/B 14. The present embodiment is not required to be specifically restricted to those construction. The construction may be modified so that either of the image D/B 13 and the machine sort information D/B 14 is installed and the data concerning the image and the data concerning the machine sort are both memorized therein. Since the data concerning the image and the data concerning the machine sort are each furnished with element data keys of one and the same format as illustrated in Fig. 9 and Figs. 6 to 8, these data can be handled in the common identical form.

Further, when the element data key of the data concerning the documents illustrated in Fig. 17 is written in the same format as that of the data concerning the image and the machine sort information, the data concerning the documents can be memorized in the D/B intended for the memorization of the data concerning the image and the machine sort information.

Further the embodiment mentioned above has been depicted as so constructed that the data for one page produced by the first page data setting means 105, the second page data setting means 133, and the element data addition means 132 are memorized in the page memory means 112. When the page memory means 112 is so adapted as to permit memorization of the data for a plurality of pages and the host computer 1 in Fig. 2 is endowed with the functions of the layout conversion means 121, the element conversion means 122, the page discrimination data alteration means 131, and the element data addition means 132, the otherwise inevitable temporary concentration of load upon the host computer 1 is precluded as mentioned below.

When the editing apparatus is not provided with the page memory means 112, since the host computer 1 is required to call out and preserve data each time the data for one page are fed out for the purpose of editing and printing, for example, or after the editing of one page is completed, it must supply data rather frequently to the first element memory means 123 and the first layout memory means 125. As the result, there are times when the host computer 1 is exposed to temporary concentration of load.

In contrast, when the editing apparatus is provided with the page memory means 112, since the data for all the pages of one book can be memorized wholly in the page memory means 112 preparatorily to the editing of the book by virtue of the page discrimination data alteration means 131 and the element data addition means 132, the host computer 1 can be used for processing all the data introduced from the various devices (such as the personal computers 31-1 to 31-N of Fig. 2) connected to the host computer 1 and installed for the purpose other than the editing by the editing apparatus until the processing of the data for the whole book such as editing and printing is completed. As the result, the operation of the host computer 1 with respect to the processing for the editing apparatus is appreciably decreased and, consequently, the load on the host computer 1 is dispersed and alleviated.

When the page memory means 112 has the data of pages of the book to be edited memorized therein, the work of editing the book can be performed even when the office is closed as on a holiday and the operation of the host computer 1 is stopped.

On the condition that inconveniences of the nature suggested above are tolerated, it may be naturally permissible to so modify the present embodiment that the data for one page output from the first and second page data setting means 105 and 133 are directly supplied to

the layout conversion means 121 and the element conversion means 122 and the data for one page fed out of the element data addition means 132 are directly supplied to the second page data setting means 133 and the display means 109.

Further, the embodiment has been depicted as so constructed that the element data key and the element data newly set by the element input means 102 and the element data key setting means 103 and the element data key and the element data supplied from the first element memory means 123 are memorized in the second element memory means 110. The otherwise possible concentration of load upon the host computer 1 can be precluded by installing the second element memory means 110 as described above and consequently allowing the data concerning the elements to be called out abundantly in advance from the first element memory means 123. When the inconveniences are tolerated, the installation of the second element memory means 110 may be omitted.

Further, the installation of the data processing means 124 illustrated in Fig. 32 is not critical.

As is clear from the description given above, the present embodiment brings about the following effect because it is so constructed that the data of a book edited by the editing apparatus itself are decomposed into data concerning the layout and data concerning the elements for storage therein and these data are then utilized for the formation of data for one page. For example, when a new book is to be produced and the new book and a book already edited by the editing apparatus have common documents and illustrations, for example, these common documents and illustrations are not required to be newly fed in and these data may be utilized for the production of the new book. Thus, the production of the new books is simplified and the operation is performed quickly.

With reference to Fig. 1, element memory means 161 memorizes the element data representing such images as documents, photographs, and illustrations and the element data key corresponding to the element data.

Layout memory means 162 memorizes the layout data, the LEN, and the layout code corresponding to the layout data.

Page memory means 163 memorizes the page data composed of material ID No. page, layout data, coordinates, and element data key as illustrated in Fig. 19.

Data memory means 169 which comprises the element memory means 161, the layout means 162, and the page memory means 163 is connected to editing means 160.

Editing information input means 167 is for introducing information necessary for the execution of the editing work. The editing means 160 carries out the editing work in accordance with the information fed in through the editing information input means 167.

The information required by the operator in executing the work of editing is supplied from the editing means

160 to the display means 168.

Security No. input means 166 is used by the operator ready for the operation of the editing means 160 to feed in his own security No.

The security No. memory means 165 memorizes the security Nos. of the operators who are authorized to operate the editing means 160.

Security No. discrimination means 164 examines the security No. fed in through the security No. input means 166 to determine whether said particular security No. is found among the security Nos. memorized in the security No. memory means 165.

Incidentally, the security No. discrimination means 164 and the security No. memory means 165 may be provided indiscriminately for any of the host computer 1, the system controller 2, and the editing devices 3-1 to 3-N.

Now, the operation of the second embodiment will be described more specifically with reference to Fig. 1, Fig. 26, and Fig. 27.

Fig. 26 is a flow chart of the second embodiment applied to the first procedure of editing mentioned above.

With reference to Fig. 1 and Fig. 26, when the operator wishes to produce a new book by the first procedure of editing, he is required to feed in the data on elements to register in the element memory means 161 at the step S201, and then input pertinent information serving as his own security No. through the security No. input means 166 such as a keyboard or a magnetic reader at the step S202. The pertinent information may be the data of a staff in charge who is responsible for the formation of a new book illustrated in Fig. 11.

When the security No. is fed in, the security No. discrimination means 164 examines the security No. at the step S203 to define whether said particular security No. is memorized in the security No. memory means 165 as the security No. of the operator authorized to produce the new book by the editing apparatus.

When the absence of the security No. in the memory is confirmed, the security No. discriminating means 164 exhibits a message announcing that the operator is not authorized to operate the editing means 160 on the display means 168 formed of CRT, for example, at the step S204. At this point, the processing is immediately terminated.

Conversely when the presence of the security No. in the memory is confirmed, the protection lock controlled by the security No. is released and the operator is allowed to proceed to produce the new book.

When the protection lock is released, the operator begins to produce the new book by feeding in various kinds of data through the editing data input means 167.

The operator feeds in the material ID No. of the book to be newly produced at the step S205, and the page number intended to be produced at the step S206.

Subsequently, the operator, for the purpose of setting the layout of the page designated, feeds in the lay-

out code at the step S207.

When the layout code is fed in, the layout information designated by said particular layout code is called out of the layout memory means 162. At the step S208, the basic layout corresponding to the layout information is exhibited on the display as illustrated in Fig. 13.

Then, at the step S209, the operator feeds in the element data key corresponding to the image he wishes to insert and consequently calls the image from the element memory means 161.

The image, at the step S210, is exhibited on the display. At this time, in which position on the basic layout the image is to be inserted is designated by the use of a mouse (not shown), for example, connected to the editing means 160.

When the position for the insertion of the image is fixed with the mouse, the coordinates (X1, Y1) of the upper left corner of the image and the coordinates (X2, Y2) of the bottom right corner of the image on the basic layout are fed into the editing means 160.

At the step S212, a decision is made as to whether the editing of one page has been completed or not. When the image to be inserted on the same page still remains, the processing returns to the step 209.

When the editing of one page is completed, such page data as illustrated in Fig. 19 are formed and memorized in the page memory means 163 at the step S213. Thereafter, the processing is moved to the step S214.

At the step S14, a decision is made as to whether the editing of one book has been completed or not. When one or more pages to be edited still remain, the processing returns to the step S207. When the editing of one book is completed, the processing is automatically terminated.

Next, the operation of the second embodiment which the operator performs in the production of a new book by the second procedure of editing will be described below. Fig. 27 is a flow chart illustrating the editing operation. In Fig. 27, the reference numerals which have equals in Fig. 26 denote identical or similar parts. The explanation of these parts, therefore, will be omitted here.

With reference to Fig. 1 and Fig. 27, when a new book is to be produced by the second procedure mentioned above, after the processing in the step S203 has been completed, material ID No. of the original book is fed at the step S220 and the page number desired to be copied out of the original book at the step S221, respectively.

At the step S222, the data for one page corresponding to the material ID No. and page number are called out of the page memory means 163 to be supplied to the editing means 160.

Subsequently the material ID No. of the new book is fed in at the step S223 and the page number of the new book at the step S224, respectively.

At the step S225, the material ID No. and page of the data for one page of the original book memorized in

the editing means 160 at the step S222 are rewritten into those of the new book fed in at the steps S223 and S224 as illustrated in Fig. 21.

At the step S226, the element data of photographs, illustrations, etc. inserted in the copied page are examined to determine whether they should be replaced with other element data or not. When the absence of necessity for the replacement is confirmed, the processing is moved to the step S213.

Conversely, when the necessity for the replacement is present, the element data are newly fed in at the step S227 and/or the editing processing similar to that in the first procedure of editing at the steps S209 - S212 is executed.

At the step S213, the data for one page rewritten at the step S225 or the data for one page edited at the step S227 are memorized in the page memory means 163 and, at the same time, the necessary part of the data concerning the elements is memorized in the element memory means.

As noted clearly from the description given above, in the second embodiment, the security control is effected by means of the operator's security No.

Now, the third embodiment which permits further elaboration of security control by the secret level of the memorized data and the security No. of the operator will be described below.

Fig. 28 is a functional block diagram illustrating the construction of an embodiment in accordance with the invention. In Fig. 28, the reference numerals which have equals in Fig. 1 denote identical or similar parts. The explanation of these parts, therefore, will be omitted here.

Editing information input means 172 serves for feeding in information necessary for executing the work of editing. Data ID No. input means 173 receives the material ID No. for the control or supervision of books to be edited or the element data key for the control of images such as photographs (hereinafter, the material ID No. and the element data key will be collectively referred to as "data ID No.").

Data ID No. memory means 170 memorizes the security No. of the operator and the data ID No. corresponding to the security No. as illustrated in Fig. 31. The operator owning the particular security No. is authorized to utilize the editing apparatus for the work of editing.

The editing means 160 is connected to the editing information input means 172 and adapted to perform the editing work in accordance with the information supplied from the editing information input means 172.

Data ID No. discrimination means 171 examines the data ID No. fed in through the data ID No. means 173 to determine whether it is memorized in the data ID No. memory means 170 as the data ID No. authorizing the operator owning the security No. fed in through the security No. input means or not.

Optionally, the data ID No. memory means 170 and

the data ID No. discrimination means 171 may be provided indiscriminately in any of the host computer 1, the system controller 2, and the various editing devices 3-1 to 3-N.

Now, the operation of the embodiment will be described in detail below with reference to Fig. 28, Fig. 29, and Fig. 30. Fig. 29 is a flow chart illustrating the embodiment as applied to the first procedure of editing. In Fig. 29, the reference numerals which have equals in Fig. 26 denote identical or similar parts. The explanation of these parts will be omitted here.

With reference to Fig. 28 and Fig. 29, when a new book is to be produced by the first procedure of editing mentioned above, the operator at the step S202 feeds in his own security No. and the security No. discriminating means 164, at the step S230 in response to the input, examines whether or not said particular security No. is memorized in the data ID No. memory means 170 as the security No. for the control of the data ID No. When the absence of the particular security No. is confirmed, the processing is terminated at the end of the step S204.

When the presence of the security No. is confirmed, all of the data ID Nos. controlled by the particular security No., namely the group of data ID Nos. of the books which the operator is authorized to edit and inspect, are selected from among the data ID Nos. memorized in the data ID No. memory means 170 at the step S231.

When "A1234" is fed in as a security No., for example, the data ID No. discriminating means 171 selects the group of a total of five data ID Nos., specifically "B10" to "B16" and "NEW" as the data ID Nos. controlled by "A1234" as shown in Table 2 of Fig. 31.

The term "NEW" as used herein refers to the data ID No. which is assigned to the operator authorized to produce the new book by the use of the editing apparatus. It does not mean the presence of any data with the data ID No. "NEW".

At the step S205, the material ID No. of the book to be newly produced is fed in as the data ID No. thereof.

At the step S232, the decision is made as to whether or not the "NEW" is present among the group of data ID Nos. mentioned above and the material ID No. is present nowhere in the data ID No. memory means 170. When the result of the decision is in the affirmative, the operator is judged as authorized to produce the new book by the editing apparatus.

Conversely, when the "NEW" is not present in the data ID Nos. or when it is present and the corresponding material ID No. is present somewhere in the data ID No. memory means 170, the operator is judged as unauthorized to use the editing apparatus for the production of the new book.

At the step S232, when the operator is not judged as authorized to use the editing apparatus for the production of the new book, the processing is terminated at the end of the step S204 similarly to the second embodiment.

When the operator is judged as authorized, the

processing is carried through the steps S206 to S209, similarly to the first procedure of editing in the second embodiment.

At the step S233, the decision is made as to whether or not the element data key fed in as the data ID No. at the step S209 is found among the group of data ID Nos. selected at the step S231.

When "A1234" is fed in as the security No. and "B12" as the element data key in the case mentioned above, for example, it is judged that "B12" is memorized in the group of data ID Nos. as the element data key corresponding to the security No. "A1234" as shown in Table a of Fig. 31.

Conversely, when "B14" is fed in as the element data key, for example, it is judged that this particular element data key is not memorized in the group of data ID Nos. corresponding to "A1234".

When the absence of the element data key from the group of data ID Nos. is confirmed, the processing is terminated at the end of the step S204. When the presence of the element data key is confirmed, the processing is carried through the steps S210 to S214, similarly to the first procedure of editing in the second embodiment.

Optionally, the procedure may be modified so that when the absence from the memory is confirmed at the step S233, the processing is returned to the step S207.

Now, the operation of the embodiment to be performed by the operator in the production of a new book in accordance with the second procedure of editing will be described below. Fig. 30 is a flow chart of the operation of the third embodiment applied to the second procedure of editing. In Fig. 30, the reference numerals which have equals in Fig. 27 and Fig. 29 denote identical or similar parts. The explanation of these parts, therefore, is omitted here.

In the production of a new book by the second procedure of editing, substantially the same processing as that of the second procedure of editing in the second embodiment is carried out as illustrated in Fig. 28 and Fig. 30. Similarly to the first procedure of editing in the third embodiment illustrated in Fig. 29, after the judgment of the step S230, the group of data ID Nos. corresponding to the security No. are selected at the step S231 and the judgment as to the introduced data ID No. is carried out at the step S232 and the step S233.

It is clear from the description given above that when the secret level of the data ID No. "B10" is higher than that of "B12", "B15", or "B16", namely when the books controlled by "B15" are more important in Fig. 31, the operator whose security No. is "A1234" is authorized to edit or revise all the books controlled by the four data ID Nos., "B10" to "B16" and produce a new book as illustrated in Table a of Fig. 31. The operator assigned the security No. of "A1235" is authorized only to edit the books controlled by the three data ID Nos. "B12", "B15", and "B16" as illustrated in Table b and is not authorized to produce a new book. The present embodiment, there-

fore, permits security control conforming to the secret level of the data.

The classification of data according to the secret level may be easily supplemented by the classification according to the contents of data to make the security control more effective.

Where the data ID Nos. "B10" and "B14" cover the data on illustrations and the other data ID Nos. that on documents, for example, the operator whose security No. is "A1236" is authorized to edit indiscriminately the data concerning the illustrations without reference to the secret level. Thus, the effective control is easily accomplished.

This arrangement proves to be advantageous when the editing system of this invention is used not exclusively by the employees of the owner's company but is used additionally by the employees of the related companies including contractors for the purpose of inspection of necessary data.

The security control involved in the use of the editing apparatus for the editing work has been described. Optionally, this system may be modified so that the security control is effected when the editing apparatus is started. As the result, that access to the system for the inspection of books and element data can be controlled.

Fig. 33 is a functional block diagram of a data processing device adapted to permit efficient and quick retrieval of accumulated data to be used for editing.

With reference to Fig. 33, the data processing device comprised n'th rank ID input means 200 for the admission of an n'th rank ID, data retrieval table memory means 202 for memorizing tables illustrated in Fig. 22, namely the (n+1)'th rank data (ID and title) corresponding to the n'th rank ID, data selection means 201 for selecting the (n+1)'th rank data from the data retrieval table memory means 202, display means 206, element data memory means 205, data existence discrimination means 203 for making a decision as to whether or not the (n+1)'th rank data are selected by the data selection means 201, namely the data are memorized in the data discrimination table memory means 202 and exhibiting the (n+1)'th rank data on the display means 206 when the presence of the data in the memory means 202 is confirmed or supplying the n'th rank ID to element data read (out) means 204 (which will be described specifically hereinafter) when the absence of the data from the memory means 202 is confirmed, element data read out means 204 for reading the element data corresponding to the n'th rank ID from the element data memory means 205 to supply to the display means 206 and information processing means 207, and information processing means 207 for performing data processing other than retrieval.

First, the data selection means 201 decodes the n'th rank ID fed in through the n'th rank ID input means 200 and calls out of the data retrieval table memory means 202 the (n+1)'th rank data corresponding to the input n'th rank ID, namely the table comprising of the (n+1)'th

rank ID including the n'th rank ID in the ID thereof and the title corresponding to the (n+1)'th rank ID as shown in Figs. 22 and 23.

When the second rank ID "AA" is fed in as an n'th rank ID through the n'th rank ID input means 200, for example, the data selection means 201 reads the table d, shown in Fig. 22, which comprises the third rank ID's "AAA", "AAB", "AAC",... each including the second rank ID "AA" and titles corresponding one to one to said third rank ID's out of the data retrieval memory means 202 as (n+1)'th rank data.

The data existence discrimination means 203 performs the judgment as to whether or not the (n+1)'th rank data are present in the data retrieval table memory means 202.

In other words, the data existence discrimination means 203 performs the judgment as to whether or not the data selection means 201 has called out the (n+1)'th rank ID and the titles corresponding to the (n+1)'th rank ID. When the third rank ID, namely the (n+1)'th rank ID, is called out, the corresponding table d is exhibited on the display means 206.

Conversely when the absence of the ID is confirmed, namely when the absence of the third rank ID is confirmed, the second rank ID which is an n'th rank ID in this case is fed out to the element data read means 204.

The element data read means 204 decodes the n'th rank ID fed out of the data existence discrimination means 203, reads the element data corresponding to the n'th rank ID from the element data memory means 205 to supply to the display means 206 and the information processing means 207.

When the third rank ID which is an (n+1)'th rank ID in this case is absent, namely when the data constituting the table d illustrated in Fig. 22 are absent from the data retrieval table memory means 202, for example, the data existence discrimination means 203 supplies "AA" as the second rank ID to the element data read means 204.

The element data read means 204 decodes the second rank ID "AA", reads out of the element data memory means 205 such element data of sentences, illustrations, etc. as defined by the title "explanation" corresponding to the second rank ID "AA", and supplies the element data to the display means 206 and the information processing means 207.

As described above, the present embodiment permits the data retrieval to be carried out efficiently because the data retrieval can be started by the introduction of a lower ID than the first rank ID.

Now, another version of the data-processing apparatus which allowed further enhancement of the efficiency of data retrieval will be described below with reference to Fig. 2, Fig. 34, and Fig. 35.

Fig. 35 is a flow chart of another typical system of data retrieval according to the present invention. In Fig. 35, the steps with S-numerals which have equals in Fig. 24 denote identical or similar steps of processing.

The n'th rank ID is fed in through the personal computer 4-1, for example, at the step S151. At the step S152, the (n+1)'th rank data including the n'th rank ID in the ID thereof are selected from the data retrieval table by the host computer.

At the step S153, the decision is made as to whether or not the (n+1)'th rank data are present, namely whether or not the (n+1)'th rank data are selected at the step S152, by the host computer 1.

When the presence of the data is confirmed at the step S152, the processing moves to the step S157. When the absence of the data is confirmed, the element data corresponding to the n'th rank ID are called out of the document D/B 12 via the host computer 1 and exhibited on the personal computer 4-1 at the step S156.

At the step S157, the decision is made as to whether or not a plurality of sets each comprising of the (n+1)'th rank ID and the title corresponding to the (n+1)'th rank ID are present in the (n+1)'th rank data.

When the presence of the plurality of the sets mentioned above is conformed, the processing moves to the step S154. When the absence of the plurality of sets is confirmed, the processing moves to the step S158.

At the step S158, the "element data" part of the only one set of data corresponding to the (n+1)'th rank ID is called out of the document D/B 12 via the host computer 1 and exhibited on the personal computer 4-1.

At the step S154, all of the (n+1)'th rank data are displayed. At the step S155, n is increased by 1. Then the processing returns to the step S151.

Now, the procedure for data retrieval will be described more specifically with reference to Fig. 34.

Fig. 34 is a functional block diagram of still another embodiment of the present invention. In Fig. 34, the reference numerals which have equals in Fig. 1 denote identical or similar parts.

The data processing apparatus comprises:

- (A) n'th rank ID input means 200 for introducing an n'th rank ID,
- (B) data retrieval quality table memory means 202,
- (C) data selection means 201 for selecting the (n+1)'th rank data from the data retrieval quality table memory means 202,
- (D) display means 206,
- (E) element data memory means 205,
- (F) data existence discrimination means 203 for making the decision as to whether or not the (n+1)'th rank data are selected by the data selection means 201, namely whether or not the data are present in said data retrieval table memory means 202 and supplying said (n+1)'th rank data to data number discrimination means 208 (being described specifically hereinafter) when the presence of the data is confirmed or supplying the n'th rank ID to first element data read means 209 (being described specifically hereinafter) when the absence of the data is confirmed,

(G) first element data read means 209 for reading element data corresponding to the n 'th rank ID from the element data memory means 205 to supply to display means 206 and information processing means 207,

(H) data number discrimination means 208 for making the decision as to whether or not a plurality of sets of data are present in the $(n+1)$ 'th rank data fed out of the data existence discrimination means 203 and exhibiting the $(n+1)$ 'th rank data on display means 206 when the presence of the plurality of sets of data is confirmed or supplying the $(n+1)$ 'th rank ID in only one set of data to second element data read means 210 (being described specifically hereinafter) when the absence of the plurality of sets of data is confirmed,

(I) second element data read means 210 for reading element data corresponding to the $(n+1)$ 'th rank ID from the element data memory means 205 to supply to the display means 206 and the information processing means 207, and

(J) information processing means 207 for performing information processing.

The first element data read means 209 has the same function as the element data read means 204 illustrated in Fig. 33.

In Fig. 34, the data selection means 201 decodes the n 'th rank ID fed in through the n 'th rank ID input means 200 and calls out of the data retrieval table memory means 202 the $(n+1)$ 'th rank data corresponding to the n 'th rank ID, namely, a table composed of the $(n+1)$ 'th rank ID including the n 'th rank ID in the ID thereof and the title corresponding to the $(n+1)$ 'th rank ID as illustrated in Fig. 22 and Fig. 23.

When the second rank ID "AA" is fed in as an n 'th rank ID through the ID input means 200 such as a keyboard, for example, the data selection means 201 reads out of the data retrieval table memory means 202 a table composed of the third rank ID's "AAA", "AAB", "AAC", ... including the second rank ID "AA" and the titles corresponding to the third rank ID's as illustrated in Table d of Fig. 22.

The data existence discrimination means 203 performs the decision as to whether or not the $(n+1)$ 'th rank data are present in the data retrieval table memory means 202.

To be more specific, the data existence discrimination means 203 performs the decision as to whether or not the table composed of the $(n+1)$ 'th rank ID and the titles corresponding one to one to the $(n+1)$ 'th rank ID is called out by the data selection means 201 and supplies the table called out, the table d in this case, to the data number discrimination means 208 when the presence of the calling of the third rank ID corresponding to the $(n+1)$ 'th rank ID as described above is confirmed.

Conversely when the absence of the third rank ID is confirmed, the second rank ID "AA" as an n 'th rank ID

is supplied to the first element data reading means 209.

The data number discrimination means 208 performs the decision as to whether or not a plurality sets of data are present in the $(n+1)$ 'th rank data fed out of the data existence discrimination means 203 and exhibits the $(n+1)$ 'th rank data on the display means 206 when the presence of the plurality of sets of data is confirmed or supplies the $(n+1)$ 'th rank ID to the second element data reading means 210 when the absence of the plurality sets of data is confirmed.

To be more specific, the data number discrimination means 208 performs the decision as to whether or not a plurality sets of data each composed of the $(n+1)$ 'th rank ID and the titles corresponding one to one to the $(n+1)$ 'th ID are present in the table d whose presence has been confirmed by the data existence discrimination means 203.

Since three sets of third rank ID's "AAA", "AAB", and "AAC" including the second rank ID "AA" are present when the second rank ID "AA" is fed in as the n 'th rank ID through the ID input means 200, in this example, the table d composed of the three sets of ID's and the titles corresponding to the three ID's is exhibited on the display means 206.

Conversely, when only one set "AAA" is present, the "AAA" is supplied to the second element data reading means 210.

The first element data reading means 209 decodes the second rank ID "AA" as an n 'th rank ID fed from the data existence discrimination means 203, calls out of the element data memory means 205 the element data corresponding to the second rank ID "AA" to supply to the display means 206 and the information processing means 207.

The second element data reading means 210 decodes the third rank ID "AAA" as an $(n+1)$ 'th rank ID fed from the data number discrimination means 208, calls out of the element data memory means 205 the element data corresponding to the third rank ID "AAA" to supply to the display means 206 and the information processing means 207.

The data processing apparatus of the present embodiment constructed as described above permits the data retrieval to be carried out more easily because the element data corresponding to the relevant ID are fed out of the element data memory means and displayed when the data selection means 201 selects only one set of $(n+1)$ 'th rank data.

Though the present embodiment has been depicted as applied to an editing apparatus, it is naturally applicable to various data processing apparatuses of the ordinary run.

As clearly noted from the description given above, the embodiment of Fig. 33 enables the data retrieval to be made at a desired rank by the introduction of the n 'th rank ID. In the case of the data whose index is perfectly remembered by the operator on account of high frequency of use, for example, a necessary part of the data

can be promptly retrieved by the injection of the index in its complete form. Where the index for the necessary part of the data is not completely known, the data can be retrieved from a desired rank by feeding in only the part of the index clearly known to the operator and then keeping under close inspection the titles on the display means.

Thus, the data processing apparatus allows the data retrieval to be carried out very efficiently when the index is perfectly known and even when the index is known only imperfectly.

Further in the case of the embodiment illustrated in Fig. 34 which is provided with the data number discrimination means, the first element data reading means, and the second element data reading means, the element data corresponding to the (n+1)'th rank ID are automatically displayed when there exists only one (n+1)'th rank data. The introduction of the (n+1)'th rank ID through the input means, therefore, is not required to be repeated for display. Thus, the data processing apparatus enjoys improved efficiency.

Industrial Applicability

As clear from the description given above, the first embodiment of the present invention attains the following effect because it enables the data of the book edited with the editing apparatus to be decomposed into data concerning layout and data concerning elements for storage in the D/B and allows data for one page to be composed by using the decomposed and memorized data.

When a new book to be produced and a book already edited with the editing apparatus have common documents, illustrations, etc, the production of the new book can be attained by utilizing such common data without requiring new introduction of the documents, illustrations, etc. Thus, the production of the new book can be accomplished simply and quickly.

In accordance with the second embodiment, the security control can be attained with high reliability because the protect function is manifested not only on such materials as books and pages which have been already edited but also on element data forming such materials. In accordance with the embodiment of the invention, the security control can be carried out more elaborately, depending on the position of the operator or the contents of data. Thus, the security control is attained with high efficiency.

The data processing apparatus permits the data retrieval to be carried out very efficiently when the index of data is perfectly known or even when the index is known imperfectly. The application of the data processing apparatus of this invention to the data retrieval from the data base in the editing apparatus, therefore, warrants highly efficient editing of a new book.

Claims

1. An editing apparatus comprising editing means (160) for editing books, editing information input means (172) for introducing information necessary for the performance of editing, data memory means (169) for memorizing data for editing, and display means (168), which editing apparatus is provided with

data ID number memory means (170) for memorizing the security numbers of operators and of data ID number groups corresponding to each particular security number, which data ID number groups indicate books which the operator having the particular security number is authorized to edit,

security number input means (166) for introducing the security numbers,

security number discrimination means (164) for performing the decision as to whether or not the security number introduced through the security number input means (166) is memorized in the data ID number memory means as a security number corresponding to a data ID number group,

data ID number input means (173) for introducing the data ID number of a book to be edited, and

data ID number discrimination means (171) for performing the decision as to whether or not the data ID number introduced through the data ID number input means (173) is registered in the data ID number memory means (170) as being of the data ID number group corresponding to the security number introduced through the security number input means (166),

and the editing means is adapted so that the editing operation thereof is controlled according to the result of the decision performed by the security number discrimination means (164) and the data reading out of the data memory means (169) is controlled depending on the result of the decision performed by the data ID number discrimination means (171).

2. An editing apparatus according to claim 1, characterized by a specific predetermined data ID number (NEW) being assigned to the security number of the operator authorized to produce a new book.

Patentansprüche

1. Eine Redigiervorrichtung, umfassend eine Redigiereinrichtung (160) zum Redigieren von Büchern, eine Redigierinformations-Eingabevorrichtung (172) zur Eingabe von für die Durchführung des Redigierens notwendigen Informationen, eine Datenspeichervorrichtung (169) zur Speicherung von Daten für das Redigieren und eine Anzeigevorrichtung (168), wobei die Redigiervorrichtung ausgestattet ist mit:
 - einer Daten-ID-Nummer-Speichervorrichtung (170) zur Speicherung der Sicherheitsnummern von Bedienern und von Daten-ID-Nummern-Gruppen, welche jeder bestimmten Sicherheitsnummer entsprechen, wobei die Daten-ID-Nummer-Gruppen Bücher angeben, zu deren Redaktion der Bediener mit der bestimmten Sicherheitsnummer berechtigt ist,
 - einer Sicherheitsnummer-Eingabevorrichtung (166) zur Eingabe der Sicherheitsnummern,
 - einer Sicherheitsnummer-Unterscheidungsvorrichtung (164) zur Entscheidung, ob oder ob nicht die durch die Sicherheitsnummer-Eingabevorrichtung (166) eingegebene Sicherheitsnummer in der Daten-ID-Nummer-Speichervorrichtung als eine Sicherheitsnummer gespeichert ist, welche einer Daten-ID-Nummer-Gruppe entspricht,
 - eine Daten-ID-Nummer-Eingabevorrichtung (173) zur Eingabe der Daten-ID-Nummer eines zu redigierenden Buches, und
 - einer Daten-ID-Nummer-Unterscheidungsvorrichtung (171) zur Entscheidung, ob oder ob nicht die durch die Daten-ID-Nummer-Eingabevorrichtung (173) eingegebene Daten-ID-Nummer in der Daten-ID-Nummer-Speichervorrichtung (170) als aus der Daten-ID-Nummer-Gruppe registriert ist, welche der durch die Sicherheitsnummer-Eingabevorrichtung (166) eingegebenen Sicherheitsnummer entspricht,
 - und wobei die Redigiereinrichtung so ausgelegt ist, daß deren Redigierbetrieb gesteuert wird entsprechend dem Ergebnis der Entscheidung, welche durch die Sicherheitsnummer-Unterscheidungsvorrichtung (164) durchgeführt wird, und daß das Auslesen der Daten aus der Datenspeichervorrichtung (169) in Abhängigkeit von dem Ergebnis der durch die Daten-ID-Nummer-Unterscheidungsvorrichtung (171) durchgeführten Entscheidung gesteuert wird.

2. Redigiervorrichtung nach Anspruch 1, **gekennzeichnet** durch das Zuordnen einer spezifischen, vorbestimmten Daten-ID-Nummer (NEW) zur der Sicherheitsnummer des Bedieners, welcher zur Erzeugung eines neuen Buches berechtigt ist.

Revendications

1. Appareil d'édition comprenant un moyen d'édition (160) pour éditer des livres, un moyen d'entrée d'informations d'édition (172) pour introduire les informations nécessaires pour effectuer l'édition, un moyen de mémoire de données (169) pour mémoriser les données pour l'édition, et un moyen d'affichage (168), lequel appareil d'édition est muni de
 - un moyen de mémoire de numéro ID de données (170) pour mémoriser les numéros de sécurité des opérateurs et des groupes de numéros ID de données correspondant à chaque numéro de sécurité particulier, lesquels groupes de numéros ID de données indiquent des livres que l'opérateur ayant le numéro de sécurité particulier est autorisé à éditer,
 - un moyen d'entrée de numéro de sécurité (166) pour introduire les numéros de sécurité,
 - un moyen de discrimination de numéro de sécurité (164) pour décider si oui ou non le numéro de sécurité introduit par l'intermédiaire du moyen d'entrée du numéro de sécurité (166) est mémorisé dans le moyen de mémoire de numéros ID de données comme un numéro de sécurité correspondant à un groupe de numéros ID de données,
 - un moyen d'entrée de numéro ID de données (173) pour introduire le numéro ID de données d'un livre qui doit être édité, et
 - un moyen de discrimination de numéro ID de données (171) pour décider si oui ou non le numéro ID de données introduit par l'intermédiaire du moyen d'entrée du numéro ID de données (173) est enregistré dans le moyen de mémoire de numéro ID de données (170) comme étant du groupe de numéros ID de données correspondant au numéro de sécurité introduit par l'intermédiaire du moyen d'entrée de numéro de sécurité (166),
 - et le moyen d'édition est prévu d'une manière telle que son opération d'édition est contrôlée conformément au résultat de la décision prise par le moyen de discrimination de numéro de sécurité (164) et les données extraites du moyen de mémoire de données (169) sont contrôlées en fonction du résultat de la décision prise par le moyen de discrimination du numéro ID de données (171).

2. Appareil d'édition selon la revendication 1, caractérisé par un numéro ID de données prédéterminé spécifique (NOUVEAU) étant affecté au numéro de sécurité de l'opérateur autorisé à produire un nouveau livre.

5

10

15

20

25

30

35

40

45

50

55

27

FIG. 1

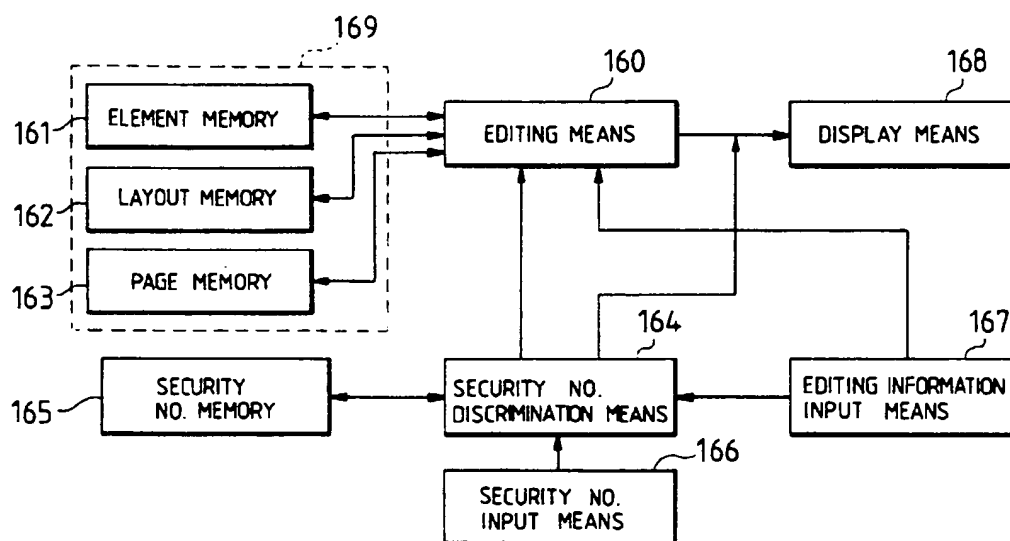


FIG. 2

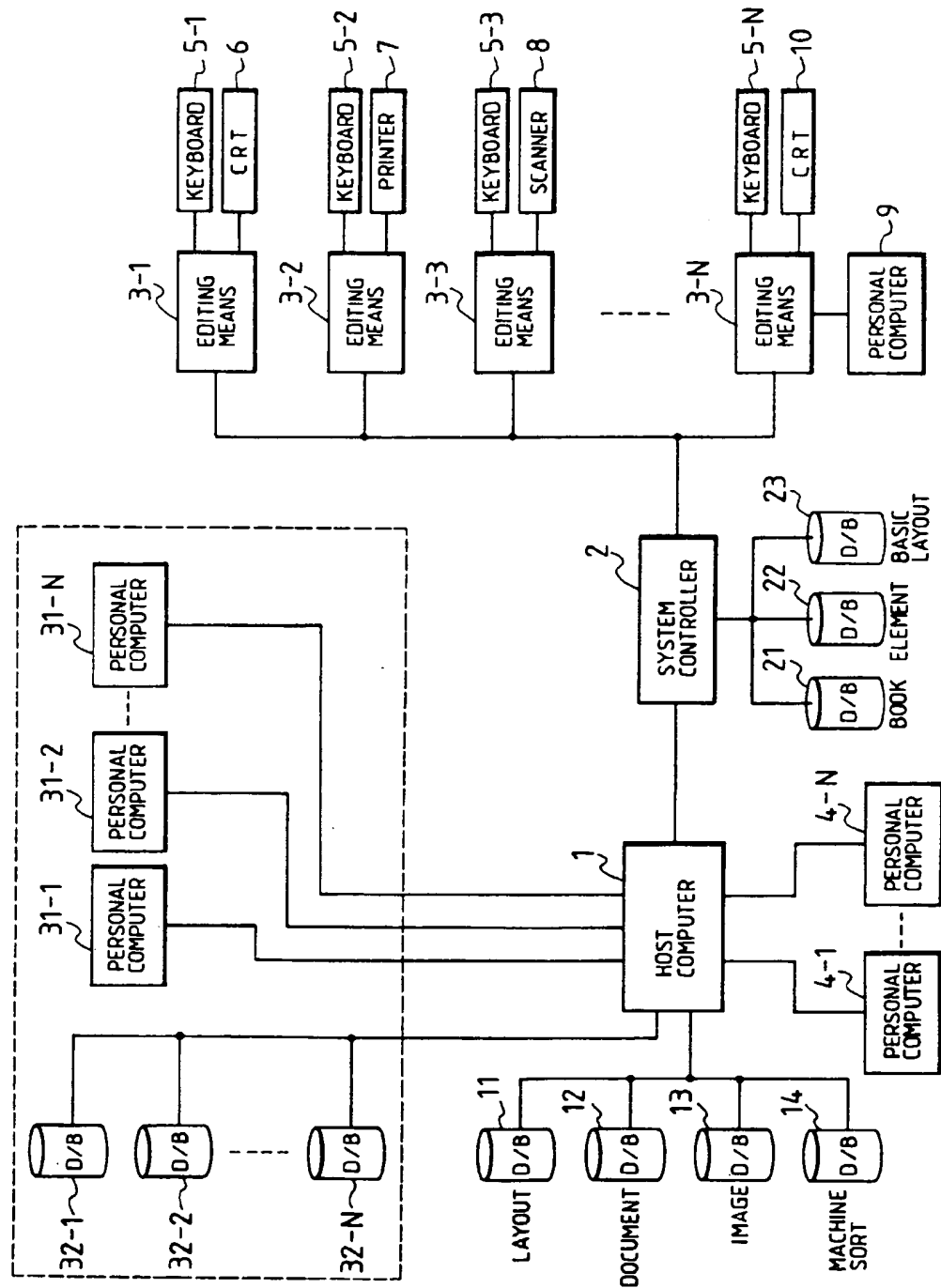


FIG. 3(1/5)

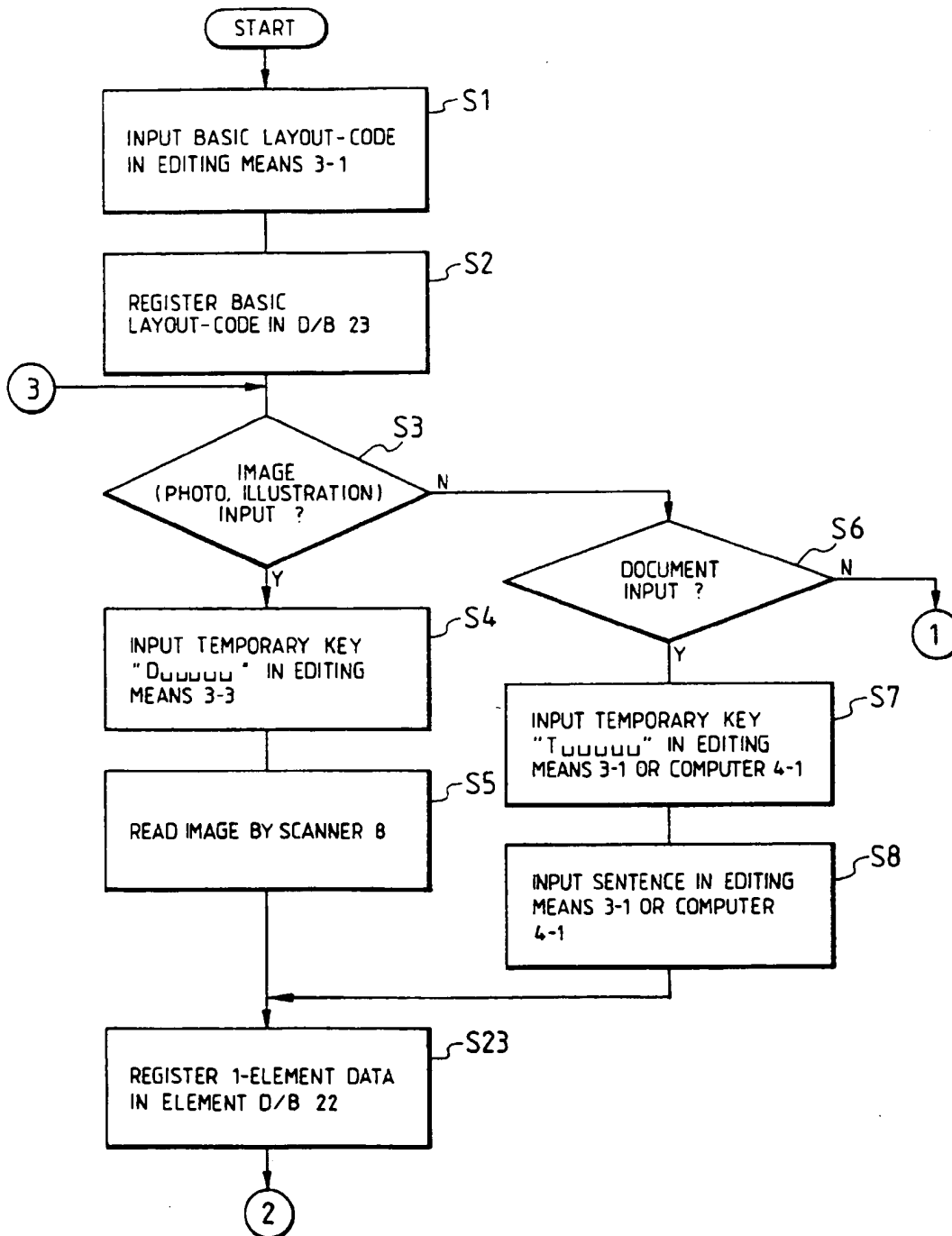


FIG. 3(2/5)

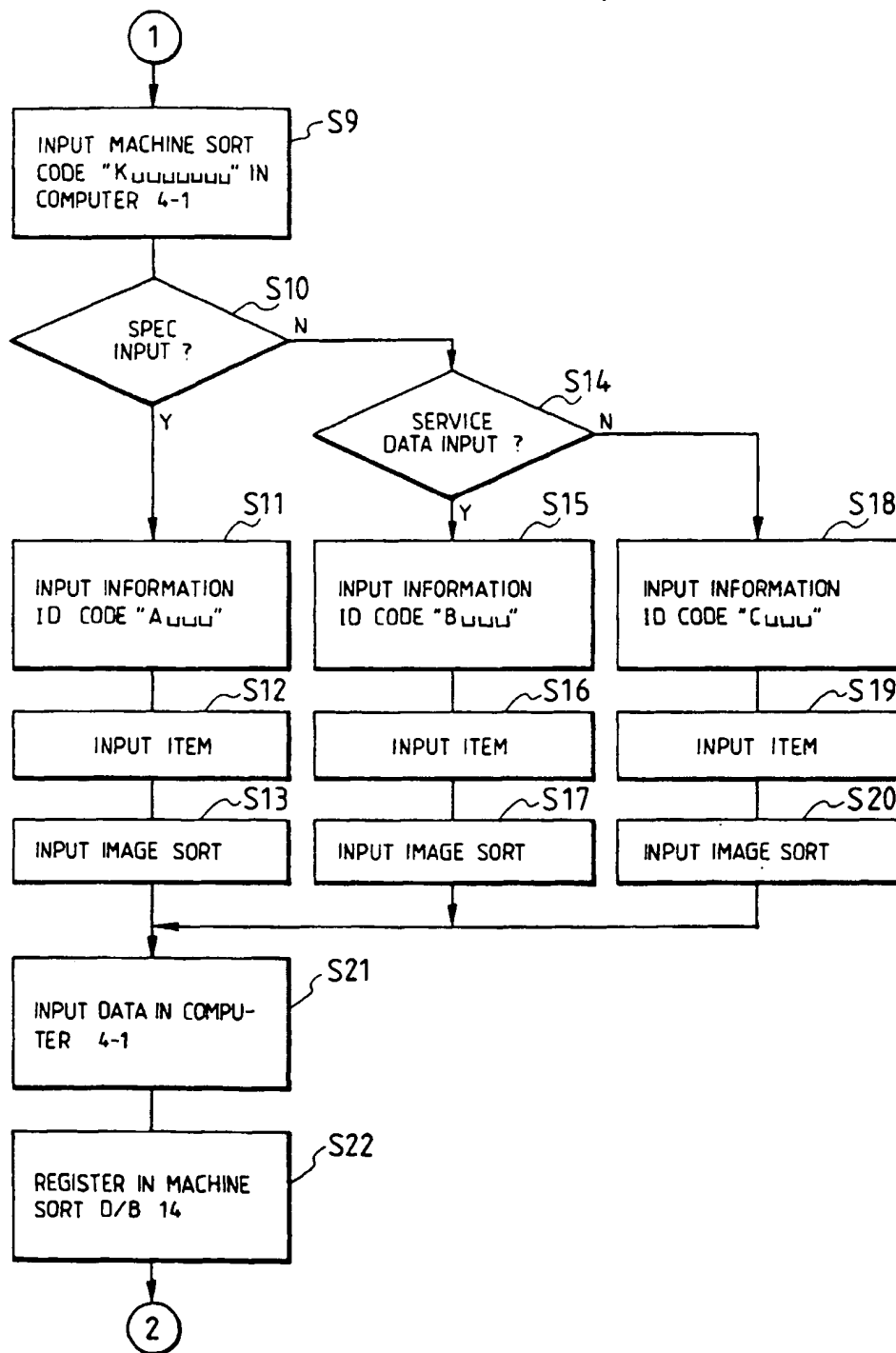


FIG. 3(3/5)

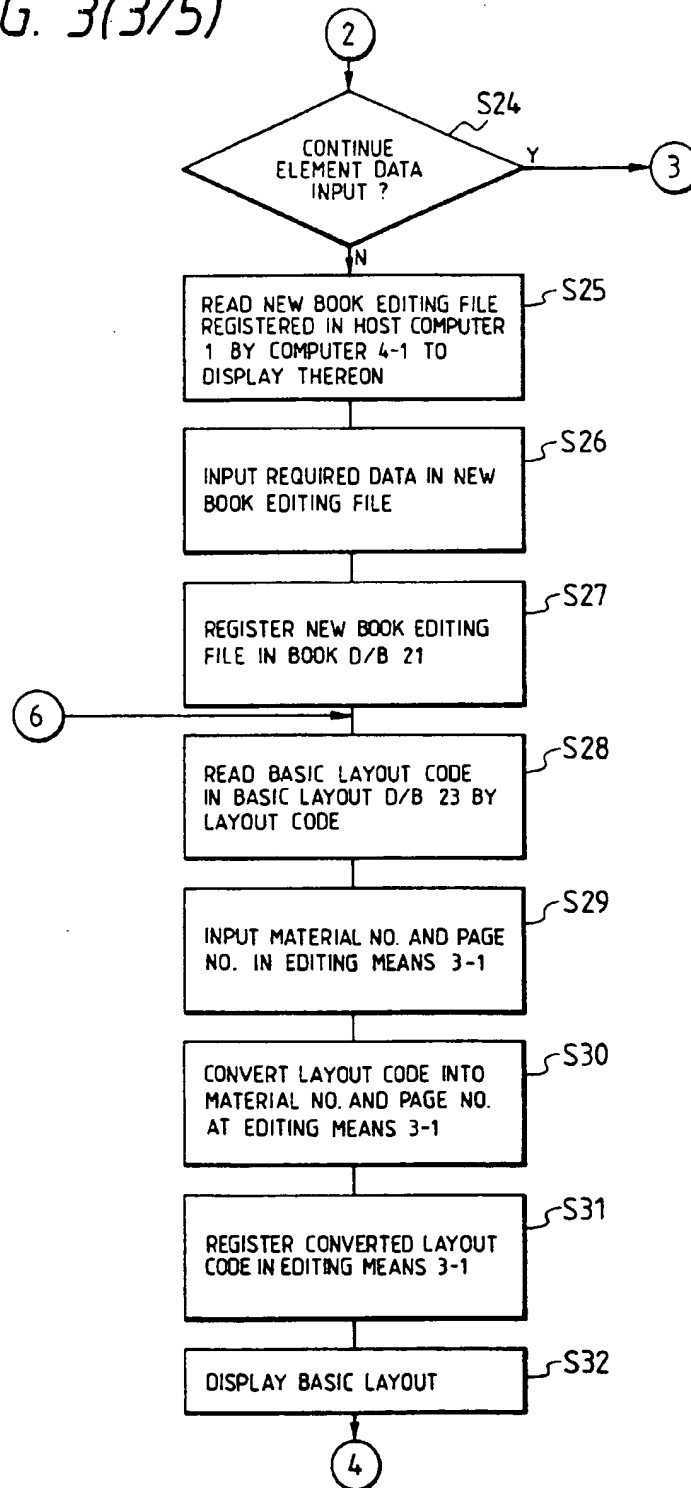


FIG. 3(4/5)

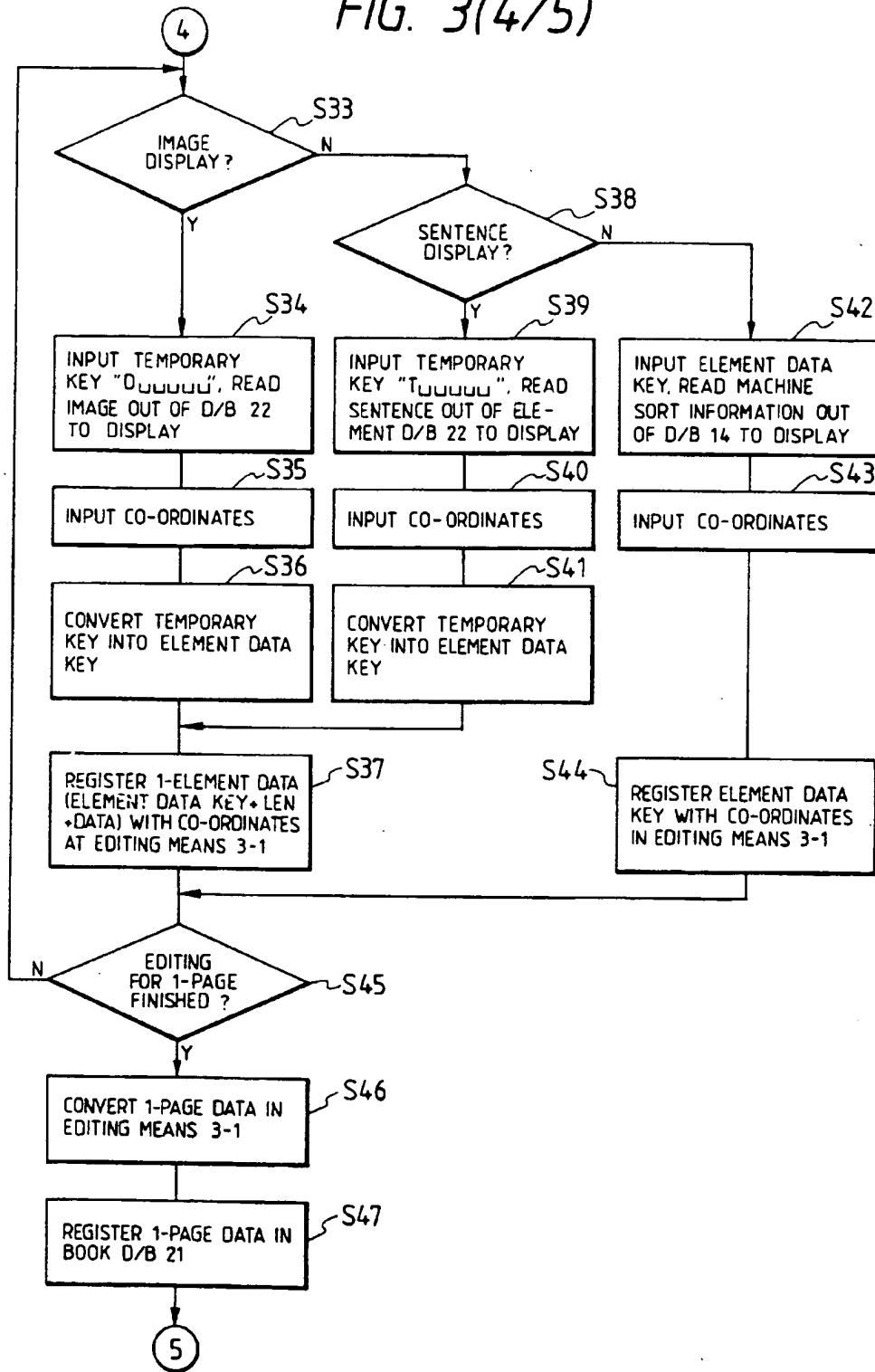


FIG. 3(5/5)

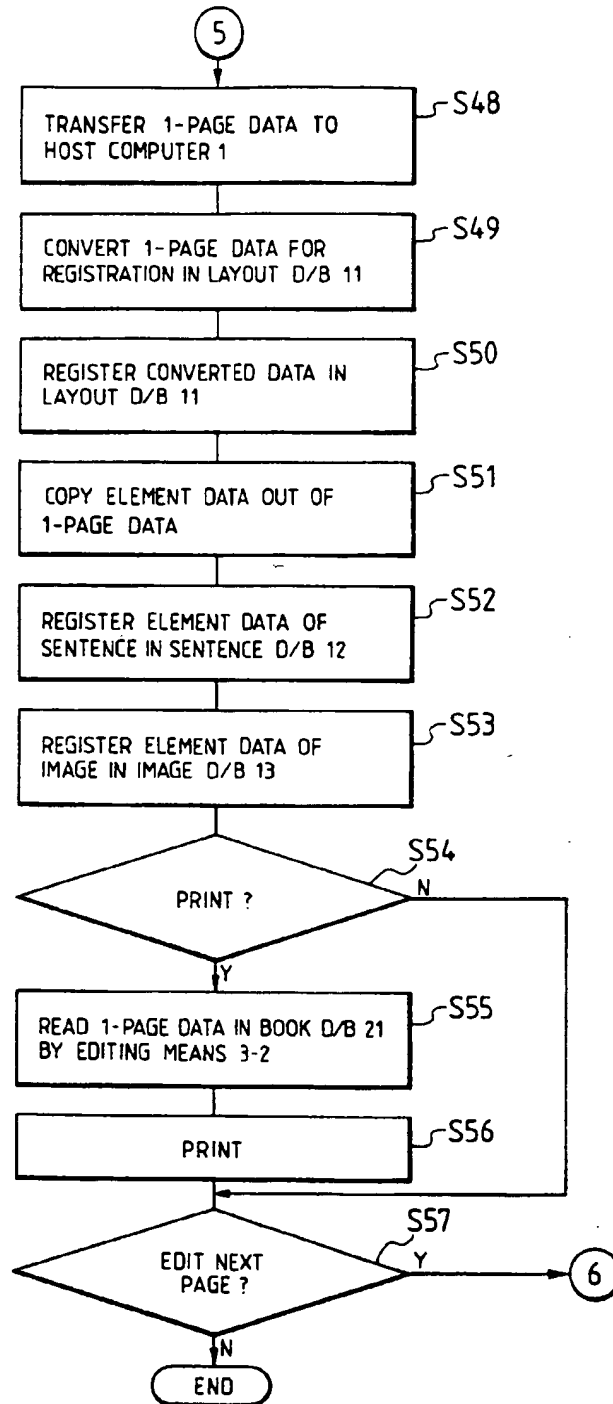


FIG. 4

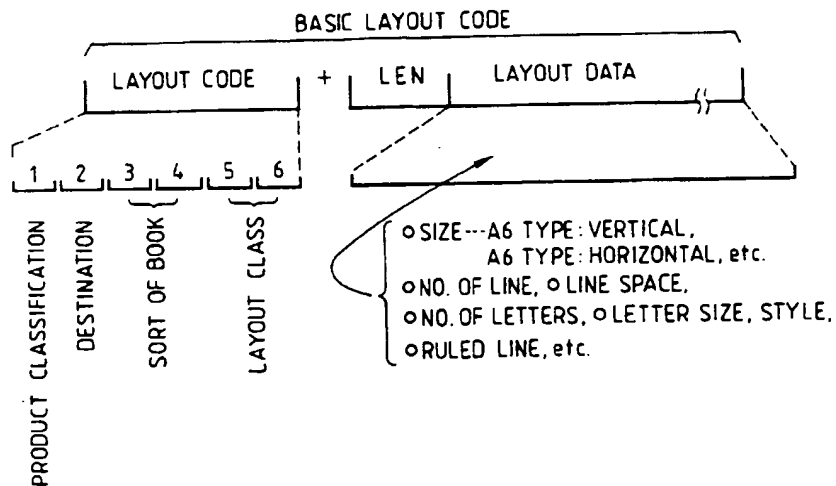


FIG. 5

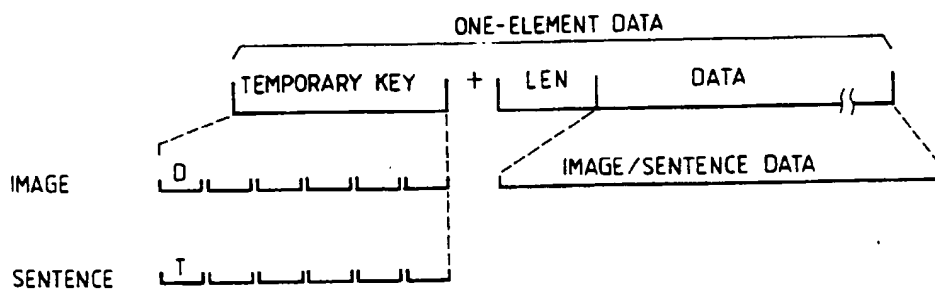


FIG. 6

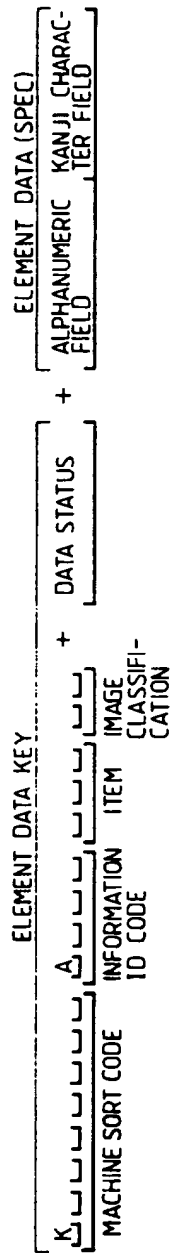


FIG. 7

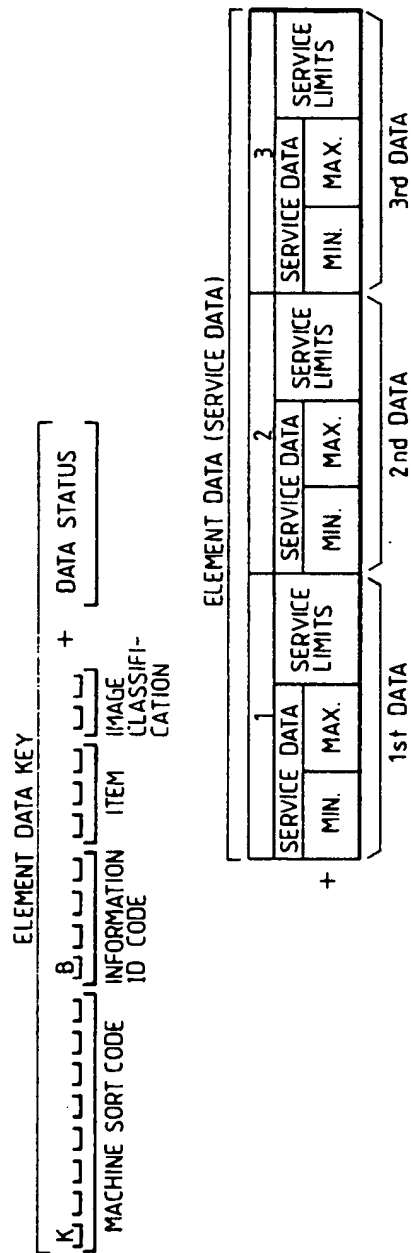
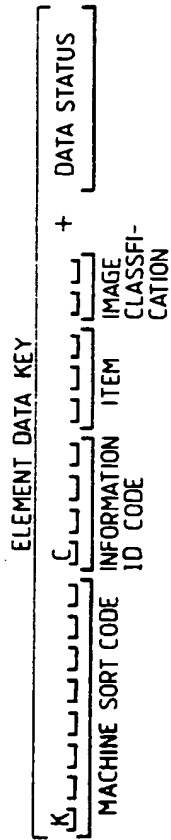


FIG. 8



ELEMENT DATA (TORQUE)

1				2				3			
RANGE		MIN.		MAX.		RANGE		MIN.		MAX.	
CENTER VALUE		DIA METER		NUMBER		CENTER VALUE		DIA METER		NUMBER	

1st DATA 2nd DATA 3rd DATA

+

FIG. 9

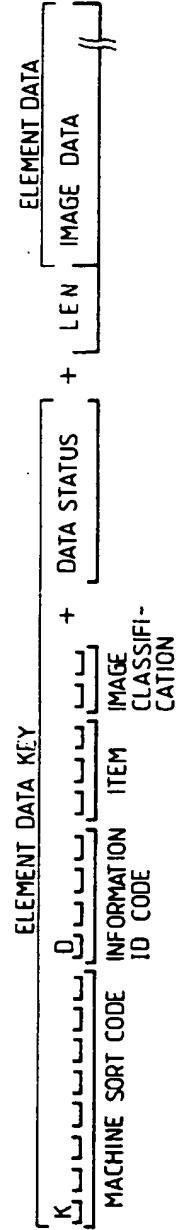


FIG. 10

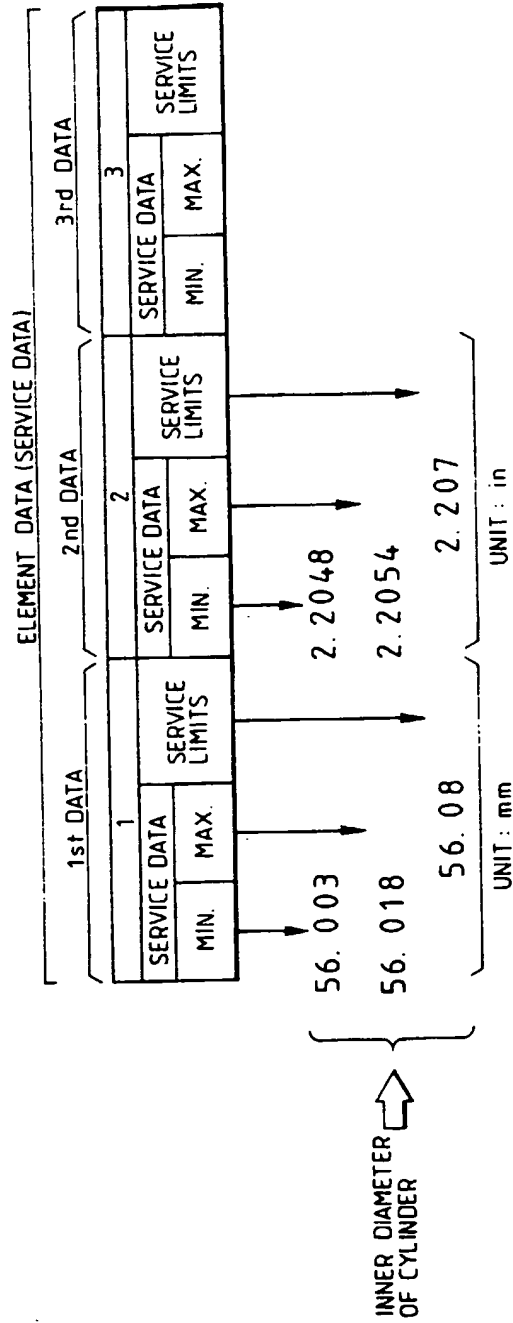


FIG. 11

FILE FOR NEW BOOK

MATERIAL ID NO.	STAFF- IN-CHARGE	STARTING PAGE NO.

FIG. 12

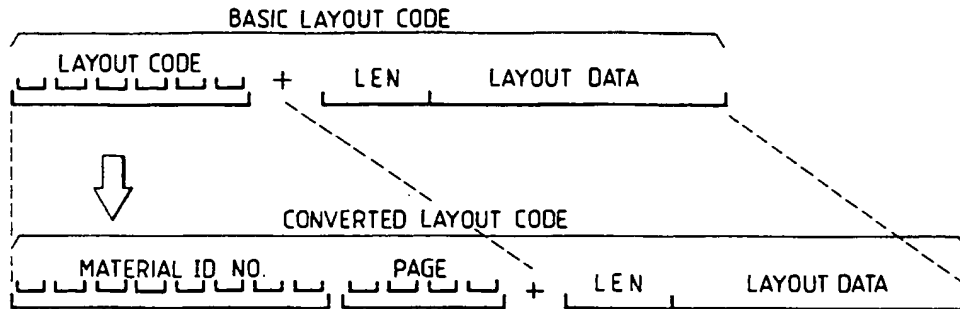


FIG. 13

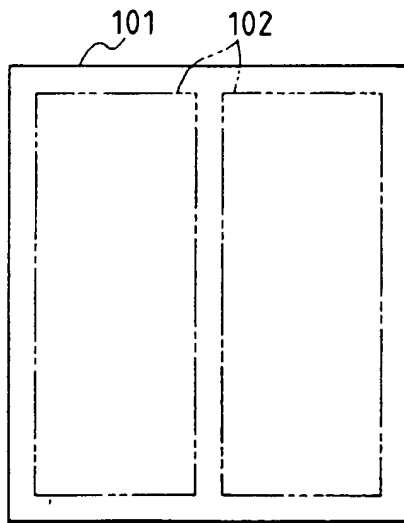


FIG. 14

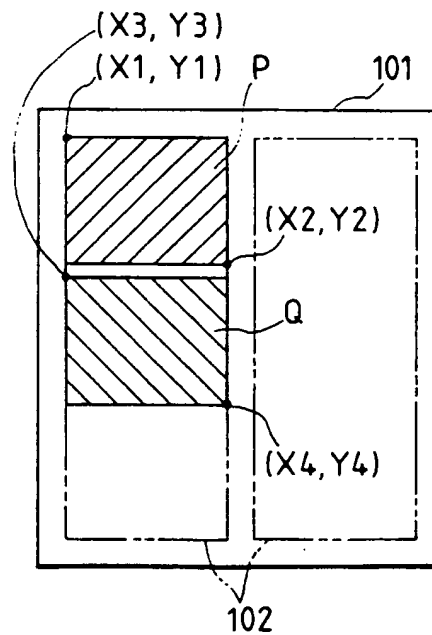


FIG. 15

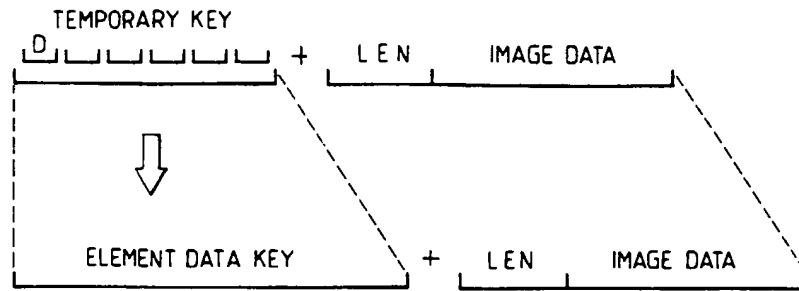


FIG. 16

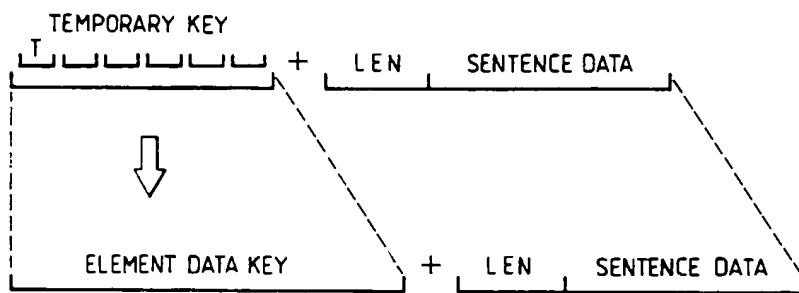


FIG. 23

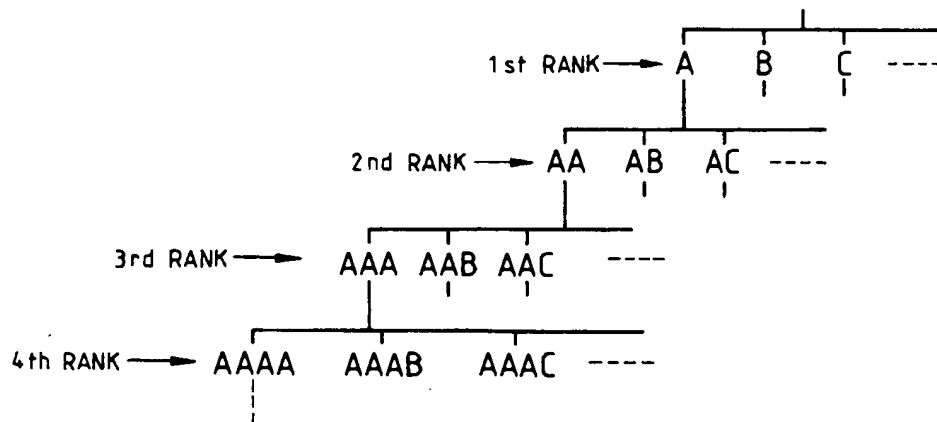


FIG. 17

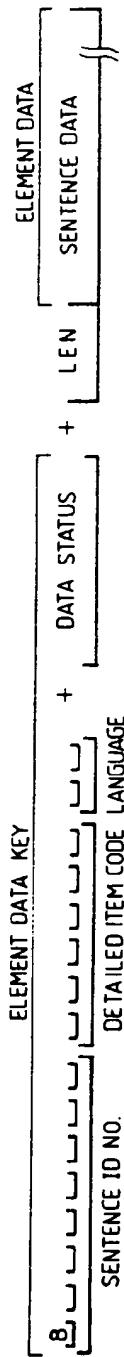


FIG. 18

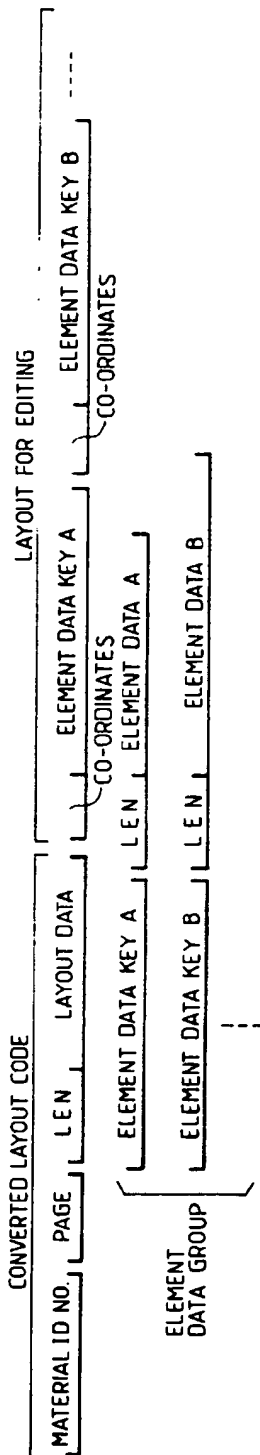


FIG. 19

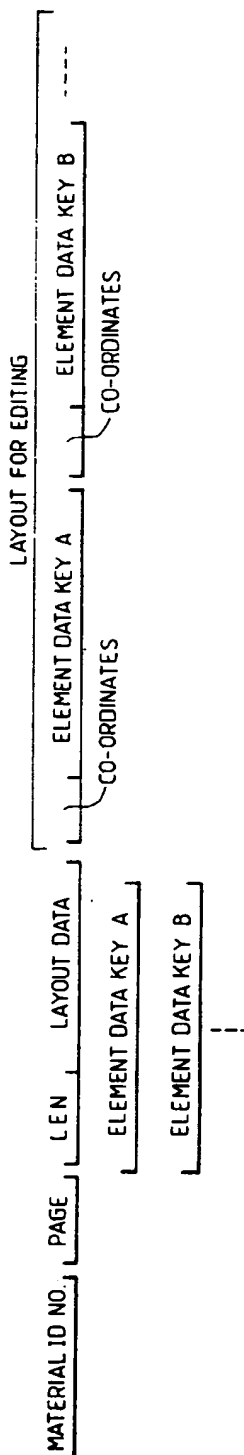


FIG. 20(1/6)

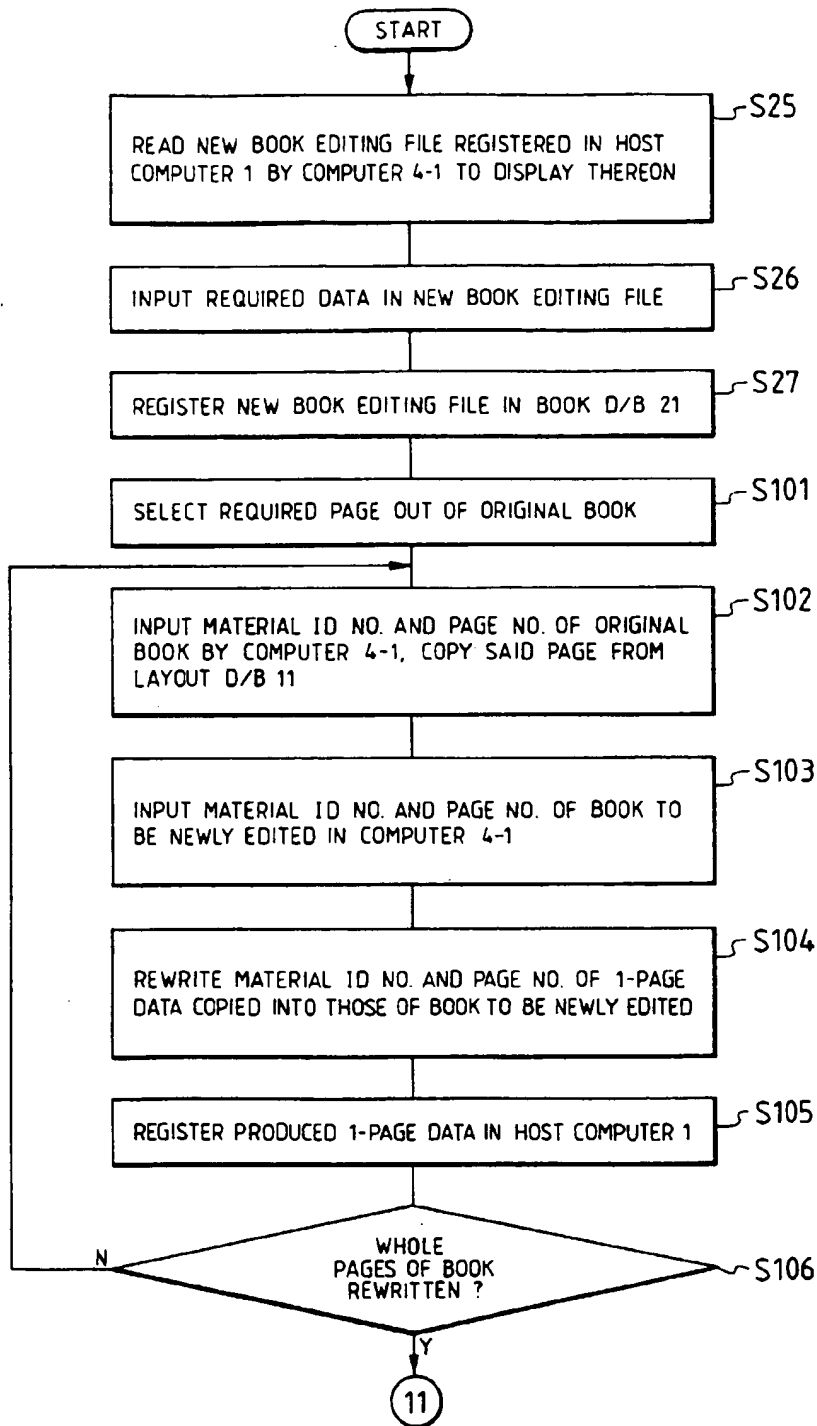


FIG. 20(2/6)

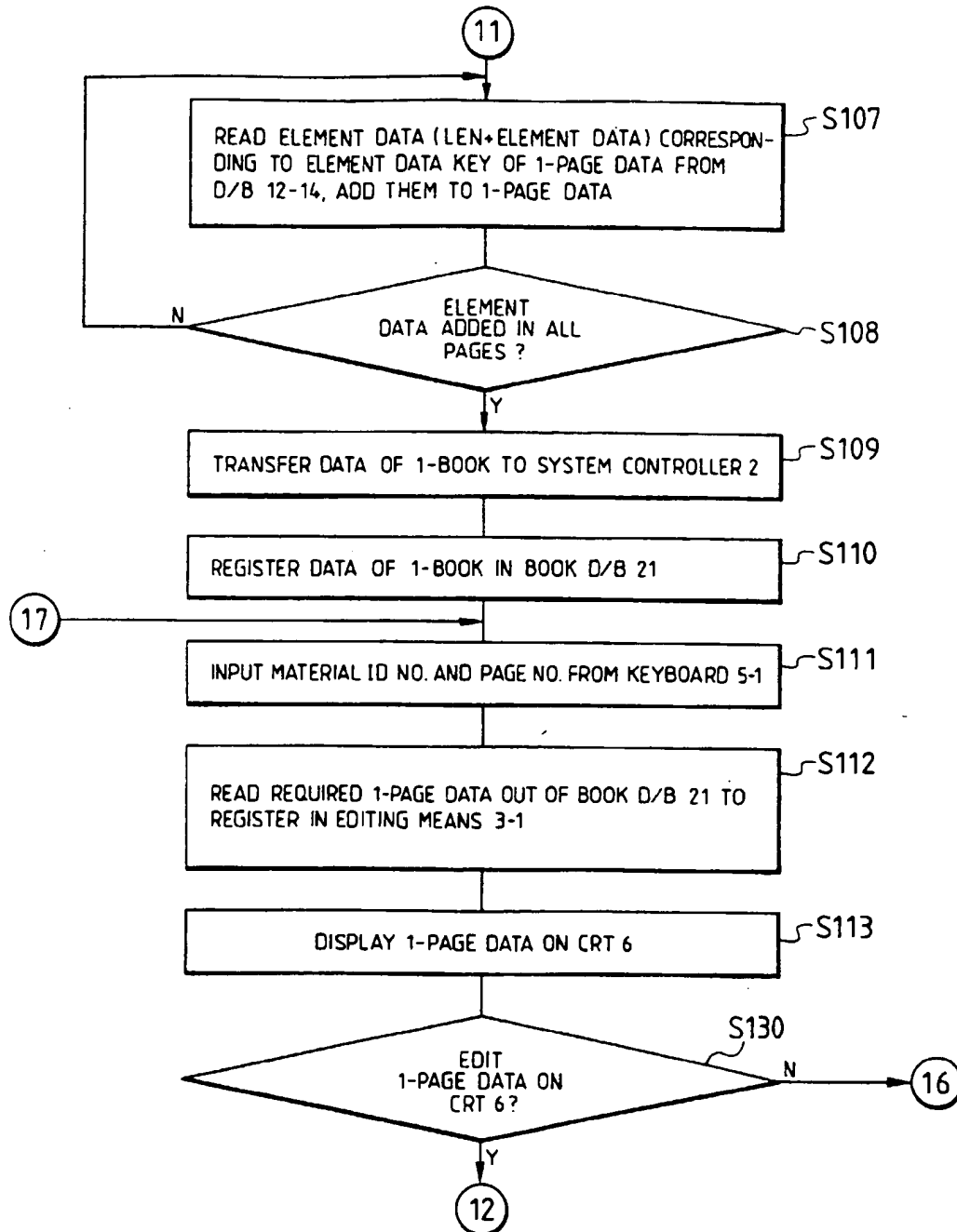


FIG. 20A

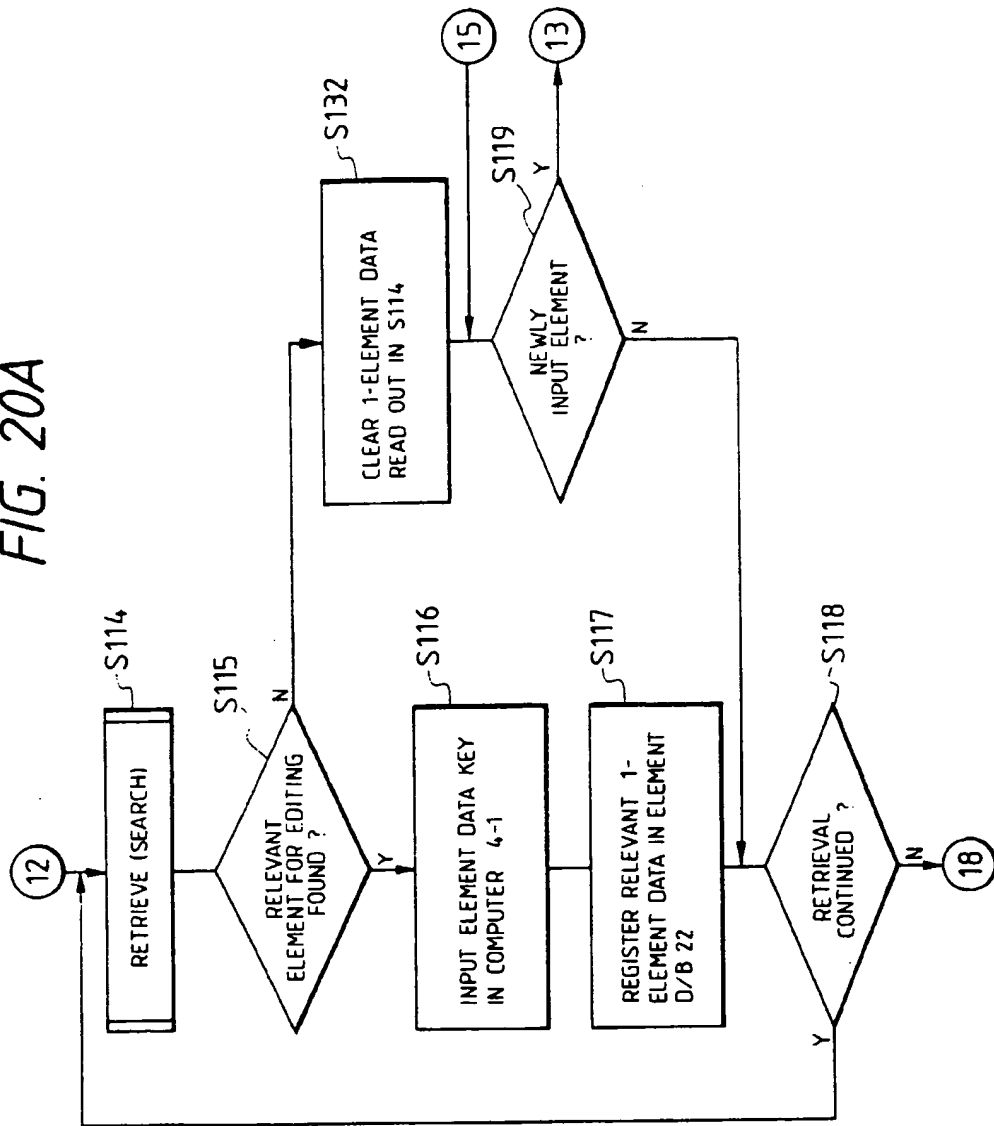


FIG. 20(3/6)

FIG. 20A
FIG. 20B

FIG. 20B

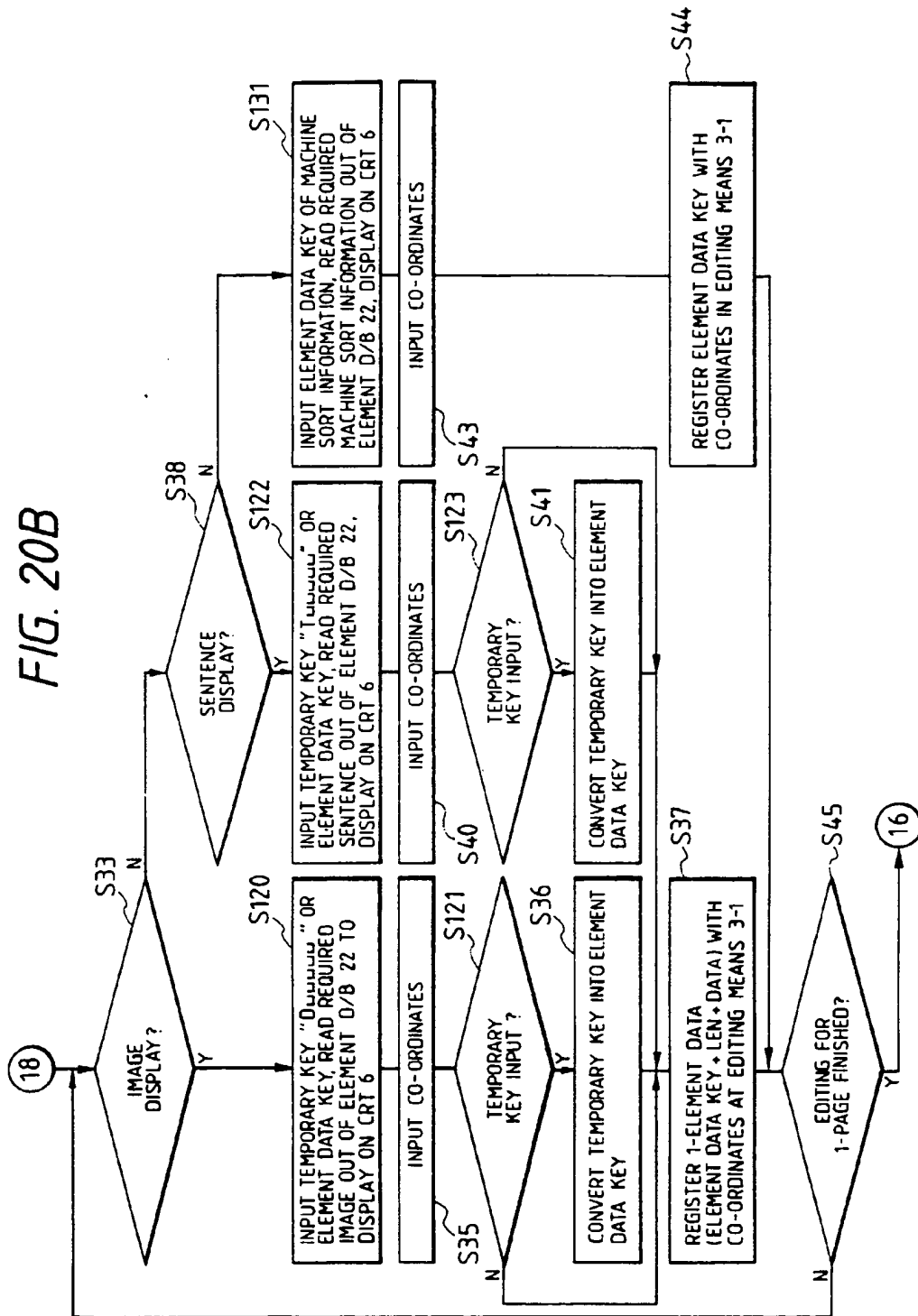


FIG. 20(4/6)

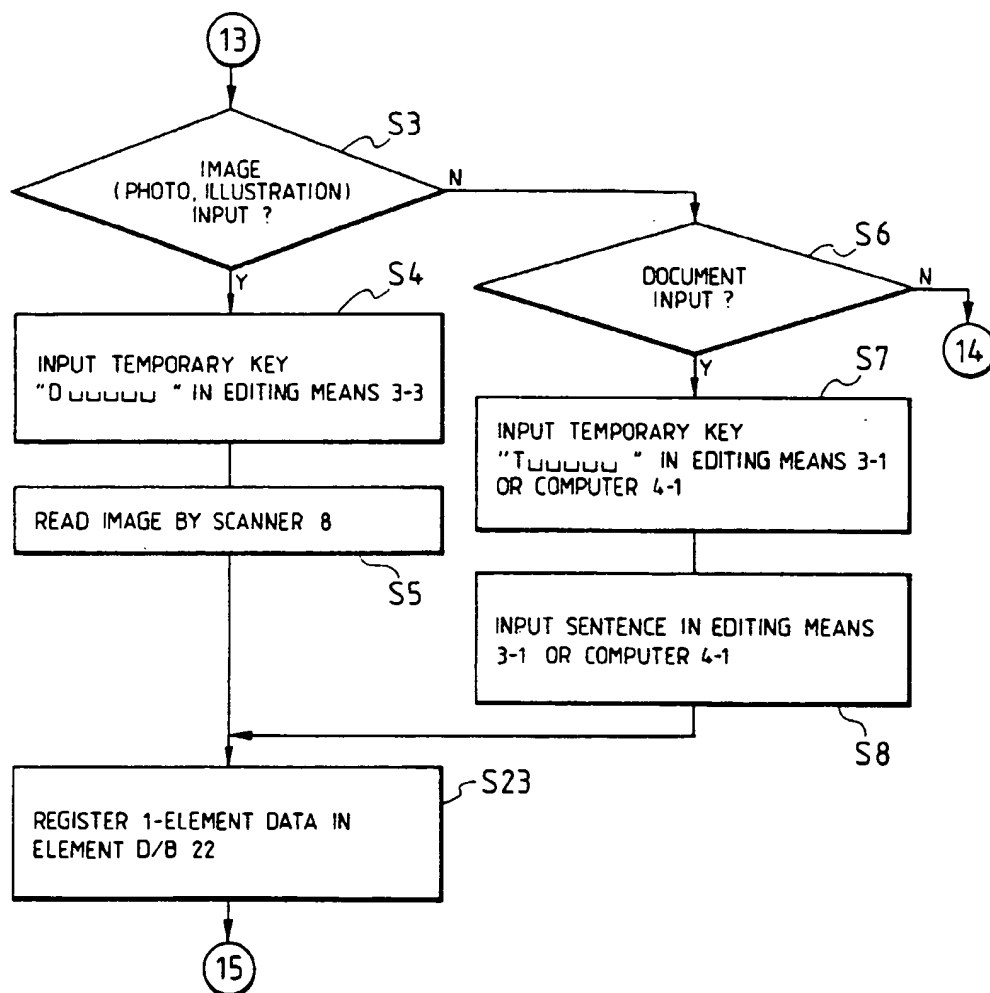


FIG. 20(5/6)

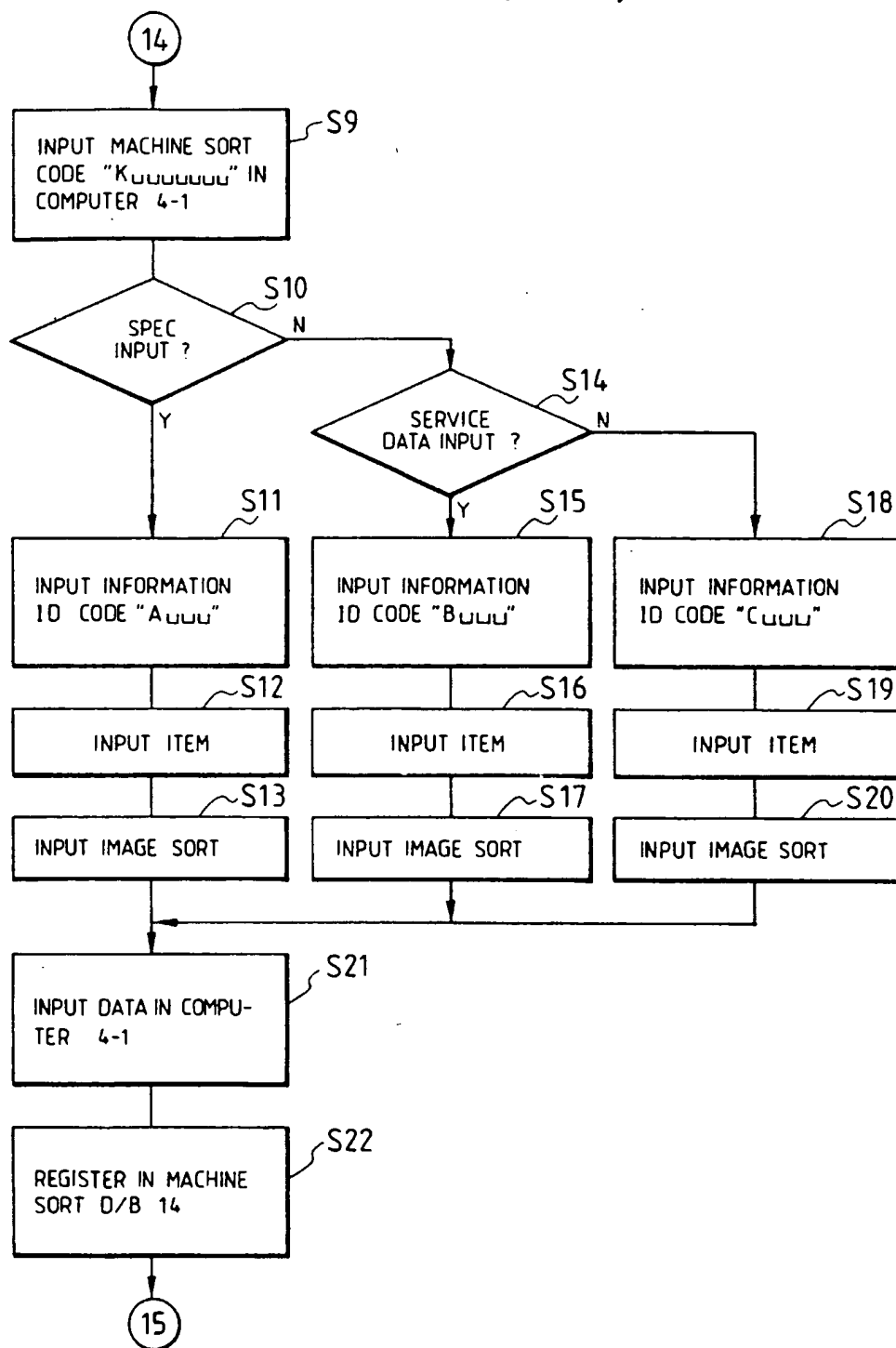


FIG. 20(6/6)

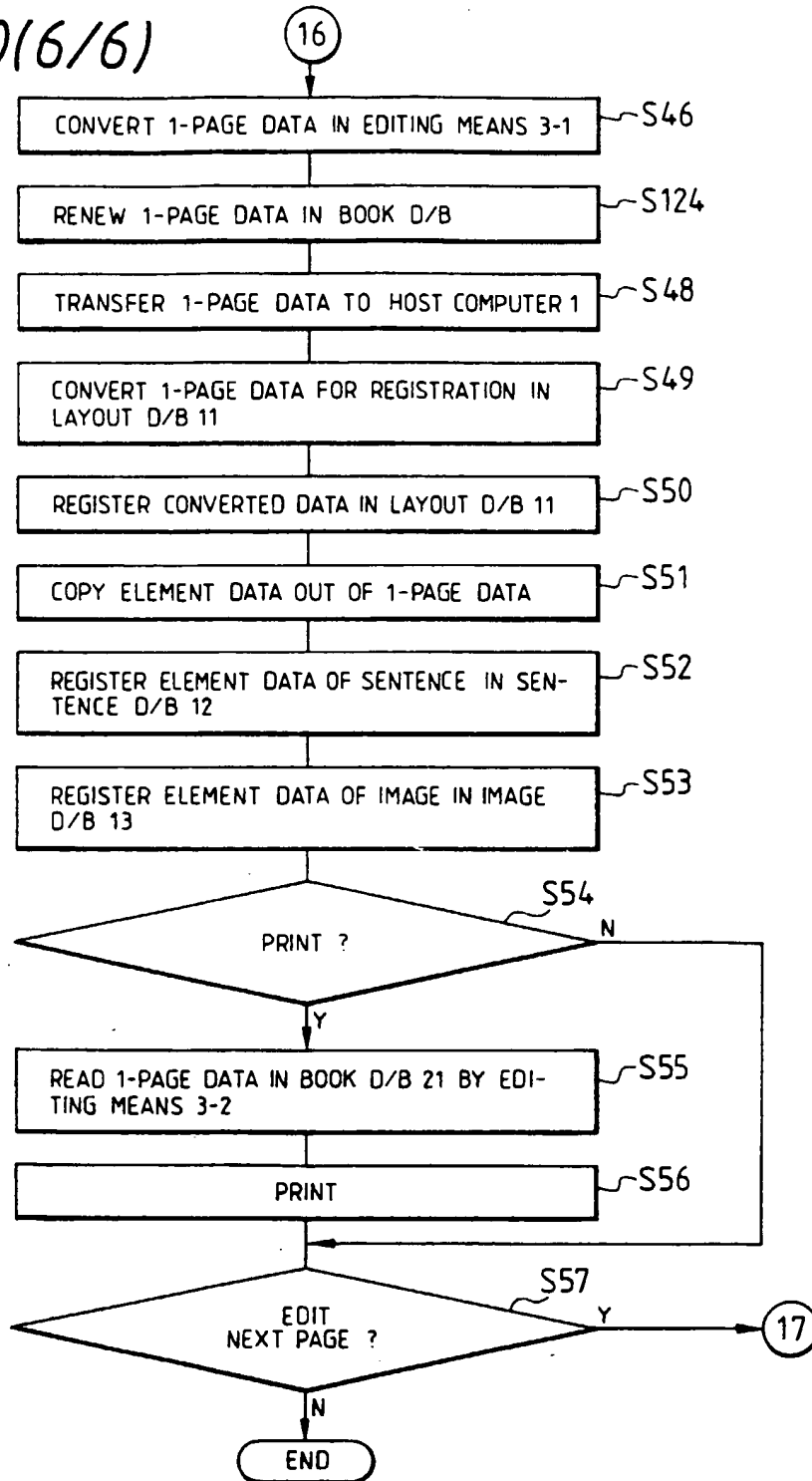


FIG. 21

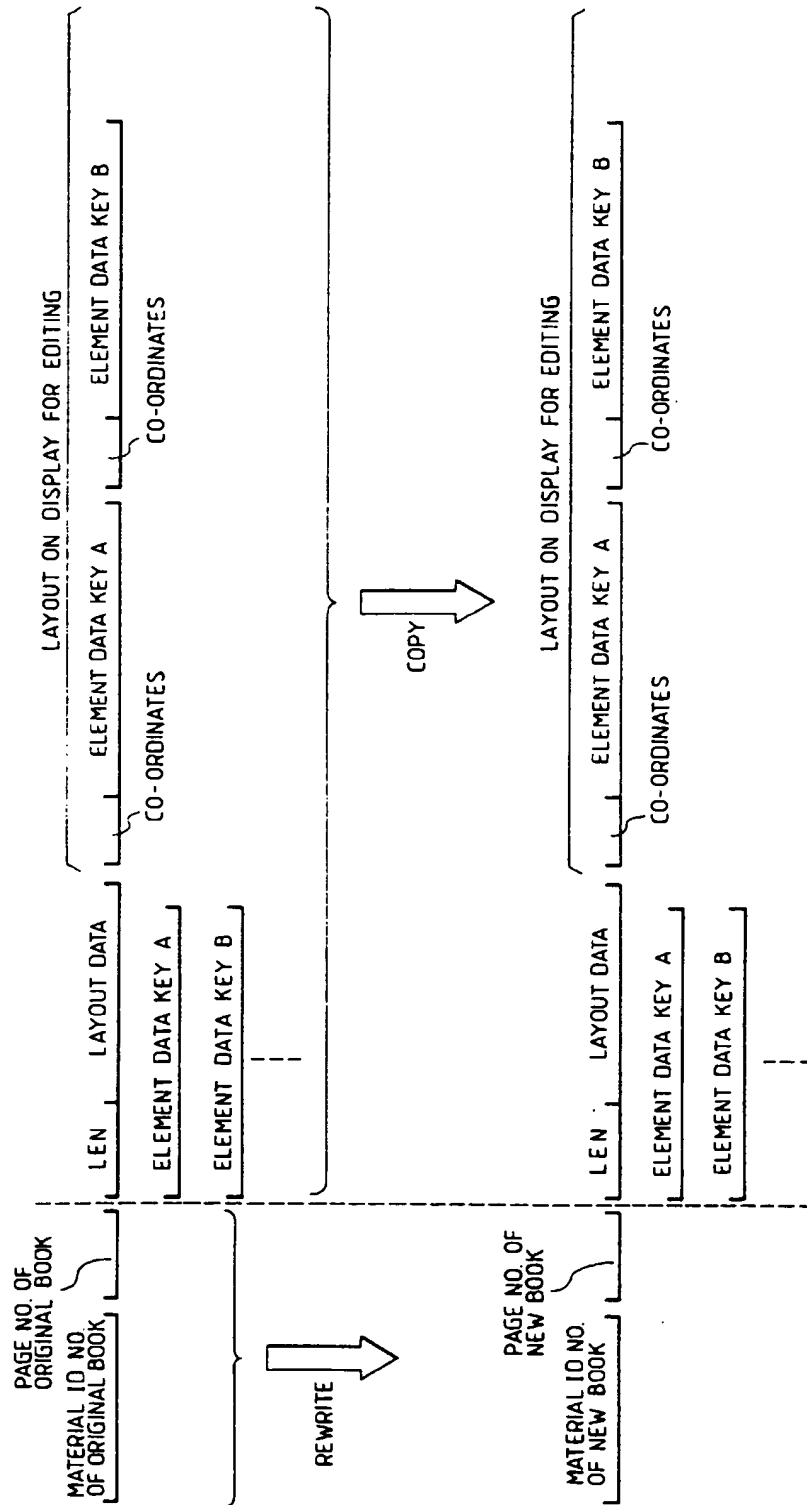


FIG. 22

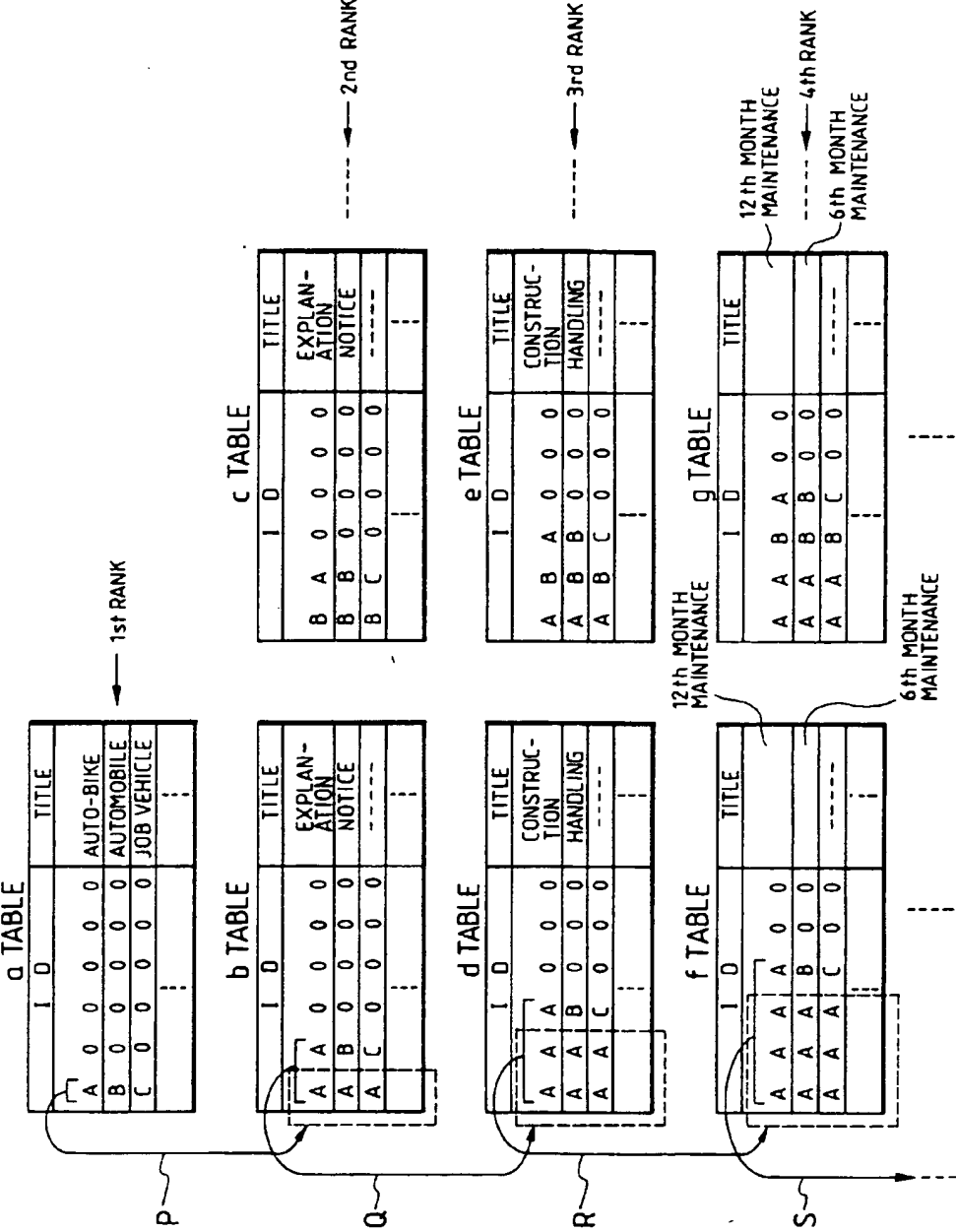


FIG. 24

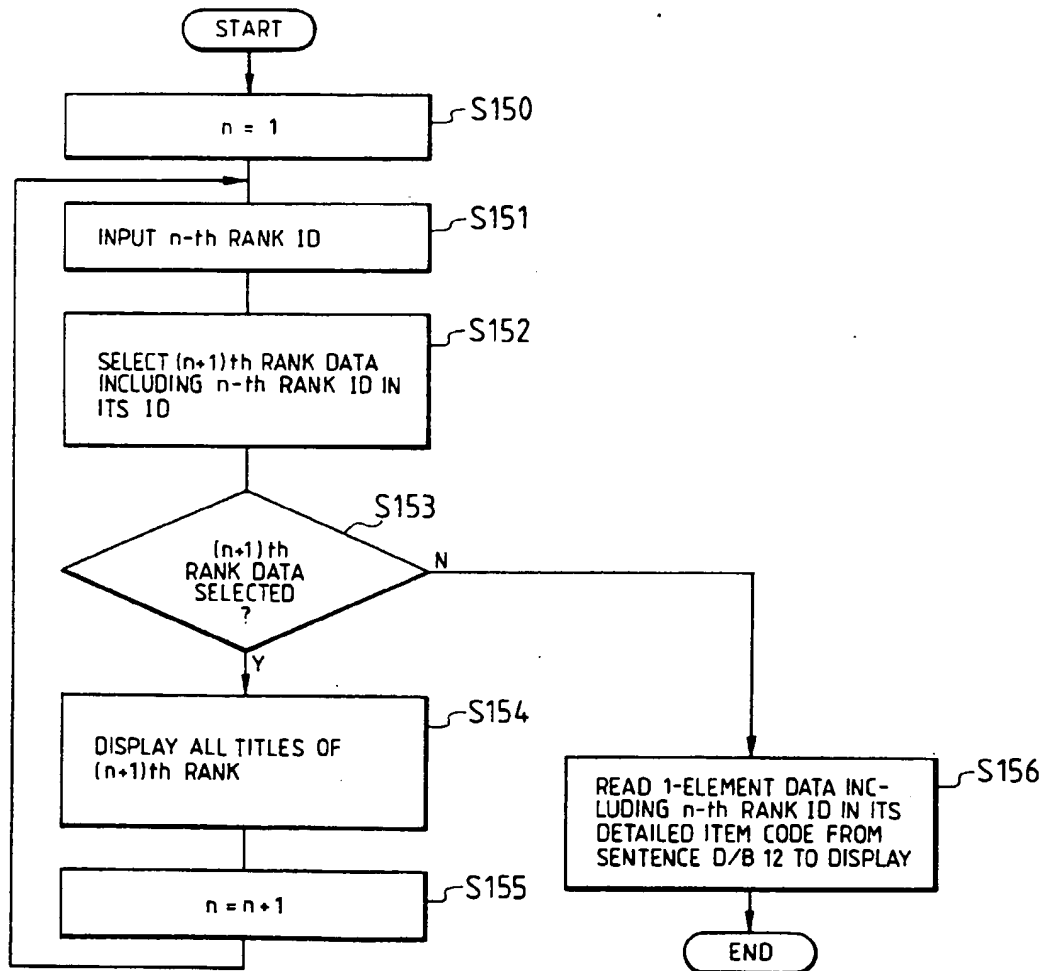


FIG. 25

CONVERSION CLASS	IMAGE CLASS	DISPLAY ON CRT
10	10	56. 003 - 56. 018 mm
	20	56. 003 - 56. 018 mm (2. 2048 - 2. 2054 in)
	⋮	⋮
20	10	2. 2048 - 2. 2054 in
	20	2. 2048 - 2. 2054 in (56. 003 - 56. 018 mm)
	⋮	⋮
40	10	56. 003 - 56. 018 mm
	20	56. 003 - 56. 018 mm (2. 2048 - 2. 2054 in)
	⋮	⋮
⋮	⋮	⋮

FIG. 26

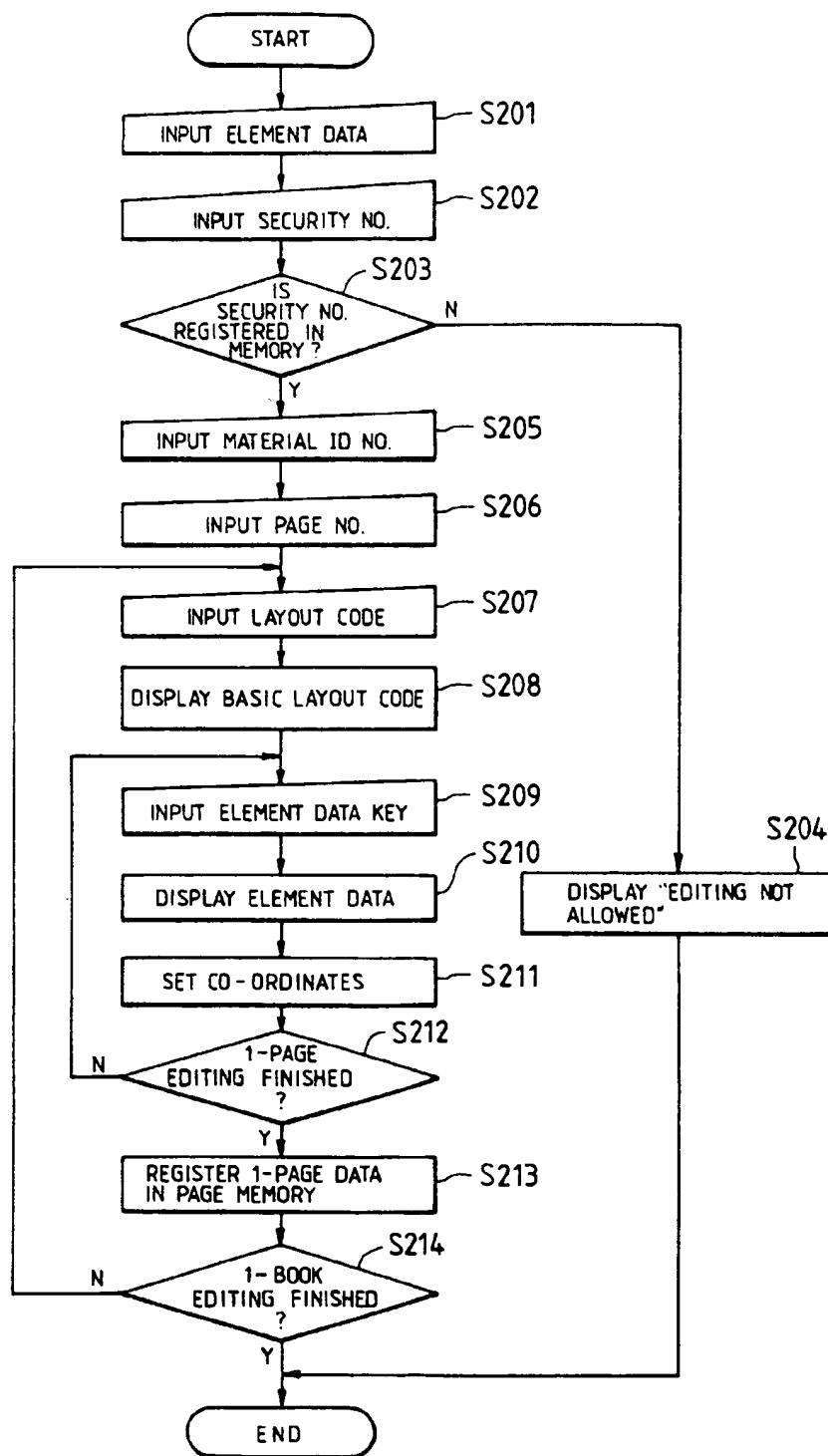


FIG. 27

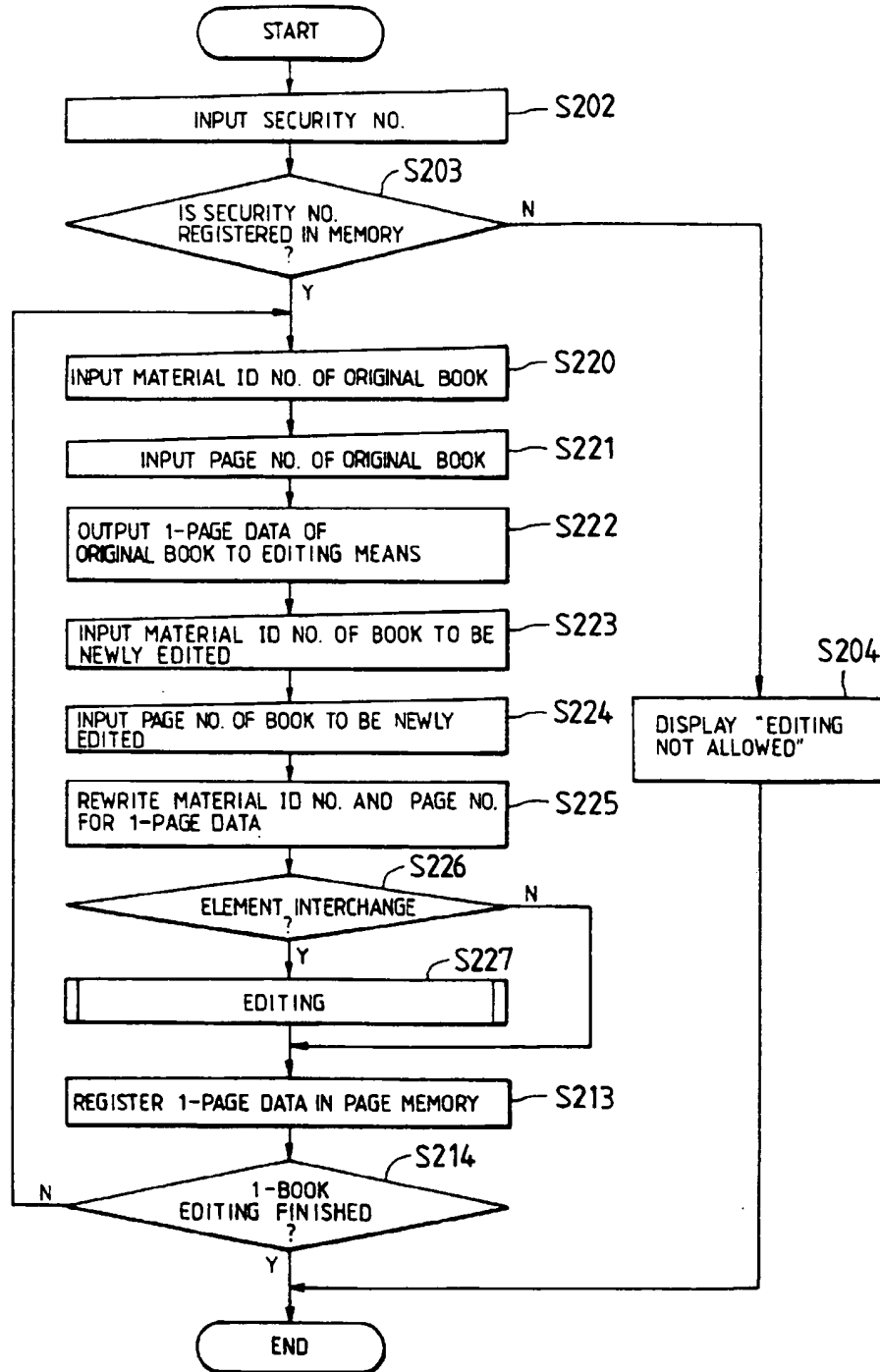


FIG. 28

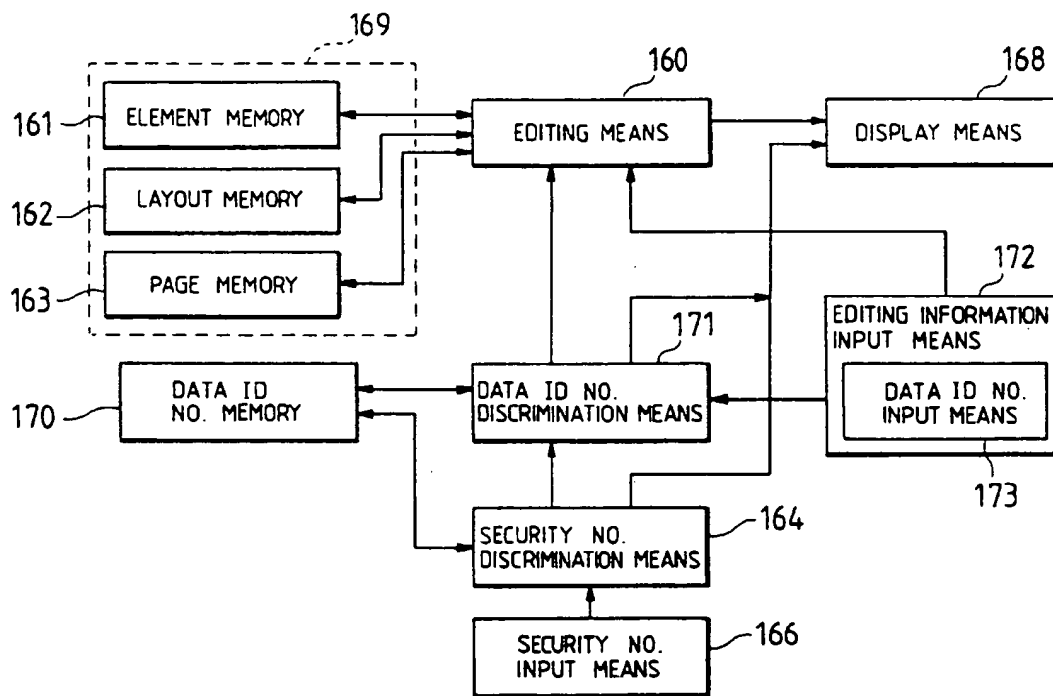


FIG. 29

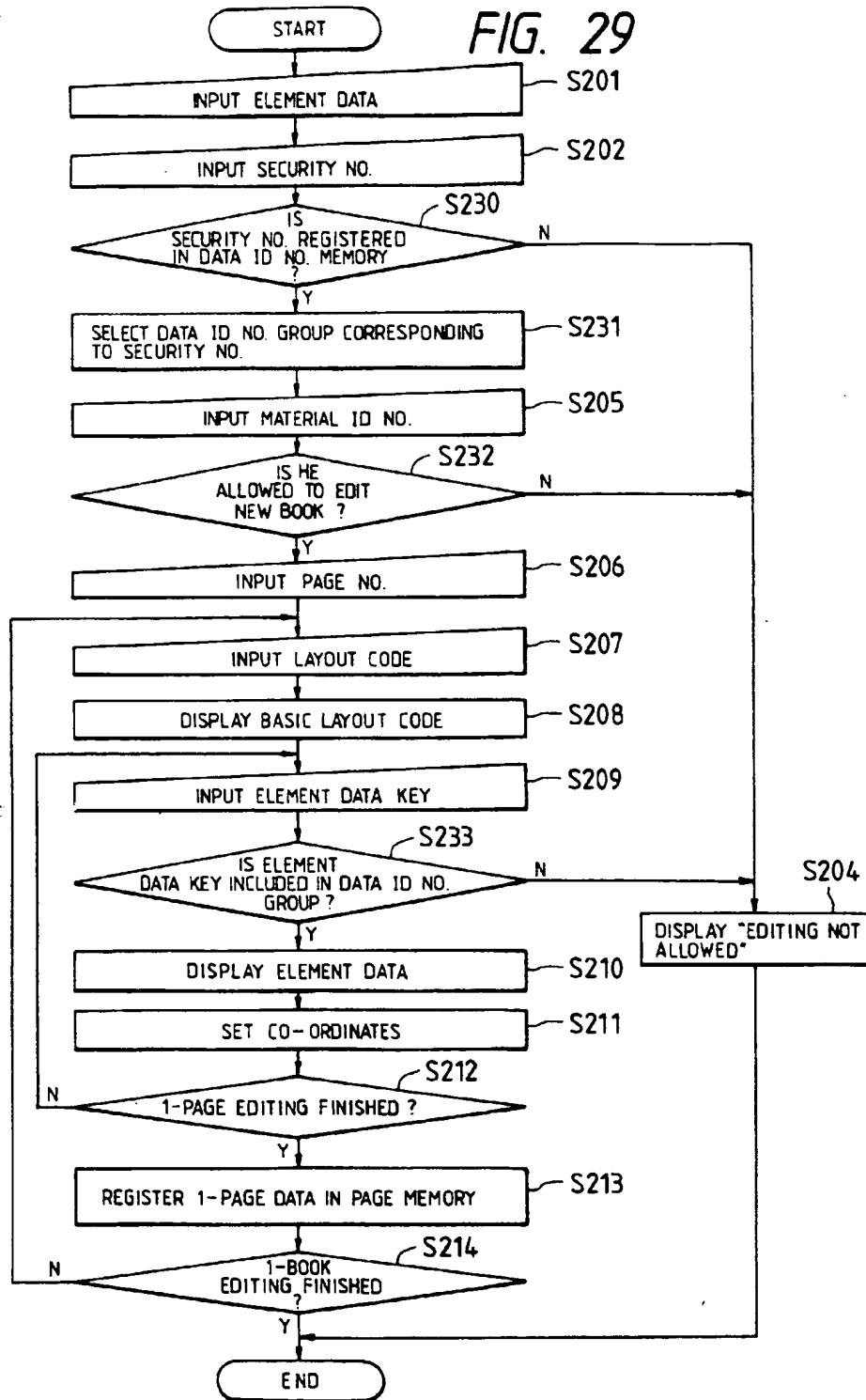


FIG. 30

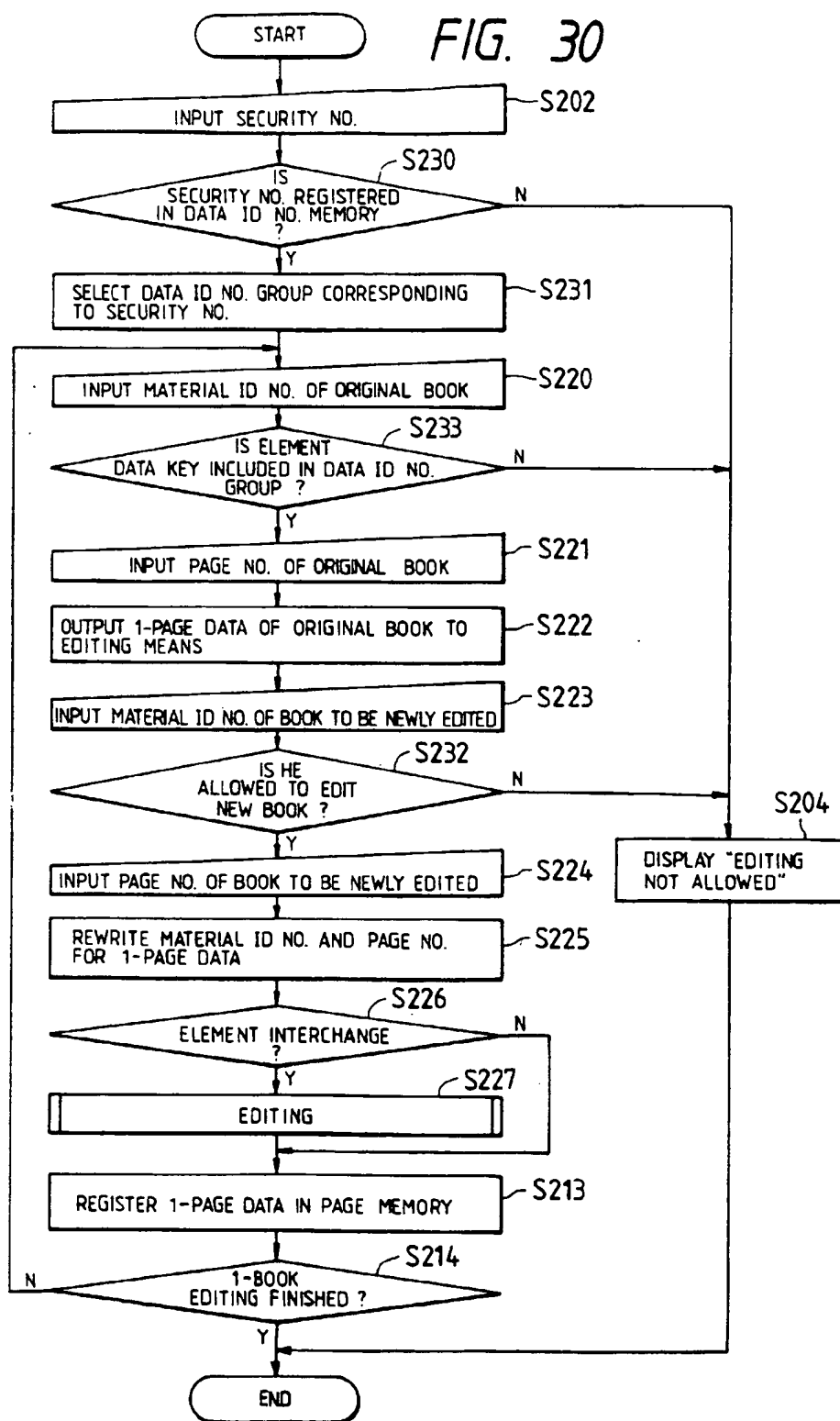


FIG. 31

TABLE a

SECURITY NO.	A 1 2 3 4
DATA ID NO.	B 1 0
	B 1 2
	B 1 5
	B 1 6
	NEW

TABLE b

SECURITY NO.	A 1 2 3 5
DATA ID NO.	B 1 2
	B 1 5
	B 1 6

TABLE c

SECURITY NO.	A 1 2 3 6
DATA ID NO.	B 1 0
	B 1 4

FIG. 32

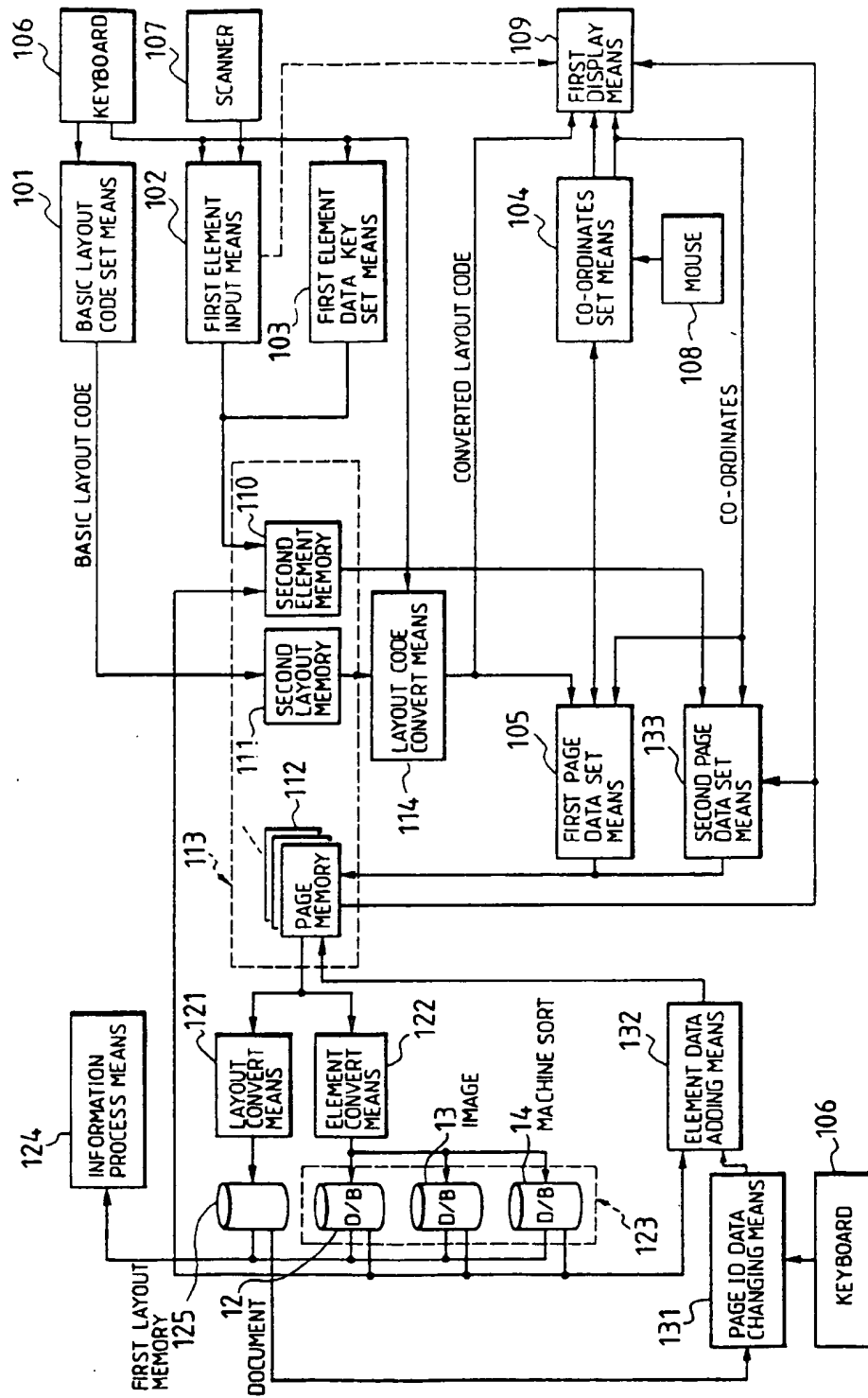


FIG. 33

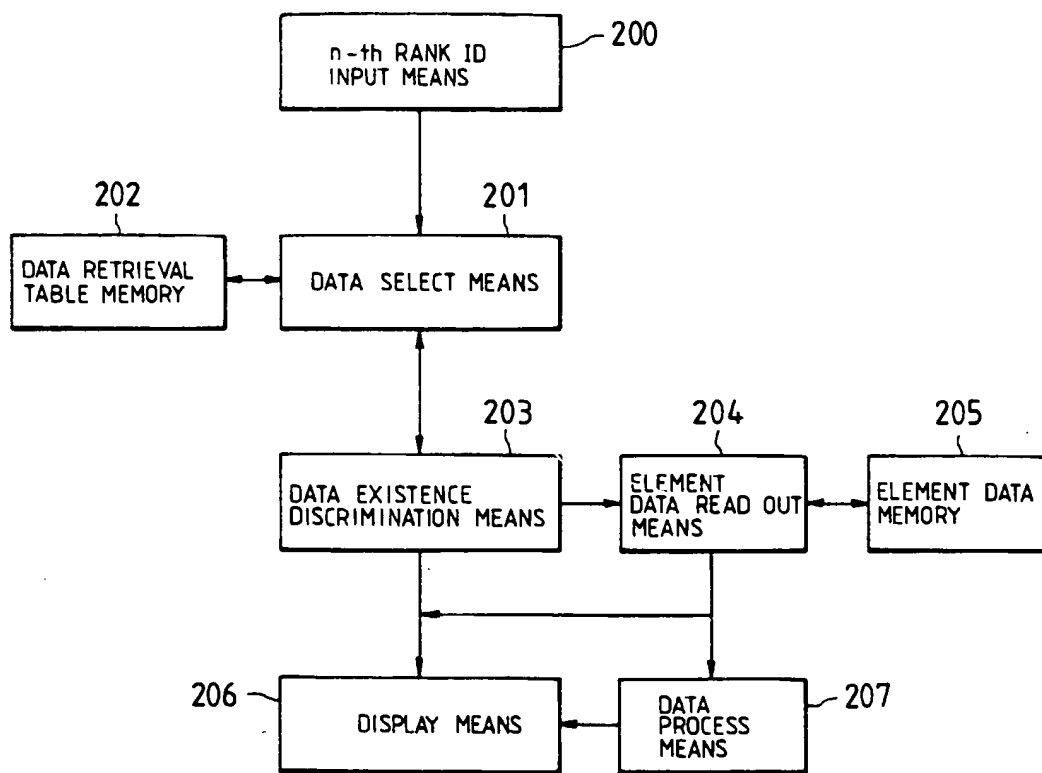


FIG. 34

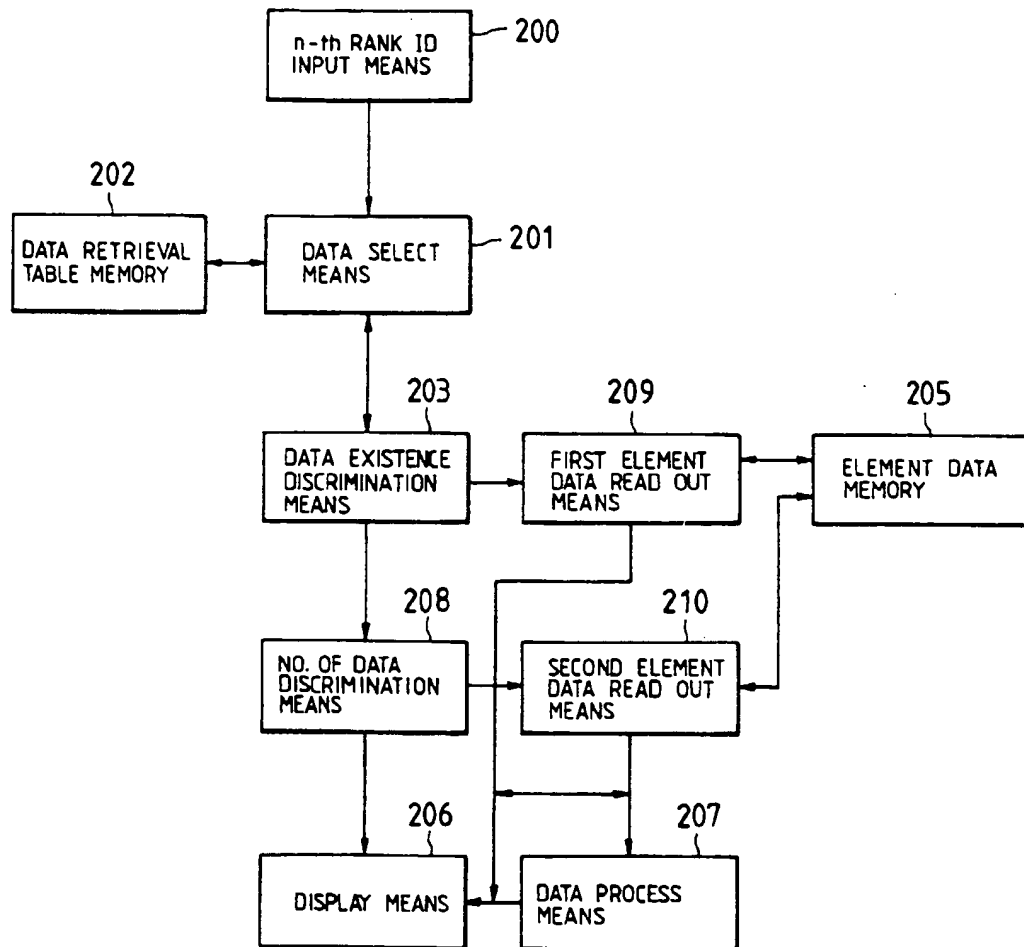
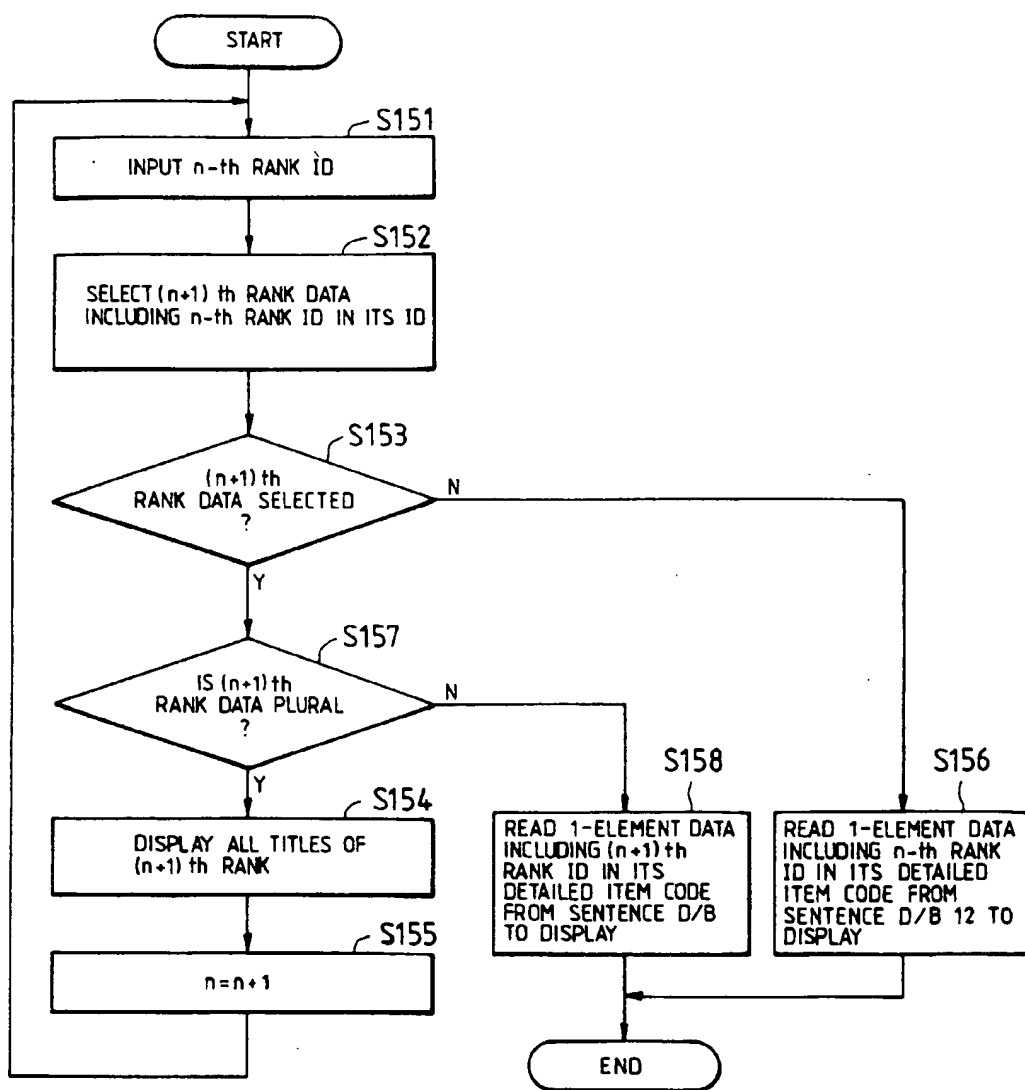


FIG. 35



Requested Patent: EP0651554A1

Title:

METHOD AND APPARATUS FOR THE ADDITION AND REMOVAL OF DIGITAL
WATERMARKS IN A HIERARCHICAL IMAGE STORAGE AND RETRIEVAL
SYSTEM. ;

Abstracted Patent: EP0651554 ;

Publication Date: 1995-05-03 ;

Inventor(s):

AXMAN MICHAEL STUART EASTMAN K (US); BARADAR ALI R EASTMAN KODAK
CO (US); MELNYCHUCK PAUL W EASTMAN KODA (US); RABBANI MAJID
EASTMAN KODAK CO (US) ;

Applicant(s): EASTMAN KODAK CO (US) ;

Application Number: EP19940420293 19941025 ;

Priority Number(s): US19930146371 19931029 ;

IPC Classification: H04N1/21; G06F1/00 ;

Equivalents: JP7212712 ;

ABSTRACT:

An image processing technique is described in the context of a hierarchical image storage and retrieval system. The method allows for the controlled addition and removal of digital watermarks from selected image components in the hierarchy. The method adds a digital watermark in a selected image resolution component and the means to remove it in an additional image component termed the watermark removal component. The method employs the encryption of the watermark removal component, and decryption with a special key, or password during authorized retrieval. This technique allows users of a distributed system the convenience of providing the entire image hierarchy on a single storage medium permitting images containing watermarks to be accessed without restriction for browsing and proofing, while the watermark removal requires knowledge and use of a controlled code.



(11) Publication number : **0 651 554 A1**

(12) **EUROPEAN PATENT APPLICATION**

(21) Application number : **94420293.6**

(51) Int. Cl.⁶ : **H04N 1/21, G06F 1/00**

(22) Date of filing : **25.10.94**

(30) Priority : **29.10.93 US 146371**

(43) Date of publication of application :
03.05.95 Bulletin 95/18

(84) Designated Contracting States :
DE GB

(71) Applicant : **EASTMAN KODAK COMPANY**
343 State Street
Rochester, New York 14650-2201 (US)

(72) Inventor : **Rabbani, Majid, Eastman Kodak Company**
Patent Legal Staff,
343 State Street
Rochester, New York 14650-2201 (US)

Inventor : **Melnychuck, Paul W., Eastman Kodak Company**
Patent Legal Staff,
343 State Street
Rochester, New York 14650-2201 (US)
Inventor : **Axman, Michael Stuart, Eastman Kodak Company**
Patent Legal Staff,
343 State Street
Rochester, New York 14650-2201 (US)
Inventor : **Baradar, Ali R., Eastman Kodak Company**
Patent Legal Staff,
343 State Street
Rochester, New York 14650-2201 (US)

(74) Representative : **Boulard, Denis et al**
Kodak-Pathé
Département Brevets
CRT-Zone Industrielle
F-71102 Chalon-sur-Saône Cédex (FR)

(54) **Method and apparatus for the addition and removal of digital watermarks in a hierarchical image storage and retrieval system.**

(57) An image processing technique is described in the context of a hierarchical image storage and retrieval system. The method allows for the controlled addition and removal of digital watermarks from selected image components in the hierarchy. The method adds a digital watermark in a selected image resolution component and the means to remove it in an additional image component termed the watermark removal component. The method employs the encryption of the watermark removal component, and decryption with a special key, or password during authorized retrieval. This technique allows users of a distributed system the convenience of providing the entire image hierarchy on a single storage medium permitting images containing watermarks to be accessed without restriction for browsing and proofing, while the watermark removal requires knowledge and use of a controlled code.

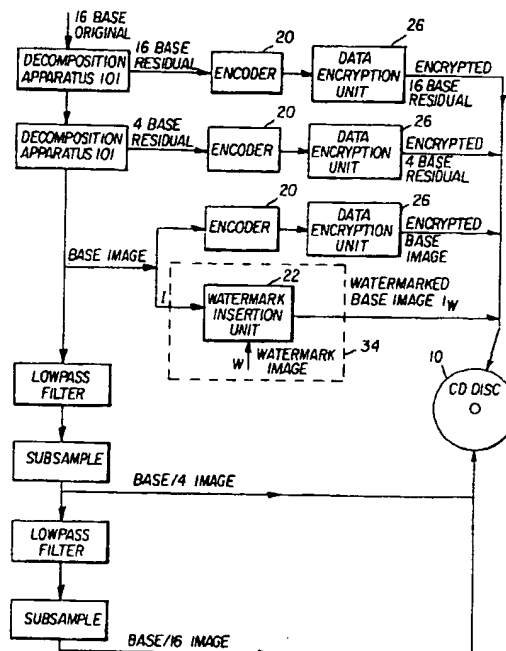


FIG. 2

EP 0 651 554 A1

Cross-reference to Related Application:

The present application is related to U.S. Patent Application Serial No. 08/026,726, entitled "Method and Apparatus for Controlling Access to Selected Image Components In An Image Storage and Retrieval System" filed March 5, 1993, by P. W. Melnychuck and assigned to Kodak, the assignee of the present application.

Technical field Of The Invention

The present invention is related to the field of digital image processing and more particularly to methods and associated apparatuses for adding and removing a digital watermark to and from a selected image resolution and the preventing of unauthorized use of associated higher resolution digital image components.

Background Of The Invention

A number of hierarchical techniques for image coding have been described in the open technical literature and in various patents. Of particular relevance to the present invention are the following publications:

P. J. Burt and E. H. Adelson, "The Laplacian Pyramid As A Compact Code," IEEE Trans. Comm., COM-31, 532-540 (1983).

J. Seberry and J. Pieprzyk, "CRYPTOGRAPHY: An introduction to Computer Security" Prentice Hall, 1988 and the following patents:

U.S. Pat No, 4,969,204 entitled "Hybrid Residual-Based Hierarchical Storage And Display Method For High Resolution Digital Images In A Multiuse Environment," by Paul W. Melnychuck and Paul W. Jones, 1990.

U.S. Pat No, 5,048,111 entitled "Hybrid Subband-Based Hierarchical Storage And Display Method For High Resolution Digital Images In A Multiuse Environment," by Paul W. Jones and Paul W. Melnychuck, 1991.

The publication by Burt, et al. teaches an encoding method for images termed the Laplacian pyramid, the Burt pyramid, or the residual pyramid. In this technique, the original image is lowpass filtered, and this lowpass image is subsampled to take advantage of its reduced bandwidth to provide an image of reduced dimension. This process of lowpass filtering and subsampling is repeated three times to generate a hierarchical structure, or pyramid of images of successively smaller dimensions. The total number of resolution levels are created depending on the application. Each lowpass image in this pyramid is then expanded to the dimensions of the next higher level by upsampling (inserting zeros) and filtering to form a prediction image for that level. This prediction image

is subtracted from its corresponding lowpass image in a subtractor to generate difference, or residual, images. The residual images corresponding to the levels of the lowpass pyramid form another pyramid which is termed the Laplacian, Burt, or residual pyramid. This technique is motivated by the fact that the residual images have a reduced variance and entropy compared to the original or lowpass images and may be quantized and entropy encoded to provide efficient storage of the data. Reconstruction is performed by interpolating the decoded lowpass image at the bottom of the lowpass pyramid and adding in the corresponding decoded residual to generate the next level in the lowpass pyramid. This process is iterated until the original image size is reached. A progressive improvement in reconstructed image quality and resolution can thus be obtained by displaying the reconstructed lowpass filtered image at each level of the pyramid. Note that errors introduced in the encoding process are propagated from one level to the next higher level in the decoding process.

The patent to Melnychuck and Jones (U. S. Pat. No. 4,969,204) teaches a modification of the Burt pyramid scheme by extending the lowpass pyramid structure to include one or more lowpass filtered images of successively smaller dimensions beyond the set described by Burt, et al. The advancement in the method of Melnychuck and Jones is that the residual pyramid is not extended to include these corresponding extended smaller dimensions. Hence, the Melnychuck and Jones pyramid contains the Burt pyramid plus additional lowpass filtered images of smaller dimensions. In a hierarchical image storage and retrieval system, the additional lowpass filtered images of smaller dimension can be retrieved directly, without interpolation and addition of residual components. In the context of the present invention, the Melnychuck and Jones pyramid provides for low resolution images that can be used for browsing or proofing. The use of these additional low resolution images for browsing and proofing means that the customer may use a simple retrieval mechanism and need not possess a more complex and hence, more expensive retrieval device that would be used to decode the higher resolution components of the pyramid. Of course, higher resolution images requiring interpolation and residual addition may be used for browsing and proofing as well.

A hierarchical image processing method will be described for the addition and removal of digital watermarks in selected image components, and for the restriction of selected high resolution image components from unauthorized use. An image hierarchy is constructed in the context of a multi-resolution environment whereby the user has the option of selecting the type of display medium and the desired resolution of this display medium. In particular, two types of display media are considered: video monitors and color hard copies, although photographic, thermal imaging,

and other types are also of interest. In Fig. 1 a prior art technique for decomposing, storing, recomposing, and displaying, a digital image using a hierarchical process is shown. An original digital image is decomposed to provide image versions at various resolutions to allow for the display of an HDTV quality image on video, an NTSC quality image with PAL/SECAM compatibility on video, one or more sub-NTSC quality images on video for overviews and browsing, and a very high quality image on color hard copy. Intermediate to the decomposition and recomposition steps, generally are inserted an encoding step, to compress the data for storage which in turn requires a decoding step when the data is read from storage.

Summary Of The Invention

The present invention places a digital watermark in a selected image resolution component and the means to remove it in an additional image component termed a watermark removal component. Encryption of the watermark removal component is used to prevent use of the image for the generation of unauthorized high quality color hard copy. A watermark is a form of graphic overlay that may contain a copyright notice or information regarding the restricted use of the image. In a distributed image system it is common to deliver an image of compromised image quality for purposes of browsing or proofing. A compromised rendition of the image is commonly distributed to prevent full utility or fulfillment of the image without proper payment for the service that generated the image. The term browsing refers to the process of image selection from a plurality of images based on some user-defined criterion. Such is the case when a user may select an image from a catalog of images depicting a particular object. The term proofing refers to the process of image selection based on the degree of desirability of a given image from a plurality of images. Such is the case when a professional portrait photographer distributes a plurality of images to a customer for selection and approval. The terms watermark, browsing and proofing described herein are not limited to the examples described above.

Upon selection of the desired image by the customer, the professional delivers a high quality rendition of the image, most often in the form of a high quality color hard copy. At all times the professional possesses the sole means of generating the high quality hard copy. In a conventional photographic system the means would be the original negatives of the images; in a digital hierarchical system according to the present invention, the means are higher resolution residual components.

In a digital imaging system, and in particular one that includes a hierarchical form of digital storage and retrieval, the professional may use a suitable digital storage medium such as a CD for the distribution of

proofs. In an unrestricted environment, the customer may choose a desired image resolution from the hierarchy for the purposes of browsing, proofing, or hard copy fulfillment. In those instances where it is desirable for the professional to deliver the digital storage medium containing the entire image hierarchy to the customer; it is also most economical to record the entire image hierarchy once onto the digital storage medium and avoid having to make a second copy containing only low resolution components for distribution. However, it is also desirable to restrict the use of selected high resolution components for the purpose of full image quality fulfillment until payment has been received. The professional may choose to provide low resolution image components for browsing or proofing, while maintaining restriction of the higher resolution components. Alternatively, he may be required to deliver a proof of high resolution. Such is the case when the image content contains information of small detail and the rendition of this detail is subject to approval via the proof. With traditional photographic prints, the professional may place a stamp, or watermark on a strategic location on the print, so as to render the print useless from a fulfillment point of view. Note with digital images that fulfillment may mean high quality video at NTSC/PAL/SECAM, HDTV, or hard copy. In the present invention, the professional places a digital rendition of the watermark on a selected image component. The removal of the watermark is done through an additional image component containing the reverse of the watermark. The customer, having possession of the digital storage medium CD would possess the means for generating his own high quality hard copy when authorized by the professional. Upon payment to the professional, the professional or his agent provides to the customer the information necessary to remove the watermark for full image quality fulfillment. In the present invention, that information would be an authorization code, key, or password that would be inputted to the image processing system accessing the storage medium, to unlock the restricted high resolution components. An advantage of this technique is that the customer may possess all information pertinent to generating high quality hard copy without the need to physically return to the professional for additional image components.

It may additionally be desirable to use some form of hierarchical image representation for the purpose of browsing or proofing in a distributed system because the hierarchy naturally provides a plurality of resolutions, and hence levels of image quality, from which to choose the proof image. No additional operation of compromising the image is necessary; the professional simply chooses at what resolution level(s) he wants to restrict access.

Systems that use a hierarchical structuring of the image data have not been employed in the past for

distribution purposes because of the lack of means to simultaneously provide low resolution components for browsing and proofing, while offering restricted access to the remaining hierarchical components for full quality image copy. Additionally, the means to generate and remove a digital watermark in a hierarchical image structure had not been previously considered.

The present invention permits the advantages of hierarchical image decomposition to create a series of residual components, direct retrieval of the additional low resolution images according to the Melnychuk and Jones pyramid, the addition and removal of a digital watermark in a selected image resolution component, and prior art encryption methods applied to the watermark removal component and the residuals, to provide for a system of browsing, proofing and restriction of the high resolution image components suitable in a distributed image system. It is assumed that the residual components and the watermark removal component are symbol encoded using the encoder box 20 in Figures 2 and 4 into a binary string of 1's and 0's either via fixed-length coding techniques (where a binary code word of a fixed-length is assigned to each symbol) or variable-length encoding techniques such as Huffman coding or arithmetic coding. The residual data may also be quantized prior to encoding, or it may be encoded in a lossless manner, i.e., without quantization. Data encryption box 26 is applied to the watermark removal component and if desired, also to the encoded quantized (or non-quantized) residual data. It is assumed that the encryption process is reversible. Hence, the decryption box 28 provides the exact data prior to data encryption.

In one embodiment of the invention a storage medium is called for having stored therein at least one low resolution digital image and at least one high resolution digital image, with said high resolution digital image encoded with a watermark that requires an authorization code for removal.

From the foregoing, it can be seen that it is a primary object of the present invention to provide a method and associated apparatus for storing and controllably retrieving digital images stored in a hierarchical format on a suitable digital storage distribution medium that allows the originator of the distribution medium to distribute the medium containing the entire image hierarchy and a controllably removable watermark for the purpose of retrieving low resolution images for browsing or proofing without compromising the originator's need to withhold the means for creating hard copies of the images without the watermark.

It is another object of the present invention to provide the means for controllably inserting and removing a watermark for a digital image.

It is another object of the present invention to pro-

vide the means for compromising a selected image component of a hierarchical formatted digital image by adding a digital watermark to the selected image component, and recording the selected image component containing the watermark as part of the image hierarchy on a digital storage distribution medium.

In association with a digital image, it is another object of the present invention to provide a means for creating a watermark removal component, and for controllably restricting access to the watermark removal component.

It is another object of the present invention to provide a means for affixing a watermark to a digital image and for controllably removing the watermark.

The above and other objects of the present invention will become more apparent when taken in conjunction with the following description and drawings wherein like characters indicate like parts and which drawings form a part of the present description.

Brief Description Of The Drawings

Fig. 1 is a block diagram illustrating the prior art Melnychuk and Jones hierarchical storage and display method.

Fig. 2 is a functional block diagram illustrating a hierarchical image decomposition technique incorporating a watermark insertion into an image component.

Fig. 3 is a functional block diagram illustrating a reconstruction technique for reconstructing the images decomposed by the system of Fig. 2.

Fig. 4 is a functional block diagram of another hierarchical image decomposition technique incorporating a watermark insertion into an image component.

Fig. 5 is a functional block diagram illustrating a reconstruction technique for reconstructing the images decomposed by the system of Fig. 4.

Detailed Description of the Invention

In the following description of the preferred embodiments, it will be assumed that the highest resolution of the image hierarchy is composed of 3072 x 2048 pixels and that this resolution is adequate to produce photographic quality originals on an appropriate digital output device. It is also assumed that a moderately high resolution level of the hierarchy composed of 1536 x 1024 pixels is adequate to generate a high quality HDTV display, or a small-sized photographic quality print on an appropriate digital output device. It is also assumed that the lowest resolution levels of 192 x 128 pixels, 384 x 256 pixels, and 768 x 512 pixels are generated and stored onto a digital storage medium such as a CD. These resolution levels are provided to give the reader an insight as to the operation of one or more embodiments of the invention

with the understanding that other resolutions or arrangements may be chosen to suit specific needs without detracting from the teachings of the present invention.

Referring now to Fig. 2, a hierarchical residual decomposition technique, for decomposing a 16BASE original image to form a 16BASE residual, a 4BASE residual, a BASE, a BASE/4, and a BASE/16 image, incorporating the teachings found substantially in Fig. 7 of the patent to Melnychuk and Jones (U. S. Pat. No. 4,969,204), in combination with the present invention is shown. The BASE image is processed in box 34 to incorporate a watermark and to provide a watermarked BASE image.

An example of a watermark insertion box 34 is given by the watermark insertion unit 22 whereby a watermark image W is combined with the input image I to create a watermarked image I_w . In this example, it is assumed that the input image I and the watermark image W are of the same size and the same bit-depth. For example, if the input image I is an 8-bit image representing the luminance component of a color image, the watermark image W would also be an 8-bit image. Similarly, the watermarked image I_w would have the same size and each pixel value would be represented with 8 bits. An example of a watermark insertion unit 22 is one where the input image I and the watermark image W are combined according to the following equation to create the watermarked image I_w

$$I_w(i,j) = I(i,j) + \alpha W(i,j)$$

Where (i,j) denotes the two-dimensional location of the pixels in the image and the operation is performed for all the pixels in the input image. The watermark image W is prepared by the originator of the storage medium and may contain the logo of the originator or any other pattern that the originator may wish to use as a watermark. The parameter α , which can be either positive or negative, controls the watermark contrast and is also selected by the originator and can vary from one image to another. Larger magnitudes of α would, in general, create a higher contrast watermark. Also, to guarantee that the watermarked image I_w has the same bit-depth as the input image I , the watermarked image I_w is clipped to the same range as the input image. For example, for an 8-bit image with pixel values in the range of 0 to 255, for every pixel location (i,j) , the value of $I_w(i,j)$ is clipped to 255 if the result of the above equation exceeds 255 and is set to zero if that result is less than zero. It should be noted that this example illustrates only one method of implementing the watermark insertion box 34 and the originator of the storage medium may incorporate any other method to generate a watermark that creates the desired effect of inhibiting the use of the image.

The BASE/16, BASE/4, and watermarked BASE images are stored on the digital storage medium 10 in direct (unencrypted) form. The BASE image, which

in this case serves as the watermark removal record, is encrypted in the data encryption unit 26. The data encryption unit 26 consists of either a private-key data encryption algorithm (also referred to as symmetric data encryption algorithm) or a public-key data encryption algorithm (also referred to as asymmetric data encryption algorithm) both of which have been explained in the prior art and in the reference book by Seberry and Pieprzyk cited before. Examples of private-key encryption algorithms that can be used in the data encryption unit 26 are either block ciphers such as the Data Encryption Standard (DES) which uses a 56-bit key and operates on blocks of data of length 64 bits at a time, or a stream cipher algorithm such as RC-4, a commercially available encryption software that uses a 40-bit key component. The encrypted BASE image is also stored on the storage medium 10. The 4BASE and 16BASE residual components are also stored on the digital storage medium 10 either in direct (unencrypted) form or in encrypted form depending on the level of security desired by the application. In the case that the encryption of any or all of the residual data are needed, either the same key used in encrypting the BASE image is used or a separate key is used. The use of multiple encryption keys provides the originator of the storage medium with more flexibility in controlling the access to the various resolutions of the image hierarchy.

For browsing or proofing, a procedure illustrated by Fig. 3 is employed. A user retrieves the BASE/16, BASE/4, or watermarked BASE image directly without decryption from the digital storage medium 10. Upon authorization, the user inputs a decryption key(s) to the data decryption unit 28 to allow the decryption of the original BASE image (and the residuals) to be performed. An example of a data decryption unit 28 is a software implementation of a decryption algorithm corresponding to the reverse operation of the encryption algorithm employed in the data encryption unit 26. One example of a set of encryption/decryption algorithms is the Data Encryption Standard (DES) which has been explained in full detail in the reference book by Seberry *et al* mentioned before. Note that the decryption key(s) must be provided by the originator of the storage medium. Upon the decryption of the BASE image and the residual components, these components can be used to arrive at full image quality fulfillment.

In a second embodiment, illustrated in Fig. 4, the 16BASE image is decomposed by decomposition apparatus 101 into a residual pyramid consisting of the 16BASE, 4BASE, and BASE. The BASE image is further decomposed to create the BASE/4 and BASE/16 images, through low pass filtering and subsampling. BASE/4 and BASE/16 are not part of the residual pyramid and hence they are available directly for display on a monitor.

A watermark, as described in the previous em-

bodiment in Fig. 2, is inserted in the BASE image in box 34 to arrive at a watermarked BASE image. This watermarked BASE image is then interpolated to the size of the 4BASE image using linear interpolation as indicated by the interpolator box 24. A difference is formed in subtractor 32 between the original 4BASE image and the interpolated watermarked BASE image to form a modified 4BASE residual that serves as the watermark removal record. The difference in this embodiment versus the first embodiment is that the watermark removal record is the modified 4BASE residual instead of the BASE image. This modified 4BASE residual is encrypted using the data encryption unit 26 as described before and is then stored on the storage media 10 along with the BASE/16, BASE/4, and watermarked BASE image in direct (un-encrypted) form. Finally, the 16BASE residual data is stored on the digital storage medium either in direct or encrypted form depending on the application.

For browsing or proofing, the system of Fig. 5 is employed. The user retrieves the BASE/16, BASE/4, or watermarked BASE image directly without decryption from the digital storage medium 10. Upon authorization, the user inputs the decryption key to the data decryption unit 28 to allow the decryption to be performed to generate the modified 4BASE residual. The watermarked BASE image is interpolated using linear interpolation and is added to the decrypted modified 4BASE residual in the reconstruction apparatus 210 to recover the original 4BASE image. If the residuals have not been quantized, the 4BASE image can be exactly recovered. In the case where the residuals have been quantized, some discrepancy between the original 4BASE image and the 4BASE image recovered according to the above scheme would exist. The degree of this discrepancy would depend on the coarseness of the quantizer employed in the quantization of the residual components. Note that the decryption key must be provided by the originator of the storage medium.

It is to be understood that in some instances it may be desirable to place a watermark upon the low resolution images to control their access.

While there has been shown what are considered to be the preferred embodiments of the invention, it will be manifest that many changes and modifications may be made therein without departing from the essential spirit of the invention. It is intended, therefore, in the annexed claims, to cover all such changes and modifications as may fall within the scope of the invention.

Parts List:

- 10 Digital storage medium (CD-Disc)
- 20 Encoder
- 22 Watermark insertion unit

- 24 Interpolator
- 26 Data encryption unit
- 28 Data decryption unit
- 30 Decoder
- 32 Subtractor
- 34 Watermark insertion box
- 101 Decomposition apparatus
- 201 Reconstruction apparatus

Claims

1. A storage medium having stored therein at least one low resolution digital image and at least one high resolution digital image, with said high resolution digital image encoded with a watermark that requires an authorization code for removal.

2. The storage medium according to claim 1 and further having stored thereon at least one additional high resolution digital image that is not encoded with a watermark and is accessed with the authorization code in place of the high resolution digital image encoded with the watermark.

3. A storage medium having stored therein at least one low resolution digital image and at least one high resolution digital image in the form of a BASE image, residual image components and a watermark component, with said low resolution digital image, said BASE image or said high resolution image formed by the combination of the BASE image with said residual image components and a watermark component being accessible without an authorization code.

4. The storage medium of claim 3 in combination with an authorization code to remove the watermark component from an accessed high resolution image.

5. A system for controlling the uncompromised use of a high resolution digital image stored on a storage medium as BASE and residual components, comprising:

means for encrypting the residual components stored on said storage medium using a watermark code;

means for accessing the BASE and encrypted residual components;

means for combining the accessed BASE and residual components to reconstruct the high resolution digital image with the watermark code; and

means for authorizing the removal of the watermark code.

6. A system for controlling the uncompromised use of a high resolution digital image comprising:

means for forming a hierarchy of lower resolution digital images from the high resolution digital image;

means for forming residual images that are a function of differences between adjacent images in the hierarchy of lower resolution digital images;

means for encrypting at least one of the formed residual images with a watermark code; 5

storage means for storing the formed hierarchy of lower resolution images and the at least one encrypted residual image;

means for reconstructing high resolution images by accessing and combining a lower resolution image with a residual image; 10

means for displaying of the at least one encrypted residual image with the watermark; and

means for controllably removing the watermark code to permit an uncompromised use of the high resolution digital image. 15

7. A recording medium having stored thereon a plurality of digital images with each of the digital images being comprised of a low resolution digital image component and at least one residual digital image component which is combinable with the low resolution digital image component to form a higher resolution digital image incorporating a watermark which is removable with an authorization code. 20

8. A method for controlling the use of a digital image stored on a storage medium in a hierarchical form comprised of a BASE image and at least one residual image component, comprising the steps of: 25

a) associating a watermark with said at least one residual image component; 30

b) permitting access to the BASE image for low resolution viewing of the digital image;

c) combining the BASE image with the at least one residual image component and an associated watermark to form the digital image for viewing, printing and/or storing; and 35

d) controllably providing a watermark removal code to remove the watermark from the formed digital image of step c.

9) A storage medium having stored thereon at least one digital image encoded with a watermark that requires an authorization code for removal. 40

45

50

55

7

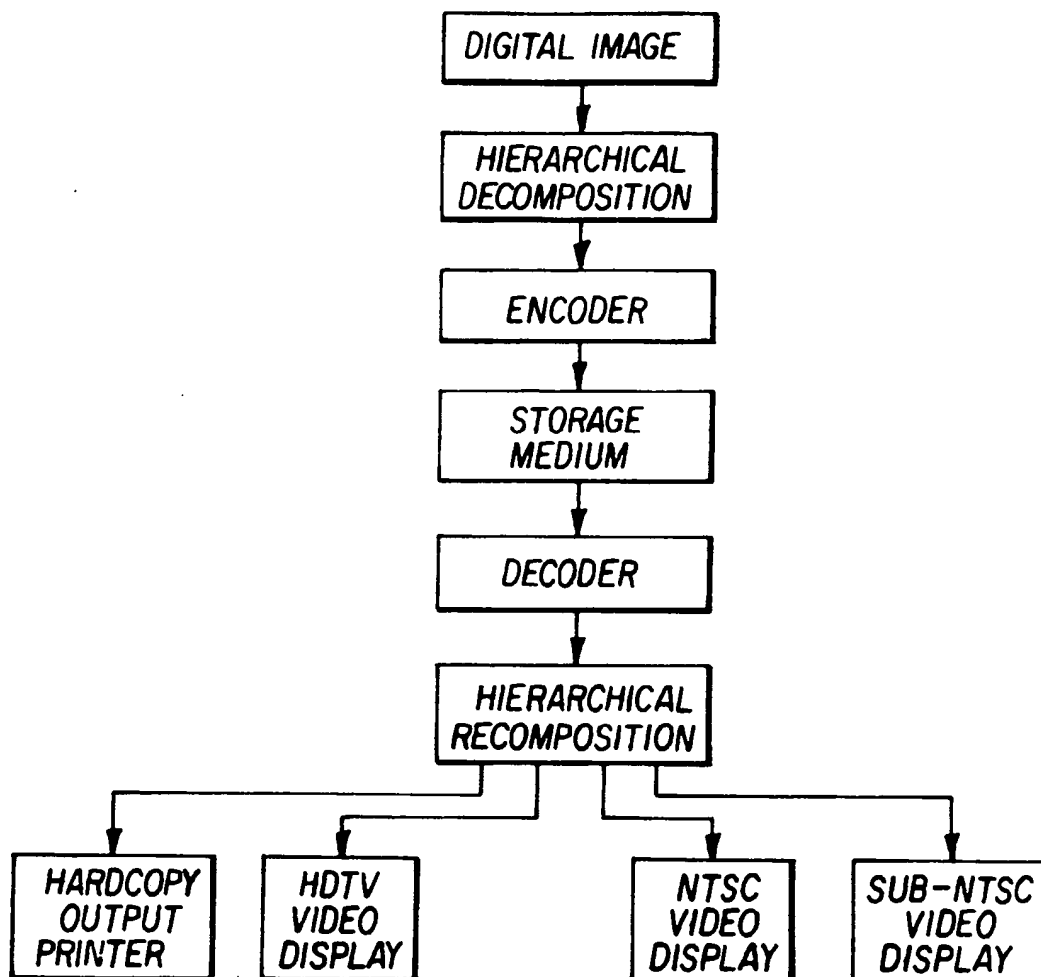


FIG. 1
(prior art)

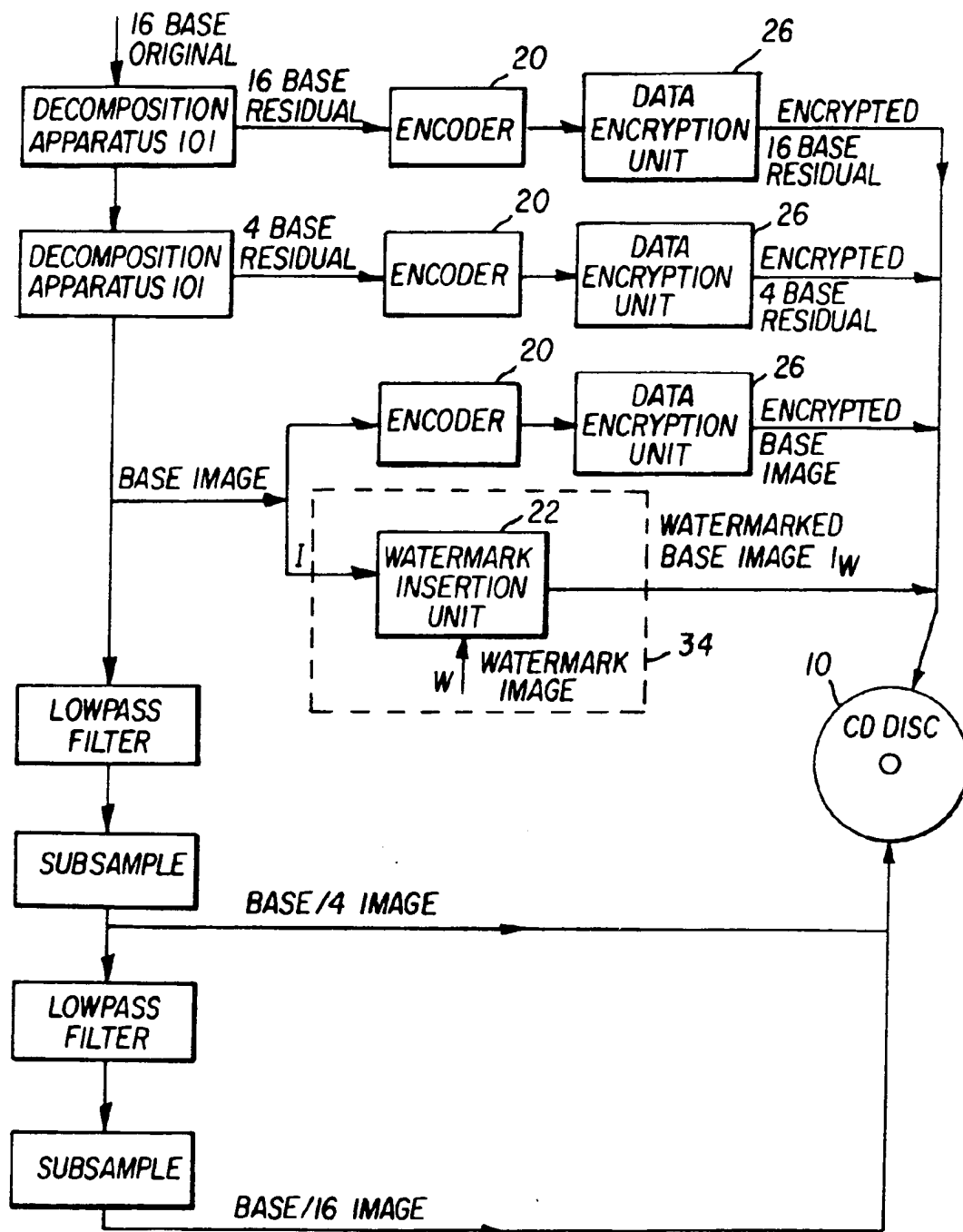
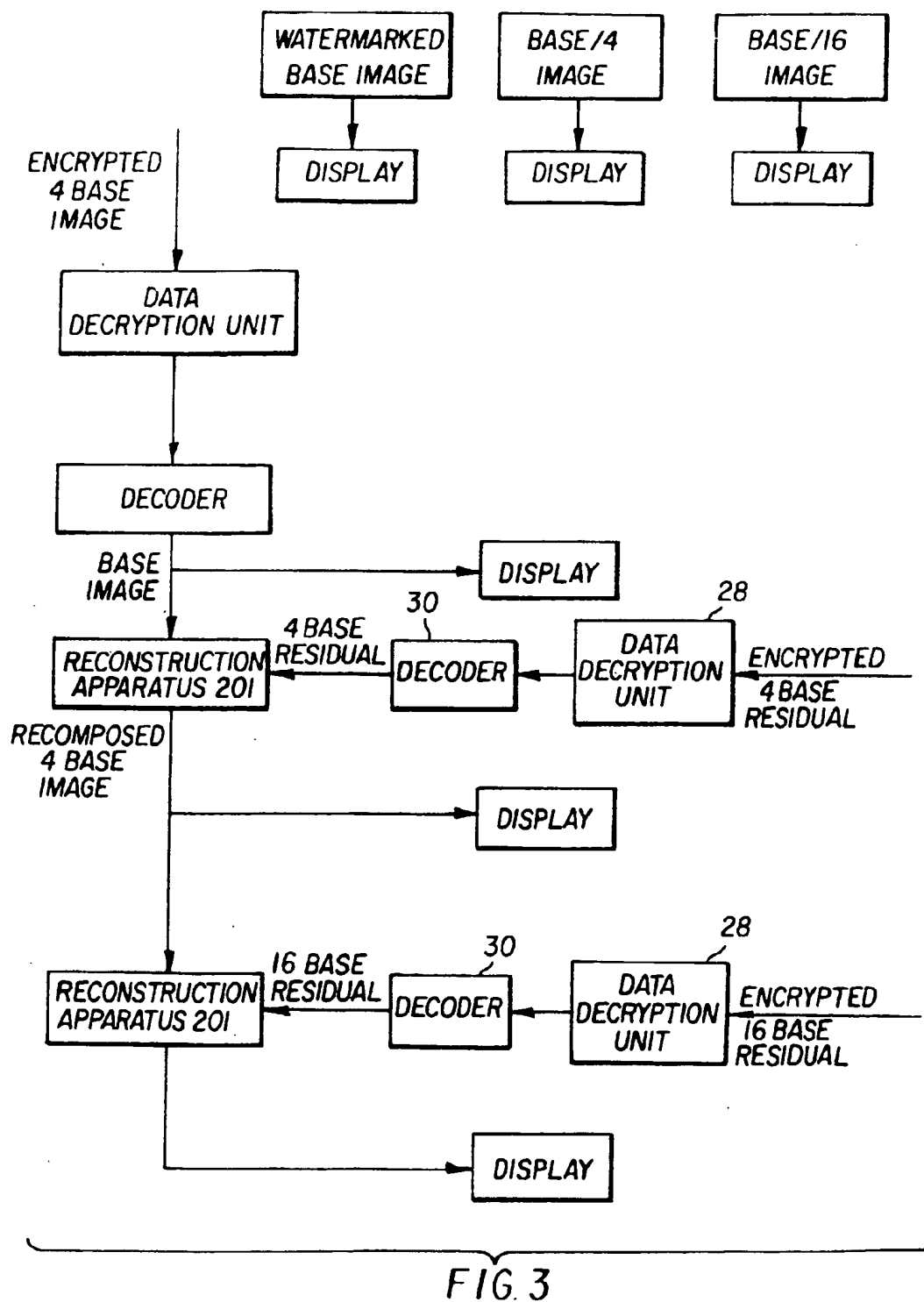


FIG. 2



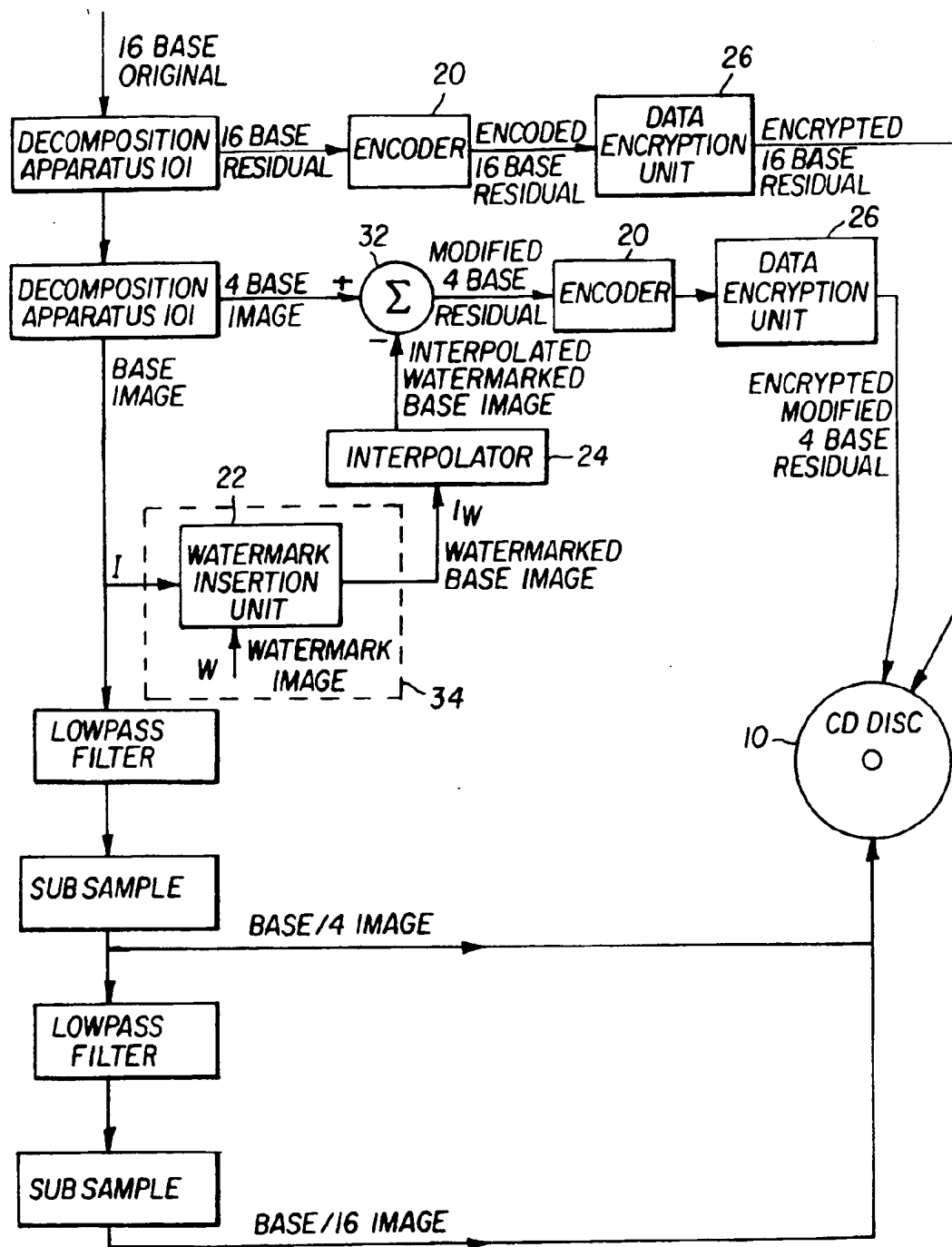
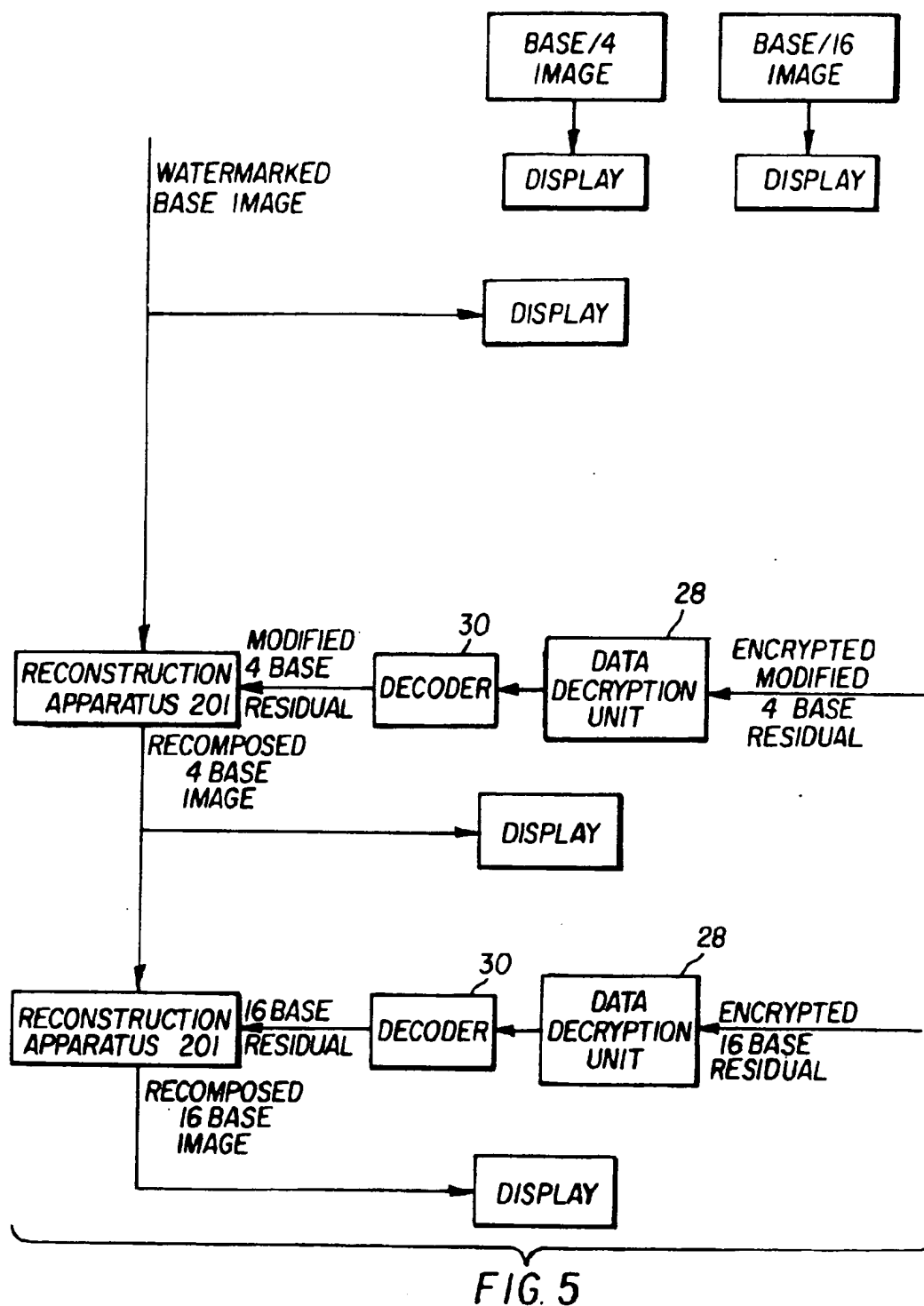


FIG. 4





European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 94 42 0293

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claims	CLASSIFICATION OF THE APPLICATION (Int. CL. 6)
D, Y	US-A-4 696 204 (MELNYCHUCK ET AL) * the whole document *	1-9	H04N1/21 G06F1/00
Y	--- ELECTRONICS AND COMMUNICATIONS IN JAPAN, vol.73, no.5, May 1990, NEW YORK, US; pages 22 - 33 N.KOMATSU ET AL 'A Proposal on Digital Watermark in Document Image Communication and Its Application to Realizing a Digital Signature' * figures 1-5 * * page 22, left column, line 1 - page 27, left column, line 23 *	1-9	
P, Y	EP-A-0 614 308 (EASTMAN KODAK) * abstract; figure 2 * * column 4, line 52 - column 5, line 4 *	2	
The present search report has been drawn up for all claims			TECHNICAL FIELDS SEARCHED (Int. CL. 6)
			H04N G06F
Place of search		Date of completion of the search	Examiner
THE HAGUE		30 January 1995	Powell, D
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons Δ : member of the same patent family, corresponding document			

EPO FORM 1503 (01.91) (P01.001)

Requested Patent: EP0084441A2

Title:

METHOD AND APPARATUS FOR THE PROTECTION OF PROPRIETARY
COMPUTER SOFTWARE. ;

Abstracted Patent: EP0084441 ;

Publication Date: 1983-07-27 ;

Inventor(s): POOLE TERENCE ERIC CYRIL;; ROGERS DAVID NIGEL ;

Applicant(s): TABS LIMITED (GB) ;

Application Number: EP19830300178 19830113 ;

Priority Number(s): GB19820001353 19820119 ;

IPC Classification: G06F13/00 ;

Equivalents: ;

ABSTRACT:

Apparatus for protecting proprietary computer software against unauthorised use comprises a store (11) for selected data, means (10) for comparing data successively communicated by a program running on a computer (1) with data from the storage means, means such as an indelible memory (15) associated with a microprocessor (14) for storing identifying data, and transmitting means (14) for sending stored identifying data to the computer. When a match is detected by the comparator, the identifying data are sent to the computer, which requires this data for continued normal running. A copy of the software cannot run on a computer without associated protection apparatus and unauthorised copies will therefore be unusable unless the protection apparatus can be obtained. For a great degree of protection, a sequence of matches and identifying data messages may be required to allow continued normal running of a program.

12

EUROPEAN PATENT APPLICATION

②¹ Application number: 83300178.7

Ⓢ Int. Cl.³: G 06 F 13/00

② Date of filing: 13.01.83

③ Priority: 19.01.82 GB 8201353

④ Date of publication of application:
27.07.83 Bulletin 83/30

⑧ Designated Contracting States:
AT DE FR GB NL SE

(71) Applicant: TABS LIMITED
Sopers House Chantry Way
Andover Hampshire SP10 1PE(GB)

(72) Inventor: Rogers, David Nigel
The Lodge 118 Ringwood Road
Verwood Wimborne Dorset(GB)

(72) Inventor: Poole, Terence Eric Cyril
The Old Rectory
Blackford Yeovil Somerset(GB)

74 Representative: Pritchard, Collin Hubert et al,
Mathys & Squire 10 Fleet Street
London EC4Y 1AY(GB)

⑤4 Method and apparatus for the protection of proprietary computer software.

57) Apparatus for protecting proprietary computer software against unauthorised use comprises a store (11) for selected data, means (10) for comparing data successively communicated by a program running on a computer (1) with data from the storage means, means such as an indelible memory (15) associated with a microprocessor (14) for storing identifying data, and transmitting means (14) for sending stored identifying data to the computer. When a match is detected by

the comparator, the identifying data are sent to the computer, which requires this data for continued normal running. A copy of the software cannot run on a computer without associated protection apparatus and unauthorised copies will therefore be unusable unless the protection apparatus can be obtained. For a great degree of protection, a sequence of matches and identifying data messages may be required to allow continued normal running of a program.

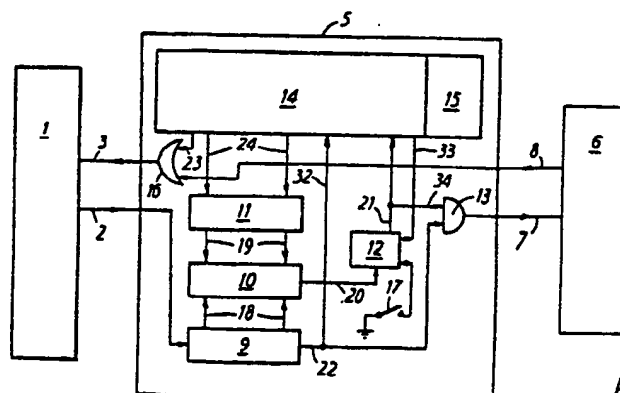


FIG 1

- 1 -

Method and
Apparatus for the Protection of Proprietary Computer Software

The present invention relates to apparatus for the protection of proprietary computer software, and in particular for protecting such software against unauthorised use.

The large amount of time and skill which is frequently involved in the writing of computer programs means that software is very often expensive. When a hardware and software package is sold to users of computer systems, a significant part of the cost will be attributable to the software. Currently, hardware costs are falling as the number of users increases and mass production techniques can be used. For software, reproduction costs are low, but initial development costs remain high. The market has become very competitive and the unscrupulous, instead of commissioning their own programs, may be tempted to take the much cheaper alternative of copying without authority the programs of others. For sophisticated computer systems which are tailored to the particular user's requirements it is relatively straightforward to monitor the system and avoid unauthorised adoption by others of the associated software. For standard systems, on the other hand, software products are frequently shipped in volume, often through distributors, to a large buying public, giving the opportunity for copying.

One proposal for protecting software is for the software package to be sold with a hardware key. The key could, for example, be physically connected to the computer data bus. With

- 2 -

such an arrangement, the software would be written to read a code, such as the serial number of the particular software package, embedded in the hardware. Provided this code matched counterparts held at suitable places in the software, the running of the program would be allowed to continue, but if there was no match, the program would be stopped from running. A person who copied the program and attempted to use it with a different hardware key or with no key at all would not be able to do so.

The above proposal has two disadvantages: one is that the key is very simple and can quite easily be copied and the other is that the key is specific to each type of hardware. Software is often written in a form which allows it to be easily transferred from one basic hardware system to another. This enables the user to upgrade his hardware without having to invest in new software at the same time and also enables the supplier to offer his software for use on the latest equipment at little extra expense to him. The simple key proposed would not therefore be of much value.

The present invention provides apparatus for protecting software which is adaptable for use on all kinds of computer hardware conforming to certain standards and which cannot easily be copied.

Accordingly, the invention provides apparatus capable of protecting proprietary computer software against unauthorised use, comprising storage means for storing selected data, comparison means for comparing data successively communicated by a program

running on a computer with data from the storage means, means for storing identifying data and transmitting means for sending stored identifying data to the computer, whereby on detection of a match in use, the transmitting means send said identifying data to the computer, said data being require for continued normal running of a proprietary program.

If the program were to be run on a computer without associated protection apparatus, the program would cease running normally as it would not receive the required identifying data.

The apparatus may be provided with communications means which enable data to be transferred to and from a computer associated with the apparatus. The apparatus may also be adapted for communication with a peripheral device. The communications means can consist of simple and standard communications hardware, making the apparatus very versatile and usable with many kinds of computer hardware.

In one form, the apparatus comprises blocking means to prevent transfer of data to the peripheral device after detection of a match and until after said identifying data has been transmitted by said transmitting means. It may also include switching means for preventing actuation of the blocking means, to allow the running of a non-proprietary program without interruption by the blocking means.

In a preferred form of the apparatus which provides particularly effective protection, said selected data comprises a code of data bits and said identifying data comprises a message

of data bits and a plurality of the data codes and data messages are stored in said respective storage means for the data codes to be successively compared and said data messages to be successively transmitted, whereby a proprietary program on an associated computer will cease running normally unless a predetermined sequence of identifying data messages is received by the program from the apparatus. The apparatus can be programmed to alter in a predetermined manner the selected group of data bits which are to be matched and to send a sequence of messages to the computer, these being required by the computer in order to allow continued normal running of the program.

The invention will now be described, by way of example, with reference to the accompanying drawing which is designated Figure 1.

The Figure shows, in block schematic form, particular protection apparatus constructed in accordance with the invention, in combination with other apparatus.

Referring to the Figure, a programmed computer 1 is connected to a peripheral device 6, which could be for example a printer, a visual display unit, or another computer. Connected between the computer 1 and peripheral device 6 is protection apparatus 5 according to the invention. Communications means 2, 3 and 7, 8 are standard RS-223-C interfaces for transmission of data to and from the computer and peripheral device.

Protection apparatus 5 comprises a shift register 9 which receives groups of data bits from the computer 1 via line 2. A further register 11 stores data in the form of an eight digit code number for example the particular code of the proprietor of the software, and is under the control of a microprocessor 14 which can, via lines 24, alter the data stored in the register 11 if required. Registers 9 and 11 are connected respectively via lines 18 and 19 to a comparator 10. The output of the comparator is connected via line 20 to a bistable 12 which is also controlled by a switch 17 and a line 33 from microprocessor 14. The output of the bistable 12 is connected via line 21 to the microprocessor 14 and line 34 to an AND-gate 13. The output of shift register 9 is directly connected with the AND-gate 13 via line 22 and also has a line 32 to the microprocessor 14. Data from gate 13 passes via line 7 to peripheral device 6 and data from device 6 may be transmitted to the computer 1 via line 8, OR-gate 16 and line 3. The microprocessor 14 is also connected to OR-gate 16 via line 23.

It will be apparent from the foregoing that microprocessor 14 receives data from and supplies data to various components of the protection apparatus 5. The microprocessor is associated with a non-volatile memory 15 which stores the multi-digit serial number of the software, or other identifying data such as a serialised message of coded characters. A program running on the computer requires transmission of the serial number (or identifying data) from the protection apparatus 5 in order to continue running.

- 6 -

The operation of the apparatus 5 is as follows. A program running on computer 1 sends data via line 2 as a sequence of bits to the shift register 9, where the bits are held a byte (8 bits) at a time for the purposes of comparison with the 8 digit code in register 11. Bits are successively passed to register 9 for comparison and, in the absence of a match with the code in register 11, the data passes via line 22 to AND-gate 13. As long as there is no match between the contents of registers 9 and 11, bistable 12 continues to enable AND-gate 13 and the data passes via line 7 to the peripheral device 6. All data passing via line 22 to the peripheral device 6 can be read by microprocessor 14 which is connected to the output of register 9 via line 32.

The bistable 12 and AND-gate 13 function as a temporary blocking means. If the comparator 10 detects a match, the state of bistable 12 is changed and the AND-gate 13 is disabled. This prevents peripheral 6, which may for example be a VDU or a printer, from displaying or printing data which is part of the checking operation and is unrelated to the purpose of the program. Bistable 12 is connected via line 21 to the microprocessor 14 and the bistable 12 prompts the microprocessor to transmit the serial number stored in memory 15 to the computer via line 23, OR-gate 16 and line 3.

The proprietary program running on the computer is so written as to require transmission of the serial number to keep on running normally, that is to say to continue to run and perform its intended tasks instead of carrying out checks. The program

- 7 -

receives and checks the serial number from apparatus 5. If the number is incorrect, the program will cease running. In an alternative form, the program may be compiled to send appropriate messages to the peripheral device 6 if the correct serial number is not received, telling the user to check that the correct apparatus and program are being used in conjunction, or warning that the program in use is an unauthorised copy.

If the serial number is found to be correct, the program is allowed to continue running normally and data passes to the apparatus via line 2, and register 9. The data is read by the microprocessor 14 via line 32, the microprocessor causes a signal to be sent to bistable 12 via line 33 which enables AND-gate 13, previously disabled on detection of a match, allowing data to pass via line 7 to peripheral device 6 which can resume its display or printing of relevant data.

If it is desired to run a non-proprietary program on computer 1, the apparatus 5 need not be disconnected as the program is permitted to run and is unaffected by the apparatus. Data from the computer 1 passes via line 2 to register 9. It is unlikely that a match will be detected by comparator 10 but to prevent a spurious match disabling gate 13, switch 17 may be closed to cause bistable 12 to keep the gate 13 enabled, allowing data to pass directly from the register 9 to gate 13 via line 22 and thence to peripheral device 6 via line 7. Data from the peripheral device may pass via line 8 through the protection apparatus, emerging

from gate 16 on line 3 to reach computer 1.

A producer of proprietary computer software can protect the software from unauthorised copying by selling the software together with apparatus according to the invention, suitably a piece of hardware in a "black box". The program is compiled so as to require specific data for continued normal running and the apparatus of the present invention provides that data. In the embodiment described above, the program is written to include the code required by register 11 of apparatus 5 to provide a match, and the match causes microprocessor 14 to send the serial number of the program to the computer 1, this being the data required to cause the program to continue running. An unauthorised person who has copied the program but has not been supplied with the apparatus 5 cannot run the program past a certain point because the serial number will not be transmitted, thus causing the running to cease. A manufacturer need not go to considerable trouble and expense, as has been necessary in the past, to stop his programs from falling into the hands of the unscrupulous because copies of proprietary programs are unusable without associated protection apparatus according to the invention.

The apparatus described and illustrated is one example of an embodiment of the invention, which has been shown for convenience as comprising a number of hardware components. It will be appreciated that these could be replaced by fixed firmware in a dedicated microcomputer to perform similar functions. The code or

other data store which in the embodiment described above was register 11 could comprise a PROM. The incoming data from the computer could be stored for comparison purposes in a memory register and the comparison carried out by the firmware through a sequence of logical operations under the instructions of the microcomputer.

The apparatus of Figure 1 is a particularly simple example, and more sophisticated forms could provide a greater degree of software protection. For example, microprocessor 14, which can control register 11 via lines 24, may be programmed to modify the contents of the register once or several times after comparator 10 has detected a first match, the program running on the computer being programmed to supply the appropriate codes via line 2. Memory 15 may store a sequence of serial numbers or messages to be released in turn, after detection of successive matches by comparator 10, to the program which requires these numbers for the performance of a sequence of tasks or checks, necessary to allow continued running of the program and for the sending of a message via line 2 for the enabling of gate 13.

Gate 13, or an equivalent device, is optional and may be included if a peripheral device is running in series with the protection apparatus. If a peripheral device is running on a different line or there is no such device, lines 2 and 3 will communicate solely with apparatus 5, and gate 13, which serves to prevent data present purely for the purpose of checking from

reaching the peripheral device, would be redundant.

The versatility of the apparatus may be increased by carrying out the initial comparison on an arbitrary time base with repeated sampling so as to determine the baud rate of the data prior to checking the pattern of bits for a match. Once a match has been detected the apparatus may allow all data unconnected with the checking operation to be transmitted to a peripheral device running on the computer.

Where the apparatus is connected in an on-line mode on a communications port on a computer, it may act as a terminal concentrator for more than one output channel. The apparatus would then be provided with several interfaces such as interface 7, 8 with associated logic in the microprocessor. The apparatus is versatile because it is connected via a standard interface to the particular user's hardware and can still be used if the hardware is updated, in association with suitably modified software.

CLAIMS:

1. Apparatus capable of protecting proprietary computer software against unauthorised use, comprising storage means for storing selected data, comparison means for comparing data successively communicated by a program running on a computer with data from the storage means, means for storing identifying data and transmitting means for sending stored identifying data to the computer, whereby on detection of a match in use, the transmitting means send said identifying data to the computer, said data being required for continued normal running of a proprietary program.
2. Apparatus as claimed in claim 1, wherein said selected data comprises a code of data bits and said identifying data comprises a message of data bits and a plurality of the data codes and data messages are stored in said respective storage means for the data codes to be successively compared and said data messages to be successively transmitted, whereby a proprietary program on an associated computer will cease running normally unless a predetermined sequence of identifying data messages is received by the program from the apparatus.
3. Apparatus as claimed in claim 1 or claim 2, including a combined storage means for storing said selected data and said identifying data.
4. Apparatus as claimed in any preceding claim, including communications means for the transfer of data to and from a computer associated with the apparatus.

5. Apparatus as claimed in any preceding claim wherein the apparatus is adapted for communication with a peripheral device.
6. Apparatus as claimed in claim 5, comprising blocking means to prevent transfer of data to the peripheral device after detection of a match and until after said identifying data has been transmitted by said transmitting means.
7. Apparatus as claimed in claim 6, including switching means for preventing actuation of the blocking means, to allow the running of a non-proprietary program without interruption by the blocking means.
8. Apparatus as claimed in any preceding claim comprising a programmed microcomputer, wherein the storage means for the selected data and the identifying data comprise programmable read only memories and the comparison means is adapted to operate under the instructions of the programmed microcomputer by carrying out a sequence of logical operations to detect a match.
9. A method for protecting proprietary computer software against unauthorised use, comprising storing selected data and identifying data in protection apparatus associated with a programmed computer, successively communicating data from the computer to the apparatus, comparing said communicated data with said selected data and, on detection of a match, transmitting said identifying data to the computer, the identifying data being required by a proprietary program for continued normal running.

Requested Patent: EP0180460A1

Title: DECODER FOR A PAY TELEVISION SYSTEM ;

Abstracted Patent: US4759060 ;

Publication Date: 1988-07-19 ;

Inventor(s): HAYASHI TOSHIHIDE (JP); KANNO MASAYOSHI (JP) ;

Applicant(s): SONY CORP (JP) ;

Application Number: US19850793554 19851031 ;

Priority Number(s): JP19850110001 19850521; JP19840229348 19841031 ;

IPC Classification: ;

Equivalents: CA1261960, DE3579785D ;

ABSTRACT:

A decoder for a pay television system incorporates a receiver for receiving scrambled programming and control data transmitted from a remote location, with the control data including data corresponding to the program fee and the program status, and a manual switch for selectively descrambling the program data when the program status is a pay-per-view program. The decoder has a storage unit for storing a credit value, transmitted from the remote location, and the decoder has timer apparatus for measuring the time during which a pay-per-view program is being received and for subtracting a program fee periodically from the value stored in the storage unit, during the time of reception of the selected program, provided the switch is operated to descramble the program.

10



Europäisches Patentamt
European Patent Office
Office européen des brevets

11 Publication number:

**0 180 460
A1**

12

EUROPEAN PATENT APPLICATION

21 Application number: 85307832.7

51 Int. Cl.: H 04 N 7/16

22 Date of filing: 29.10.85

30 Priority: 31.10.84 JP 229348/84
21.05.85 JP 110001/85

71 Applicant: SONY CORPORATION,
7-35 Kitashinagawa 8-Chome Shinagawa-ku,
Tokyo 141 (JP)

36 Date of publication of application: 07.05.86
Bulletin 86/19

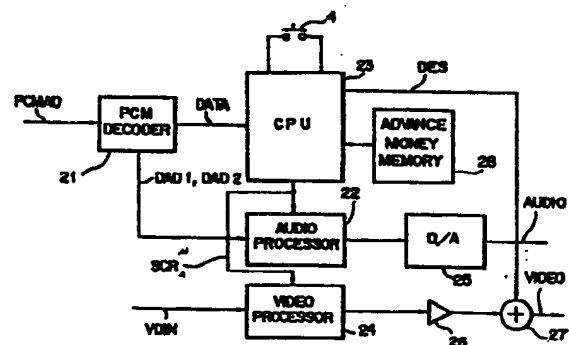
72 Inventor: Hayashi, Toshihide c/o Patents Division, Sony
Corporation 8-7-35 Kitashinagawa, Shinagawa-ku
Tokyo 141 (JP)
Inventor: Kanno, Masayoshi c/o Patents Division, Sony
Corporation 8-7-35 Kitashinagawa, Shinagawa-ku
Tokyo 141 (JP)

54 Designated Contracting States: DE FR GB NL

74 Representative: Thomas, Christopher Hugo et al, D
Young & Co 10 Staple Inn, London WC1V 7RD (GB)

64 Decoders for pay television systems.

17 A decoder control circuit for a pay television system includes a descrambler (22, 24) for descrambling a selected video programme transmitted from a broadcast centre (9), a decoder (21) for receiving and decoding control data transmitted from the centre (9), the control data including at least programme feed data and programme status data, an advance money memory (28) for storing money data transmitted from the centre (9), and a central processing unit (23) for subtracting a programme fee from the money data stored in the advance money memory (28) upon reception of the selected programme.



EP 0 180 460 A1

ACTORUM AG

DECODERS FOR PAY TELEVISION SYSTEMS

This invention relates to decoders for pay television systems.

Broadcasting systems include cable television (CATV) systems and direct-broadcasting satellite (DBS) systems. These are frequently used for pay television systems, using an appropriate converter.

5 In a pay television system of this type, a conventional decoder for decoding a programme of a desired channel has two main status modes; a free mode and a pay mode. The free mode allows free reception of a television programme, and the pay mode represents a status mode in which a subscriber is charged for reception of a television programme.

10 More particularly, the pay mode is further classified into a flat fee mode, a tier level pay mode and a pay-per-view mode. In the tier level pay mode, a tier level representing the rank of programmes to be received by the decoder is predetermined. A user subscribes to a desired tier level and pays fees to a broadcast station or centre, corresponding to the subscribed
15 tier level. In the pay-per-view mode, the user reserves desired programmes, and pays the programme fee only for the reserved programmes to a broadcast centre.

In the pay-per-view mode, user procedures are cumbersome. When the user wishes to watch a pay programme in the pay-per-view mode
20 according to a first known procedure, the broadcast centre posts a programme schedule to each user. The user telephones the broadcast centre by a predetermined date before a desired programme is to be broadcast (say a week or a day beforehand) so as to reserve the desired programme. At this time, the user pays the fee for the programme. When the desired
25 programme is on the air, the broadcast centre sends a reservation confirmation signal, hereinafter referred to as an identification (ID) signal, to a user who has reserved the corresponding programme. When a user's receiver or a decoder in the receiver receives the ID signal, a scrambled programme signal is descrambled by the decoder, so that the user can watch
30 the programme.

In the pay-per-view system described above;

(A) the user must telephone the broadcast centre to reserve the desired programme, which is time-consuming and may be inconvenient,

5 (B) each programme has a reservation due date, so that the user cannot reserve the desired programme when the reservation due date has passed,

(C) fees cannot be refunded even if the user does not watch the reserved programme, and

10 (D) an idle time is required to send the ID signal to all reserved users at the beginning of every reserved programme.

According to a second, improved procedure, a pay-per-view status signal is sent from the broadcast centre to each user. When the user wishes to watch a programme represented by the pay-per-view station signal, he depresses a pay-per-view switch located on his tuner. The scrambled
15 programme is then descrambled, and the user can watch the programme. When the user actually watches the programme, charge data transmitted with the subscribed programme is stored in an account memory of the decoder. The centre periodically checks the contents of the account memory of each decoder, using a telephone line, and collects fees or bills
20 the applicable charge.

This effectively solves the drawbacks of the first procedure. However, since the broadcast centre must periodically check the account memories of all users, the checking system is complicated. Moreover, since fee collection is performed by use of a telephone line, an auto-dial unit and
25 a modem (modulator/demodulator) are required, so that the required user unit and centre unit are complex and expensive.

According to the present invention there is provided a decoder control circuit for a pay television system, the decoder control circuit comprising:

30 means for descrambling a selected video programme transmitted from a centre at a remote location;

characterised by:

means for receiving and decoding control data transmitted from said centre, said control data including at least programme feed data and programme
35 status data;

an advance money memory for storing money data transmitted from said centre; and

control means for subtracting a programme fee from the money data stored in said advance money memory upon reception of said selected programme.

5 According to the present invention there is also provided a decoder for a pay television system, the decoder comprising an advance account memory for storing deposit data sent from a broadcast centre, means for subtracting programme fees from the contents of the advance account memory for every reception of a pay programme, and means for displaying a
10 warning display when the contents of the advance account memory equals less than a predetermined value.

 The pay-per-view mode may be classified into two sub-modes to prevent unreasonable payment. A first sub-mode constitutes a time pay-per-view for determining the charge per unit of time, and the second sub-
15 mode constitutes a programme pay-per-view mode for determining a charge per programme. The tuner decodes a programme status code signal sent from the broadcast centre and selects either the time pay-per-view sub-mode or the programme pay-per-view sub-mode.

 The broadcast centre can determine which sub-mode applies to
20 programmes of different sorts, without interfering with the user's ability to view programmes. For instance, the programme pay-per-view mode may be employed for a special sports programme such as a boxing match programme, while other programming employs the other sub-mode. In this way a reasonable charge for any programme can be made.

25 The invention will now be described by way of example with reference to the accompanying drawings, throughout which like parts are referred to by like references, and in which:

 Figure 1 is a schematic diagram of a pay-per-view system applied to a DBS utilizing a broadcast satellite;

30 Figure 2 is an illustration of a data format of a pulse code modulation (PCM) data signal used in the DBS;

 Figure 3 is a block diagram of a tuner unit shown in Figure 1;

 Figure 4 is a flow chart illustrating the operational mode of the tuner shown in Figure 1;

Figure 5 is a flow chart illustrating the pay-per-view mode of the tuner shown in Figure 1; and

Figure 6 is a flow chart illustrating a control programme continuously executed by the central processing unit (CPU) of the tuner.

5 In the system of Figure 1, a broadcast wave from a broadcast satellite 13 is directly received by a receiving parabolic antenna 2 and supplied to a DBS tuner 1. Audio and video signals reproduced by the DBS tuner 1 are supplied to a television monitor 14. An unscrambled broadcast signal is reproduced by a decoder in the tuner 1 in the free mode. However,
10 in the pay mode or the pay-per-view mode, a scrambled broadcast signal is descrambled and reproduced on the television screen.

 When the pay-per-view mode is utilized by the user, the user deposits funds using a cash card 8 or the like, to advance money from an account within a bank 6 to a broadcast centre 9. The cash card 8 is conveniently
15 used with an automatic teller machine (ATM) (not shown). The deposit data is transmitted from a computer 7 of the bank 6 to a computer 10 of the broadcast centre 9. The desired programme is transmitted from the broadcast centre 9 to the user through a satellite broadcast link. The deposit data is inserted in the data coded in a PCM data signal transmitted
20 during a vertical blanking period of the video signal. The deposit data is transmitted from a transmitter 11 to the antenna 2 of each user through a broadcast parabolic antenna 12 by way of the satellite 13.

 The tuner 1 of each user has an advance money memory, and the deposit data is stored therein. The content of the storage can be displayed
25 at any time on a display 5.

 The broadcast centre 9 transmits a television signal having a PCM data signal PCMAD with the format shown in Figure 2, during a vertical blanking period of the video signal. The PCM audio signal PCMAD incorporates a plurality of data slots for data of different types. Two slots
30 correspond to two (alternately used) channels of audio data DAD1 and DAD2. Also included are a slot for the deposit money data DCIN (representing the amount of the deposited fees) and a slot for address data DADD, including the user identification (ID) code. A data code field DDCCD is added to the above data string, and an error check code field DBCH is
35 also added to the end of the data string.

The data code slot DDCD contains a frame sync code FSYN as the initial data thereof. The following data consist of the mode data word MODE, a programme status code word PROM, a channel code CHCD representing a broadcast channel number, first and second range bits REG1 and REG2 used to expand the compressed audio data, a scramble sync code SCRM used for decrambling the signal, a charge code CHAG representing the programme fees, a data code DATE, and user bits USBT.

The programme status code PROM comprises a 4-bit code signal representing the pay mode assigned to the current broadcast programme. For example, in one embodiment the programme status code PROM is set to be "0000" in the free mode and "0001" in the pay mode.

The pay-per-view mode is further classified into a time pay-per-view mode represented by a code "0101" and a programme pay-per-view mode represented by a code "0110".

The free and pay mode are, respectively, a mode for allowing free reception of programmes and a mode for subscription programmes on a monthly basis in the same manner as in the known system. The time pay-per-view sub-mode is set to charge for the length of listening time, and the programme pay-per-view sub-mode is set to charge predetermined fees for the programme regardless of reception time.

The broadcast centre 9 transmits with each programme the programme status code PROM added to the data code DDCD. A proper pay mode is predetermined in accordance with the contents of the programme by the broadcast centre 9, so that the appropriate programme status code PROM is added to the data code DDCD, and the resulting broadcast signal is transmitted to the user. Thus, fees can be charged in accordance with charging levels suitable for the respective programmes.

In the tuner 1 which receives the broadcast signal, the PCM data signal PCMA is supplied to a PCM decoder 21 (Figure 3), located within the tuner 1 (Figure 1). The PCM decoder 21 decodes the PCM audio signal PCMA to extract the first and second channel audio data DAD1 and DAD2 (Figure 2) which are supplied to an audio processor 22. The other data DATA are supplied to a central processing unit (CPU) 23 which is preferably a microcomputer. A video signal VDIN (which may be scrambled) in the broadcast signal is supplied to the input of a video processor 24, which is arranged to unscramble the video signal if necessary.

As shown in Figure 4, a step SP1 inspects the code word PROM and controls the subsequent operation accordingly. When the programme status code PROM represents the time pay-per-view mode or the programme pay-per-view mode (step SP1) a pay-per-view programme SP2 runs under the control of the CPU 23 to perform a processing step SP3.

However, when the programme status code PROM represents the free mode, a free mode programme SP4 is executed. In this case, the CPU 23 causes the audio processor 22 to decode the audio data DAD1 and DAD2 and the video processor 24 to decode the video input signal VDIN.

When the programme status code PROM represents the pay mode, a pay programme SP5 is executed under control of the CPU 23. The CPU 23 determines in a step SP6 whether or not the user ID signal included in the data DATA is the same as a unique user ID assigned to the tuner 1. If YES in the step SP6, the CPU 23 supplies the scramble sync code SCRM to the audio and video processors 22 and 24, so as to cause them to perform descrambling of the audio data DAD1 and DAD2 in a step SP7. Therefore, the audio and video signals are both reproduced. However, if the result is NO in the step SP6, the CPU 23 does not supply the scramble sync code SCRM to the audio and video processors 22 and 24, so as not to cause them to perform descrambling of the audio and data DAD1 and DAD2.

The audio signal reproduced by the audio processor 22 is supplied to a digital-to-analog (D/A) converter 25. The D/A converter 25 generates an audio output signal AUDIO. The video signal reproduced by the video processor 24 is supplied to a buffer amplifier 26. A video output signal VIDEO is generated as the sum output of an adder 27.

In the pay-per-view mode, the CPU 23 performs the processing step SP3, as shown in Figure 5. When the processing step SP3 is started, the CPU 23 checks in a step SP11 whether or not a pay-per-view switch 4 is turned on.

As shown in Figure 1, the pay-per-view switch 4 is mounted on the operation panel of the tuner 1. After the user has entered a desired channel with a ten-key pad 3, he depresses the pay-per-view switch 4, if the programme is a pay-per-view programme, which sets the tuner 1 in the pay-per-view mode. The reception channel number is displayed on the display 5. The above sequential operations are performed under the control of the CPU 23.

If the result of the step SP11 is NO (Figure 5), the CPU 23 determines that the user has to set the tuner 1 in the pay-per-view mode. Control then advances to a step SP12, and no descrambling is performed. Thereafter, the flow returns repeatedly to the step SP11. The CPU 23 thus
5 waits until the user turns on the pay-per-view switch 4.

When the pay-per-view switch 4 is turned on by the user, the result of the step SP11 is YES. The CPU 23 then supplies the scramble sync code SCRM to the audio and video processors 22 and 24 for descrambling. The CPU 23 thus controls the generation of both the audio and video output
10 signals AUDIO and VIDEO.

When the content of the programme status code PROM represents the time pay-per-view mode, the CPU 23 fetches this data in a step SP14, and the flow advances to a step SP15.

The CPU 23 checks in the step SP15 the length of viewing time in the
15 time pay-per-view mode. More specifically, the CPU 23 checks whether or not a unit time has been counted by a timer incorporated in the CPU 23. If the result is NO in the step SP15, the control flow returns to the step SP11. The CPU 23 waits until counting of the unit time by the timer is completed by a loop of the steps SP11, SP13, SP14, SP15 and SP11.

20 When the unit time has elapsed, the result of the step SP15 is YES. The control flow then advances to a charge step SP16, in which one unit time fee is subtracted from the contents of an advance money memory 28 (Figure 3). This may be a fixed quantity, or a quantity designated by the CHAG field of the PCM data. The control flow returns to the step SP11
25 again to reset the timer. The CPU 23 waits for the predetermined unit time to elapse by the loop of steps SP11, SP13, SP14, SP15 and SP11.

In the same manner as described above, the CPU 23 sends out the audio and video output signals AUDIO and VIDEO which are descrambled and reproduced by the audio and video processors 22 and 24 while the time pay-
30 per-view mode programme is being received by the user. A fee corresponding to the viewing time of the programme is subtracted from the contents of the advance money memory 28 in the step SP16.

When the user wishes to stop receiving the programme in the time pay-per-view mode, he merely turns off the pay-per-view switch 4. In this
35 case, the step SP11 is determined by the CPU 23 to be NO, and the CPU 23 then causes the audio and video processors 22 and 24 to disable descrambling by the loops of steps SP11, SP12, SP11 ... , etc.

When the user wishes to watch programmes in the programme pay-per-view mode, the user simply turns on the time pay-per-view switch 4 of the tuner 1 in the same manner as in the time pay-per-view mode. In this case, after the step SP11 has been determined by the CPU 23 to be YES, the
5 CPU 23 controls the audio and video processors 22 and 24 to perform descrambling in the step SP13.

In this case, the programme status code PROM included in the data DATA represents the programme pay-per-view mode detected by the CPU 23 in a step SP21. The control flow advances to a decision step SP22. The
10 CPU 23 determines in the step SP22 whether the current programme being received is the same as the programme being received when the step SP22 last had control or whether the programme being received has been changed. If the result is NO in the step SP22, the CPU 23 determines that the same programme is being received. Then the control flow returns to the step
15 SP11. The user can continuously watch the programme in the programme pay-per-view mode by a loop of the steps SP11, SP13, SP21, SP22 and SP11.

When the user wishes to change the programme being received, the step SP22 is determined by the CPU 23 to be YES. Then a charge is subtracted from the contents of the advance money memory 28 at the step
20 SP23, and the flow returns to the step SP11.

Once the programme has changed, the mode is set to continue the changed programme. The CPU 23 monitors this programme by the loop of the steps SP11, SP13, SP21, SP22 and SP11.

Every time one programme has been completed or finished, the CPU
25 23 performs charge processing, so that the user is charged per programme, irrespective of the elapsed time.

When the user operates the pay-per-view switch 4, the mode of the tuner 1 is set to be either the time pay-per-view mode or the programme pay-per-view mode. The CPU 23 supplies to an input of the adder 27 (Figure
30 3) a display signal DES representing that the current programme is set in the time or programme pay-per-view mode. For example, numerical value "1" representing the time pay-per-view mode or numerical value "2" representing the programme pay-per-view mode is displayed on the screen of the monitor 14. The user can easily see the charging mode being used for
35 the current programme.

Figure 6 illustrates a flow chart of operations continuously checked by the CPU 23. Normally, control stays in one of the loops illustrated in Figure 5. These loops are represented in Figure 6 by a fetch charge code step SP31, which passes control through a return step SP38. The return step SP38 returns control to a step SP30 which normally passes control to the fetch charge code step SP31, to define the loop of Figure 5 which is active (for the appropriate sub-mode). Periodically, the check routine step SP30 sends control to two other paths illustrated in Figure 5. Preferably this is accomplished by a signal from a timer of the CPU 23 which interrupts the normal control loop, to check the balance data maintained in the advance money memory 28 at periodic intervals, or to update the data stored in the advance money memory 28 in response to detection of data within the DCIN slot of PCMAD (Figure 2). In the loops represented by the step SP31, the charge code data CHAG superimposed on the programme by PCMAD is detected by the CPU 23, and, as previously described, when the programme or channel is changed, the step SP31 is executed.

A step SP32 receives control periodically by a timer interrupt. Alternatively, execution of the charge step SP16 or SP23 (Figure 5) may include the setting of a flag which causes the step SP30 to pass control to the step SP32. In the step SP32, the CPU 23 compares the balance of the advance money memory 28 with the charge code CHAG fetched by the step SP31. When the CPU 23 determines that the balance of the advance money memory 28 is smaller than the data represented by the charge code CHAG, the CPU 23 causes the monitor 14 to display a message representing a request for deposit in a step SP37. In this case, the CPU 23 inhibits the SCRM signal, so that the programme cannot be watched. The deposit data is checked in a step SP33. The step SP30 passes control to the step SP33 periodically, by a timer interrupt, so that the control data PCMAD may be checked for deposit information. Alternatively, the step SP30 may check each incoming control word PCMAD for deposit data and branch to the deposit routine step SP33 whenever data is detected within the DCIN field. When deposit money data DCIN is detected as present in the step SP33 (Figure 6), the CPU 23 checks the ID code included in the address data DADD. In this case, if the result is YES in a step SP34, the deposit money data DCIN is added to the content of the advance money memory 28 in a step 36. When all the routines have been completed in Figure 6, the flow

returns from the step SP38. Preferably, the advance money memory 38 comprises a non-volatile memory device such as an MNOS semiconductor RAM.

5 With the above arrangement, the pay-per-view mode for designating a pay programme is classified into time and programme pay-per-view sub-modes. Charging is determined in units of time or programmes. Therefore, a reasonable charging system suitable for the purposes of users can be established.

10 A desired programme need not be reserved by telephone, and the user can conveniently enjoy the DBS system. If the user does not watch a reserved programme, no charge is made. Moreover, when the pay-per-view switch 4 is depressed, the user can watch any programme at any time, provided the sum stored in the advance money memory 28 is enough.

15 Since the user is not charged through a telephone line (bidirectional communication system), the configuration of the user's receiver and the broadcast centre system can be simplified and made less expensive. The user simply receives a message via the step SP37 representing a request for a deposit when the value of the advance account memory is less than the predetermined value which is needed for viewing a particular programme.
20 The user pays a fee in accordance with the message, or else has it automatically deducted from the advance money memory 28, so that a simple efficient system can be provided. In the pay-per-view system of this embodiment, the transmission line from each user to the centre comprises a link through a bank or the like. Such a link may be located at the user's
25 home, or a public link may be used such as the link of a bank's ATM. Programme fee data, deposit amount data, and the like are sent from the broadcast centre 9 to the user through a transmission line using the regular broadcast channel as shown in Figure 1. Shortage of a deposit amount is automatically displayed by the user's monitor 14. Therefore, a
30 conversational two-way communication between the user and the broadcast 9 is provided, although only a one-way line terminating at the user's home is actually used.

35 The broadcast centre 9 may, but need not, send the ID signal to all users when payment data is obtained. Many control words relative to payment data can be sent to users within a limited time, and additional information such as a text or message can be sent, if desired.

The above embodiment exemplifies a DBS system. However, the present invention can be applied to other pay broadcast systems such as a CATV system. The invention can also be applied to an audio pay broadcast system.

5

Alternatively, in the above embodiment, the fee corresponding to the actual viewing time detected by the timer incorporated in the CPU 23 can be subtracted from the contents of the advance account memory periodically, by means of a timer interrupt procedure, which is enabled during the course of a pay mode.

CLAIMS

1. A decoder control circuit for a pay television system, the decoder control circuit comprising:
means (22,24) for descrambling a selected video programme transmitted from a centre (9) at a remote location;
5 characterised by:
means (21) for receiving and decoding control data transmitted from said centre (9), said control data including at least programme feed data and programme status data;
an advance money memory (28) for storing money data transmitted from
10 said centre (9); and
control means (23) for subtracting a programme fee from the money data stored in said advance money memory (28) upon reception of said selected programme.
- 15 2. A decoder control circuit according to claim 1 wherein said control means (23) includes a manual switch (4) for selectively allowing descrambling of said programme when said programme is a pay-per-view programme, as represented by pay-per-view status data in said control data.
- 20 3. A decoder control circuit according to claim 1 wherein said control means (23) includes means to inhibit the reception of said programme when said programme fee data exceeds the money data stored in said advance money memory (28).
- 25 4. A decoder control circuit according to claim 3 wherein said control means (23) further includes display means (14) for displaying a message giving reasons for inhibiting the reception of said programme.
- 30 5. A decoder control circuit according to claim 1 wherein said advance money memory (28) includes means for verifying an identification code transmitted with said control data and for selectively prohibiting the storing of said money data when said identification data transmitted in association with said money data does not coincide with the individual identification code of the decoder with which said control circuit is associated.

6. A decoder control circuit according to claim 2 wherein said control means (23) includes timer means arranged to measure the time during which said programme is received, and for controlling said subtraction based on the measure of reception time.

5

7. A decoder control circuit according to claim 6 including means for activating said timer means when said manual switch (4) is turned on, and for deactivating said scrambling when said manual switch (4) is turned off during the period of reception of a time pay-per-view programme.

FIG. 1

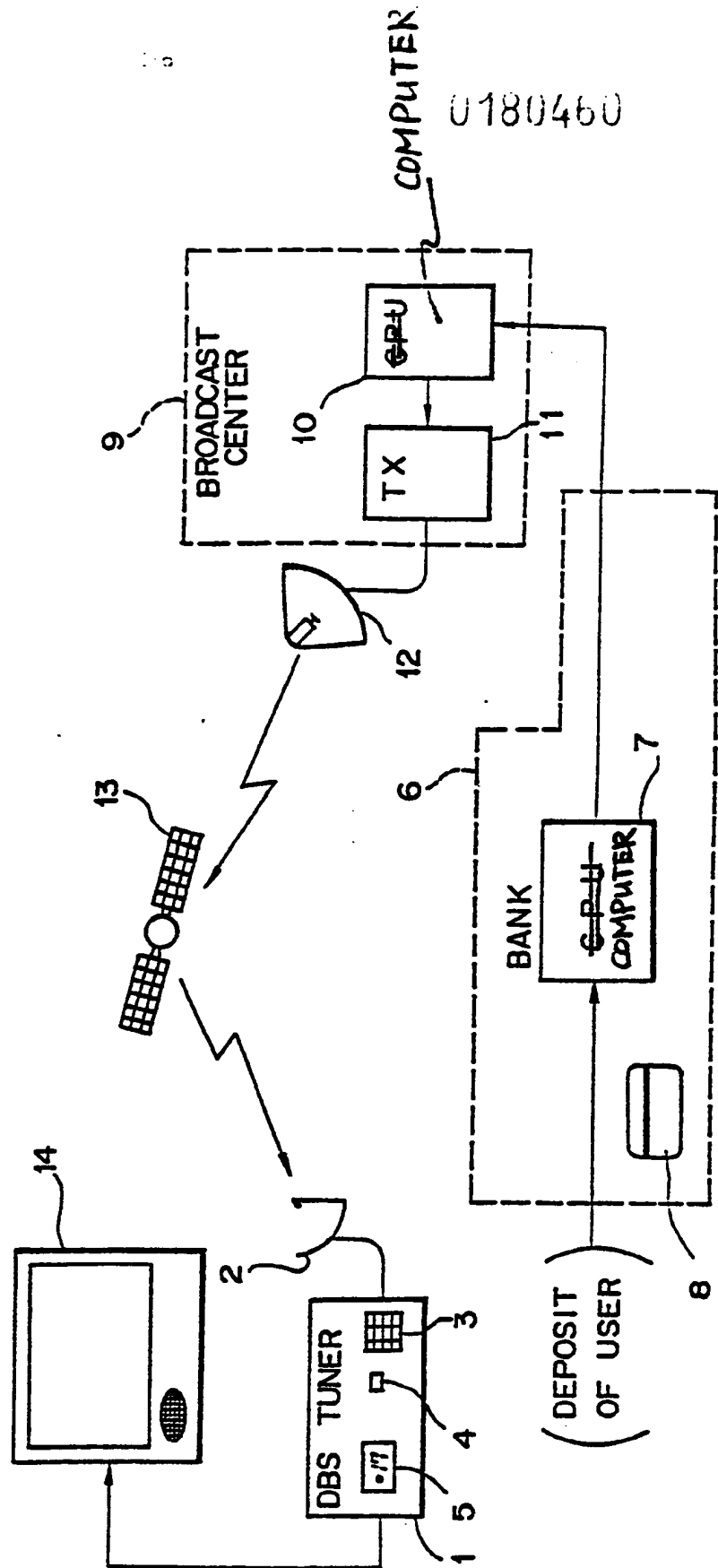


FIG. 2

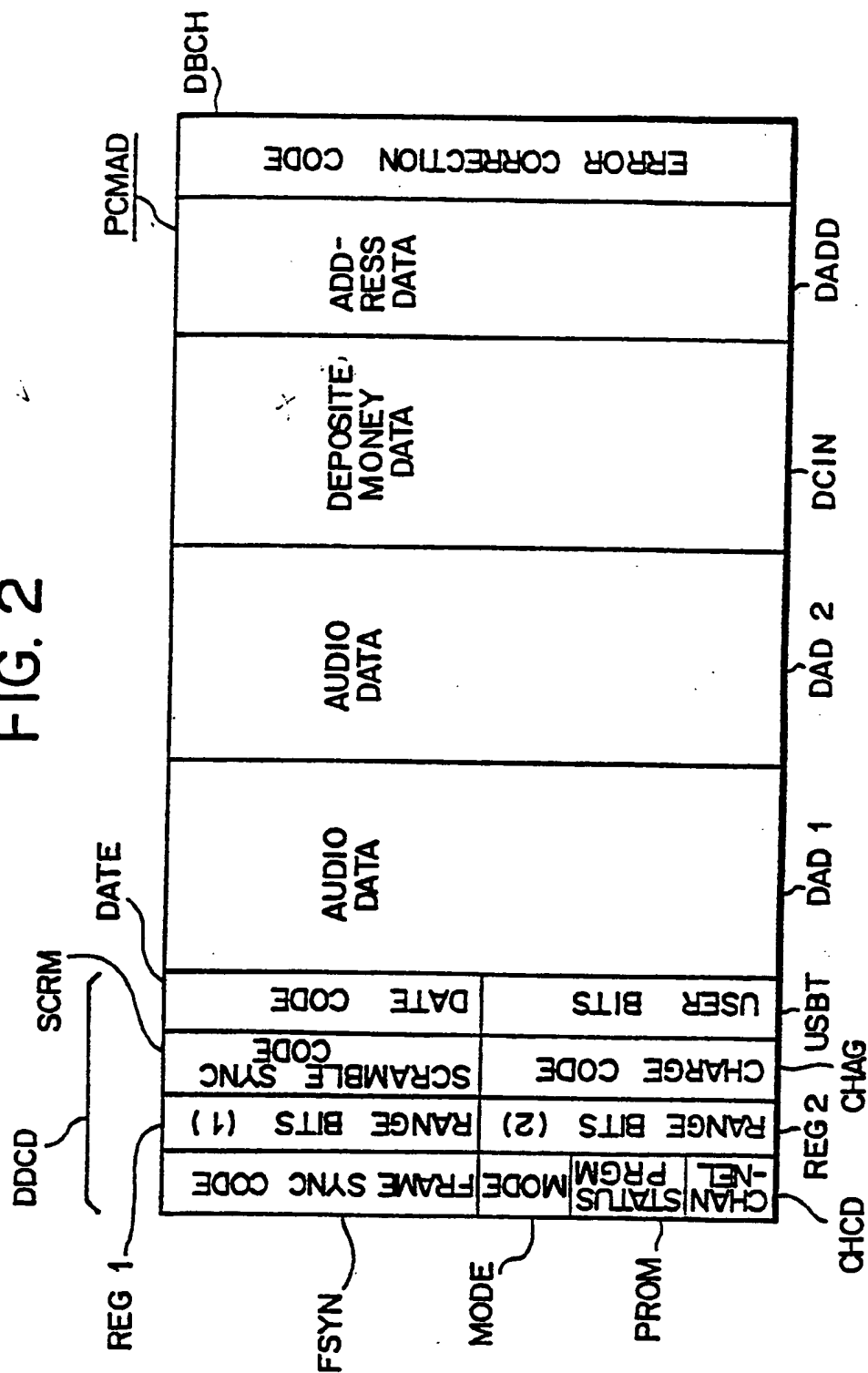


FIG. 3

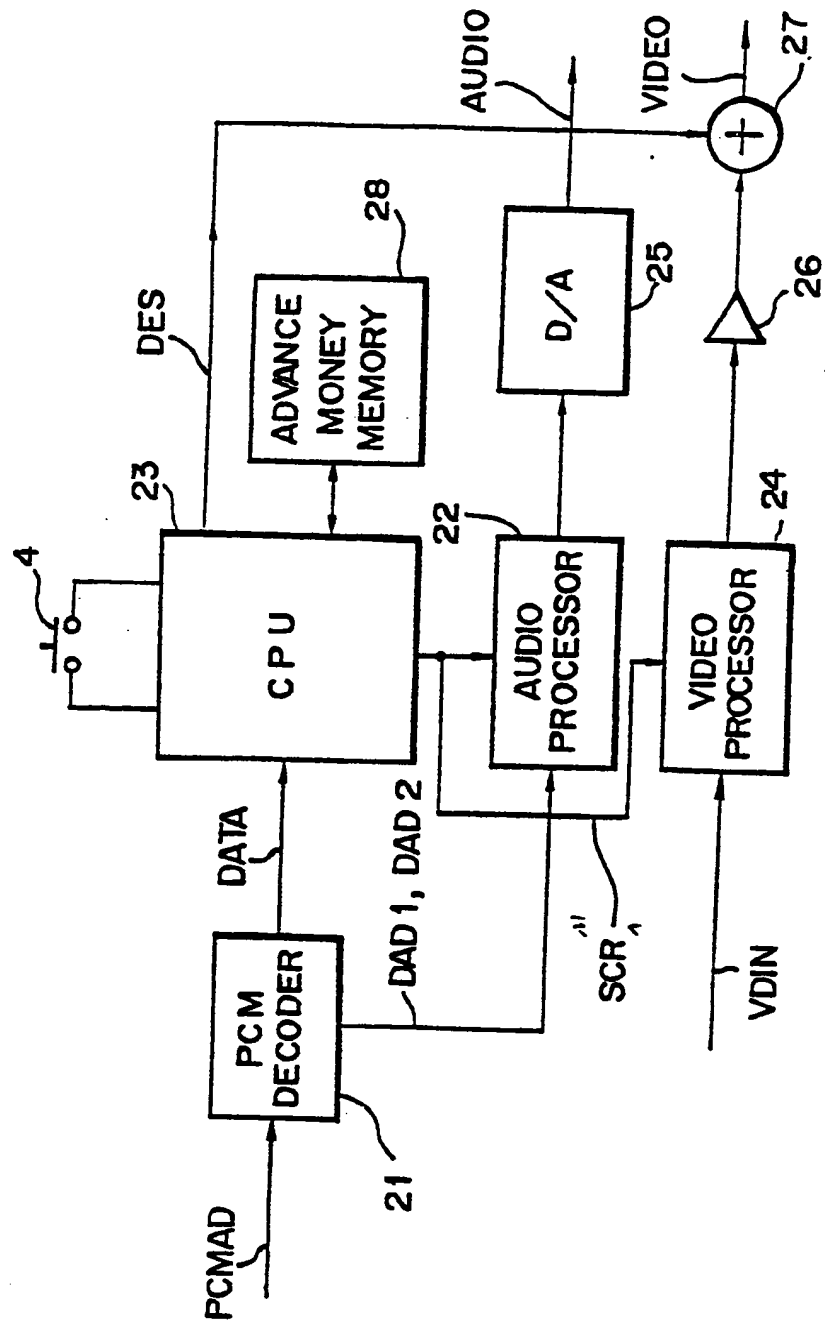


FIG. 4

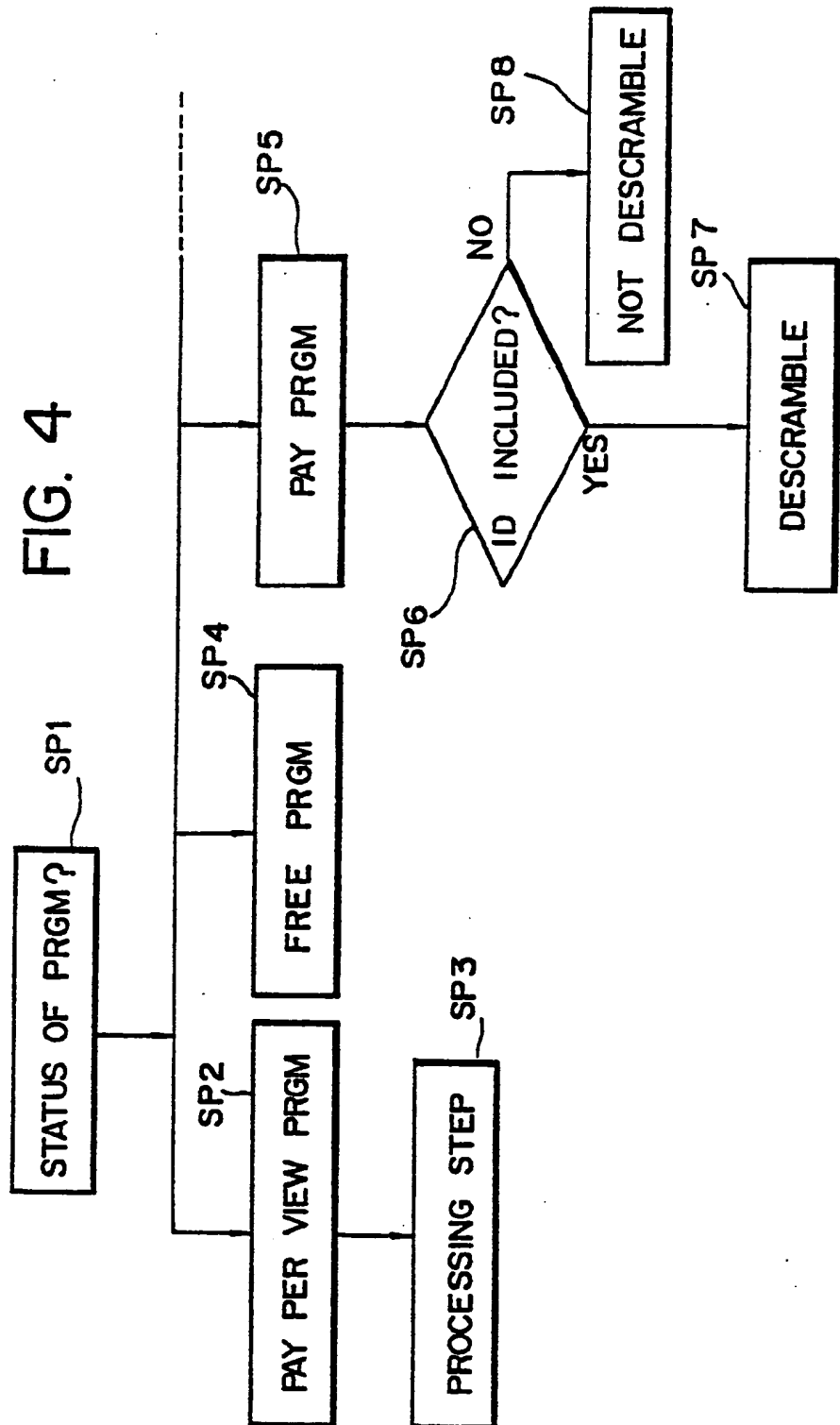


FIG. 5

0180460

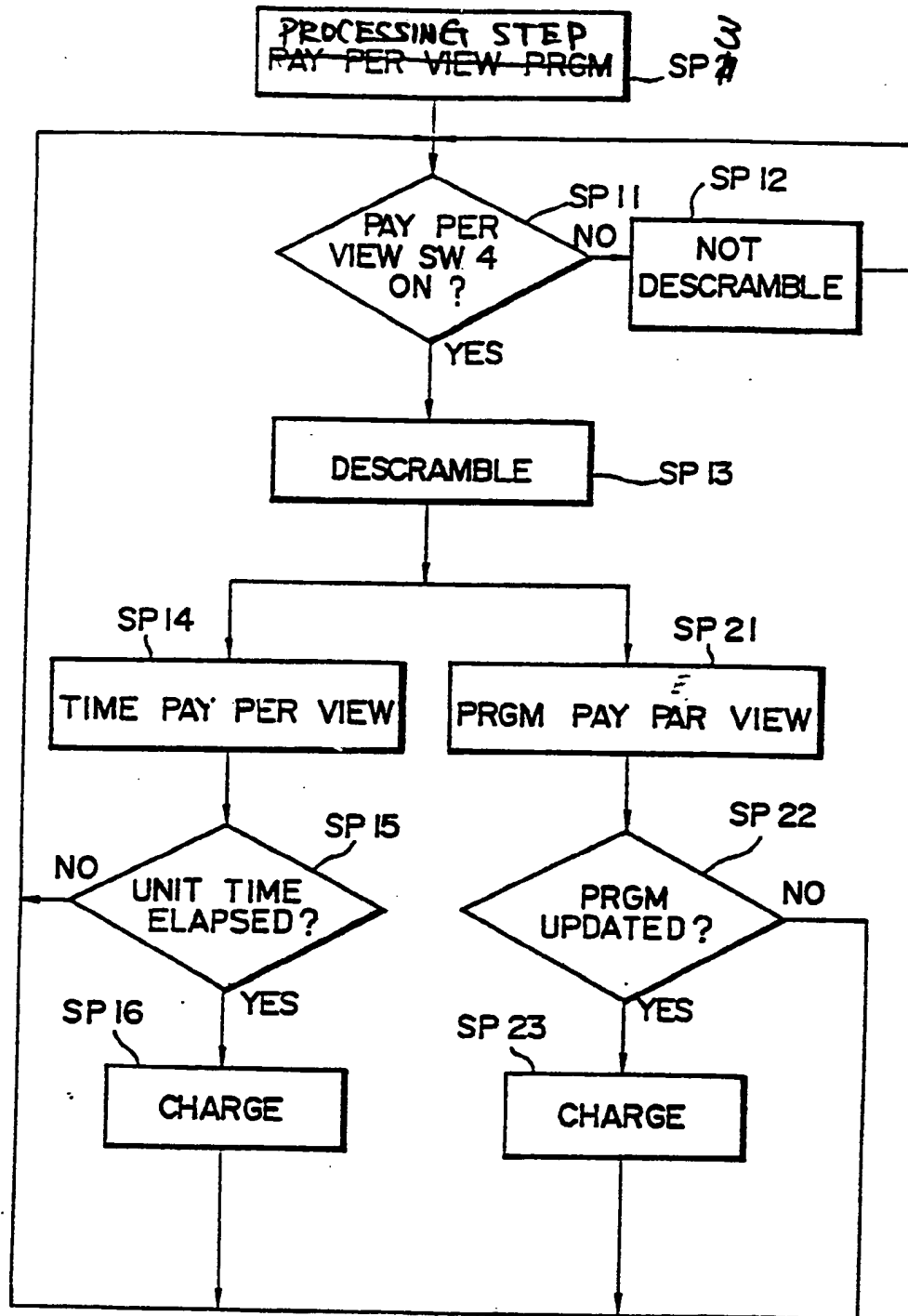
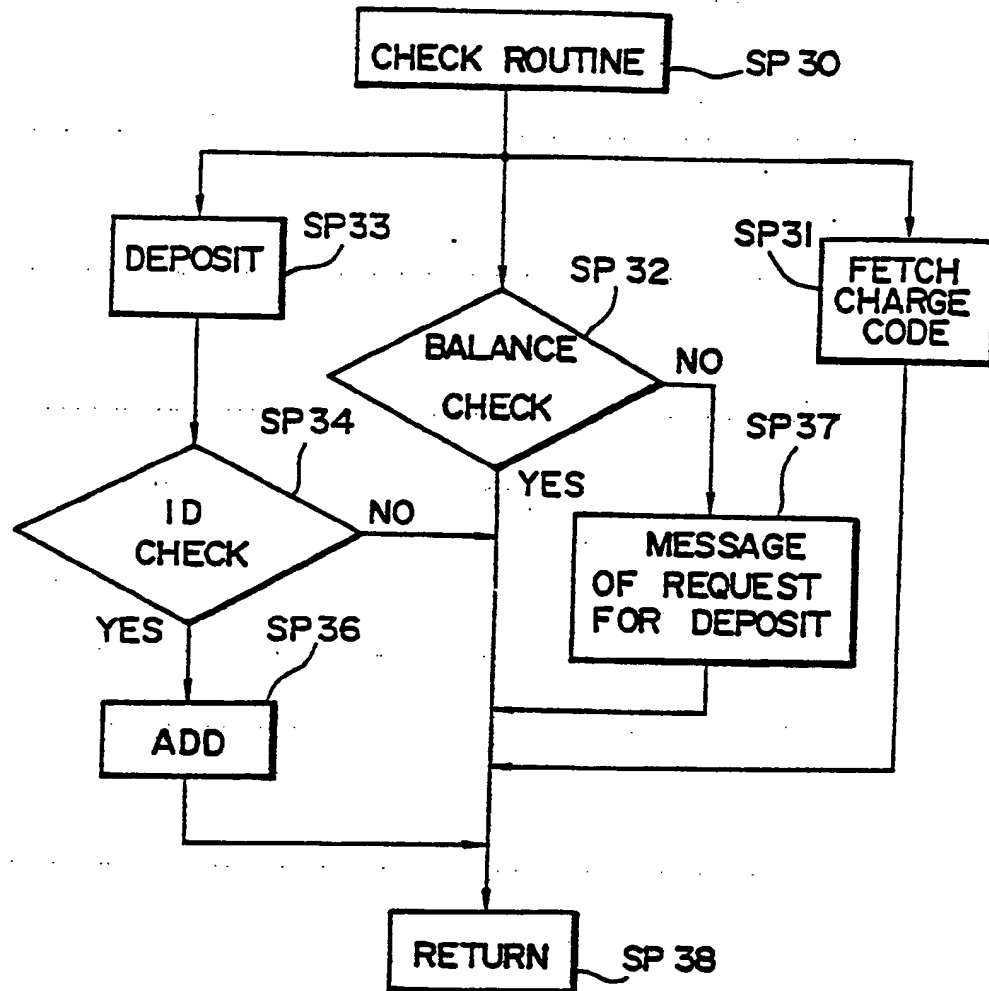


FIG. 6





EUROPEAN SEARCH REPORT

Application number

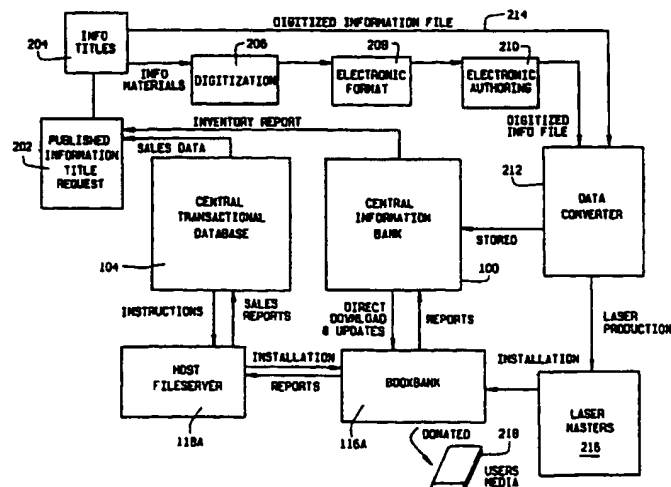
Petitioner Apple Inc. - Exhibit 1012, p. 574



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : H04L 9/32		A1	(11) International Publication Number: WO 99/49615
			(43) International Publication Date: 30 September 1999 (30.09.99)
(21) International Application Number: PCT/US98/22238		(81) Designated States: AL, AM, AT, AT (Utility model), AU (Petty patent), AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, CZ (Utility model), DE, DE (Utility model), DK, DK (Utility model), EE, EE (Utility model), ES, FI, FI (Utility model), GB, GE, GH, GM, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (Utility model), SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).	
(22) International Filing Date: 21 October 1998 (21.10.98)			
(30) Priority Data:			
09/049,321	27 March 1998 (27.03.98)	US	
09/175,559	20 October 1998 (20.10.98)	US	
(63) Related by Continuation (CON) or Continuation-in-Part (CIP) to Earlier Application			
US	Not furnished (CIP)		
Filed on	Not furnished		
(71) Applicant (for all designated States except US): MICROTOME, INC. [US/US]; 150 S. Price Road, St. Louis, MO 63124 (US).		Published With international search report.	
(72) Inventor; and			
(75) Inventor/Applicant (for US only): SAIGH, Michael, M. [US/US]; 150 S. Price Road, St. Louis, MO 63124 (US).			
(74) Agents: BEULICK, John, S. et al.; Armstrong, Teasdale, Schlafly & Davis, Suite 2600, One Metropolitan Square, St. Louis, MO 63102-2740 (US).			

(54) Title: INFORMATION DISTRIBUTION SYSTEM



Content Presentation

For
specific
user

(57) Abstract

An information distribution system, in accordance with one form of the present invention, includes a central information bank (100) and a central transactional data base (104) coupled to point-of-sale delivery systems (202). Information flows between each point-of-sale delivery system and the central information bank and central transactional data base via communication network such as the telephone network, a satellite network, or any other network suitable for the transfer of information. The point-of-sale delivery systems may take one of many forms including a point of purchase delivery system, a point of rental delivery system, a "book bank" (116A) subsystem, a promotional delivery system, or any combination of such systems. Information exchanged within the system is controlled using various levels of encryption and is monitored to prevent unauthorized exchange.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

-1-

INFORMATION DISTRIBUTION SYSTEMField of the Invention

The present invention relates to a system for distributing information in electronic form and more particularly, relates to a communication network for transmittal information between a central information bank and a user interface.

Background of the Invention

With the current information publishing and distribution system, information usually is delivered as fixed printed images on paper or fixed in other media forms such as tapes, diskettes, cartridges, laser disk, or compact disk. Prior to and at various points in the delivery process, the information usually is warehoused. Eventually, the information is delivered to retail outlets scattered throughout a distribution territory. Upon receipt of the information, the retailers either store the information or display the materials for resale.

The present publishing and distribution system has many disadvantages. One disadvantage is the amount of time and labor required for preparing, printing, producing and distributing information. Another disadvantage of the current system is the lack of control over the production quantity of information, i.e., the number of copies made and sold. The current distribution system is further disadvantaged by the time, work and costs required in publishing and distributing information updates.

In an attempt to improve the dissemination of some types of information, bulletin board networks have been established. Networks, such as Internet, also have been or are being established. Known networks generally utilize a telephone network or some other network as a communication media and can be accessed using commercially available software and almost any type of computer. As presently operated, however, such networks are unsuitable for the distribution of proprietary information and information which is intended for limited copying. The

-2-

free transfer of information using such networks provides little or no protection for copyright and proprietary information owners.

Summary of the Invention

The present invention, in one aspect, is an information distribution system
5 which overcomes disadvantages and shortcomings of the current information publishing and distribution system. An information distribution system, in accordance with one form of the present invention, includes a central information bank and a central transactional data base coupled to point-of-sale delivery systems. The central information bank and central transactional data base do not necessarily
10 have to be co-located and can be implemented on different, but coupled, computer systems. Information flows between each point-of-sale delivery system and the central information bank and central transactional data base via a communication network such as the telephone network, a satellite network, or any other network suitable for the transfer of information.

15 More specifically, information obtained from publishers is digitized, i.e., converted to an electronic form, to create a master copy in a uniform electronic format. Information obtained from publishers in digitized format simply is converted into the uniform electronic format. The master copies are stored in the central information bank.

20 The central transactional data base performs a record keeping function. Particularly, the transactional data base records and stores information related to each transaction performed at each point-of-sale site. Upon request, the transactional data base transmits sales data to a requesting publisher.

The point-of-sale delivery systems may take one of many forms including
25 a point of purchase delivery system, a point of rental delivery system, a "book bank" subsystem, a promotional delivery system, or any combination of such systems. In the point of purchase delivery system, information is downloaded, i.e., copied, onto a user's storage media for later access by the user. The point of rental

-3-

delivery system is similar to the point of purchase system except that in addition to downloading information, an automatic erasure time period designation is downloaded. As explained in more detail hereinafter, the time period designation is utilized so that upon expiration of the designated time, the downloaded information is automatically erased from the user's storage media. The book bank subsystem is a sub-network established between authorized users, such as employees of a corporation. Each user within the sub-network with the proper authorization, or approval, can access designated information stored within the sub-network. Such a configuration enables sharing of information. In the promotional delivery system, promotional and other commercial information can be accessed for viewing and ordering of products.

A most important element of each point-of-sale delivery system is the user interface, sometimes referred to herein as the "Book Bank". The term Book Bank, as used herein, refers to the interface between the network and the user. Although the term Book Bank may imply "booktype" material, such term is not so limited. The material may be of many types such as movies, music, video, images, text, audio, and computer software material.

The Book Bank is a self-service, user interactive information vending device. Each Book Bank contains a high volume, local memory storage having, a customized portfolio of the most demanded information products for the particular site at which the Book Bank is located. Other information is transferred, via commercial communication networks (i.e., telephone networks, cable systems, satellite or cellular system or other similar communication networks), to a Book Bank for supplemental, secondary and less demanded purposes. A central processing unit contained within the Book Bank and coupled to the Book Bank local memory, storage controls downloading and dynamic encryption of the information.

In one embodiment, a user may select portions or entire contents of one or more books. The selected information may then be combined and downloaded to

-4-

the user's storage device, for example a cartridge. The cartridge, in one embodiment, includes a unique identification number and a determined amount of memory for storing the selected information.

Widespread use of the present invention should greatly simplify, and reduce the costs associated with, the publication and distribution of information. Particularly, the present architecture reduces the amount of time and resources required for the distribution of information. Further, information updates can easily be made simply by updating the master copy stored in the central information bank and then either writing over the copies stored in each Book Bank with the updated master copy or downloading the updated master copy and storing both the old and updated versions in each Book Bank. Each Book Bank contains an electronic index of the various information titles accessible from the Book Bank. In addition, the number of production quantities of a particular work can be readily controlled using the central transactional data base to track the number of copies made and sold, within the network, for each work.

The present invention also readily enables controlling reproduction of information and greatly simplifies updating of text, and the dynamic encryption of text provides copyright and proprietary information owners sufficient confidence in the present network to allow such information to be transmitted on the network.

20

Brief Description of the Drawings

These and other objects and advantages of the present invention, together with further features and advantages thereof, will become apparent from the following detailed specification when read together with the accompanying drawings, in which:

25

Fig. 1 illustrates one embodiment of the present information distribution system architecture;

Fig. 2 illustrates information flow in the system architecture shown in Fig. 1;

-5-

Fig. 3 is a block diagram illustration of a point of purchase delivery system;

Fig. 4 is a more detailed block diagram illustration of the host fileserver shown in Fig. 3;

Fig. 5 is a perspective view of a Book Bank embodiment;

5 Fig. 5A is an alternative Book Bank embodiment;

Fig. 6 is a block diagram illustration of the Book Bank circuitry;

Fig. 7 is a block diagram illustration of an end user's storage media;

Fig. 8 illustrates the information flow for the point of purchase configuration;

10 Fig. 9 is a block diagram illustration of certain elements of a point of rental delivery system;

Fig. 10 is a block diagram illustration of certain elements of a Book Bank subsystem;

15 Fig. 11 is a block diagram illustration of certain elements of a promotional delivery system; and

Fig. 12 is a flow chart illustrating the encryption process implemented in accordance with the present invention; and

Fig. 13 is another Book Bank embodiment.

Detailed Description of the Drawings

20 The following sections provide a brief overview of the present system and a detailed description of the system architecture. Following the detailed architecture description is a detailed description of point of sale delivery configurations. A detailed description of the various levels of encryption which may be used in the present system is then provided.

-6-

A. Brief Overview

In accordance with one embodiment of the present invention, information is distributed from a central information bank to a user's personalized storage medium. Information to be so distributed by the present system is received
5 from outside sources either electronically, over various communication networks (e.g., telephone lines, cable systems, cellular systems or other similar commercial communication networks) or from various storage mediums (e.g., magnetic or electronic disks, cartridges, or tape reels or compact disks, laser disks, tape cassettes, etc.), or in hard copy format. If information is received in a hard copy
10 format, it is initially converted to a standard digital format (e.g., ASCII text, DOS text or other similar standard commercially available text format) by scanning or direct transcription. Then the information is digitized, formatted, compressed and initially encrypted to form an electronic master copy which is stored in the central information bank. The master copy is duplicated electronically and dispatched
15 electronically through a communication network, such as a telephone or satellite network, to a point-of-sale delivery system. Book Banks form a part of such a delivery system, and the electronic copies are retained in the Book Banks for downloading into a user's personalized storage medium. Initially, a user selects the information to be downloaded and a tracking entry is made into a transactional
20 database to record the transfer. Prior to and during downloading of the copy on the user's storage medium, the information is dynamically encrypted utilizing a varying level of encryption which is dependent upon a variety of variables, for example, an economic value of the information. A "dynamic" encryption process is utilized so that only the electronic reader associated with the user card used to access the
25 information from the Book Bank and download the information to the user storage cartridge can be utilized to display the information in an understandable text format.

-7-

B. System Architecture

Figure 1 illustrates one embodiment of the present information distribution system. The system is shown, for illustration purposes only, as being implemented across the world. Referring to Figure 1, a central information bank 100 is the central "library," or storage location, for information. Peripheral information banks 102A-F, coupled to central information bank 100, are libraries, or storage locations, for community oriented information. For example, the information stored in central information bank 100 accessed most often from the San Francisco bay area peripheral information bank 102A may not be accessed often from the peripheral information bank for Rome, Italy 102E. In any event, central information bank 100 is coupled to each peripheral information bank 102A-F to enable sharing of information. As explained in more detail hereinafter with respect to peripheral information bank 102F, each peripheral information bank 102A-F is coupled to one or more point-of-sale sites.

A central transactional data base 104 coupled to the central information bank 100 and the peripheral information banks 102A-F, serves a central record keeping function for central information bank 100 and peripheral information banks 102A-F. Central information bank 100 and central transactional data base 104 preferably, are commercially available main frame computers, such as an IBM main frame computer. The particular main frame model selected depends on the amount of information to be centrally stored in the network, the extent of record keeping functions to be performed, and the speed at which transfer and processing of information is to occur. Importantly, the present invention is not limited to any one particular computer to serve as the central information bank and/or the central transactional data base.

As shown in Figure I is an exploded view of the various couplings between central information bank 100 and transactional data base 104, peripheral information bank 102F and various point-of-sale delivery sites, particularly, point of purchase sites 108A-C, point of rental sites 110A-D, promotional sites 112A-D,

-8-

and Book Bank subsystem sites 114A-C. Each point of purchase site 108 includes a point of purchase transactional database, represented by a box, and a user interface, represented by a circle. As explained above, the user interface is sometimes referred to herein as the "Book Bank." Specifically, point of purchase
5 site 108A contains Book Bank 116A and transactional data base 118A, site 108B contains Book Bank 116B and transactional data base 118B, and site 108C contains Book Bank 116C and transactional data base 118C. Since the central information bank 100 and peripheral information bank 102F, and specifically peripheral information bank memory storage unit 106A, also could serve as Book Banks, such
10 units are illustrated as circles. Further details regarding Book Banks and transactional data bases are provided below in Section C.

As illustrated in Figure 1, each point-of-sale delivery system, such as systems 112A, 108A-B, 110A, and 114A-B, may be networked directly to peripheral information bank 102F, or the point-of-sale delivery system, such as
15 systems 108C, 110B-D, 112B, 112D and 114B-C, may be networked to the point of purchase site 108B, which is networked to the peripheral information bank 102F. Point-of-sale delivery system configurations are explained in more detail below in Section C. At the level illustrated in Figure 1, however, it is important to understand that the delivery systems may be integrated into various combinations,
20 such as a promotional point of rental system as shown by 110B and 112B, or a promotional point of purchase system as shown by 108B and 112C or a combination of a promotional, point of purchase, and point of rental systems as shown by 108C, 110D and 112D.

Communication network links between the central information bank 100
25 central transactional data base 104, peripheral information banks 102A-F, and point of sale sites can be made utilizing one or a combination of many commercial available networks such as telephone, satellite or cable networks or any other medium suitable for transmitting information in digitized format. Many well-known protocols could be used in connection with the present system. For

-9-

example, if the Internet is used as the "backbone" network, the well-known TCP/IP protocol could be used.

Figure 2 illustrates the flow of information in accordance with the embodiment of the system architecture illustrated in Figure 1. For ease of illustration only, peripheral memory storage unit 106A is consolidated into central information bank 100, and peripheral transactional data base 106B is consolidated into central transactional data base 104. It should be understood, of course, that communication links between the peripheral information bank 102F and central information bank 100 and central transactional data base 104 are provided.

As illustrated by the inputs provided to block 202, a publisher will receive inventory reports from the central information bank 100 and sales data from central transactional data base 104. Based on this and other information the publisher can determine whether to place additional information on the network. For ease of reference such information is sometimes referred to herein as "information titles" as shown in block 204. If the information is not present in an electronic format, then the information is digitized 206, disposed in an electronic format 208 and then undergoes electronic authoring 210. The digitized information is then transmitted to a data converter 212 for converting the digitized information into a uniform format. For example, if the central information bank 104 and central transactional database 104 are DOS-based systems, the data converter will convert the information into a DOS format. If the information titles are in a digitized format, the information titles are transmitted directly to data converter 212 for direct conversion into the uniform format as illustrated by line 214.

Once the data is in a uniform, digitized format, it undergoes an initial encryption and compression to both reduce the amount of storage space required to store the data and to make the data ready for being transmitted with less risk of unauthorized use while being transmitted through a communications network. The compression is accomplished through the use of one of the commercially available

-10-

compression protocols. The initial encryption is performed using one of the standard available encryption protocols as discussed below in Section D.

Once in uniform, encrypted and digitized form, the information titles are stored in central information bank 100. An electronic index listing all titles
5 available and accessible by author, title, subject or ISBN codes is prepared. As new information titles are added, the electronic index is updated to include the new titles. The information titles may then be downloaded to Book Bank 116A. The information titles and corresponding electronic index information may, in addition to or rather than being stored in central information bank 100, be disposed on laser
10 disk masters as illustrated at block 216. Laser disk masters 216 can then be installed directly into Book Bank 116A.

Prior to downloading desired information titles, the user may access an electronic index which contains all the information titles available for downloading from Book Bank 116A. Through the electronic index, the user obtains the listing
15 for available information titles by author's name, by specific title of the work, by ISBN code or by subject matter. Once compiled, a listing of the available information titles included in the index category selected and the other necessary information to allow the user to purchase or rent any information title contained in the index category listed is displayed on the video screen. Using the video listing,
20 the user selects any title listed thereon and obtains a printout of the relevant information through the printer slot 342. Upon proper access by a user, the information titles may then be downloaded from Book Bank 116A onto a user's storage media 218.

After downloading of information and corresponding electronic index
25 information from central information bank 100 or installation of laser masters 216 to Book Bank 116A, inventory reports are generated by Book Bank 116A and transmitted to central information bank 100. These inventory reports reflect the information titles presently stored in Book Bank 116A. These reports are then sent to publisher 202. Also, a download completion report is sent from Book Bank

-11-

116A to transactional data base 118A, sometimes referred to herein as a host
fileserver, which in turn generates a status sales report. The sales report is
transmitted to central transactional data base 104. Transactional data base 104
sends the necessary action instruction back to host fileserver 118A and a transaction
5 report to publisher 202 for uses such as accounting and auditing.

For publishers who wish to allow a user to be able to produce a hard printed
copy of a portion of an information title for study purposes, when the information
titles are being downloaded into the master file, special authorization codes are
included in the data. The codes accompany the information title to the storage
10 medium, or cartridge, of the user. The codes limit the particular amount of the
information title which the user may produce in hard copy. With such coding in
place, the user may print, from the cartridge, the allowable amount of text as a hard
copy. The cartridge retains information relating to such printing and restricts
further printing once the limits have been reached. The user determines, within the
15 defined limits, or authorized purposes, the portion of the text to be produced as a
hard copy by using the high lighting features of the reader programming to make
a selection.

C. Point-of-Sale Delivery System Configurations

The point-of-sale delivery systems, as previously discussed, are classified
20 by function. The functions include one or more of the following: (1) point of
purchase delivery system, (2) point of rental delivery system, (3) book bank
subsystem, and (4) promotional delivery system. The configurations for each of
these functions are separately discussed in detail below.

1. Point of Purchase Delivery System

25 A point of purchase system is illustrated in block form in Figure 3. The
point of purchase system is described herein for illustration purposes only as a
system from which books can be purchased. As pointed out above, however, the

-12-

system is not limited to books and other media capable of being expressed in electronic form such as computer software, music and video could be purchased utilizing the present system.

The point of purchase system illustrated in Figure 3 includes a Book Bank
5 302 coupled to host fileserver 304. Server 304 is coupled to a customer service terminal 306 (of course, there could be more than one terminal) and a cashier's station 308A which is further interconnected to other cashier stations 308B-D. Server 304 also is coupled to an institution network 310 which in turn connects to institution terminals 312A-E. Service terminal 306, cashier stations 304A-D and
10 institution network 310 are connected to server 304 via a computer communication link such as a commercially available computer networking system such as CompuServe or Internet. Book Bank 302 and server 304 are connected to central information bank 100 and central transactional database 104 as hereinbefore explained with reference to Figs. 1 and 2.

15 Cashier stations 308A-D are in serial, linear networking connections which allows the addition and removal of a number of cashier stations at any time. This configuration accommodates extra cashier stations required during rush seasons or rush hours and the desire to remove cashier stations for better utilization of space after the rush seasons. Customer service terminal 306 has local processing
20 capability that provides customer services such as personal identification initiation, personal identification number changes, processing of complimentary books, book refunds, customer information entries and updates. The customer services terminal 306 can also provide the retail outlet with internal administration and the management functions, such as the book inventory cards management, the book list
25 management, book requests, book reports, financial reports, and E-Mail and Bulletin Board management.

Referring now to Figure 4, point of purchase fileserver 304 is shown in more detail. Particularly, server 304 includes a central processing unit (CPU) 316, a primary power supply 318, an uninterrupted power supply 320 to assure

-13-

continuous operation during power failure, and a high density storage 322 that holds all the programs and the data bases required for server 304 operation.

Server 304 has four (4) interfaces, i.e., a network interface 324, a maintenance interface 326, a customer service station interface 328 and a cashier station interface 330. CPU 316 transmits instructions to Book Bank 302, creates transaction data bases and reports, and processes orders from cashier stations 308A-D and customer service terminals 306A-D.

From network interface 324, server 304 communicates with central transaction data base 104 for electronic filing of transaction reports and communicates with Book Bank 302 to give Book Bank 302 downloading instruction orders and to receive the status reports and the inventory reports from Book Bank 302. Server 304 also is coupled, through network interface 324 to a Book Bank subsystem to receive subsystem reports in order to give instructions and orders whenever necessary, as hereinafter discussed. External network systems such as institutional or corporate network systems with local merchants terminals, community bulletin board services and others can also be coupled to the network interface 324. The network interface 324 also allows two-way connecting with interbank networks such as Cirrus, Plus or other similar data transfer network. Coupling to merchants' terminals, promotional system provides local merchants and the local business direct access to update their promotions and coupons. Maintenance interface 326 enables remote or on-site diagnosis and repair of server 304.

Customer service station interface 328 provides for communication between server 304 and customer service terminals 306A-D to handle customer service transactions. Customer service terminals 306A-D are illustrated as being coupled through a data switch 332 to a printer 334. Cashier station interface 330 provides that cashier stations 308A-D can communicate with server 304.

Figure 5 illustrates one embodiment of Book Bank 302. Book Bank 302 includes a high resolution color graphic display 336 which is a touch screen device

-14-

used to display, for example, instructions, messages, and status reports to the user, indexing information and to receive the user's touch screen input selections. Book Bank 302 also has a keypad 338 that is for the user to input a personal identification as well as other inputs. A magnetic code or other generally accepted card reader 5 341, shown as an insertion slot, is provided for customers' transactions with a bank card, credit card or some other form of debit card. A bar code reader 340, shown as an insertion slot, is provided to allow users to insert cards containing ISBN codes for desired information titles for reading by the Book Bank. ISBN codes may also be manually inserted by typing the relevant numbered keys on the keypad 338. A 10 printer slot 342 also is provided to enable the user to access the output of Book Bank 302, a printer (not shown in Fig. 5), as hereinafter described, to retrieve receipts and transactions reports and ISBN access vouchers. Book Bank 302 also includes a base member 344 with a cut-out portion 346 to enable a user to stand comfortably at keypad 338. Importantly, Book Bank 302 also includes a cartridge 15 slot 348 for the user to input a reading cartridge, as explained in detail hereinafter, to obtain a copy of the information selected for downloading.

In another embodiment and as shown in Figure 5A, the physical embodiment of Book Bank 302 may be altered. More specifically and in one embodiment, Book Bank 302 is positioned on a desk 339 using a personal computer 20 349, for example those available from IBM Corporation. The user operates Book Bank 302 as described above with reference to Figure 5 except, the user may for example, sit in a chair (not shown). Such a configuration may be used, for example, in a corporate, dormitory, library, or other similar environment where the user may be accessing information for longer periods of time or in a professional 25 type environment. In other embodiments, readers 340 and 341 and cartridge slot 348 each may be located within computer 349 or in separate devices which are electrically coupled to computer 349.

Figure 6 is a block diagram description of Book Bank 302 circuitry. Particularly, Book Bank 302 includes a central processing unit (CPU) 350 which

-15-

is coupled to display 336, keypad 338, magnetic strip reader 341 and bar code reader 340. Although CPU 350 is illustrated as one unit, it is contemplated that CPU 350 could be a parallel processor or distributed processor arrangement. Selection of CPU 350 type depends, of course, on the amount of information to be processed, the desired speed of processing and costs. CPU 350 also is coupled to an automatic teller machine (ATM) module 352 to allow transactions with ATM cards. CPU 350 is coupled to a media driver 354 which enables users to insert personalized media for acknowledgment or other functions as hereinafter discussed. Book Bank 302 also includes a primary local storage device 356 provided for the storage of all information masters selected for loading into Book Bank 302 and related index information. A secondary storage device 358 is provided to hold other programs, instructions and transaction related information. A buffer memory 360 is utilized to speed up downloading in order to accommodate high volume users during the peak seasons. A printer 362 is provided to print coupons on demand, receipts and various reports for the users. A power supply 364 provides power to printer 362. CPU 350, secondary storage device 358 and local storage 356. An uninterrupted power supply 366 coupled to primary power supply 364 assures continuous operation even during power down time.

CPU 350 is coupled to a network interface 368 to provide communication to central information bank 100, host fileserver 304 or a Book Bank subsystem, as hereinafter discussed. CPU 350 also is coupled to a wireless communication port 370, which in turn is coupled to an antenna 372. Wireless communication port 370 enables compatibility with an alternative communication media in the event that such media is required.

Figure 7 illustrates, in block diagram form, the structure for a user's personalized media storage cartridge 374. As explained hereinafter, a user inserts cartridge 374 into cartridge slot 348 for downloading of the information selected from Book Bank 302. The downloaded information is stored, in an encrypted format, on cartridge 374 together with relevant basic index information copied from

-16-

the electronic index contained in Book Bank 202 at the time of initial downloading. Cartridge 374 is compatible with readers to enable the user to view information stored on the cartridge. Cartridge 374 includes reading software 376 which, as explained in more detail hereinafter, performs sequential encryption and decryption of information. Registry correspondence segment 378 also is provided. A book file registry 380 is created at the time of downloading information onto cartridge 374. Encrypted bookfiles 382 together with relevant electronic index information are stored on cartridge 374 as well as a non-erasable permanently marked serial number 384. Cartridge 374 also contains a commercial operating system environment 386 and free disk space 388.

Figure 8 illustrates the user process and component processing which occurs when a user utilizes the point-of-purchase system described above. Particularly, once the user enters the site 390, and if the user a first-time patron 392, the user will complete a user's application form 394. The user will then take the completed application and a picture I.D. to customer service station 306A, where the user will select and input a personal identification number (PIN) and a password 396. The customer service clerk will open an account for the customer 398. The user-selected password is automatically matched with a sequentially created customer account number within the central data banks. Using the keypad accompanying the cashier station, the clerk types in the name, address and social security number for the user. The written application will then be inserted into the printer slot accompanying the cashier station. While loading the customers' information, the central data bank reviews the information to determine if there are any prior problems with the customer or other discrepancies. When the verification process is completed, the designated customer account number for the user is printed on the users application. Then, the clerk places, into the card slot, a user identification card (for example, a plastic card having dimensions of the standard credit card and containing a magnetic strip on the back on which can be placed magnetic coded information). The card is then embossed with the user's name and account number

-17-

and the magnetic strip is encoded with the applicable user codes (account number, card number and designated password information). The written application is then transmitted to the central data storage center for retention. The user now is able to use the issued card to make purchases or to rent the use of information titles. The machine used to emboss and encode the cards is a standard commercially available machine of the type currently being used in connection with the issuance of a bank's card, credit cards or debit cards.

In addition to obtaining a personal identification card, a new enrollee purchases a reader/computer or other acceptable reading device (such as a special computerized interface, or an audio or video playback device). Each such device is assigned a unique serial number and a special code number. In one embodiment, the serial number is contained on a read only memory chip enclosed within the device. All of the many cartridges which accompany each such reading device is encoded in such a manner that information recorded on the cartridge can only be read by the related reading device. This is accomplished by a simple program contained within the permanent memory of the device. In one embodiment, if the special code on the device is not the same as the number which the cartridge is seeking, then the cartridge will cause to be displayed in the reading device the words "cartridge cannot be read by this device" and to not allow any further access to any information contained on the cartridge by the particular reading device in question. If the numbers match, further access will be allowed. At the time a reading device is purchased, the clerk enters the serial number for the device into the central data bank through the cashiers station. The central data bank contains a list of the serial numbers of all approved reading devices and the corresponding special code number. The personal identification card for the customer purchasing the reading device is placed by the clerk into an appropriate slot on the cashier's station and the magnetic strip on the identification card is encoded with the applicable serial number for the reading device being purchased. Thereafter, whenever the user desires to obtain additional cartridges for reading by his or her

-18-

reading device, the user needs only present his personal identification card and the cartridge to be properly coded to the clerk and by inserting the cartridge into the cartridge slot and the identification card into the card slot and pressing the designated button on the cashiers station, the new cartridge will be correctly
5 encoded to be readable by the user's reading device.

The requirement that reading device codes and cartridge codes match, before access will be allowed, means that issued cartridges will not be readily readable by multiple reading devices. Multiple device reading will require special programming and the granting of special allowances, or approved purposes. In one
10 embodiment, by purpose encoding the information, a publisher may expand or limit access to those users having a proper authorization or defined purpose. For example, a publisher may purpose encode a portion of a selected book to be readable by any user, i.e., any purpose encoded. This type of purpose encoding may be used on, for example, promotional books or some governmental publications.
15 Other information may be purpose encoded to limit access to a single user to only view the information, i.e., classified material. The requirement reduces the possibility of unauthorized use of information titles.

In operation, the customer takes the customer identification card to the book display area for shopping 400. If the customer previously opened an account, the
20 customer will not have to go through the above described process and can proceed directly to shopping area 400, where the customer will select a book inventory card matching his book selection. The book inventory card has the book ISB numbers, a bar code and information related to the particular information title, author, publisher, and edition date printed thereon. The customer brings the selected book
25 inventory card to the cashier's station 308A. The cashier magnetically reads the codes on the customer's I.D. card and scans or manually enters the bar codes on book inventory cards 402. The customer then makes a proper payment, and the customer codes and information title bar codes are transmitted to host files server 304. Server 304 searches the existing customer account file to match the

-19-

identification (i.e., pin and password) and will generate a downloaded book list file based on the bar codes from the book inventory cards or as manually loaded. Server 304 downloads the file to Book Bank 302 which electronically generates a portfolio of information titles ready to be downloaded on demand. The user can then proceed to Book Bank 302 at any later time and insert the identification card into the slot 340 of Book Bank 302 and a coded point-of-purchase cartridge 374 into Book Bank cartridge slot 348 to identify himself with a personal identification number, as illustrated in step 404. The user also enters a password 406 into the key pad 338. When Book Bank CPU 350 matches the personal identification number with a downloaded list portfolio, Book Bank CPU 350 starts downloading the requested information from local storage 356, through buffer memory 360, to media driver 354 which copies the information onto cartridge 374. As part of the downloading process, the data is dynamically encrypted to make the data uniquely readable only by authorized reading devices. The dynamic encrypting is described below in Section D. After downloading, the user removes cartridge 374 and then inserts cartridge 374 into his personal reader/computer to access the information acquired. The reader/computers are configured for long-term reading applications. The reading application software is stored on cartridges with the ability to read the applicable software on the cartridges permanently stored within the memory of the reader/computers or other authorized reading device.

A user may select portions of selected information to combine and download. In one embodiment, the user may select at least one book and select at least one portion of each selected book. If more than one portion is selected, where each portion includes up to the entire selected book, the portions are combined and downloaded to user's cartridge 374. For example, for a specific college course, a student may be required to download specific chapters from ten different books. After selecting the ten books and ten specific chapters of the selected books, the selected information is combined, encrypted using the determined level of protection, and downloaded to the students cartridge 374. Similarly, a user may

-20-

select individual tracks from music information to combine and download the selected tracks to a single cartridge 374 for playback at a later time.

2. Point of Rental Delivery System

If a user is not interested in obtaining a permanent copy of a particular work but requires a copy for a period of time, e.g., a semester, the user may prefer to visit a point of rental site rather than a point of purchase site. A point of rental system is illustrated in Figure 9. The rental system is identical to the point of purchase system previously described herein (e.g., includes a host fileserver) except with respect to the differences pointed out below. In many instances, a single site or Book Bank may serve as a point of purchase system site and a point of rental system site and a point of delivery system for promotional or commercial information site or any combination thereof. As shown in Figure 9, the point of rental system includes Book Bank user terminal hubs 410A-B coupled to terminals 412A-E, and customer service station 414. User terminals 412A-E allow a customer to do an information title search and index search of the Book Bank memory and to transmit other information between Book Bank 302 and himself. Customer service station 414 combines the function of customer service as well as the cashier's station. For example, at customer service station 414, a credit customers' debit card can be credited and the ATM operation can be overridden, via ATM module 416, if necessary. Information can be printed out from customer service station 414 via printer 418.

Point of rental storage media 420 is used in the rental system. Point of rental media 420 is the same as point of purchase media 374 except that media 420 includes an automatic erasure mechanism that erases the information downloaded after the expiration of a preset time interval. More specifically, when information is downloaded from Book Bank 302 onto media 420, Book Bank 302 also downloads a "time stamp" equal to the time period for which the user has paid to retain a copy of the information. The time stamp could take the form of a value

-21-

loaded into a memory location on media 420, which value corresponds to the rental time period. During usage, the actual usage time elapsed is subtracted from the electronically stamped time period. Once the user has consumed all of the usage hours authorized, the information title will self-destruct, i.e., be deleted from media

5 420. This can be achieved simply by calling a stored program which erases the information associated with the memory location where the time value is stored. For example, when the value of the memory location where the authorized usage time is stored is zero, the stored erase program would be called upon to erase the information associated with the "zero" time usage authorization.

10 Another method for automatic erasure is for each rental or library cartridge to contain a real time clock and independent rechargeable power supply. When the cartridge is initially encoded for use, the real time clock mechanism is activated. As rented information titles are being downloaded, an expiration date is logged into the index information for each title. Any time after the real time clock on the

15 cartridge reaches the designated expiration date, access to the relevant information title is denied. If use of the title is not extended, after the expiration of an additional number of days, use of the cartridge will cause a permanent erasure of the information title from the cartridge memory. With this method, if the real time clock falls to operate, the cartridge will become unreadable without repair. To

20 repair a defective cartridge, the user need only bring the cartridge together with his personal identification card to the nearest service center where the real time clock will either be repaired or the relevant information titles will be loaded onto a replacement cartridge. The user will be credited for any lost time while the cartridge was unreadable. A service center could be located, for example, near each

25 separate point of delivery site.

The automatic erasure program could be created as an operating system module or as a separate executable program designed to be "terminate and stay resident" (TAR). A module integral with the operating system is preferred since

-22-

such a structure ensures that if the operating system is viable, the automatic erasure module is viable.

If the user still needs more time with any particular information titles, the user may return to the point of rental site and "re-rent" the information.

5 Alternatively, it is contemplated that the user could renew the rental via a modem coupled to the reader.

With respect to the user process for renting information, when a point of rental patron enters a point of rental site, the user will use a valid ATM card, bank card, credit card, or some other debit card and proceed directly to a user terminal
10 412A-E. Using such a terminal, the user can perform information title searching and download an order entry to Book Bank 302. When the download entry is complete, the user will go to Book Bank 302, insert a rental media, an identification card, and a credit card, bank card, or debit card into Book Bank 302 for the transaction approval. If the transaction is not cleared or if the ATM system is not
15 working properly, the patron can proceed to the customer service center and have the attendant manually override the ATM process, if appropriate. If the user does not have a valid ATM credit card or debit card, the user will go to the customer service center, pay the service clerk to receive credit on the Book Bank debit card. Then the customer may proceed to the user terminal where the user downloads the
20 order entry.

After the transaction approval is cleared, the patron inserts point of rental media 420 into Book Bank media driver 354, has his personal identification card scanned and enters a password. The information is dynamically encrypted and downloaded from Book Bank 302 to media 420 with an electronic stamp of the
25 number of hours of usage authorized for each information title or an expiration date. After the downloading, the user will apply this media on the personal reader/computer to access the information on the media.

Typical examples for the point of rental site are libraries (commercial, education or public access) and book rental shops. The information downloaded

-23-

by the user may be free of charge to the users such as in the case of a library, or may incur certain rental fees at a predetermined rate, such as in the case of a rental shop or library charging on a per page use basis. Any given point of rental site may operate as a traditional library in allowing free use to library members for a limited
5 period of time or may operate as a rental shop where fees are collected from users in accordance with the period of use allowed.

3. Book Bank Subsystem

A Book Bank subsystem couples to a Book Bank and host fileserver as described in more detail below. The central element of the subsystem is a Book
10 Bank which is a modified version of the point-of-purchase Book Bank 302. The subsystem is specifically configured for the collective use by members or the staff of a commercial or business entity or a corporation. It delivers and it recalls information titles among authorized users within the business or corporate entity, and provides the capability of limiting the number of copies of a given work that
15 may be distributed to other authorized users. If all of the licensed copies of any information titles have been checked out by the staff of an organization, then no other users may access the same information title within that particular subsystem until one or more of the licensed copies of the particular information is uploaded or recalled to the subsystem or additional copies are purchased. In one
20 embodiment, the information is purpose encoded so as to limit a purpose to which access by a user to the information is allowed. For example, a central corporate library may allow specific users, i.e., R&D personnel, to selected information, i.e., pending patent applications. All non-R&D personnel users without the proper purpose, or authorization code, are prevented from accessing the information.

25 Instead of purchasing the unlimited use of a limited number of copies, a commercial or business entity may lease the limited use of an unlimited number of copies or the use of a specified portion of a given information title. Under such circumstances, the commercial or business entity would be charged each time the

-24-

subsystem is accessed from a participating work station for the portion of a specified information titled accessed and for the period of time the access occurs. By restrictions encoded on the interface between a participating work station and the subsystem, while accessing information from the subsystem, the ability of the work station to perform certain operations would be restricted. The restricted operations would be those related to the duplication or transmission of data related to information titles being accessed through the subsystem.

More specifically, and referring to Figure 10, Book Bank subsystem 422 contains a high resolution color graphic display 424 coupled to CPU 426 to display the instructions or status of subsystem 422. Subsystem 422 also includes a keyboard 428 with limited access to the system for keying selections for operating certain given functions such as product display. Subsystem 422 has a media driver 430 for the downloading of information and a local storage 432 which holds a portfolio of information the business entity has ordered for use. A secondary storage 434 is also provided to hold all the software programs that control and perform the functions of subsystem 422. Subsystem 422 further includes a power supply 436 and an uninterrupted power supply 438 to assure continuous operation during power failure downtime. A printer 440 is provided to print various reports.

Book Bank subsystem 422 has a network interface 442 that connects subsystem 422 to Book Bank 302 and host fileserver 304. Network interface 442 also may couple to the corporate or business entity network system 444. With such a structure, the corporate entity may transmit or download its own corporate proprietary information through Book Bank subsystem 422.

Media port extension interface 446 provides access by an adequate number of media drivers to the desired corporate terminals for corporate network stations. Media driver 430 is connected to the terminals or stations by a proprietor driver card. The corporate administration can utilize the dynamic encryption and the dynamic downloading function of Book Bank subsystem 422 to incorporate and accommodate the corporate proprietary information. The corporate proprietary

-25-

information may be transmitted to Book Bank subsystem 422 using an encryption process and then downloaded selectively to the destination port and to the properly identified authorized personnel. Book Bank subsystem 422 is not only a customized corporate library of copyrighted and proprietary information, but also
5 is a corporate document security device that encrypts and dispatches the corporate documents and the corporate confidential proprietary information in the corporate network system. As part of the network interface connection linking each participating work station to the subsystem and allowing access to encrypted information, a separate unit, e.g., a memory storage unit restricts certain operations
10 which may be performed from the work station so long as the work station has access to encrypted information from the subsystem. The restrictions limit or prevent operations related to the duplication or transmission of data.

4. Promotional Delivery System

The promotional system is a point of delivery system for promotional and
15 commercial information. It distributes promotional and commercial information in electronic format and users may either view the digitized promotional and commercial information at the site or download the information to their personalized media for later viewing. User's call access the promotional and commercial information including the dynamic viewing electronically of
20 advertising available discounts, commercials, special promotional events, software demos and product catalogs. Users may even shop electronically by manipulating the promotional and commercial information and placing orders through E-Mail from a personal reader/computer or by ordering directly from an interactive promotional Book Bank. The promotional Book Bank has the same structure as
25 Book Bank 302 for the point-of-purchase system.

A promotional system in accordance with the present invention is illustrated in block diagram form in Figure 11. As in the other point-of-sale systems, Book Bank 302 is networked to host fileserver 304. The promotional system further

-26-

includes a number of promotional units 448A-D which electronically display and promote products. Unit 448A is coupled directly to central transactional data base 104 and central information bank 100 while units 448B-D are coupled to host fileserver 304. Unit 448A receives information from merchant terminals 450-A-D
5 and host fileserver 304, receives information via merchant terminals 450E-G. More specifically, host fileserver 304 receives advertising and special offer updates from the local businesses, national or regional advertisers, and corporate sponsors through merchants terminals (MT) 450E-G. The host fileserver 304 is also networked to a central transaction data base which, in turn, provides a report to the
10 publishers, advertisers, accounting, auditing firms, merchandise vendors, and others.

The promotional Book Bank allows selective downloading of promotional and commercial information to the user's point of rental media (see discussion in Section B, System Architecture, for explanation of such downloading) for the user's
15 private review and personal shopping at his convenience. The promotional and the commercial information downloaded will self-destruct (i.e., automatically erase) at the expiration of a pre-determined time interval as explained above with respect to point of rental delivery systems. The promotional Book Bank also provides a user interactive self-service vending feature. The user may order products or
20 information electronically via the network. Some of the promotional functions are: coupons on demand, virtual shopping, catalog sales, demos, subscription orders, electronic applications of credit cards, calling cards, or other types of services. Some public domain information distributed such as community events, ticket sales, institutional events or even public bulletins could also be distributed with the
25 promotional information as a free or low cost service to the community.

The promotional and the commercial information flow is very similar to the information flow within the point-of-sale delivery system. However, rather than a publisher or copyright information owners, the information sources are local

-27-

businesses, national or regional advertisers, and appropriate sponsors through advertising agents and other entities.

5. Information Tracking

In one embodiment, for each exchange, or download, of information, a tracking entry is transmitted, or stored, in an appropriate transactional database, for example central transactional database 104, to record movement, or transfer, of information from a first location to a second location. By reviewing these tracking entries, an information owner may monitor movement of the information and take appropriate action. For example, a tracking entry may be recorded in transactional database 104 each time any information is copied into, removed from, or copied from, Book Bank 302. Based upon the tracking entries, the information owner may charge the receiving or transferring party a fee as defined by the owner of the information. The fee charged may be based on a variety of factors including, but not limited to, an economic value of the information, use or purpose of the information, number of users, and the availability from other sources. The tracking entries may also include additional data so that the information owner may determine who transferred the information, the amount of information transferred, type of transfer, i.e., rental for specified period of time, and the time of transfer. Tracking entries, in one embodiment, are recorded for all transfers, i.e., Book Bank 302 to cartridge 374 and central information bank 100 to Book Bank 302. Utilizing these entries, an information owner may also determine the type of information that is being transferred, the number of transfers, and the identification of the information receivers. For example, the information may be used to determine if certain promotional materials are being received by targeted users, or to determine responses to different information pricing strategies.

-28-

D. Encryption

The above-described point-of-sale delivery systems have the capability of performing dynamic encryption of data as the data is downloaded onto a user's storage media. Dynamic encryption refers to the process in which the Book Bank works together with the storage media to perform a proprietary encryption of downloaded data. In one embodiment, different levels of encryption are utilized based on a series of factors or variables. These variables include, but are not limited to, an economic life, a market value, a general availability, a replacement cost, time sensitivity, and potential number of users, of the information. For example, today's TV listings may have a low level, or complexity, of encryption as a result of the low market value, low replacement cost, and general availability from many sources. Conversely, a multi-volume legal treatise may, for example, have a high, or complex, level of encryption as a result of the limited availability, replacement cost, and long economic life of the information. Based on the described factors, a source, i.e., publisher, of information may decide the appropriate level of encryption for each portion of information from the initial transmission to central information bank 100 to a user's cartridge 374. The level of information encryption, in any location, i.e., bank 100, may be higher or lower than another location, i.e., cartridge 374. More specifically, each time information is downloaded, or transmitted, the level of encryption may be independently altered, or determined. For example, the level of encryption at central information bank 100 may be different, i.e., higher or lower, than the level of encryption of a book downloaded to a user's cartridge 374.

In addition to dynamic encryption, other encryption may be performed as illustrated in Figure 12. Figure 12 illustrates a three level encryption process. For example, prior to transmitting information on the network, the data may be encrypted. This facilitates preventing unauthorized users from accessing the transmitted information on the network. In addition to the pre-transport encryption, the data, may be encrypted prior to being placed in a book bank. Publishers or other

-29-

owners of the information may have approval authority over this level of encryption to provide such information owners with satisfaction that the data is adequately protected.

Once the data is stored in the book bank, dynamic encryption techniques
5 may be used when downloading the data onto storage media. The storage media (Figure 7) includes a proprietary environment for building, reading, viewing and processing. The media also has a commercial operating system environment for processing information files. An information file directory registry forms a part of the proprietary application, and a file directory pointer is contained in the operating
10 system application.

The dynamic encryption process, in one form, uses the permanent serial numbers stored in the storage media, the user's personal identification number, a personal signature code number, and a password to further encrypt the data stored in the book bank as the data is downloaded to a user's storage media. The
15 personalized variables and codings are combined with various individualized information file variables to form an individualized data structure for the data downloaded to the user's personalized media. As a result, those information files are individualized pertaining to the media, the version of software, the information file itself, and other variables.

20 The dynamic encryption assists in reducing the possibility of the unauthorized use of proprietary or other information by causing all information downloaded through the point-of-sale delivery system to be readable and accessible by a selected number of user readers/computers. Specifically, data storage medium accessible from one reader/computer will not be accessible using another reader/
25 computer unless such access has been prearranged such as by providing the other reader/computer with an identical user identification number and password.

Examples of well-known encryption algorithms which may used in performing the above described three level encryption include the Z8068 Data Ciphering Processor (DCP). The DCP contains the structure to encrypt and decrypt

-30-

data using National Bureau of Standards encryption algorithms. It may be used in a variety of environments including in dedicated controllers, communication concentrators, terminals and peripheral task processors in general processor systems. DCP provides a high throughput rate using cipher feedback, electronic code book or cipher block chain operating modes. The provisions of separate ports for key input, clear data and enciphered data enhances security. The host system communicates with the DCP using commands entered in the master port or through auxiliary control lines. Once set up, data can flow through the DCP at high speeds because input, output and ciphering activities can be performed concurrently.

In alternative embodiments, encryption and decryption may be performed in dedicated hardware and/or software functions. For example, each reader and cartridge 374 may include a dedicated encryption integrated circuit (IC) and a dedicated decryption IC to maximize the transfer speed of the information. The level of encryption and decryption may altered by adding additional functions and by enabling or disabling the additional levels.

With respect to dynamic encryption, the following describes one of many methods of dynamic encryption which could be used. Particularly, each regularly used alpha or numeric symbol is assigned a corresponding number as illustrated in Table 1.

Table 1

symbol	A	B	C	D	E	F	G	H	I	J	K
code	1	2	3	4	5	6	7	8	9	10	11
symbol	L	M	N	O	P	Q	R	S	T	U	V
code	12	13	14	15	16	17	18	19	20	21	22
symbol	W	X	Y	Z	0	1	2	3	4	5	6
code	23	24	25	26	27	28	29	30	31	32	33

-31-

symbol	7	8	9	.	,	;	:	+	-	x	
code	34	35	36	37	38	39	40	41	42	43	44

The serial number stored on the cartridge would be used to determine how many slots the code should shift to the left at the start the encrypting. For example, if the serial number ended with six, before starting of encryption, the code would be shifted to the left by six places. Table 2 illustrates the code table after the shift.

Table 2

	symbol	A	B	C	D	E	F	G	H	I	J	K
	code	7	8	9	10	11	12	13	14	15	16	17
10	symbol	L	M	N	O	P	Q	R	S	T	U	V
	code	18	19	20	21	22	23	24	25	26	27	28
	symbol	W	X	Y	Z	0	1	2	3	4	5	6
	code	29	30	31	32	33	34	35	36	37	38	39
15	symbol	7	8	9	.	,	;	:	+	-	x	
	code	40	41	42	43	44	1	2	3	4	45	6

The selected user password then is used to determine after how many symbols the code should again shift to the left. As an example, if the password were ROSE, then using the codes from Table 2, the numeric statement for rose would be 24212511. When the corresponding numbers are added together until reaching, a number between 1 and 10, the number reached in our example is 9 [18.9]. So after every 9th letter, the codes would be shifted another 6 spaces to the

-32-

left. After the encrypting of 9 letters, the codes would be set as set forth in Table 3.

Table 3

5	symbol	A	B	C	D	E	F	G	H	I	J	K
	code	13	14	15	16	17	18	19	20	21	22	23
	symbol	L	M	N	O	P	Q	R	S	T	U	V
	code	24	25	26	27	28	29	30	31	32	33	34
	symbol	W	X	Y	Z	0	1	2	3	4	5	6
	code	35	36	37	38	39	40	41	42	43	44	1
10	symbol	7	8	9	.	,	;	:	+	-	x	
	code	2	3	4	5	6	7	8	9	10	11	12

Because the fact that the encrypting tables are constantly shifting, under this simple method, the phrase "My brown dog has fleas." would be encrypted as follows:

15	19	31	6	8	24	21	29	20	6
	16	27	19	12	20	13	31	18	24
	23	19	37	11					

Decoding using only Table 1, the coded phrase would read as follows:

	S	4	F	H	X	U	2	T	F
20	P	0	S	L	T	M	4	R	X
	W	S	.	K					

Without knowing other information, it would be very difficult to find a pattern that would allow one to decode the symbols.

Knowing the placement of the codes relative to the symbols at the start of the encryption process and the number of symbols between shifts, decoding an

-33-

encrypted phrase is simply a reversal of the process applying each of the tables in reverse.

There are any number of similar methods of dynamic encryption which use a different manner of determining how encryption codes will vary as one proceeds
5 through the data to be encrypted. The objective, of course, is to make decoding difficult by avoiding obvious patterns associated with conventional language and number usage.

E. Tamper Protection

In one embodiment of the present invention, access to the information is
10 monitored, or recorded, to determine attempted unauthorized access to the information. If an unauthorized access is recorded, or stored, onto a user's media, for example cartridge 374, the next time that user attempts to download additional information to cartridge 374, an unauthorized access message may be transmitted to notify the appropriate party, for example the cashier. As a result of the
15 unauthorized access message, the cashier may revoke user's cartridge 374, notify the proper authorities, or record an entry into the user's account for future action. More specifically and in one embodiment, the unauthorized access is determined by first reading, or recording, the specific identification data from the information requester, or receiver. If the data provided by the information receiver is
20 determined to not match, i.e., is unequal, predefined values, the unauthorized access message is recorded and information exchange is prevented. The data determination may be completed using known comparison hardware and/or software functions.

Additionally, the unauthorized access message may be generated if a user
25 having an incorrect purpose, or authorization code, attempts to access unauthorized information. For example, in a corporate environment, if a user attempted to access information for which the user did not have the proper authorization code, an unauthorized access message is generated and may be sent to, for example a system

-34-

administrator or a security official. Different level of unauthorized access messages may also be generated. For example, a high level message may be generated if a user attempts to decrypt the information stored in various locations within the system, for example Book Bank 302 using an unauthorized device. A lower level
5 message may be generated if a remote user has attempted to access data that is one level above that user's authorized level.

F. Other Embodiments

In another embodiment, Book Bank 302 may be configured to capture and exchange real-time information. For example, as a professor presents material to
10 students in a classroom, the professor's presentation may be captured and converted into copyrighted text and exchanged with remote users. This conversion may be completed using known voice to text conversion systems using a known computer system. The professor's presentation may be supplemented with previously prepared, or concurrently prepared, written text. The text may be digitized and
15 properly integrated into the text using known methods. Remote users may receive information from the professor's lecture in real-time as the material is presented or may receive the information at a later time. Remote users receive only that information which the remote user is authorized to receive from Book Bank 302 as described above.

20 In yet another embodiment Book Bank 302 is configured to receive audio, video, and/or computer software code. For example and in one embodiment shown in Figure 13, Book Bank 302 is coupled to a Video Cassette Recorder (VCR) 600, a stereo system 610 including a cassette recorder/player 620, a Compact Disk (CD) player and/or recorder 630, a television 640, and a computer 650. As described
25 above, authorized information is received from Book Bank 302, and in one embodiment is stored to a storage device, for example a memory device 660. The memory device 660 may be a plurality of memory cells, for example Read Access Ram (RAM), Read Only Memory (ROM), a rotating storage unit, i.e., a hard disk,

-35-

a magnetic storage media, i.e., magnetic tape, or other storage media, for example an optical storage media. After the remote user has selected the appropriate information to receive from Book Bank 302, the information is stored in device 660. Device 660 is configured to transfer the stored information to the selected
5 playback device, i.e., Video Cassette Recorder (VCR) 600, stereo system 610, cassette recorder/player 620, CD player/recorder 630, television 640, or computer 650. For example, in one embodiment, the remote user downloads, or receives, the entire contents of a top ten music album. The contents of the album is stored in device 660. As described above, the information may be permanently stored or may
10 be stored for a fixed period of time or number of uses. After downloading the information to device 660, the remote user may transfer the information to stereo system 610 for listening. In another embodiment, the information may be transferred to, or through, device 660 to one of the other components, i.e., cassette recorder 620, VCR 610, CD recorder 630, or computer 650. To limit unauthorized
15 copying or playback, the information may be playback using only those components, i.e., cassette recorder 620, VCR 610, CD recorder 630, or computer 650, coupled to device 660. For example, the remote user may download a feature movie by saving the movie on a tape using VCR 610. The remote user may then playback the movie as authorized as long as the tape is playback in VCR 610 that
20 is coupled to device 660. Similarly, the remote user may download a software program so that the information is stored in device 660 or in a storage media in computer 650. Depending upon the authorization code of the software, the program may be configured to execute only from computer 650 when computer 650 is coupled to device 660.

25 The above described system facilitates controlled and monitored exchange of information between many types of information owners, distributors, and users. By using the described system, a user may obtain many types of authorized information. The user may, as determined by the information owner, purchase, rent, or obtain without charge, the authorized information. The information, in one

-36-

embodiment, is encrypted using various levels, or complexities, of encryption to prevent unauthorized access. The level of encryption depends upon a variety of factors or variables, for example, the economic life of the information. For example, Book Bank 302 may include information representing a reference dictionary and a top ten music album. Information from the reference dictionary and the album may have the same or different levels of encryption. In addition, students from a determined class may access the reference dictionary information without charge as the result of the school purchasing an unlimited use copy of the information, however, those same students would be required to purchase any information downloaded from the album. Additionally, the type of access may differ for different portions of the information. For example, a first track of the album information may be coded so that anyone may download the information without charge, however, the remaining tracks of the album information may be coded to require payment to download.

15 While the present invention has been described with respect to specific embodiments, many modifications, variations, substitutions, and equivalents will be apparent to those skilled in the art. Accordingly, the invention is to be considered as limited only by the spirit and scope of the appended claims.

-37-

CLAIMS:

1. Apparatus for facilitating obtaining text of a book, comprising:
a storage device having stored therein text of a plurality of books;
a processor connected to said storage device, said storage device further having
5 stored therein a program for controlling said processor, said processor operative
with the program to:
receive a book selection request;
receive a user identification associated with the book selection request; and
output encrypted text of the selected book if the user identification and book
10 selection are valid utilizing a determined level of book encryption.
2. Apparatus in accordance with Claim 1 wherein said processor is further
operative with the program to purpose encode the text of the selected book so as to
limit a purpose which access by the user to the text is authorized.
3. Apparatus in accordance with Claim 1 wherein said processor is further
15 operative with the program to encrypt the text of the selected book utilizing an
identifier associated with the user.
4. Apparatus in accordance with Claim 1 wherein said determined level
of book encryption is based on at least an economic value of the selected book.

-38-

5. Apparatus in accordance with Claim 1 further comprising at least one local unit communicatively coupled to said processor, said local unit comprising a memory for storing, in electronic form, information transmitted to said unit from said processor, and a local unit processor for controlling transfer of information
- 5 stored in said unit to electronic storage media of system users, said local unit configured to encrypt the information when the information is to be transferred to the electronic storage media, said local unit configured to encrypt the information utilizing a determined level of information encryption.
6. Apparatus in accordance with Claim 5 wherein the information stored
- 10 on the user's storage media comprises a personal signature code number and serial number.
7. Apparatus in accordance with Claim 5 wherein said determined level of information encryption is based upon at least an economic value of the selected book.
8. Apparatus in accordance with Claim 7 wherein the level of information
- 15 encryption is not equal to the level of the book encryption.
9. Apparatus in accordance with Claim 1 wherein said processor is further operative with the program to determine attempted unauthorized access to the output copies of the text of a book.

-39-

10. A method for operating a computer to obtain text of a book, comprising:

inputting into the computer a book selection request;

inputting into the computer a user identification associated with the book
5 selection request; and

outputting encrypted text of the selected book if the user identification and book selection are valid utilizing a determined level of book encryption.

11. A method in accordance with Claim 10 wherein inputting the computer a book selection comprises the steps of:

10 selecting at least one of the books in the memory storage; and
selecting the text of at least a portion of each the selected books.

12. A method in accordance with Claim 11 wherein inputting the computer a book selection further comprises the steps of:

determining if more than one portion is selected; and

15 if more than one portion is selected, then combining the selected portions.

13. A method in accordance with Claim 11 further comprising the step of generating tracking information corresponding to the selected portions and the selected books.

14. A method in accordance with Claim 10 further comprising the step of
20 purpose encoding the outputted text of the selected book so as to limit a purpose which access by the user to the text is authorized.

15. A method in accordance with Claim 10 further comprising the step of determining a level of book encryption.

-40-

16. A method in accordance with Claim 15 wherein determining a level of book encryption comprises the step of selecting at least one of a plurality of levels of encryption code.

17. A method for operating a processor communicatively coupled to a network to obtain text of a book, the network being coupled to a memory storage having stored therein text of a plurality of books, said method comprising:
determining a book selection request;
inputting into the processor the book selection request;
inputting into the processor a user identification associated with the book selection request; and
outputting encrypted text of the selected book if the user identification and book selection request are valid utilizing a determined level of encryption.

18. A method in accordance with Claim 17 wherein determining the book selection request comprises the steps of:
reviewing the books in the memory storage;
selecting the text of at least a portion of at least one of the books in the memory storage.

19. A method in accordance with Claim 17 wherein determining the book selection request comprises the steps of:
selecting at least one books in the memory storage;
selecting the text of at least a portion of each selected book in the memory storage;
combining the selected portions of the selected books.

-41-

20. A method in accordance with Claim 19 wherein selecting the text of at least a portion of each selected book in the memory storage comprises the step of selecting the entire book.

21. A method in accordance with Claim 19 wherein selecting the text of at least a portion of each selected book in the memory storage comprises the step of selecting at least one word of each selected book.

22. A method in accordance with Claim 19 wherein selecting the text of at least a portion of each selected book in the memory storage comprises the step of selecting at least one section of each selected book.

23. A method in accordance with Claim 19 further comprising the step of generating tracking information corresponding to the selected books in the memory storage.

24. A method in accordance with Claim 23 further comprising the step of generating tracking information corresponding to the selected portions of the selected books.

25. A method in accordance with Claim 17 further comprising the step of purpose encoding the outputted text of the selected book so as to define a purpose of use which access by the user to the text is authorized.

26. A method in accordance with Claim 17 further comprising the step of determining unauthorized access to the encrypted text.

-42-

27. Apparatus for facilitating obtaining text of a book, comprising:
a storage device having stored therein text of a plurality of books;
a processor connected to said storage device, said storage device further having
stored therein a program for controlling said processor, said processor operative
5 with the program to:
receive a book selection request;
receive a user identification associated with the book selection request; and
dynamically encrypting text of the selected book utilizing a determined level
of encryption if the book selection request and user identification are valid, the
10 dynamic encryption performed using a user identifier as the text is outputted.
28. Apparatus in accordance with Claim 27 wherein said apparatus is
communicatively coupled to a communications network.
29. Apparatus in accordance with Claim 27 wherein said processor is
further operative with the program to purpose encode the text of the selected book
15 so as to limit a purpose which access by the user to the text is authorized.
30. Apparatus in accordance with Claim 27 further comprising at least one
local unit communicatively coupled to said processor, said local unit comprising a
memory for storing, in electronic form, information transmitted to said unit from
said processor, and a local unit processor for controlling transfer of information
20 stored in said unit to electronic storage media of system users, said local unit
configured to encrypt the information when the information is to be transferred to
the electronic storage media, said local unit configured to encrypt the information
utilizing a determined level of encryption.

Supp. r
in
94
filing
box
not
filed
in time

-43-

31. A method for operating a computer to obtain information, comprising:
inputting into the computer an information selection request;
inputting into the computer a user identification associated with the
information selection request; and
5 outputting encrypted information of the selected information if the user
identification and information selection are valid utilizing a determined level of
information encryption.
32. A method in accordance with Claim 31 wherein inputting the computer
an information selection comprises the step of selecting at least one portion of
10 information in the memory storage.
33. A method in accordance with Claim 32 wherein inputting the computer
a book selection further comprises the steps of:
determining if more than one portion is selected; and
if more than one portion is selected, then combining the selected portions.
- 15 34. A method in accordance with Claim 32 further comprising the step of
generating tracking information corresponding to the selected portions of
information.
35. A method in accordance with Claim 31 further comprising the step of
purpose encoding the encrypted information so as to limit a purpose which access
20 by the user to the information is authorized.
36. A method in accordance with Claim 31 wherein the information
includes at least one of visual, music, software, and video information.

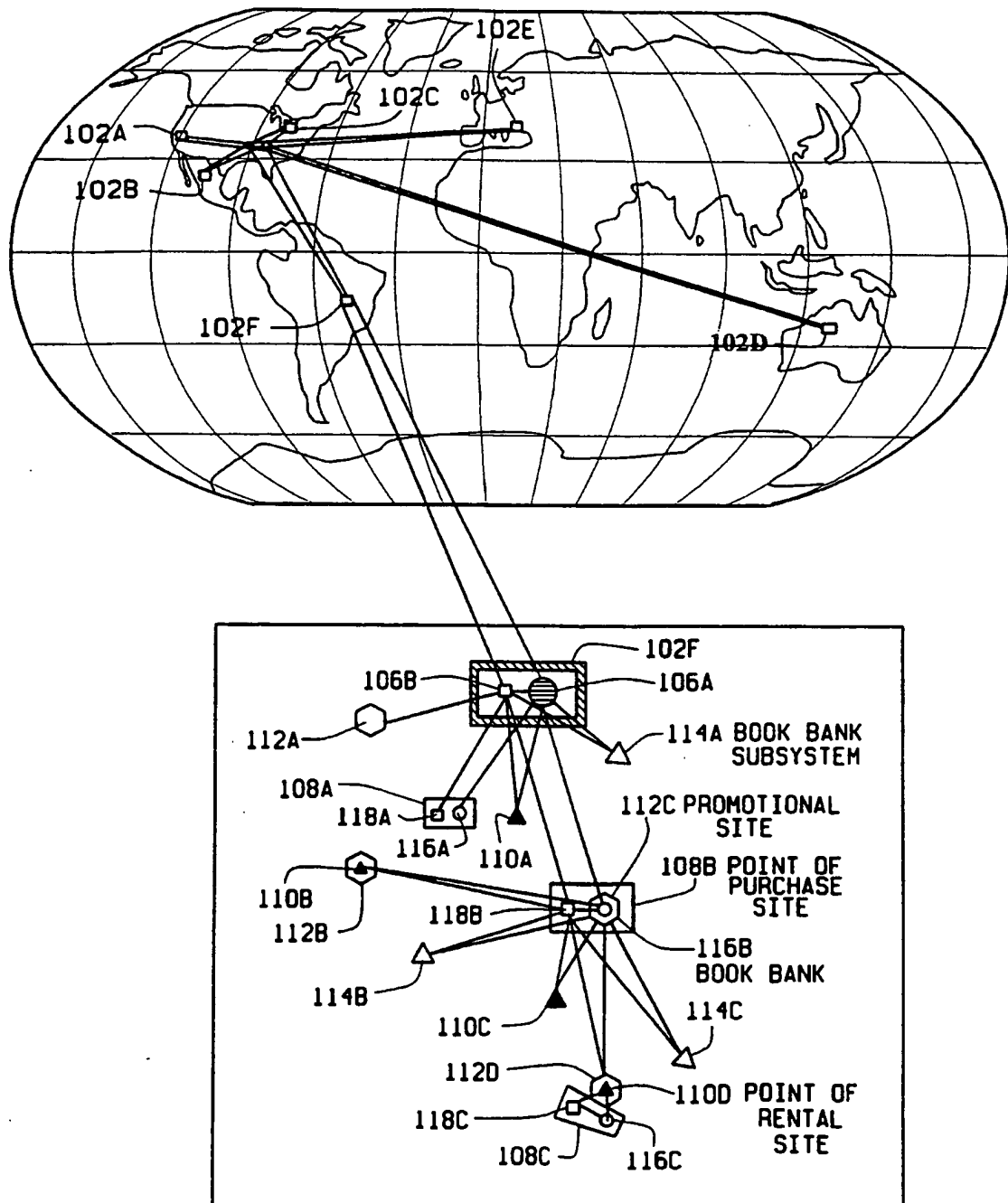


FIG. 1

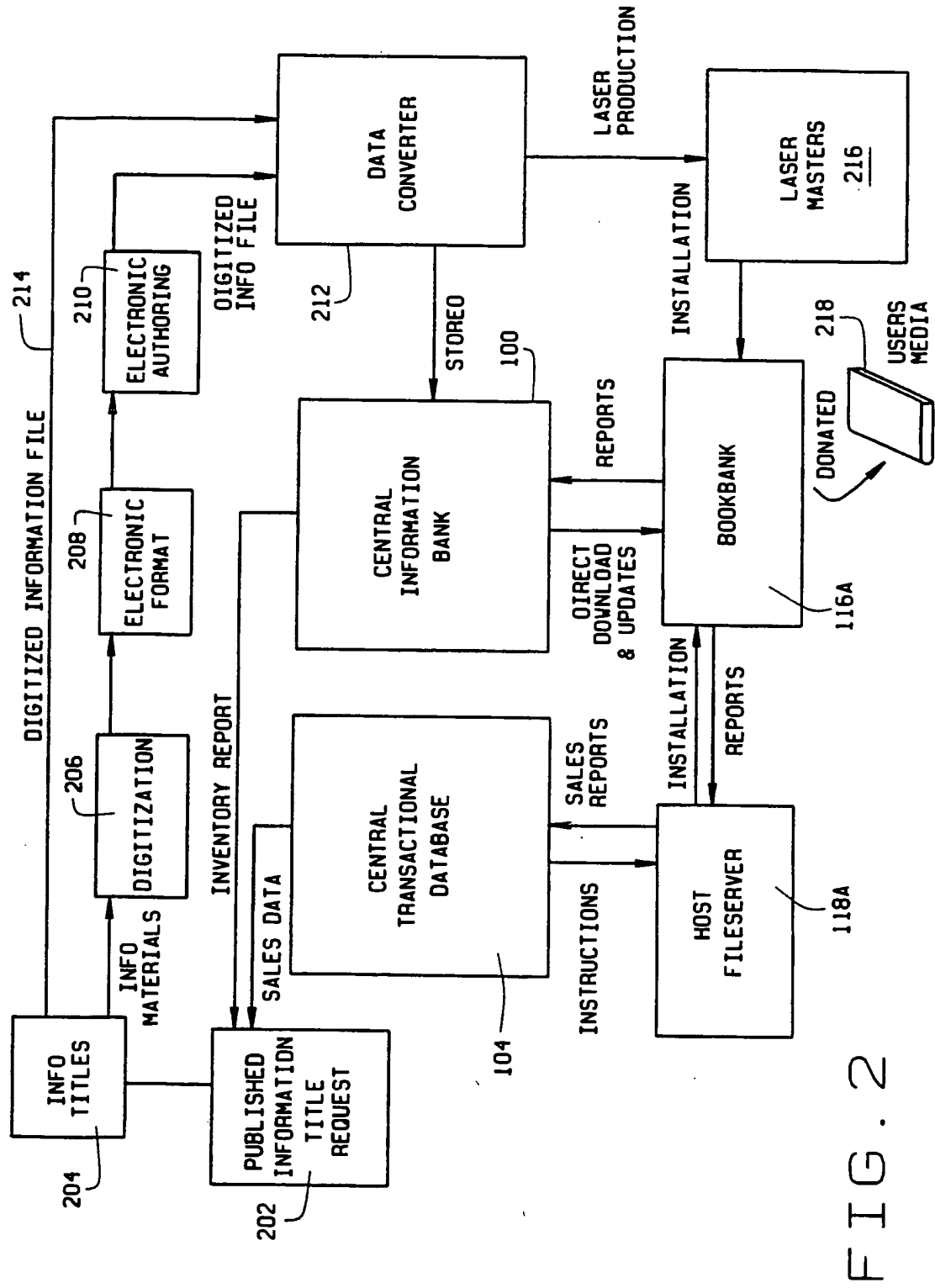


FIG. 2

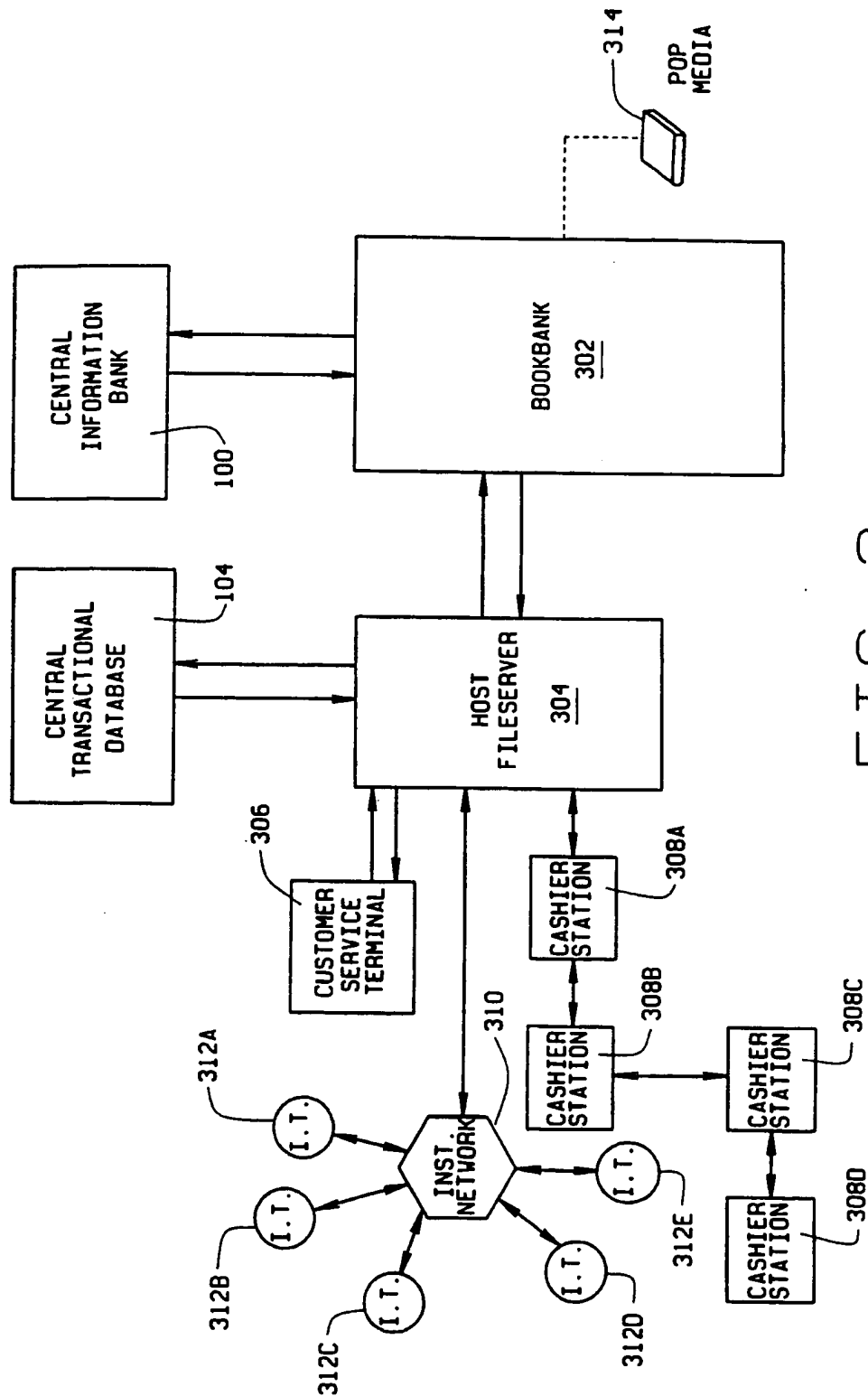


FIG. 3

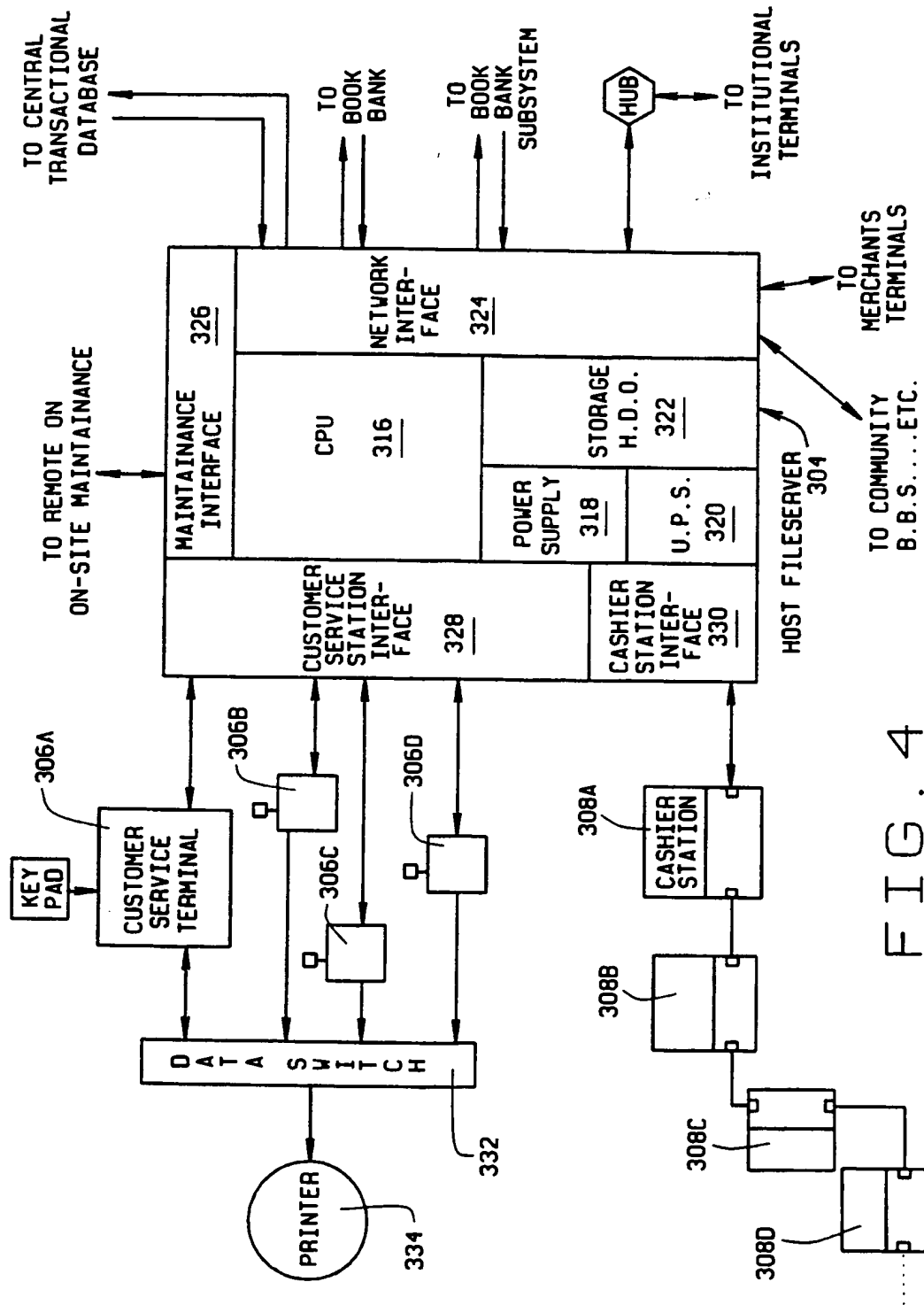
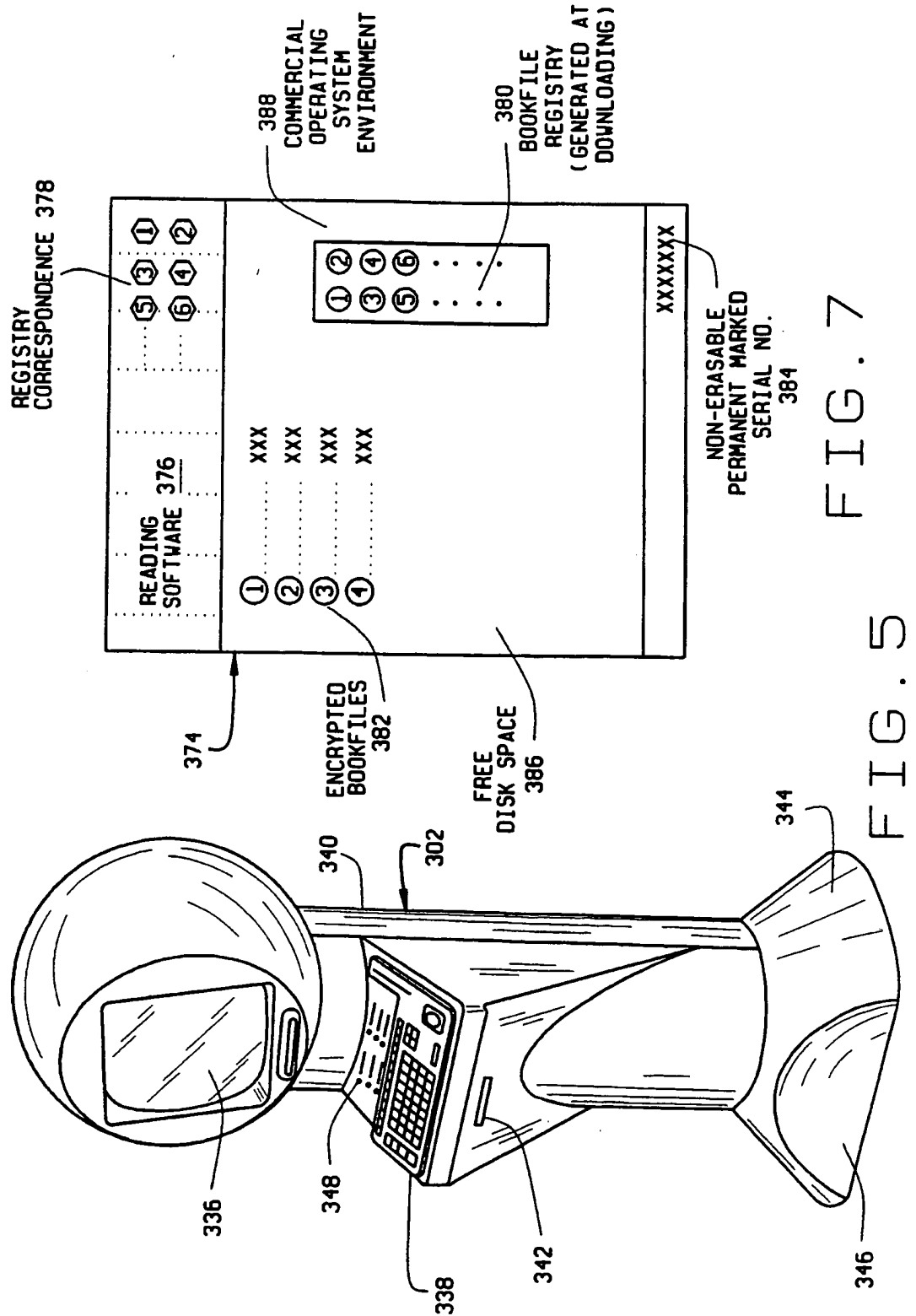


FIG. 4



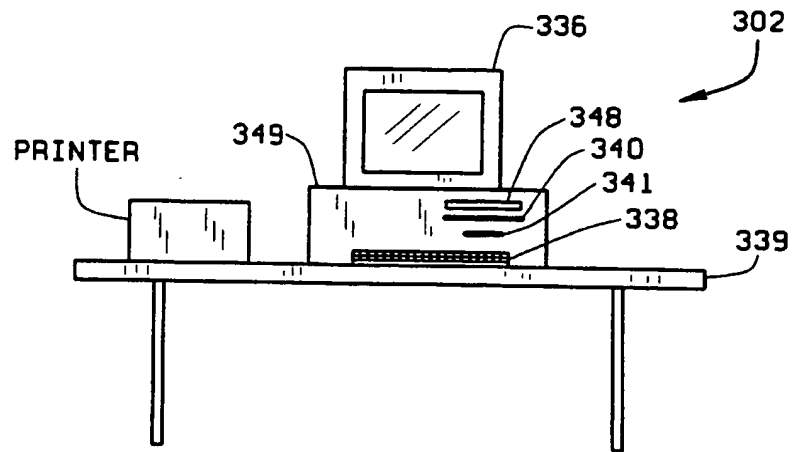


FIG. 5A

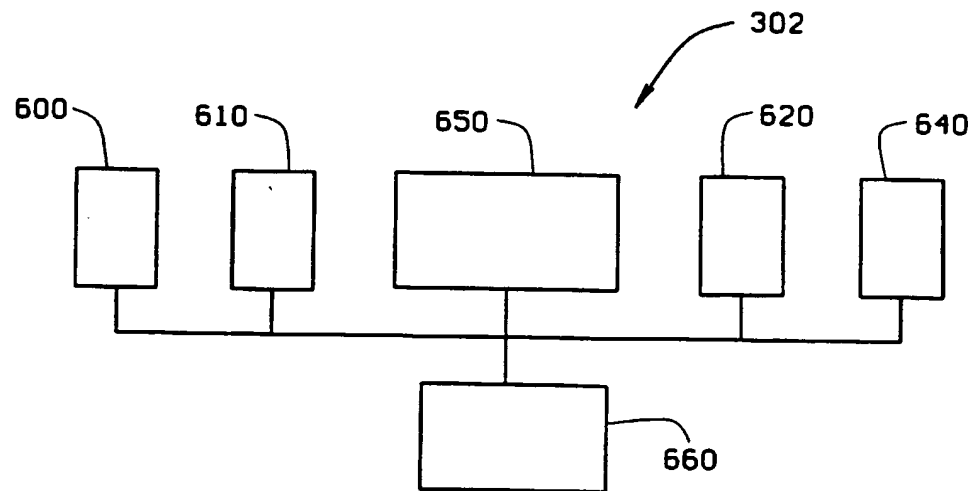


FIG. 13

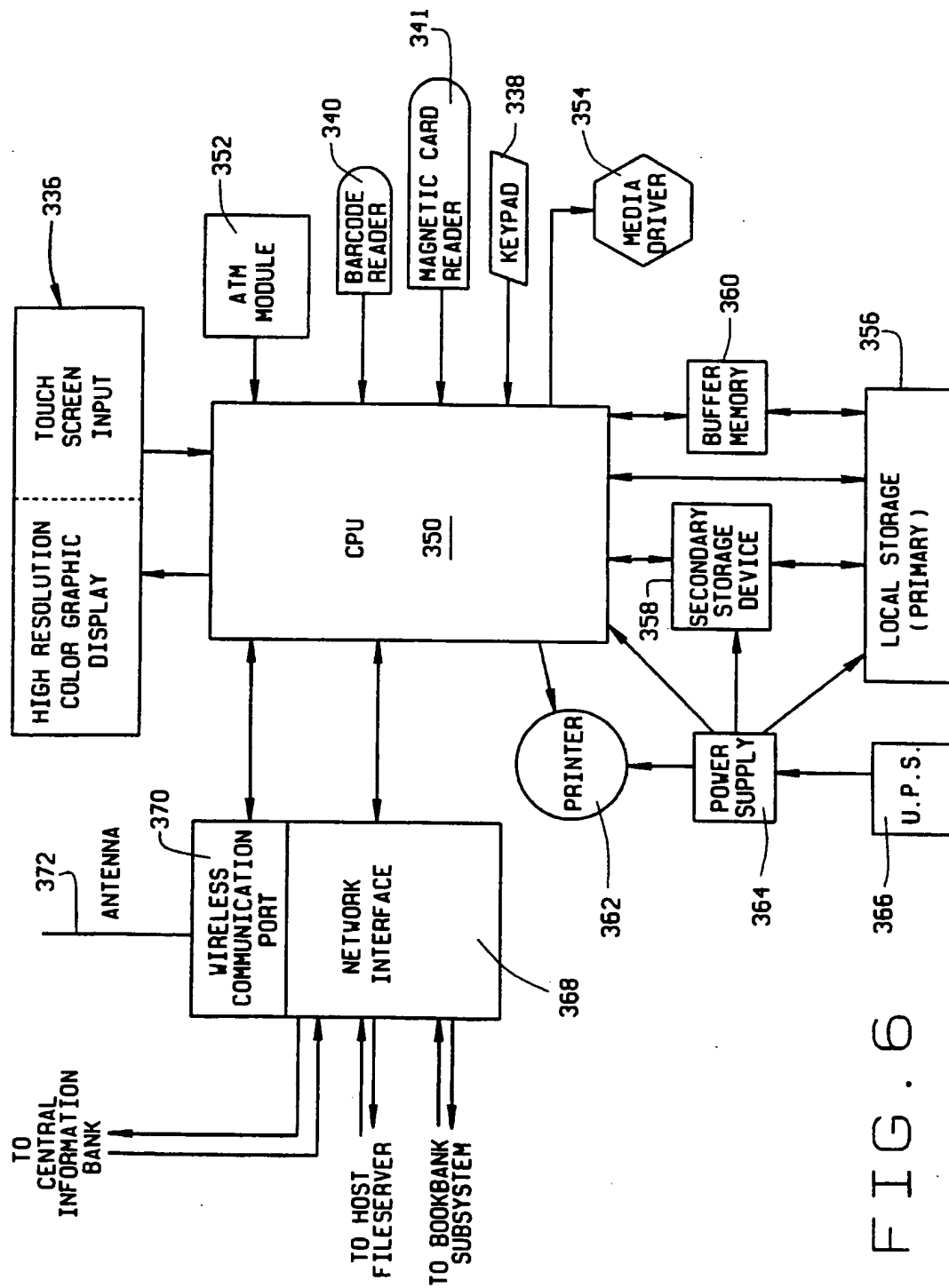


FIG. 6

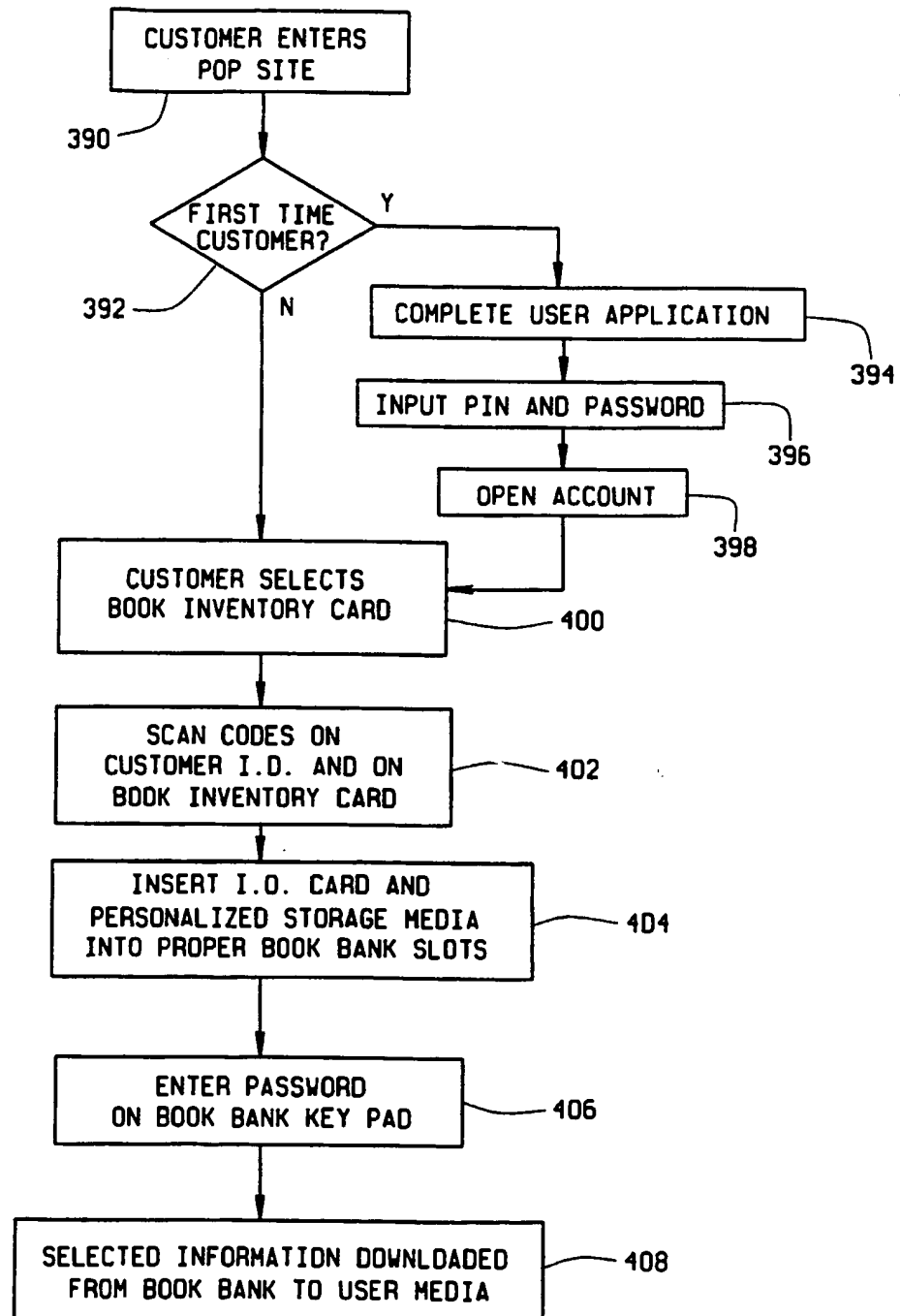


FIG. 8

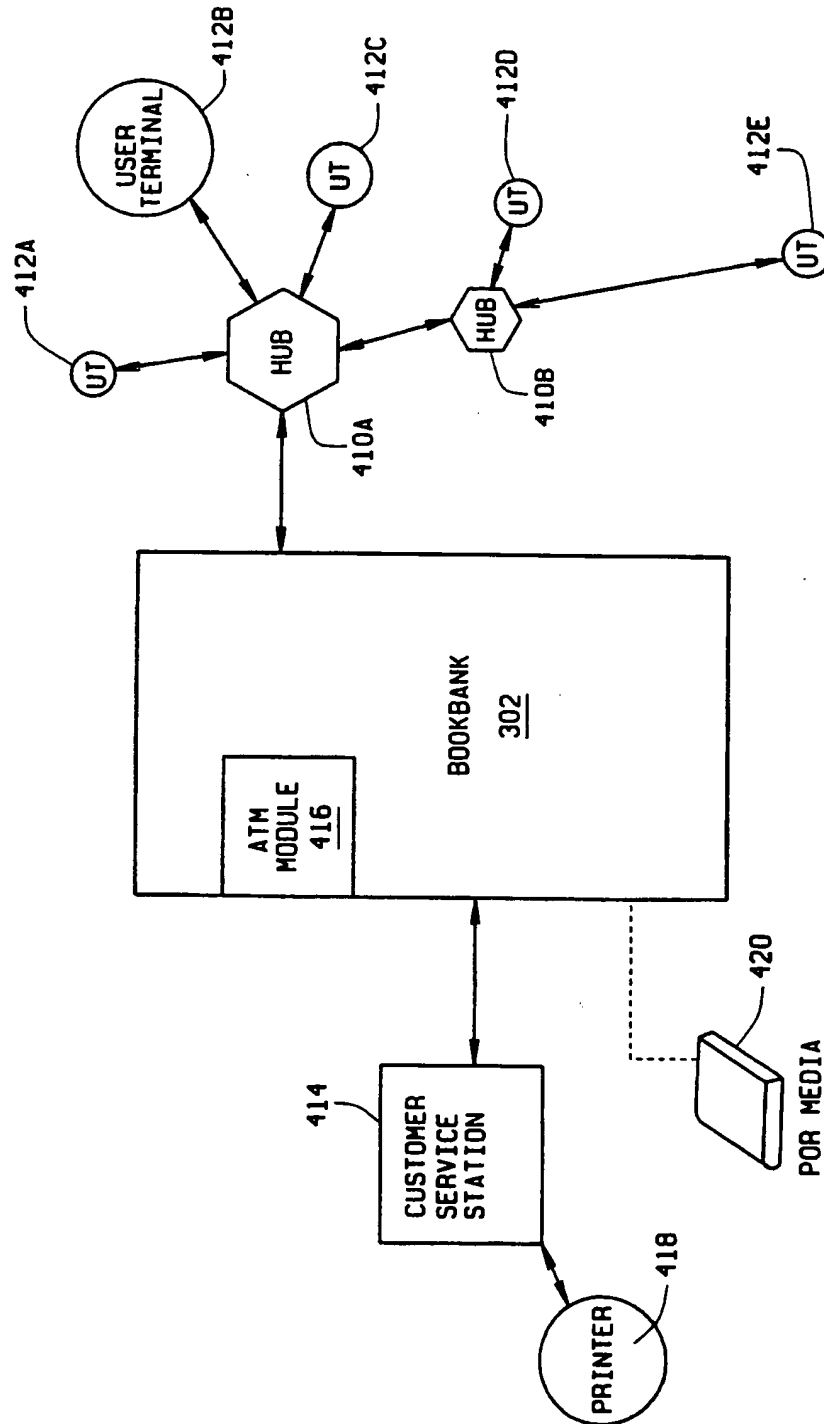


FIG. 9

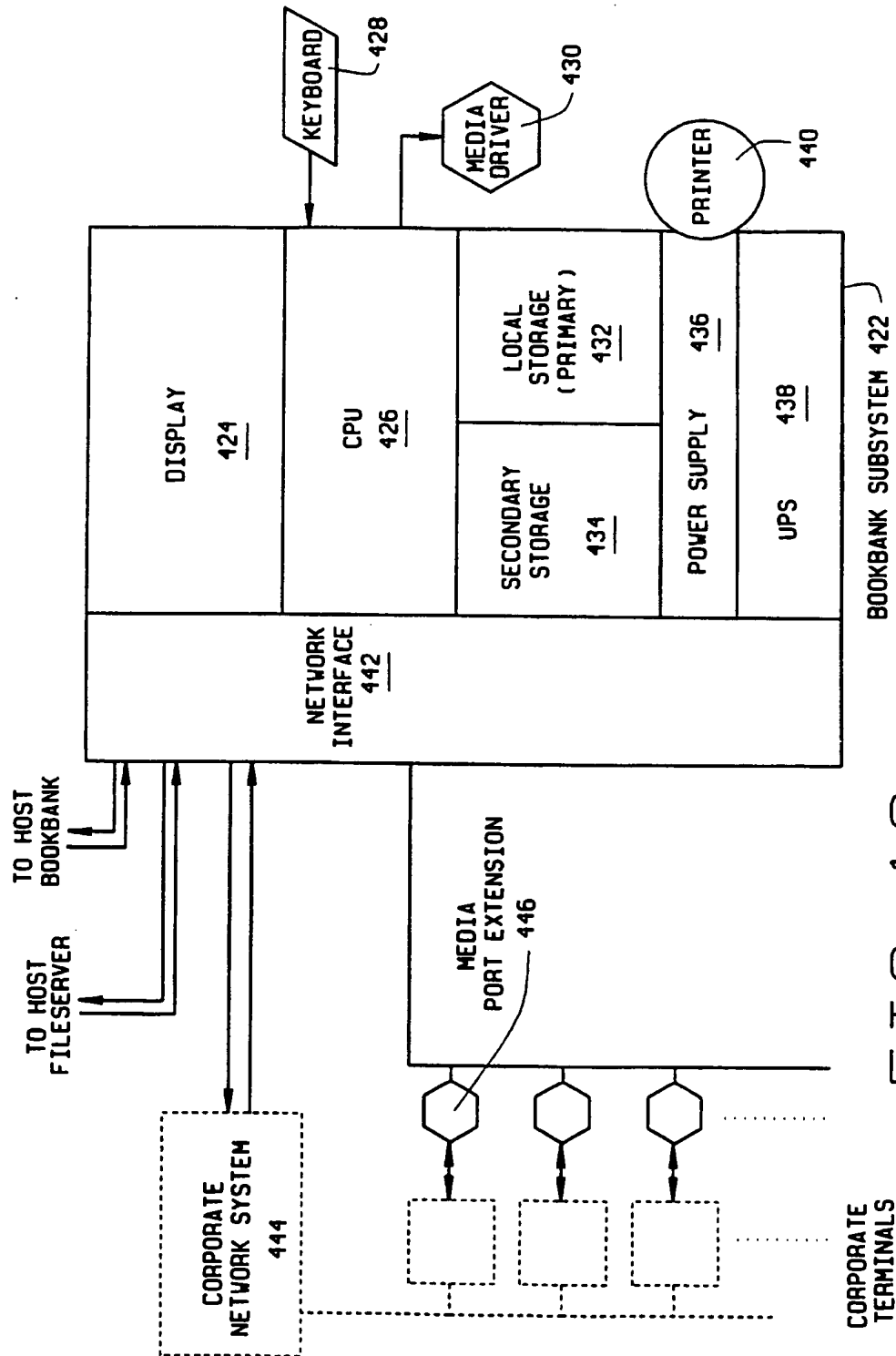
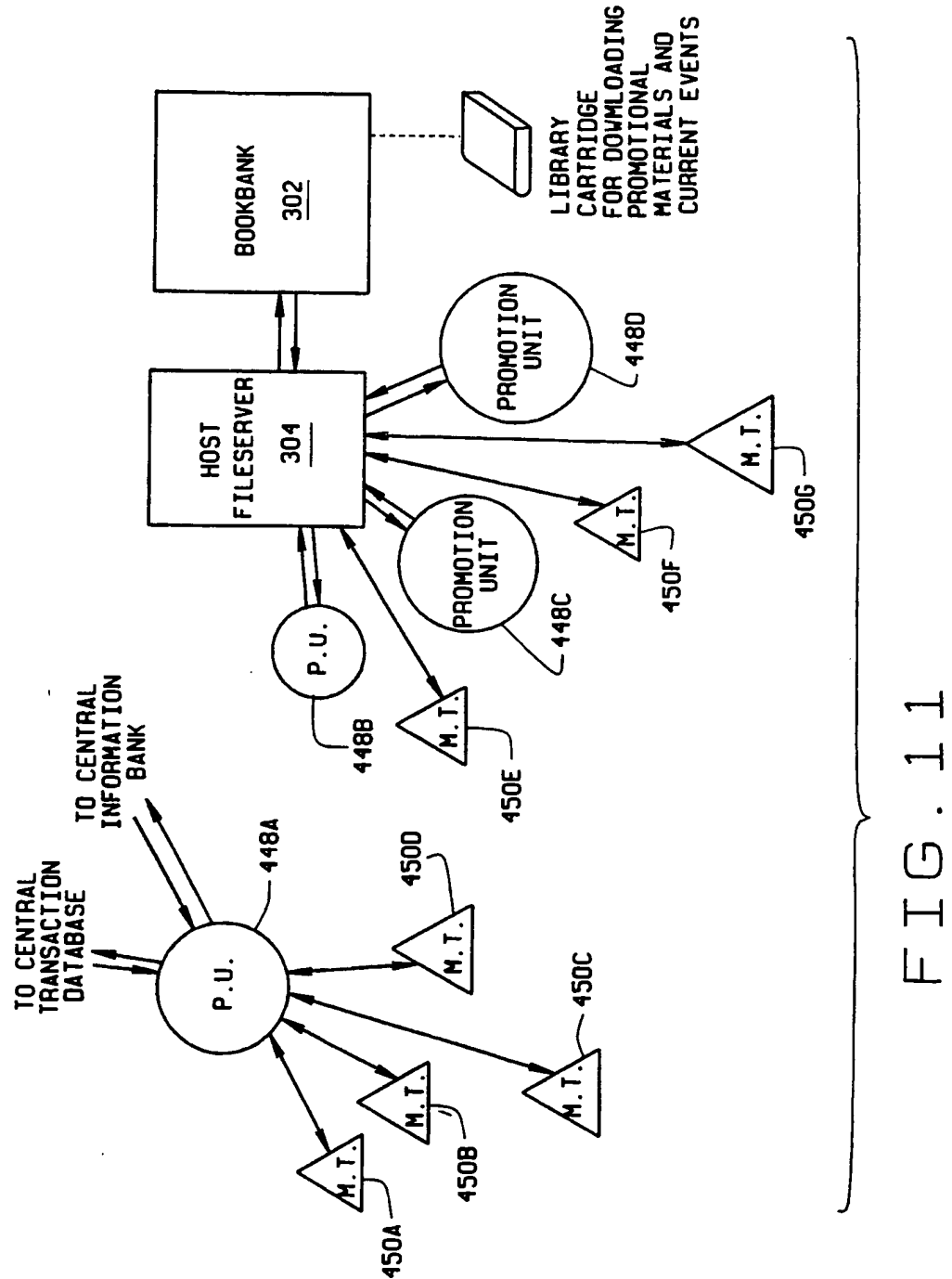


FIG. 10



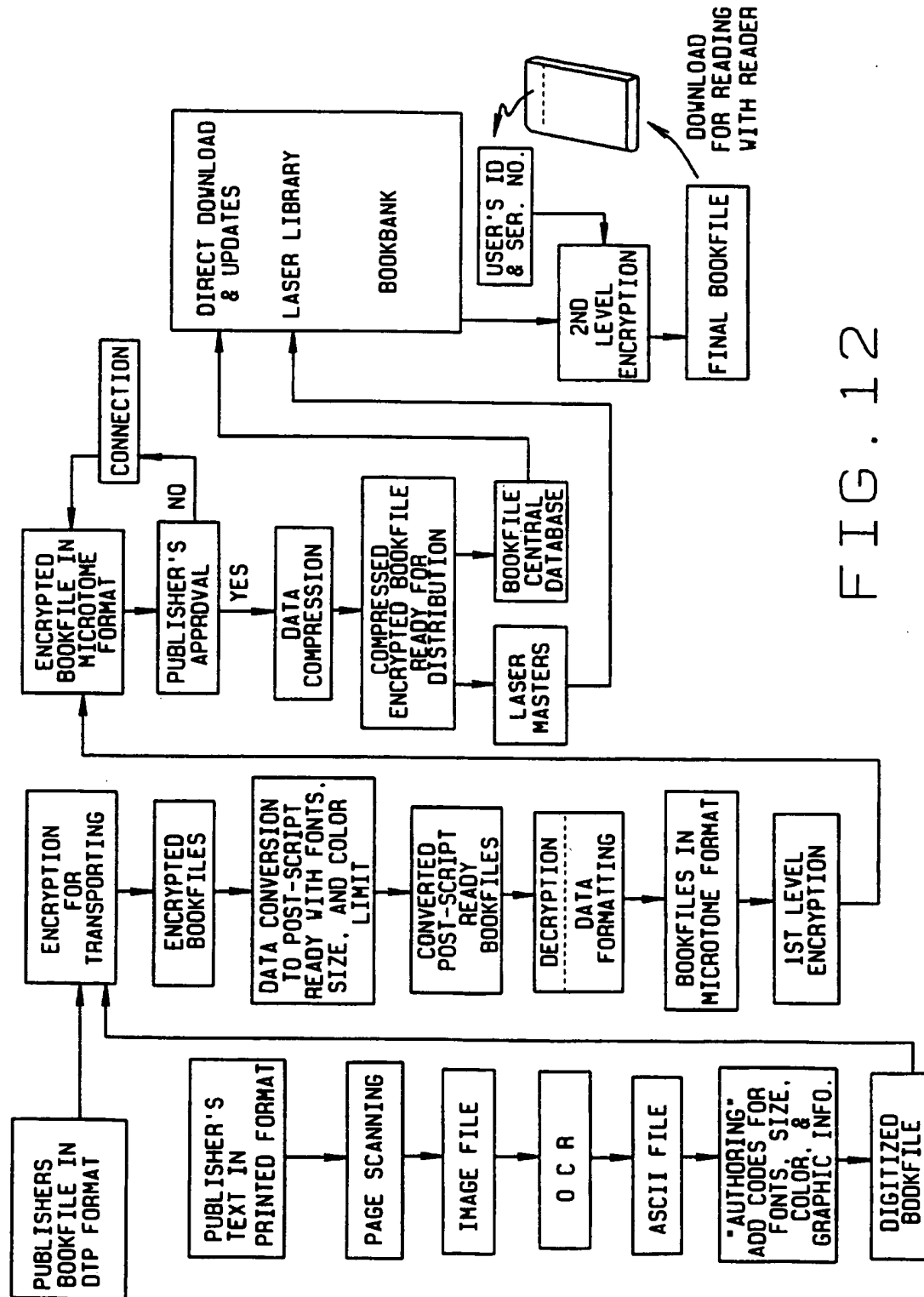


FIG. 12

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/22238

A. CLASSIFICATION OF SUBJECT MATTER														
IPC(6) : H 04 L 9/32 US CL : 380/4,25,49 According to International Patent Classification (IPC) or to both national classification and IPC														
B. FIELDS SEARCHED														
Minimum documentation searched (classification system followed by classification symbols) U.S. : 380/4,25,49														
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched														
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) Please See Extra Sheet.														
C. DOCUMENTS CONSIDERED TO BE RELEVANT														
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.												
Y	US 5,708,709 A(ROSE) 13 January 1998, column 3, lines 11-47, column 5, lines 1-7, column 7, lines 16-31, column 8, lines 65-67, column 9, lines 1-8	1-3,5-6,8-14,16-36												
Y	US 5,680,453 A (AKIYAMA ET AL.) 21 October 1997, abstract, column 3, lines 39- 48, column 5, lines 62-66, column 6, lines 58-67	1-3,5-6,8-14,16-36												
A	US 5,532,920 A (HARTRICK ET AL.) 02 July 1996, column 4, lines 39-65, column 9, lines 6-15, column 16, lines 40-67	1-3,5-6,8-14,16-36												
A	US 5,694,469 A (LE RUE) 02 December 1997, abstract, column 4, lines 20-33, column 3, lines 51-67, column 5, lines 48-53	1-3,5-6,8-14,16-36												
☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.														
<table border="0"> <tr> <td>* Special categories of cited documents:</td> <td>*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</td> </tr> <tr> <td>*A* document defining the general state of the art which is not considered to be of particular relevance</td> <td>*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</td> </tr> <tr> <td>*B* earlier document published on or after the international filing date</td> <td>*Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</td> </tr> <tr> <td>*L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</td> <td>*A* document member of the same patent family</td> </tr> <tr> <td>*O* document referring to an oral disclosure, use, exhibition or other means</td> <td></td> </tr> <tr> <td>*P* document published prior to the international filing date but later than the priority date claimed</td> <td></td> </tr> </table>			* Special categories of cited documents:	*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention	*A* document defining the general state of the art which is not considered to be of particular relevance	*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone	*B* earlier document published on or after the international filing date	*Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art	*L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*A* document member of the same patent family	*O* document referring to an oral disclosure, use, exhibition or other means		*P* document published prior to the international filing date but later than the priority date claimed	
* Special categories of cited documents:	*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention													
A document defining the general state of the art which is not considered to be of particular relevance	*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone													
B earlier document published on or after the international filing date	*Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art													
L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*A* document member of the same patent family													
O document referring to an oral disclosure, use, exhibition or other means														
P document published prior to the international filing date but later than the priority date claimed														
Date of the actual completion of the international search 05 MARCH 1999		Date of mailing of the international search report 14 APR 1999												
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230		Authorized officer GAIL HAYES <i>James R. Matthews</i> Telephone No. (703) 305-9711												

Form PCT/ISA/210 (second sheet)(July 1992)*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/22238

B. FIELDS SEARCHED

Electronic data bases consulted (Name of data base and where practicable terms used):

APS

search terms: download,transmit,request,send,receive,software,information,program,content,file,video,image
s,www,internet,world wide web,cipher,cypher,encrypt,encode,scramble,access code code,PIN,secret code,key

A Publishing and Royalty Model for Networked Documents

by Theodor Holm Nelson

INTRODUCTION For some years the Xanadu* project has been planning a royalty publishing service for networked documents. The proposed system has a number of aspects, some of which are subtle.

THE ROLES "Publication" under most systems of law means making a document public. A *publisher*, then, is whoever, or whatever entity, commits the act of publishing. The role of publisher corresponds to the paper publisher: that person or entity which takes the initiative of publication, receives the profits and is sued for the contents.

(The role of *author* may or may not be different from the publisher; but his or her presence is not formally acknowledged within the contractual system; it is assumed that the publisher has made appropriate arrangements with the author, and it is up to the publisher to pay the author in whatever way they have agreed.)

The role of *service provider* is like that of *printer* and *distributor* in the paper world. The publisher contracts with the service provider for the material to be reproduced and distributed, just as the publisher now does in the paper world.

The role of *customer* is like that of "customer" in the ordinary paper world. But in the paper world, granularity is large: magazines, books and newspapers are the units of sale. In our model, the unit of sale can be as small as one byte.

Note, of course, that these roles may overlap. In the network community we anticipate a customer will often be both author and publisher, as well.

THE PUBLISHING ARRANGEMENT Publishing consists of network storage and delivery of documents, voluntarily and explicitly put on line by publishers, and delivery by such fragments as customers request. Nothing is sent but what the reader asks for. The customer pays on a per-byte basis for all published materials sent by the server.

* "Xanadu" is a service and trademark for services and software of Project Xanadu.

THE ROYALTY ARRANGEMENT

The publisher sets the price-per-byte of the document, or of its sections, if they differ. The customer sends for arbitrary portions (up to the whole document), paying the royalty for each byte transmitted. (Note that other royalty arrangements have been mooted during the life of the Xanadu project.)

THE CONTRACTS**Contract between publisher and service provider**

The publisher contracts with the service provider for the storage of the document and its sale by arbitrarily small fragment. The service provider promises to send the royalties for each sent byte. The service provider also agrees to forward materials to other service providers as needed to provide the service throughout the network.

In this contract, the publisher also represents that he/she/it is the *rightful* publisher, and further agrees to be responsible for any disagreeable consequences under law or tort (violations of national security, privacy, copyright, etc.). These same understandings ordinarily hold in paper publishing, but are not made explicit.

Contract between customer and service provider

The service provider agrees to send materials on demand to the customer. The customer agrees with the service provider to pay for materials sent—both royalties at rates specified by the publishers, and delivery fees to the service provider (as separately negotiated). The customer further agrees only to “fair use” of materials received—a copy for use, printed out if desired, and copies for backup—but no further distribution.

Contract between author and publisher

This is assumed, but not formally required. The author agrees to the sale of materials by the small fragment, and to the various consequences thereof.

TRANSLUSION

In our software (still under development), we implement a special pointer which we call *transclusion*, a handy term for *virtual inclusion by reference across a document boundary*. A transclusion pointer from Document Y to a paragraph in Document Z means that the paragraph is logically and virtually a part of Document Y.

**SPECIAL CONSEQUENCES
OF TRANSLUSION FOR
QUOTATION**

A problem of universal concern is the issue of copyright violation (from the point of view of publishers) or the restriction of freedom (from the point of view of authors wishing to quote other documents). We believe our model nicely resolves the two motivational thrusts.

The transclusion pointer means that any author is free to quote any document already published under this system, since the publisher of the other document has already given contractual permission for sale by fragment. The quoted materials are thus purchased automatically by the reader from the original publisher at the time of delivery.

CONCLUSION There is little question that publishing with royalty on electronic networks will become a principal feature of the world of information. Sale only of whole documents is a frustrating practice with limited usefulness. Sale by user-specified fragment makes transclusion widely practical, making both possible and fair to all parties many varieties of use which are currently frustrated within the system of copyright.

However, without contractual recognition of the varied possible ramifications, many parties may get into difficult situations.

BIBLIOGRAPHY Nelson, Theodor Holm, *Literary Machines 93.1*. \$25 prepaid (\$40 foreign) from Mindful Press, 3020 Bridgeway #295, Sausalito, CA 94965.

Xanadu Operating Company, "Xanadu Hypermedia Server Developer Documentation," July 1992. \$150 prepaid (\$200 foreign) from Mindful Press, 3020 Bridgeway #295, Sausalito, CA 94965.

Theodor Holm Nelson
Project Xanadu
3020 Bridgeway #295
Sausalito, CA 94965

CONCLUSION

A Method for Protecting Copyright on Networks

by Gary N. Griswold

ABSTRACT This solution to copyright protection uses software envelopes which authenticate each access by communicating with an authorization server on a wide area network. It decrypts the information for display, print, or copying when the authorization is approved. This method is specifically suited to controlling information which has been delivered to customer machines over a wide area network.

MOTIVATION Many celebrate the freer environment of electronic networks: the ease of data modification, copying, and multiple use usher in a relaxed attitude toward copyright. They believe that copyright holders must accept the less controlled environment of electronic networks. However, they are ignoring the property rights granted to authors and publishers in article 1, section 8, item 8 of the U.S. Constitution. The decision to place intellectual property on electronic networks is the prerogative of rights holders.

Because publishers do not share this relaxed vision of copyright, the current providers of electronic services are delivering information which does not require extraordinary protection, for example: open discussions, such as USENET; perishable information, such as news services; and government information, such as patent databases. However, any new system which wishes to leverage its content from the trillions of dollars in intellectual property already existing in the world, must address the property owner's concerns of property protection, or risk losing their cooperation.

Mr. Timothy King, Vice President of Corporate Development at John Wiley and Sons, has identified the following key concerns.

- Will the integrity of information be preserved?
- Will attribution for all information be ensured?
- Will the quality of the content and form of information be maintained? Will creators and copyright holders be able to control the use of their work and to receive compensation for that use?

The legal problem must be solved. The High Performance Computing and Communications Act of 1991 (HPCC) specifically requires that the National Research and Education Network (NREN) include a means to protect copyright:

(c) NETWORK CHARACTERISTICS. — The Network shall —

...

- (5) be designed and operated so as to ensure the continued application of laws that provide network and information resources security measures, including those that protect copyright and other intellectual property rights, and those that control access to data bases and protect national security;
- (6) have accounting mechanisms which allow users or groups of users to be charged for their usage of copyrighted materials available over the Network and, where appropriate and technically feasible, for their usage of the Network;²

To date, the problem remains unresolved. In his December 8th, 1992 presentation to Congress on the NREN, Dr. Allen Bromley, Director of the Office of Science and Technology Policy, had the following to say about the current status of copyright protection.

The technical mechanism appropriate to protect copyright of material distributed over the network is as yet unclear. ... Because consensus has not been reached in this complex area, implementation of technical measures on the Network has not yet been scheduled.³

There are an abundance of applications which require a solution to this problem if they are to be performed legally and without negative implications for publishers. Libraries, which are currently using FAX for inter-library loan, are looking forward to delivering the journals over the NREN. Likewise, the Colorado Alliance of Research Libraries (CARL) Uncover Project and Engineering Information's Article Express are looking forward to NREN delivery. The CUPID project (Consortium of University Publishing and Information Distribution) is planning a distributed network architecture that will permit university presses to establish servers containing their copyrighted products in electronic form. These university press servers will be used for distributed publishing on the Internet. Libraries have made extensive progress in putting bibliographic information on-line, and look forward to implementing digital libraries in which they deliver copyrighted information. Also, information retrieval systems, such as Wide-Area Information Service (WAIS), deliver the query result to the machine of the customer. At present, such systems are not being used for the delivery of information that requires protection. One can also conceive of additional applications which could appear once adequate copyright protection were available. For example, news could be delivered by broadcast over the NREN, but only received by subscribers. Means to filter the information to subscriber requirements would also be part of such a system. Journal subscriptions could be delivered electronically. That is, each month a copy of the latest journals could be file transferred to the machines of each subscriber. Also, an electronic retail service could be provided so that customers could search by author, title, and subject indexes and request electronic delivery of titles they wished to purchase.

BACKGROUND

Many solutions to this problem have been suggested. The following is a discussion of some of the more important.

Many have suggested a simple system: A library charges for each transmitted article and pays the publisher or the Copyright Clearance Center a royalty for each copy. This method is being used effectively by CARL (Colorado Alliance of Research Libraries) for FAXed journal articles.⁴ However, as we move to the electronic distribution of information, the ease with which information can be repeatedly distributed, for no fee after the first distribution, threatens the prudence of using this approach on computer networks.

Digital signature use of public key encryption has been suggested as a means to protect copyright. A hashing algorithm is used to create a unique number from the content of a document. This number is encrypted with the private key of the originator. The receiver of such a document can obtain the public key of the assumed source of the document from a central key facility.⁵ However, while this important technology verifies the source and content of the document, it does nothing to prevent the creation or use of copies.

Public key encryption has also been suggested as a way to encrypt information. By using the public key of the receiver, only the receiver can decrypt it with their private key. However, while this is mathematically very secure, nothing prevents people from distributing encrypted information along with their private keys. The elegant security of public key encryption prevents anyone from identifying the source of the offending private key and copyright infringement.

John H. Ryder and Susanna Smith describe a simple solution for the electronic dissemination of software. Before the customer receives the copyrighted software product in working form, he or she is presented with a number of screens of text which display a license agreement. The customer must follow certain steps on the keyboard to signify that they agree to the terms of the license agreement.⁶ However, while this method makes certain the customer understands their licensing rights, it does nothing to insure that the customer lives up to those obligations.

Martin E. Hellman describes a means to limit access and bill usage of software, video games, video disks, and videotapes. This is accomplished via an encrypted authorization code, which contains information related to an identification of the computer, a product, a number of uses requested, and a random or non-repeating number. When entered into the customer's base unit, the authorization code permits use of the specified software product for the specified time.⁷

Victor H. Shear describes a system and method to meter the usage of distributed databases, such as CD-ROM. This method describes a hardware module which must be part of the computer used to access the distributed database. This module retains records of the intellectual property viewed. Once the module becomes full, it must be removed and delivered to someone who will charge for the usage and set the module back to zero.⁸

Hellman's and Shear's methods both require hardware modules, which must be constructed into the customer's computer, in order to control access. These methods will not be practical until a very large number of computers contain these modules. Hardware manufacturers will be

hesitant to include these modules in the design of their computers until there is sufficient demand for these specific systems.

TECHNOLOGY

A solution to the copyright protection problem is described in the following section. Patent applications have been filed on the pivotal aspects of the innovation.^{9,10,11}

Description of the Innovation

Our approach is as follows: copyrighted information is transmitted in an encrypted form, and is transmitted in a software "envelope". The copyrighted information and the software envelope together comprise an executable program which can decrypt the copyrighted information and present it to the user. The capabilities of the envelope intentionally limit the user's access to the copyrighted information to those capabilities which are appropriate under copyright law for the specific kind of copyrighted information contained. For database information, the software envelope should enable the user to search indexes and display text. For CAD information, the software envelope should permit the display of the information and permit the user to manipulate attributes of the display. For video information, the software envelope should display the video. For audio information, the software envelope should display the audio information. For text, the software envelope should display and turn pages. For hypertext information, the software envelope should allow the user to thread through the information. These are only some of the ways these software envelopes can control different kinds of copyrighted information.

Finally, the software envelope uses a method to check for authorization to access and to track the usage of the software envelope and copyrighted information over the same telecommunication network used to transmit them to the user. The tracking method works as follows. Automatic messages are sent between the software envelope and a central authorizing site. Each time a customer starts to use a copyrighted work, a message is automatically sent from the work. Also, at a regular interval, additional messages are sent. Sent at regular intervals, they are a measure of use. When the messages arrive at the central authorizing server they are verified. A reply is sent back, which is an authorization to continue or a denial of authorization. If no valid message returns, a denial is assumed by the software envelope. Whenever a denial is received or assumed, the use of the software or copyrighted information product is discontinued. The diagram in Figure 1 illustrates this method of tracking copyrighted information.

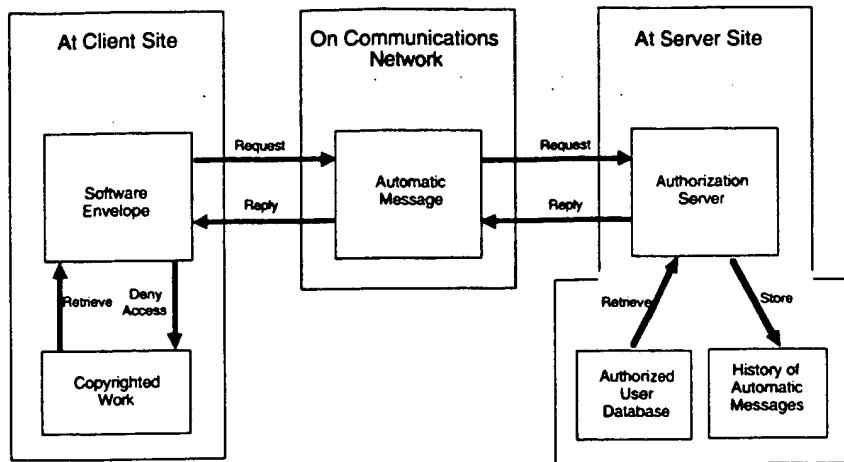


Figure 1 - Overview of Copyright Tracking Mechanism

Benefits of the Innovation

The system of authorization and usage measurement capabilities described above can be used to license information products in a variety of ways to suit a variety of information licensing policies. It can be used to enforce site licenses by preventing off-site access and limiting the number of concurrent uses. It can be used to limit duration of use, analogous to returning a book to a library, by disabling use of an information product after a period of time. It can be used to implement an electronic subscription by providing an unending duration of use of the product on one machine. It can also be used to meter and charge for each use of the information.

The software envelope would provide the user with the ability to view the information product, but it would not provide any way to edit or extract from it. This is needed, because otherwise the displayed information could be used as a source from which to create a new copy which is not subject to this copyright protection scheme. Second, it would insure the authenticity of the information products, by preventing the automatic creation of altered copies. Third, it would interfere with plagiarism, which has become an increasing problem because of the abundance of easily copyable electronic information. Fourth, it would prevent the automated generation of derivative works.

Other Licensing Requirements

So far, we have only discussed controlling licenses for viewing information, but the same method can be used to control licensed printing. While the rightsholder may choose to give the customer a license to view and to print, they could require an additional expense for the action of printing. In this case, the authorization request would indicate that printing is requested and the reply would indicate whether the customer is licensed. The act of printing would be recorded for the purpose of charging. In some computer operating system environments, insuring the security of the document will require the installation of a special print server, which is capable of decrypting while printing.

This system permits unlimited copying on the network, and yet limits the use of those copies to licensed customers. However, a customer may need to take an electronic copy of a document onto a machine which is not connected to the Internet. For machines which contain internally readable serial numbers or firmware private keys, we can license and control the act of making copies. Each copy made will contain the internal identifiers of the machine on which it is to run. It will still be encrypted, and requires a similar software envelope for presentation. Instead of checking for further authorization over the network, the software envelope checks that it is running on the machine to which it is licensed.

Network Infrastructure

This method assumes the existence of a network used in the delivery of electronic information. This network should also be capable of sending connectionless datagrams. Analog telephone is both too slow for sending large amounts of data, and would require an explicit telephone call with each use of an information product. Integrated Services Digital Network (ISDN) telephone, because of its minimum 64 K bps speed, would be much more suitable for the transmission of information products. Also, the authorization datagrams which this method requires could be sent over the signaling channel without placing a call. Similarly, on the Internet, the authorization datagrams can be most efficiently transmitted and processed as User Datagram Protocol (UDP) datagrams. Digital Cellular would also be a very suitable network.

DEMONSTRATION PROTOTYPE

Capabilities

At this time, we have a demonstration version of our technology running on the Internet. The system consists of three main programs: 1) a license authorization program called "authorize"; 2) a program for creating protected files called "product"; 3) and a program for viewing the protected files called "read". The authorization server runs on one machine on the Internet in Albany NY, and will control access to any documents created using the "product" program. Copies of "product" and "read" are available upon request.

Limitations

While the above prototype has many capabilities, it has many limitations which make it less than a commercial product. While it does register the creation of new protected products, authorizes access, tracks usage, and permits customers to register upon receiving a denial, it does not include a customer billing module or a publisher payment module. While the software envelope provides the essential features needed to display the decrypted information, it lacks the user interface quality one would expect in a commercial product. Finally, the viewer program is written to run on Sparcstations. Versions are not yet available for other computers. Despite all of the above limitations, the Demonstration Prototype performs an important service by demonstrating how licenses can be managed over the Internet.

COMMERCIAL PROTOTYPE

The next step will be to develop a commercial prototype of the technology. We will be able to proceed with this step as soon as the necessary funding is available. This system should be limited in the number of products sold and the number of customers serviced in order to facilitate revision of the system as we learn from its use. However, this system should provide the full scope of functionality required in a commercial version. That is, it should manage licenses for viewing, printing and node-locked copying, and it should maintain a full database about its customers and publishers, which should be used to bill customers and pay publishers. The system should provide a higher quality presentation program which is available on a wide variety of platforms. Such a viewer could be developed by InfoLogic, but it would be more efficient to have the developers of an existing viewer integrate InfoLogic's license control mechanism into their viewer. Finally, the license server will be redundantly implemented to guarantee 100% uptime.

APPLICATIONS

There are a variety of applications for which the described method of copyright license management would be very useful. These include: electronic retailing, inter-library loan, library circulation, and distributed information services. The following is a description of how each of these applications could function using the copyright protection mechanisms described in this report.

Electronic Retailing

Publishers and printers have automated their methods of production so that typeset copies of books or journals exist in electronic forms, such as Standard Graphics Markup Language (SGML) or Postscript. From these electronic copies, the pages are printed. These same electronic forms are a useful source for electronic distribution. In addition, scanned copies of older books are a source of electronic distribution.

After printing their books and journals, the publisher could license the electronic sources to the electronic retailer. The only task the publisher needs to perform is signing the license agreement. There is no need for a second tier of distribution. The electronic retailer could offer to pay for each copy delivered to the customer. Considering the absence of printing, inventory, warehousing, and returns, the publisher could earn a considerably larger margin than they receive on paper copies. Considering the absence of two-tier distribution in this model, the electronic retailer could sell the copies for less than the cost of paper copies.

Those currently connected to the Internet include most universities; most national laboratories; most private research laboratories doing government work, or collaborating with universities; and a growing number of smaller organizations, especially technical. As a result of this profile, it appears that PSP/STM (Professional Scholarly Publishing and Scientific, Technical and Medical) are the publishing segments where the demand will occur first.

To begin using the system, the customer would request a copy of the electronic retailer's client program over the network. The client program could be delivered free, or for a nominal charge. The first time the customer used this client program, they would be asked to enter identify-

ing information. This program would enable them to browse through the title, author, and subject catalog of books and journals in the electronic retail server. They could request any book, whereupon they would be required to enter charging information, such as a credit card number. The book or journal would be delivered to them electronically.

For universities and organizations the system would permit the site licensing of the information, while at the same time permitting the licensing to individuals or licensing by the duration of time used. People would be able to share electronic documents freely, and all accesses to a site licensed document within the site would be permitted. However, if someone off the licensed site were to receive a copy, they would be denied access when they attempted to access it.

Inter-library Loan and Document Delivery

Inter-library loan and document delivery services are very similar, except that one is a library service and the other commercial; one usually pays copyright royalties while the other usually does not. Using this copyright management method they become even more similar.

When a document is requested for delivery, it is located, scanned into a computer, and immediately converted to an encrypted file. The protected file can be transferred to the requester's machine and a licensing entry permitting one concurrent use of the document can be made at the same time. Once received, the document can be freely accessed by the requester on the machine to which the document was sent. Should the requester pass the document along to others, they will not be able to access the document until they have secured a license to the document. At the same time that they receive a denial of access from the license server, they will be given the opportunity to enter charging information on the screen which will permit them to access the information.

On a periodic basis, the license management system will generate administrative reports which detail the following: 1) library charges for documents delivered; 2) library receipts for documents provided; 3) copyright royalties for documents provided; 4) copyright royalties for additional licensees added to previously delivered documents. These documents could be the basis for payments between libraries and the Copyright Clearance Center.

Library Circulation

A possible use of this technology is for each library to maintain a license server to manage the copies of books and periodicals which have been checked out from their library in electronic form. In addition to the technology previously described, the digital library card catalog must contain a record of the number of copies owned and number of copies borrowed for each item in the electronic card catalog. Such a system would work as follows.

Each time someone wishes to check out an electronic copy of a book or periodical, the current "number owned" by the library and the current "number checked out" from the library would need to be looked up to be certain that a copy is available.

When a book or article is checked out from the library, a licensing entry for the user would be entered into the license database. A termina-

tion date, such as two weeks, would be entered in the license to represent the borrowing period. The card catalog's record of the number of copies checked out from the library would need to be updated to indicate that the copy has been removed from the library.

When the two-week borrowing period of the book or periodical terminates, the copyrighted work would cease to be accessible by the library patron, even though the copy still exists on his or her computer. On a nightly basis, the library's system could look in the licensing database for copies which have terminated on that day and decrease the "number of copies checked out" shown on the electronic card catalog. This action is analogous to returning the book or periodical to the library shelf.

Advantages of Standardization

If this technology were consistently implemented by libraries and electronic retail services, it would be possible for the holder of a copy checked out from the library to purchase the same item from a retail service. The customer would use the software envelope of a retail service to try to access the library copy of the document. Upon getting a denial of access, they would fill out the charging information requested on their screen by the electronic retailer. Once this step was completed, they would be purchasing a copy of the book or periodical.

Distributed Information Services

Currently, providers of on-line services fill their large computers with quantities of information and charge the customers for the use of the infrastructure needed to access that information. Using the methods in this paper, much more efficient information services are possible. For example, one could provide a bibliographic information retrieval service at no cost, since money would be made on the sale of information.

Before using this system, the customer would need to provide certain charging information, such as corporate purchase orders, or credit card numbers. The customer would search the on-line bibliographic database for documents on particular topics. Once documents are selected by the user, the documents or abstracts of the documents could be delivered to the user by file transfer. Access to the information could be measured in a variety of ways. By default, it may make sense to charge the customer for the time each document is accessed. Time would be measured in intervals, such as every 15 minutes. In addition, the customer could be charged for printing out a copy of the documents. Finally, the customer could be given the opportunity to purchase permanent electronic copies that they may store and view at any time without further charge. The license servers can be apprised of these events by automatic messages, sent between the software envelopes and the license server.

CONCLUSION

One of the side effects of these methods of distribution is to lower the amount of infrastructure needed to deliver information, because most of the information access occurs on the customer's own computer. Lowering the cost can in turn lower price and thus increase profit. Any lowering of price of the currently expensive electronic information is apt to

increase demand. We need to build into our selling systems a positive feedback loop which would lower costs of operation, to lower prices, and increase demand. Increased demand would lower the per unit production costs, which increases demand even more. At the same time, we must retain and even increase the use of peer review and editorial filtering to insure the availability of the highest quality information. This technology facilitates the lowering of operational costs, while providing a mechanism to compensate for the time and effort that went into production.

NOTES

1. Tim King, "Critical Issues for Providers of Network Accessible Information", *EDUCOM*; Summer 1991, Page 82.
2. High Performance Computing and Communications Act of 1991 (HPCC), Section 15 USC 55112 (c).
3. Dr. Allen Bromley, Director of the Office of Science and Technology Policy, "The National Research and Education Network Program: A Report to Congress", December 1992, Page 2.
4. CARL Systems, Inc., Uncover and Uncover2—the Article Access and Delivery Solution, unpublished article, 1992.
5. Public-Key Cryptography Standards, RSA Data Security, Inc., June 1991.
6. John H. Ryder and Susanna R. Smith, "Self-verifying Receipt and Acceptance System for Electronically Delivered Data Objects", *United States Patent 4,953,209*; August 28, 1990.
8. Victor H. Shear, "Database Usage Metering and Protection System and Method", *United States Patent 4,977,594*, December 11, 1990.
9. Gary N. Griswold, "License management system for information products located at user site periodically requesting usage authorization via communication network", *Application for International PCT patent*, 1992.
10. Gary N. Griswold, "System and method for protecting and licensing information products on an electronic network", *Application for United States Patent*, 1992.
11. Gary N. Griswold, "System and method for protecting and licensing software on an electronic network", *Application for United States Patent*, 1991.

BIOGRAPHY

Gary Griswold is President of InfoLogic Software, Inc., a consulting firm which develops software in technical applications including: Very Large Scale Integrated (VLSI), CAD, Image Recognition, Computer Aided Software Engineering (CASE), Manufacturing Automation, and Management Information Systems. Recently, his primary technical interest has been copyright protection for networked information. He holds an M.S. (Union College, Schenectady, NY) and a B.S. (University of Washington, Seattle).

Gary Griswold
InfoLogic
1223 Peoples Avenue
Troy, NY 12180
Tel: (518) 276-4840
e-mail: gary@infologic.com

FAX: (518) 276-4841

Dyad: A System for Using Physically Secure Coprocessors

by J. D. Tygar and Bennet Yee

ABSTRACT Physically secure coprocessors, as used in the Dyad project at Carnegie Mellon University, provide easily implementable solutions to perplexing security problems. This paper presents the solutions to five problems: (1) protecting the integrity of publicly accessible workstations; (2) tamper-proof accounting/audit trails; (3) copy protection; (4) electronic currency without centralized servers; and (5) electronic contracts.

INTRODUCTION AND MOTIVATION

The Dyad project at Carnegie Mellon University is using physically secure coprocessors to achieve new protocols and systems addressing a number of perplexing security problems. These coprocessors can be produced as boards or integrated circuit chips and can be directly inserted in standard workstations or PC-style computers. This paper presents a set of security problems and easily implementable solutions that exploit the power of physically secure coprocessors.

Standard textbook treatments of computer security assert that physical security is a necessary precondition to achieving overall system security. While meeting this condition may seem reasonable for yesterday's computer centers with their large mainframes, it is no longer so easy today. Most modern computer facilities consist of workstations within offices or of personal computers arranged in public access clusters, all of which are networked to file servers. In situations such as these where computation is distributed, physical security is very difficult—if not impossible—to realize. Neither computer clusters, nor offices, nor networks are secure against intruders. An even more difficult problem is posed by a user who may wish to subvert his own machine; for example, a user who wishes to steal a copy protected program can do so by gaining

Partial support was provided by the Avionics Laboratory, Wright Research and Development Center, Aeronautical Systems Division (AFSC), U.S. Air Force, Wright-Patterson AFB, OH 45433-6543 under Contract F33615-90-C-1465, ARPA Order No. 7597. Additional support was provided in part under a Presidential Young Investigator Award, Contract No. CCR-8858087 and matching funds from Motorola Inc. and TRW. Additional partial support was provided by a contract from the U.S. Postal Service. We gratefully acknowledge the generous support of IBM through equipment grants and loans.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Government.

read access to the system memory while the system runs the nominally "execute only" program. By making the processing power of workstations widely and easily available, we've also made the system hardware accessible to casual interlopers. How do we remedy this?

Researchers have recognized the vulnerability of network wires and have brought the tools from cryptography to bear on the problem of non-secure communication networks, and this has led to a variety of key exchange and authentication protocols [15, 16, 20, 35, 37, 45, 46, 53, 54] for use with end-to-end encryption to provide privacy on network communications. Others have noted the vulnerability of workstations and their disk storage to physical attacks in the office workstation environment, and this has led to a variety of *secret sharing* algorithms for protecting data from isolated attacks [24, 42, 48]. Tools from the field of consensus protocols can also be applied [24]. These techniques, while powerful, still depend on some measure of physical security.

Cryptography allows us to slightly relax our assumptions about physical security; with cryptography we no longer need to assume that our network is physically shielded. However, we still need to make strong assumptions about the physical protection of hosts. We cannot entirely eliminate the need for physical security.

All security algorithms and protocols depend on physical security. Cryptographic systems depend on the secrecy of keys, and authorization and access control mechanisms crucially depend on the integrity of the access control database. The use of physical security to provide privacy and integrity is the foundation upon which security mechanisms are built. With the proliferation of workstations to the office and to open computation clusters, the physical security assumption is no longer valid. The recent advent of powerful mobile computers only exacerbates this problem, since the machines may easily be physically removed.

The gap between the reality of physically unprotected systems and this assumption of physical security must be closed. With traditional mainframe systems, the security firewall was between the users' terminals and the computer itself—the mainframe was the physically secure component in the system.¹ With loosely administered, physically accessible workstations, the security partition can no longer encompass all the machines. Indeed, with most commercially available workstations, the best that can be found is a simple lock in the front panel which can be easily picked or bypassed—there really is no physically secure component in these systems.

This paper discusses the use of physically secure processors to achieve new, powerful solutions to system security problems. (Physically secure coprocessors were first introduced in [6].) A secure coprocessor embodies a physically secure hardware module; it achieves this security by advanced packaging technology [62]. We focus on systems and protocols that can exploit the physical shielding to achieve novel solutions to challenging problems. There are many applications that need to use secure coprocessors; we discuss five of them here.

1. Consider the problem of protecting the integrity of publicly accessible workstations. For normal workstations or PCs, it is very easy to steal or modify data and programs on the hard disks. Operating system software could be modified to log keystrokes

to extract encryption keys that you may have used to protect data. There is neither privacy nor integrity when the attacker has physical access to the machine, even if we prevent the attacker from adding Trojan horses to hardware (e.g., a modified keyboard which records keystrokes or a network interface board which sends the contents of the system memory to the attacker).

2. The problem of providing tamper-proof audit trails and accounting logs is similar to that of workstation integrity, except that instead of protecting largely static data (operating system kernels and system programs), the goal is to make the generated logs unforgeable. For normal workstations or PCs, nothing prevents attackers from modifying system logs to erase evidence of intrusion. Similarly, secure system accounting is impossible because nothing protects the integrity of the accounting logs.
3. The problem of providing copy protection for proprietary programs is also insoluble on traditional hardware. Distributing software in encrypted form does not help, since the user's machine must have the software decrypted in its memory to run it. Because we cannot guarantee the integrity of the machine's operating system, we have no assurances as to the privacy of this in-memory copy of the software.
4. Another difficult problem is that of providing electronic currency without centralized control. Any electronic representation of currency is subject to duplication—data stored in computers can always be copied, regardless of how our software may choose to interpret them. When electronic currency no longer remains on trusted, centralized server machines, there is no way to guarantee against tampering.

Given that we cannot trust the system software on our publicly accessible computers, any electronic currency on our machines might be arbitrarily created, destroyed, or sent over a network to the attacker. Alternatively, an untrustworthy user can record the state of his computer prior to "spending" his electronic currency, after which he simply resets the state of his computer to the saved state. Without a way to securely manage currency, attackers may "print" money at will. Furthermore, the attacker may take advantage of a partitioned network in order to use the same electronic currency in transactions with machines in different partitions. Since no communication is possible between these machines, users (or computers acting as service-providers) have no way to check for duplicity.

5. A closely related problem arises when we want to provide "electronic contracts" that obligate secure coprocessors to perform certain actions or enforce certain restrictions. The notion of an electronic contract provides a mechanism for controlling the configuration of security constraints listed in the above items. Note that it does not suffice to merely cryptographically sign contracts; we must ensure that contracts are enforced on all machines that are parties in the contract.

All of these problems are vulnerable to physical attacks which result in a loss of privacy and integrity. Software protection systems crucially rely on the physical security of the underlying hardware and are completely useless when the physical security assumption is violated.

We can, however, close the assumption/reality gap in computer security. By adding physically secure coprocessors to computer systems, real, practical security systems can be built. Not only are secure coprocessors necessary and sufficient for security systems to be built; placing the security partition around a coprocessor is the natural model for providing security for workstations. Moreover, they are cost effective and can be made largely transparent to the end user.

The rest of this paper outlines the theory of secure coprocessors. First we discuss a model for physically secure coprocessors and describe a number of platforms that use secure coprocessor technology. Then we consider several important security problems that are solved by using secure coprocessors. We next present a hierarchy of traditional and new approaches to physical security, and demonstrate naturally induced security partitions within these systems. We give an approach which allows secure coprocessors to be integrated into existing operating systems; we continue with a machine-user authentication section, which tackles the problem of verifying the presence of a secure coprocessor to users. Finally, we discuss previous work. For the sake of readers who are not computer scientists, we include a glossary of technical terms at the end of the paper.

SECURE COPROCESSORS

What do we mean by the term *secure coprocessor*? A secure coprocessor is a hardware module containing (1) a CPU, (2) ROM, and (3) NVM (non-volatile memory). This hardware module is physically shielded from penetration, and the I/O interface to this module is the only means by which access to the internal state of the module can be achieved. (Examples of packaging technology are discussed later in this section.) Such a hardware module can store cryptographic keys without risk of release. More generally, the CPU can perform arbitrary computations (under control of the operating system) and thus the hardware module, when added to a computer, becomes a true coprocessor. Often, the secure coprocessor will contain special-purpose hardware in addition to the CPU and memory; for example, high speed encryption/decryption hardware may be used.

Secure coprocessors must be packaged so that physical attempts to gain access to the internal state of the coprocessor will result in resetting the state of the secure coprocessor (i.e., erasure of the NVM contents and CPU registers). An intruder might be able to break into a secure coprocessor and see how it is constructed; the intruder cannot, however, learn or change the internal state of the secure coprocessor except through normal I/O channels or by forcibly resetting the entire secure coprocessor. The guarantees about the privacy and integrity of non-volatile memory provide the foundations needed to build security systems.

Physical Assumptions for Security

All security systems rely on a nucleus of assumptions. For example, it is often assumed that it is infeasible to successfully cryptanalyze the encryption system used for security. Our basic assumption is that the coprocessor provides private and tamper-proof memory and processing. Just as attackers can exhaustively search cryptographic key spaces, it may be possible to falsify the physical security hypothesis by expending enormous resources (possibly feasible for very large corporations or government agencies), but we will assume the physical security of the system as an axiom. This is a physical work-factor argument, similar in spirit to intractability assumptions of cryptography. Our secure coprocessor model does not depend on the particular technology used to satisfy the work-factor assumption. Just as cryptographic schemes may be scaled to increase the resources required to penetrate a cryptographic system, current security packaging techniques may be scaled or different packaging techniques may be employed to increase the work-factor necessary to successfully bypass the physical security measures.

In the section on applications, we will see examples of how we can build secure subsystems running partially on a secure coprocessor by leveraging off the physical security of the coprocessor.

Limitations of Model

Even though confining all computation within secure coprocessors would ideally suit our security needs, in reality we cannot—and should not—convert all of our processors into secure coprocessors. There are two main reasons: the first is the inherent limitations of the physical security techniques in packaging circuits, and the second is the need to keep the system maintainable. Fortunately, as we shall see in the section below, the entire computer need not be physically shielded. It suffices to physically protect only a portion of the computer.

Current packaging technology limits us to approximately one printed circuit board in size to allow for heat dissipation. Future developments may eventually relax this and allow us to make more of the solid-state components of a multiprocessor workstation physically secure, perhaps an entire card cage; the security problems of external mass storage and networks, however, will in all likelihood remain a constant.

While it may be possible to securely package an entire multiprocessor in a physically secure manner, it is likely to be impractical and is unnecessary besides. If we can obtain similar functionalities by placing the security concerns within a single coprocessor, we can avoid the cost of making all the processors (in multiprocessors) secure.

Making a system easy to maintain requires a modular design. Once a hardware module is encapsulated in a physically secure package, disassembling the module to fix or replace some component will probably be impossible. Moreover, packaging considerations, as well as the extra hardware development time required, imply that the technology used within a secure coprocessor may lag slightly behind the technology used within the host system—perhaps by one generation. The right balance between physically shielded and unshielded components will depend on the class of applications for which the system is intended. For many applications, only a small portion of the system must be protected.

Potential Platforms

Several real instances of physically secure processing exist. This subsection describes some of these platforms, giving the types of attacks which these systems are prepared against, and the limitations placed on the system due to the approaches taken to protect against physical intrusion.

The μ ABYSS [62] and Citadel [64] systems provide physical security by employing board-level protection. The systems include an off-the-shelf microprocessor and some non-volatile (battery backed) memory, as well as special sensing circuitry which detects intrusion into a protective casing around the circuit board. The security circuitry erases the non-volatile memory before attackers can penetrate far enough to disable the sensors or to read the memory contents from the memory chips. The Citadel system expands on μ ABYSS, incorporating substantially greater processing power; the physical security mechanisms remain identical.

Physical security mechanisms must protect against many types of physical attacks. In the μ ABYSS and Citadel systems, it is assumed that in order for intruders to penetrate the system, they must be able to probe through a hole of one millimeter in diameter (probe pin voltages, destroy sensing circuitry, etc). To prevent direct intrusion, these systems incorporate sensors consisting of fine (40 gauge) nichrome wire, very low power sensing circuits, and a long life-time battery. The wires are loosely but densely wrapped in many layers about the circuit board and the entire assembly is then dipped in a potting material. The loose and dense wrapping makes the exact position of the wires in the epoxy unpredictable. The sensing electronics can detect open circuits or short circuits in the wires and erase the non-volatile memory if intrusion is attempted. The designs show that physical intrusion by mechanical means (e.g., drilling) cannot penetrate the epoxy without breaking one of these wires.

Another physical attack is the use of solvents to dissolve the potting material to expose the sensor wires. To block this kind of attack, the potting material is designed to be chemically "stronger" than the sensor wires. This implies that solvents will destroy at least one of the wires—and thus create an open-circuit condition—before the intruder can bypass the potting material and access the circuit board.

Yet another physical attack uses low temperatures. Semiconductor memories retain state at very low temperatures even without power, so an attacker could freeze the secure coprocessor to disable the battery and then extract the memory contents at leisure. The designers have blocked this attack by the addition of temperature sensors which trigger erasure of secrets before the low temperature reaches the dangerous level. (The system must have enough thermal mass to prevent quick freezing—by being dipped into liquid nitrogen or helium, for example—so this places some limitations on the minimum size of the system.)

The next step in sophistication is the high-powered laser attack. Here, the idea is that a high powered (ultraviolet) laser may be able to cut through the protective potting material and selectively cut a run on the circuit board or destroy the battery before the sensing circuitry has time to react. To protect against such an attack, alumina or silica is added to the epoxy potting material which causes it to absorb ultraviolet light. The

generated heat will cause mechanical stress, which will cause one or more of the sensing wires to break.

Instead of the board-level approach, physical security can be provided for smaller, chip-level packages. Clipper and Capstone, the NSA's proposed DES replacements [3, 56, 57] are special-purpose encryption chips. The integrated circuit chips are designed in such a way that key information (and perhaps other important encryption parameters—the encryption algorithm is supposed to be secret as well) are destroyed when attempts are made to open the integrated circuit chips' packaging. The types of attacks which this system can withstand are unknown.

Another approach to physically secure processing appears in *smart-cards* [30]. A smart-card is essentially a credit-card-size microcomputer which can be carried in a wallet. While the processor is limited by size constraints and thus is not as powerful as that found in board-level systems, no special sensing circuitry is necessary since physical security is maintained by the virtue of its portability. Users may carry their smart-cards with them at all times and can provide the necessary physical security. Authentication techniques for smart-cards have been widely studied [1, 30].

These platforms and their implementation parameters together provide the technology envelope within which secure coprocessor hardware will likely reside and this envelope will provide constraints on what class of algorithms is reasonable. As more computation power moves into mobile computers and smart-cards and better physical protection mechanisms are devised, this envelope will grow larger with time.

APPLICATIONS

Because secure coprocessors can *process* secrets as well as store them, they can do much more than just keep secrets confidential. We can use the ability to compute privately to provide many security related features, including (1) host integrity verification; (2) tamper-proof audit trails; (3) copy protection; (4) electronic currency; (5) and electronic contracts.² None of these are realistically possible on physically exposed machines.

Host Integrity Check

Trojan horse software dates back to the 1960s, if not earlier. Bogus login programs are the most common, though games and fake utilities are also widely used to set up back doors as well. Computer viruses exacerbate the problem of host integrity—the system may easily be inadvertently corrupted during normal use.

The host integrity problem can be partially ameliorated by guaranteeing that all programs have been inspected and approved by a trusted authority, but this is at best an incomplete solution. With computers getting smaller and workstations often physically accessible in public computer clusters, attackers can easily bypass any logical safeguards to modify the disks. How can you tell if even the operating system kernel is correct? The integrity of the computer needs to be verified. The integrity of the kernel image and system utilities stored on disk must be verified to be unaltered since the last system release.³

There are two main cases to examine. The first is that of stand-alone workstations that are not connected to any networks, and the second is

that of networked workstations with access to distributed services such as AFS [52] or Athena [4]. While publicly accessible stand-alone workstations have fewer avenues of attack, there are also fewer options for countering attacks. We will examine both cases concurrently in the following discussion.

Using a secure coprocessor to perform the necessary integrity checks solves the host integrity problem. Because of the privacy and integrity guarantees on secure coprocessor memory and processing, we can use a secure coprocessor to check the integrity of the host's state at boot-up and have confidence in the results. At boot time, the secure coprocessor is the first to gain control of the system and can decide whether to allow the host CPU to continue by first checking the disk-resident bootstrap program, operating system kernel, and all system utilities for evidence of tampering.

The cryptographic checksums of system images must be stored in the secure coprocessor's NVM and protected both against modification and (depending on the cryptographic checksum algorithm chosen) against exposure. Of course, tables of cryptographic checksums can be paged out to host memory or disk after first checksumming and encrypting them within the secure coprocessor; this can be handled as an extension to normal virtual memory paging. We have more to say on this subject in the section on system architecture. Since the integrity of the cryptographic checksums is guaranteed by the secure coprocessor, we can detect any modifications to the system objects and protect ourselves against attacks on the external storage.

One alternative model is to eliminate external storage for networked workstations—to use trusted file servers and access a remote, distributed file system for all external storage. Any paging needed to implement virtual memory would go across the network to a trusted server with disk storage.

What are the difficulties with this trusted file server model? First, note that non-publicly readable files and virtual memory pages must be encrypted before being transferred over the network and so some hardware support is probably required anyway for performance reasons. A more serious problem is that the workstations must be able to authenticate the identity of the trusted file servers (the host-to-host authentication problem). Since workstations cannot keep secrets, we cannot use shared secrets to encrypt and authenticate data between the workstation and the file servers. The best that we can do is to have the file servers use public key cryptography to cryptographically sign the kernel image when we boot over the network, but we must be able to store the public keys of the trusted file servers somewhere. With exposed workstations, there's no safe place to store them. Attackers can always modify the public keys (and network addresses) of the file servers so that the workstation would contact a false server. Obtaining public keys from some external key server only pushes the problem one level deeper—the workstation would need to authenticate the identity of the key server, and attackers need only to modify the stored public key of the key server.

If we page virtual memory over the network (which we assume is not secure), the problem only becomes worse. Nothing guarantees the privacy or integrity of the virtual memory as it is transferred over the

network. If the data is transferred in plaintext, an attacker can simply record network packets to break privacy and modify/substitute network traffic to destroy integrity. Without the ability to keep secrets, encryption is useless for protecting their memory—attackers can obtain the encryption keys by physical means and destroy privacy and integrity as before.

A second alternative model, which is a partial solution to the host integrity problem, is to use a secure-boot floppy containing system integrity verification code to bring machines up. Let's look at the assumptions involved here. First, note that we are assuming that the host hardware has not been compromised. If the host hardware has been compromised, the "secure" boot floppy can easily be ignored or even modified when used, whereas secure coprocessors cannot. The model of using a secure removable medium for booting assumes that untrusted users get a (new) copy of a master boot floppy from the trusted operators each time a machine is rebooted from an unknown state. Users must not have access to the master boot floppy since it must not be altered.

What problems are there? Boot floppies cannot keep secrets—encryption does not help, since the workstation must be able to decrypt them and workstations cannot keep secrets (encryption keys) either. The only way to assure integrity without completely reloading the system software is to check it by checking some kind of cryptographic checksum on the system images.

There are a variety of cryptographic checksum functions available, and all obviously require that the integrity of the checksums for the "correct" data be maintained: when we check the system images on the disk of a suspect workstation, we must recompute new checksums and compare them with the original ones. This is essentially the same procedure used by secure coprocessors, except that instead of providing integrity within a piece of secure hardware we use trusted operators. The problem then becomes that of making sure that operators and users follow the proper security procedures. Requiring that users obtain a fresh copy of the integrity check software and data each time they need to reboot a machine is cumbersome. Furthermore, requiring a centralized database of all the software that requires integrity checks for all versions of that software on the various machines will be a management nightmare. Any centralized database is necessarily a central point of attack. Destroying this database will deny service to anybody who wishes to securely bootstrap their machine.

Both secure coprocessors and secure boot floppies can be fooled by a sufficiently faithful emulation of the system which simulates a normal disk during integrity checks, but secure coprocessors allow us to employ more powerful integrity check techniques to provide better security. Furthermore, careless use (i.e., reuse) of boot floppies becomes another channel of attack—boot floppies can easily be made into viral vectors.

Along with integrity, secure coprocessors offer privacy; this property allows the use of a wider class of cryptographic checksum functions. There are many cryptographic checksum functions that might be used, including Rivest's MD5 [44], Merkle's Snefru [31], Jueneman's Message Authentication Code (MAC) code [26], IBM's Manipulation Detection Code (MDC) [25], chained DES [60], and Karp and Rabin's family of fingerprint functions [28]. All of these require integrity; the last three

require privacy of keys. The strength of these rely on the difficulty of finding *collisions*—two different inputs with the same checksum. The intractability arguments for the first four of these are based on conjectured numbers of bit operations required to find collisions, and so are weak with respect to theoretical foundations. MDC, chained DES, and the fingerprint functions also keep the identity of the particular checksum function used secret—with MDC and DES it corresponds to keeping encryption keys (which select particular encryption functions) secret, and with fingerprint functions it corresponds to keeping an irreducible polynomial (which defines the fingerprint function) secret. DES is less well understood than the Karp-Rabin functions.

The secrecy requirement of MDC, chained DES, and the Karp-Rabin functions is a stronger assumption which *can* be provided by a secure coprocessor and it allows us to use cryptographic functions with better theoretical underpinnings, thus improving the bounds on the security provided. Secrecy, however, cannot be provided by a boot floppy. The Karp-Rabin fingerprint functions are superior to chained DES in that they are much faster and much easier to implement (thus the implementation is less likely to contain bugs), and there are no proven strong lower bounds on the difficulty of breaking DES.

Secure coprocessors also greatly simplify the problem of system upgrades. This is especially important when there are large numbers of machines on a network: systems can be securely upgraded remotely through the network. Furthermore, it's easy to keep the system images encrypted while they are being transferred over the network and while they are resident on secondary storage. This provides us with the ability to keep proprietary code protected against most attacks. As noted below in the section on copy protection, we can run (portions of) the proprietary software only within the secure coprocessor, allowing vendors to have execute-only semantics—proprietary software need never appear in plaintext outside of a secure coprocessor.

The later section on operational requirements discusses the details of host integrity check as it relates to secure coprocessor architectural requirements, and the section on key management discusses how system upgrades would be handled by a secure coprocessor. Also relevant is the problem of how the user can know if a secure coprocessor is running properly in a system; our section on machine-user authentication discusses this.

Audit Trails

In order to properly perform system accounting and to provide data for tracing and detecting intruders on the host system, audit trails must be kept in a secure manner. First, note that the integrity of the auditing and accounting logs cannot be completely guaranteed (since the entire physically accessible machine, including the secure coprocessor, may be destroyed). The logs, however, can be made tamper evident. This is quite important for detecting intrusions—forging system logs to eliminate evidence of penetration is one of the first things that a system cracker will attempt to do. The privacy and integrity of the system accounting logs and audit trails can be guaranteed (unless the secure coprocessor is removed or destroyed) simply by holding them inside the secure

coprocessor. It is awkward to have to keep everything inside the secure coprocessor since accounting and audit logs can grow very large and resources within the secure coprocessor are likely to be tight. Fortunately, it is also unnecessary.

To provide secure logging, we use the secure coprocessor to seal the data against tampering with one of the cryptographic checksum functions discussed above; we then write the logging information out to the file system. The sealing operation must be performed within the secure coprocessor, since all keys used in this operation must be kept secret. By later verifying these cryptographic checksums we make tampering of log data evident, since the probability that an attacker can forge logging data to match the old data's checksums is astronomically low. This technique reduces the secure coprocessor storage requirement from large logs to the memory sufficient to store the cryptographic keys and checksums, typically several words per page of logged memory. If the space requirement for the keys and checksums is still too large, they can be similarly written out to secondary storage after being encrypted and checksummed by master keys.

Additional cryptographic techniques can be used for the cryptographic sealing, depending on the system requirements. Cryptographic checksums can provide the basic tamper detection and are sufficient if we are only concerned about the integrity of the logs. If the accounting and auditing logs may contain sensitive information, privacy can be provided by using encryption. If redundancy is required, techniques such as secure quorum consensus [24] and secret sharing [48] may be used to distribute the data over the network to several machines without greatly expanding the space requirements.

Copy Protection

A common way of charging for software is licensing the software on a per-CPU, per-site, or per-use basis. Software licenses usually prohibit making copies for use on unlicensed machines. Without a secure coprocessor, this injunction against copying is unenforceable. If the user can execute the code on any physically accessible workstation, the user can also read that code. Even if we assume that attackers cannot read the workstation memory while it is running, we are implicitly assuming that the workstation was booted correctly—verifying this property, as discussed above, requires the use of a secure coprocessor.

When secure coprocessors are added to a system, however, we can quite easily protect executables from being copied and illegally utilized by attackers. The proprietary code to be protected—or at least some critical portion of it—can be distributed and stored in encrypted form, so copying it without obtaining the code decryption key is futile.⁴ Public key cryptography may be used to encrypt the entire software package or a key for use with a private key system such as DES. When a user pays for the use of a program, a digitally signed certificate of the public key used by his secure coprocessor is sent to the software vendor. This certificate is signed by a key management center verifying that a given public key corresponds to a secure coprocessor, and is *prima facie* evidence that the public key is valid. The corresponding private key is

stored only within the NVM of the secure coprocessor; thus, only the secure coprocessor will have full access to the proprietary software. What if the code size is larger than the memory capacity of the secure coprocessor? We have two alternatives: we can use *crypto-paging* or we can split the code into protected and unprotected segments.

We discuss crypto-paging in greater detail in the section on system architecture below, but the basic idea is to dynamically load the relevant sections of memory from the disk as needed. Since good encryption chips are fast, we can decrypt on the fly with little performance penalty. Similarly, when we run out of memory space on the coprocessor, we encrypt the data as we flush it out onto secondary storage.

Splitting the code is an alternative to this approach. We can divide the code into a security-critical section and an unprotected section. The security-critical section is encrypted and runs only on the secure coprocessor. The unprotected section runs in parallel on the main host processor. An adversary can copy the unprotected section, but if the division is done well, he or she will not be able to run the code without the secure portion.

A more primitive version of the copy protection application for secure coprocessors originally appeared in [63].

Electronic Currency

With the ability to keep licensed proprietary software encrypted and allow execute access only, a natural application would be to allow for charging on a pay-per-use basis. In addition to controlling access to the software according to the terms of software licenses, some mechanism must be available to perform cost accounting, whether it is just keeping track of the number of times a program has run or keeping track of dollars in the users' account. More generally, this accounting software provides an *electronic currency* abstraction. Correctly implementing electronic currency requires that account data be protected against tampering—if we cannot guarantee integrity, attackers will be able to create electronic money at will. Privacy, while perhaps less important here, is a property that users expect for their bank balance and wallet contents; similarly, electronic money account balances should also be private.

There are several models that can be adopted for handling electronic funds. The first is the cash analogy. Electronic funds can have similar properties to cash: (1) exchanges of cash can be effectively anonymous; (2) cash cannot be created or destroyed; and (3) cash exchanges require no central authority. (Note that these properties are actually stronger than that provided by currency—serial numbers can be recorded to trace transactions, and national treasuries regularly print and destroy money.)

The second model is the credit cards/checks analogy. Electronic funds are not transferred directly; rather, promises of payment—perhaps cryptographically signed to prove authenticity—are transferred instead. A straightforward implementation of the credit card model fails to exhibit any of the three properties above. By applying cryptographic techniques, anonymity can be achieved [10], but the latter two requirements remain insurmountable. Checks must be signed and validated at central authorities (banks), and checks/credit payments en route “create”

temporary money. Furthermore, the potential for reuse of cryptographic signed checks requires that the payee must be able to validate the check with the central authority prior to committing to a transaction.

The third model is analogous to a rendezvous at the bank. This model uses a centralized authority to authenticate all transactions and so is even worse for large distributed applications. The bank is the sole arbiter of the account balance and can easily implement the access controls needed to ensure privacy and integrity of the data. This is essentially the model used in Electronic Funds Transfer (EFT) services provided by many banks—there are no access restrictions on deposits into accounts, so only the depositor for the source account needs to be authenticated.

Let us examine these models one by one. What sort of properties must electronic cash have? We must be able to easily transfer money from one account to another. Electronic money must not be created or destroyed by any but a very few trusted users who regulate the electronic version of the Treasury.

With electronic currency, integrity of the accounts data is crucial. We can establish a secure communication channel between two secure coprocessors by using a key exchange cryptographic protocol and thus maintain privacy when transferring funds. To ensure that electronic money is conserved (neither created nor destroyed), the transfer of funds should be failure atomic, i.e., the transaction must terminate in such a way as to either fail completely or fully succeed—transfer transactions cannot terminate with the source balance decremented without having incremented the destination balance or vice versa. By running a transaction protocol such as two-phase commit [8, 12, 65] on top of the secure channel, the secure coprocessors can transfer electronic funds from one account to another in a safe manner, providing privacy as well as ensuring that money is conserved throughout. With most transaction protocols, some “stable storage” for transaction logging is needed to enable the system to be restored to the state prior to the transaction when a transaction aborts. On large transaction systems this typically has meant mirrored disks with uninterruptible power supplies. With the simple transfer transactions here, the per-transaction log typically is not that large, and the log can be truncated once transactions commit. Because each secure coprocessor needs to handle only a handful of users, large amounts of stable storage should not be needed—because we have non-volatile memory in secure coprocessors, we only need to reserve some of this memory for logging. The log, the accounts data, and the controlling code are all protected from modification by the secure coprocessor, so account data are safe from all but bugs and catastrophic failures. Of course, the system should be designed so that users should have little or no incentive to destroy secure coprocessors that they can access—which should be natural when their own balances are stored on secure coprocessors, much like cash in wallets.

Note that this type of decentralized electronic currency is *not* appropriate for smart cards unless they can be made physically secure from attacks by their owners. Smart cards are only quasi-physically-secure in that their privacy guarantees stem solely from their portability. Secrets may be stored within smart cards because their users can provide the

physical security necessary. Malicious users, however, can easily violate smart card integrity and insert false data.

If there is insufficient memory within the secure coprocessor to hold the account data for all its users, the code and the accounts database may be cryptographically paged to host memory or disk by first obtaining a cryptographic checksum. For the accounts data, encryption may also be employed since privacy is typically desired as well. The same considerations as those for checksums of system images apply here as well.

This electronic currency transfer is analogous to the transfer of rights (not to be confused with the copying of rights) in a capability-based protection system. Using the electronic money—e.g., expended when running a pay-per-use program—is analogous to the revocation of a capability.

What about the other models for handling electronic funds? With the credit card/check analogy, the authenticity of the promise of payment must be established. When the computer cannot keep secrets for users, there can be no authentication because nothing uniquely identifies users. Even when we assume that users can enter their passwords into a workstation without having the secrecy of their password compromised, we are still faced with the problem of providing privacy and integrity guarantees for network communication. We have similar problems as in host-to-host authentication in that cryptographic keys need to be exchanged somehow. If communication is in plaintext, attackers may simply record a transferral of a promise of payment and replay it to temporarily create cash. While security systems such as Kerberos [53], if properly implemented, can help to authenticate entities and create session keys, they revert to the use of a centralized server and have similar problems to the bank rendezvous model.

With the bank rendezvous model, a "bank" server supervises the transfer of funds. While it is easy to enforce the access controls on account data, this suffers from problems with non-scalability, loss of anonymity, and easy denial of service from excessive centralization.

Because every transaction must contact the bank server, access to the bank service will be a performance bottleneck. The system does not scale well to a large user base—when the bank system must move from running on a single computer to several machines, distributed transaction systems techniques must be brought to bear in any case, so this model has no real advantage over the use of secure coprocessors in ease of implementation. Furthermore, if the bank host becomes inaccessible, either maliciously or as a result of normal hardware failures, no agent can make use of any bank transfers. This model does not exhibit graceful degradation with system failures.

The model of electronic currency managed on a secure coprocessor not only can provide the properties of (1) anonymity, (2) conservation, and (3) decentralization, but it also degrades gracefully when secure coprocessors fail. Note that secure coprocessor data may be mirrored on disk and backed up after being properly encrypted, and so even the immediately affected users of a failed secure coprocessor should be able to recover their balance. The security administrators who initialized the secure coprocessor software will presumably have access to the decryption keys for this purpose. Careful procedural security must be

required here, both for the protection of the decryption key and for auditing for double spending, since dishonest users might attempt to back up their secure coprocessor data, spend electronic money, and then intentionally destroy their coprocessor in the hopes of using their electronic currency twice. The amount of redundancy and the frequency of backups depends on the reliability guarantees desired; in reliable systems secure coprocessors may continually run self-checks when idle and warn of impending failures.

Contract Model

Our electronic contract model is built on the following two secure coprocessor-provided primitive objects: (1) unforgeable tokens and (2) computer-enforced contracts.

Tokens are protected objects that are conserved by the secure coprocessors; they are freely transferable, but they can be created and destroyed only by the agent that issued them. Tokens are useful as electronic currency and to represent the execute-only right to a piece of software (much as in capability systems). In the case of rights such as execute-only rights, the token provides access to cryptographic keys that may be used (only) within the secure coprocessors to run code.

Contracts are another class of protected objects. They are created when two parties agree on a contract draft. Contracts contain binding clauses specifying actions that each of the parties must perform—or, in reality, actions that the secure coprocessors will enforce—and “method” clauses that may be invoked by certain parties (not necessarily restricted to just the parties who agreed on the contract). Time-based clauses and other event-based clauses may also exist. Contractual obligations may force the transfer of tokens between parties.

Contract drafts are typically instantiated from a contract template. We may think of a contract template as a standardized contract with blanks which are filled in by the two parties involved, though certainly “custom” contracts are possible. Contract negotiation consists of an offerer sending a contract template along with the bindings (values with which to fill in blanks) to the offeree. The offeree either accepts or rejects the contract. If it is accepted, a contract instance is created whereby the contract bindings are permanent, and any immediate clauses are executed. If the draft is rejected, the offeree may take the contract template and re-instantiate a new draft with different bindings to create a counter-offer, whereupon the roles of offerer and offeree are reversed.

From the time that a contract is accepted until it terminates, the contract is an active object running in one or more secure coprocessors. Methods may be invoked by users or triggered by external events (messages from the host, timer expiration). The method clauses of a contract are access controlled: they may be optionally invoked by only one party involved in the contract—or even by a third party who is under no contractual obligations. Contractual clauses may require one of the parties to accept further contracts of contractual obligations. Contractual clauses may require one of the parties to accept further contracts of certain types. One example of this is a requirement for some action to be performed prior to a certain time. Another is a contract between a distributor and a software house, where the software house requires the

distributor to accept sales contracts from users for upgrading a piece of software.

SECURITY PARTITIONS IN NETWORKED HOSTS

Network hosts, regardless of whether they use cryptography, have a de facto security partitioning that arises because different system components have different vulnerabilities to various attacks. Some of these vulnerabilities diminish when cryptography is used; similarly, the use of a secure coprocessor can be thought of as adding another layer with fewer vulnerabilities to the partitioning. By bootstrapping our system using a secure coprocessor and thus ensuring that the correct operating system is running, we can provide privacy and integrity guarantees on memory that were not possible before. In particular, public workstations can use secure coprocessors and cryptography to guarantee the privacy of disk storage and provide integrity checks. Let us see what kind of privacy/integrity guarantees are already available in the system and what new ones we can provide.

Subsystem	Vulnerabilities	
	Integrity	Privacy
Secure Coprocessor	None	None
Host RAM	On-line Physical Access	On-line Physical Access
Secondary Store	Off-line Physical Access	Off-line Physical Access
Network (communication)	On-line Remote Access	On-line Remote Access

Table 1: Subsystem Vulnerabilities Without Cryptographic Techniques

Table 1 shows the vulnerabilities of various types of memory when no cryptographic techniques are used. That memory within a secure coprocessor is protected against physical access is one of our axioms, and correctly using that to provide privacy and integrity at the logical level is a matter of using the appropriate software protection mechanisms. With the proper protection mechanisms within a secure coprocessor, data stored within a secure coprocessor can be neither read nor tampered with. Since we assume that we have a working secure coprocessor, we will also assume that the operating system was booted correctly and thus host RAM is protected against unauthorized logical access.⁵ It is not, however, well protected against physical access—it is a simple matter to connect logic analyzers to the memory bus to listen passively to memory traffic. Furthermore, replacing the memory subsystem with multi-ported memory in order to allow remote unauthorized memory accesses is also a conceivable attack. While the effort required to do this in a way that is invisible to users may make it impractical, this line of attack can certainly not be entirely ruled out. Secondary storage may be more easily attacked than RAM since the data can be modified off-line; to do this, however, an attacker must gain physical access to the

disk. Network communication is completely vulnerable to on-line eavesdropping and off-line analysis, as well as on-line message tampering. Since networks are inherently used for remote communication, it is clear that these may be remote attacks.

What protection guarantees can we provide when we use encryption? By using encryption when appropriate, we can guarantee privacy. Integrity of the data, however, is not guaranteed. The same vulnerabilities which allowed data modifications still exist as before; tampering, however, can be detected by using cryptographic checksums as long as the checksum values are stored in tamper-proof memory. Note also that the privacy that can be provided is relative to the data usage. If data in host RAM is to be processed by the host CPU, encrypting it within the secure coprocessor is useless—the data must remain vulnerable to on-line physical attacks on the host since it must appear in plaintext form to the host CPU. If the host RAM data is simply serving as backing store for secure coprocessor data pages, however, encryption is appropriate. Similarly, encrypting the secondary store via the host CPU protects that data against off-line privacy loss but not on-line attacks, whereas encrypting that data within the secure coprocessor protects that data against on-line privacy attacks as well, as long as that data need not ever appear in plaintext form in the host memory.

Subsystem	Vulnerabilities	
	Integrity	Privacy
Secure Coprocessor	None	None
Host RAM	On-line Physical Access	Host Processor Data
Secondary Store	Off-line Physical Access (detectable)	None
Network (communication)	On-line Remote Access (detectable)	None

Table 2: Subsystem Vulnerabilities With Cryptographic Techniques

For example, if we wish to send and read secure electronic mail, the encryption and decryption can be performed in the host processor since the data must reside within both hosts for the sender to compose it and for the receiver to read it. The exchange of the encryption key used for the message, however, requires secure coprocessor computation: the encryption for the key exchange needs to use secrets that must remain within the secure coprocessor, regardless of whether the key exchange uses a shared secret key or a public key scheme.⁶

SYSTEM ARCHITECTURE

This section discusses one possible architecture for a secure coprocessor software system. We will start off with a discussion of the constraints placed upon a secure coprocessor by the operational requirements of a security system—during system initialization and during normal, steady-state operation. We will next refine these constraints, examining various security functions and what their assumptions imply about trade-offs in a

secure coprocessor. Following this, we will discuss the structure of the software in a secure coprocessor, ranging from a secure coprocessor kernel and its interactions with the host system to user-level applications.

Operational Requirements

We will start by examining how a secure coprocessor must interact with the host hardware and software during the bootstrap process and then proceed with the kinds of system services that a secure coprocessor should provide to the host operating system and user software. The first issue to consider is how to fit a secure coprocessor into a system. This will guide us in the specification of the secure coprocessor software.

To be sure that a system is bootstrapped securely, secure hardware must be involved in the bootstrap process. Depending on the host hardware—whether a secure coprocessor could halt the boot process if it detects an anomaly—we may need to assume that the bootstrap ROM is secure. To ensure this, the system's address space either could be configured such that the boot vector and the boot code are provided by a secure coprocessor directly or we may simply assume that the boot ROM itself is a piece of secure hardware. Regardless, a secure coprocessor verifies the system software (operating system kernel, system related user-level software) by checking the software's signature against known values. We need to convince ourselves that the version of the software present in external, non-secure, non-volatile store (disk) is the same as that installed by a trusted party. Note that this interaction has the same problems faced by two hosts communicating via a non-secure network: if an attacker can completely emulate the interaction that the secure coprocessor would have had with a normal host system, it is impossible for the secure coprocessor to detect this. With network communication, we can assume that both hosts can keep secrets and build protocols based upon those secrets. With secure coprocessor/host interaction, we can make very few assumptions about the host—the best that we can do is to assume that the cost of completely emulating the host at boot time is prohibitive.

At boot time, the primary duty of a secure coprocessor is to make sure that the system boots up securely; after booting, a secure coprocessor's role is to aid the host operating system by providing security functions not otherwise available. A secure coprocessor does not enforce the system's security policy—that is the job of the host operating system; since we know from the secure boot procedure that the correct operating system is running, we may rely on the host to enforce policy. When the system is up and running, a secure coprocessor provides the following security services to the host operating system: the host may use the secure coprocessor to verify the integrity of any data in the same manner that the secure coprocessor checks the integrity of system software; it may use the secure coprocessor to encrypt data to boost the natural security of storage media (see section above on security partitions); and it may use the secure coprocessor to establish secure, encrypted connections with remote hosts (key exchange, authentication, private key encryption, etc).⁷

Secure Coprocessor Architecture

The bootstrapping procedure described above made assumptions about the capability of a secure coprocessor. Let us refine what requirements we have on the secure coprocessor software and hardware.

When a secure coprocessor verifies that the system software is the correct version, we are assuming that a secure coprocessor has secure, tamper-proof memory which remembers a description of the correct version of the system software. If we assume that proposed functions such as MD5 [44], multi-round Snefru [31], or IBM's MDC [25] are one-way hash functions, then the only requirement is that the memory is protected from writing by unauthorized individuals. Otherwise, we must use cryptographic checksums such as Karp and Rabin's technique of *fingerprinting*, which uses a family of hash functions with good error-detection capabilities. This technique requires that the memory be protected against read access as well, since both the hash value and the index selecting the particular hash function must be secret. In a similar manner, cryptographic operations such as authentication, key exchange, and secret key encryption all require that secrets be kept. Thus a secure coprocessor must have memory that is inaccessible by everybody except the secure coprocessor—enough private NVM to store the secrets, plus possibly volatile private memory for intermediate calculations in running the protocols.

There are a number of architectural tradeoffs for a secure coprocessor, the crucial dimensions being processor speed and memory size. They together determine the class of cryptographic algorithms that are practical.

The speed of the secure coprocessor may be traded off for memory in the implementation of the cryptographic algorithms. We observed in [54] that Karp-Rabin fingerprinting may be sped up by about 25% with a 256-fold table-size increase. Intermediate size tables may be used to yield intermediate speedups at a slightly higher increase in code size. Similar tradeoffs can be found for software implementations of the DES.

The amount of real memory required may be traded off for speed by employing cryptographic techniques: we need only enough private memory for an encryption key and a data cache, plus enough memory to perform the encryption if no encryption hardware is present. Depending on the throughput requirements, hardware assist for encryption may be included—where software is used to implement encryption, private memory must be provided for intermediate calculations. A secure coprocessor can securely page its private memory to either the host's physical memory (and perhaps eventually to an external disk) by first encrypting it to ensure privacy. Cryptographic checksums can provide error detection, and any error correcting encoding should be done *after* the encryption. This cryptographic paging is analogous to paging of physical pages to virtual memory on disk, except for different cost coefficients, and well-known analysis techniques can be used to tune such a system. The variance in costs will likely lead to new tradeoffs: cryptographic checksums are easier to calculate than encryption (and therefore faster modulo hardware support), so providing integrity alone is less expensive than providing privacy as well. On the other hand, if the

computation can reside entirely on a secure coprocessor, both privacy and integrity can be provided for free.

Secure Coprocessor Software

With partitioned applications that must have parts loaded into a secure coprocessor to run and perhaps paging of secure coprocessor tasks, a small, simple security kernel is needed for the secure coprocessor. What makes this kernel different from other security kernels is the partitioned system structure.

Like normal workstation (host) kernels, the secure coprocessor kernel must provide separate address space if vendor and user code is to be loaded into the secure coprocessor—even if we implicitly trust vendor and user code, providing separate address spaces helps to isolate the effects of programming errors. Unlike the host's kernel, many services are not required: terminal, network, disk, and other device drivers need not be part of the secure coprocessor. Indeed, since both the network and disk drives are susceptible to tampering, requiring their drivers to reside in the secure coprocessor's kernel is overkill—network and file-system services from secure coprocessor tasks can simply be forwarded to the host kernel for processing. Normal operating system services such as printer service, electronic mail, etc. are entirely inappropriate in a secure coprocessor—these system daemons can be eliminated entirely.

The only services that are crucial to the operation of the secure coprocessor are (1) secure coprocessor resource management; (2) communications; (3) key management; and (4) encryption services. Within resource management we include task allocation and scheduling, virtual memory allocation and paging, and allocation of communication ports. Under communications we include both communication among secure coprocessor tasks and communication to host tasks; it is by communicating with host system tasks that proxy services are obtained. Under key management we include the management of secrets for authentication protocols, cryptographic keys for protecting data as well as execute-only software, and system fingerprints for verifying the integrity of system software. With the limited number of services needed, we can easily envision using a microkernel such as Mach 3.0 [22]: we need to add a communications server and include a key management service to manage non-volatile key memory. The kernel must be small for us to trust it; we have more confidence that it can be debugged and verified.

Key Management

A core portion of the secure coprocessor software is code to manage keys. Authentication, key management, fingerprints, and encryption crucially protect the integrity of the secure coprocessor software and the secrecy of private data, including the secure coprocessor kernel itself. A permanent part of a bootstrap loader, in ROM or in NVM, controls the bootstrap process of the secure coprocessor itself. Like bootstrapping the host processor, this loader verifies the secure coprocessor kernel before transferring control to it.

The system fingerprints needed for checking system integrity must reside entirely in NVM or be protected by encryption while being stored on an external storage device—the key for which must reside solely in

the secure NVM. If the latter approach is chosen, new keys must be selected⁸ to prevent replay attacks where old, potentially buggy secure coprocessor software is reintroduced into the system. Depending on the cryptographic assumptions made in the algorithm, the storage of the fingerprint information may require just integrity or both integrity and secrecy. For the cases of MD4, MDC, and Snefru, integrity of the integrity check information is sufficient; for the case of the Karp-Rabin fingerprint, both integrity and secrecy are required.

Other protected data held within the secure coprocessor's NVM include administrative authentication information needed to update the secure coprocessor software. We assume that a security administrator is authorized to upgrade secure coprocessor software, and that only the administrator may authenticate his identity properly to the secure coprocessor. The authentication data for this operation can be updated along with the rest of the secure coprocessor system software; in either case, the upgrade must appear transactional, that is, it must have the properties of *permanence*, where results of completed transactions are never lost; *serializability*, where there is a sequential, non-overlapping view of the transactions; and *failure atomicity*, where transactions either complete or fail such that any partial results are undone. The non-volatility of the memory gives us permanence automatically, if we assume that only catastrophic failures (or intentional sabotage) can destroy the NVM; serializability, while important for multi-threaded applications, can be easily enforced if we permit only a single upgrade operation to be in progress at a time (this is an infrequent operation and does not require parallelism); and the failure atomicity guarantee can be provided easily as long as the non-volatile memory subsystem provides an atomic store operation. Update transactions need not be distributed nor nested; this simplifies the implementation immensely.

MACHINE-USER AUTHENTICATION

With secure coprocessors, we can perform all the necessary security functions to verify the integrity of the host system. The secure coprocessor may believe that the host system is clean, but how is the user to be convinced of this? After all, the secure coprocessor within the computer may have been replaced with a Trojan horse unit.

Smart-Cards

One solution to this is the use of smart-cards. Users can use advanced smart-cards to run an authentication procedure to verify the secure coprocessor's identity. Since secure coprocessors' identity-proofs can be based on a zero-knowledge protocol, no secret information needs to be stored in smart-cards unless smart-cards are to also aid users in authenticating themselves to systems, in which case the only secrets would be those belonging to the users. By the virtue of their portability, users can carry smart-cards at all times and thus provide the physical security needed.

Remote Services

Another way to verify that a secure coprocessor is present is to ask a third-party entity—such as a physically sealed third-party computer—to

check for the user. Often, this service can also be provided by normal network-servers machine such as file-servers. The remote services must be difficult to emulate by attackers. Users may rely on noticing the absence of these services to detect that something is amiss with the secure coprocessor. This necessarily implies that these remote services must be available *before* the users authenticate to the system.

Unlike authentication protocols reliant on accessing central authentication servers, this authentication happens once, at boot time. The identity being proven is that of the secure coprocessor—users may be confident that the workstation contains an authentic secure coprocessor if access to *any* normal remote service can be obtained. This is because in order to successfully authenticate to obtain the service, attackers must either break the authentication protocol, break the physical security in the secure coprocessor, or bypass the physical security around the remote server. As long as the remote service is sufficiently complex, attackers will not be able to emulate it.

RELATIONSHIP WITH PREVIOUS WORK

Partitioning security is not new. The method of embodying physical security in a secure coprocessor, however, *is* new, and it has been made possible only recently due to advances in packaging technology [62]. Certainly, the need for physical security is widely described in standard textbooks. For example, one book states that “physical security controls (locked rooms, guards, and the like) are an integral part of the security solution for a central computing facility.”[18]

We can trace several analogs to this approach of partitioning security in previous work. The logical partitioning of security in the literature [58, 61] of dividing the system into a “Trusted Computing Base” (TCB) and applications in some sense heralds this idea—the security partition was firmly drawn between the user and the machine; it not only included the logical security of the operating system part of the TCB, but also the physical security of the TCB hardware installation (machine rooms, etc).

Systems such as Kerberos [53] move that security partition for distributed systems toward including just one trusted server behind locked doors. This approach, however, still has serious security problems: client machines are often physically exposed and users are provided with no real assurances of their logical integrity, and the centralized server approach offers attackers a central point of attack—the system catastrophically fails when the central server is compromised [5]. Certainly, it does not offer much in terms of providing fault tolerance with distributed computing.

More recently, the partitioning in Strongbox [54] more clearly points the way toward minimizing the number of assumptions about trusted components in a secure system and clearly defining the security partition boundaries and security assumptions. In that system, the base security system was divided into trusted servers which, assuming protected address spaces, allowed security to be bootstrapped to application servers and clients. Unfortunately, while the system has better degradation properties, it could deliver system integrity assurances only by assuming trusted-operator-assisted bootstrapping. Table 3 shows the various types of systems and their basic assumptions as well as typical cryptographic assumptions.

System Type	Basic Assumptions	Cryptographic Assumptions
Conventional Non-distributed e.g., Unix password	Physical Security of Central Mainframe	DES cannot be inverted
Conventional Distributed e.g., Kerberos	Physical Security of Authentication Server	DES cannot be inverted
Self-Securing Distributed e.g., Strongbox	Physical Security of a Quorum of White Pages Servers	DES cannot be inverted; factoring is hard
Secure Coprocessors e.g., Dyad	Physics (Tampering destroys cryptographic data)	DES cannot be inverted; factoring is hard

Table 3: Basic Assumptions of Security Systems

The secure coprocessor approach minimizes the basic assumptions and can address all of the problems with the approaches cited above. By implementing cryptographic protocols within a secure coprocessor, we can be assured that they will execute correctly and that the secrets required by the various protocols are indeed kept secret. By using the secure coprocessor to verify the integrity of the rest of the system, we can give users greater assurance that the system has not been compromised and that the system has securely bootstrapped.

In addition to the work mentioned above, there are many other relevant works on security related issues: [3, 56, 57, 63] discuss issues in the design and implementation of physically secure system components. Research on cryptosystems and cryptographic protocols which are important tools for secure network communication can be found in [2, 5, 7, 11, 15, 16, 17, 19, 20, 21, 24, 29, 34, 35, 37, 42, 45, 47, 48, 49, 53]. More general information on some of the number theoretic tools behind many of these protocols may be found in [33, 36, 40, 51]. The tools for checking data integrity are described in [27, 28, 38, 41].

Research on protection systems and general distributed system security may be found in [39, 43, 46].

[9] provides a logic for analyzing authentication protocols, and [23] extends the formalism.

General security/cryptography information can be found in [14], the new proposed federal criteria for computer security [61], and in older standards such as the "Orange book" [58] and the "Red book" [59]. General information on cryptography can be found in [13, 32].

GLOSSARY OF TERMS

At the suggestion of the editors of this anthology, we include a small glossary of technical terms which may be unfamiliar to readers who are not computer scientists. Readers who are interested in operating systems may wish to read [50]; those who are interested in cryptography may wish to read [13, 32].

authentication The process by which identity (or other credentials) of a user or computer are verified. *Authentication protocols* are series of stylized exchanges which prove identity. The simplest example is that of the *password*, which is a secret shared between the two parties involved; password-based schemes are cryptographically weak because any eavesdropper who overhears the password may subsequently impersonate one of the parties.

More sophisticated authentication protocols use techniques from *cryptography* to eliminate the problem with eavesdroppers. In these systems, the password is used as a *key* to parameterize the *cryptographic function*. Some of these protocols depend on the strength of the particular cryptographic function involved and may leak a little bit of information about the keys used each time the protocol is run. A more powerful class of authentication protocols known as *zero knowledge authentication* probably do not leak any information. Zero knowledge protocols are an important special case of zero knowledge proofs, which have a number of important applications in computer science.

authentication protocol See *authentication*.

bootstrap The process of initializing a computer.

checksum A small output value computed using some known checksum function from input data. The checksum function is chosen so that changes in the input will likely result in a different output checksum. Checksums are intended to guard against the corruption of the original data, typically due to noisy communication channels or bad storage media.

checksum, cryptographic A checksum computed using a function where it is infeasible (other than by exhaustively searching through all possible input) to find another input which would have the same output value. Cryptographic checksums may be used to guard against malicious as well as unintentional modification of the data, since the correct checksum may be delivered by a (more expensive) tamper-proof means, and the recipient of the data may recompute the checksum to verify that the data has not been corrupted.

Cryptographic checksums are generally computed by applying a conjectured⁹ *one-way hash function* to the input. One-way hash functions have the property that they are computationally infeasible (except by exhaustive search) to invert—that is, given only the output value, one cannot easily find an input to the function that would give that output.

Another approach to cryptographic checksums is based on parameterizing the checksum function by a secret key. The Karp-Rabin *fingerprint* system uses secret keys to checksum data, forcing attackers to guess the secret key correctly; the secret key and resultant checksum must be delivered via a secure communication channel which guaran-

tees against both tampering and eavesdropping.

ciphertext See *cryptography*.

client-server model A model of distributed computing where *client* software on one computer makes requests to *server* software on the same or a different computer. Examples of the client-server model include distributed file systems, electronic mail, and file transfer.

client See *client-server model*.

cryptography The enciphering and deciphering of messages in secret codes. The enciphered messages are known as *ciphertext*; the original (or decoded) messages are known as *plaintext*. *Cryptographic functions* are used to transform between plaintext and ciphertext, and *keys* are used to parameterize the cryptographic function used (or alternatively, *select* the cryptographic function from a set of functions). The security of cryptographic systems depends on the secrecy of the keys. Generally cryptographic systems may be classified into two different kinds: *private key* (or symmetric) systems; and *public key* (or asymmetric) systems.

In private key systems, a single value or *key* is used to parametrically encode and decode messages. Thus, both the sender and the receiver of enciphered messages must know the same key. In contrast, public key systems are parameterized by pairs of keys, one for encryption and the other for decryption. Knowing one of the two keys in a pair does not help in determining the other. Thus, the encryption key may be widely published while the decryption key is kept secret; anyone may send encrypted messages that can only be read by the owner of the secret key.

Alternatively, the decryption key may be widely published while the encryption key is kept secret; this is used in *digital* or *cryptographic signature* schemes where the owner of the encryption key uses the secret key to encrypt, or *sign* a digital document. The result is a *digital signature*, which may be decrypted by anyone to verify that it corresponds to the original digital document. Since the secret key is known by the owner of that key, the cryptographic signature serves as evidence that the data has been processed by that person.

Cryptographic systems are attacked in two main ways. The first is that of *exhaustive search*, where the attacker tries all possible keys in an attempt to decipher messages or derive the key used. The second class is that of *short cut* attacks where properties of the cryptographic system are used to speed up the search.

Cryptographic attacks may be further classified by the amount of information available to the attacker. A *ciphertext-only attack* is one where the only information available to the attacker is the ciphertext. A *known-plaintext attack* is one where the attacker has available corresponding pairs of plaintext and ciphertext. A *chosen-plaintext attack* is one where the attacker may choose plaintext messages and obtain the corresponding ciphertext in an attempt to decrypt other messages or derive the key. A *chosen-ciphertext* attack is one where the attacker may choose some ciphertext messages and obtain their corresponding plaintext in an attempt to derive the key used.

cryptographic checksum See checksum, cryptographic.

cryptographic signature See *cryptography*.

cryptography, private key See *cryptography*.

cryptography, public key See *cryptography*.

daemon a program which runs unattended, providing system services to users and application programs; examples includes electronic mail transport, printer spooling, and remote login service.

Data Encryption Standard A U.S. federal data encryption standard adopted in 1976. The NSA recommended that the Federal Reserve Board use DES for electronic funds transfer applications. [60]

digital signature See *cryptography*.

executable binary A file containing an executable program; typically includes standardized headers.

fingerprint See *checksum, cryptographic*.

kernel The program which is the core of an operating system, providing the lowest level services upon which all other programs are built. There are two major schools of designing kernels. The traditional method of *monolithic kernels* places all basic system services within the kernel. In a more modular kernel design approach, microkernels provide many of the system services in a set of separate system servers rather than the kernel itself.

known-plaintext attack See *cryptography*.

microkernel See *kernel*.

monolithic kernel See *kernel*.

National Security Agency (NSA) The U.S. agency responsible for military cryptography and signals intelligence; recently more active in civilian cryptography.

nonvolatile memory (NVM) Memory that retains its contents even after power is removed.

one-way hash function See *cryptography*.

operating system The collection of programs which provide the interface between the hardware and the user and application programs.

paging See *virtual memory*.

plaintext See *cryptography*.

private key cryptography See *cryptography*.

public key cryptography See *cryptography*.

server See *client-server model*.

virtual memory A technique of providing the appearance of having more primary memory than actually exists by moving primary memory (RAM) contents to/from secondary storage (disk) as needed. The transfer of sections of this virtual memory between physical memory and secondary storage is called *paging*.

zero knowledge protocol See *authentication*.

NOTES

1. Even greater security would be achieved if the terminals were also secure; otherwise the users would have the right to wonder whether their every keystroke is being spied upon.
2. In recent work [55], we have also demonstrated the feasibility of cryptographically protecting postage franking marks printed possibly by PC-based electronic postage meters. Because such meters are located at customer sites, secure coprocessors were crucial for the protection of cryptographic keys.
3. Sufficiently sophisticated hardware emulation can fool both users and any integrity checks. If an attacker replaced a disk controller with one which would provide the expected data during system integrity verification but would return Trojan horse data (system programs) for execution, there would be no completely reliable way to detect this. Similarly, it would be very difficult to detect if the CPU were substituted with one which fails to correctly run specific pieces of code in the operating system protection system. One limited defense against hardware modifications is to have the secure coprocessor do behavior and timing checks at random intervals. There is no absolute defense against this form of attack, however, and the best that we can do is to make such emulation difficult and force the hardware hackers to build more perfect Trojan horse hardware.
4. Allowing the encrypted form of the code to be copied means that we can back up the workstation against disk failures. Even giving attackers access to the backup tapes will not release any of the proprietary code. Note that our encryption function should be resistant to known-plaintext attacks, since executable binaries typically have standardized formats.
5. We can assume that the operating system provides protected address spaces. Paging is assumed to be performed on either a local disk which is immune to all but physical attacks or a remote disk via encrypted network communication (see section below on coprocessor architecture). If we wish to protect against physical attacks for the former case, we may need to encrypt the data anyway or ensure that we can erase the paging data from the disk prior to shutting down.
6. The public key encryption requires no secrets and may be performed in the host; signing the message, however, requires the use of secret values and thus must be performed within the secure coprocessor.
7. Presumably the remote hosts will also contain a secure coprocessor, though everything will work fine as long as the remote hosts follow the appropriate protocols. The final design must take into consideration the possibility of remote hosts without secure coprocessors.
8. One way is to use a cryptographically secure random number generator, the state of which resides entirely in NVM.
9. Theorists have not been able to prove any particular function to be one-way. However, there are several functions that seem to work in practice.

REFERENCES

- [1] M. Abadi, M. Burrows, C. Kaufman, and B. Lampson. Authentication and delegation with smart-cards. Technical Report 67, DEC Systems Research Center, October 1990.
- [2] W. Alexi, B. Chor, O. Goldreich, and C. P. Schnorr. RSA and Rabin functions: Certain parts are as hard as the whole. *SIAM Journal on Computing*, 17(2):194—209, April 1988.
- [3] R. G. Andersen. The destiny of DES. *Datamation*, 33(5), March 1987.
- [4] E. Balkovich, S. R. Lerman, and R. P. Parmelee. Computing in higher education: The Athena experience. *Communications of the ACM*, 28(11):1214—1224, November 1985.
- [5] S. M. Bellovin and M. Merritt. Limitations of the Kerberos authentication system. Submitted to *Computer Communication Review*, 1990.
- [6] Robert M. Best. Preventing software piracy with crypto-microprocessors. In *Proceedings of IEEE Spring COMPCON 80*, page 466, February 1980.
- [7] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal on Computing*, 13(4):850—864, November 1984.
- [8] Andrea J. Borr. Transaction monitoring in Encompass (TM): Reliable distributed transaction processing. In *Proceedings of the Very Large Database Conference*, pages 155—165, September 1981.
- [9] Michael Burrows, Martin Abadi, and Roger Needham. A logic of authentication. In *Proceedings of the Twelfth ACM Symposium on Operation Systems Principles*, 1989.
- [10] David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030—1044, October 1985.
- [11] Ben-Zion Chor. Two Issues in Public Key Cryptography: RSA Bit Security and a New Knapsack Type System. *ACM Distinguished Dissertations*. MIT Press, Cambridge, MA, 1986.
- [12] C. J. Date. *An Introduction to Database Systems Volume 2. The System Programming Series*. Addison-Wesley, Reading, MA, 1983.
- [13] Donald Watts Davies and W. L. Price. *Security for Computer Networks: An Introduction to Data Security in Teleprocessing and Electronic Funds Transfer*, 2nd Edition. Wiley, 1989.
- [14] Dorothy Denning. *Cryptography and Data Security*. Addison-Wesley, Reading, MA, 1982.
- [15] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-26(6):644—654, November 1976.
- [16] Uriel Feige, Amos Fiat, and Adi Shamir. Zero knowledge proofs of identity. In *Proceedings of the 19th ACM Symp. on Theory of Computing*, pages 210—217, May 1987.
- [17] U. Fiege and A. Shamir. Witness indistinguishable and witness

- hiding protocols. In Proceedings of the 22nd ACM Symp. on Theory of Computing, pages 416—426, May 1990.
- [18] Morrie Gasser. Building a Secure Computer System. Van Nostrand Reinhold Co, New York, 1988.
- [19] S. Goldwasser and M. Sipser. Arthur Merlin games versus zero interactive proof systems. In Proceedings of the 17th ACM Symp. on Theory of Computing, pages 59—68, May 1985.
- [20] Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing, 1982.
- [21] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. In Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing, May 1985.
- [22] David Golub, Randall Dean, Alessandro Forin, and Richard Rashid. Unix as an Application Program. In Proceedings of the Summer 1990 USENIX Conference, pages 87—95, June 1990.
- [23] Nevin Heintze and J. D. Tygar. A critique of Burrows', Abadi's, and Needham's a logic of authentication. To Appear.
- [24] Maurice P. Herlihy and J. D. Tygar. How to make replicated data secure. In Advances in Cryptology, CRYPTO-87. Springer-Verlag, August 1987. To appear in Journal of Cryptology.
- [25] IBM Corporation. Common Cryptographic Architecture: Cryptographic Application Programming Interface Reference, sc40-1675-1 edition.
- [26] R. R. Jueneman, S. M. Matyas, and C. H. Meyer. Message authentication codes. IEEE Communications Magazine, 23(9):29—40, September 1985.
- [27] Richard M. Karp. 1985 Turing award lecture: Combinatorics, complexity, and randomness. Communications of the ACM, 29(2):98—109, February 1986.
- [28] Richard M. Karp and Michael O. Rabin. Efficient randomized pattern-matching algorithms. Technical Report TR-31-81, Aiken Laboratory, Harvard University, December 1981.
- [29] Michael Luby and Charles Rackoff. Pseudo-random permutation generators and cryptographic composition. In Proceedings of the 18th ACM Symp. on Theory of Computing, pages 356—363, May 1986.
- [30] J. McCrindle. Smart Cards. Springer Verlag, 1990.
- [31] R. Merkle. A software one-way function. Technical report, Xerox PARC, March 1990.
- [32] C. Meyer and S. Matyas. Cryptography. Wiley, 1982.
- [33] G. L. Miller. Riemann's hypothesis and a test for primality. Journal of Computing and Systems Science, 13:300—317, 1976.

- [34] R. M. Needham. Using cryptography for authentication. In Sape Mullender, editor, *Distributed Systems*. ACM Press and Addison-Wesley Publishing Company, New York, 1989.
- [35] Roger M. Needham and Michael D. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12):993—999, December 1978. Also Xerox Research Report, CSL-78-4, Xerox Research Center, Palo Alto, CA.
- [36] I. Niven and H. S. Zuckerman. *An Introduction to the Theory of Numbers*. Wiley, 1960.
- [37] Michael Rabin. Digitized signatures and public-key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, Laboratory for Computer Science, Massachusetts Institute of Technology, January 1979.
- [38] Michael Rabin. Fingerprinting by random polynomials. Technical Report TR-81-15, Center for Research in Computing Technology, Aiken Laboratory, Harvard University, May 1981.
- [39] Michael Rabin and J. D. Tygar. An integrated toolkit for operating system security (revised version). Technical Report TR-05-87R, Center for Research in Computing Technology, Aiken Laboratory, Harvard University, August 1988.
- [40] Michael O. Rabin. Probabilistic algorithm for testing primality. *Journal of Number Theory*, 12:128—138, 1980.
- [41] Michael O. Rabin. Probabilistic algorithms in finite fields. *SIAM Journal on Computing*, 9:273—280, 1980.
- [42] Michael O. Rabin. Efficient dispersal of information for security and fault tolerance. Technical Report TR-02-87, Aiken Laboratory, Harvard University, April 1987.
- [43] B. Randell and J. Dobson. Reliability and security issues in distributed computing systems. In *Proceedings of the Fifth IEEE Symposium on Reliability in Distributed Software and Database Systems*, pages 113—118, January 1985.
- [44] R. Rivest and S. Dusse. The MD5 message-digest algorithm. Manuscript, July 1991.
- [45] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120—126, February 1978.
- [46] M. Satyanarayanan. Integrating security in a large distributed environment. *ACM Transactions on Computer Systems*, 7(3):247—280, August 1989.
- [47] A. W. Schift and A. Shamir. The discrete log is very discreet. In *Proceedings of the 22nd ACM Symp. on Theory of Computing*, pages 405—415, May 1990.
- [48] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612—614, November 1979.
- [49] Adi Shamir and Eli Biham. Differential cryptanalysis of DES-like

- cryptosystems. In *Advances in Cryptology, CRYPTO-90*. Springer-Verlag, August 1990.
- [50] Abraham Silberschatz, James L. Peterson, and Peter B. Galvin. *Operating System Concepts*, 3rd Edition. Addison-Wesley, 1991.
 - [51] R. Solovay and V. Strassen. A fast Monte-Carlo test for primality. *SIAM Journal on Computing*, 6:84—85, March 1977.
 - [52] Alfred Z. Spector and Michael L. Kazar. Wide area file service and the AFS experimental system. *Unix Review*, 7(3), March 1989.
 - [53] J. G. Steiner, C. Neuman, and J. I. Schiller. Kerberos: An authentication service for open network systems. In *USENIX Conference Proceedings*, pages 191—200, Winter 1988.
 - [54] J. D. Tygar and Bennet S. Yee. Strongbox: A system for self securing programs. In Richard F. Rashid, editor, *CMU Computer Science: 25th Anniversary Commemorative*. Addison-Wesley, 1991.
 - [55] J. D. Tygar and Bennet S. Yee. Cryptography: It's not just for electronic mail anymore. Technical Report CMU-CS-93-107, Carnegie Mellon University, March 1993.
 - [56] U. S. National Institute of Standards and Technology. Capstone chip technology press release, April 1993.
 - [57] U. S. National Institute of Standards and Technology. Clipper chip technology press release, April 1993.
 - [58] U.S. Department of Defense, Computer Security Center. Trusted computer system evaluation criteria, December 1985.
 - [59] U.S. Department of Defense, Computer Security Center. Trusted network interpretation, July 1987.
 - [60] U.S. National Bureau of Standards. Federal information processing standards publication 46: Data encryption standard, January 1977.
 - [61] U.S. National Institute of Standards and Technology and National Security Agency. Federal criteria for information technology security, December 1992. Draft.
 - [62] Steve H. Weingart. Physical security for the mABYSS system. In *Proceedings of the IEEE Computer Society Conference on Security and Privacy*, pages 52—58, 1987.
 - [63] Steve R. White and Liam Comerford. ABYSS: A trusted architecture for software protection. In *Proceedings of the IEEE Computer Society Conference on Security and Privacy*, pages 38—51, 1987.
 - [64] Steve R. White, Steve H. Weingart, William C. Arnold, and Elaine R. Palmer. Introduction to the Citadel Architecture: Security in Physically Exposed Environments. Technical Report RC16672, Distributed Security Systems Group, IBM Thomas J. Watson Research Center, March 1991. Version 1.3.
 - [65] Jeannette Wing, Maurice Herlihy, Stewart Clamen, David Detlefs, Karen Kietzke, Richard Lerner, and Su-Yuen Ling. The Avalon language: A tutorial introduction. In Jeffery L. Eppinger, Lily B. Mummert, and Alfred Z. Spector, editors, *Camelot and Avalon: A Distributed Transaction Facility*. Morgan Kaufmann, 1991.

BIOGRAPHY

Dr. J. Douglas Tygar is Associate Professor of Computer Science, Carnegie Mellon University, where he is active in computer security and cryptography research, and directs the Dyad and StrongBox projects.

CMU

Pittsburgh PA 15213-3891

Telephone: (412) 268-6340

E-mail: tygar@cs.cmu.edu

Bennet Yee is a doctoral candidate in the School of Computer Science at Carnegie Mellon University. He received a B.S. in Mathematics and a B.S. in Computer Engineering from Oregon State University in 1986. His primary research interest is in computer security, and his current research centers on Dyad, a project with Doug Tygar that explores the use of physically secure coprocessors to solve otherwise intractable security problems.

CMU

Pittsburgh PA 15213-3891

Telephone: (412) 268-7571

E-mail: bsy@cs.cmu.edu

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☒ **BLACK BORDERS**

☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

☐ **FADED TEXT OR DRAWING**

☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

☐ **SKEWED/SLANTED IMAGES**

☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

☐ **GRAY SCALE DOCUMENTS**

☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**

☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

DEPOSIT, REGISTRATION AND RECORDATION IN AN ELECTRONIC COPYRIGHT MANAGEMENT SYSTEM

by Robert E. Kahn

ABSTRACT

This document proposes the development of a testbed for deposit, registration and recordation of copyright material in a computer network environment. The testbed will involve the Library of Congress and provide for electronic deposit of information in any of several standard formats, automated submission of claims to copyright, notification of registration and support for on-line clearance of rights in an interactive network. "Digital signatures" and "privacy enhanced mail" will be used for registration and transfer of exclusive rights and other copyright related documents. Electronic mail will be used for licensing of non-exclusive rights with or without recordation. Verification and authentication of deposits can be carried out within the testbed using the original digital signatures. A system of distributed redundant "Repositories" is assumed to hold user deposits of electronic information. The testbed provides an experimental platform for concept development and evaluation, a working prototype for system implementation and a basis for subsequent deployment, if desired.

INTRODUCTION AND BACKGROUND

Deposit, registration and recordation of copyright material and its associated claims to rights have generally been handled manually. Over the past two decades, the economics of information technology has enabled an electronic foundation for such material and claims. The key elements of this foundation are the personal computers, workstations, computer networks and peripheral devices such as scanners, printers and digital storage systems which have now become sufficiently powerful and cost effective to be put into widespread use. It is now essential that the underlying systems used to manage copyright be conformed to be compatible with the promise of this new computer networking environment. This paper addresses several essential steps that should now be taken to facilitate that process.

In the current manual system, claims to copyright are registered with the Copyright Office, Library of Congress. Deposits are accepted and stored in physical form including tapes and diskettes as well as paper and other substances. Notification of registration is also made in physical form. In addition, documents transferring copyright ownership and other documents pertaining to copyright may be submitted to the Copyright Office for recordation. While an on-line record of recent registrations and

recordations may be accessed at the Copyright Office, there is only limited external dissemination of this information in electronic form for access at remote sites.

This approach requires considerable physical storage at the Library of Congress for deposited materials which can only increase over time. Materials stored in physical form will slowly degrade unless deposited in digital media in which case the contents may be reproduced subsequently without loss of information but at some cost for duplication. Even if it is available digitally, much, if not most, of this material will not generally be accessible on-line from any source. Rights to use the information in a computer network environment cannot usually be acquired easily or quickly, even if the identity of the rightsholder is accurately known. Fortunately, these limitations can also be overcome with the use of information technology and only minor modification to the current manual system.

COMPONENTS OF THE PROPOSED SYSTEM

This document proposes building a testbed to develop and evaluate key elements of an electronic copyright management system. These elements include:

- a. Automated copyright registration and recordation
- b. Automated transactional framework for on-line clearance of rights
- c. Privacy enhanced mail and digital signatures to facilitate on-line transactions
- d. Methodology for deposit, registration, recordation and clearance

Current registration and recordation activities of the Library of Congress would be maintained and enhanced in the proposed testbed. It provides for repositories and recordation systems both within and without the Library of Congress, which would serve as agents for authors and other copyright owners which seek to register works with the library. In addition, the testbed provides for automated rights clearance, outside of but linked to the library, which would accelerate permissions and royalty transfers between users and rightsholders.

Electronic Copyright Management Testbed

A testbed is proposed to develop and evaluate these concepts and to obtain experience in the implementation and operation of an experimental system (see Figure 1). The proposed testbed consists of a Registration and Recording System (RRS), a Digital Library System (DLS) and a Rights Management System (RMS). The RRS will be operated by the Library of Congress and will permit automated registration of claims to copyright and recordation of transfer of ownership and other copyright related documents. The RRS would also provide evidence of "chain of title." The DLS will be a distributed system involving authors, publishers, database providers, users, and numerous organizations both public and private. It will be a repository of network accessible digital information and contain a powerful network based method of deposit, search and retrieval. The RMS will be an interactive distributed system that grants

certain rights on-line and permits the selective use of copyright material on the network.

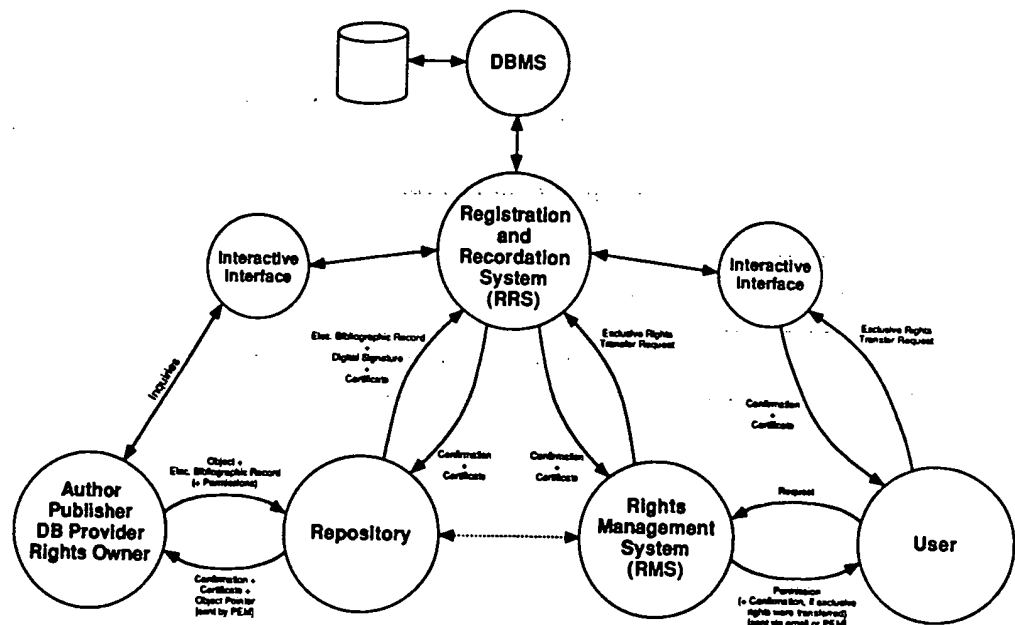


Figure 1. System Components and Information Flow.

Information may be stored in the DLS, located within the DLS and retrieved from the DLS using any of several mechanisms such as file transfer, electronic mail or agents such as Knowbot programs. Material may be imported into the DLS from other independent systems, from paper and other sources or exported from the DLS to other independent systems, to paper or to other materials such as CD-ROM, DAT, and microcassettes. The electronic copyright management system described in this document would be directly linked to the DLS.

The testbed would contain a digital storage system connected to an applications gateway (which is, in turn, connected to multiple communication systems including the Internet) to which documents would be submitted. The storage system would constitute an experimental repository for information. The applications gateway would be designed to support multiple access methods including direct login. The RRS and RMS would be servers connected to the Internet. Initially, they would be on a common machine, but they could later be easily separated. After development, the RRS would be relocated to the Library of Congress or its designated agent prior to being placed in operation. After initial implementation, the repository and the RMS would be replicable at other sites.

Electronic Bibliographic Records

An electronic bibliographic record (EBR) is created by the user for each digital document submission and supplied with the document for registration. The EBR is also suitable for use in cataloging and retrieval. The EBR may be supplied to other systems without the actual document but with a pointer to it. The EBR must contain a unique name for the

document per author. If a name is provided that has already been used by the same author, it will be rejected with an explanation. An acknowledgment of deposit will be returned to the user along with a unique numerical identifier and a retrieval pointer to the document, and, in the event of a claim to copyright, a certificate of registration from the RRS.

Claims Registration

When the EBR indicates a claim to copyright, the RRS will be supplied a copy of the EBR by the repository along with a digital signature (to be described shortly) that can be used to verify the accuracy of a deposit at a later time. The actual work would remain in the repository. The digital signature consists of a few hundred bytes of data and is approximately the size of the EBR. It should allow the authenticity of the retrieved document to be formally established at any time for legal and other purposes.

Repositories

The RRS need not be collocated with a repository. It is expected that an operational RRS would be operated by the Library of Congress. The repositories would be operated by the Library of Congress as well as other organizations or individuals. Deposits in certain qualified repositories will constitute deposit for public record purposes. The Library of Congress will maintain its own repository of selected deposits.

Although a set of distributed repositories is envisioned for a widely deployed system, the proposed testbed will only have a single repository for experimentation. The repositories would be established in such a way as to insure the survival of the deposited information with perhaps different degrees of confidence (much like the treasury, banks and brokerage houses, for example). Certain information would probably not be deposited for purposes of registration and might be stored at the users local site or in a commercial repository. Highly valued information could be stored in rated repositories (5-star down to 1-star) with varying degrees of backup and corresponding costs. The most critical information, as determined by Copyright Office regulations, might be stored at the Library of Congress or the National Archives as a safeguard. The structure of such a system of repositories should be developed as part of the project.

The advantages of a distributed repository system are:

1. Large amounts of physical storage is not required to be made available at the Library of Congress.
2. Access to the original documentation is guaranteed by the DLS to the confidence level selected by the user's choice of repository (again like the banks).
3. Repositories serve as interfaces to the users, thus offloading and insulating any central servers and systems such as the RRS from potentially large user loadings and specialized customer service requests.
4. Access to the RRS in transaction mode is available only to authorized repositories and RMSs that are qualified to use the RRS in that mode. An individual author, a collective licensing organization, a government or corporate entity or others may run an RMS. Authors and other copyright owners, as well as users may also

connect directly to the RRS through a separate interactive user interface.

The Computer Network Environment

There are three specific actions of concern in a network environment. One is the movement of information already contained in a computer network environment thereby greatly facilitating the creation of multiple copies in multiple machines in fractions of a second. The second is the importation of external information, such as print material or isolated CD-ROM based material, which must first be scanned or read into the system before it can be used. The third is export of internal network based information to paper using digital printers or facsimile machines or copied to separable media such as tape or DAT for external transport to others. Some of these actions, such as local use on paper in very small quantities, may or may not be covered by fair use provisions. However, non fair use actions would require approval of rightsholders.

In addition to the above three actions, there is a fourth action that is facilitated by the computer network environment. Information in digital form has the property of being easily manipulated on a computer to produce derivative works. Such derivative works can also be easily moved about in a computer network environment and be subject to further manipulation by other parties. The technology makes it possible for parallel and concurrent manipulation of such information to result in an exponential proliferation of such derivative works.

Rights Management System

The four actions described above form a basis for a rights management system. In general, there will be many such systems operated by rightsholders or their agents for required permissions on either an exclusive or non-exclusive basis for a given type of action. To locate an RMS, a user requires the existence of a domain server that knows about the network names and addresses of all RMS servers. Transactions involving rights may be handled by direct exchange on-line between the user system and the corresponding RMS. Typically, this exchange would occur rapidly on-line, and we refer to this as the interactive clearance of rights. Privacy enhanced electronic mail would be available for exclusive licenses and other transfers of rights. Non-exclusive licenses might be handled by regular electronic mail.

Transfer of copyright ownership would usually involve recordation in the RRS and could conceivably be handled automatically by the RMS on behalf of the rightsholder and the user to facilitate matters. The confirmation from the RRS would also be passed back to the rightsholder and user directly or via the RMS using privacy enhanced mail. Various enabling mechanisms in the normal screen-based computer interface could be developed and invoked by a user to achieve rapid clearance. If included in the user interface, this capability would have the effect of creating an instant electronic marketplace for such information.

Recordation is defined to mean the official keeping of records of transfers of copyright ownership and other documents pertaining to copyright by the Copyright Office, Library of Congress. For legal purposes, proof of official registration of claims and recordations will be

provided by the Copyright Office (via the RRS). Other registrations (at repositories) and non-exclusive licenses (via RMSs) will be certified by privacy enhanced mail. It will be up to the parties to such registrations and recordations to maintain electronic records of their transactions. These could also be stored within the DLS.

Identification Systems

The electronic copyright management system actually requires several types of domain servers. First, documents can be easily retrieved via the DLS if the citation is accurately known or through one or more search and browsing processes otherwise. However, the mapping of a bibliographic pointer (to the designated repository) into its network name and address may require a separate server. Second, the above mentioned domain server for RMSs is needed. Third, the date and time that transactions have been requested and taken may need to be formally validated. An electronic notary and time server would provide such a capability as part of the privacy enhanced mail system.

Retrieval, Appearance and Submission of Documents

Retrieval of documents from the DLS is generally a two-step process. The initial step is to identify the document and to retrieve its EBR. This record will also identify the rightsholder and any terms and conditions on the use of the document or a pointer to a designated contact for rights and permissions. Rules would have to be formulated and posted to inform clearly what obligations a user incurs when accessing the system. For example, it may be specified that a submitted request with a valid EBR will then be taken to mean acceptance of the terms and conditions, including any implementation and usage restrictions or payment requirements. The rightsholder may also wish to place restrictions on the appearance of documents for certain uses.

As part of the process of document submission, a valid EBR will have been produced which can be used in the author's system. Each author or other owner of copyright (or such owner's successor in title or duly authorized agent) will maintain his or her own collection of EBRs. Searches and requests will typically be made to the user's home system unless the rights have been transferred or delegated elsewhere (e.g. to a publisher, agent, or database provider). In applying for registration of claims to copyright at the Copyright Office, a user could be required to certify that he or she has the rights to the material and sign the submission digitally.

PRIVACY AND AUTHENTICATION TECHNOLOGY

This section briefly describes several key technologies to handle privacy and authentication in the digital network environment. Four such technologies are described below, namely: 1) Public Key Cryptography, 2) Digital Signatures, 3) Privacy Enhanced Mail, and 4) Notarization.

Public Key Cryptography

In conventional cryptography, a mathematical function and a "secret key" are shared by parties who wish to communicate confidentially. Each message to be sent is "encrypted" using the function and key and the

recipient(s) "decrypt" it using the same function and key. This may be thought of as sharing a locked box in which several individuals have the key and any of them can lock or unlock the box at will.

In the late 1970's, two Stanford University researchers, Martin Hellmann and Whitfield Diffie speculated that it might be possible to devise paired cryptographic functions which had the interesting property that one function would encrypt and the other would decrypt. In fact, the concept was slightly more sophisticated in that any message encrypted with either one of the functions could only be decrypted by the other. In other words, having access to the function which did the encrypting does not help when it is time to decrypt. Using the box analogy, the public key cryptography system would be like having a box with a two-key lock. If one of the keys is used to lock the box, the other must be used to unlock it. A person holding a key used for locking could not use it for unlocking.

One of the biggest problems with conventional cryptography is that the keys must be kept secret and must be distributed by secure means. The notions of Hellman and Diffie opened up a new way of thinking about key management. One key could be made public (e.g. the one to be used for encryption) and the other kept private. Anyone knowing the public part of a pair of keys could use it to prepare a message which would remain confidential until the person knowing the private key used it to decrypt the message. The public keys could be listed in public directories without any special protection since knowing them did not help anyone decrypt messages encrypted using the public key. This feature makes it far simpler to manage key distribution since the public part need not be protected.

Three researchers at MIT, Rivest, Shamir and Adelman developed a pair of functions meeting the requirements specified by Diffie and Hellman. These functions are now known as the RSA algorithms (from the last names of the inventors).

Digital Signatures

Since either key of a public key cryptography pair can be used to perform the initial encryption, an interesting effect can be achieved by using the secret key of the pair to encrypt messages to be sent. Anyone with access to the public key can decrypt the message and on doing so successfully, knows that the message must have been sent by the person holding the corresponding secret key. The use of the secret key acts like a "signature" since the decryption only works with the matching public key.

Buyers could send digitally signed messages to sellers and the sellers could verify the identity of the sender by looking up the public key of the sender in a public directory and using it to verify the source of the message by successfully decrypting it.

Privacy-Enhanced Mail (PEM)

Public key cryptography can be combined with electronic mail to provide a flexible way to send confidential messages or digitally signed messages or both. In actual practice, a combination of public key, conventional secret key and another special function called cryptographic hashing is used to implement the features of privacy-enhanced mail. The

public key algorithms require a substantial amount of computing power compared to conventional secret key algorithms. The older secret key algorithms, such as the Data Encryption Standard (DES) developed by the National Institutes of Standards and Technology (NIST), are much more efficient. Consequently, confidential messages are typically encrypted using a conventional secret key which, itself, is sent, encrypted in the public key of the recipient. Thus, only the recipient can decrypt the conventional secret key and, eventually, decrypt the message.

To send digitally-signed messages, each message is run through a "hashing" algorithm which produces a compressed residue which is then encrypted in the private key of the sender. The message itself is left in plain text form. The recipient can apply the same hashing algorithm and compare the compressed residue against the one that was sent (after decrypting it with the sender's public key).

One of the basic problems with this application of public key cryptography is knowing whether the public key found in the directory for a given correspondent is really that correspondent's key or a bogus one inserted by a malicious person. The way this is dealt with in the Privacy-Enhanced Mail system is to create certificates containing the name of the owner of the public key and the public key itself, all of which are digitally signed by a well-known issuing authority. The public key of the issuing authority is widely publicized so it is possible to determine whether a given certificate is valid. The actual system is more complex because it has a hierarchy of certificate issuers, but the principles remain the same.

Notarization

Using digital signatures, it is possible to establish an on-line notarization service which accepts messages, time-stamps them and digitally signs them, then returns them in that form. If the person desiring notarization digitally signs the message at the time it is sent to the notarizing service, then it will be possible, later, to establish that the person requesting the notarizing had the document/message in question at the time it was notarized. One can imagine that the originator of a message might have it notarized for the record and the recipient might independently do so. By this means, for instance, evidence of a contract's existence in the hands of each party at particular times might be established.

VERIFICATION, AUTHENTICATION AND CERTIFICATION

The verification process uses stored digital signatures to ascertain whether a given copy is identical to the version which was originally deposited. If any portion of the copy differs from the original, the verification process will fail. Authentication or formal certification of deposits may be provided to a requesting party in traditional ways or via electronic mail. Privacy enhanced mail would be used to certify the authenticity of a deposit, as well as to certify registration and recordation records, for legal purposes.

CONCLUSION

The deployment of an electronic deposit, registration and recordation capability for use in a computer network environment would greatly facilitate and accelerate the move to a network base for information creation and dissemination. The system would be compatible with the current manual system and would support the ability of the Library of Congress to provide automated registration and recordation services. It would provide a foundation for straightforward and easy expansion and evolution and provide a direct linkage for the Library of Congress to the DLS. It would provide a prime working example for all other kinds of activities where claims registration and rights management come into play. Verification and authentication of copies of deposits may be performed electronically using digital signatures. Formal certification of deposits, as well as registration and recordation records, using privacy enhanced mail may be provided for legal purposes. A testbed which demonstrates the relevant concepts and ideas can be implemented within a two to three year period with initial limited use within a year.

Robert E. Kahn, Ph.D.
President
Corporation for National Research Initiatives
Suite 100
1895 Preston White Drive
Reston, VA 22091-5434

THIS PAGE BLANK (USPTO)

Internet Billing Service Design and Prototype Implementation

by Marvin A. Sirbu*

ABSTRACT A group of students in the M.S. program in Information Networking at Carnegie Mellon University have designed and implemented a prototype of an Internet Billing Service—an electronic credit card service for the Internet environment. The service provides account management, authentication, access control, credit verification, management reporting, billing, and collection services to network-based service providers.

INTRODUCTION A worldwide data networking infrastructure is gradually falling into place which will allow consumers and service providers to interact in a vast electronic marketplace. In France some 9,000 services are available over the Minitel network. In the U.S., access to electronic databases generates billions of dollars a year in business. As networked computers proliferate at home and in the workplace, more and more consumers are in a position to shop in the electronic marketplace.

Already the Internet, a loose confederation of independently managed networks, links eight million users and some one million computers on 10,000 separate subnetworks in more than 40 countries. Once used only by universities and research organizations, the Internet is used today by individuals and corporations for a wide range of commercial purposes including disseminating information, searching remote databases, and providing access to specialized computer resources. Major information service providers such as Dialog and BRS can now be reached via the Internet.

While it is relatively simple for an entrepreneur to set up a small computer capable of providing information to the worldwide Internet community, it is much more difficult to arrange a mechanism to charge users for the services rendered and to collect payments. Current billing mechanisms for electronic services are costly and inconvenient for both service providers and end users. In the absence of a centralized billing

* This report represents the collective work of 13 students in CMU's Master of Science Program in Information Networking and is derived from their final project report: Richard Batelaan, Richard Butler, Chun Yi Chan, Tie Ju Chen, Michael Evenchick, Paul Hughes, Tao Jen, John Jeng, Jon Millett, Michael Riccio, Ed Skoudis, Chris Starace, and Peter Stoddard. The project was directed by William Arms, John Leong and Dennis Smith of CMU. Dhananjay Gode served as Teaching Assistant and prepared the first draft of this paper.

service, users must initiate a service agreement with each service provider before using its services, and must keep track of its access point, password, and bills. Also, there is no central directory of service providers. It is uneconomical for small service providers to advertise, check credit, authenticate users, control access, bill and collect payments, maintain audit trails, and keep usage statistics.

Both service providers and end users need a reliable, easily accessible, fast and inexpensive intermediary, a billing service. The billing service could be compared to an electronic credit card for services on a network. It would allow small service providers to concentrate on providing services by contracting out the functions of billing users and collecting payments.

This document describes the design and implementation of a prototype of a computer-based Internet Billing Server (IBS) developed by a project team from the Information Networking Institute (INI) at Carnegie Mellon University. The project team had two major tasks: (1) specify functional requirements for a full-scale billing server, and (2) design and develop a prototype which, while satisfying only a subset of these requirements, demonstrates the feasibility of the concept. Specification of the full set of requirements brings out important issues and ensures that the prototype design has no major flaws which limit its extensibility to a full-scale system. We will summarize the full set of requirements, noting in passing where the actual prototype differs. Our design demonstrates how an Internet Billing Server could facilitate the emergence of a real electronic marketplace.

UNIQUE CHARACTERISTICS OF NETWORK-BASED MARKETS

The design of a network billing server is difficult because markets for electronic services are different from markets for physical goods and non-electronic services. A network billing server must be designed to take these differences into account, in order to avoid fraud and disputes. These differences are outlined below.

First, in a network, the users, the service providers, and the billing service are geographically separated. Credit cards are designed for situations where the users physically present their credit cards at the time of purchase so that merchants may validate their signature. Although credit cards are used today for placing orders over the phone, such methods are highly insecure; ordering over a network makes it difficult to verify the identity of the parties. Indeed, to reduce fraudulent charges, many merchants will only ship goods ordered over the phone to the billing address of the credit card holder. Secure network authentication protocols, such as Kerberos, may be part of a solution but the legal liability and responsibility of participants in an electronic market is not well defined.

Second, given the high processing speeds of electronic services, a user can accidentally run up a huge bill within a matter of seconds without having an opportunity to cancel it. In contrast, if there is a mistake in ordering a physical product, it can be corrected before the product is shipped or the product can be returned after delivery. Even though a non-electronic service cannot be returned, the user can still cancel it while it is being performed: one can leave a hotel if its service is not satisfactory

or is too expensive. Providing a similar capability for halting the provision of network-based services in midstream is complex.

Third, in contrast to physical goods, it is difficult to determine the price for an electronic service in advance. For example, if the price of a database query were based on the number of bytes of information it generates, it would be difficult to determine the query's price without searching the database. It is infeasible to let the user have a look at the information to assess its value; it is also infeasible to price information solely upon objective measures such as its size in bytes. This difficulty in judging the quality of information may give rise to disputes which are difficult to resolve. This makes it important to carefully define the legal role of the billing server.

Fourth, electronic information can be easily duplicated and redistributed. This makes product "returns" meaningless because a user can copy an electronic file before returning it. It is also much easier to copy and redistribute an electronic version of a book than copy and redistribute a printed version of the same book.

The INI Internet Billing Server is able to address some, but not all, of these issues. Security is provided using passwords and encryption. The IBS also provides a capability for setting and enforcing spending limits. The billing server provides a flexible mechanism for charging for network services, and for price negotiation, but it does not pretend to resolve the problem of determining a service's value. Nor does it address the issue of illegal copying and redistribution of purchased information.

FUNCTIONS PROVIDED BY A BILLING SERVER

A billing server plays the same role vis-à-vis end users and service providers as a credit card company does vis-à-vis cardholders and merchants. Consider a credit card holder going to rent a car. He begins by identifying himself to the rental car company by presenting his credit card and driver's license. He negotiates with the car company for the service he desires, the cost per day and the maximum number of days he expects to keep the car. The merchant then verifies the customer's credit with the credit card issuer and places a hold on the customer's credit for the maximum amount of the rental. When the customer returns the car, the transaction is complete, and the car rental agency sends a final invoice to the credit card company, with a copy to the consumer. At the end of the billing cycle, the credit card company sends a bill for all purchases charged to the card, including the car rental, and the customer sends back his payment. The credit card company pays the merchant after deducting its fees.

Our model of transactions in the network marketplace is similar to the car rental scenario: a customer or end user—through his computer—interacts over a network with two other computer systems: the *service provider*, and the *Internet Billing Server*, as illustrated in Figure 1.

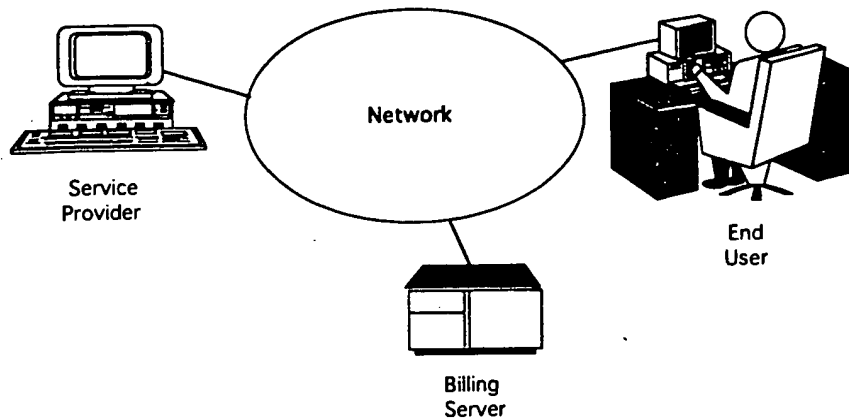


Figure 1. Electronic Marketplace

All of the steps described above for renting a car have their counterparts in the network marketplace. However, instead of face-to-face communications as in the car rental scenario, the end user's computer interacts with the service provider's computer over the network. Each step in the interaction forms part of the Internet Billing Protocol (IBP), a proposed standardized method of interaction among an end user's computer, the service provider's computer, and the Internet Billing Server computer.

The Internet Billing Server is more than a computer and a set of standardized protocols, however; it is a model for a business which provides valuable services to network marketplace entrepreneurs. The Internet Billing Service acts as a factor for the service provider, providing prompt payment while taking over all aspects of billing and collections. To be successful as a business, the Internet Billing Service must satisfy two sets of customers. It must attract merchants by giving them easy access to a large group of customers, and providing them a cost-effective way to receive payment for services provided. It must make it as easy as possible for service providers to make use of the Internet Billing Server, working with the providers to modify both client and server software to implement the Internet Billing Protocol. It must attract end users by providing a powerful and flexible capability for managing end-user accounts and by giving end users access to a large number of service providers.

While we have described the Internet Billing Service as an independent business, large organizations often have a need to create an internal equivalent of an Internet Billing Server. For example, the central administration at a university such as CMU could operate a billing server as a single mechanism to bill for online library access, computing services, printing services, and electronic mail. While we recognize this as another potential application for the billing server software developed in this project, the focus has been on the design and implementation of a public, third-party billing server.

As designed by the project team, the INI Internet Billing Server provides the following functions to service providers and end users:

- *Account Management.* The IBS enables end users to establish an account relationship with the Billing Server which will permit

them access to any number of service providers. Service providers establish accounts which enable them to use the IBS to bill their clients for services rendered.

- *Authentication:* The IBS provides a service for authenticating both end users and service providers prior to any transaction and for secure communications between the parties.
- *Access Control:* The IBS provides access control for both end users and service providers. Information associated with an end user account can specifically designate a list of services that may be accessed, or a list of services that specifically may not be accessed.
- *Price Negotiation:* Using the Internet Billing Protocol, the end user may determine the services available from the service provider and the posted prices. The Internet Billing Server can record the mutual agreement of the end user and the service provider on a set of prices, and the maximum amount which the end user has authorized for this set of transactions.
- *Credit Verification:* The Internet Billing Service will verify to the service provider that the customer has sufficient credit to pay for the proposed transaction up to the agreed cap.
- *Final Invoice:* At the conclusion of the transaction, the service provider can send a final invoice to the Billing Server using the IBP. The Internet Billing Server will send an authenticated copy of the invoice to the end user.
- *Periodic Billing:* The Internet Billing Server will generate periodic billing statements to customers detailing all of their transactions and the sums owed.
- *Collections:* The Internet Billing Service will collect funds from end users and make payments from these funds to service providers.
- *Directory Services:* The Internet Billing Service provides a "white pages" and "yellow pages" service for identifying service providers.
- *Help Service:* The Internet Billing Server provides an online help manual service.
- *Software Libraries.* In the client server model, every service provider must make available to end users client software capable of accessing the service provider's service. A successful Internet Billing Service must provide a set of library routines which make it simple to upgrade both client and server software to support the Internet Billing Protocol. These modules are shown logically in Figure 2.

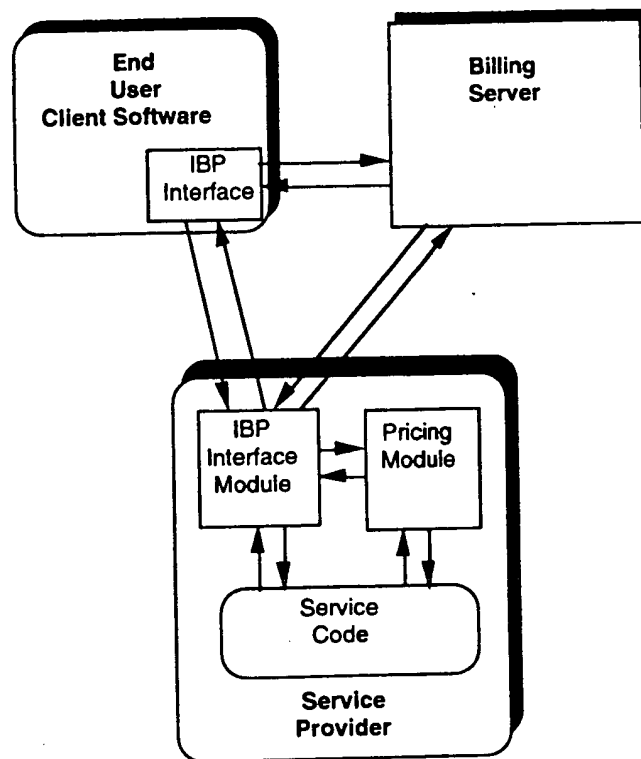


Figure 2.

Interface Libraries Provided by Internet Billing Service

DESIGN OBJECTIVES

In developing the Internet Billing Server and the Internet Billing Protocol we were guided by several fundamental considerations.

- The Internet Billing Server will operate in a transaction-oriented environment. All communications between the parties will be based on a remote procedure call communications paradigm. This is in sharp contrast to most current network-based information services. These typically have been provided via large timeshared computers. Users log in over low-speed networks from dumb terminals—or PCs emulating dumb terminals—and are charged by connect time. However, as desktop computers have replaced dumb terminals and networks have increased in speed, a new information access paradigm has emerged: client-server. In this paradigm, powerful desktop computers running user-friendly client software interact with remote servers on a transaction basis. In a few seconds large files of information can be requested and transferred from servers to clients. File Transfer Protocol, Gopher, and Wide Area Information Service (WAIS) are but a few of the client-server protocols used by numerous clients and servers on the Internet. In a client-server environment there is no notion of connect time. Accordingly, services must be billed on a per-transaction basis.

- The billing server should have high availability since, in its absence, the service providers will not be able to offer their services to end users.
- The billing server should be scalable. It is difficult to predict the initial number of customers and the growth pattern, even though the number of potential customers is large. These latter two points suggest that the billing server should be designed to run on replicated, distributed computers, thus providing modular scalability and high availability through redundancy.
- Communications between the parties—end user, service provider, and billing server—should be based on widely available telecommunications standards to ensure the largest market for the services.
- Secure authentication and encryption are critical because all three parties will be connected via insecure public networks. Without a secure authentication mechanism there is a substantial potential for fraud.
- Before using a service, the end user must understand and agree to the prices and terms of the exchange. The transaction protocol in the billing system must support an initial price negotiation between the end user and the service provider. The billing server should be informed about the outcome of this negotiation by both the service provider and the end user. To avoid disputes the billing server should make sure that the user and the service provider have the same version of the agreement.
- The users should be able to limit their financial exposure on a transaction by specifying a spending cap. If the cost of an ongoing transaction exceeds this spending cap then the end user should be able to choose whether to abort the transaction, or continue it by raising the spending cap.
- The billing server should not become a bottleneck slowing the speed of interaction between the end user and the service provider. In particular, the billing server should not be a gateway for communication between the end user and the service provider. Interactions with the billing server should be as few and as simple as possible.
- The billing server software should help users in their account management. It should support hierarchical accounts so that corporate users can get bills aggregated by organizational units such as departments, regions or divisions. Similarly, a provider of multiple services may use an account hierarchy to organize information on the use of each service.

With these general requirements in mind, the project team prepared a detailed requirements document specifying all of the capabilities required of the Internet Billing Server.

DESIGN OF THE INTERNET BILLING SERVER

Our prototype Internet Billing Service was implemented using widely available technology. The Billing Server prototype is designed to run on a Digital Equipment Corporation workstation class machine running the

Ultrix operating system. It was written in C and uses the Ingres Database Management System. It also uses Transarc Corporation's Base Development Environment (BDE) to provide multithreading, which allows the prototype to process concurrent requests efficiently. For communication between the billing server, end users, and service providers, the prototype uses the remote procedure call (RPC) portion of the Distributed Computing Environment (DCE) provided by the Open Software Foundation and the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol suite. Implementations of DCE are available for the OS2/2.x and Microsoft's Windows NT operating systems as well as Unix.

Authentication is implemented using the Kerberos protocol developed at M.I.T. All communications between the parties are encrypted for security using the Data Encryption Standard (DES) encryption method.

Code libraries enabling rapid modification of client and server software to support the Internet Billing Protocol were written in C. As a test of the complete system, we modified versions of File Transfer Protocol (FTP) client and server software to make use of the Internet Billing Server. Using these software packages, a service provider could distribute information using FTP and bill for it using the Internet Billing Server.

TRANSACTION SEQUENCE

Figure 3 illustrates the sequence of steps involved in the use of the Internet Billing Server.

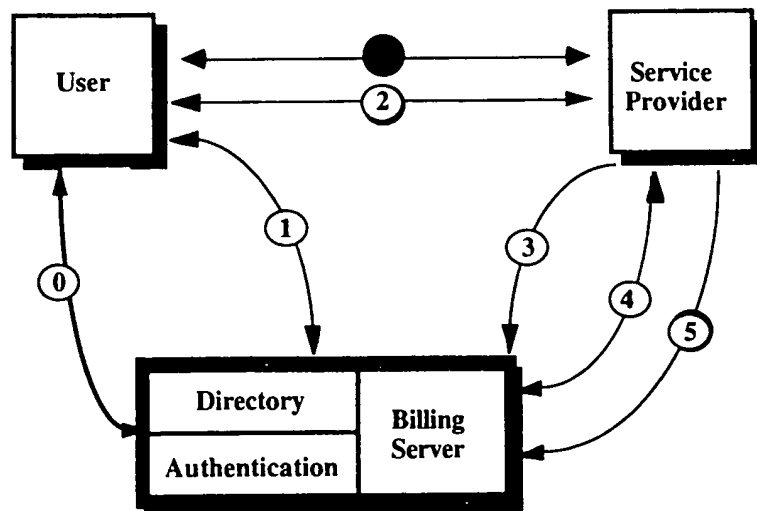


Figure 3: Transaction Sequence for the Billing Server

Step 0. Establishing an Account

Prior to engaging in a network-based transaction, an end user must first establish an account with the Internet Billing Server. In our design, any number of accounts may be organized in a hierarchical fashion by allowing each account to have *sub-accounts*, each of which is also an account. See Figure 4 for an example of a single hierarchical account structure.

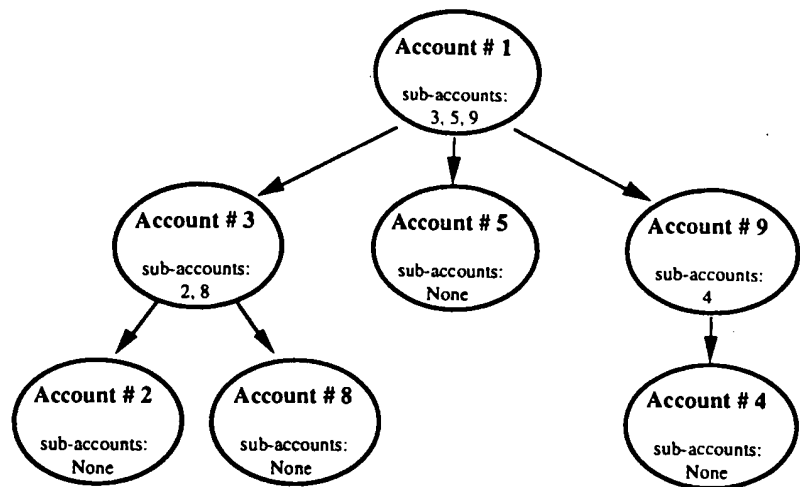


Figure 4: An Example of an Account Hierarchy

The hierarchical structure represents authority over accounts; the end user of a parent account has authority over the end user of a sub-account. Every hierarchy has an Account Administrator, who is able to view financial information, and modify certain account characteristics for all the accounts in the hierarchy.

Billing and usage information can be aggregated by various branches of the hierarchical structure, or detailed information for each node in the structure can be supplied. An organization should have the ability to give managers privileges to modify some of the information for the accounts of their subordinates. This hierarchical structure allows the account environment within the Internet Billing Server to mirror the environment within organizations.

Account hierarchies may be individually or collectively billable. In the first case each account is fully billable, i.e., it contains all the financial information such as balance due, adjustments, payments and usage information; the hierarchy is used merely to provide aggregate billing information for management and control. This model may be more appropriate for decentralized organizations. In the second type of hierarchy only the parent or root of the hierarchy is fully billable, i.e., only the root account contains the full billing information for the hierarchy whereas for other accounts only the usage information is listed. The prototype only supports hierarchies where each account is a fully billable account.

Service providers which offer multiple services may want to order their accounts in a hierarchy to help maintain valuable marketing and usage information. Since each distinct service requires a unique Kerberos identifier and an account will not provide multiple Kerberos identifiers, each service must be given a separate account within the Internet Billing Server. By allowing these accounts to be placed into a hierarchical structure, the Internet Billing Server can make one aggregated payment to the service provider instead of a separate payment for each service. In addition, hierarchical accounts make it easier to supply the service

providers with one statement containing usage information for all of their services.

Step 1 - User Authentication and Access Control

Since a network is a mutually suspicious environment, the service providers, the end users, and the billing server must authenticate each other prior to any transaction. This step may be compared to credit card users showing their driver's license to prove their identity while using a credit card. As noted, we use a Kerberos-based authentication system for secure communication between end users, service providers, and the billing server. Cross-realm Kerberos authentication may be required in the full-scale system if large users authenticate end users within their organization and then ask the billing server to accept their authentication. However, our prototype does not need cross-realm Kerberos authentication because it functions within a small group of end users and service providers. All communication is encrypted using the Data Encryption Standard for security.

After authentication, the end users may directly request access to a specific service provider, or may search through an index of service providers classified by service categories to select the service they want. The prototype does not support the directory service. The billing server checks the access control lists of both the end user and the service provider to ensure that the end user is allowed to access the requested service. In the full set of requirements, the end-user and service-provider accounts may have two types of access control lists: (1) two positive access control lists—one listing *specific* service providers/end users and the other listing *categories* of service providers/end-users; and (2) similarly, two negative access control lists. End users' lists specify which service providers they can (positive lists) or cannot (negative lists) access; similarly, service providers' lists specify which end users are allowed or not allowed access to them.

The negative access control lists override the positive lists, and determine which specific services or service categories cannot be accessed from the account. Corporate users could use positive access lists to allow access only to company-approved service providers. Parents could use negative access lists, analogous to 900 telephone service blocking, to prevent their children from accessing frivolous or high-cost services. We have implemented only negative access lists for end users and only positive access lists for service providers.

If access is allowed, the billing server issues the end user a Kerberos ticket for the service provider; that authenticates the end user to the service provider. As mentioned before, the end user must have the client software specific to the service provider (for example FTP or Internet Gopher interface software) in order to access the service provider.

Steps 2 and 3 - Price Negotiation and Spending Cap

After getting a Kerberos ticket from the billing server, the end user and the service provider negotiate a price for the requested service and a spending cap for the transaction. This is called an agreement. Note that the end user is communicating with the service provider's computer, not with a human representative of the service provider.

The end user sends a copy of the agreement, encrypted with his private key, to the service provider who forwards the end user's copy to the billing server, along with his own copy of the agreement. This prevents an unscrupulous service provider from changing the agreement before sending it to the billing server. It also reduces the communication load on the billing server, since it receives only one combined message rather than two separate messages from the service provider and the end user.

The full-scale billing server allows renegotiation of spending caps if the initial spending cap proves to be insufficient. However this capability was not implemented in the prototype.

Step 4 - Verifying Spending Cap and Credit

The billing server decrypts the two copies of the agreement and compares the end user's version with the service provider's version. If the two copies match, then the billing server checks if the end user has sufficient funds to pay for the transaction and places a hold on the end user's funds in the amount of the spending cap. It then sends an authorization to the service provider.

In the full-scale server, the end users can specify their preferred payment method; this could be historical billing, advance deposit or credit card. With historical billing the user receives a bill for the services that they used at the end of a specified period of time. Advance payment means that the user deposits funds with the billing server before using services, and receives a periodic statement of the services used and the funds remaining. With credit card billing, their credit card is billed when accumulated charges reach a specified limit. In addition to these three options, corporate users can use purchase orders, a form of historical billing, for making payments. The prototype allows deposit in advance as the only payment method.

Step 5 - Performing the Service

Messages are exchanged between the client and the service provider to perform the service—e.g. retrieve information, perform calculations, or spool a print file.

Step 6 - Generating an Invoice

After the service provider has rendered the service, it sends the billing server an invoice detailing the services performed and the actual amounts to be charged. The billing server checks whether the price information on the invoice is identical to the price information received earlier during the price negotiation stage. This protects the users from unscrupulous service providers. The billing server then forwards this invoice to the user. Since the identity and credit capacity of the end users were previously checked by the billing server, the service providers have assured payment for their services. The service providers are required to maintain an audit trail to handle customer inquiries which cannot be resolved by the billing server.

ACCOUNT MANAGEMENT

End user accounts can go through various states, as illustrated in Figure 5. To open an account, the end user account administrator sends a request to the billing server. Once all of the information required for the creation of an account is entered, the account enters the "new" state. An account cannot begin to access services until the billing server's account administrator verifies and approves the account characteristics and the billing server's financial administrator verifies and approves the financial information. Once these verifications are complete, the account is activated. An account which has been activated enters the "active" state and is then allowed to accumulate charges for services it accesses through the billing server.

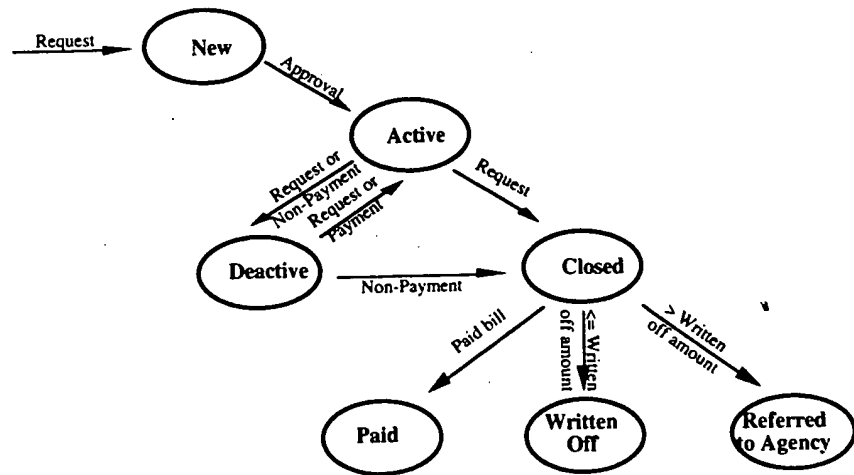


Figure 5: Account State Diagram

An account goes into the "deactive" state if it has an overdue balance for an unreasonable period of time. It goes back to the "active" state if the balance is paid. An account can also enter the "closed" state by end user request or if payment is not received while it is in the "deactivated" state.

Accounts may enter the "paid," "written off," or "referred to agency" states after being in the "closed" state. An account enters the "paid" state if the final balance due is paid in full. An account enters the "written off" state if the billing server financial administrator determines that payment for the balance due will not be received. An account enters the "referred to agency" state if the billing server's financial administrator refers the account to a collection agency.

Since the prototype handles only debit model accounts where users pay in advance, there is no need for an approval process. The "new" state is not needed. Again because of payment in advance and the credit check performed during transactions, end users' accounts cannot owe money to the billing server. Therefore, the "deactive," "paid," "written off," and "referred to agency" states are also not needed. In the prototype, when an account is "closed" it is removed from the database. Therefore account states are not supported by the prototype.

Users can access their own account information at the billing server through an interface that allows them to view financial information, and to modify certain account characteristics. The full-scale billing server also provides on-line help to its users. The help, which can be accessed through a keyword search, consists of text screens describing how to perform basic operations. Since on-line help is not central to the billing server, it is not implemented in the prototype.

CONCLUSION

What distinguishes the Carnegie Mellon project from other piecemeal or service-specific solutions is its comprehensive analysis of the network services billing problem. The project has made two contributions: (1) it has highlighted the complex and challenging issues involved in the design of the Internet Billing Server; and (2) it has demonstrated the feasibility of its proposed solution through the successful design and implementation of the prototype. Even though the prototype implements only a subset of the full requirements and may have to be significantly modified, it is an important first step. A commercial service based on the concepts in the INI Internet Billing Server could be the key to the rapid growth of entrepreneurial service providers in the Internet environment.

For further information, please contact The Information Networking Institute at Carnegie Mellon University, Pittsburgh, Pennsylvania 15213. Tel: (412)-268-7195.

REFERENCES

1. John Quarterman. "In Depth. (What can businesses get out of the Internet?)" *Computerworld*, February 22, 1993, p. 81.
2. For a complete report of the project including the full Requirements Specification and Prototype Design Documents, contact the Information Networking Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213-3890.
3. Jennifer Steiner, Clifford Neuman and Jefferey Schiller. "Kerberos: An authentication service for open network systems." *USENIX Winter Conference*, 9-12 February 1988, Dallas, Texas.
4. Richard Batelaan, et.al. *An Internet Billing Server: System Requirements*. Carnegie Mellon University Information Networking Institute, August 1992.

THIS PAGE BLANK (USPTO)

Metering and Licensing of Resources: Kala's General Purpose Approach

by Sergiu S. Simmel and Ivan Godard

ABSTRACT This paper describes the licensing and metering capabilities of Kala¹, a persistent data server. Kala offers a suite of low-level primitives for constructing both simple and sophisticated licensing (pay-per-user) and metering (pay-per-use) models. Kala allows the licensing and/or metering of access to any software facilities, both data (passive resources) and executable code and associated services (active resources). A few examples model concrete business needs, bridging the gap between technologically motivated mechanism and business motivated policies.

INTRODUCTION This section motivates the work on economic grounds. It also introduces several terms used throughout the paper.

Components, Subassemblies and Applications

It is natural to think of software as being an assemblage of *components*. Software structure starts with small-grain components such as functions, classes, and data values. These are grouped in *subassemblies*, such as a library or a subsystem. Successive composition finally yields what we all think of as *applications* — a conventional packaging of functionality.

The basic distinction between components, subassemblies and applications is that of granularity. In electronic hardware the analogous entities are electronic components (such as integrated circuits and passive components), boards (usually ready to be plugged into bus connectors), and systems (such as personal computers).

For simplicity, we will use the term *component* to mean both components and subassemblies. We will use the term *subassembly* only if there is need to distinguish them from small-grain components.

Both program components and data components exist. The former includes such examples as operating system subsystems, runtime libraries, specialized classes, etc. The latter includes font collections, clip-art sheets, economic indicators, and so on.

Components, subassemblies, and applications go through an economic cycle which includes two mechanisms relevant to our discussion: the distribution mechanism and the revenue collection mechanism.

Distribution is the mechanism by which a component is made available to another component or final consumer (end-user). Several distribution techniques are conventionally used throughout the industry: embedding (static linking), runtime linking, and runtime loading.

Revenue Collection is the mechanism by which payment for a component is made to reach the producing vendor, regardless of the context in which the component is actually used. Conventionally, revenue collection is done at the time the component is distributed.

The Economics of Components

Markets for software applications have been established for some time. There is a market for subassemblies as well, although substantially smaller and with much less potential for growth and diversification. The market for small-grain components is very small, although a great many are given away free or bundled with larger units.

Many analysts attribute the relative lack of an open market for components to the tight coupling between delivery and revenue collection. The reasons include, among others:

- **Hard to Control Scarcity.** Distribution cannot be effectively controlled because information is easy to duplicate. This leads to large percentages of fraudulent use.² Since the software industry needs to collect enough revenue to pay its own bills, the result is higher prices paid by a small fraction of users, with a secondary effect of diminishing market sizes.
- **Hard to Select.** It is difficult and inconvenient to "test-drive" software components and applications, because users are asked to pay the high price of acquisition before being able to determine whether the software is of adequate quality or even needed.
- **Hard to Get Fair Revenue.** It is difficult to predict actual distribution volumes at the time redistribution arrangements are made. For simplicity, many such arrangements are based on a flat fee. Inevitably, many end up being unfair to one of the parties, with negative effects on the entire industry.

An unprofitable component market deprives the software industry at large of:

- **Well Crafted Components.** Components should be produced by the best specialists in the relevant technology. However, there is no incentive for them to enter this market, since there is little chance that their efforts would pay off.
- **Higher Quality Infrastructure.** Most of the industry's focus is on producing applications — the only merchandise one can make real money on. This strong bias has negative long-term effects on the quality of the software infrastructure and the specialized components out of which the industry builds these applications.
- **Slower development of technology.** Fewer quality components leads to less reuse and more reinventing the wheel.

The natural solution to all of these problems is to decouple delivery and revenue collection: make the delivery mechanism direct, easy, and largely free, and then use a separate mechanism that insures collection

and proper allocation of revenue. The only charges at distribution time should be to cover media manufacturing (including printed documentation, if any), and physical transportation, if any. These charges can also be eliminated in many (but not all) instances through delivery via high-speed networks.

A simple solution, indeed. Why has it not happened? Reasons are multiple, including:

- **Cultural Barriers.** We've done business in one way for over 40 years now. The basic methods were established in an era where mainframe software was the predominant product. Additional techniques were added with the advent of standalone personal computers. While most of the software (by revenue) to be sold over the next decade will not fall into either of these categories, the culture that generated the practices is still strong.
- **Opposition by Monopolies.** The large monopolies perceive change that may establish a fairer way of compensating value, emphasizing quality, and encouraging the small but highly skilled producer to be not in their best immediate interest. While one could argue that this is just a matter of perception, and that in fact change is in everybody's best long-term interest, the perceptions remain and strongly influence the attitude of many of the larger participants in this industry.
- **Absence of Supporting Infrastructure.** To actually effect such a change, a simple, secure, flexible, and general purpose infrastructure must exist and support the new manner of collecting revenue well. In its absence, the discussion remains academic.

While we recognize the importance and seriousness of the first two barriers listed above, we are not addressing them here.³ This paper is about the third barrier. Our thesis is that such a technological infrastructure now exists. We are presenting both a model and a concrete, industrial-quality, commercial implementation of it, as part of the Kala technology and persistent data server product.

Revenue Collection

To be useful in practice, the revenue collection mechanism must satisfy several requirements:

- a. **Be safe.** The mechanism must address safety very seriously, so that it seen as truly solving the revenue loss problem, as opposed to replacing it with another variant of it. Thus, the mechanism must be foolproof against major fraud.
- b. **Be recursive.** The mechanism must not only allow the collection of revenue, but also the allocation of some portion of it to subsidiary suppliers. This reflects the components-made-out-of-components structure of software.
- c. **Be flexible.** The mechanism must be just that: a mechanism. It should allow the component vendor the maximum flexibility to establish policies and reflect business arrangements and special cases.

- d. **Be efficient.** The mechanism must introduce very little, if any, overhead to the normal functions of the components and applications it supports.
- e. **Be invisible.** The mechanism must be largely transparent, and simplify customers' lives, rather than become a nuisance like dangle chains.

There are two principal arrangements possible between the producer and the consumer of a component:

- **Pay-per-use.** This is an arrangement whereby the consumer pays the producer for as much of the producer's software as the consumer's software actually uses. The measurement of use is specified as part of the arrangement. This technique is often referred to as *metering*.
- **Pay-per-user.** This is an arrangement whereby the consumer's software is permitted to use the producer's software for a fixed period of time in return for a fixed fee, independent of whether or how much the consumer's software actually uses the producer's software. This technique is usually known as *licensing*, and sometimes referred to as *pay-per-copy*.

The pay-per-user technique is certainly the dominant one in today's software industry, but the pay-per-use technique is hardly unknown. Pay-per-use is used extensively by utilities (your gas and electricity consumption is metered, and so is most of your telephone use), postal services (through the now widespread postal meters), city services (parking meters), music industry (jukeboxes), etc.

In the information industries, pay-per-use is employed extensively by information database providers (e.g., reference searches, legal databases, medical databases), bulletin board operations, and consumer information providers (e.g., CompuServe, Prodigy).

For component distribution and revenue collection purposes, we submit the following additional requirement:

- f. **Provide Dual Technique.** Both pay-per-use and pay-per-user arrangements must be supported; metering becomes the default mechanism in the absence of any pre-paid license.

We motivate this requirement with an example involving end-users and an application. Similar examples can be constructed to involve software components only.

Suppose that you are the manager of a medium-size software development group, say 30 people. You have 25 software engineers and 5 technical writers. Your platform is a network of Unix workstations. You must purchase an authoring system to be the technical writers' main tool, and also to be used occasionally by the software engineers. Typically, you'll be presented with products based on floating licenses.

You know that your writers will use the system every day, intensively. You also know that your engineers will rarely use it, but once a month they will all want to use it simultaneously to write their status reports to the management. The question is: how many floating licenses should you buy?

If you buy 5, then either no engineer will be able to use it, or they will constantly fight over licenses with the writers. This decision will waste you time, energy, and perhaps even people!

If you buy 30, then everybody will be happy, except for your CFO: you will end up with 25 licenses sitting around unused for most of the time. This decision will waste you quite a bit of money!

If you buy any number between 5 and 30, you'll still waste purchasing money and still have to force people to come to work at odd hours to abide by their licenses.

You don't have to worry about any of the above if the application offers both pay-per-user and pay-per-use. You buy 5 licenses to satisfy the predictable, steady use by the writers. You also pre-pay some amount of use, and have the application run off the meter any time more than 5 people try to use it. If you run out of meter, you call your vendor with your credit card and buy more. Problem solved: save both headaches and money at the same time with a simple but flexible approach.

THE RESOURCE MODEL

This section introduces the resource management conceptual model of metering and licensing. We introduce several new notions, including resource, vendor, account, etc. We also explain in detail the concept of acquiring a resource, and the resource acquisition algorithm.

Resources and Sub-resources

A *resource* is an abstraction representing access to a software component, such as:

- software subsystem (e.g., a math library, a persistence library, etc.), or
- data (e.g., a font family, an encyclopedia entry, a stored movie, etc.), or
- generally, an object or cluster of objects.

The relationship between a resource and the related software component is such that one can only access the component via the associated resource.

Accounts, Vendors and End-Users

Each resource has an *account*, which contains the current balance, measured in *meter units*. If the account has a negative balance, the resource owes units to one or more resources. If the account has a positive balance, the resource is owed units by one or more resources.

Meter units are the currency in which all transactions between resources take place. Meter units are converted to cash when revenue is distributed to or collected from vendors.

Each component and the corresponding resource is owned by a *vendor*. The vendor can be the manufacturer of that component, an agent for the manufacturer, or anyone who has acquired the legal right to sell access to that component.

For accounting purposes, end-users are also represented by resources. Thus, each end-user's resource has an account. This account is debited any time the end-user uses metered resources, and credited any time the end-user purchases more meter units for cash. The *end user* may be a

real person (based on whatever the local system uses for user identifier), budget pseudo-people (virtual users set up simply as means to implement budgeting classifications), or an installation as a whole if finer grain accounting is not required.

Periodically payments are made to the vendors whose resource's accounts had positive balances.

In summary, resources deal in meter units, while vendors (and users) deal in monetary currency (e.g. U.S Dollars, Deutsche Marks, etc.).

The Structure of a Resource

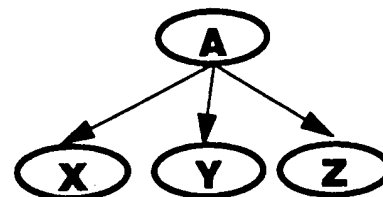
A software component *uses* other software components to implement its functionality. Correspondingly, a resource *uses* other resources, which become its sub-resources. In Figure 1, resource A uses resources X, Y, and Z, which are its sub-resources.

Each resource is identified by a *resource identifier* (or a *rid*). Resource identifiers are universally unique, so that accounting integrity is preserved.

Thus, conceptually, a resource has the following components:

- a resource identifier,

FIGURE 1. A Resource and its Sub-resources



- a set of sub-resources,
- an account with a balance, and
- an associated software component.

Resources are implemented as persistent objects, subject to the same persistence and visibility properties as any other object in the Kala system.

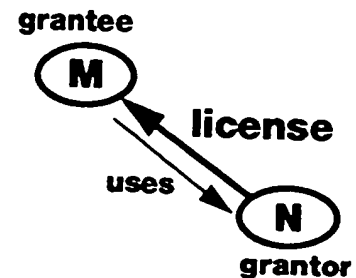
Licensed vs. Metered Use

The use relationship between a resource and any of its sub-resources can be based either on a license or on a metered basis. The subsections below explore in detail each of these two alternatives.

Licensed ("pay-per-user") use

A license is a deal between a grantee resource M and a grantor resource N whereby N grants to M's component access to N's component for a certain period of time, in return for a pre-negotiated license fee (see Figure 2).

FIGURE 2. A licensed use relation between resources



In this context, the *license fee* is the monetary exchange between the vendors that own the two resources M and N, or between an end-user represented by resource M and the vendor that owns resource N.

A license is implemented as a persistent object in the system. The mere existence of a license object implies that the license fee has been paid. The connection between the existence of a license object and the actual payment for the license can be enforced by the software under certain circumstances, explored in more detail in the Section on "The Architecture".

The *license duration* is the time for which a license exists. License durations can be expressed either as elapsed time (measured in days) or as a fixed date, denoting the license's expiration time. A *perpetual license* is a license whose duration is infinite. A *temporary license* is a license whose duration is finite, and usually shorter than the usefulness of the granting component.

Thus, a license consists of:

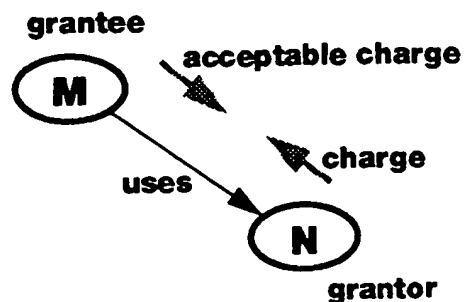
- **Grantor.** This is the rid (resource id) of the resource granting access to its component.
- **Grantee.** This is the rid of the resource whose component gains access to another component. This may be wildcarded, i.e. 'any using resource' or specified by predicate.
- **Duration.** This is the time validity of the license.

A licensed use engenders a fee regardless of whether or not the licensed resource (the grantor) is used or not. Through a license, the grantee gains access to the grantor's associated component without additional charge, but the grantee need not actually access it.

Metered ("pay-per-use") Use

A metered charge is an exchange between a grantee resource M and a grantor resource N taking place at execution time, whereby N grants to M's component access to N's component in return for a metered fee.

FIGURE 3. A metered use relation between resources



As in the licensed case, there is a grantor resource and a grantee resource. However, the relation between them is established at runtime, based on two pieces of information:

- **The potential grantor's provisional charge.** Each resource defines charges (in meter units) that it will ask for if its component is requested. A *provisional charge* is an object that specifies both a potential grantor and a potential grantee, thus allowing different charges to be applied to different requestors. These party specifications can be wildcarded. The model does not specify how the charge is defined, what are the terms, etc. This is a matter of policy, left to the users of the resource manager to negotiate and define.
- **The potential grantee's acceptable provisional charge.** Each resource defines a method by which it decides whether it will accept a charge or not. While the model allows this definition to be associated with a resource, it does not specify how this acceptance should be determined. This is a matter of policy, left to the users of the resource manager to define for their resources.

The metered use involves a runtime provisional charge acceptance step. Once the potential grantee accepts the potential grantor's provisional charge, the access is granted. Thereafter, metering engenders a fee only if the granted resource is actually used. Upon access (use), the resource manager debits metered units from the grantee's account and credits them to the grantor's account.

The transfer is done by the resource manager after completion of all use of the granted component, according an actual charge definition presented by the grantor resource and accepted by the grantee. This actual charge is checked for error and inconsistency against the provisional charge and acceptance is recorded during the resource acquisition phase prior to the grant of access.

The Resource Graph

For both licensed and metered use relationships between resources and their sub-resources, the totality of resources that make up an application form a directed graph, called the *resource graph*. The resource graph is not a tree because some components may be independently used by several different components that enter into the making of an application.

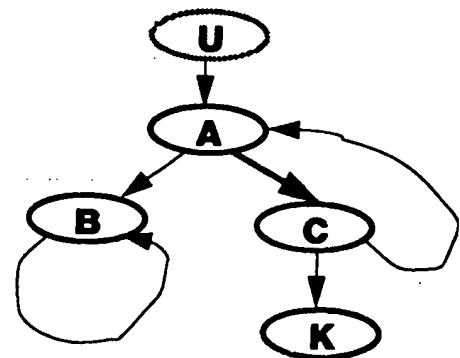
For the purposes of the model detailed in this paper, we assume the resource graph to be static. That is, its structure is determined at a time prior to the execution of the application. For example, this could be at static linking time, or at application definition time.⁴ In other words, for simplicity we assume that the knowledge of all components that could be *potentially* used by an application is present before the application is actually launched.

The model is easily extended to allow for a dynamic resource graph. This provides for the general case in which the set of components *potentially* used by an application is not known at application launch time.

The resource graph has a root node, corresponding to the resource associated with the application. In the example in Figure 4, the root node is A. The set of all nodes in the graph is obtained by the transitive closure of the use relationship.

The example in Figure 4 shows the resource U corresponding to the end-user who runs the application. It also shows B and C having self-pointing arrows. These arrows indicate that both B and C add value beyond the value acquired from their own sub-resources (K in C's case, none in B's case), and therefore their own use is not free.

FIGURE 4. An application resource graph



For an application to run successfully, the top level code (the application's main program) must gain access to all its components, and so on recursively. This translates into the resource A *acquiring* its sub-resources, its sub-resources acquiring their own sub-resources, and so forth until all resources in the resource graph have been acquired.

Acquiring a Resource: The Basic Algorithm

Acquiring a resource is a recursive process, starting from the root of a resource graph and working its way down until all resources have acquired all their sub-resources. There are three kinds of information that are used in the process:

- the resource graph,
- the existing license objects, and
- the provisional charge and charge acceptance definitions associated with each resource.

When an attempt is made to acquire a resource X, the following algorithm is followed:

Step 1: For each of X's sub-resources Y:

- 1.a Attempt to acquire Y.
- 1.b If successful, then go to next sub-resource, if any.
- 1.c If Y requests a charge, and there is a license from Y to X or any parent of X, then acquire Y with license.
- 1.d If Y requests a charge, and there is no license for Y, then accumulate charge.

Step 2: Add local added value to accumulated charge.

Step 3: If X's parent has a license for X, then

- 3.a Accept provisional charges from all X's sub-resources, and remember to pay all charges to X's sub-resources from X's account.
- 3.b Return success.

Step 4: If X's parent has no license for X, then propose to charge accumulated charge to parent.

Step 5: If X's parent accepts provisional charge, then

- 5.a Accept provisional charges from all X's sub-resources, and remember to pay all charges to X's sub-resources from the actual charge received from X's parent.
- 5.b Return success.

Step 6: If X's parent refuses the provisional charge, then return failure.

As the algorithm shows, negotiations take place between a resource and each of its sub-resources. The negotiations take place entirely outside the resource manager, which is totally unaware of the nature, methods, and means of these negotiations. A negotiation may be a complex dialog between the two resources, or may be empty (no negotiation at all).

However determined, the result of the negotiation is communicated to the resource manager by both parties. The potential grantor communicates to the resource manager a *provisional charge* (in Step 4), while the potential grantee communicates a *provisional acceptable charge* (in Steps 3a and 5a).

The resource is able to acquire the sub-resource if it either has a license for it or is ready to accept metered charges, based on the provisional description of the charges presented by the sub-resource. If neither of these happens, the resource is unable to acquire the sub-resource, and the entire algorithm fails, all the way up to the root of the resource graph. In other words, an application can execute if and only if it is able to acquire all resources in its resource graph.⁵

When proposing or accepting a provisional charge, the resources inform the resource manager of the fact by supplying a pair of two numbers (a range). The lower number represents the *provisional minimum charge*: if the provisional charge is accepted then the grantor resource's account will be credited at least that many meter units, whether or not the grantee accepts the final actual charge (see Section on

“Charging and Disbursing”). The provisional minimum charge may be zero, but not negative.

The upper number represents the *provisional maximum charge*. It means that the grantor resource’s account will be credited no more than that many meter units (see Section on “Charging and Disbursing”). If the final actual charge exceeds the provisional maximum charge, the resource manager considers this an error (a sign of potential run-away charges), credits the grantor’s account with only the provisional minimum charge, and refuses to acquire the grantor resource for the grantee resource until the bug is fixed. The provisional maximum charge may be infinity, but not smaller than the provisional minimum charge.

Charging and Disbursing

After having successfully acquired all the resources it needs, an application can now execute. As it runs, some resources are actually used. Some resources may never be used.

As they are used, those components that have been acquired on a metered basis (as opposed to a licensed basis) tally their running charges *using their internal algorithms and internal data structures to hold the running tallies*. At the same time, those components which acquired other components on a metered basis may also keep a tally of the charges they are expecting to eventually receive from those components based on the actual pattern of usage. This tally too is performed entirely by internal algorithms, possibly based upon information about the grantor resources which was obtained during the negotiation phase.

At the end of the application execution, the grantor resource presents the accumulated *actual charge* tally and the grantee resource presents the accumulated expected charge tally (as an actual acceptable charge) to the resource manager, which compares them to each other and against the provisional charges specified by the grantor and provisional limits accepted by the grantee resource.

If the actual charges conform to the agreement represented by the agreed provisional charges, the resource manager transfers the amount of metered units actually charged from the account of the grantee resource to the account of the grantor resource. The same process occurs for all components which were provided on a metered basis. These transfers are known as *actual charges*. They may include charges to the end-user’s account.

The tally by the grantee of expected actual charges is intended to provide a check on the veracity of the grantor beyond the rather broad limits of the provisional agreement. The grantee presents the expected actual charge to the resource manager in the form of a range, and so need not be exact in its calculation of the expected actual charges. Indeed if the grantor is trusted, the grantee may omit the expected tally altogether and present the resource manager with an expected actual charge range of zero to infinity, effectively taking the grantor’s word for the actual charges.

Accounting

Periodically, the resource accounts must be converted into cash, so that the corresponding vendors can be paid cash for the use of their

components. This can be done at specific times (such as whenever end-users refill their own resource accounts), or regularly (for example, on a quarterly basis).

The *resource accounts conversion* is an activity of summarizing the balances of all resources in a resource manager installation, communicating the summary to the agency that does the conversion (the equivalent of the bank), the agency paying the corresponding vendors the equivalent amounts of cash, and finally resetting the paid accounts to zero, ready for the next cycle.

The resource accounts conversion can be carried out manually or mechanically. A manual process involves running a batch program at the installation site, which will create an accounting dump file; sending the accounting dump file to the metering agency (the resource manager vendor); and finally reinitializing vendor account balances at the site.

The same activity can be carried out entirely mechanically over a modem or other transmission line. The activity can be manually started or could even be started automatically (by the resource manager itself), thus making it largely transparent to the end users.

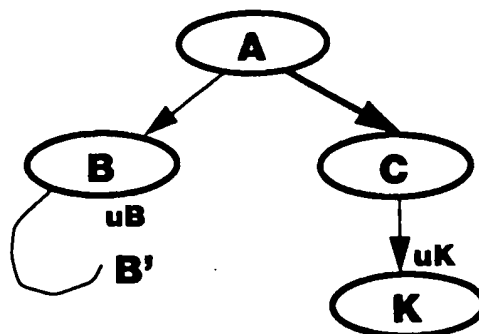
A RESOURCE ACQUISITION EXAMPLE

To provide a simple example, this section traces the execution of the basic resource acquisition algorithm presented in Section on "Acquiring a Resource: The Basic Algorithm".

The Resource Graph

The example involves four resources: A (the root resource, likely standing for the application), B, C, and K. The resource graph is shown in Figure 5.

FIGURE 5. A sample resource graph



Here, resource A uses resources B and C, and resource C uses resource K. Resource B does not use anything else, but adds its own value B'.

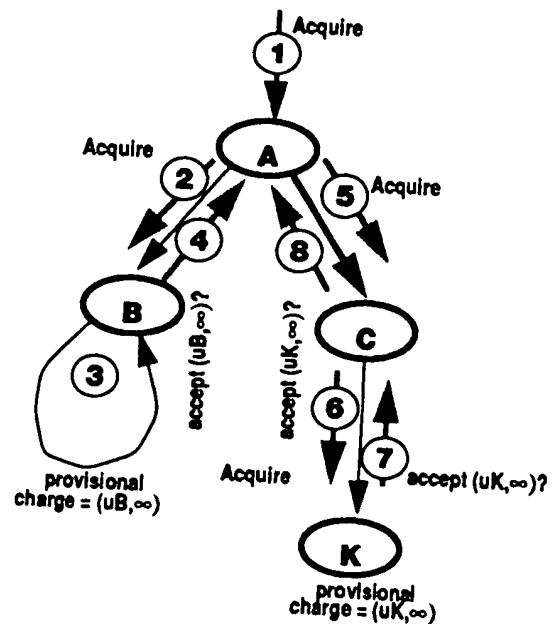
We also assume that A has a license for C, and that there are no other licenses. Let's assume that K submits a provisional charge with a minimum of uK meter units and a maximum of infinity (with the actual charge computed on usage), and that B' proposes a provisional charge with a minimum of uB units and a maximum of uB units as well (fixed fee).

Tracing Through a Resource Acquisition

The execution of the algorithm in Section on "Acquiring a Resource: The Basic Algorithm" to acquire resource A entails the following sequence of events:

1. The end-user attempts to run the application, i.e. the end user's resource attempts to acquire the application's resource A.
2. A attempts to acquire B (cf. A's step 1).
3. B has no sub-resources, but has a local charge of a minimum uB meter units (cf. B's step 2).
4. B proposes the provisional charge (uB, uB) back to A (cf. B's step 4). This provisional charge is accumulated by A (cf. A's step 1d).
5. A attempts to acquire C (cf. A's step 1a).
6. C attempts to acquire K (cf. C's step 1a).
7. K has no sub-resources, but has a provisional local charge of (uK, ∞) which it proposes to C (cf. K's step 4).
8. C has accumulated provisional charges of $\{(uK, \infty)\}$ (cf. C's steps 1 and 2), and proposes them up to A.

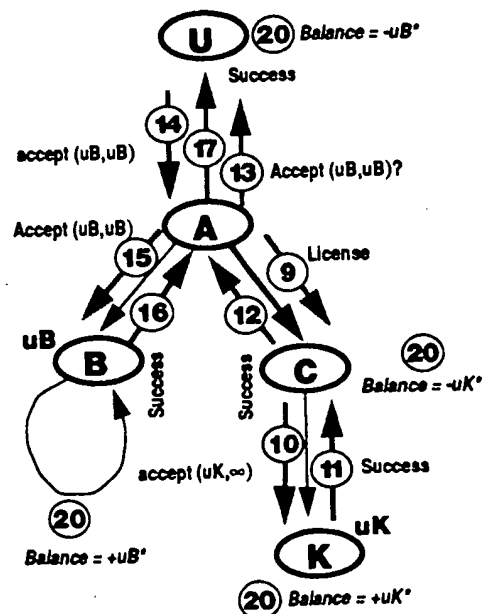
FIGURE 6. Acquiring resource A (first 8 steps)



9. A has a license for C, and presents it in response to C's provisional charge (cf. A's step 1c).
10. C accepts K's provisional charge, but must commit to pay out of its own account (cf. C's step 3a).
11. Since C accepted K's provisional charge, K returns success (cf. K's step 5b).
12. C has successfully acquired all its sub-resources, and it returns success (cf. C's step 3b).

13. A still has an accumulated provisional charge (from B) of $\{(uB, uB)\}$, and it asks the end-user resource whether it accepts this provisional charge (cf. A's step 4).
14. The end-user resource accepts, either by silent pre-specified provisional acceptance of the charges or interactively by responding "Yes" to the "Accept charges?" dialog box (cf. A's step 5).
15. A accepts B's provisional charge of (uB, uB) meter units (cf. A's step 5a).
16. B receives the acceptance to its provisional charge and returns success (cf. B's step 5b).
17. A has acquired all its sub-resources, and it returns success (cf. A's step 5b).
18. All components of the resource graph have returned success, and so their respective components have been acquired. A proceeds with execution of its associated component.
19. During execution, K internally accumulates the actual charges for the actual use made of its component (as opposed to the provisional charges used during resource acquisition). B need not do this as it is using flat fee charging.

FIGURE 7. Acquiring resource A (remaining steps)



20. At completion of execution (signaled by A) B and K present their final bills based on actual usage to the resource manager. B presents a uB^* actual charge equal to the uB provisional. K presents a uK^* actual charge, greater than or equal to the provisional charge uK .
21. The user resource, which knows that it is expecting a uB charge from B, presents an acceptance of a $\{(uB, uB)\}$ actual charge to the resource manager. C, which has been keeping rough track of

the use it has been making of K, presents an acceptance of a $\{uK, uK\}$ actual charge to the Resource Manager, where uK is somewhat larger than the uK^* computed by K but low enough to catch any error or cheating by K.

22. After verification, the Resource Manager debits uB^* units from the user (resource U) and credits them to B, and uK^* units from C and credits them to K. The transaction is complete.

Resulting Charges

Since the charge contracts between A and B, and between C and K involve metered usage, the balances of the U (end-user), B, C, and K accounts are modified:

- U's balance was debited uB^* meter units, to pay for the use of B (there was no license for B, nor was there one for any of its ancestors up the chain to U).
- B's balance was credited uB^* units.
- C's balance was debited uK^* meter units, to pay for the use of K (C had to pay this itself, because A had a license for C, and so C could not ask A for payment).
- K's account was credited uK^* units.

THE ARCHITECTURE

This section provides an overview of the architecture supporting the main activities of the resource model.

Three Activities

The resource management model involves three distinct activities and related sets of objects:

- **The Definition Activity.** This activity creates and modifies resource and license objects. This activity is described in detail in the Section on "The Definition Activity".
- **The Resource Acquisition and Charging Activity.** This activity takes place every time an application executes for an end-user. This activity is described in detail in the Section on "The Resource Model".
- **The Accounting Activity.** This activity takes place periodically, both at each installation and at the account agency site. This activity is described in the Section on "The Accounting Activity".

The Definition Activity

Definition is the activity by which resources are defined, licenses are installed, and meters (resource accounts) are filled with meter currency. While the times and frequency of the resource definitions differ from those of installing licenses and filling up meters, the notions are related and share several common properties:

- They all involve interactions between a *customer*, using one or more applications based on Kala's resource manager, and a *vendor*,

able to provide the customer with both the application and related licenses and meter currency.

Note that by "vendor" we don't necessarily mean a vendor of software — ultimately, it is not software that's being sold here. By vendor, we mean any agency that has acquired the legal right to sell and administer licenses and meter currency for various components and applications. This may be a software distributor, a software manufacturer, an Independent Software Vendor (ISV), an independent licensing and metering authority, or the manufacturers of the resource management software itself.

- They all reflect cash transactions between the customer and the vendor. The cash exchanges don't necessarily have to mimic the definition actions. However, each such definition action is related to either a past, a concurrent, or a future cash exchange between the vendor and the customer.
- As a consequence of the previous item, they all must be totally safe, thus offering trusted protection against fraud. The Kala Resource Manager insures this safety through a cookie exchange protocol described below.

The Cookie Exchange Protocol

The "Cookie Exchange Protocol" is a mechanism designed to insure the safe and unforgeable definition of resources, installation of licenses and filling of meters at the customer sites. The mechanism is based on the exchange of "magic numbers" between vendors and customers. These magic numbers are also known as *cookies*.

For each definition action, two cookies are involved:

- a. The Customer Cookie.** This cookie is generated by the customer on the customer computer, and transmitted (either manually or mechanically) to the vendor. The customer generates the customer cookie using the Cookie utility program or a version of it embedded in the application itself. The customer cookie encodes information that makes it unique, reflecting this particular customer site (specifically, this particular Kala-based resource manager instance). The customer cookie identifies the customer uniquely throughout the subsequent activity.
- b. The Vendor Cookie.** This cookie is generated by the vendor in response to a customer cookie and to information the vendor has about the associated cash transaction (such as whether the customer's check cleared or the credit card transaction went through). The vendor cookie encodes the originating customer cookie, the vendor's identity, and the action to be performed on the basis of customer payment. It is generated by the vendor, using a unique per-vendor copy of the NewCookie program. It is passed back to the customer, who uses it to perform the definition action.

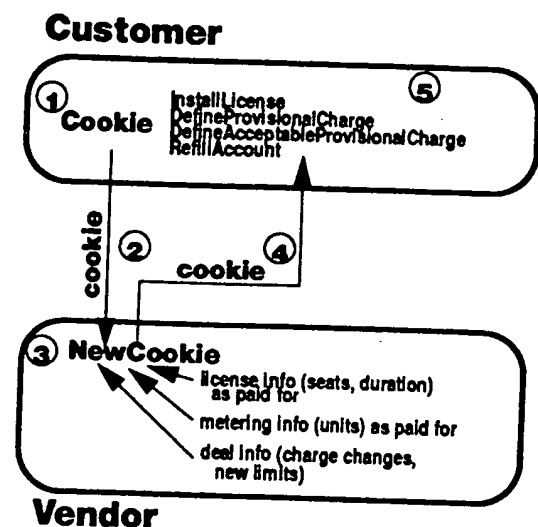
The cookie exchange protocol is safe against fraud as long as:

- The vendor's NewCookie program is safeguarded by the vendor. Since the vendor has material interests in preventing fraud, we

assume that it will take good precautionary measures to insure the program's safety (for example, installing it only on a physically protected, standalone (un-networked) machine.

- The customer's site is able to generate customer cookies that uniquely identify the site. Satisfaction of this requirement is guaranteed by the mechanism Kala uses to guarantee universal uniqueness of identifiers [6].
- Valid cookies from NewCookie only work on sites which generated the original customer cookie.
- Invalid cookies don't work at all.

FIGURE 8. The cookie information flow



Since the customer cookie is not useful to anyone other than the vendor who generates a vendor cookie with it, and the vendor cookie is useful only to the originator of the customer cookie the vendor cookie was generated from, cookies can be transmitted safely over unsafe communication media, such as regular electronic mail or voice telephone.

The application developer may embed the customer cookie generation in the application itself, so it can benefit from the already existing Graphical User Interface (GUI), communication facilities, etc. This is done by calling the Kala Cookie API function, which returns a datum of type *cookie*. The datum can then be either displayed to the end-user (for manual communication to the vendor), or embedded into a message silently sent to the vendor via some communication link (e.g., modem connection, electronic mail, etc.).

Defining Resources

The sub-activity of defining resources is part of the process of "installing" the software on the customer's computer from the delivery medium (e.g., diskettes, tapes, network, CDs, etc.). A resource is defined in the Resource Manager for each installed component.

To define a resource, the installation program calls `DefineResource`, a function that is part of Kala's API. For example, the following defines the `K` in the example resource graph in the Section on "The Resource Graph". This resource has a resource identifier `K`, no sub-resources, and represents some given component. The definition is controlled by a vendor-supplied `magicNumber`.

```
rid K = ...;
cookie magicNumberK = ...;
DefineResource (K, nilRow,
               magicNumber, component, state);
```

The following code fragment defines the `C` resource in the same example in the Section on "The Resource Graph":

```
rid C = ...;
cookie magicNumberC = ...;
DefineResource (C, Only (K, 1),
               magicNumberC, componentC,
               stateC);
```

Defining Charges and Acceptable Charges

Once a resource is defined, one can define the provisional charge the resource would present to a potential grantee (another resource attempting to acquire this resource) using the `DefineProvisionalCharge` function. For example, the following defines the charge presented by the `K` resource defined above (see the section on "Defining Resources"):

```
span uK = ...;
DefineProvisionalCharge (K,
                        Range (uK, maxInt),
                        magicNumberK);
```

Note that `DefineProvisionalCharge` (like `DefineResource`) is controlled by a vendor cookie. The same is true for the next function, `DefineProvisionalAcceptableCharge`.

This defines an acceptable charge that a grantee (using) resource can accept when charged back by any of its sub-resources:

```
DefineProvisionalAcceptableCharge (
    C, K, Range(uK, maxInt),
    magicNumberC);
```

Installing Licenses and Refilling Meters

Finally, an application's code can install licenses or refill meters using two additional Kala API functions: `InstallLicense` and `RefillAccount`. They both require a vendor cookie to operate.

For example, the following installs 2 one-year licenses of resource `C` to resource `A`:

```
rid A = ..., C = ...;
InstallLicense (A, C, 2, 365,
               magicNumberC);
```

To refill `U`'s account with 100,000 meter units, you call:

```
rid U = ...;
cookie magicNumberU = ...;
/* from meter vendor */
RefillAccount (U, 100000,
               magicNumberU);
```

The Accounting Activity

The accounting activity has three parts; the first and the last take place at the customer site, and the middle takes place at the accounting vendor's site:

- a. **Balance Information Collection.** This activity takes place periodically. It consists of (i) summarizing the account balances (their magnitude, not the actual units or money) for all resources defined at the customer site, and (ii) communicating this data to the accounting vendor. The communication can be either manual (via paper, removable magnetic media, etc.) or mechanized (via modem and telephone lines, high-speed network, etc.).
- b. **Accounting.** This activity takes place at the accounting vendor site. The balance for each resource is merged with the corresponding accounts for the same resource from other customers, and those with net positive balances trigger cash payments to the corresponding vendors.
- c. **Balance Reinitialization.** Upon successful delivery of balance summary information (per step a. above), the balance of each resource account at the customer site is brought to zero.

The Resource Acquisition and Charging Activity

This activity was described in detail in the Section on "Acquiring a Resource: The Basic Algorithm". Here, we need only mention two Kala API functions called in the process. The first is the function that starts the resource acquisition activity: `AcquireResource`. This function is called by the application as part of its initialization. For example, if the application's resource id is `r`, the following call acquires the application component on behalf of the end-user:

```
p = AcquireResource (r,
                    ResourceOfClient (myCid), 1);
```

The call above makes use of the second relevant Kala API function: `ResourceOfClient`. Given a client identifier (for example, `myCid`, the well known wildcard identifier of the calling client process itself), `ResourceOfClient` returns the resource identifier associated with that client.

If resource acquisition (and all sub-acquisition) was successful, `AcquireResource` returns a pointer to the component associated with the acquired resource. The Resource Manager automatically loads the component from the persistent store. If the resource acquisition activity fails, the pointer is `nil`. Because the components are kept on protected and inaccessible persistent store by Kala, the application has no way to get at any of the components other than through a successful `AcquireResource`.

COMMON BUSINESS MODELS

The resource management model and its implementation as part of the Kala technology is useful in practice only as long as it is able to support useful and desired business models. A good test is to explore its support for a few simple ones in wide use.

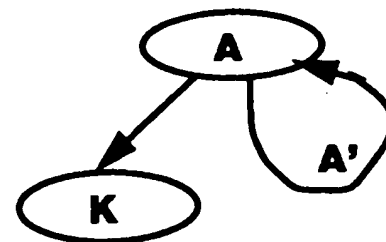
While this section only explores a few simple models, many more can be implemented atop Kala's resource management primitives, opening the doors to creative business deals that are both mechanically and legally enforceable, and inexpensive to administer.

Perpetual Floating Licensing

Contemporary software implemented on network computers commonly employs perpetual floating license schemes. These schemes allow up to a specific number of users to use the application concurrently. The number can be increased by purchasing more licenses. Once purchased, a license never expires. If a site has N licenses and the $(N+1)^{\text{th}}$ user attempts to access the application, she gets a message informing her that the system is temporarily out of licenses, and that she needs to wait until one of the current users logs off this application.

Such a perpetual floating licensing scheme can be implemented trivially using Kala's resource management mechanism. Perpetual licenses are implemented as Kala licenses with infinite durations (represented as `maxInt` values). For each application (or component) subject to such a licensing scheme, a resource will be created as part of that application's (or component's) installation.

FIGURE 9. Perpetual Floating Licensing Example



For example, let's assume an application that uses Kala to store its persistent data. The application uses no other component that is subject to resource management. The resource graph is shown in Figure 9. **A** is the resource associated with the application, and **K** is the resource associated with Kala itself, viewed as a software subassembly.

We assume that Kala's own installation (the `coldKala` program) installs the **K** resource. The application's own installation program contains the following code fragment:

```

rid A = ...;
mid midOfAnEntryPoint = ...;
DefineResource (A,
    Only (Account (K, 0)),
    receivedCookie,
    midOfAnEntryPoint, nilMid);
  
```

In the fragment above, `midOfAnEntryPoint` is the identifier of application **A**'s entry point segment, to be loaded into memory from the persistent store and branched to for execution upon a successful acquisition of **A**'s resource.

The application has a user interface (e.g., part of its GUI) that provides the application administrator the means to:

- generate customer cookies to request additional floating licenses,
- add new (paid for) licenses, using vendor cookies, communicated either by telephone (voice), or mechanically via some form of connection between the application site and the vendor's site (e.g., modem connection over regular phone lines), and
- inquire about the status of licenses and meters, such as how many licenses are currently installed, etc.).

The application can run either with or without a license for Kala. In the former case, no further exchange takes place between A and K. In the latter case, the meter units consumed by application execution will be debited from A's account and credited to K's account when application execution ceases.

A typical user site scenario is:

- The application administrator (likely the same person who administers networks, etc.) clicks on the application's menu item reading "Add another license".
- The application presents the administrator with a dialog box. The application administrator fills out the quantity desired (defaulting to 1) and the credit card number and expiration date (no default, for privacy reasons). Then he clicks on the OK button.
- The application silently sends electronic mail to the vendor, containing the above information and a locally generated client cookie.
- The vendor silently responds with a vendor cookie.
- The application receives the electronic mail from the vendor and issues a call to

```
DefineLicense (A, installation, 1,
               forever, vendorCookieA);
```

to install the license to A.

- If the agreement between A's vendor and Kala's vendor specified that each sale of a license for A will also include a sale of a license for Kala (one of the many possible arrangements), then a license to Kala is also installed:

```
DefineLicense (K, A, 1, forever,
               vendorCookieK);
```

Note that a perfectly valid alternative is to assume that licenses for Kala are obtained through a completely separate channel. For example, Kala may already be installed on that network computer in support of other applications, and blanket licenses (that is, licenses to Kala for anyone who needs them) may already exist.

- The application sends e-mail to the application administrator, notifying him that the license(s) have been successfully installed.

There are many variations and extensions of the perpetual floating license scheme, both with respect to what is being licensed and to how the licenses are administered. The example above suggests a more mechanized, transparent and easy-to-use approach, involving a minimum amount of effort on the customer's side and a minimum amount of labor on the vendor's side. Indeed, a win-win situation.

Flat Annual Royalty Arrangements

Another commonly practiced licensing scheme is that of a flat annual royalty. In this scheme, if an application A uses a component B, A's vendor obtains the permission to dispense an unlimited number of licenses to B in return for a flat annual fee. Other legal limitations may occur, and some may be enforceable through software. However, we will ignore them here for simplicity. We also assume that A's vendor sells its application under a perpetual license agreement.

When A's vendor pays B's vendor the agreed-upon annual fee, B's vendor gives A's vendor a NewCookie module that generates cookies for 1-year licenses from B to A. A's vendor delivers the B upgrade module to its own customers as part of an annual upgrade of the A application. The B upgrade module silently installs up-to-date licenses to B, so A's customers can continue to use the embedded B without charge (they may actually not be even aware of the existence of B!).

This scheme assumes an explicit module upgrade. While these upgrades of B-to-A licenses are not relevant to A's customers, they can be hidden inside other kinds of upgrades of A, such as annual A software releases, etc.

If the customer does not install the upgrade, A will continue to run even though the local (sub)license from B to A has expired, so long as the B component was written to fall back to pay-per-use. Absent a license, B will run off a meter so long as the end-user's resource accepts the provisional and actual charges. This inconvenience can be avoided by providing the customer with strong enough incentives to upgrade.

If A's vendor (who has sold perpetual licenses to its customers) fails to buy and deliver to its customer the necessary B module upgrade (so that the customer can use B as part of A without charge), then the customer has the same legal complaint against A as for any other failure to deliver contracted upgrades. This presents an incentive to A's vendor to provide the necessary upgrades.

As in the example in Section on "Perpetual Floating Licensing", the process can be made quite smooth by employing silent transfer mechanisms such as electronic mail or direct connect via modems over phone lines. Since most sites (organizational and residential) are now equipped with such devices, and since the amount of data to be transferred is fairly modest, these options are now more practical than ever.

Metered Use of Kala as a Repository

Another simple application occurs where passive objects are provided under a metered arrangement. For example, a vendor of clip art may place an entire clip art collection on a Kala-managed CD-ROM (see [7],[8] for more information on Kala's persistent store functionality).

The vendor distributes the collection for no or low cost (perhaps enough to cover manufacturing costs in part). The arrangement is that each clip is paid for on a usage basis: a small amount is charged every time a drawing program loads a clip and inserts it into a drawing.

The drawing program is implemented so that it cannot be used to make any further copies of the clip. Kala's Resource Manager doesn't really enforce this if the clip is to be truly passive, i.e. usable by a lot of applications. The application must have an internal (memory) representa-

tion for the art clip, and Kala has no means to prevent the application from writing out the internal representation and reusing it. Here, the actual protection comes from the fact that commercial drawing package developers will have no incentive to allow such loopholes — it is in their best interest to protect the economic interest of clip artists, so that they can continue to supply them with high quality clip art.

An individual programmer may find ways to break this protection; however, if the cost of clips is low enough, it would be too much trouble for what it was worth. Pricing strategy plays an important role here. The vendor of a clip art library should price the product such that if every clip was referenced once, the total meter credit would be more than what he would sell an unlimited license for anyway.

In the implementation, Kala holds clips as persistent data. Access to these clips is securely controlled using Kala's data visibility functionality. Drawings using these clips can also be held as Kala data.

Each clip has a resource object associated with it. Each "handle" to a clip is set up to acquire the clip's associated resource. The model supports embedded clips as well.

The end-user (or the site, if accounting is on a site basis) periodically fills up his account with some quantity of meter currency. The end-user resource accepts charges as long as its account balance is positive.⁶

When a clip needs to be loaded into a drawing, if the proposed charge is accepted by the end-user resource, the actual charge is credited to the clip's resource account. Periodically (for example, on a quarterly basis or when the customer purchases more meter currency), the accumulated accounting information is sent to the vendor.

Clip art may be supplied by many artists. The clip art vendor can, using the detailed accounting data received from the customer, distribute the proportional revenues to the clip artists. Thus, the small but highly skilled contributor can get actual revenue from his or her work, without placing an excessive burden on the revenue collection system — a simple accounting computation. The underlying mechanism is very similar to the one ASCAP uses to convey a portion of each coin in a jukebox to the author of the song played.

Free (Unlicensed) Use

A degenerated use of the model occurs when a component is offered for free use to anyone who needs it. In this case, the code fragment that installs such a "nil license" for a resource X is:

```
DefineLicense (X, anyone, maxInt,
               forever, aCookie);
```

Here, a virtually unlimited number of users can use component X indefinitely.

SUMMARY OF THE API

This section summarizes Kala's resource management interface. The interface (API) is shown as C function declarations, although other language interfaces will be supported.

Resources, Relationships and Accounts

DefineResource defines a resource and its "use" relationship to a suite of other resources, expressed as a row of resource. It associates a resource with a component.

```
void
DefineResource (rid resource,
                rowRid subResource,
                cookie magicNumber,
                mid component,
                mid state);
```

InstallLicense defines one or more pay-per-user (license) relationships between a grantor resource and a grantee resource, based on a cash payment. The license is for a given number of seats, and has a given time validity, measured in days.

```
void
InstallLicense (rid grantor,
                rid grantee,
                span seats,
                duration validity,
                cookie magicNumber);
```

RefillAccount credits a resource's account with a given number of meter units, based on a cash payment.

```
void
RefillAccount (rid resource,
               span units,
               cookie magicNumber);
```

Provisional and Actual Charges

DefineProvisionalCharge states that a resource can allow access to its associated component (see **DefineResource** in the Section on "Resources, Relationships and Accounts") in return for an expected charge. A provisional charge is defined to be a range, indicating a minimum and a maximum (see the Section on "Resources and Sub-resources").

```
void
DefineProvisionalCharge
    (rid grantor,
     rid grantee,
     urange charge,
     cookie magicNumber);
```

DefineAcceptableProvisionalCharge states that a grantee resource is willing to accept an indicated provisional charge.

```
void
DefineAcceptableProvisionalCharge
    (rid grantor,
     rid grantee,
     urange charge,
     cookie magicNumber);
```

Charge allows a grantor resource to present a grantee resource with a given actual charge, measured in meter units.

```
void
Charge (rid grantor,
        rid grantee,
        span units);
```

AcceptCharge allows a grantee resource to accept an actual charge from a grantor resource. The accepted actual charge is given as a min/max urange (range of unsigned integers), so that the match does not have to be exact.

```
void
AcceptCharge (rid grantor,
              rid grantee,
              urange charge);
```

Acquiring Resources

AcquireResource acquires a resource for a potential grantee resource (the caller of this function). If successful, it secures access to the resource's component for the grantee's component. It can acquire the resource a quantity number of times. If successful, it returns a pointer to the grantor component, now in memory.

```
pointer
AcquireResource (rid resource,
                 rid grantee,
                 span quantity);
```

ResourceOfClient returns a client's associated resource, expressed by its identifier (rid). The client is identified by its client unique identifier (cid).

```
rid
ResourceOfClient (cid client);
```

Cookies

Cookie generates a new local installation cookie, to be passed to the vendor, along with some request and cash payment.

```
cookie
Cookie (void);
```

Accounting

CreateAccountingSummary computes the summary of accounts in an internal format and uses Kala facilities to create a new Kala persistent datum to hold the summary. The newly created datum is pointed to by a Kala handle located at <kin, basket>, using usual Kala addressing [7]. The resulting datum can thereafter be copied to a file, sent to the vendor site via electronic mail, or moved between Kala installations using regular Kala facilities.

```
mid
CreateAccountingSummary (kid kin,
                        bid basket);
```

CONCLUSIONS

In the Section on "Revenue Collection" we outlined a few requirements for a practical revenue collection mechanism independent of the software distribution mechanism. To conclude this brief overview of Kala's

resource management functionality, we are reviewing how Kala meets these requirements.

- a. **Be safe.** Kala meets this requirement by a combination of several factors: (i) the resource model requires components to be accessible *only* via their resource objects; (ii) Kala provides a secure storage of data, whereby applications have full control over who can see what and when, via the data visibility primitives (see [7] for a complete discussion of Kala's main data visibility mechanism, the Kala Basket); and (iii) the dual cookie device permits rights to be communicated between vendors and customers without the need for expensive or cumbersome communication safety provisions.
- b. **Be recursive.** Kala's resource management model meets this requirement by recognizing that, with respect to licensing or metering, arbitrary components are no different from applications which are no different from passive data. The model explicitly calls for resource graphs.
- c. **Be flexible.** Kala's model does not impose any policies. It imposes no practical restrictions on how the business deal is structured, how the licenses or the meters are administered, what mechanism is used to communicate between vendors and customers, etc. As long as a policy does not violate the other requirements discussed here (for example, the safety requirement), it is implementable using Kala's resource management primitives. This feature sharply distinguishes Kala from all other license managers (commercial or research). The immediate effect is simplicity and interface economy.
- d. **Be efficient.** Kala's model was designed to minimize the number of communications between the application/component code and the resource manager. The result is an implementation that brings very little overhead, unnoticeable in practice.
- e. **Be invisible.** Kala's toolkit approach makes it possible to fully integrate all license and meter management with the application (and under some schemes even hide them inside).
- f. **Provide Dual Techniques.** Kala provides support for both the "pay-per-user" and "pay-per-use" approaches, and does so by linking them together, so that "pay-per-use" becomes the default in the absence of available licenses.

A restricted version of this model has been part of the Kala persistent data server product since its 2.2 α version. This model is part of Kala's 3.x version. Extensions to it are also expected.

NOTES

1. Kala is a Trademark of Penobscot Development Corporation. Portions of the technology described herein are covered by pending US and international patents.

2. Many have pointed out that there is more software and information being used without payment than there is legitimate use.

3. We are, however, encouraging everyone to explore these topics and devise ways to overcome these serious barriers.
4. Application definition with respect to the component repository, i.e., at application installation.
5. This restriction can be relaxed in the case of a dynamic resource graph.
6. It is possible to allow account balances to become negative. This corresponds to extending some amount of credit to a customer, much as a charge account bank would do.

BIBLIOGRAPHY

- [1] Cox, Brad, *What if There Is A Silver Bullet*, Journal of Object Oriented Programming, June 1992.
- [2] Hemnes, Thomas M.S., Esq., *Software Revenue Generation in Network Environments*, Ropes and Gray, Massachusetts Computer Software Council Annual Legal Update Program, November 1992.
- [3] Miller, Mark, *On Software Pay-Per-Use*, private conversations, June 1992 and January 1993.
- [4] Mori, Ryoichi and Maraji Kawahara, *Superdistribution: An Overview and The Current Status*, Technical Reports of the Institute of Electronics, Information, and Communication Engineers, Volume 89, Number 44.
- [5] Penobscot Development Corporation, *The Kala Archives*, available via anonymous ftp from world.std.com at ~ftp/pub/kala. Send mail to kala-request@world.std.com for more information.
- [6] Penobscot Development Corporation, *The Kala Forum*, by free on-line subscription only. Send requests to kala-request@world.std.com.
- [7] Simmel, Sergiu S. and Ivan Godard, *The Kala Basket — A Semantic Primitive Unifying Object Transactions, Access Control, Versions, and Configurations*, Proceedings of OOPSLA'91, October 1991, pp. 240-261.
- [8] Simmel, Sergiu S. and Ivan Godard, *Objects of Substance*, BYTE, December 1992, Volume 14, Number 17, pp. 167-170.
- [9] Simmel, Sergiu, *Software Licensing and Metering with Kala — Infrastructure for a New Economics of Software*, Hotline on Object-Oriented Technology, Volume 3, Number 4, pp. 3-7.

GLOSSARY OF TERMS

account — constituent of a resource, holding a balance of meter units

actual charge — the amount of meter units a grantor resource asks a grantee resource to pay for its actual use of the grantor's services during an execution

actual acceptable charge — the amount of meter units a grantee resource expects or is willing to be charged by a grantor resource for its actual use of the grantor's services during an execution

- component** — small-grain constituent of a software system
- customer cookie** — magic number generated by the customer on the customer computer, and transmitted (either manually or mechanically) to the vendor
- definition** — activity by which resources are defined, licenses are installed, and meters (resource accounts) are filled with meter currency
- distribution** — mechanism by which a component is made available to another component or final consumer (end-user)
- end-user** — real person (based on whatever the local system uses for user identifier), budget pseudo-people, or an installation as a whole if finer grain accounting is not required
- license** — deal between a grantee resource M and a grantor resource N whereby N grants to M's component access to N's component for a certain period of time, in return for a pre-negotiated license fee
- license fee** — monetary exchange between the vendors that own the two resources M and N, or between an end-user represented by resource M and the vendor that owns resource N
- license duration** — time for which a license exists
- licensing** — see *pay-per-user*
- meter units** — currency in which all transactions between resources take place
- metering** — see *pay-per-use*
- pay-per-use** — arrangement whereby the consumer pays the producer for as much of the producer's software as the consumer's software actually uses
- pay-per-user** — arrangement whereby the consumer's software is permitted to use the producer's software for a fixed period of time in return for a fixed fee, independent of whether or how much the consumer's software actually uses the producer's software
- provisional charge** — a formulation of the amount of meter units a grantor resource may charge if the resource is used
- provisional acceptable charge** — a formulation of the amount of meter units a resource M would accept to be charged for the use of another resource N
- resource** — abstraction representing access to a software component
- resource accounts conversion** — activity of summarizing the balances of all resources in a Resource Manager installation, communicating the summary to the agency that does the conversion (the equivalent of the bank), paying the corresponding vendors the equivalent amounts of cash, and finally resetting the paid accounts to zero, ready for the next cycle
- revenue collection** — mechanism by which payment for a component is made to reach the producing vendor, regardless of the context in which the component is actually used
- resource graph** — the graph of resources related by the user relationship, with the application's resource as the entry node (root)

rid — resource identifier

subassembly — medium-grain constituent of a software system

vendor — manufacturer of a component, an agent for the manufacturer, or anyone who has acquired the legal right to sell access to a component

vendor cookie — magic number generated by the vendor in response to a customer cookie and to information the vendor has about the associated cash transaction (such as whether the customer's check cleared or the credit card transaction went through)

BIOGRAPHY

Sergiu S. Simmel, President and Co-Founder of Penobscot Development Corporation, has been involved in the KALA project since 1987. He holds a Master's Degree in Computer and Information Sciences from University of Minnesota. He has been working in the computer industry as a software engineer and technical manager since 1981. His areas of expertise include CASE systems, hypermedia document management, and object oriented databases.

Ivan Godard, Chief Technologist, is Kala's main designer and implementor, being the main force of the KALA project throughout. Mr. Godard has been involved in the computing industry since 1968 as a scientist, engineer, consultant, and entrepreneur. He contributed to the Aldo168 Revised Report and the design of the Ada Language (the "green" version). His areas of expertise include language design, translation technologies, and object oriented databases. Mr. Godard has taught computer science at graduate and post-graduate level at several universities including Carnegie-Mellon University and University of Maine.

The authors can be reached at:

Penobscot Development Corporation
One Kendall Square, Building 200, Suite 2200
Cambridge, MA 02139-1564
voice: (617) 267-KALA
fax: (617) 859-9597
infor@Kala.com

Copyright (c) 1993, Penobscot Development Corporation

THIS PAGE BLANK (USPTO)

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:



BLACK BORDERS



IMAGE CUT OFF AT TOP, BOTTOM OR SIDES



FADED TEXT OR DRAWING



BLURRED OR ILLEGIBLE TEXT OR DRAWING



SKEWED/SLANTED IMAGES



COLOR OR BLACK AND WHITE PHOTOGRAPHS



GRAY SCALE DOCUMENTS



LINES OR MARKS ON ORIGINAL DOCUMENT



REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY



OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

Permissions Headers and Contract Law

by Henry H. Perritt, Jr.

ABSTRACT Law and technology must work together to minimize free riding on the intellectual contributions of authors and publishers. Contract-law and evidence-law doctrines can protect contributors in well-designed digital library systems, but unduly relying on encryption and other sophisticated technologies to protect intellectual property frustrates the vision of an open architecture for electronic publishing because they impose transaction costs disproportionate to the risk.

INTRODUCTION One of the ways to protect intellectual property on the NREN is through a digital library concept. Under this concept, a work would have attached to it a "permissions header," defining the terms under which the copyright owner makes the work available. The digital library infrastructure, implemented on the NREN, would match request messages from users with the permissions headers. If the request message and the permissions header match, the user would obtain access to the work. This concept encompasses major aspects of electronic contracting, which is already in wide use employing electronic data interchange (EDI) standards developed by ANSI Committee X12.¹

This paper explains the relationship between the digital library concept and EDI practice, synthesizing appropriate solutions for contract law, evidence, and agency issues that arise in electronic contracting. The question of how electronic signatures should work to be legally effective is an important part of this inquiry. The paper also defines particular types of service identifiers, header descriptors, and other forms of labeling and tagging appropriate to allow copyright owners to give different levels of permission, including outright transfer of the copyright interest, use permission, copying permission, distribution permission, display permission, and permission to prepare derivative works. The paper considers how payment authorization procedures should work in conjunction with a permissions header and digital library concept in order to integrate the proposed copyright licensing procedures with existing and anticipated electronic payment authorization systems. The paper necessarily considers whether existing standards approaches related to SGML (Standard Graphics Markup Language) and X12 are sufficient or whether some new standards development efforts will be necessary for implementation of the concepts. The paper considers the

relationship between technology and law in enforcing intellectual property, and emphasizes that the traditional adaptation of legal requirements to levels of risk is appropriate as the law is applied to new technologies.

There are certain common issues between the intellectual property question and other applications of wide area digital network technology. The question of signatures and writings to reflect the establishment of duties and permissions and the transfer of rights is common to the intellectual property inquiry and to electronic commerce using EDI techniques. There also are common questions involving rights to use certain information channels: First Amendment privileges, and tort liability. These are common not only to technological means of protecting intellectual property but to all forms of wide area networking.

THE PROBLEM

The law recognizes intellectual property because information technology permits one person to get a free ride on another person's investment in creating information value. Creative activity involving information usually is addressed by copyright, although patent has a role to play in protecting innovative means of processing information.² The concept of intellectual property arose in the context of letterpress printing technology. Newer technologies like xerography and more recently small computer technology and associated word processing and networking have increased the potential for free rides and accordingly increased the pressure on intellectual property.

The concern about free ride potential is especially great when people envision putting creative works on electronic publishing servers connected to wide area networks intending to permit consumers of information products to access these objects, frequently combining them and generally facilitating "publishing on demand" rather than the well known publishing just in case, typified by guessing how many copies of a work will sell, printing those in advance, and then putting them in inventory until someone wants them.

The concern is that it will be too easy to copy an entire work without detection and without paying for it. Worse, it will be easy to copy an entire work and resell it either by itself or as a part of a new derivative work or collection.

But technology is capable of protecting investment in new ways as well as offering the potential for a free ride. Computer networks make it possible to restrict access and to determine when access occurs. Depending on how new networks are designed, they may actually reduce the potential for a free ride. The digital library is one way of realizing that potential. Professor Pamela Samuelson has observed that the digital library model replaces intellectual property with a system of technological controls.³

DIGITAL LIBRARY CONCEPTS

Basic Concepts

A digital library is a set of information resources ("information objects") distributed throughout an electronic network. The objects reside on servers (computers with associated disk drives connected to the

network). They can be retrieved remotely by users using "client" workstations.

Origin of Concepts

The phrase "digital library" and the basic concept were first articulated in a 1989 report growing out of a workshop sponsored by the Corporation for National Research Initiatives.⁴ From its inception, the digital library concept envisioned retrieval of complete information resources and not merely bibliographic information.⁵

The technologies for remote retrieval of complete information objects using electronic technologies are in wide use through the WESTLAW, Dialog, LEXIS, NEXIS, and National Library of Medicine databases. These remotely accessible databases, however, unlike the digital library, involve a single host on which most of the data resides. The digital library concept envisions a multiplicity of hosts (servers).

Recent Developments

The remotely accessible database host concept is converging with the digital library concept as more of the electronic database vendors provide gateways to information objects actually residing on other computers. This now is commonplace with WESTLAW access to Dialog, and Dialog's gateways to other information providers.

The most explicit implementation of the digital library concept is the Wide Area Information Service (WAIS), which implements ANSI standard Z.39.50.⁶ WAIS permits a remote user to formulate a query that is applied to a multiplicity of WAIS servers, each of which may contain information responsive to the query. The WAIS architecture permits search engines of varying degrees of sophistication, resident on WAIS information servers, to apply the query against their own information objects, reporting matches back to the user.⁷ Future implementations of WAIS will permit automatic refinement of searches according to statistical matching techniques.

The Corporation for National Research Initiatives (CNRI) has proposed a test bed for an electronic copyright management system.⁸ The proposed system would include four major elements: automated copyright recording and registration, automated on-line clearance of rights, private electronic mail, and digital signatures to provide security. It would include three subsystems: a registration and recording system (RRS), a digital library system (DLS), and a rights management system (RMS). The RRS would provide the functions enumerated above and would be operated by the Library of Congress. It would provide "change of title" information.⁹ The RMS would be an interactive distributed system capable of granting rights on line and permitting the use of copyrighted material in the digital library system. The test bed architecture would involve computers connected to the Internet performing the RRS and RMS functions.

Digital signatures would link an electronic bibliographic record (EBR) with the contents of the work, ensuring against alteration after deposit.¹⁰ Multiple RMS servers would be attached to the Internet. A user wishing to obtain rights to an electronically published work would interact electronically with the appropriate RMS. When copyright ownership is

transferred, a message could be sent from the RMS to the RRS,¹¹ creating an electronic marketplace for copyrighted material.

The EBR submitted with a new work would "identify the rights holder and any terms and conditions on the use of the document or a pointer to a designated contact for rights and permissions."¹² The EBR, thus, is apparently equivalent to the permissions header discussed in this paper. Security in the transfer of rights would be provided by digital signatures using public key encryption, discussed further, *infra* in the section on encryption.

Basic Architectural Concepts

The digital library concept in general contemplates three basic architectural elements: a query, also called a "knowbot" in some descriptions; a permissions header attached to each information object; and a procedure for matching the query with the permissions header.

Two kinds of information are involved in all three architectural elements: information about the content of information objects desired and existing, and information about the economic terms on which an information object is made available. For example, a query desiring court opinions involving the enforcement of foreign judgments, evidencing a desire to download the full text of such judicial opinions and to pay up to \$1.00 per minute of search and downloading time, would require that the knowbot appropriately represent the subject matter "enforcement of foreign judgments." It also requires that the knowbot appropriately represent the terms on which the user is willing to deal: downloading and the maximum price. The permissions header similarly must express the same two kinds of information. If the information object to which the permissions header is attached is a short story rather than a judicial opinion, the permissions header must so indicate. Or, if the information object is a judicial opinion and it is about enforcement of foreign judgments, the permission header may indicate that only a summary is available for downloading at a price of \$10.00 per minute. The searching, matching, and retrieval procedure in the digital library system must be capable of determining whether there is a match on both subject matter and economic terms, also copying and transmitting the information object if there is a match.

Comparison to EDI

Electronic Data Interchange (EDI) is a practice involving computer-to-computer commercial dealing without human intervention. In the most widespread implementations, computers are programmed to issue purchase orders to trading partners, and the receiving computer is programmed to evaluate the terms of the purchase order and to take appropriate action, either accepting it and causing goods to be manufactured or shipped, or rejecting it and sending an appropriate message. EDI is in wide use in American and foreign commerce, using industry-specific standards for discrete commercial documents like purchase orders, invoices, and payment orders, developed through the American National Standards Institute.

There obviously are similarities between the three architectural elements of the digital library concept and EDI. There is a structured way

of expressing an offer or instruction, and a process for determining whether there is a match between what the recipient is willing to do and what the sender requests.

There is also, however, an important difference. In the digital library concept, a match results in actual delivery of the desired goods and services in electronic form. In EDI practice, the performance of the contractual arrangement usually involves physical goods or performance of nonelectronic services.

Nevertheless, the digital library and EDI architectures are sufficiently similar and, it turns out, the legal issues associated with both are sufficiently similar to make analogies appropriate.

Elements of Data Structure

For purposes of this paper, the interesting parts of the data structure are those elements that pertain to permission, more than those elements that pertain to content of the information object to which the header is attached. Accordingly, this section will focus only on permissions-related elements, after noting in passing that the content part of the header well might be a pointer to an inverted file to permit full text searching and matching.

The starting point conceptually for identifying the elements of the permissions header are the rights exclusively reserved to the copyright owner by § 106 of the copyright statute. But these exclusive rights need not be tracked directly because the owner of an information object is free to impose contractual restrictions as well as to enjoy rights granted by the Copyright Act. Accordingly, it seems that the following kinds of privileges in the requester should be addressed in the permissions header:

- outright transfer of all rights
- use privilege, either unrestricted or subject to restrictions
- copying, either unlimited or subject to restrictions like quantitative limits
- distribution, either unlimited or subject to restrictions, like geographic ones or limits on the markets to which distribution can occur
- preparation of derivative works.

Display and presentation rights, separately identified in § 106, would be subsumed into the use element, because they are particular uses.

The simplest implementation would allow only binary values for each of these elements. But a binary approach does not permit the permissions header to express restrictions, like those suggested in the enumerated list. Elements could be defined to accept the most common kinds of restrictions on use, and quantitative limits on copying, but it would be much more difficult to define in advance the kinds of geographic or market-definition restrictions that an owner might wish to impose with respect to distribution.

In addition to these discrete privileges, the permissions header must express pricing information. The most sensible way of doing this is to have a price associated with each type of privilege. In the event that different levels of use, copying, or distribution privilege are identified, the data structure should allow a price to be associated with each level.

A complicating factor in defining elements for price is the likelihood that different suppliers would want to price differently. For example, some would prefer to impose a flat fee for the grant of a particular privilege. Others might wish to impose a volume-based fee, and still others might wish to impose a usage or connect-time based fee. The data structure for pricing terms must be flexible enough to accommodate at least these three different approaches to pricing.

Finally, the data structure must allow for a specification of acceptable payment terms and have some kind of trigger for a payment approval procedure. For example, the permissions header might require presentation of a credit card number and then trigger a process that would communicate with the appropriate credit card database to obtain authorization. Only if the authorization was obtained would the knowbot and the permissions header "match."

There is a relationship between the data structures and legal concepts. The knowbot is a solicitation of offers. The permissions header is an offer. The matching of the two constitutes an acceptance. The "envelope" discussed elsewhere in these proceedings could be the "contract."

There are certain aspects of the data structure design that are not obvious. One is how to link price with specific levels of permission. Another is how to describe particular levels of permission. This representation problem may benefit from the use of some deontic logic, possibly in the form of a grammar developed for intellectual property permissions. Finally, it is not clear what the acceptance should look like. Conceptually, the acceptance occurs when the knowbot matches with a permissions header, but it is unclear how this legally significant event should be represented.

THE ROLE OF ENCRYPTION

The CNRI test bed proposal envisions the use of public key encryption to ensure the integrity of digital signatures and to ensure the authenticity of information objects. Public key encryption permits a person to encrypt a message - like a signature - using a secret key, one known only to the sender, while permitting anyone with access to a public key to decrypt it. Use of public key cryptography in this fashion permits any user to authenticate a message, ensuring that it came from the purported sender.¹³ A related technology called "hashing" permits an encrypted digital signature to be linked to the content of a message. The message can be sent in plain text (unencrypted) form, but if any part of it is changed, it will not match the digital signature. The digital signature and hashing technologies thus permit not only the origin but also the content integrity of a message of arbitrary length to be authenticated without necessitating encryption of the content of the message. This technology has the advantage, among others, that it is usable by someone lacking technological access to public key encryption. An unsophisticated user not wishing to incur the costs of signature verification nevertheless can use the content of the signed information object.

It is well recognized that encryption provides higher levels of security than other approaches. But security through encryption comes at a price. Private key encryption systems require preestablished relationships and exchange of private keys in advance of any encrypted communication.

The burdens of this approach have led most proponents of electronic commerce to explore public key encryption instead. But public key systems require the establishment and policing of a new set of institutions. An important infrastructure requirement for practicable public key cryptography is the establishment and maintenance of certifying entities that maintain the public keys and ensure that they are genuine ones rather than bogus ones inserted by forgers. A rough analogy can be drawn between the public key certifying entities and notaries public. Both kinds of institutions verify the authenticity of signatures. Both kinds require some level of licensing by governmental entities. Otherwise the word of the "electronic notary" (certifying entity) is no better than an uncertified, unencrypted signature. In a political and legal environment in which the limitations of regulatory programs have been recognized and have led to deregulation of major industries, it is not clear that a major new regulatory arrangement for public key encryption is practicable. Nevertheless, experimentation with the concept in support of digital library demonstration programs can help generate more empirical data as to the cost and benefits of public key encryption to reinforce electronic signatures.

On the other hand, it is not desirable to pursue approaches requiring encryption of content. No need to encrypt the contents is apparent in a network environment. Database access controls are sufficient to prevent access to the content if the permissions header terms are not matched by the knowbot. On the other hand, if the electronic publishing is effected through CD-ROMs or other physical media possessed by a user, then encryption might be appropriate to prevent the user from avoiding the permissions header and going directly to the content.

Encrypted content affords greater security to the owner of copyrighted material, because someone who has not paid the price to the copyright owner must incur a much higher cost to steal the material. But the problem is everyone must pay a higher price to use the material. One of the dramatic lessons of the desktop computer revolution was the clear rejection of copyright protection in personal computer software. The reasons that copy protection did not survive in the marketplace militate against embracing encryption for content. Encryption interferes with the realization of electronic markets, because producer and consumer must have the same encryption and description protocols. Encryption burdens the processing of electronic information objects because it adds another layer. Some specific implementations of encryption require additional hardware at appreciable costs.

Digital libraries cannot become a reality until consumers perceive that the benefits of electronic formats outweigh the costs, compared to paper formats. Encryption interferes with electronic formats' traditional advantages of density, reusability, editability, and computer search ability; also, by impairing open architectures, they may perpetuate some of paper's advantages with respect to browsability.¹⁴

The need for encryption of any kind depends upon whether security is available without it. That depends, in turn, on the kinds of free rides that may be obtainable and the legal status of various kinds of electronics transactions in the digital library system.

LEGAL ISSUES**Copyright: What legal effect is intended?**

The design of the permissions header and the values in the elements of the header must be unambiguous as to whether an outright transfer of a copyright interest is intended or whether only a license is intended. If an outright transfer¹⁵ is intended, then the present copyright statute requires a writing signed by the owner of the rights conveyed.¹⁶ Recordation of the transfer with the copyright office is not required, but provides advantages in enforcing transferee rights.¹⁷ On the other hand, non-exclusive licenses need not be in writing nor registered. If the electronic transaction transfers the copyright in its entirety, then the rights of the transferor are extinguished, and the rights of the transferee are determined by the copyright statute. The only significant legal question is whether the conveyance was effective.

On the other hand, when the copyright is not transferred outright but only certain permissions are granted or certain rights conveyed, the legal questions become more varied. Then, the rights of the transferor and the obligations of the transferee are matters of contract law. It is important to understand the degree to which the contract is enforceable and how it is to be interpreted in the event of subsequent disputes. The following sections consider briefly the first sale doctrine as a potential public policy obstacle to enforcing contractual restrictions different from those imposed by the copyright statute and then explore in greater depth whether electronic techniques satisfy the formalities traditionally required for making a contract, whether they adequately ensure against repudiation, and whether they provide sufficient information to permit predictable interpretation of contractual obligations and privileges.

First Sale Doctrine

The first sale doctrine may invalidate restrictions on use. It is impermissible for the holder of a patent to impose restrictions on the use of a patented product after the product has been sold. Restrictions may be imposed, however, on persons who merely license the product.¹⁸ The rationale for this limit on the power of the owner of the intellectual property interest is that to allow limitations on use of the product would interfere with competition beyond what the Congress - and arguably the drafters of the Constitution - intended in setting up the patent system.

The first sale doctrine applies to copyright owners.¹⁹ Indeed, because of the First Amendment's protection of informational activity, the argument against restrictions after the first sale may be even stronger in the copyright arena than in the patent arena.

The first sale doctrine is potentially important because it may invalidate restrictions imposed on the use of information beyond what is authorized by the Copyright Act and by common law on trade secrets. Thus, there may be serious questions about the legal efficacy of use restrictions, although such restrictions are common in remote database service agreements. The vendors could argue that the limitations pertain to the contractual terms for delivery of a service rather than use of information as such. The characterization avoids the overlap with copyright and thus may also avoid the conflict between federal policy and contract enforcement.²⁰

Contract Formation Issues

The law does not enforce every promise. Instead, it focuses its power only on promises surrounded with certain formalities to make it likely that the person making the promise (the "promisor") and the person receiving the promise (the "promisee") understood that their communication had legal consequences. A threshold question for the digital library system is whether the traditional formalities for making a contract are present when the contract is made through electronic means. The digital library system considered in this paper clearly contemplates that a contract is formed when the knowbot and the permissions header achieve a match. In this respect, the digital library concept converges with EDI, where trading parties contemplate that a contract to perform services or deliver goods is formed when a match occurs either upon the receipt of a purchase order or upon the transmission of a purchase order acknowledgment.

It is not altogether clear, however, whether the match between values and computer data structures meets contract formation requirements, particularly those expressed in various statutes of frauds. Statutes of frauds require "writings" and "signatures" for certain kinds of contracts - basically those contemplating performance extending beyond a period of one year.²¹

In many instances, the digital library contract will be fully performed almost instantaneously upon delivery of the information object after the knowbot and the permissions header match. In such a case, the statute of frauds is not a problem and its requirements need not be satisfied. In other cases, however, as when the intent of the owner of the information object is to grant a license to do things that will extend beyond one year, the statute of frauds writing and signature requirements must be met.

Historical application of statutes of frauds by the courts clearly indicates that there is flexibility in the meaning of "writing" and "signature." A signature is any mark made with the intent that it be a signature.²² Thus an illiterate person signs by making an "X," and the signature is legally effective. Another person may sign a document by using a signature stamp. Someone else may authorize an agent to sign his name or to use the signature stamp. In all three cases the signature is legally effective. There may of course be arguments about who made the X, or whether the person applying the signature stamp was the signer or his authorized agent, but these are evidentiary and agency questions, not arguments about hard and fast contract-law requirements.

Under the generally accepted legal definition of a signature, there is no legal reason why the "mark" may not be made by a computer printer, or for that matter by the write head on a computer disk drive or the data bus in a computer random access memory. The authorization to the computer agent to make the mark may be given by entering a PIN (personal identification number) on a keyboard. To extend the logic, there is no conceptual reason to doubt the legal efficacy of authority to make a mark if the signer writes a computer program authorizing the application of a PIN upon the existence of certain conditions that can be tested by the program. The resulting authority is analogous to a signature pen that can be operated only with a mechanical key attached to

somebody's key ring, coupled with instructions to the possessor of the key.

Which of these various methods should be selected for particular types of transactions must not depend on what the law requires, because the law permits any of these methods. Rather, it must depend on the underlying purposes of the legal requirement and which method best serves those purposes.

The real issue is how to prove that a particular party made the mark. In other words, the contingency to be concerned about is repudiation, not absence of formalities. Repudiation should be dealt with through the usual evidentiary and fact finding processes rather than artificial distinctions between signed and unsigned documents.

Authority is skimpier on how flexible the "writing" requirement is. The best approach is to borrow the fixation idea from the copyright statute and conclude that a writing is "embodiment in a copy . . . sufficiently permanent or stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration."²³

The most important thing conceptually is to understand the purpose of the writing and signature requirements. They have two purposes: awareness or formality, and reliability of evidence. Signature requirements, like requirements for writings and for original documents, have an essentially evidentiary purpose. If there is a dispute later, they specify what kind of evidence is probative of certain disputed issues, like "who made this statement and for what purpose?" The legal requirements set a threshold of probativeness. Surely the values in a knowbot as well as the values in a permissions header constitute a "mark," and someone who knowingly sets up potential transactions in a digital library scheme can have the intent that the mark be a signature.

When a contract is made through a signed writing, it is more likely that the parties to the contract understand what they are doing. They are aware of the legal effect of their conduct because the writing in the signature involves a greater degree of formality than a simple conversation.

The awareness/formality purpose can be served by computerized contracting systems. This is so not so much because the computers are "aware" of the affect of their "conduct." Rather, it is true because the computers are agents of human principals. The programming of the computer to accept certain contract terms is the granting of authority to the computer agent to enter into a contract. The fact that a principal acts through an agent engaging in conduct at a later point in time never has been thought to defeat contract formation in the traditional evolution of agency and contract law. Nor should it when the agent is a computer.

Fulfillment of the evidentiary purpose depends on the reliability of the information retained by the computer systems making up the digital library. Such systems must be designed to permit the proponent of contract formation to establish the following propositions if the other party to the purported contract attempts to repudiate it.

1. It came from computer X.
2. It accurately represents what is in computer X²⁴ now.²⁵
3. What is in computer X now is what was in computer X at the time of the transaction.

4. What was in computer X at the time of the transaction is what was received from the telecommunications channel.²⁶
5. What was received from the telecommunications channel is what was (a) sent, (b) by computer Y.

Two other questions relate to matters other than the authenticity of the message:

6. Computer Y was the agent of B.
7. The message content expresses the content of the contract (or more narrowly, the offer or the acceptance).²⁷

Factual propositions 1-4 can be established by testimony as to how information is written to and from telecommunications channel processors, primary storage, and secondary storage. Factual proposition 5 requires testimony as to the accuracy of the telecommunications channel and characteristics of the message that associate it with computer Y. Only the last proposition (number 5) relates to signatures, because signature requirements associate the message with its source.²⁸ The other propositions necessitate testimony as to how the basic message and database management system works. It is instructive to compare these propositions with the kinds of propositions that must be established under the business records exception to the hearsay rule when it is applied to computer information.

Those propositions may be supported with non-technical evidence, presented by non-programmers. A witness can lay a foundation for admission of computer records simply by testifying that the records are generated automatically and routinely in the ordinary course of business. The more inflexible the routine, and the less human intervention in the details of the computer's management of the database, the better the evidence.²⁹

The ultimate question is trustworthiness, and if the computer methods are apparently reliable, the information should be admitted unless the opponent of admissibility can raise some reasonable factual question undercutting trustworthiness.³⁰

CONTRACT INTERPRETATION ISSUES

Assuming that the permissions header and knowbot constitute sufficient writings to permit a contract to be formed and that the signature requirement also is met, through digital signature technology or otherwise, there still are difficult contract interpretation questions. Contract interpretation questions arise not only after contractual relationships are formed, but also in connection with deciding whether there has been offer and acceptance, the prerequisites to contract formation.³¹ Contract interpretation always seeks to draw inferences about what the parties intended. When contract interpretation issues arise at the contract formation stage, the questions are what the offeror intended the content of the offer to be and what the offeree intended the content of the purported acceptance to be. The proposed digital library system envisions extremely cryptic expressions of offer and acceptance — by means of codes. The codes have no intrinsic meaning. Rather, extrinsic reference must be made to some kind of table, standard, or convention associating particular codes with the concepts they represent. Extrinsic evidence is available to

resolve contract interpretation questions when the language of the contract itself is ambiguous, and perhaps at other times as well.³² The codes in the permissions header and knowbots certainly are ambiguous and become unambiguous only when extrinsic evidence is considered. So there is no problem in getting a standard or cable into evidence. The problem is whether the parties meant to assent to this standard.

In current EDI practice, this question is resolved by having parties who expect to have EDI transactions with each other sign a paper trading partner agreement, in which the meaning of values or codes in the transaction sets is established.³³ But requiring each pair of suppliers and users of information in a digital library to have written contracts with each other in advance would defeat much of the utility of the digital library. Thus the challenge is to establish some ground rules for the meaning of permissions header and knowbot values to which all participants are bound. There are analogous situations. One is a standard credit card agreement that establishes contractual terms among credit card issuer, credit card subscriber, and merchant who accepts the credit card. The intermediary — the credit card company — unilaterally establishes contract terms to which the trading partners assent by using and accepting the credit card.³⁴ Also, it is widely recognized that members of a private association can, through their constitution and bylaws, establish contractual relationships that bind all of the members in dealing with each other.³⁵ In the digital library system, similar legal arrangements can establish the standards by which electronic transactions between permissions header and knowbots will bind the transferor and the transferee of information.

Third Party Liability

It is not enough merely to ensure that the licensee is contractually bound. Trading partners also must ensure that the participants in funds transfers have enforceable obligations. For example, if the digital library system envisions that the information object would not be released to the purchaser without simultaneous release of a payment order, the supplier may be interested in enforcing the obligations of financial intermediaries who handle the payment order. This implicates the federal Electronic Funds Transfer Act, and Article 4A of the Uniform Commercial Code, regulating wire transfers.

SOLUTIONS

Satisfy the Business Records Exception to the Hearsay Rule

The discussion of contract formalities earlier in this paper concluded that legally enforceable contracts can be formed through electronic means and that the significant legal questions relate to reliability of proof and intent of the parties to be bound by using the electronic techniques. This section considers the reliability of proof further. Traditional evidence law permits computer records to be introduced in evidence when they satisfy the requirements of the business records exception: basically that they are made in the ordinary course of business, that they are relied on for the performance of regular business activities, and that there is no independent reason for questioning their reliability.³⁶

The business records exception shares with the authentication concept, the statute of frauds, and the parol evidence rule a common concern with reliability.³⁷ The same procedural guarantees and established practices that ensure reliability for hearsay purposes also ensure reliability for the other purposes. Under the business records exception, the proponent must identify the source of a record, through testimony by one familiar with a signature on the record, or circumstantially.³⁸ The steps in qualifying a business record under the common law, which since have been relaxed,³⁹ were:

- proving that the record is an original entry made in the routine course of business
- proving that the entries were made upon the personal knowledge of the proponent/witness or someone reporting to him
- proving that the entries were made at or near the time of the transaction
- proving that the recorder and his informant are unavailable.⁴⁰

These specific requirements are easier to understand and to adapt to electronic permissions and obligations formed in a digital library system by understanding the rationale for the business records exception. The hearsay rule excludes out-of-court statements because they are inherently unreliable, primarily because the maker of the statement's demeanor cannot be observed by the jury and because the maker of the statement is not subject to cross examination. On the other hand, there are some out-of-court statements that have other guarantees of reliability. Business records are one example. If a continuing enterprise finds the records sufficiently reliable to use them in the ordinary course of business, they should be reliable enough for a court. The criteria for the business records exception all aim at ensuring that the records really are relied upon by the business to conduct its ordinary affairs.

The Manual for Multidistrict Litigation suggests steps for qualifying computer information under the business records exception:

1. The document is a business record.
2. The document has probative value.
3. The computer equipment used is reliable.
4. Reliable data processing techniques were used.⁴¹

Key to adapting the business records exception to electronic permissions in a digital library system are points 3 and 4. Establishing these propositions and the evidentiary propositions set forth elsewhere in this paper requires expert testimony. Any designer of a digital library system must consult with counsel and understand what testimony an expert would give to establish these propositions. Going through that exercise will influence system design.

Reinforce the Evidentiary Reliability by Using Trusted Third Parties

The evidentiary purpose of contract formation requirements can be satisfied by using a trusted third party as an intermediary, when the third party maintains archival records of the transactions. The third party lacks any incentive for tampering with the records and when the third party's

archiving system is properly designed, it can provide evidence sufficient to establish all of the propositions.

This third party intermediary concept is somewhat different from the concept of a certifying agent in digital signature systems. To be sure, the custodian of transaction records envisioned by this section could be the same as the certifying entity for public and key encryption, but the custodian role can be played in the absence of any encryption. Indeed, the digital library itself is a good candidate for the custodian role. The library has no incentive to manipulate its records in favor of either the producers of information value or the consumers. In order to carry out its affairs, it must use these transactional records in the ordinary course of business, thereby making it likely that digital library records would qualify under the business records exception.

Standardization

Obviously, the digital library concept depends upon the possibility of an automated comparison between the knowbot and the permissions header. This means that potential requesters of information and suppliers of information must know in advance the data structures for representing the elements of the permissions header and the knowbot. This requires compatibility. Compatibility requires standardization. Standardization does not, however, necessarily require "Standards" in the sense that they are developed by some bureaucratic body like ANSI. It may simply imply market acceptance of a particular vendor's approach. Indeed, each digital library might use different data structures. All that is necessary is that the structure of the knowbot and the structure of the permissions header be compatible within any one digital library system. Also, as demands emerge for separate digital libraries to communicate with each other, there can be proprietary translation to assure compatibility between systems much as common word processing programs translate to and from other common formats and much as printers and word processing software communicate with each other through appropriate printer drivers. In neither of these cases has any independent standards organization developed a standard that is at all relevant in the marketplace.

Standardizing the elements of knowbot and permissions headers involves content standardization, which generally is more challenging than format standardization.⁴² A permissions header/knowbot standard is a system for representing legal concepts and for defining legal relations. As such, the standard is basically a grammar for a rule-based substantive system in a very narrow domain.⁴³ The data elements must correspond to legally meaningful relational attributes. The allowable values must correspond to legally allowable rights, obligations, privileges and powers. In other words, the standard setter must meet many of the challenges that a legal expert system designer working with Hohfeldian frameworks must meet.⁴⁴ This adds a constraint to the standards setting process. Unlike setting format standards, where the participants are free to agree on an arbitrary way of expressing format attributes, participants in setting a content standard must remain within the universe of permissible content. The set of permissible values is determined by the law rather than being determined only by the imagination of format creators.

**ENFORCEMENT AND
BOTTLENECKS**

One of the many profound observations by Ithiel de Sola Pool⁴⁵ was that copyright always has depended upon technological bottlenecks for its enforceability. The printing press was the original enforcement bottleneck. Now, a combination of the printing press and the practical need to inventory physical artifacts representing the work constitute the enforcement bottlenecks. As technologies change, old bottlenecks disappear and enforceability requires a search for new bottlenecks. When there are single hosts, like WESTLAW, Dialog, LEXIS, and CompuServe, access to that host is the bottleneck. The problem with distributed publishing on an open architecture internet is that there is no bottleneck in the middle of the distribution chain corresponding to the printer, the warehouse or the single host.

If new bottlenecks are to be found, they almost surely will be found at the origin and at the point of consumption. Encryption and decryption techniques discussed elsewhere in this volume concentrate on those bottlenecks as points of control. It also is possible that rendering software could become the new bottleneck.

Even with those approaches, however, a serious problem remains in that the new technologies make it difficult or impossible to distinguish between mere use and copying. Thus the seller cannot distinguish between an end user⁴⁶ and a potential competitor. On the other hand, the new technologies permit a much better audit trail, potentially producing better evidence for enforcement adjudication.

If network architectures for electronic publishing evolve in the way that Ted Nelson suggests with his Xanadu concept⁴⁷, the real value will be in the network and the pointers, not in the raw content. Thus, the creative and productive effort that the law should reward is the creation and production and delivery of pointers, presentation, distribution, and duplication value. If this is so, then technological means will be particularly important, foreclosing access by those lacking passwords and other keys and limiting through contract what a consumer may do with the information.

In such an architecture, the law either will be relatively unimportant because technology can be counted on to prevent free riding, or the law will need to focus not on prohibiting copying or use without permission, but on preventing circumvention of the technological protections. Thus, legal approaches like that used to prevent the sale of decryption devices for television broadcasts and legal issues associated with contract enforcement may be more important than traditional intellectual property categories.

WEIGHING RISKS AND COSTS

The law generally imposes sensible levels of transaction costs. Usually, transaction costs are proportional to the risk. Figure 1 shows a continuum of risk and transaction cost in traditional and new technologies. A real estate closing involves significant risks if there is some dispute later about the transaction. Therefore, the law affords much protection, including a constitutional officer called a registrar of deeds who is the custodian of records associated with the transaction. The risk level analogous to this in electronic publishing might be access to an entire library including access software as well as contents. Next on

the continuum is a transaction involving a will or power of attorney. There, the risk is substantial because the maker of the instrument is not around to help interpret it. The law requires relatively high levels of assurance here, though not as great as those for real estate transactions. The law requires witnesses and attestation by a commissioned minor official called a notary public. The electronic publishing analogy of this level of risk might be the contents of an entire CD-ROM.

traditional transaction	institutions	electronic equivalent
real estate closing	registrar of deeds	entire library - software and contents
will/power of attorney	witnesses, notary public	contents of entire CD-ROM
auto purchase	UCC financing statement	complete work - transfer of copyright
television set purchase	written sale agreement	complete work - use permission
box of diskettes	handshake	part of a work - use permission

Figure 1 - Cost effectiveness = risk-proportional security

Next in level of risk is the purchase of a large consumer durable like an automobile. The law requires somewhat less, but still significant, protections for this kind of transaction: providing for the filing and enforcement of financing statements under the Uniform Commercial Code. The electronic publishing analogy might be the transfer of copyright to a complete work. Next along the risk continuum is the purchase of a smaller consumer durable like a television set. Here, the law typically is reflected in written agreements of sale, but no special third party custodial mechanisms. The electronic publishing analogy might be use permission for a complete work.

At the other end of the continuum is the purchase of a relatively small consumer item, say a box of diskettes. Neither the law nor commercial practice involves much more than the exchange of the product for payment, with no written agreement or anything else to perform channeling, cautionary, evidentiary, or protective functions. The electronic publishing analogy might be use permission for part of a work.

Realizing the potential of electronic publishing in distributed information networks requires sensitivity to the transaction costs of too much security. Requiring \$10 boxes of diskettes to be sold like real property would impose unacceptable transaction costs. Similarly, an encrypted

object combined with rendering software is probably inconsistent with an open architecture. Because of the difficulty of setting standards for such technologies, this approach to intellectual property protection probably would be effectuated by proprietary approaches, thus frustrating the vision of an open market for electronic publishing.

CONCLUSION

Realization of the digital library vision requires a method for collecting money and granting permission to use works protected by intellectual property. The concept of a knowbot and a permissions header attached to the work is the right way to think about such a billing and collection system. Standards for the data structures involved must be agreed to, and systems must be designed to satisfy legal formalities aimed at ensuring awareness of the legal significance of transactions and reliable proof of the terms of the transactions.

In the long run, not only must these technological issues be resolved, with appropriate attention to levels of risk and protections available under traditional legal doctrines, but also further conceptual development must be undertaken. Proponents of electronic publishing over wide area networks need to think about the appropriate metaphors: whether it is a library or a bookstore, if a library whether with or without Xerox machines, if a bookstore whether it is a retail bookstore or a mail order operation. Then, thought must be given to how standards will be set. Finally, and most important, much more needs to be understood about the need for third party institutions. There is a good deal of enthusiasm for public key encryption. Yet the vulnerability of public key encryption systems is in the integrity of the key authority. In traditional legal protections, the third party custodians or authenticating agents like notaries public and registrars of deeds receive state sanction and approval, and in the case of registrars of deeds, public funding. We must be clearer as to whether a similar infrastructure must be developed to protect against substantial risks and the use of EDI and electronic publishing technologies.

Finally, and perhaps most importantly, we must be thoughtful about what legal obligations, imposed on whom, are appropriate. The suggested paragraphs 102(e) and (f) in the High Performance Computing Act look very much like King James I's licensing of printing presses. They also look like the FBI's proposal to prohibit the introduction of new technologies until certain conformity with past legal concepts is assured. Such approaches make the law a hurdle to new technology - an uncomfortable position for both law and technology.

NOTES

1. The use of EDI techniques to meter usage and determine charges for use of intellectual property is an example of billing and collection value in a typology of different types of value that can be produced in electronic marketplaces for information. See Henry H. Perritt, Jr., *Market Structures for Electronic Publishing and Electronic Contracting* in Brian Kahin, ed., *BUILDING INFORMATION INFRASTRUCTURE: ISSUES IN THE DEVELOPMENT OF THE NATIONAL RESEARCH AND EDUCATION NETWORK* (Harvard University and McGraw-Hill 1992) (developing typology for different types of value and explaining how market

structures differ for the different types); Henry H. Perritt, Jr., *Tort Liability, the First Amendment, and Equal Access to Electronic Networks*, 5 Harv.J.Law & Tech. 65 (1992) (using typology of ten types of value to analyze access by competing producers of value).

2. See, e.g., U.S.Pat. No. 5,016,009, Data compression apparatus and method (May 14, 1991); U.S. Pat. No. 4,996,690, Write operator with gating capability (Feb. 26, 1991); U.S. Pat. No. 4,701,745, Data compression system (Oct. 20, 1987); Multi Tech Systems, Inc. v. Hayes Microcomputer Products, Inc., 800 F. Supp. 825 (D. Minn. 1992) (denying summary judgment on claim that patent for modem escape sequence is invalid).

3. Comments on the 8/21 draft of "Knowbots in the Real World" from the intellectual property workshop participants, page 6 (author unknown, source unknown). Professor Samuelson also observed that the workshop, despite its title, actually did not focus much on intellectual property issues. See fn. 4, below.

4. Corporation for National Research Initiatives, *Workshop on the Protection of Intellectual Property Rights in a Digital Library System: Knowbots in the Real World-May 18-19, 1989* (describing digital library system).

5. See generally Clifford A. Lynch, *Visions of Electronic Libraries* (libraries of future can follow acquisition-on-demand model rather than acquiring in advance of use; Z39.50 protocol will facilitate realization of that possibility, citing Robert E. Kahn & Vinton G. Serf, *An Open Architecture for a Digital Library System and a Plan for Its Development*. The Digital Library Project, volume 1: The World of Knowbots (draft) (Washington D.C.: Corporation for National Research Initiatives; 1988)).

6. Clifford A. Lynch, *The Z39.50 Information Retrieval Protocol: An Overview and Status Report*, ACM Sigcomm Computer Communication Review at 58 (describing Z39.50 as an OSI application layer protocol that relieves clients from having to know the structure of data objects to be queried, and specifies a framework for transmitting and managing queries and results and syntax for formulating queries).

7. Brewster Kahle, Wide Area Information Server Concepts (Nov. 3, 1989 working copy; updates available from Brewster @THINK.com. (describing WAIS as "open protocol for connecting user interfaces on workstations and server computers") (describing information servers as including bulletin board services, shared databases, text searching and automatic indexing and computers containing current newspapers and periodicals, movie and television schedules with reviews, bulletin boards and chat lines, library catalogues, Usenet articles).

8. Robert E. Kahn, *Deposit, Registration, Recordation in an Electronic Copyright Management System* (August 1992) (Corporation for National Research Initiatives, Reston, Virginia). CNRI claims a trademark in "knowbot" and "digital library".

9. Kahn 1992 at 4.

10. Kahn 1992 at 6.

11. Kahn 1992 at 10.

12. Kahn 1992 at 12.

13. Kahn 1992 at 15.

14. Browsability through techniques like the collapsible outliner function in Microsoft Word for Windows and competing products requires more chunking and tagging value in the form of style and text element codes. Handling this

additional formatting information through encryption and description processes is problematic.

15. "A 'transfer of copyright ownership' is an assignment, mortgage, exclusive license, or any other conveyance, alienation, or hypothecation of a copyright or of any of the exclusive rights comprised in a copyright, whether or not it is limited in time or place of effect, but not including a non-exclusive license " 17 U.S.C. § 101 (1988).

16. 17 U.S.C. § 204(a) (1988); *Valente-Kritzer Video v. Pinckney*, 881 F.2d 772, 774 (9th Cir. 1989) (affirming summary judgment for author; oral agreement unenforceable under Copyright Act); *Library Publications, Inc. v. Medical Economics Co.*, 548 F. Supp. 1231, 1233 (E.D. Pa. 1982) (granting summary judgment against trade book publisher who sought enforcement of oral exclusive distribution agreement; transfer of exclusive rights, no matter how narrow, must be in writing), *aff'd mem.*, 714 F.2d 123 (3d Cir. 1983).

17. 17 U.S.C. § 205 (1988) provides constructive notice of the contents of the recorded document, determining priority as between conflicting transfers, and determines priority as between recorded transfer and non-exclusive license. The former requirement for transfers to be recorded in order for the transferee to maintain an infringement, 17 U.S.C. § 205(d), was repealed by the Berne Act Amendments § 5.

18. Under *Adams v. Burke*, 84 U.S. (17 Wall.) 453 (1873), a patentee must not attempt to exert control past the first sale. In general, use restrictions may be placed only on licensees, consistent with *General Talking Pictures v. Western Elec.*, 304 U.S. 175 (1938). See generally *Baldwin-Lima-Hamilton Corp. v. Tatnall*, 169 F. Supp. 1 (E.D. Pa. 1958) (applying no control after purchase rule).

19. See *Red-Baron-Franklin Park, Inc. v. Taito Corp.*, 883 F.2d 275, 278 (4th Cir. 1989) (purchase of video game circuit boards did not create privilege to perform video game under first sale doctrine); *United States v. Moore*, 604 F.2d 1228, 1232 (9th Cir. 1979) (pirated sound recording not within first sale doctrine in criminal copyright infringement prosecution). But see *Mirage Editions, Inc. v. Albuquerque A.R.T. Co.*, 856 F.2d 1341, 1344 (9th Cir. 1988) (first sale doctrine did not create privilege to prepare derivative work by transferring art in book to ceramic tiles).

20. The way in which the first sale doctrine would impact the electronically imposed use restrictions is by frustrating a breach-of-contract lawsuit by the licensor against a licensee who exceeds the use restrictions. The licensee exceeding the use restrictions would argue that it violates public policy to enforce the restrictions and therefore that state contract law may not impose liability for their violation. See generally *Restatement (second) of Contracts § 178* (1981) (stating general rule for determining when contract term is unenforceable on grounds of public policy).

21. In addition, the Copyright Act itself requires signed writings for transfers of copyright interests. 17 U.S.C. § 204(a). (1988).

22. Michael S. Baum & Henry H. Perritt, Jr., *ELECTRONIC CONTRACTING, PUBLISHING AND EDI LAW* ch. 6 (1991) (contract, evidence and agency issues) [hereinafter "Baum & Perritt"]. Accord, *Signature Requirements Under EDGAR*, Memorandum from D. Goelzer, Office of the General Counsel, SEC to Kenneth A. Fogash, Deputy Executive Director, SEC (Jan. 13, 1986) (statutory and non-statutory requirements for "signatures" may be satisfied by means other than manual writing on paper in the hand of the signatory . . . "In fact, the electronic transmission of an individual's name may legally serve as that person's signature, providing it is transmitted with the present intention to authenticate.").

23. 17 U.S.C. § 101 (1988). For copyright purposes, a work is created, and therefore capable of protection, when it is fixed for the first time. 17 U.S.C. § 101 (1988). "[I]t makes no difference what the form, manner, or medium of fixation may be—whether it is in words, numbers, notes, sounds, pictures, or any other graphic or symbolic indicia, whether embodied in a physical object in written, printed, photographic, sculptural, punched, magnetic, or any other stable form, and whether it is capable of perception directly or by means of any machine or device 'now known or later developed.'" 1976 U.S. Code Cong. & Admin. News 5659, 5665. The legislative history further says that, "the definition of 'fixation' would exclude from the concept [representations] purely of an evanescent or transitory nature—reproductions such as those projected briefly on a screen shown electronically on a television or other video display, or captured momentarily in the 'memory' of a computer." 17 U.S.C. § 102 note (excerpting from House Report 94-1476).

24. Or, more likely, what is on the computer medium read by computer X, such as a magnetic cartridge used for archival records. Further references in the textual discussion to "what is in computer X now" should be understood to include such computer-readable media.

25. Cf. R. Peritz, *Computer Data and Reliability: A Call for Authentication of Business Records Under the Federal Rules of Evidence*, 80 Nw.U.L.Rev. 956, 980 (1986) (proof that a printout accurately reflects what is in the computer is too limited a basis for authentication of computer records).

26. In some cases, the electronic transaction will be accomplished by means of a physical transfer of computer readable media. In such a case, this step in the proof would involve proving what was received physically.

27. See generally Peritz, 80 Nw.U.L.Rev. at 979 (citing as examples of authentication *Ford Motor Credit Co. v. Swarens*, 447 S.W.2d 53 (Ky. 1969) (authentication by establishing relationship between computer-generated monthly summary of account activity and the customer reported on); *Ed Guth Realty, Inc. v. Gingold*, 34 N.Y.2d 440, 315 N.E.2d 441, 358 N.Y.S.2d 367 (1974) (authentication of summary of taxpayer liability and the taxpayer)).

28. Of course, a paper document signed at the end also is probative of the fact that no alterations have been made. In this sense, a signature requirement telescopes several steps in the inquiry outlined in the text.

29. *United States v. Linn*, 880 F.2d 209, 216 (9th Cir. 1989) (computer printout showing time of hotel room telephone call admissible in narcotics prosecution). See also *United States v. Miller*, 771 F.2d 1219, 1237 (9th Cir. 1985) (computer-generated toll and billing records in price-fixing prosecution based on testimony by billing supervisor although he had no technical knowledge of system which operated from another office; no need for programmer to testify; sufficient because witness testified that he was familiar with the methods by which the computer system records information).

30. See *United States v. Hutson*, 821 F.2d 1015, 1020 (5th Cir. 1987) (remanding embezzlement conviction, although computer records were admissible under business records exception, despite trustworthiness challenged based on fact that defendant embezzled by altering computer files; access to files offered in evidence was restricted by special code).

31. Restatement (Second) of Contracts §§ 17, 24, 35 (1981).

32. John E. Murray, Jr., *Murray on Contracts* § 89, 3rd ed. (Charlottesville, VA: Michie, 1990).

33. See Baum & Perritt § 2.6; The Electronic Messaging Services Task Force, *The Commercial Use of Electronic Data Interchange—A Report and Model Trading Partner Agreement*, 45 Bus.Law. 1645 (1990); Jeffrey B. Ritter, *Scope of the Uniform Commercial Code: Computer Contracting Cases and Electronic Commercial Practices*, 45 Bus.Law. 2533 (1990); Note, *Legal Responses to Commercial Transactions Employing Novel Communications Media*, 90 Mich.L.Rev. 1145 (1992).
34. *Garber v. Harris Trust & Savings Bank*, 432 N.E.2d 1309, 1311-1312 (Ill. App. 1982) ("each use of the credit card constitutes a separate contract between the parties;" citing cases). It is not quite this simple, because both merchant and credit card customer have separate written contracts with the credit card issuer. But there is no reason that a supplier of information to a Digital Library System and all customers of that system might not have their own contracts with the Digital Library System in the same fashion.
35. *Rowland v. Union Hills Country Club*, 757 P.2d 105 (Ariz. 1988) (reversing summary judgment for country club officers because of factual question whether club followed bylaws in expelling members); *Straub v. American Bowling Congress*, 353 N.W.2d 11 (Neb. 1984) (rule of judicial deference to private associations, and compliance with association requirements, counseled affirmance of summary judgment against member of bowling league who complained his achievements were not recognized). But see *Wells v. Mobile County Board of Realtors, Inc.*, 387 So.2d 140 (Ala. 1980) (claim of expulsion of realtor from private association was justifiable and bylaws, rules and regulations requiring arbitration were void as against public policy; reversing declaratory judgment for defendant association).
36. F.R.E. 803(6) (excluding business records from inadmissibility as hearsay); 28 U.S.C. § 1732 ("Business Records Act" permitting destruction of paper copies of government information reliably recorded by any means and allowing admission of remaining reliable record).
37. See Peritz, 80 Nw.U.L.Rev. at 978-80, 984-85 (noting body of commentator opinion saying that business records exception and authentication are parallel ways of establishing reliability).
38. See F.R.E. 901(b)(4) (appearance, contents, substance, internal patterns, as examples of allowable authentication techniques).
39. Peritz, 80 Nw.U.L.Rev. at 963-64 (identifying steps and trend resulting in F.R.E.).
40. Peritz, 80 Nw.U.L.Rev. at 963.
41. Peritz, 80 Nw.U.L.Rev. at 974 (reporting four requirements of Manual, and endorsing their use generally).
42. See Henry H. Perritt, Jr., *Format and Content Standards for the Electronic Exchange of Legal Information*, 33 Jurimetrics 265 (1993) (distinguishing between format and content standardization).
43. Marc Lauritsen, Senior Research Associate at Harvard Law School, has written about the relationship between substantive legal systems and the field of artificial intelligence.
44. Anne Gardner, *An Artificial Intelligence to Legal Reasoning* (Cambridge, MA: MIT Press, 1987); Kevin Ashley, *Modeling Legal Argument* (Cambridge, MA: MIT Press, 1990).
45. Ithiel de Sola Pool, *Technologies of Freedom* (Cambridge, MA: Harvard University Press, 1983), p. 16-17, 249.

46. It may not be particularly important to limit competition by consumers, because the consumers will never have the pointers and the rest of the network infrastructure.

47. This concept was discussed at the conference.

BIOGRAPHY

Henry H. Perritt, Jr., Professor of Law at Villanova University School of Law, was Deputy Undersecretary of Labor in the Ford Administration, and worked on telecommunications on President Clinton's Transition Team. He holds a B.S. and S. M. from M.I.T. and a J.D. from Georgetown, and has written ten books and 30 articles.

Henry H. Perritt, Jr.
Villanova University School of Law
Villanova, PA 19085
(215) 645-7078
FAX (215) 896-1723
Internet: perritt@ucis.vill.edu.

Petitioner Apple Inc. - Exhibit 1012, p. 757

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FI	Finland	MI	Mali
AU	Australia	FR	France	MN	Mongolia
BB	Barbados	GA	Gabon	MR	Mauritania
BE	Belgium	GB	United Kingdom	MW	Malawi
BF	Burkina Faso	GN	Guinea	NI	Netherlands
BG	Bulgaria	GR	Greece	NO	Norway
BJ	Benin	HU	Hungary	PL	Poland
BR	Brazil	IE	Ireland	RO	Romania
CA	Canada	IT	Italy	RU	Russian Federation
CF	Central African Republic	JP	Japan	SD	Sudan
CG	Congo	KP	Democratic People's Republic of Korea	SE	Sweden
CH	Switzerland	KR	Republic of Korea	SN	Senegal
CI	Côte d'Ivoire	LI	Liechtenstein	SU	Soviet Union
CM	Cameroon	LK	Sri Lanka	TD	Chad
CS	Czechoslovakia	LU	Luxembourg	TG	Togo
DE	Germany	MC	Monaco	US	United States of America
DK	Denmark	MG	Madagascar		
ES	Spain				

- 1 -

MANAGEMENT INTERFACE AND FORMAT FOR LICENSE MANAGEMENT SYSTEM

BACKGROUND OF THE INVENTION

15 This invention relates to methods of operation of computer systems, and more particularly to a method and system for managing the licensing of software executed on computer systems.

20 In U.S. Patent 4,937,863, issued to Robert, Chase and Schafer and assigned to Digital Equipment Corporation, the assignee of this invention, a Software Licensing Management System is disclosed in which usage of licensed software may be monitored in a computer system to determine if a use is within the scope of a license. The system maintains a database of licenses for software products,

- 2 -

delivering the license document may be in the form of a network, or may be a phone line using modems, or may include physical delivery by disks or CD ROMs, for example. Likewise, the method of delivery of the software products being licensed, i.e., the applications programs 17 to be executed on the CPUs 16, is not material to the license management facility of the invention; the products are delivered by some appropriate means, e.g., the communications link 30 and the networks 21 and 22, by CD ROMs or disks physically distributed, etc.

Although shown in Figure 1 as operating on a distributed system, in the simplest case the license management facility of the invention may be operated on a single CPU. The license management program 11 and the applications program 17 may be executing on the same CPU, in which case the license document would be stored in a database 23 as before, on this CPU, and the calls from the unit 18 to the license server would be local instead of RPCs. As in the distributed system, however, the licensed product would still not have access to the license document, but instead could only make inquiries to the server program, even if all are executing on the same CPU.

In operation of the distributed system of Figure 1, the producer 28 gives the issuer 25 authority to grant licenses on its behalf (the producer and issuer can be a single entity or multiple entities). The license document generator program 26, under control of a user (a person), generates a license (usually the result of negotiation between the user of program 26 and a user of the server 10). This license is called a product use authorization, and it is transmitted by the link 30 to the server 10. The license management program in the server 10 stores the product use authorization in the database 23, and, if delegation is an authorized option, may distribute parts of the authorized use to the delegatee servers 13.

- 3 -

where it is likewise stored in a database. Thereafter, administration of the license is only in response to inquiries from user nodes 16. When execution of a program 17 begins, the unit 18 is invoked to check on the availability of a license for this particular node. The unit 18 sends (as by an RPC) a request to the license management program 14 (or 11 if there is no delegatee), where the product use authorization stored in database 23 is checked to see if use is authorized. If so, a return is sent to the user node 16, granting permission to continue. When the program 17 has finished executing, the unit 18 again is invoked to signal to the license management program, again by an RPC, that the authorization is released, so the license management program can take appropriate action, e.g., log the use in log 24, etc.

To implement these operations, the license management program 11 or 14 contains several functions, including a client interface 31, a database interface 32, a management interface 33, and an interserver interface 34 for communicating with the delegates 13 (if any). The client interface 31, as described below, handles the requests received from the user nodes 16, and returns resulting from these requests. The database interface 32 handles the storing and retrieval of license information in the database 23, and logging license usage activity to log 24, and retrieval of this data. The management interface 33 handles the tasks of receiving the product use authorizations from the issuer 25 and maintaining the database 23 via the database interface 32. The interserver interface 34 handles the task of communicating with the delegatee servers 13, including transmitting the assigned parts of the product use authorizations, or communicating with other license servers that may be separately executing the license management function; for example, calls for validating calling cards may be made to another such server.

- 4 -

If there are no delegates or no other license servers, then of course the interserver interface 34 has no function, and is idle.

5 The license document or "product use authorization" forming the basis for the license management activity of the program 11 on the server 10 may be illustrated as a data structure containing the information set forth in Figure 2; in actual practice the product use authorization is preferably a more abstract data arrangement, not in such a rigidly structured format as illustrated. For example, the product use authorization as well as similar documents stored in the database 23, or passed between components of the system of Figure 1, may be of the so-called tag-length-value data format, where the data structure begins with an identifying tag (e.g., PUA or product use authorization) followed by a field giving the length, followed by the value itself (the content). One type of data treatment using this tag-length-value format is an international standard referred to as ASN.1 or Abstract Syntax Notation. In any event, the document 35 illustrated in Figure 2 is merely for discussing the various items of data, rather than representing the way the information is stored. Some of the fields shown here exist at some times and not others, and some are optional; the product use authorization may also include additional fields not shown or discussed here. Also it should be noted that copies of parts of this type of document are made for the delegates, so this representation of Figure 2 is a composite of several documents used in the system of Figure 1. The document 35 includes fields 36 identifying the software product by product name, producer, version numbers, release date, etc. The issuer 25 is identified in field 37, and the licensee (usually the owner of the license server 10) identified in field 38. The essential terms of the license grant are then defined in fields 40-46. The start date and end date are specified in fields 40; these store the exact time (date, hour, minute, second, etc.) when the license becomes valid and

- 5 -

when it ends, so licenses may be granted to start at some future time and to end at a particular time. Note that the previous practice has been to specify only the ending date, rather than also a start date as employed here. Each of the nodes, including issuer 25, servers 10 and 13, and user nodes 16, maintain a time value by a local clock referenced to a standard, so inherent in the license management facility is the maintaining of a time standard to compare with the start and end date information in the fields 40. The units granted are specified in field 41; the units are an arbitrary quantitative measure of program usage. In a delegatee server 13, the units field 41 will have some subset of the units field in the original product use authorization. As units are granted to users 16 or delegated, the remaining units available for grant are indicated in a subfield 42 in the copy of the document used by the server. The management policy occupies fields 43-46, and includes style, context, duration and LURDM (license use requirements determination method), as will be explained. The style field 43 specifies whether the licensed units are controlled by an "allocative" style or "consumptive" style, or some other "private" algorithm, where styles are ways used to account for the consumption or allocation of the units. The context field 44 specifies the location and environment in which product use or license management occurs, i.e., a CPU or an individual user or a network, etc. Duration field 45 indicates whether the license granted to a user is by assignment, by transaction, or immediate. The LURDM field 46 indicates the license use requirements determination method, in some cases using a license use requirements table (LURT) seen as field 47, as will be described.

Additional fields 48-54 in the product use authorization 35 of Figure 2 define features such as delegation authorization, calling authorization, overdraft

- 6 -

authorization, combination authorization, token, signature, checksum, etc. These will be described in the following paragraphs.

5 If the delegation field 48 is true, a license server 10 may distribute license units to multiple servers 13. A time limit may be imposed, i.e., units can be delegated to other hardware systems until they time out. Delegation allows an administrator to distribute units to improve response time and increase the resilience of the system. For example, the communication network 21 may include a satellite link to a remote facility where the local server 13 has a number of clients or users 16, in which case the calls to the server 13 would be completed much quicker than would be the case if calls had to be made to the server 10. Also, delegation may be used as a method of allocating licensed units within a budget for administrative purposes. Usually the delegation authorization is a feature that is priced by the issuer, i.e., a license granting 1000 units with delegation authorization is priced higher than without this authorization.

15 The field 49 contains a calling authorization and/or a caller authorization. If the caller authorization in field 49 is true, the product is permitted to receive calls from other named products requesting use of the product, and if conditions are met (identified caller is authorized) the server can grant a calling card, as described below. If the calling authorization is true, the product can make calls to other products. If neither is true, then the product can neither make or receive calls using the calling card feature. Referring to Figure 1, if product 17a wishes to make a remote procedure call to a feature of product 17b running on a different user node 16, it makes a call to its server 13 including a request for a calling card, and, if permitted, the return to product 17a includes a calling card 25 49a. The product 17a then makes a call to product 17b in the usual manner of

- 7 -

5 RPCs, sending along the calling card 49a, which the product 17b then verifies by
a call to its server 13 before executing the called procedure and issuing its return
to product 17a. The feature of calling cards is important for distributed
applications. For example, if a product is able to execute faster in a distributed
10 system by assigning tasks to other CPUs, then the issue is presented of which
license policy is needed, i.e., does every node executing a part of the task have to
be licensed and consume or receive allocation of a unit, or just the one managing
the task? This is resolved for most applications by use of this calling card concept.
The product use authorization for such a product has the calling authorization
15 field 49 enabled, so calling cards can be issued. This feature is typically separately
priced.

 The combination authorization field 50 of Figure 2 determines whether or
not license requests from a user node 16 can be satisfied by combining units from
multiple product use authorizations. It may be advantageous to purchase licenses
15 with different policy values, and use units from certain product use authorizations
only for overflow or the like. Or, for other reasons, it may be advantageous to
"borrow" and "lend" units among delegated servers or user nodes. This function
is permitted or denied by the content of field 50.

 The overdraft field 51 determines whether or not a requested allocation
20 from a user node 16 will be nevertheless granted, even though the units available
field 42 is zero or too small to permit the requested use. Overdrafts can be
unlimited, or a specific overdraft pool can be set up by a server 10, for a
customer's internal administrative purposes. That is, the overdraft value may be
unlimited in the original license, but limited or zero for internally distributed
25 copies of the license. Thus, the product use authorization sent by the issuer 25 to

- 8 -

the customer may have overdrafts permitted by the field 51, but the customer may deny overdraft permission for its own budgeting purposes. In any event, if overdraft is permitted, additional fees have to be paid to the issuer at some accounting period, when the logged usage from log 24 indicates the available units have been exceeded. If overdraft is denied, then the units 18 of the user nodes making request allocations are structured to inform the products 17 that a license grant is not available. The intent is not to prevent the application program from running; the license server merely informs the application whether or not the license manager determines that it is authorized to run. The application can itself be structured to shut itself down if not authorized to run, or it can be structured to shut down certain functions (e.g., ability to save files, ability to print, etc.), or it can be structured to continue in a fully functional manner. The purpose of the license management facility is not that of enforcement, nor that of "copy protection", but instead is merely that of license management.

An optional token field 52 is available in the product use authorization 35 of Figure 2. This field can contain comments or other information desired by the issuer or user. For example, a telephone support number may be included in the token field, then when the product 17 shows its "help screen" the number is inserted. This number would be part of the argument, i.e., data transmitted to the user node 16, when the server 10 makes a return following a request allocation message from the user. This field may also be used to store information used in a "private" style, where the information from this field returned to the user node is employed by the application program 17 or the stub 19 to determine if the application can be activated.

- 9 -

The signature field 53 in the product use authorization 35 is a part of a validation mechanism which provides important features. This field contains a digital signature encoded to reflect the data in the license itself, as well as other encoding methods not known to customers, so it cannot be duplicated unless the encoding algorithm is known. In a preferred embodiment, a so-called "public/private key" system of encoding is used for the signature field 53. The encoding algorithm used to generate the signature 53 is known to the issuer 25, using a private key, and anyone knowing the public key can decode the signature to determine if it is valid but cannot determine the encoding algorithm so it cannot produce a forged signature. So, if the server 10 knows the public key which is unique to the issuer 25, it can determine if a license document 35 is genuine, but it cannot itself generate license documents. However, if the server possesses a valid license document that gives it the right to delegate, then it will be assigned its own private key (different from all other issuers or servers) and its delegates 13 will be able to determine if a valid delegated license is delivered to them as they will be given the public key for the servers 13. The field 53 will thus contain both the original signature from the issuer 25 and the license server's signature when delivered to a delegatee 13. The decoding algorithm using a public key for any signatures is thus used by the license server 10 or delegatee 13 to make sure a product use authorization 35 is authentic before it is stored in the database 23. Related to the digital signature 53 is a checksum field 54, which merely encodes a value related by some known algorithm to the data in the product use authorization 35 itself. This field may be used merely to check for corruption of the data as it is stored, recalled, and transmitted within the system. That is, the checksum is used for data validation rather than security.

- 10 -

Two concepts central to the license management system implemented using the license document or product use authorization 35 of Figure 2 are the "license units", specified in field 41 or 42 and the "context", specified in field 44. License units are an abstract numerical measure of product use allowed by the license. When a product 17 (or a function or feature of a product) makes a license-checking request, the license management program 11 on server 10 computes how many license units are required to authorize this particular use of the product, and this is the license units requirement, in some cases using the LURDM field 46. A "context" is a set of tagged values which define the location and environment in which product use or license management occurs. Context values may be specified in field 44 of the product use authorization 35 of Figure 2 to restrict the environments in which the license may be managed and in which product use may occur. A context template may also be specified in the field 44 to indicate which parts of the complete context of product use (sub-contexts) are significant in differentiating product uses for the purposes of unit allocation; when this is specified, it allows separate product uses to share license units in a controlled way.

The two general types of policies specified in field 43 are allocative and consumptive. An allocative policy grants to the holder a specific number of license units (field 41) and specifies the policy which must be used to account for the allocation of these units. A software product 17 which is being managed by an allocative license will require verification that the appropriate number of license units have been allocated to it prior to performing services to the user. Typically, this allocation of units occurs either at the time of activation of the product 17 or at the time that product use is enabled on a particular platform (user CPU 16). The units typically remain allocated to the product 17 throughout the period that the product is running or is enabled to run. Upon termination of

- 11 -

processing or disabling, the allocated units are deallocated and made available for allocation to other instances of the software product 17 (other users 16 activating the product). In general, as long as any license units remain unallocated in field 42, the holder of the license is contractually authorized to increase his utilization of the licensed product. The usage does not deplete the license, however, as the units are returned to the units-available field 42 after a user is finished, and can be granted again to another user.

A consumptive unit based license, indicated in policy field 43, grants to the holder a specific number of initial license units (from field 42) and specifies the policy used to account for the consumption of those units. A software product 17 which is being managed by a consumptive license will cause an appropriate number of license units to be consumed to reflect the services provided by the product. Once consumed, units cannot be reused. Thus, the number of units available for future use declines upon every use of the licensed software product 17. This may also be referred to as a "metered" policy, being conceptually similar to measured consumption of electricity, water, etc. When the number of available units in field 42 reaches zero, the license may require that further use of the product is prohibited, or, the agreement may permit continued decrementing of the number of available units; the result is the accumulation of a negative number of available units in the field 42. It is anticipated that most consumptive unit based licenses will consider negative units to represent an obligation of the license holder to pay the license issuer 25. The transaction log 24 maintains an audit trail for providing a record of the units used in a consumptive license.

Referring to Figure 3, the major elements of the management policy are set forth in a table, where the possible entries for the fields 43, 44, 45 and 46 are

- 12 -

listed. For the style entry 43, the possibilities are allocative and consumptive as just described, plus a category called "private" which represents a style of management undefined at present but instead to be created especially for a given product, using its own unique algorithm. It is expected that most licenses may be administered using the named alternatives of Figure 3, but to allow for future expansion to include alternatives not presently envisioned, or to permit special circumstances for unique software, the "private" choices are included, which merely mean that the product 17 will generate its own conditions of use. It is important to note that, except for the "private" alternative, the license management is totally in control of the license management program 11 on the license server 10 (or delegatee 13), rather than at the product 17. All the product 17 does, via the unit 18, is to make the request inquiry to the server 10 via the client interface 31, and report when finished.

The context field 44 specifies those components (sub-contexts) of the execution-context name which should be used in determining if unit allocations are required. License data is always used or allocated within, or for the benefit of, some named licensing context, and context can include "platform contexts" and "application contexts". Platform contexts are such things as a specific network, an execution domain, a login domain, a node, a process ID or a process family, a user name, a product name, an operating system, a specific hardware platform, as listed in Figure 3. Applications contexts are information supplied from the application (the product 17), such as may be used in a "private" method of determining license availability. The context name can use several of these, in which case the context name is constructed by concatenating the values of all subcontexts into a single context name, e.g., a VAX 3100 platform using VMS operating system.

- 13 -

The duration field 45 defines the duration of an allocation of license units to a specific context or the duration of the period which defines a valid consumptive use. For durations of type "Assignment," the specification of a reassignment constraint is also provided for, as discussed below. There are three types of duration, these being "transaction," "assignment" and "immediate" as seen in Figure 3.

The transaction duration type, when specified for an allocative policy, indicates that license units should be allocated to the specified context upon receipt of a license request and that those units should be deallocated and returned to the pool of available units upon receipt of a corresponding license release from a user node 16. Abnormal termination of the process or context having made the original license request will be semantically equivalent to a license release. On the other hand, when specified for a consumptive policy, this duration type indicates that license units should be allocated to the specified context upon receipt of a license request and permanently removed from the available units pool (field 42) upon receipt of a license release which reflects successful completion of the transaction. Upon receipt of a license release which carries an error status or upon abnormal termination of the processor context having made the original license request, the allocated units will be deallocated and returned to the pool of available units (field 42).

The assignment duration type in Figure 3 (field 45 of Figure 2) imposes the constraint that the required units must have been previously assigned to a specific context. The sub-contexts which must be specified in the assignment are those given in the context-template. A "reassignment constraint" may be imposed, and this is a limitation on how soon a reassignment can be made. For example, a

- 14 -

reassignment constraint of 30-days would require that units assigned to a specific context could not be reassigned more often than every 30-days; this would prevent skirting the intent of the license by merely reassigning units whenever a user of another context made a request allocation call for the product. Related to this assignment constraint, a "reallocation limit" may also be imposed, to state the minimum duration of an allocation; where there is a context template of process, the intent is to count the number of uses of the software product at a given time, but where software runs in batch rather than interactive mode it may run very quickly on a powerful machine, so a very few concurrent uses may permit almost unlimited usage - by imposing a reallocation constraint of some time period, this manner of skirting the intent of the license may be constrained.

The immediate duration type (field 45 of Figure 2) is used to indicate that the allocation or consumption of an appropriate number of license units from the pool of available units (field 42) should be performed immediately upon receipt of a license request. Receipt of license release or abnormal terminations will then have no impact on the license management system. When specified as the duration for an allocative policy, the effect will be simply to check if an appropriate number of license units are available at the time of a license request. When specified as the duration for a consumptive policy, the effect will be to deduct the appropriate number of license units from the available pool at the time of a license request, and, thereafter, abnormal termination, such as a fault at the user CPU 16 or failure of the network link, will not reinstate the units.

The LURDM or license unit requirement determination method, field 46, has the alternatives seen in Figure 3 and stores information used in calculating the number of units that should be allocated or consumed in response to a license

- 15 -

request. If this field specifies a table lookup kind, this means license unit requirements are to be determined by lookup in the LURT (field 47) which is associated with the current license. If a constant kind is specified, this indicates that the license units requirements are constant for all contexts on which the licensed product or product feature may run. A private LURDM specifies that the license unit requirements are to be determined by the licensed product 17, not by the license management facility 11. The license unit requirements tables (LURTs) provide a means by which issuers of licenses can store information describing the relation between context (or row selector) and unit requirements. The license units requirements determination method (LURDM) must specify "table lookup" for the LURT to be used, and if so a row selector must be specified, where a valid row selector is any subcontext, e.g., platform ID, user name, time of day, etc. An example of an LURT fragment is shown in Figure 4, illustrating the license unit requirements table mechanism. In this example, the row selector is "platform-ID" so the platform-ID value determines which row is used. The issuer of this LURT of Figure 4 has established three unit requirement tiers for use in determining the unit requirements for that issuer's products. The reason for the tiers is not mandated by the license management system, but the issuer 25 (actually the user of the program 26) would probably be establishing three pricing tiers, each reflecting a different perspective on the relative utility of different platforms in supporting the use of various classes of product 17. The first column in Figure 4, Column A, specifies the use requirements for a class of products whose utility is highly sensitive to the characteristics of the specific platform on which they are run. This can be seen by observing that the unit requirements are different for every row in Column A. Products which use the second column (Column B) appear to have a utility which is more related to the class of platform on which they run. This is indicated by the fact that all the PC

- 16 -

platforms share a single value which is different from that assigned to the VAX platform. The final column (Column C) is for use with a class of products which is only supported on the VAX platform. Figure 4 is of course merely an example, and the actual LURT created by the license document generator 26 and stored in the license database 23 (as field 47 of the product use authorization 35) can be of any content of this general format, as desired by the license issuer.

Instead of always selecting the rows in LURT tables according to the platform ID of the execution platform, in order to handle the breadth of business practices that need to be supported by the license management facility, the LURT mechanism is extended by providing a "row selector" attribute in the LURT class structure. No default is provided although it is expected that the normal value for the row selector attribute will be "platform ID."

In the system of patent 4,937,863, a concept similar to that of the LURT of Figure 4 was provided, with rows selected by the platform ID and columns selected by some arbitrary means, typically according to product type. The system of this invention allows flexibility in the selection of both LURT row and column while continuing to provide backwards compatibility for licenses defined within the constraints of patent 4,937,863.

Some examples will illustrate potential uses for the row selector attribute. A customer may only want to pay for the use of a product during one or two months of the year; the product may be FORTRAN and the reason for this request may be that the company has a fairly stable set of FORTRAN subroutines that are given regular "annual maintenance" only during the months of May and June. To handle this customer's needs, the FORTRAN product would generate

- 17 -

an application subcontext which would contain a value representing the month of the year. Then, a LURT table would be defined with twelve rows, one for each month of the year. In some column, probably column A, a negative one (-1) would be placed in each month except for May and June. These two months would contain some positive number. The product use authorization would then have a LURDM field specifying a LURT for use to determine the units requirement, and would name this custom LURT table. The effect would be that the PUA could only be used during the months of May and June since negative one is interpreted by license managers to mean "use not authorized." This mechanism could also be used to do "time of day" charging. Perhaps charging fewer units per use at night than during the day. Also, if a subcontext was used that contained a year value, a type of license would be provided that varied in its unit requirements as time passed. For instance, it might start by costing 10-units per use in 1991 but then cost one unit less every year as time passed, eventually getting to the point where the unit requirement was zero.

Another example is font names. A specific customer may purchase a license giving it the right to concurrent use of 100-units of a large font collection; some of the fonts may cost more to use than others. For instance, Times Roman might cost 10-units per use while New Century Schoolbook costs 20-units per use. The problem is, of course, making sure that charges are properly made. The solution is to build a LURT table with a specified application subcontext as its row selector. A row is then created for each font in the collection and in Column A of the LURT, the number of units required to pay for use of the font would be specified. The print server would then specify the name of a font as the value of the application subcontext whenever it does an *lm_request_allocation()* call. This will allow charges to be varied according to font name.

- 18 -

5 A further example is memory size. Some products are more or less valuable depending on the size of memory available to support them. A software vendor wishing to determine unit requirements based on memory size will be able to do so by building LURT tables with rows for each reasonable increment of memory (probably 1-megabyte increments). Their applications would then sense memory size (using some mechanism not part of the license management facility) and pass a rounded memory size value to the license manager in a private context.

10 Other examples are environment and operating system. Some products may be valued differently depending on whether they are being run in an interactive mode or in batch. This can be accomplished by building LURT rows for each of the standard platform subcontexts that specify environment. Regarding operating system, it has been considered desirable by many to have a single product use authorization permit the use of a product on any number of operating systems, this conflicts with some vendors policies who do not want to have to create a single price for a product that applies to all operating systems. 15 Thus, if an operating system independent license were offered for a C compiler, the price would be the same on MS-DOS, VMS, and/or UNIX. Clearly, it can be argued that the value of many products is, in part, dependent on the operating system that supports them. By using a row selector of operating system (one of the standard platform subcontexts), license designers could, in fact, require 20 different numbers of units for each operating system. However, it might be more desirable to base the row selection on a private application subcontext that normally had the same value as the operating system subcontext. The reason for this is that the license designer might want to provide a default value for operating system names that were unknown at the time the LURT rows were defined. If 25 this is the case, the product would contain a list of known operating systems and

- 19 -

pass the subcontext value of "Unknown" when appropriate. The LURT row for "Unknown" would either contain a negative one (-1) to indicate that this operating system was unsupported or it would contain some default unit requirement.

5 Another example is variable pricing within a group. One of the problems with a "group" license is that there is only one unit requirements field on the PUA for a group. Thus, all members of the group share a single unit requirement. However, in those cases where all members of the group can be appropriately licensed with a constant unit requirement yet it is desired to charge different amounts for the use of each group member, a LURT can be built that has rows defined for each group member. The row selector for such a group would be the standard platform subcontext "product name."

10 Many different types of license can be created using different combinations of contexts, duration and policy from the table of Figure 3. As examples, the following paragraphs show some traditional licensing styles which can be implemented using the appropriate values of the product use authorization fields 43-46.

15 A "system license" as it is traditionally designated is a license which allows unlimited use of a product on a single hardware system. The correct number of units must be allocated to the processor in advance and then an unlimited product use is available to users of the system. The product use authorization would have in the context field 44 a context template for a node name, the duration field would be "assignment" and the policy style field 43 would be "allocative".

- 20 -

5 A "concurrent use" license is one that limits the number of simultaneous uses of a licensed product. Concurrent use license units are only allocated when the product is being used and each simultaneous user of the licensed product requires their own units. In this case the context template, field 44, is a process ID, the duration field is "transaction" and the policy style 43 is "allocative".

10 A "personal use" license is one that limits the number of named users of a licensed product. This style of licensing guarantees the members of a list of users access to a product. Associated with a personal use type of product use authorization there is a list of registered users. The administrator is able to assign these users as required up to the limit imposed by the product use authorization; the number of units assigned to each user is indicated by the LURDM. It may be a constant or it may vary as specified in a LURT. The context template is "user name", the duration is "assignment", and the policy is "allocative".

15 A "site license" is one that limits the use of a licensed product to a physical site. Here the product use authorization contains for the context template either "network name" or "domain name", the duration is "assignment" and the policy style field 43 is "allocative".

20 Generally, a license to use a software product is priced according to how much benefit can be gained from using the product, which is related to the capacity of the machine it will run on. A license for unlimited use on a large platform such as a mainframe, where there could be thousands of potential users at terminals, would be priced at a high level. Here the style would be "allocative", the context template = "node", the duration = "assignment" and the LURDM may be "Column A" - the units, however, would be large, e.g., 1000. At the other end

- 21 -

of the scale would be a license for use on a single personal computer, where the field values would be the same as for the mainframe except the units would be "1". If a customer wanted to make the product available on the mainframe but yet limit the cost, he could perhaps get a license that would allow only five users at any given time to use the product; here the fields in the product use authorization would be: units = 5; style = allocative; context template = process; duration = transaction; LURDM = constant, 1-unit. This would still be priced fairly high since a large number of users may actually use the product if a session of use was short. A lower price would probably be available for a personal use license where only five named persons could use the product, these being identified only in the license server 10, not named by the license issuer 25. Here the fields in the product use authorization are: units = 5; style = allocative; context template = user name; duration = transaction; LURDM = constant, 1-unit.

An additional feature that may be provided for in the product use authorization 35 is license combination. Where there are multiple authorizations for a product, license checking requests sent by user nodes 16 may be satisfied by combining units from multiple authorizations. Individual product use authorizations may prohibit combined use. Thus, a licensee may have a license to use a product 17 on an allocative basis for a certain number of units and on a consumptive basis for another number of units (this may be attractive from pricing standpoint); there might not be enough units available for a particular context from one of these licenses, so some units may be "borrowed" from the other license (product use authorization), in which case a combination is made.

The interface between the program executing on the client or user 16 and the license server 10 or its delegates 13 includes basically three procedure calls:

- 22 -

a request allocation, a release allocation and a query allocation. Figure 5 illustrates in flow chart form some of the events occurring in this client interface. The request allocation is the basic license checking function, a procedure call invoked when a software product 17 is being instantiated, functioning to request an allocation of license units, with the return being a grant or refusal to grant. Note that a product may use request allocation calls at a number of points in executing a program, rather than only upon start-up; for example, a request allocation may be sent when making use of some particular feature such a special graphics package or the like. The release allocation call is invoked when the user no longer needs the allocation, e.g., the task is finished, and this return is often merely an acknowledge; if the style is consumptive, the caller has the opportunity via the release allocation call to influence the number of units consumed, e.g., decrease the number due to some event. The query allocation call is invoked by the user to obtain information about an existing allocation, or to obtain a calling card, as will be described.

The request allocation, referred to as *lm_request_allocation()*, is a request that license units be allocated to the current context. This function returns a grant or denial status that can be used by the application programmer to decide whether to permit use of the product or product feature. The status is based on the existence of an appropriate product use authorization and any license management policies which may be associated with that product use authorization. License units will be allocated or consumed, if available, according to the policy statement found on the appropriate product use authorization. The product would normally call this function before use of a licensed product or product feature. The function will not cause the product's execution to be terminated should the request fail. The decision of what to do in case of failure to obtain allocation of license

- 23 -

units is up to the programmer. The arguments in a request allocation call are the product name, producer name, version, release date, and request extension. The product name, producer name, version and release date are the name of the software product, name of producer, version number and release date for specifically identifying the product which the user is requesting an allocation be made. The request extension argument is an object describing extended attributes of the request, such as units required, LURT column, private context, and comment. The results sent back to the calling node are a return code, indicating whether the function succeeded and, if not, why not, and a grant handle, returned if the function completes successfully, giving an identifying handle for this grant so it can be referred to in a subsequent release allocation call or query allocation call, for example.

The release allocation, referred to as *lm_release_allocation()*, is an indication from a user to the license manager to release or consume units previously allocated. This function releases an allocation grant made in response to a prior call to request allocation. Upon release, the license management style 38 determines whether the units should be returned to the pool of available units or consumed. If the caller had specified a request extension on the earlier call to request allocation which contained a units-required-attribute, and the number of units requested at that time are not the number of units that should be consumed for the completed operation, the caller should state with the units-consumed argument how many units should be consumed. The arguments of the release allocation are: grant handle, units consumed, and comment. The grant handle identifies the allocation grant created by a previous call to request allocation. The units-consumed argument identifies the number of units which should be consumed if the license policy is consumptive; this argument should only be used

in combination with an earlier call to request allocation which specified a units requirement in a request extension. Omission of this argument indicates that the number of units to be consumed is the same as the number allocated previously. The comment argument is a comment which will be written to the log file 24 if release units are from a consumptive style license or if logging is enabled. The result is a return code indicating if the function succeeded, and, if not, why not.

The query allocation, or *lm_query_allocation()*, is used by licensed products which have received allocations by a previous request allocation call. The query is to obtain information from the server 10 or delegatee server 13 about the nature of the grant that has been made to the user and the license data used in making the grant, or to obtain a calling card (i.e., a request that a calling card be issued). Typically, the item read by this query function is the token field 52 which contains arbitrary information encoded by the license issuer and which may be interpreted as required by the stub 19 for the licensed product software 17, usually when a "private" allocation style or context is being employed. The arguments in this procedure call are the grant handle, and the subject. The grant handle identifies the allocation grant created by a previous call to request allocation. The subject argument is either "product use authorization" or "calling card request"; if the former then the result will contain a public copy of the product use authorization. If this argument is a calling card request and a calling card which matches the previous constraints specified in that request can be made available, the result will contain a calling card. If the subject argument is omitted, the result will contain an instance of the allocation. The results of the query allocation call are (1) a return code, indicating whether the function succeeded, and, if not, why not, and (2) a result, which is either an allocation, a product use authorization or a calling card, depending on type and presence of the subject argument.

- 25 -

Referring to Figure 5, the flow chart shows the actions at the client in its interface with the server. When the software product 17 is to be invoked, the unit 18 is first executed as indicated by the block 60, and the first action is to make a request allocation call via the stub 19, indicated by the block 61. The client waits for a return, indicated by the loop 62, and when a return is received it is checked to see if it is a grant, at decision block 63. If not, the error code in the return is checked at block 64, and if a return code indicates a retry is possible, block 65, control passes back to the beginning, but if no retry is to be made then execution is terminated. If the policy is to allow use of the product 17 without a license grant, this function is separately accounted for. If the decision point 63 indicates a grant was made, the grant handle is stored, block 66, for later reference. The program 17 is then entered for the main activities intended by the user. During this execution of product 17, or before or after, a query allocation call can be made, block 67, though this is optional and in most cases not needed. When execution of the program 17 is completed, the grant handle is retrieved, block 68, and a release allocation call is made, block 69. A loop 70 indicates waiting for the return from the server, and when the return received it is checked for an error code as before, and a retry may be appropriate. If the release is successfully acknowledged, the program exits.

Referring to Figure 6, the actions of the server 10 or delegatee server 13 in executing the license management program 11 or 14, for the client interface, are illustrated in flow diagram form. A loop is shown where the server program is checking for receipt of a request, release or query call from its clients. The call would be a remote procedure call as discussed above, and would be a message communicated by a network, for example. This loop shows the decision blocks 71, 72 and 73. If a release allocation call is received, a list of products for which

- 26 -

authorizations are stored is scanned, block 74, and compared to the product identity given in the argument of the received call, block 75. If there is no match, an error code is returned to the client, block 76, and control goes back to the initial loop. If the product is found, the authorization is retrieved from the database 23, block 77 (there may be more than one authorization for a given product, in which case all would be retrieved, but only one will be referred to here) and all of the information is matched and the calculations made depending upon the management policy of Figures 3 and 4, indicated by the decision block 78. If a grant can be made, it is returned as indicated at block 79, or if not an error code is returned, block 80. If a release allocation call is received, indicated by a positive at the decision block 72, the grant handle in the argument is checked for validity at block 81. If no match is found, an error code is returned, block 82, and control passes back to the initial loop. If the handle is valid, the authorization for this product is retrieved from the database 23 at block 83, and updated as indicated by the block 84. For example, if the license management style is allocative, the units are returned to the available pool. Or, in some cases, no update is needed. The authorization is stored again in the database, block 85, and a return made to the client, block 86, before control passes back to the initial loop. If the decision block 73 indicates that a query allocation call is received, again the grant handle is checked at block 87, and an error code returned at block 88 if not valid. If the grant handle matches, the authorization is retrieved from the database 23, at block 89, and a return is made to the client giving the requested information in the argument, block 90.

The basic allocation algorithm used in the embodiment of the license management system herein described, and implemented in the method of Figures 5 and 6, is very simple and can handle a very large proportion of known license

- 27 -

unit allocation problems. However, it should be recognized that a more elaborate and expanded algorithm could be incorporated. Additions could be made in efforts to extend the allocation algorithm so that it would have specific support for optimizing unit allocation in a wider variety of situations. Particularly, sources of non-optimal allocations occurring when using the basic allocation algorithm are those that arise from combination and reservation handling.

The first step is formation of full context. The client stub 19 is responsible for collecting all specified platform and application subcontexts from the execution environment of the product 17 and forwarding these collected subcontexts to the license management server 13 or 10. The collection of subcontexts is referred to as the "full context" for a particular license unit allocation request.

The next step is retrieval of the context template. When the license manager receives an *lm_request_allocation()*, it will look in its list of available product use authorizations (PUA) to determine if any of them conform to the product identifier provided in the *lm_request_allocation()* call. The product identifier is composed of: product name, producer, version, release date. If any match is found, the license manager will extract from the matching PUA the context template. This template is composed of a list of subcontexts that are relevant to the process of determining unit requirements. Thus, a context template may indicate that the node-ID subcontext of a specific full context is of interest for the purposes of unit allocation. The context template would not specify any specific value for the node-ID; rather, it simply says that node-ID should be used in making the allocation computation.

The next step is masking the full context. Having retrieved the context template, the license manager will then construct an "allocation context" by filtering the full context to remove all subcontexts which are not listed in the context template. This allocation context is the context to be used in determining allocation requirements.

5

Then follows the step of determining if the request is new. The license manager maintains for each product use authorization a dynamic table which includes the allocation contexts of all outstanding allocations for that PUA (i.e., allocations that have been granted but have not yet been released). Associated with each entry in this table is some bookkeeping information which records the number of units allocated, the full context, etc. To determine if a recent *lm_request_allocation()* requires an allocation of units to be made, the license manager compares the new allocation context with all those allocation contexts in the table of outstanding allocations and determines if an allocation has already been made to the allocation context. If the new allocation context does not already exist in the table, an attempt will be made to allocate the appropriate number of units depending on the values contained in the LURDM structure of the PUA and any LURTs that might be required. If an allocation context similar to that specified in the new allocation request does exist in the table, the license manager will verify that the number of units previously allocated are equal to or greater than the number of units which would need to be allocated to satisfy the new allocation request. If so, the license manager will return a grant handle to the application which indicates that the allocation has been made (i.e., it is a "shared allocation" - the allocated units are shared between two requests.) If not, the license manager will attempt to allocate a number of units equal to the

10

15

20

25

difference between the number previously allocated and the number of units required.

5 The step of releasing allocations (Fig. 6, blocks 84-85) occurs when the license manager receives an *lm_release_allocation()* call; it will remove the record in its dynamic allocation table that corresponds to the allocation to be released. Having done this, the license manager will then determine if the allocation to be removed is being shared by any other allocation context. If so, the units associated with the allocation being released will not be released. They will remain allocated to the remaining allocation contexts. Some of the units might
10 be released if the license manager determines that the number of allocated units exceeds the number needed to satisfy the outstanding allocation contexts. If this is the case, the license manager will "trim" the number of allocated units to an appropriate level.

15 In summary, the two things that make this algorithm work are (1) the basic rule that no more than one allocation will be made to any single allocation context, and (2) the use of the context template to make otherwise dissimilar full contexts appear to be similar for the purposes of allocation.

20 The license designer's task, when defining basic policy, is then to determine which contexts should appear to be the same to the license manager. If the license designer decides that all contexts on a single node should look the same (context template = node-ID), then any requests that come from that node will all share allocations. On the other hand, a decision that all contexts should be unique (i.e., context template = process-ID) will mean that allocations are never shared.

- 30 -

and stores a unit value indicating the number of licensing units for each product. When a user wishes to use a licensed product, a message is sent to the central license management facility requesting a license grant. In response to this message, the facility accesses the database to see if a license exists for this product, and, if so, whether units may be allocated to the user, depending upon the user's characteristics, such as the configuration of the platform (CPU) which will execute the software product. If the license management facility determines that a license can be granted, it sends a message to the user giving permission to proceed with activation of the product. If not, the message denies permission.

While the concepts disclosed in the patent 4,937,863 are widely applicable, and indeed are employed in the present invention, there are additional functions and alternatives that are needed in some applications. For example, the license management system should allow for simultaneous use of a wide variety of different licensing alternatives, instead of being rigidly structured to permit only one or only a few. When negotiating licenses with users, vendors should have available a wide variety of terms and conditions, even though a given vendor may decide to narrow the selection down to a small number. For example, a software product may be licensed to a single individual for use on a single CPU, or to an organization for use by anyone on a network, or for use by any users at terminals in a cluster, or only for calls from another specific licensed product, or any of a large number of other alternatives. A vendor may have a large number of products, some sold under one type of license and some under others, or a product may be a composite of a number of features from one or more vendors having different license policies and prices; it would be preferable to use the same license management system for all such products.

Distributed computing systems present additional licensing issues. A distributed system includes a number of processor nodes tied together in a network of servers and clients. Each node is a processor which may execute programs locally, and may also execute programs or features (subparts of programs) via the network. A program executing on one node may make remote procedure calls to procedures or programs on other nodes. In this case, some provision need be made for defining a license permitting a program to be executed in a distributed manner rather than separately on a single CPU, short of granting a license for execution on all nodes of a network.

In a large organization such as a company or government agency having various departments and divisions, geographically dispersed, a software license policy is difficult to administer and enforce, and also likely to be more costly, if individual licenses are negotiated, granted and administered by the units of the organization. A preferred arrangement would be to obtain a single license from the software producer, and then split this license into locally-administered parts by delegation. The delays caused by network communication can thus be minimized, as well as budgetary constraints imposed on the divisions or departments. Aside from this issue of delegation, the license management facility may best be operated on a network, where the licensing of products run on all nodes of the network may be centrally administered. A network is not necessary for use of the features of the invention however, since the license management can be implemented on a single platform.

Software products are increasingly fragmented into specific functions, and separate distribution of the functions can be unduly expensive. For example, a spreadsheet program may have separate modules for advanced color graphics, for

- 32 -

accessing a database, for printing or displaying an expanded list of fonts, etc. Customers of the basic spreadsheet product may want some, none or all of these added features. Yet, it would be advantageous to distribute the entire combination as one package, then allow the customer to license the features separately, in various combinations, or under differing terms. The customer may have an entire department of the company needing to use the spreadsheet every day, but only a few people who need to use the graphics a few days a month. It is advantageous, therefore, to provide alternatives for varied licensing of parts or features of software packages, rather than a fixed policy for the whole package.

Another example of distribution of products in their entirety, but licensing in parts, would be that of delivering CD ROMs to a customer containing all of the software that is available for a system, then licensing only those parts the customer needs or wishes to pay fees for rights to use. Of course, the product need not be merely applications programs, operating systems, or traditional executable code, but instead could also include static objects such as printer fonts, for example, or graphics images, or even music or other sound effects.

As will be explained below, calling and caller authorizations are provided in the system according to one feature of the invention, in order to provide technological support for a number of business practices and solve technical problems which require the use of what is called "transitive licensing." By "transitive licensing" is meant that the right to use one product or feature implies a right to use one or more other products or features. Transitive licenses are similar to group licenses in that both types of license consist of a single instrument providing rights of use for a plurality of products. However, transitive licenses differ from group licenses in that they restrict the granted rights by specifying that

- 33 -

the licensed products can only be used together and by further specifying one or more permitted inter-product calling/caller relationships. Some examples may help to clarify the use and nature of a transitive license: the examples to be explained are (1) two products sold together, (2) a give-away that results from narrow choices of licensing alternatives, (3) a client licensing method in a client/server environment, (4) impact of modular design, and (5) the impact of distributed design.

A software vendor might have two products for sale: the first a mail system, and the second a LEXISTM-like content-based text retrieval system. Each of these products might be valued at \$500 if purchased separately. Some customers would be satisfied by purchasing the rights to use only one of these products. others might find that they can justify use of both. In order to increase the likelihood that customers will, in fact, purchase both products, it would not be surprising if the software vendor offered his potential customers a volume discount, offering the two products for a combined price of \$800. The customers who took advantage of this combined offer would find that they had received two products, each of which could be exploited to its fullest capabilities independently from the other. Thus, these customers would be able to use the content based retrieval system to store and retrieve non-mail documents. However, from time to time, the vendor may discover that particularly heavy users of mail wish to be able to use the content based retrieval system only to augment the filing capabilities provided by the standard mail offering. It is likely that many of these potential customers would feel that \$800 is simply too much to pay for an extended mail capability. The vendor might then consider offering these customers a license that grants mail users the right to use the content-based retrieval system only when they are using mail and prohibits the use of content

- 34 -

based retrieval with any other application that might be available on the customers system. This type of license is referred to below a "transitive license," and it might sell for \$600.

5 Another example is a relational database product (such as that referred to as Rdb™) designed for use on a particular operating system, e.g., VMS. This relational database product has two components: (1) A user interface used in developing new databases, and (2) a "run-time" system which supports the use of previously developed databases. The developers of the database product might spend quite a bit of effort trying to get other products made by the vendor of the database product to use it as a database instead of having those other products build their own product-specific databases. Unfortunately, the other product designers may complain that the cost of a run-time license for the database product, when added to the cost of licenses for their products, would inevitably make their products uncompetitive. Thus, some mechanism would be needed that would allow one or another of the vendor's products to use the run-time system for the relational database product in a "private" manner while not giving unlicensed access to products of other vendors. No such mechanism existed, prior to this invention; thus, the vendor might be forced to sell the right to use its run-time system for the database product with its proprietary operating system license. 10 Clearly, this combined license would make it possible for the vendor's products to use its database product without increasing their prices; however, it also would make it possible for any customers and third-parties to use the database product without paying additional license fees. However, had the system of the invention been available, the vendor could have granted transitive licenses for the run-time component of its database product to all the vendor's products. Essentially, these 15 licenses would have said that the database run-time could be used without an

20
25

- 35 -

additional license fee if and only if it was used in conjunction with some other of the vendor's products. Any customer wishing to build a new relational database application or use a third-party application that relied on the vendor's database product would have had to pay the vendor for its database run-time license.

5 A proposed client/server licensing method provides yet another example of a problem which could be solved by transitive licensing. Typically, a client is only used by one user at a time, while a server can support an arbitrary number of clients depending on the level of client activity and the capacity of the machine which is supporting the server. While traditionally, server/client applications have
10 been licensed according to the number of clients that a server could potentially support, this may not be the most appropriate method for licensing when the alternatives afforded by the invention are considered. The business model for the proposed client/server method requires that each client be individually licensed and no explicit licensing of servers is required to support properly licensed clients.
15 Such a licensing scheme makes it possible to charge customers only for the specific number of clients they purchase. Additionally, it means that a single client can make use of more than one server without increasing the total cost of the system. The solution to this transitive licensing problem would be to provide a mechanism that would allow the clients to obtain license unit allocations and then pass a
20 "proof" of that allocation to any servers they may wish to use. Servers would then support any clients whose proofs could be verified to be valid. On the other hand, if a client that had not received a proof of allocation attempted to use a server, the server would obtain a license allocation for that client session prior to performing any services. Such a solution has not been heretofore available.

- 36 -

As the complexity and size of the software systems provided to customers increases, it is found that the actual solution provided to customers is no longer a single product. Rather, customers are more often now offered solutions which are built up by integrating an increasing number of components or products, each of which can often stand alone or can be part of a large number of other solutions. In fact, a product strategy may rely almost exclusively on the vendor's engineering and selling a broad range of specialized components that can only be fully exploited when combined together with other components into a larger system. Such components include the relational database runtime system mentioned above, mail transport mechanisms, hyperinformation databases, document format conversion services, time services, etc. Because these components are not sold on their own merits, but rather on their ability to contribute to some larger system, it is unlikely that any one customer will be receiving the full abstract economic value of any one of the components once integrated into a system. Similarly, it can be observed that the value of any component once integrated into a larger system varies greatly from system to system. Thus, it may be found that a mail transport mechanism contributes a large part of a system whose primary focus is mail, however, it will contribute proportionally less of the value of a system that provides a broader office automation capability. As a result of these observations, the job of the business analyst who is attempting to find the "correct" market price for each component standing on its own, is more complex. In reality, the price or value of the component can only be determined when considering the contribution of that component to the full system or solution in which it is integrated. Attempting to sell the components at prices based on their abstract, independent values will simply result in overpriced systems.

- 37 -

Transitive license styles are particularly suited to dealing with pricing of modular components, since component prices can be clearly defined in relation to the other components or systems which they support. Thus, a vendor can charge a price of \$100 for the right to use a mail transport system in conjunction with one product, yet charge \$200 for the use of the same mail transport system when used by another product.

In addition to the "business" reasons for wanting to support transitive licensing, there is also a very good technical reason that arises from the growing tendency of developers to build "distributed products" as well as the drive toward application designs that exploit either tightly or loosely coupled multiprocessor systems; the availability and growing use of remote procedure calls has contributed to this tendency. This technical problem can be seen to arise when considering a product which has a number of components, each of which may run in a different process space and potentially on a different computer system. Thus, there might be a mail system whose user interface runs on one machine, its "file cabinet" is supported by a second machine and its mail transport system runs on yet a third machine. The simple question which arises is: "Which of the three components should check for licenses?" Clearly it must be ensured that no single component can be used if a valid license is not present. Thus, the answer to the question will probably be that all three components should check for licenses. However, the question is then presented: "Where are the licenses to be located?". This can become more complex.

Increasingly, the distributed systems being built are being designed so that it is difficult to predict on which precise machine any particular component will run. Ideally, networks are supposed to optimize the placement of functions

- 38 -

5 automatically so that the machine with the most available resource is always the one that services any particular request. This dynamic method of configuring the distribution of function servers on the network makes it very difficult for a system or network manager to predict which machines will run any particular function and thus very difficult for him to decide on which machines software licenses should be loaded.

10 Even if a system manager could predict which machines would be running the various application components and thus where the license units should be loaded, the situation would still be less than ideal. The problem arises from the fact that each of the components of the application would be independently making requests for license unit allocations. This behavior will result in a difficult problem for anyone trying to decide how many license units are required to support any one product. Given the mail example, the problem wouldn't exist if it were assumed that all three components (i.e., user interface, file cabinet, and transport system) were required by the design of the mail system to be in use simultaneously. If this were the case, it could be simply assumed that supporting a single activation of the mail system would require three units. However, in a real mail system, it will be inevitably discovered that many users will only be using just the user-interface and file-cabinet components of the system at one time. Thus, there will be some unused units available which could be used to authorize additional users. This situation might not be what is desired by the software vendor.

20 The problem of providing license support to multi-component products which are dynamically configured could be solved by viewing each of the product components as a distinct licensable product and by treating the problem as one

- 39 -

of transitive licensing, but a mechanism for accomplishing this has not been available. Essentially, a single license document would be created that stated that if any one of the components had successfully obtained a license to run, it could use this grant to give it the right to exploit the other components. Thus, in the example above, the user might start the mail system by invoking its user interface. This user interface code would then query the license management facility for a license allocation and once it has received that allocation, it would pass a proof of allocation to the other mail components that it uses. Each of the other components would request that the license management system validate that the "proof" is valid prior to performing any service; however, none of the other components would actually require specific allocations to be made to them. In this way, the complexity of licensing and managing networks of distributed applications can be significantly reduced.

SUMMARY OF THE INVENTION

In accordance with one embodiment of the invention, a license management system is used to account for software product usage in a computer system. The system employs a license management method which establishes a management policy having a variety of simultaneously-available alternative styles and contexts. A license server administers the license, and each licensed product upon start-up makes a call to the license server to check on whether usage is permitted, in a manner similar to that of patent 4,937,863. The license server maintains a store of the licenses, called product use authorizations, that it administers. Upon receiving a call from a user, the license server checks the product use authorization to determine if the particular use requested is

- 40 -

permitted, and, if so, returns a grant to the requesting user node. The license server maintains a database of product use authorizations for the licensed products, and accesses this database for updating and when a request is received from a user. While this license management system is perhaps of most utility on a distributed computer system using a local area network, it is also operable in a stand-alone or cluster type of system. In a distributed system, a license server executes on a server node and the products for which licenses are administered are on client nodes. However, the license management functions and the licensed products may be executing on the same processor in some embodiments.

The product use authorization is structured to define a license management policy allowing a variety of license alternatives by components called "style", "context", "duration" and "usage requirements determination method". The style may be allocative or consumptive. An allocative style means the units of the license may be allocated temporarily to a user when a request is received, then returned to the pool when the user is finished, so the units may be reused when another user makes a request. A consumptive style means the units are deducted from an available pool when a user node makes a valid request, and "consumed", not to be returned for reuse. The context value defines the context in which the use is to be allowed, such as on a particular network, by a particular type of CPU, by a particular user name, by a particular process, etc. The duration value (used in conjunction with the style component) concerns the time when the license units are to be deducted from the available pool of units, whether at the time of request, after a use is completed, etc. A usage requirements determination method may be specified to define or provide information concerning the number of license units charged in response to a license request from a user node; for example, some CPU platforms may be charged a larger number of license units

than others. A table may be maintained of usage requirements, and the determination method may specify how to access the table, for example. The important point is that the user node (thus the software product) can only make a request, identifying itself by user, platform, process, etc., and the license management facility calculates whether or not the license can be granted (that is, units are available for allocation), without the user node having access to any of the license data or calculation. There is a central facility, the license server, storing the license documents, and, upon request, telling the licensed products whether they can operate under the license terms.

An important feature of one embodiment is that the license administration may be delegated to a subsection of the organization, by creating another license management facility duplicating the main facility. For example, some of the units granted in the product use authorization may be delegated to another server, where the user nodes serviced by this server make requests and receive grants.

The license management facility cannot create a license itself, but instead must receive a license document (a product use authorization) from an issuer of licenses. As part of the overall license management system of the invention, a license document generator is provided which creates the product use authorizations under authority of the owner of the software, as negotiated with customers. Thus, there are three distinct rights in the overall license management facility of the invention: (1) the right to issue licenses, (2) the right to manage licenses, and (3) the right to use the licensed products. Each one of these uses the license document only in prescribed ways. The license issuer can generate a license document. The license manager (or license server as referred to herein) can grant products the right to use under the license, and can delegate parts of the

- 42 -

5 licensed units for management by another server, as defined by the license document; the way of granting rights to products is by responding to certain defined calls from the products. And, the licensed products can make certain calls to the license server to obtain grants of rights based upon the license document, inquire, or report, but ordinarily cannot access the document itself.

10 As explained above, transitive licensing is an important feature of one embodiment. This is the provision of a mechanism for one user node to get permission to use another software product located on another user node; this is referred to as a calling authorization and a caller authorization, using a "calling card," and these are examples of the optional features which must be specifically permitted by the product use authorization. A user node must obtain permission to make a procedure call to use a program on another node; this permission is obtained by a request to the license server as before, and the permission takes the form of a calling card. When a calling card is received by a second node (i.e.,
15 when the procedure call is made), a request is made by the second node to the license server to verify (via the product use authorization) that the calling card is valid, and a grant sent to the user node if allowed. In this manner, all nodes may have use of a program by remote calls, but only one consumes license units.

20 Another important feature of one embodiment is a management interface which allows a license manager to modify the license policy components of a license document maintained by at a license server in its database. Usually the license manager can only make modifications that restrict the license policy components to be more restrictive than originally granted. Of course, the management interface is used to make delegations and assignments, if these are
25 authorized.

- 43 -

The license document interchange format is an important feature, in that it allows the license management system to be used with a wide variety of software products from different vendors, so long as all follow the defined format. The format uses data structures that are defined by international standards.

5 An important function is the filter function, used in the management interface and also in the client interface to select among elements in the data structures.

BRIEF DESCRIPTION OF THE DRAWINGS

10 The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, however, as well as other features and advantages thereof, will be best understood by reference to the detailed description of specific embodiments which follows, when read in conjunction with the accompanying drawings, wherein:

15 Figure 1 is a diagram in block form of a distributed computer system which may be used to implement the license management operations according to one embodiment of the invention;

 Figure 2 is a diagram of the content of a license document or "product use authorization" generated by the license document generator and stored by the license server in the system of Figure 1;

- 4 -

Figure 3 is a diagram of the alternatives for license style, context and duration making up the license management policy implemented in the system of Figure 1, according to one embodiment of the invention;

5 Figure 4 is a diagram of an example of a fragment of a license use requirements table (LURT) used in the system of Figure 1, according to one embodiment of the invention;

Figure 5 is a logic flow chart of a program executed by a user node (client), in the system of Figure 1, according to one embodiment of the invention;

10 Figure 6 is a logic flow chart of a program executed by a license server, in the system of Figure 1, according to one embodiment of the invention; and

Figure 7 is a diagram of the calls and returns made in an example of use of calling cards in the system of Figure 1.

Figure 8 is a diagram of an LDIF document identifier, according to an standard format;

15 Figure 9 is a syntax diagram of an LDIF document;

Figure 10 is a diagram of an LDIF document structure;

Figures 11, 13, 15, 17, 18, 19, 21-28 and 31-43 are syntax diagrams for elements of various ones of the LDIF data structures;

- 45 -

Figure 16 is a diagram of a license data structure;

Figures 12, 14 and 20 are examples of descriptions of data elements using a standard notation;

Figures 29 and 30 are examples of context templates used in the license management system;

Figures 44 and 45 are tables of attributes specific to filter and filter item type; and

Figure 46 is notation in a standard format for an example of a filter.

DETAILED DESCRIPTION OF SPECIFIC EMBODIMENTS

Referring to Figure 1, a license management facility according to one example embodiment of the invention is centered around a license server 10, which typically includes a CPU located in the customer's main office and executing a license management program 11 as will be described, under an operating system 12. The license server 10 communicates with a number of delegates 13 which likewise include CPUs in departments or divisions of the company or organization, each also executing a license management program 14 under an operating system 15. The license management program 14 is the same as the program 11 executing on the main server 10; the only difference in the functions of server 10 and servers 13 is that the latter have a delegated subset of the license units granted to the server 10, as will be described. The CPUs 13 are in turn servers for a number of

- 46 -

users 16, which are CPU nodes where the licensed programs 17 are actually executed. The programs 17 executing on the user CPUs 16 are applications programs (or operating systems, etc.) which have added to them units 18 and 19, according to the invention, allowing them to make inquiry to the their server 13 (or 10) before executing and to report back after executing, using a client stub 19 in the manner of remote procedure calls, in one embodiment. A user node 16 may have many different programs 17 that may be executed, and the various user nodes 16 would usually each have a set of programs 17 different from the other user nodes, all of which would be administered by the license management program 14 or 11. The terms "program" and "licensed product" are used in reference to the element 17, but it is understood that the products being administered may be segments of programs, or functions or features called by another program, or even merely data (such as printer fonts), as well as complete stand-alone applications programs. The license server 10 communicates with the delegatee servers 13 by a network 21, as is usual in large organizations, and the delegatee servers 13 each communicate with their user nodes 16 by networks 22; these networks may be of the Ethernet, token ring, FDDI types or the like, or alternatively, the user nodes 16 may be merely a cluster of terminals on a multiuser system with the delegatee being a host CPU. The particular hardware construction of the user nodes, server nodes, communication networks, etc., and the operating systems 12 or 15, are of no concern regarding the utility of the features of the invention, the only important point being that the user CPUs 16 of the software products 17 in question can communicate readily and quickly with their respective server nodes 13 or 10. In one embodiment, remote procedure calls (RPCs) are used as the communication medium for the interfaces between components of the system, handling the inquiries and grants as will be described.

- 47 -

A remote procedure call is similar to a local procedure call but is made to a procedure located on a remote node, by way of a communications network.

5 The function of the unit 19 is that of a client stub, in a remote procedure call sense. The calls to the license server 10 are made through this stub 19, and returns are received by the stub 19 and passed on to the program 17. The stub 19 is responsible for obtaining the network addresses of other nodes on the network, such as the server 10. Also, the stub 19 is responsible for determining the context (as defined below) for passing on to the server 10. The unit 18 functions to execute a "private" type of license availability determination if this is used, rather than this task being done by the application program 17, but if the ordinary method of determination is employed (using the license server) as is usually the case, the unit 18 is merely code that starts the execution and passes calls and returns back and forth between the program 17 and the unit 19.

15 The license server 10, using the license management program 11, maintains a license data file 23 comprising a number of license documents or licenses (product use authorizations), and also maintains a log 24 which is a record of the usage activity of all of the user CPUs 16 of each of the licensed programs. The delegatee servers 13 would maintain similar license databases and logs. The license server 10 has no authority to originate a license, but instead must receive a license from a license issuer 25. The issuer 25 is again a CPU executing a license document generator program 26 under an operating system 27. The license issuer 25 may be under control of the producer 28 of the programs or software products being licensed, or may be controlled by a distributor who has received the authority to grant licenses from the producer or owner 28. The communications link 30 between the license issuer 25 and the license server 10 for

- 48 -

5 This mechanism permits the system of the invention to dispose of the cumbersome, explicit support of license types having different scope such as the cluster licenses, node licenses, and process licenses found in prior license management systems including that of patent 4,937,863. Instead of defining a limited set of scopes (cluster, node, etc.), the system of this invention provides a general mechanism which allows an effectively unlimited range of allocation scopes to be defined.

10 Transitive licensing, as referred to above, is supported by the system of the invention by (1) calling authorizations, which are statements made in field 49 of the product use authorization 35 for one product (the "caller") to permit that product to call another product (the "callee"), and, (2) caller authorizations, which are statements made in field 49 of the product use authorization for one product (the "callee") to permit it to be called by another product (the "caller").

15 If calling or caller authorizations are to be exploited by products, then whenever one product calls another product, it must pass the callee a calling card 49a. This calling card 49a is an encoding of an identification of the caller as well as a statement by the license management system that a license unit allocation has been made to the caller which is passing the calling card. This calling card is then passed by the callee to the license management system for validation and, if the
20 either the product use authorization of the caller carries an appropriate calling authorization or the product use authorization of the callee carries an appropriate caller authorization, the use of the callee by the caller will be authorized without requiring any additional license unit allocations.

.. 49 -

Referring to Figure 7, the intercomponent interactions that occur when either calling or caller authorizations are being used are illustrated. This figure shows a license management server 10, a caller product 17a named "Product-1" and a callee product 17b named "Product-2". When Product-1 starts to run, it will make an *lm_request_allocation()* call to the license management server 10 to obtain a grant handle for an allocation of some number of units of the Product-1 license. Either immediately, or at some later time, but always prior to making a call to Product-2, Product-1 will call *lm_query_allocation()*, passing the grant handle received earlier and specifying that it wants a calling card for the product named "Product-2." If the field 49 of the product use authorization 35 used to satisfy the grant represented by the grant handle carries a calling authorization in field 49 naming "Product-2," the license manager will create a calling card 49a which includes the statement that a calling authorization exists and pass this calling card back to Product-1. If the calling authorization does not exist, the calling card passed to Product-1 will contain a statement to that effect.

Once Product-1 has successfully obtained a calling card 49a from the license manager, it will then make a call to Product-2, passing the calling card along with any other initialization parameters that would normally be used when starting Product-2. Product-2 will then pass that calling card to the license manager as part of its *lm_request_allocation()* call and the license manager will determine if the calling card is valid. Note that calling cards become invalid once the process which received the calling card makes an *lm_release_allocation()* call or terminates abnormally. If the calling card is valid, and it indicates that a calling authorization is present, the license manager will verify this statement and if found to be true, will return a grant handle to Product-2. If, on the other hand, the calling card carries an indication that no calling authorization is present, the

- 50 -

5 license manager will attempt to find a product use authorization for Product-2 that contains a caller authorization naming Product-1 as an authorized caller. If the caller authorization is found, a grant handle will be passed back to Product-2. If not, the license manager will ignore the calling card and proceed with the normal *lm_request_allocation()* logic.

10 The requirement to be passing calling cards between products requires that both the caller and the callee be "aware" of the fact that calling and caller authorizations may be used. This is one of the few examples of a requirement for a product 17 to become actively involved in the licensing problem when using the licensing management system of the invention. However, since the use of calling/caller authorizations is a fairly "sophisticated" and powerful feature, it is considered acceptable to impose this burden on application coders.

MANAGEMENT INTERFACE

15 Referring to Figure 1, the license management program 11 executing on a server 10 includes a license management interface 33 which functions to allow a user at a console for the server 10 CPU or at a remote terminal to implement certain necessary operations. The management interface 33 is essentially the tools or mechanisms available to the license manager at the licensee's site to (a) load the various licenses received from issuers 25 into the database 23 and make them
20 available for request allocation calls from the users, (b) remove the licenses from the machine when expired, (c) to make delegations if permitted, (d) to make assignments, (e) to make reservations, etc. Whatever the license manager is allowed to do to modify the license for his special circumstances (within the

- 51 -

original grant, of course), he does it by the mechanism of the management interface 33. Some licenses are not modified at all, but merely loaded. In a multiple machine environment, as on a network, there is considerable modification, as it is necessary to make sure the correct number of units are distributed onto the correct machines, the right people have access, other people don't have access, etc. Thus, in a network environment, there is extensive use of the management interface 33.

In reference to the terminology used in describing the management interface, as well as the license management system in general, it is helpful to note that the documentation conventions, data declarations, macro declarations, etc., for the object management used in one embodiment of the invention are according to the standards set forth in *OSI Object Management API Specification, Version 2.0*, X.400 API Association and X/Open Company Limited, 24 August 1990, a published document.

The specific operations available to the management interface 33 are to allow a manager to open and close a management session, register (load) objects in the license database 23, obtain a list of objects in the license database 23, and control a cursor (a cursor is a movable pointer to a member of a list of items). Once an object in the license database 23 is identified with the cursor, certain changes may be made in the object by a write function. For example, certain fields of a license document of Figure 2 or an LURT of Figure 4 may be changed in only specified ways as will be explained.

The operation of opening a session goes by the name of *lm_open_session()* and is used to establish a license management service session between a

- 52 -

management client and the service. Opening a session also creates a workspace to contain objects returned as a result of functions invoked within the session. Object management Objects can be created and manipulated within this workspace. Objects created within this workspace, and only such objects, may be used as Object arguments to the other license management service management functions used during the session established by a call to this function. More than one session may exist simultaneously.

The arguments that go with a *lm_open_session()* call are (a) the binding handle, which is binding information that defines one possible binding (a client-server relationship), and (b) a comment which will be inserted in the log file 24 if logging is enabled. The results from a *lm_open_session()* call are (a) a return code indicating whether the function succeeded, and, if not, why not, (b) a session, which is an established license management session between the management client and the license management service, and (c) a workspace that will contain all objects returned as a result of functions invoked in the session.

The close session call is referred to by *lm_close_session()* and functions to terminate the lm session. This function terminates the license service management session and makes the argument unavailable for use with other interface functions. The arguments that go with a *lm_close_session()* call are (a) the session which identifies the established lm session between the management client and the license management service, and (b) a comment which will be inserted in the log file if logging is enabled. The result of the call is a return code indicating whether the function succeeded, and, if not, why not.

- 53 -

The list function returns a set of selected objects in the license database 23, and uses the name *lm_list_licenses()*. This function is used to search the license database 23 and return a cursor which represents the first of one or more objects which match the specified filter. The specified filter will be applied to each object in the license database 23; all objects for which the filter evaluates true will be included in the object list accessible by the *set_cursor* function. The arguments that go with *lm_list_licenses()* are (a) session which identifies an established session between the management client and the license management service, and (b) a filter which is an object used to select license database 23 objects; license database objects will only be included in the object list headed by the cursor if they satisfy the filter - the constant no-filter may be used as the value of this argument if all license data objects are to be included in the object list. The results of the *lm_list_licenses()* call are (a) a return code indicating whether the function succeeded, and, if not, why not, and (b) a license list upon successful completion of this call containing a cursor which represents the first of one or more objects in the current license database 23 for which the specified filter evaluates true.

The register function is to register objects in the license database 23, and uses the name *lm_register()*. This function is used to register (i.e., load or create) new objects, or modify existing objects, in the license database 23; the objects which may be registered include only those which are subclasses of the license data class or history objects. The arguments are (a) session, which identifies an established session between the management client and the license management service, (b) license data object which is to be registered; if this argument is omitted, the comment argument is a required argument and a history object containing the comment will be registered in the license database 23, and (c)

- 54 -

comment, which will be inserted in the log file if logging is enabled. The result is a return code indicating whether the function succeeded, and, if not, why not. The errors possible when it does not succeed include data-expired, duplicate-object, no-such-session, memory-insufficient, network-error, etc., indicated by this return code.

The set cursor function establishes a new cursor, and is called by *lm_set_cursor()*. The arguments are (a) session, which identifies an established session between the management client and the license management service, (b) forward, which is a boolean value indicating if the direction in which the cursor is to be moved is forward or reverse, (c) filter which is used to eliminate cursors from the search for the next cursor that are not wanted; a new cursor will only be set if it satisfies the filter - the constant no-filter may be used as the value of this argument if any cursor is to be considered as the target cursor, and (d) the cursor which is to be used as the starting point in searching for the new cursor. The results are (a) a return code indicating whether the function succeeded, and, if not, why not, and (b) next-cursor, which is the requested cursor. The error codes in the return code may be end-of-list, not-a-cursor, etc.

After a session is opened, and an object such as a product use authorization or a LURT has been identified by the cursor, using the functions explained above, the management interface 33 is able to execute certain object management interface functions such as write or copy. By this mechanism, the management interface can modify certain limited attributes. None of these attributes can be modified in such a way that they reduce constraints established by corresponding attributes in the license data objects. The more important attributes which can be modified by the management interface 33 using this mechanism are:

- 55 -

(a) assignment: an assignment of some or all of the units granted on the associated product use authorization;

(b) reservation: a reservation of some or all of the units granted on the associated product use authorization;

5 (c) delegation: a delegation of the right to manage some or all of the units granted on the associated product use authorization, or if the associated license data is not a product use authorization, the delegation is of the right to use that license data;

10 (d) backup delegation: a statement of the right to manage some or all of the units granted on the associated product use authorization; this right is only active at times when the delegating server is not available;

(e) allocation: an allocation of units to a specific context;

15 (f) allocation period: the minimum duration of a single allocation - all allocated units cannot be allocated to a new context until a time period equal to the allocation period has passed since the units were last allocated;

(g) termination date: a date which is to override the value specified as the end date of the product use authorization 40 - this date must be earlier than specified;

20 (h) delegation permitted: an override of the delegation permitted flag of the associated license data;

(i) overdraft: the current overdraft level;

(j) overdraft logging: an override of the overdraft logging attribute of the associated product use authorization;

25 (k) comment: a comment created by the licensee;

(l) extended info: information not defined by the architecture which may be of use in managing the license data.

- 56 -

5 It will be noted that an assignment and a reservation are identical, the only difference being that a reservation is something optional, while an assignment is something that is required. If the duration is Assignment in the policy declaration of Figure 3, the license manager must assign some or all of the units before units can be allocated. Reservations, on the other hand, are made by the license manager using the management interface, regardless of the policy.

10 Thus, there are certain attributes that can be changed by a license administrator using the management interface at the server 10, but none of these can result in obtaining more extensive rights to use than granted by the product use authorization. In each case, the license administrator can limit the rights which will be allocated to users in some way that may be appropriate for the administrator for control purposes.

LICENSE DOCUMENT INTERCHANGE FORMAT

15 The major structural components of an ASN.1 encoded document which conforms to the specifications for the license management system discussed above will be described. The object identifier that is assigned to this data syntax, according to one embodiment, is that specified in ASN.1 as seen in Figure 8. The International Standards Organization or ISO, as it is referred to, defines how bit patterns are chosen to uniquely identify an object type, so the bit pattern set forth
20 in Figure 8 would precede each document used in the license management system so the document could be identified as being a document conforming to the prescribed License Document Interchange Format.

- 57 -

A document encoded according to this format is represented by a value of a complex data type called "license document interchange format document" of LDIFDocument, in this embodiment. A value of this data type represents a single document. This self-describing data structure is of the syntax defined in the international standard ASN.1 referred to above. The X/Open standard referred to above defines the conventions that must be used in employing this syntax, while the syntax itself is described in an OSI (Open Systems Interconnect, a standard administered by ISO) document identified as X.409 (referenced in the X/Open document identified herein).

The LDIFDocument data type consists of an ordered sequence of three elements: the document descriptor, the document header, and the document itself. Each of these elements are in turn composed of other elements. The overall structure of the LDIFDocument data type will be described, and the nature of the document descriptor and document header types. Then, the document content elements will be described in detail, as well as the various component data types used in the definition of the descriptor, the header and the content.

The LDIFDocument represents a single license document, with the syntax being shown in Figure 9 and the high-level structure of an LDIF document in graphical form being seen in Figure 10. The DocumentDescriptor of Figure 9 is a description of the document encoding, the DocumentHeader contains parameters and processing instructions that apply to the document as a whole, and the DocumentContent is the content of the document, all as explained below.

Referring to Figure 9, what this says is that an LDIFDocument is composed of (::= means "is composed of") a number of elements, the first thing in an

- 58 -

LDIFDocument is a bit pattern (tag) according to an international standard, indicating a certain type of document follows, which is indicated here to be "private" or vendor selected, the number 16373 in this case. Following the bit pattern which functions as a "starting delimiter" it is "implicit" that a "sequence" of elements must follow, where a sequence is distinguished from a set. A sequence is one or more of the elements to follow, whereas a set is exactly one of the elements to be listed. Implicit means that any file identified as LDIFDocument must have a sequence data type, rather than some other type. In the case of Figure 9, the sequence is document-descriptor, document header and document content; the document-content is mandatory, whereas the first two are optional. If an element in the sequence begins with a "0" it is a document-descriptor, "1" means a document-header, and "2" means it is a document-content. Again, it is implicit that the data following is of the format DocumentDescriptor, etc., in each case, and these are defined in Figure 11, Figure 13 and Figure 15.

Each file is in the tag-length-value format mentioned above, and also each element of a file containing multiple elements is of the tag-length-value format. The data stream could be examined beginning at any point, and its content determined by first looking for a tag, which will tell what data structure this is, then a length field will say how long it is, then the content will appear. These structures are nested within one another; a document containing several product-use-authorizations would be an LDIFDocument of the format of Figure 9, with a number of DocumentContent elements of Figure 15 following, with the length given for the LDIFDocument spanning the several PUAs, and the length given for each PUA being for the one PUA.

In Figure 11, the elements major-version and minor-version are seen to be "implicit integer". This means that because the element is of the type major-version, etc., it must be an integer. Various other implicit types are given in other syntax diagrams, such as character-string, boolean, etc.

5 In Figure 15, the license body is identified as being of the type "choice" meaning it can be one of PUA, LURT, GroupDefinition, KeyRegistration, etc. Thus, knowing this is a license-body does not mean the data type of the object is known; it is a bit further where the kind of a license-body becomes known. The definition of a license body is not implicit, but instead is a choice type.

10 The contents of the various data elements will now be described in detail with reference to Figures 11-43. Using these detailed descriptions, the exact format of each of the elements used in the LDIF can be interpreted.

15 The license document descriptor or DocumentDescriptor consists of an ordered sequence of four elements which specify the version level of the LDIF encoding and identify the software that encoded the document, with the syntax being shown in Figure 11. An example of the way a product called PAKGEN V1.0 is expressed in the DocumentDescriptor encoding is shown in Figure 12. The fields in the DocumentDescriptor syntax are major-version, minor-version, encoder-identifier and encoder-name. The major-version field is the primary
20 indicator of compatibility between LDIF processors and the encoding of the present document; this major-version field is updated if changes are made to the system encoding that are not backward compatible. The minor-version field is the revision number of the system encoding. The encoder-identifier field is a registered facility mnemonic representing the software that encoded the document;

- 60 -

the encoder-identifier can be an acronym or abbreviation for the encoder name -
this identifier is constant across versions of the encoder. The encoder-identifier
should be used as a prefix to Named Value Tags in Named Value Lists to identify
the encoder of the named value. The encoder-name field is the name of the
product that encoded the document; the encoder-name string must contain the
version number of the product.

The document header or `DocumentHeader` contains data that pertains to
the document as a whole, describing the document to processors that receive it;
the syntax is shown in Figure 13. An example of a document header is shown in
Figure 14, using the hypothetical product `PAKGEN V1.0` of Figure 12. The
private-header-data contains the global information about the document that is not
currently standardized; all interpretations of this information are subject only to
private agreements between parties concerned, so a processor which does not
understand private header data may ignore that data. The Title field is the user-
visible name of the document. The Author field is the name of the person or
persons responsible for the information content of the document. The Version
field is the character string used to distinguish this version of the document from
all other versions. The Date field is the date associated with this document. Note
that the nature and significance of the Title, Author, Version, and Date fields can
vary between processing systems.

The content of an LDIF document is represented by a value of a complex
data type called `DocumentContent`. An element of this type contains one or more
`LicenseData` content element using a syntax as shown in Figure 15. There are no
restrictions on the number, ordering or context of `LicenseData` elements. The
structure of a `LicenseData` element is represented in Figure 16. No restrictions

- 61 -

are made on the number, ordering, or context of LicenseData elements. The license-data-header field of Figure 16 specifies that data, common to all types of license data, which describes the parties to the licensing agreement, the term of the agreement, and any constraints that may have been placed on the management of the license data encoded in the license body. The license-body is an element that contains one content element, including: product use authorizations, license unit requirements tables, group definitions, key registrations, and various forms of delegations. The Management-Info is an element that contains information concerning the current state of the license data; this element is not encoded by Issuers.

The license data header, called LicenseDataHeader, is represented as a syntax diagram in Figure 17. The license-id field provides a potentially unique identification of the encoded license data, so issuers of license data can generate unique license-ids to distinguish each issuance of license data; however, the architecture does not require this to be the case, since the only architectural restriction is that no two objects in any single license management domain may have the same value for license-id. The licensee field identifies the party who has received the rights reflected in the license data; there are at least two parties involved in all transfers of license data, first, the issuer of the license data, and second, the licensee or recipient of that data - it is anticipated that individual licensees will specify to those issuing them licenses what the licensee fields on their license data should contain. the term field identifies the term during which the license data may be used; the validity of license data can be limited by issuers to specific time ranges with given starting and ending dates, which are carried in the term element - attempts to use license data or products described by that data either before the start date or after the end date will result in conforming license

- 62 -

managers denying access to the license. Management-constraints identifies constraints placed on the right to manage the associated license data; these constraints can include (a) limiting the set of contexts permitted to manage the data, (b) limiting the set of platforms which may benefit from that management, and (c) limiting the right to backup and delegate the managed data. The signature provides the digital signature used by the issuer to sign the license data and identifies the algorithm used in encoding the signature. Issuer-comment is a comment provided by the issuer and associated with the license data.

The IssuerComment is of an informational nature and does not impact the process of authorizing product or feature use. This field is not included in the fields used to generate the signature for a license, thus, even if specified by an issuer, the IssuerComment can be omitted from a license without invalidating the license. If specified, the IssuerComment should be stored in the appropriate license data base with the associated license data. The IssuerComment can be retrieved by products which use the system and may be of particular utility to products in the "Software Asset Management" domain which are intended to extend or augment the administrative or accounting facilities or basic system components. Some examples of potential uses for this field are order information, additional terms and conditions, and support information. For order information, some issuers may wish to include with their loadable license data some indication of the purchase order or orders which caused the license data to be issued; licensees may find it useful to include this data in their license databases to assist in the license management process. For additional terms and conditions, the system will never provide automatic means for the management of all possible license terms and conditions, and so some issuers may wish to include summaries of non-system managed terms and conditions in the comment as a reminder. For

support information, the IssuerComment could be used to record the phone numbers or addresses of the responsible individuals within the issuing organization who should be contacted if there are problems with the data as issued.

5 A product use authorization as previously discussed in reference to Figure 2 is used to express the issuance of a right to use some product, product feature, or members of some product group. As such, it records the identity of the product for which use is authorized and specifies the means that will be used by the license manager to ensure that the licensee's actual use conforms to the terms and conditions of the license. Figure 18 illustrates a syntax diagram for a
10 ProductUseAuthorization. Product-id identifies the name of the producer of the product or product feature of which usage rights are being granted as well as the name of that product; in addition, issuers of product use authorizations may specify a range of versions and/or releases whose use is controlled by the specific product use authorization. Units-granted - Contains the number of units of
15 product use which are granted by the license. Management-policy defines the policy which is to be used in managing the granted software usage rights; this definition specifies the Style, Context-Template, Duration, and License Unit Requirements Determination Method which must be used. The calling-authorizations and caller-authorizations are as explained above in reference to
20 calling cards. The execution-constraints field identifies constraints placed on the characteristics of execution contexts which may be authorized to benefit from the units granted by this Product Use Authorization. The product-token field contains product specific data not interpreted in any way by any processors conformant with the architecture; software product producers 28 use this array to augment the
25 capabilities of conformant license managers.

Some anticipated uses of the token field include language support, detailed feature authorizations, and product support number. For language support, a token could be constructed which contains a list of local language interface versions whose use is authorized; thus, if a product were available in English, German, French and Spanish, a token could be constructed listing only English and German as the authorized languages. For detailed feature authorizations, some license issuers will wish to have very fine control over the use of features in a complex product; however, they may not wish to issue a large number of individual Product Use Authorizations to "turn on" each feature, so these vendors could construct tokens which contain lists of the features authorized or whose use is denied. For product support number, some issuers may wish to include on the product use authorization, and thus make available to the running product, some information concerning the support procedures for the product; for example, an issuer might include the telephone number of the support center or a support contract number, and the product could be designed to retrieve this data from the license manager and display it as part of Help dialogues.

The LURT's or license use requirements tables of Figure 4 provide a means by which issuers of licenses, whose LURDM is dependent on the type of platform on which the product is run, can store information describing the relationship between the platform type and unit requirements. A syntax diagram for a LURT is shown in Figure 19. In Figure 20, an example of how the LURT of Figure 4 might be encoded is illustrated. Lurt-name specifies the name by which the LURT is to be known to conforming license managers. The rows field models a list of multicolumn lurt rows. Platform-id identifies the platform for which this LurtRow provides license unit requirements. The lurt-columns field provides a list of one or more lurt column values; the first value provided is

- 65 -

assigned to column-1 of the lurt-row, the second value provided is assigned to column-, etc. A lurt column value of -1 indicates that use of the product or feature is not authorized, while a lurt column value of 0 or greater indicates the number of units that must be allocated in order to authorize product use on the platform described by this lurt-row. All unspecified columns (e.g., columns whose number is greater than the number of column values provided in the lurt columns element) will be considered to contain the value -1.

In reference to Figure 19, to use the row-selector feature mentioned above, the platform-ID element would be replaced with *row-selector* which would be implicit of Context. Also, in Figure 34 described below, in the lurdm-kind element, *row-selector* would be included if the row-select feature is to be used.

As discussed above, Figure 4 provides an example of a hypothetical LURT, illustrating the LURT mechanism, where the issuer of this LURT table has established three unit requirement tiers for use in determining the unit requirements for that issuer's products. Figure 20 provides an example of how the LURT presented in Figure 4 might be encoded.

A group definition is used to define and name a license group. Once so defined, the name of this group can be used on product use authorizations in the same manner as a product name. Since a single product use authorization specifies the management policy for all members of the group, the members of that group must be compatible in their licensing styles, i.e., a personal use type product can not be mixed with a concurrent use product in the same group. Figure 21 shows a group definition syntax diagram. Group-name is the name which must appear on Product Use Authorizations for this group. Group-version

- 66 -

specifies the current version of this group; the requirements for matching between the version information on a product use authorization and that on a specified group definition are the same as those rules which require matching between produce use authorizations and the Release Date data provided by products. Group-members lists those products or features which are components of the named group.

A key registration is used by a producer 28 or issuer 25 who have been registered as authorized license issuers and provided with an appropriate public and private key pair. The key registration identifies the public key which is to be used by conforming license managers 10 in evaluating signatures 53 created by the named issuer 25 or producer 28. A key registration syntax diagram is shown in Figure 22. Key-owner-name provides the name which must be used in either of, or both, of the Producer and Issuer fields of license data generated by the issuer; the key-owner-name must be identical to that specified in the Issuer field of the header record. Key-algorithm identifies the registered algorithm that is to be used when producing digital signatures with this key. Key-value identifies the public key.

An issuer delegation is typically issued by a producer 28 and authorizes the named issuer 25 to issue licenses for products produced by the producer. An issuer delegation syntax diagram is shown in Figure 23. Delegated-issuer-name identifies the name which must appear in the Issuer field of any Product Use Authorization generated using the License Issuer Delegation. Delegated-product-id identifies the products whose licenses the named issuer is authorized to issue. Delegated-units-granted, if specified, indicates that the use of this IssuerDelegation is to be managed in the style of a consumptive license; the value of this attribute

- 67 -

gives the number of units for which license documents may be generated (i.e., if granted 1000 units by a Producer, an Issuer can only issue 1000 units.) Template-authorization provides a "template" Product Use Authorization whose attribute values must be included on any Product Use Authorization generated using this IssuerDelegation; in the case of attributes which have a scalar value (i.e., Version, Release Date, etc.), the Issuer may issue licenses with more restrictive values than those specified on the Template Authorization. Sub-license-permitted indicates whether the Issuer identified on this IssuerDelegation may issue an IssuerDelegation for the delegated-product-id.

A license delegation, as shown in a syntax diagram of Figure 24, is used to delegate the right to manage license data. Such delegations are created by the licensee (by the license manager), if authorized by the issuer 28. A backup delegation, also shown in Figure 24, is used by one license management facility to authorize another to manage the delegated rights in the case that the delegating license manager is not running. The delegated-units field specifies the number of units whose management is being delegated; this may only be specified when a product use authorization is being delegated. Delegation-distribution-control defines the mechanisms by which the distribution and refreshing of the delegation will be accomplished. Delegatee-execution-constraints identifies any constraints which are placed on the execution-context of the Delegatee; these constraints are applied in addition to those which are a part of the delegated License Data. Assignment-list identifies any assignments of the delegated units that must be respected by the delegatee. Delegated-data stores a copy of the LicenseData received from the issuer that is the subject of the delegation; the delegated data is not provided when the LicenseDelegation element is included in a DelegationList.

- 68 -

The management information or ManagementInfo element records information concerning the current state of the LicenseData with which it is associated. A syntax diagram of the ManagementInfo element is shown in Figure 25. The assignments field identifies a list of one or more assignments which may be outstanding for the units on the associated product use authorization. Reservations identifies a list of one or more reservations which may be outstanding for the units on the associated product use authorization. Delegations identifies a list of all outstanding delegations. Backup-delegations identifies all outstanding backup delegations. the allocations field provides detailed information about outstanding allocations which involve units from the associated product use authorization. Registration-date is the date on which the LicenseData was registered in the license database. Registrar is the context which caused the LicenseData to be registered. Local-comment is a comment field. Termination-date means a license defined date after which the license data may not be used; this date must be earlier than the end-date specified in the license data's term record. The extended-info field allows additional information concerning the state of the LicenseData and its handling by the license manager that is not standardized.

The defined types of elements will now be described. These defined type are:

Allocation	ManagementPolicy
Assignment	Member
Context	NamedValue
DistributionControl	NamedValueList
ExecutionConstraints	ProductID
IntervalTime	Signature

- 69 -

LicenseID	Term
LUDRM	Version
ManagementConstraints	

5 The allocation element records the information concerning a single unit allocation, and is shown in a syntax diagram in Figure 26. Allocation-context specifies the context to which the allocation was made. The allocation-lur field specifies the license unit requirement which applies to the allocation-context; this license unit requirement is calculated without consideration of any allocation sharing which may be possible. The allocation-group-id field identifies the
10 "allocation-group" for the current allocation, in which an unshared allocation will always have an allocation group id of 0; allocations which utilize shared units will have an allocation group id which is shared by all other allocations sharing the same units.

15 The assignment element is shown in syntax diagram in Figure 27. Assigned-units identifies the number of units which are assigned. Assignment-term identifies the start and end of the assignment period. Assignee identifies the context to which the assignment is made.

20 The context element is shown in syntax diagram in Figure 28. The SubContext-type field identifies the type of subcontext, and this type can be either standard or private; if standard, the type value will be taken from the standard-subcontext-type enumeration: (a) network-subcontext means the subcontext value identifies a network; (b) execution-domain-subcontext means the subcontext value is the name of the management domain within which the caller is executing; (d) login-domain-subcontext means the subcontext value is the name of the

management domain within which the user of the caller was originally authenticated or "logged in"; (d) node-subcontext means the subcontext value is the name of a node; (e) process-family-subcontext means the subcontext value is an implementation specific identifier for a group of related processes; (f) process-ID-subcontext means the subcontext value is an implementation specific process identifier; (g) user-name-subcontext means the subcontext value is a user name; (h) product-name-subcontext means the subcontext value is the same as the product name found on the Product Use Authorization; (i) operating-system-subcontext means the subcontext value is a character string representation of the name of the operating system; (j) platform-ID-subcontext means the subcontext value is an identifier that describes the hardware platform supporting the context. The subcontext-value field is the value of the subcontext.

As discussed above, license data is always used or allocated within, or for the benefit of, some named licensing context. This context name is constructed by concatenating the values of all subcontexts into a single context name. A Context Template specifies those components of the context name which should be used in calculating license unit requirements. The management system determines the need to perform a unit allocation each time license units are requested. The full context on whose behalf the allocation should be made is obtained for each requested authorization. The system will mask the full context to exclude all sub-contexts not specified in the context template and then determine if the resulting context already has units allocated to it. If not, units will be allocated according to the specification of the LURDM, otherwise, the units previously allocated will be shared by the new context. Thus, if a given product authorization contains a context specification of NODE + USER_NAME, each context which requests license unit allocations and which has a unique pair

of NODE + USER_NAME subcontext values will require an explicit grant of license units to be made. On the other hand, any contexts which share the same pair of NODE and USER_NAME subcontext values will be able to "share" a single allocation of license units. The requirement for specific allocations of units and the ability to share units is exhibited in Figure 29 which attempts to provide a "snapshot" of the units allocated for the product FOOBAR V4.1 at a particular instance. It is seen from the figure that although presented with five unique full contexts, only four of them are unique when looking only at those portions of each context which are described by the Context Template (ie: NODE + USER_NAME). A unit allocation must be made for each of the four instances of unique contexts, when masked by the Context Template. The fifth context can share allocated units with another context. Thus, the total requirement to support product use as described in this example would be 40-units (ie: four allocations of ten units each). Significant changes in the unit requirements can be achieved by making small modifications to the Context Template. Figure 30 shows the same contexts as in Figure 29 but a Context_Template of NODE. The total unit requirement for this example would be three units (three allocations of ten units each) rather than the forty units required in the previous example.

The distribution control element defines the mechanism that will be used for distributing the subject delegation and records some status information concerning the distribution of that delegation. A syntax diagram of the distribution control element is shown in Figure 31. Distribution-method identifies the means by which the delegation will be distributed, and the alternatives are refresh-distribution, initial=distribution-only, and manual-distribution. Refresh-distribution means the license manager shall be responsible for the initial distribution of the delegation and for ensuring that refresh delegations are

-72 -

properly distributed. Initial-distribution-only means the license manager shall be responsible for the initial distribution of the delegation, however, distribution of refresh delegations will be made by some other means. Manual-distribution means the distribution of the delegation will be under the control of some other mechanism (perhaps a license asset manager). Current-start-date is the time that the last successful initial or refresh delegation distribution was performed. Current-end-date identifies the last date on which the most recent delegation distribution was performed. Refresh-interval identifies the period of time between attempts to refresh the delegation; the refresh-interval may not be longer than the maximum-delegation-period and should normally be less than that in order to ensure that refresh delegations are distributed prior to the expiration of the previous delegations that they are replacing. Retry-interval identifies the amount of time to wait for an unsuccessful distribution attempt to try again. Maximum-retry-count identifies the maximum number of times that an unsuccessful distribution attempt may be retried. Retries-attempted records the number of unsuccessful retry attempts which have been made since the last successful initial or refresh delegation distribution was performed.

The execution constraints elements place limits on the environments and contexts which may receive allocations. A syntax diagram of the execution constraints element is shown in Figure 32. Operating-system contains a list of zero or more operating systems on which the use of the subject license is authorized; if no operating systems are specified, it is assumed that license use is authorized on all operating systems. Execution-context specifies a list of zero or more full or partial context names which identify the contexts within which products described by the license data may be executed; if no context names are specified, the licensed products may be executed in any context controlled by the licensee.

- 73 -

Environment-list identifies those environments within which the licensed product may be used.

The interval time element is defined by the syntax `IntervalTime ::= UTCTime`.

5 The license ID element uniquely identifies the license data it is associated with, and is described by the syntax diagram of Figure 33. Here issuer uniquely identifies the issuer of the license data as well as the name space within which the LicenseID Number is maintained. While the issuer name will typically be the same as the name of the issuer's company or personal name, this is not a
10 requirement. For instance: The issuer name for Digital Equipment Corporation is "DEC," an abbreviation of the corporate name. Valid contents of the Issuer field are maintained in the an Issuer Registry. The serial-number provides a unique identification or serial number for the license data. The amendment field is an integer which is incremented each time license data is amended by its issuer,
15 with the first version of any license data carries the amendment number 0; an amendment can only be applied to license data if that license data has identical Issuer and Number values and an amendment number less than the number of the amendment to be applied.

20 The license units requirements determination method or LURDM element is shown in syntax diagram in Figure 34. The combination-permitted field indicates whether conforming license managers are permitted to combine together into a common pool the units from different product use authorizations if those produce use authorizations have the same product record value; for example, if combination is permitted and a single license manager discovers in its database

- 74 -

two 500-unit authorizations for the use of DEC Cobol, the license manager would be permitted to combine these two authorizations into a logical grant of 1000 units. The overdraft-limit modifies the behavior of a conforming license management facility in those cases where it is found that there are zero or fewer
5 license units available for use at the time of a request for the allocation or consumption of additional license units. Operation of overdraft is different depending upon whether allocative, or consumptive style is being used. In using with allocative style, an allocation is granted even though the remaining units are zero or less, up to the overdraft-limit. In using with consumptive style, the license
10 is authorized to accumulate a negative balance of license units, up to the overdraft-limit. Overdraft-logging-required indicates whether all license grants which are the result of overdraft use must cause a log record to be generated. When the allocation-size field is non-zero, then all unit allocations and delegations must be made in sizes which are whole number multiples of the allocation-size
15 value. Lurdm-kind identifies the method by which license unit requirements will be calculated once the requirement for an allocation has been discovered, the permitted alternatives being (a) LURT which specifies that license unit requirements are to be determined by lookup in the LURT which is associated with the current license, (b) Constant which specifies that license unit
20 requirements are constant for all platforms on which the licensed product or product feature may run, and (c) Private-LURDM which specifies that license unit requirements are to be determined by the licensed product, not by the license management facility. The named-lurt-id specifies the name of the LURT table to be used in determining license unit requirements if the LURDM-kind is specified as LURT; if the LURDM-kind is specified as LURT and no table is explicitly
25 named, the name of the table to be used is constructed from the issuer name on the product use authorization. Lurdm-value specifies the LURT column to be

-75-

used when LURDM-kind = LURT; however, when LURDM-kind = Constant, the Lurdm-value field contains the precise number of units to be allocated or consumed. Default-unit-requirement specifies the unit requirement value to be used when the appropriate LURT does not have a row corresponding to the appropriate platform ID; when specified on a product use authorization with Style = Allocative, the context template will change to Process + Product_Specific and the Duration will change to Transaction in cases of unrecognized Platform ID's.

The management constraints element is shown in a syntax diagram in Figure 35. The management-context field specifies a list of zero or more partial context names which identify the specific contexts within which the license data may be managed. If no management contexts are specified, the license data may be managed within any context controlled by the licensee. The contexts used in specifying Management Context Constraints may only contain the Network, Domain, and Node subcontexts. Specifying a list of management contexts does not effect whether or not the license data can be used within other contexts. For example, unless otherwise restricted, license data with a specified management context can be remotely accessed from or delegated to other nodes in a network. The management-scope field defines the maximum permitted size of the license management domain within which the license data may be managed or distributed, these being single-platform, management-domain, or entire-network. Single-platform constrains the license management domain for the subject license data to be no larger than a single platform. Management-domain constrains the license management domain for the subject license data to be no larger than a single management domain. Entire-network constrains the license management domain for the subject license data to be no larger than a single wide area network; that

-76-

network which contains the platform on which the license units were initially loaded. Although technology may not exist to detect the interorganizational boundaries of a wide area network (i.e., what is on the Internet as opposed to being on a company's own network), the assumption is that interorganization and internetwork sharing of licenses will normally be considered a violation of license terms and conditions. The backup-permitted field indicates if the Issuer has authorized the use of backup delegations for this data. Delegation-permitted indicates if the Issuer has authorized the licensee to delegate this data. Maximum-delegation-period identifies the longest interval during which a delegation may be valid; by default, delegations have a life of 72-hours.

The major elements of the management policy specification are shown in Figure 3, as previously discussed. A syntax diagram for the management policy element is shown in Figure 36. For the Style field, three fundamental styles of license management policy are supported, allocative, consumptive, and private-style, as explained above. Only one of these styles may be assigned to any single product use authorization. The Context-template specifies those components (sub-contexts) of the execution-context name which should be used in determining if unit allocations are required. The Duration defines the duration of an allocation of license units to a specific context or the duration of the period which defines a valid consumptive use. For durations of type "Assignment," the specification of a Reassignment Constraint is also provided for. Three types of Duration_Kind are supported, these being Transaction, Assignment and Immediate, as explained above. The dur-determination-method stores information used in calculating the number of units that should be allocated or consumed in response to a license request. The allocation-sharing-limit identifies the largest number of execution contexts that may share an allocation made under this management policy; an

- 77 -

allocation-sharing-limit of 0 indicates that the number of execution contexts that may share an allocation is unlimited. The reassignment-constraint specifies a minimum duration of assignment; although there is normally no constraint placed on how frequently granted units may be reassigned, an issuer may constrain
5 reassignment by specifying this minimum duration of an assignment, in which case reassignment of assigned units will not be supported until the amount of time specified in the Reassignment Constraint has passed. If an assignment of some particular set of units has been delegated and the delegation period for that delegation has not terminated, cancellation of the delegation must be performed
10 prior to reassignment.

The member element identifies a specific licensed product which may be part of a calling authorization or group definition, and is shown in syntax diagram in Figure 37. Member-product identifies the product which is a member. Member-signature is constructed from the product and token fields of the called
15 member structure as well as the product and issuer fields of the calling product. Member-token provides the data which should be used as the product token for this member.

Named values are data elements with a character string tag that identifies the data element, and have a syntax as shown in Figure 38, which also shows the
20 syntax for ValueData and named value list. A named value list models a list of named values, with an example being shown in Figure 39. In Figure 38, Value-Name uniquely identifies the value; no standard value names are defined, and the period character can be used as a part of the value name to form a hierarchical tag registry at the discretion of the issuer. Value-data is the data that has been
25 named; data types are selected from the possible Value Data types, seen in the

Figure. Value-boolean means the named data is a boolean value. Value-integer means the named data is an integer value. Value-text means the named data is a StringList value. Value-general means the named data is a stream of bytes in any format. Value-list means the named data is a list of named data values.

5 The product ID explicitly identifies the product which is the subject of the license data with which it is associated, with the syntax for ProductID being shown in Figure 40. The version and release date fields provide a mechanism for defining which specific instances of the licensed product are described in the associated license data. The Producer field is a registered name which identifies
10 the producer of the licensed feature; in the case of Group Names, the Producer is always also the Issuer of the group. The Product-name identifies a licensed software feature. The First-version identifies the earliest version of the product whose use is authorized. The Last-version identifies the latest version of the product whose use is authorized. The First-release-date identifies the earliest
15 release of the product whose use is authorized. The Last-release-date identifies the latest release of the product whose use is authorized. Conforming license managers are required to interpret the contents of these fields in the most restrictive way possible. Thus, if a license is issued with Last-version = 3.0 and a Last-release-Date of 1-Jan-1991, then the use of version 2.0 of the licensed
20 product would be unauthorized if it had a release date of 2-Jan-1991. If either a First-version or First-release-date is specified without a matching Last-version or Last-release-date, use of the produce is authorized for all versions or release dates following that specified. Similarly, if either a last-version or Last-release-date is specified without a matching First-version or First-release-date, use of the produce
25 is assumed to be authorized for all versions or release dates prior to that specified. Issuers should typically only specify one of either First-version or First-release-

- 79 -

date. This is the case since it is anticipated that these fields will typically refer to events which occurred prior to the moment of license data issuance. Thus, it should normally be possible for the issuer to state unambiguously with only one of these two fields which is the oldest implementation of the product that is to be authorized. The architecture does permit, however, both fields to be used in a single product authorization.

The signature element is used to establish the integrity and authorship of the license data with which it is associated. A syntax diagram for the signature element is shown in Figure 41. The Signature-algorithm field identifies the registered algorithm that was used to produce the digital signature. Signature-parameters are the values of the algorithm's parameters that are to be used; the need for and syntax of parameters is determined by each individual algorithm. Signature-value is an enciphered summary of the information to which the signature is appended; the summary is produced by means of a one-way hash function, while the enciphering is carried out using the secret key of the signer (Issuer).

The term element defines an interval during which the license data is valid, and is shown in syntax diagram form in Figure 42. The fields are start-date and end-date. Start-date identifies the first date of the term; if not specified, the license data is considered valid on any date prior to the end-date. End-date identifies the last date of the term; if not specified, the license data is considered valid on any date after the Start-date. While the Start-date is always either omitted or specified as an absolute date, the End-date can be either absolute or relative. If the End-date is specified as a relative or "interval" date and the Start-date has been omitted, the date of license registration will be used as the effective

- 80 -

5 start date in computing the valid term of the license data. It should be noted that the system does not specify the mechanism by which system dates are maintained by platforms supporting system components. Instead, the system always accepts that system time returned to it as correct. Thus, the reliability of the management of license data which specifies terms is dependent on the time management function of the underlying platform.

10 The version element identifies a four-part version of the licensed software product or feature. A syntax diagram of the version element is shown in Figure 43. The schematics of each of the four parts is not detailed, but it is required that producers who wish to permit version ranges to be specified on product use authorizations ensure that the collating significance of the four parts is maintained. When comparing versions, Part-1 is considered first, then Part-2, then Part-3, and finally, Part-4. Part-1 identifies a major modification to the versioned object. Part-2 identifies a modification to the versioned object which is less significant
15 than a modification which would cause a change in the Part-1 value. Part-3 identifies a modification to the versioned object which is less significant than a modification which would cause a change in the Part-2 value. Part-4 identifies a modification to the versioned object which is less significant than a modification which would cause a change in the Part-3 value.

20 FILTERS

An important feature is the use of filters in the license management program 11, including the client interface 31 and the management interface 33. A filter is used to select items in the license database 23, for example. Various

- 81 -

selection mechanisms are used in picking out or doing lookups in database technology; filters are one of them. The filter engine used in the license management system 11 of Figure 1 is generally of a known construction, with the exception of the select filter item type as will be described, which allows a
5 complex rather than a flat data format to be selected from. The feature that is of importance to this embodiment is the way of specifying items as an input to the filter function , rather than the filter function itself. Thus, there is described below a template for specifying input to the filter engine. This is as if a form were used as the input, with blanks on the form; by filling in certain blanks these
10 would be the items selected on, the blanks not filled in would be "don't care".

An instance of the class *filter* is a basis for selecting or rejecting an object on the basis of information in that object. At any point in time, a filter has a value relative to every object - this value is false, true or undefined. The object is selected if and only if the filter's value is true. This concrete class has the
15 attributes of its superclass - *Object* - and the specific attributes listed in the table of Figure 44.

A filter is a collection of simpler filters and elementary filter-items together with a Boolean operation. The filter value is undefined if and only if all the component filters and filter-items are undefined. Otherwise, the filter has a
20 Boolean value with respect to any object, which can be determined by evaluating each of the nested components and combining their values using Boolean operation (components whose value is undefined or ignored). The attributes specific to *filter* as shown in Figure 44 are (a) *filter items* which are a collection of assertions, each relating to just one attribute of an object, (b) *filters* which are a

- 82 -

collection of simple filters, and (c) *filter type* which is the filter's type, of one of the following values: And, Or, Not.

An instance of the class *filter item* is a component of a *filter*. It is an assertion about the existence or values of a single attribute of a license data object or one or its subobjects. This concrete class has the attributes of its superclass - *object* - and the specific attributes listed in the table of Figure 45.

The value of a filter item is undefined if: (a) the Attribute Types are unknown, or (b) the syntax of the Match Value does not conform to the attribute syntax defined for the attribute type, or (c) a required Attribute is not provided. The attributes specific to *filter item* as shown in Figure 45 are (a) *filter item type* which identifies the type of filter item and thereby the nature of the filter, and its value must be one of

equality	less
inequality	present
greater or equal	select
less or equal	request candidates
greater	simulate request

(b) *attribute type* which identifies the type of that attribute whose value or presence is to be tested; the value of All Attributes may be specified, (c) *match value* which is the value which is to be matched against the value of the attribute, (d) *filter* which identifies the filter to be used in evaluating a selected subobject of the current object; the filter is ignored if the *filter item type* is not *select* or if the specified attribute type is not present in the object, and upon evaluation of the *filter* the value of *filter item* will be set to that of the *filter*, (e) *initial substring*, if present, this is the substring to compare against the initial portion of the value of

- 23 -

the specified attribute type, (f) *substring*, if present, this is the substring(s) to compare against all substrings of the value of the specified attribute type, (g) *final substring*, if present, this is the substring to compare against the final portion of the value of the specified attribute type, and (h) *license request*, if present, this is license request against which the appropriate license data objects should be evaluated; this attribute may only be specified if the value of the filter item type is either Request Candidates or Simulate Request.

An instance of enumeration syntax *Filter Type* identifies the type of a filter. Its value is chosen from one of the following: (a) *And* means the filter is the logical conjunction of its components; the filter is true unless any of the nested filters or filter items is false, or if there are no nested components, the filter is true; (b) *Or* means the filter is the logical disjunction of its components; the filter is false unless any of the nested filters or filter items is true, or, if there are no nested components, the filter is false; (c) *Not* means the result of the filter is reversed; there must be exactly one nested filter or filter item, and the filter is true if the enclosed filter or filter item is false, and is false if the enclosed filter or filter item is true.

An instance of enumeration syntax *Filter Item Type* identifies the type of a filter item. Its value is chosen from one of the following: (a) *Equality* which means the filter item is true if the object contains at least one attribute of the specified type whose value is equal to that specified by Match Value (according to the equality matching rule in force), and false otherwise; (b) *Inequality* which means the filter item is true if the object contains at least one attribute of the specified type whose value is not equal to that specified by Match Value (according to the equality matching rule in force), and false otherwise; (c) *Greater*

- 84 -

or *Equal* which means the filter item is true if the object contains at least one attribute of the specified type whose value is equal to or greater than the value specified by Match Value (according to the matching rule in force), and false otherwise; (d) *Less or Equal* which means the filter item is true if the object contains at least one attribute of the specified type whose value is equal or less than the value specified by Match Value (according to the matching rule in force), and false otherwise; (e) *Greater* which means the filter item is true if the object contains at least one attribute of the specified type whose value is greater than the value specified by Match Value (according to the matching rule in force), and false otherwise; (f) *Less* which means the filter is true if the object contains at least one attribute of the specified type, whose value is less than the value specified by Match Value (according to the matching rule in force), and false otherwise; (g) *Present* which means the filter item is true if the object contains at least one attribute of the specified type, and false otherwise; (h) *Select* which means the filter item is true if the object contains at least one attribute of the specified type which has an object syntax and when the Filter is evaluated against the attributes of that object the Filter is true, and false otherwise; (i) *Request Candidates* which means the filter item is true if the object against which it is evaluated is one which could be used to provide some or all of the units requested by the specified License Request; the evaluation is made independently of any outstanding allocations or preallocations; and (j) *Simulate Request* which means the filter item is true if the object against which it is evaluated is one which would be used to provide some or all of the units requested by the specified License Request.

The Request Candidates and Simulate Request filter item types are of special use in testing and prototyping of systems by a license manager at a

- 85 -

licensee's site. For example, the license manager can simulate the effect of potential assignments, the effect of a population of certain types on a network, etc.

As an example, Figure 46 shows how a filter may be constructed to identify "All Product Use Authorizations issued by Digital for the Product 'Amazing Graphics System' which contains a calling authorization for Digital's 'Amazing Database' Product". This example is in the international standard format referred to as X.409 as mentioned above.

Filters can also used in a request allocation, being specified in a request extension as explained above. That is, a filter is one of the optional items in a request extension. For example, if a user wanted to use a version of WordPerfect with French language extension, and there were version with and without on the network, his request allocation would have a request extension that specified a filter for "French" in the token field. In this manner, a product can describe itself more richly. The filter in the request extension can be a Required filter or a Preferred filter, meaning the feature such as "French" is either absolutely necessary, or merely the preferred.

While this invention has been described with reference to specific embodiments, this description is not meant to be construed in a limiting sense. Various modifications of the disclosed embodiments, as well as other embodiments of the invention, will be apparent to persons skilled in the art upon reference to this description. It is therefore contemplated that the appended claims will cover any such modifications or embodiments as fall within the true scope of the invention.

-86-

WHAT IS CLAIMED IS:

1 1. A method of managing use of licensed software items, said
2 software items separately executable on a computer system or
3 accessible by said computer system, the computer system including
4 a processor and one or more nodes, comprising the steps of:

5 maintaining by said processor a store of license
6 authorizations for said software items; each license authorization
7 including an indication of license management policy for a software
8 item, said indication having a plurality of sets of policy
9 components, said sets of policy components granting alternatives of
10 specified restrictive rights to selectively access and execute said
11 software items in said system; said indication of license
12 management policy being in the format of an encoded document of a
13 data type consisting of an ordered sequence of elements;

14 accessing said store by said processor to modify in said store
15 one or more of said specified restrictive rights of said policy
16 components of an identified license authorization;

17 accessing said store by said processor using a filter to
18 obtain information from said license authorization for a selected
19 software item, in response to a request from a node, and

20 comparing an identification of said node and said software
21 item with said information, to produce and send to said node a
22 grant or refusal of said request.

1 2. A method according to claim 1 including the step of
2 receiving said license authorizations , for storing in said store,

1 from a license grantor external to said processor, and wherein said
2 step of accessing said store to modify in said store one or more of
3 said specified restrictive rights employs management functions
4 executable on said processor but not on said nodes or said license
5 grantor to identify a license authorization in said store.

1 3. A method according to claim 1 wherein said indication is
2 in the format of an encoded document of a data type consisting of
3 an ordered sequence of three elements, the three elements including
4 a document descriptor, a document header and the document content.

1 4. A method according to claim 1 wherein said filter
2 specifies one or more of said attributes and a Boolean operator for
3
4 each selected attribute.

1 5. A method according to claim 2 wherein said step of
2 accessing said store to modify one or more of said policy
3 components is to allow grant of rights to use which are more
4 restrictive than said specified restrictive rights.

1 6. A method according to claim 2 including the steps of:

2

3 sending a request by a user of one of said software items to
4 obtain permission to use said software item; said request
5 identifying the user and said software item;

1 accessing said store to obtain information from said license
2 authorization for said software item, in response to said request,
3 and comparing said identification of said user and said software
4 item with said information, to produce a grant or refusal of said
5 request for sending to said user.

1 7. A method according to claim 6 wherein said store is
2 maintained by a license server, and said request is sent to said
3 server and wherein said request is in the form of a remote
4 procedure call, and said grant or refusal sent to said user is a
5 return of said procedure call.

6

1 8. A method according to claim 7 wherein said license
2 authorization is a data arrangement specified as a product use
3 authorization, and said product use authorization is received by
4 said server from an issuer, and wherein said server and said users
5 are nodes on a computer network.

1 9. A method according to claim 2 wherein said policy
2 components include a termination date, and said management
3 functions can modify said termination date to an earlier
4 termination date and wherein said policy components include a right
5 of delegation of a right to grant said requests to another server,
6 and said management functions can modify said right of delegation
7 to remove said right of delegation.

1 10. A method according to claim 2 including storing in
2 association with said license authorization a number of management
3 attributes, and said management functions being able to modify said
4 management attributes.

1 11. A method according to claim 10 wherein said management
2 attributes include a reservation of units of license use granted by
3 said license authorization so that said units will not be granted
4 to a user in response to said request, and wherein said management
5 attributes include an allocation of units of license use to a
6 specific context.

1 12. A method according to claim 10 wherein said management
2 attributes include an allocation period which is the minimum
3 duration of an allocation of units, and wherein said management
4 attributes include permission to enable a backup delegation of the
5 right to grant said requests.

1 13. A system for managing use of licensed software products,
2 comprising: means for maintaining a store of license documents, one
3 for each said product; each license document including an
4 indication of license policy having plurality of sets of policy
5 components granting specified restrictive rights to use said
6 software products, said policy components in each set providing
7 alternatives;

8 a management interface for accessing said store to modify

1 selected ones of said components of an identified license
2 authorization.

1 14. A system according to claim 13 including:

2 means for sending a request from a user of one of said
3 products to obtain permission to use said product; said request
4 identifying the user and said product;

5 means for accessing said store to obtain information from said
6 license document for said product, in response to said request, and
7 for comparing said identification of said user and said product
8 with said information, and with constraints imposed by said policy
9 components, to produce a grant or refusal of said request and send
10 said grant or refusal to said user.

1 15. A system according to claim 13 wherein said management
2 interface can modify said selected ones of said components to allow
3 grant of rights to use which are more restrictive than said
4 specified restrictive rights and wherein said means for
5 maintaining, and said means for accessing and sending to said user
6 are all located at a server on a distributed network, and said
7 means for sending a request is located at a user node on said
8 network.

1 16. A system according to claim 14 wherein said request is in
2 the form of a remote procedure call, and said grant or refusal sent
3 to said user is a return of said procedure call, and wherein said

1 license document is a data arrangement specified as a product use
2 authorization, and said product use authorization is received by
3 said server from a license issuer.

1 17. A system according to claim 13 wherein said policy
2 components include a termination date, and said management
3 functions can modify said termination date to an earlier
4 termination date, and wherein said policy components include a
5 right of delegation of a right to grant said requests to another
6 server, and said management functions can modify said right of
7 delegation to remove said right of delegation.

1 18. A system according to claim 15 including means for storing
2 in association with said license authorization a number of
3 management attributes, wherein said management functions are able
4 to modify said management attributes and wherein said management
5 attributes include a reservation of units of license use granted by
6 said license authorization so that said units will not be granted
7 to a user in response to said request.

1 19. A system according to claim 18 wherein said management
2 attributes include an allocation of units of license use to a
3 specific context.

1 20. A system according to claim 18 wherein said management
2 attributes include an allocation period which is the minimum

1 duration of an allocation of units, and include permission to
2 enable a backup delegation of the right to grant said requests.

1 21. A method according to claim 3 wherein said document
2 descriptor includes an encoding method version number, and encoder-
3 identifier and an encoder-name, and wherein said document-header
4 includes a title, an author, a version and a date for the software
5 item.

1 22. A method according to claim 3 wherein said document
2 content includes at least one of the following:

3 a product-use-authorization;
4 a license-use-requirements-table;
5 a group-definition;
6 a key-registration;
7 a delegation.

1 23. A method according to claim 3 wherein said document-
2 content includes a license-data-header, and said license-data-
3 header describes the parties to the license document, the term of
4 the agreement and constraints that may have been placed on
5 management of the license data.

1 24. A method according to claim 3 wherein said document-
2 content includes management-info, where the management-info may
3 include at least one of the following:

1 an assignment;
2 a reservation;
3 a delegation;
4 a backup delegation;
5 an allocation;
6 a registration date;
7 a registrar;
8 a comment;
9 a termination-date.

1 25. A method according to claim 3 wherein:
2 said document descriptor includes an encoding method
3 version and a date for the software item;
4 said document content may include at least one of the
5 following: a product-use-authorization, a license-use-requirements-
6 table, a group-defination, a key-registration, and a delegation;
7 said document-content selectively includes a license-
8 data-header, and said license-data-header describes the parties to
9 the license document, the term of the agreement and constraints
10 that may have been placed on management of the license data;
11 said document-content may have been placed on management
12 of the license data;
13 said document-content selectively includes management-
14 info, where the management-info may include at least one of the
15 following: an assignment, a reservation, a delegation, a backup
16 delegation, an allocation, a registration date, a registrar, and a

1 comment.

1 26. A method according to claim 3 wherein said store is
2 maintained by a license server, and said request is sent to said
3 server, and wherein said server and said users are nodes on a
4 computer network.

1 27. A method according to claim 3 wherein said request is in
2 the form of a remote procedure call, and said grant or refusal sent
3 to said user is a return of said procedure call, and wherein said
4 license authorization is received by said server from an issuer.

1 28. A method according to claim 3 including the steps of:
2 sending a request by a user of one of said software items to obtain
3 permission to use said software item; said request identifying the
4 user and said software item;
5 sending said grant or refusal to said user.

1 29. Apparatus for managing use of licensed software items,
2 comprising:
3 means for maintaining a store of license authorizations
4 for said software items; each license authorization including an
5 indication of license management policy for a software item, said
6 indication being in the format of an encoded document of a data
7 type consisting of an ordered sequence of three elements, the three
8 elements including a document descriptor, a document header and the

1 document content;

2 means for sending a request by a user of one of said
3 software items to obtain permission to use said software item; said
4 request identifying the user and said software item;

5 means for accessing said store to obtain information from
6 said license authorization for said software item, in response to
7 said request, and comparing said identification of said user and
8 said software item with said information, to produce a grant or
9 refusal of said request;

10 means for sending said grant or refusal to said user.

1 30. Apparatus according to claim 29 wherein said document
2 descriptor includes an encoding method version number, and an
3 encoder-identifier and an encoder-name, and wherein said document-
4 header includes a title, an author, a version and a date for the
5 software item.

1 31. Apparatus according to claim 29 wherein said document
2 content includes at least one of the following:

3
4 a product-use-authorization;
5 a license-use-requirements-table;
6 a group-definition;
7 a key-registration;
8 a delegation.

9

1 32. Apparatus according to claim 29 wherein said document-
2 content includes a license-data-header, and said license-data-
3 header describes the parties to the license document, the term of
4 the agreement and constraints that may have been placed on
5 management of the license data.

1 33. Apparatus according to claim 29 wherein said document-
2 content includes management-info, where the management-info may
3 include at least one of the following:
4 an assignment;
5 a reservation;
6 a delegation;
7 a backup delegation;
8 an allocation;
9 a registration date;
10 a registrar;
11 a comment;
12 a termination-date.

1 34. Apparatus according to claim 29 wherein:
2 said document descriptor includes an encoding method
3 version number, and encoder-identifier and an encoder-name;
4 said document-header includes a title, an author, a
5 version and a date for the software item;
 said document content may include at least one of the
following: a product-use-authorization, a license-use-requirements-

table, a group-definition, a key-registration, and a delegation;

said document-content may include a license-data-header, and said license-data-header describes the parties to the license document, the term of the agreement and constraints that may have been placed on management of the license data;

said document-content may include management-info, where the management-info may include at least one of the following: an assignment, a reservation, a delegation, a backup delegation, an allocation, a registration date, a registrar, and a comment.

1

2

3

4

5

6

35. Apparatus according to claim 29 wherein said store is maintained by a license server, and said request is sent to said server, and wherein said request is in the form of a remote procedure call, and said grant or refusal sent to said user is a return of said procedure call.

1

2

3

36. Apparatus according to claim 29 wherein said license authorization is received by said server from an issuer, and wherein said server and said users are nodes on a computer network.

1

2

3

4

5

6

37. A method of storing license documents by a server for a license management system, comprising the steps of:

maintaining a store of license documents for software items; each license document including an indication of license management policy for a software item, said indication being in the format of an encoded document of a data type consisting of an ordered

1 sequence of three elements, the three elements including a document
2 descriptor, a document header and the document content;

3 accessing said store to obtain information from a selected one
4 of said license documents for a software item, in response to a
5 request, and referencing said indication of license management
6 policy, to produce a grant or refusal of said request.

1 38. A method according to claim 37 wherein said document
2 descriptor includes an encoding method version number, an encoder-
3 identifier and an encoder-name, and wherein said document-header
4 includes a title, an author, a version and a date for the software
5 item.

1 39. A method according to claim 37 wherein said document
2 content includes at least one of the following:

3 a product-use-authorization;
4 a license-use-requirements-table;
5 a group-definition;
6 a key-registration;
7 a delegation.

1 40. A method according to claim 4 wherein said step of
2 selecting by a filter may select on one or more of the attributes:
3 issuer, producer, product name, product use authorization, calling
4 authorization, and wherein said store is maintained by a license
5 server, and said request is sent to said server.

1 41. A method according to claim 4 wherein said request is in
2 the form of a remote procedure call, and said grant or refusal sent

1 to said user is a return of said procedure call.

1 42. A method according to claim 40 wherein said license
2 authorization is a data arrangement specified as a product use
3 authorization, and said product use authorization is received by
4 said server from an issuer, and wherein said server and said users
5 are nodes on a computer network.

1 43. Apparatus for managing use of licensed software items,
2 comprising:

3 means for maintaining a store of license authorizations for
4 said software items; each license authorization including an
5 indication of license management policy for a software item, said
6 indication being an encoded document containing a number of
7 attributes defining said license policy;

8 filter means for selecting from said store, said filter means
9 specifying one or more of said attributes and a Boolean operator
10 for each selected attribute;

11 means for sending a request by a user of one of said software
12 items to obtain permission to use said software item; said request
13 identifying the user and said software item;

14 means for accessing said store to obtain information from said
15 license authorization for said software item, in response to said
16 request, and comparing said identification of said user and said
17 software item with said information, to produce a grant or refusal
18 of said request; and

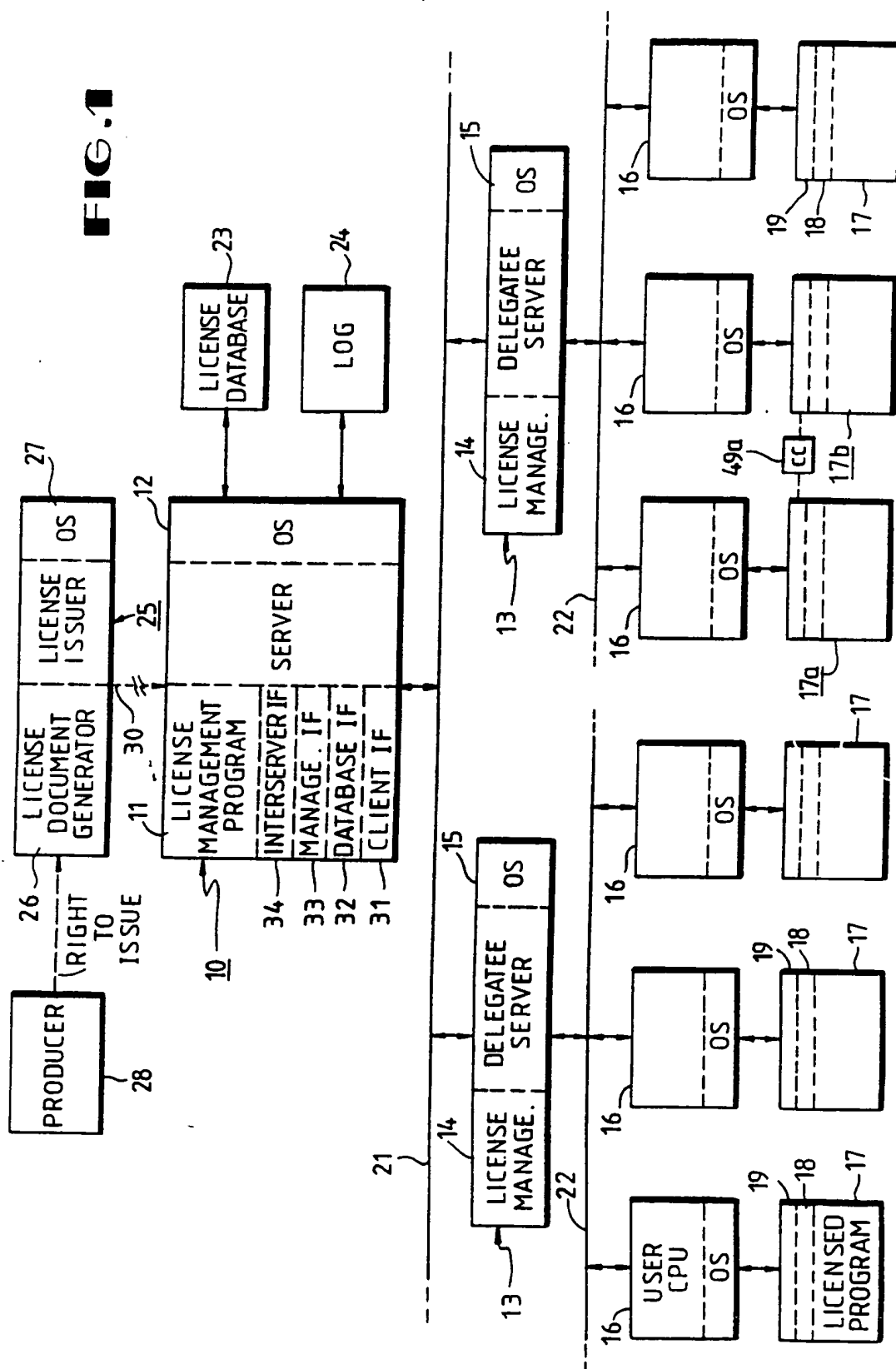
1 means for sending said grant or refusal to said user.

1 44. Apparatus according to claim 43 wherein said filter means
2 may select on one or more of the attributes: issuer, producer,
3 product name, product use authorization, calling authorization, and
4 wherein said store is maintained by a license server, and said
5 request is sent to said server, and wherein said request is in the
6 form of a remote procedure call, and said grant or refusal sent to
7 said user is a return of said procedure call.

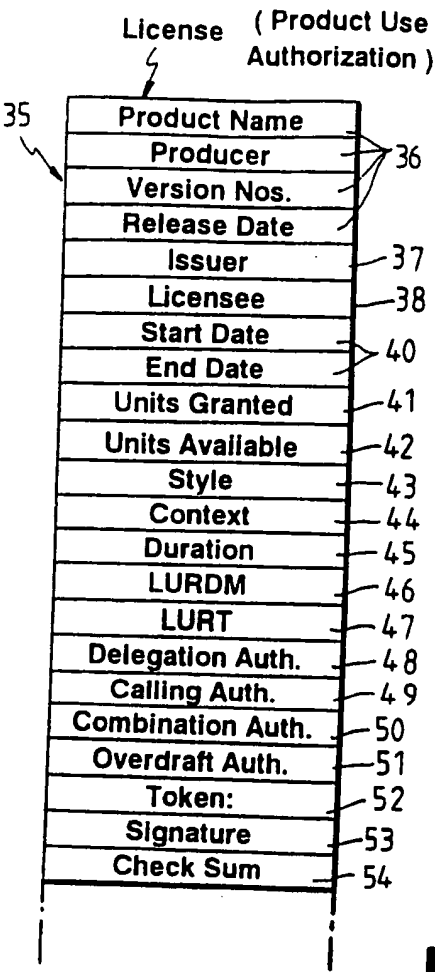
1 45. Apparatus according to claim 43 wherein said license
2 authorization is a data arrangement specified as a product use
3 authorization, and said product use authorization is received by
4 said server from an issuer, wherein said server and said users are
5 nodes on a computer network.

1/32

THE



SUBSTITUTE SHEET



License Unit Requirements Table			
Row Selector	Columns		
Platform ID	A	B	C
PC-0	10	230	-1
PC-1	12	230	-1
VAX 6210	158	300	150

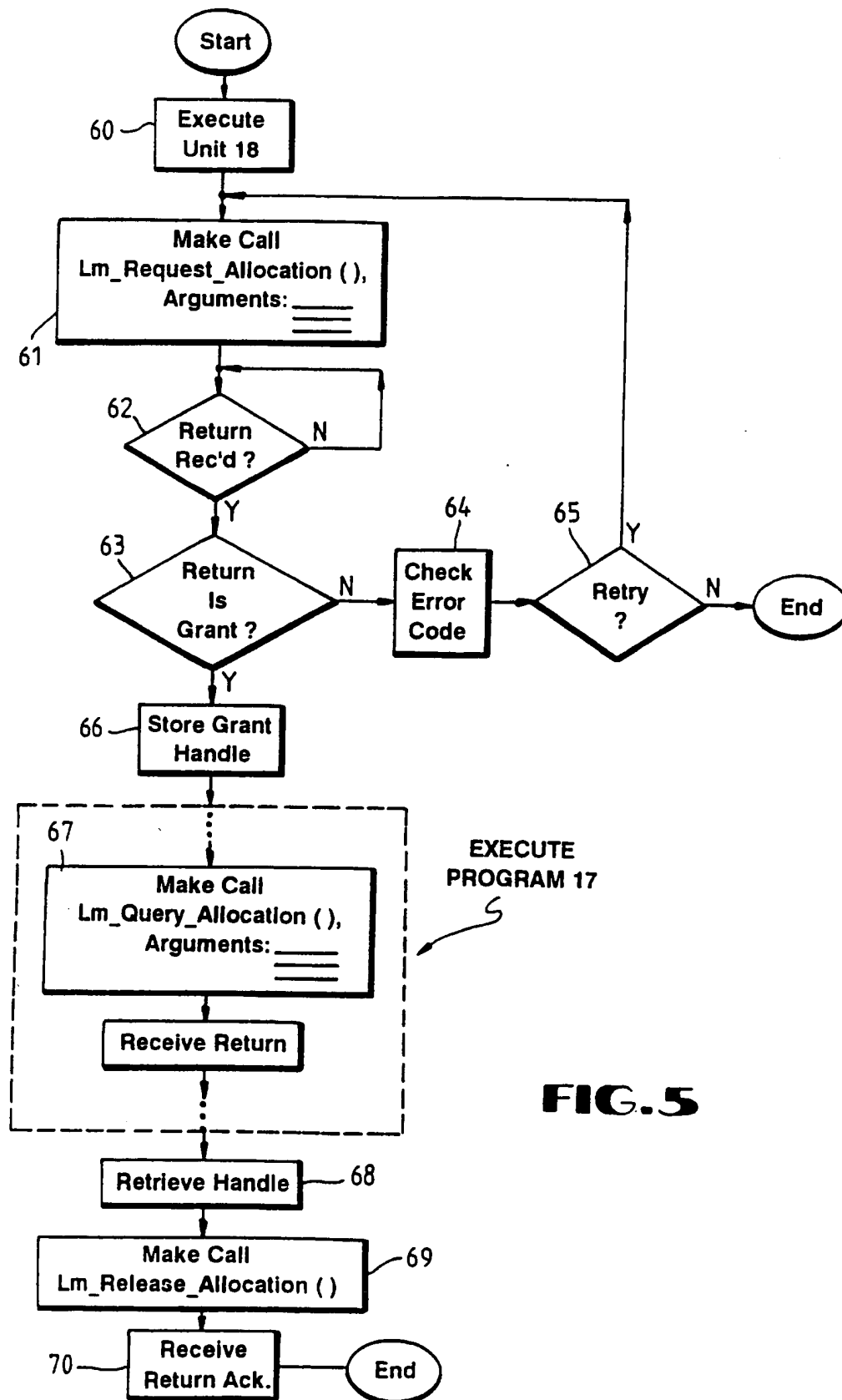
FIG.4

FIG.2

43	44	45	46
Style	Context	Duration	LURDM
Allocative	Network	Transaction	Constant
Consumptive	Execution_Domain	Assignment	Table Lookup
Private	Login_Domain	Immediate	Private
	Node_ID		
	Process_Family		
	Process		
	User_Name		
	Product_Name		
	Operating_System		
	Platform_ID		
	Private		

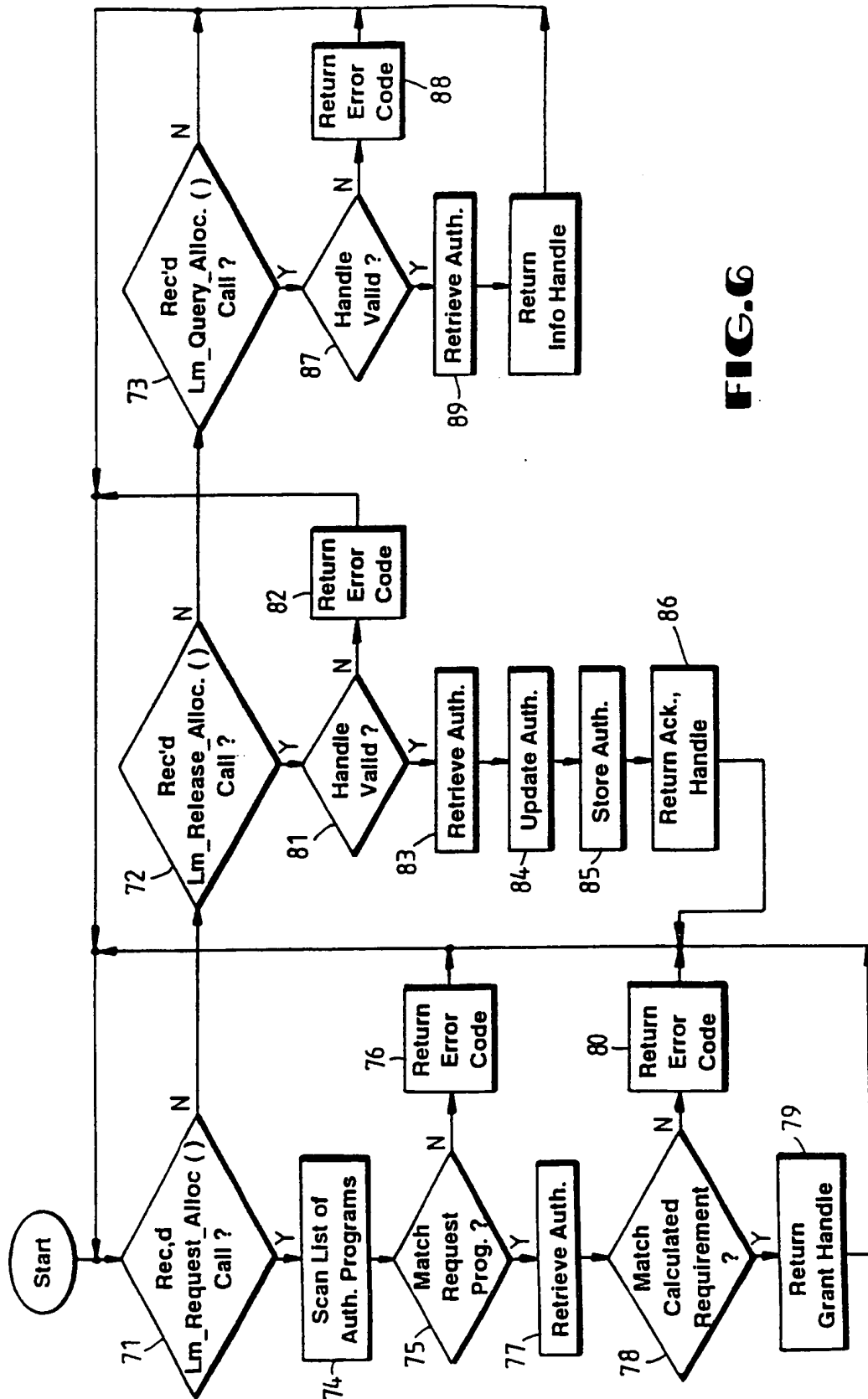
FIG.3

3/32

**FIG. 5**

SUBSTITUTE SHEET

4/32

**FIG. 6**

SUBSTITUTE SHEET

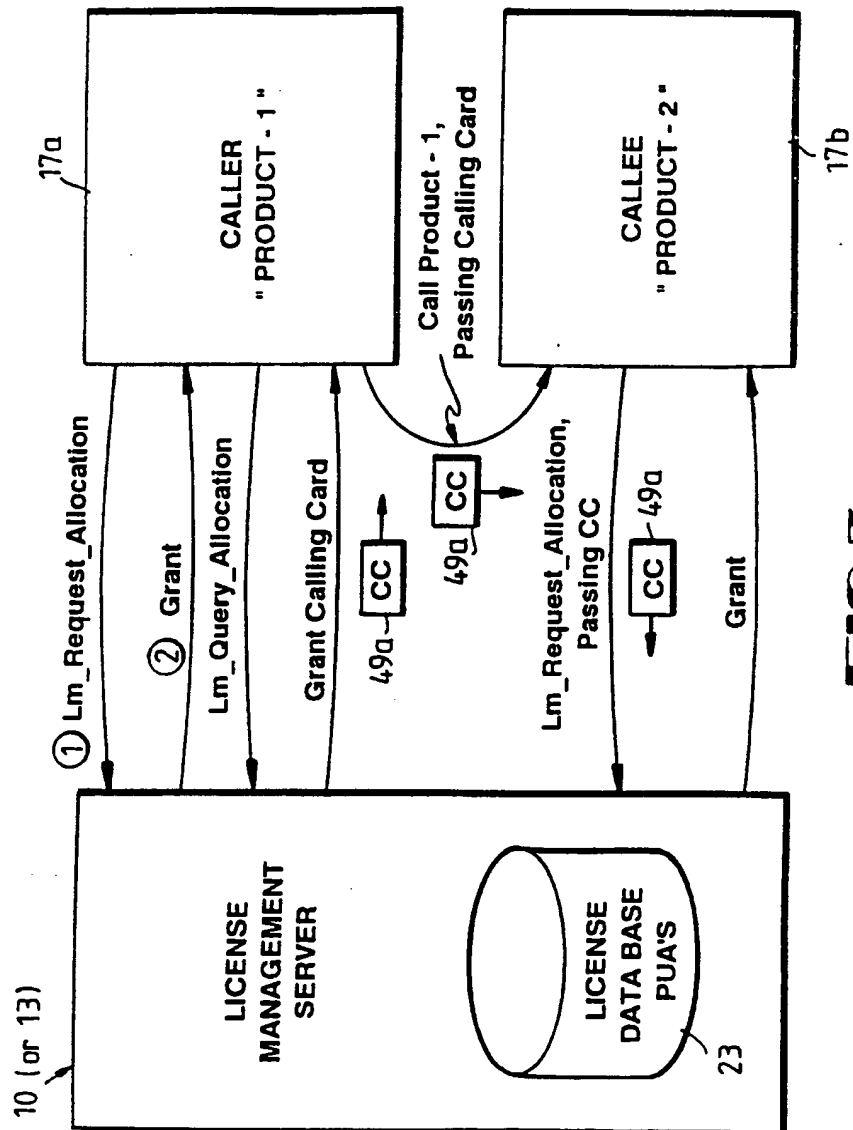


FIG.7

SUBSTITUTE SHEET

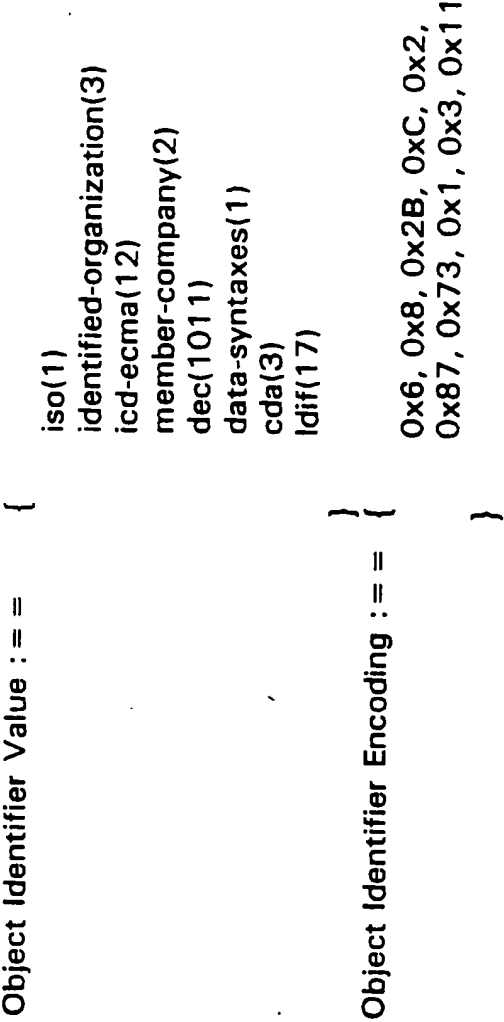


FIG. 8 LDIF Object Identifier

```
LDIFDocument ::= [PRIVATE 16373] IMPLICIT SEQUENCE {  
    document-descriptor [0] IMPLICIT DocumentDescriptor OPTIONAL,  
    document-header [1] IMPLICIT DocumentHeader OPTIONAL,  
    document-content [2] IMPLICIT DocumentContent  
}
```

FIG. 9 LDIF Document Syntax Diagram

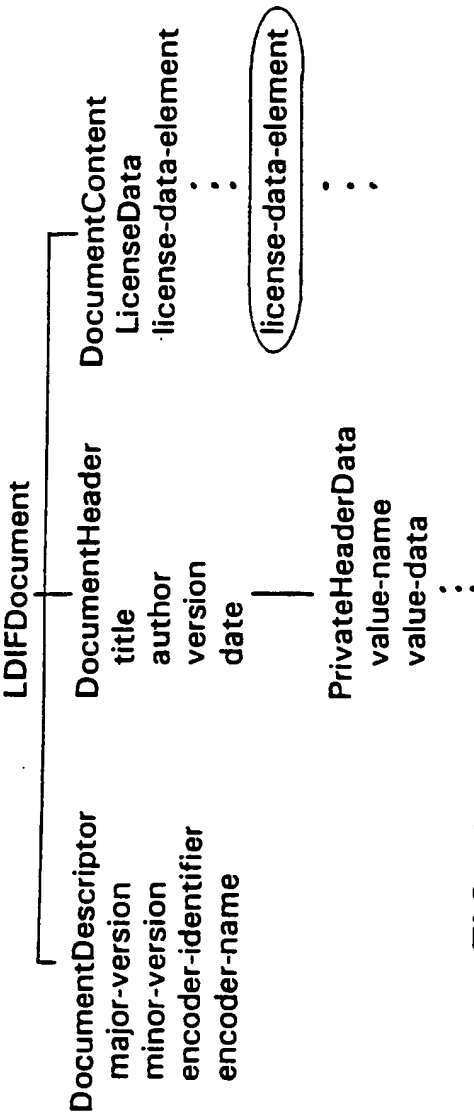


FIG. 10 LDIF Document Structure

8/32

```

DocumentDescriptor ::=
    major-version
    minor-version
    encoder-identifier
    encoder-name
    SEQUENCE {
        [0] IMPLICIT INTEGER OPTIONAL,
        [1] IMPLICIT INTEGER OPTIONAL,
        [2] IMPLICIT Character-String OPTIONAL,
        [3] IMPLICIT Character-String OPTIONAL
    }

```

FIG. 11 Document Descriptor Syntax Diagram

```

Pakgen DocumentDescriptor ::= {
    major-version 1,
    minor-version 0,
    encoder-identifier "PAKGEN",
    encoder-name {Character-String "PAK Generator V1.0"}
}

```

FIG. 12 Document Descriptor Example

SUBSTITUTE SHEET

9/32

```

DocumentHeader      ::=
private-header-data
title
author
version
date
SEQUENCE {
    [0] IMPLICIT NamedValueList OPTIONAL,
    [1] IMPLICIT Character-String OPTIONAL,
    [2] IMPLICIT Character-String OPTIONAL,
    [3] IMPLICIT Character-String OPTIONAL,
    [4] IMPLICIT UTCTime OPTIONAL
}

```

FIG. 13 Document Header Syntax Diagram

```

example-header document-header ::= {
    title {Character-String "PAKGEN Licenses with Associated LURT data"}
    author {Character-String "Tom Jones, Foobar, Inc. License Department"}
    version {Character-String "VO.1"}
    date "198801021100-0500"
}

```

FIG. 14 Document Header Example

SUBSTITUTE SHEET

10/32

```

Document Content      ::= SEQUENCE OF LicenseData

LicenseData
  license-data-header
  license-body
    product-use-authorization
    license-units-requirements-table
    group-definition
    key-registration
    issuer-delegation
    license-delegation
    backup-delegation
  management-info
}
  ::= SEQUENCE {
    [0] IMPLICIT LicenseDataHeader,
    [1] CHOICE {
      [0] IMPLICIT ProductUseAuthorization,
      [1] IMPLICIT LURT,
      [2] IMPLICIT GroupDefinition,
      [3] IMPLICIT KeyRegistration,
      [4] IMPLICIT IssuerDelegation,
      [5] IMPLICIT LicenseDelegation,
      [6] IMPLICIT BackupDelegation
    },
    [2] IMPLICIT ManagementInfo OPTIONAL
  }

```

FIG. 15 Document Content Syntax Diagram

SUBSTITUTE SHEET

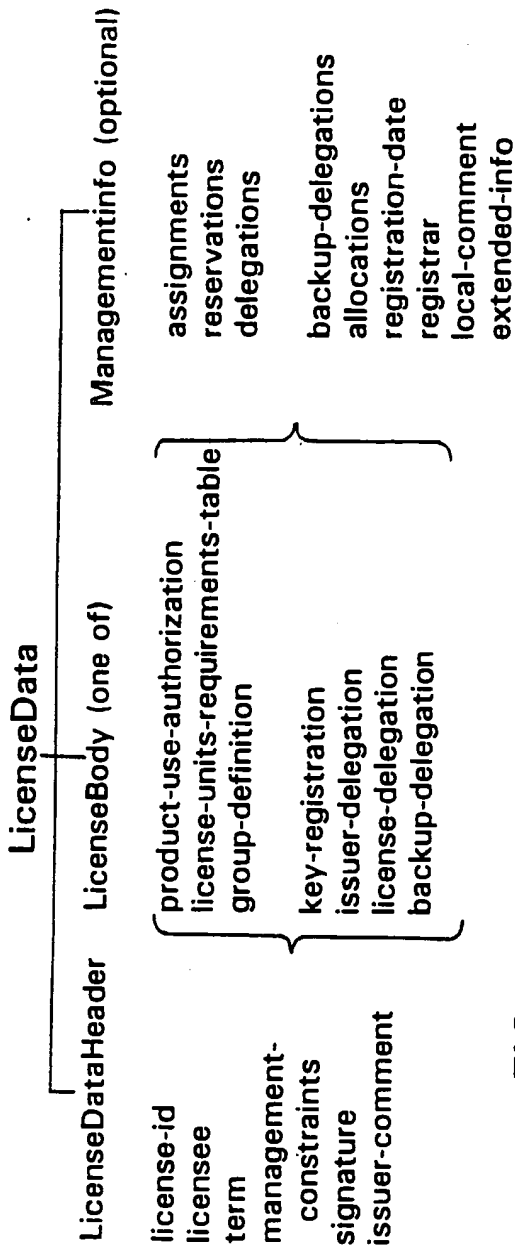


FIG. 16 License Data Structure

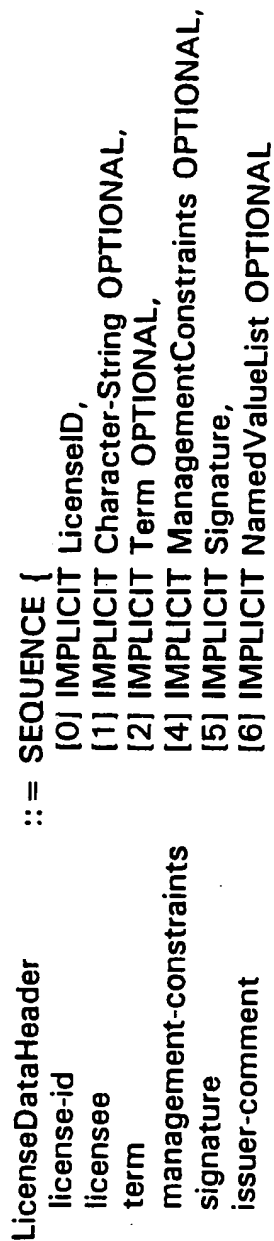


FIG. 17 License Data Header Syntax Diagram

```
ProductUseAuthorization ::= SEQUENCE {
    product-id [0] IMPLICIT ProductID,
    units-granted [1] IMPLICIT INTEGER,
    management-policy [2] IMPLICIT ManagementPolicy,
    calling-authorizations [3] IMPLICIT SEQUENCE OF Member OPTIONAL,
    caller-authorizations [4] IMPLICIT SEQUENCE OF Member OPTIONAL,
    execution-constraints [5] IMPLICIT ExecutionConstraints OPTIONAL,
    product-token [6] IMPLICIT NamedValueList OPTIONAL
}
```

FIG. 18 Product Use Authorization Syntax Diagram

```
LURT ::= SEQUENCE {
    lurt-name [0] IMPLICIT Character-String,
    rows [1] IMPLICIT RowList
}
RowList ::= SEQUENCE OF LurtRow
LurtRow ::= SEQUENCE {
    platform-id [0] IMPLICIT Character-String,
    lurt-columns [1] IMPLICIT SEQUENCE OF INTEGER
}
```

FIG. 19 License Unit Requirement Table Syntax Diagram

13/32

```

Example LURT ::= {
  lurt-name { Character-String "Example LURT" }
  rows {
    LurtRow {
      {Character-String "PC-0"}
      {{10} {230} {-1}}
    }
    LurtRow {
      {Character-String "PC-1"}
      {{12} {230} {-1}}
    }
    LurtRow {
      {Character-String "VAX 6210"}
      {{158} {300} {150}}
    }
  }
}

```

FIG. 20 Example Encoding of LURT

SUBSTITUTE SHEET

```
Group Definition
group-name
group-version
group-release-date
group-members
 ::= SEQUENCE {
    [0] IMPLICIT Character-String,
    [1] IMPLICIT Version,
    [2] IMPLICIT UTCTime,
    [3] IMPLICIT SEQUENCE OF Member
 }
```

FIG. 21 Group Definition Syntax Diagram

```
KeyRegistration
key-owner-name
key-algorithm
key-value
 ::= SEQUENCE {
    [0] IMPLICIT Character-String,
    [1] IMPLICIT Character-String,
    [2] IMPLICIT OCTET STRING
 }
```

FIG. 22 Key Registration Syntax Diagram

15/32

```

IssuerDelegation
  delegated-issuer-name
  delegated-product-id
  delegated-units-granted
  template-authorization
  sub-license-permitted
::= SEQUENCE {
  [0] IMPLICIT Character-String,
  [1] IMPLICIT SEQUENCE OF Member,
  [2] IMPLICIT INTEGER OPTIONAL,
  [3] IMPLICIT ProductUseAuthorization OPTIONAL,
  [4] IMPLICIT BOOLEAN DEFAULT FALSE
}

```

FIG. 23 Issuer Delegation Syntax Diagram

```

LicenseDelegation
  delegated-units
  delegated-distribution-control
  delegatee-execution-constraints
  assignment-list
  delegated-data
::= SEQUENCE {
  [0] IMPLICIT INTEGER OPTIONAL
  [1] IMPLICIT DistributionControl,
  [2] IMPLICIT ExecutionConstraints OPTIONAL,
  [3] IMPLICIT AssignmentList OPTIONAL,
  [4] IMPLICIT LicenseData OPTIONAL
}

```

FIG. 24 License Delegation & Backup Delegation Syntax Diagrams

SUBSTITUTE SHEET

16/32

ManagementInfo	::= SEQUENCE {
assignments	[0] IMPLICIT AssignmentList OPTIONAL,
reservations	[1] IMPLICIT AssignmentList OPTIONAL,
delegations	[2] IMPLICIT DelegationList OPTIONAL,
backup-delegations	[3] IMPLICIT DelegationList OPTIONAL,
allocations	[4] IMPLICIT AllocationList OPTIONAL,
registration-date	[5] IMPLICIT UTCTime,
registrar	[6] IMPLICIT Context,
local-comment	[7] IMPLICIT NamedValueList OPTIONAL,
termination-date	[8] IMPLICIT UTCTime OPTIONAL,
extended-info	[9] IMPLICIT NamedValueList OPTIONAL
	}

FIG. 25 ManagementInfo Syntax Diagram

SUBSTITUTE SHEET

AllocationList	::= SEQUENCE OF Allocation
Allocation	::= SEQUENCE {
allocation-context	[0] IMPLICIT Context,
allocation-lur	[1] IMPLICIT INTEGER,
allocation-group-id	[2] IMPLICIT INTEGER OPTIONAL
	}

FIG. 26 Allocation Syntax Diagram

AssignmentList	::= SEQUENCE OF Assignment
Assignment	::= SEQUENCE {
assigned-units	[0] IMPLICIT INTEGER,
assignment-term	[1] IMPLICIT Term,
assignee	[2] IMPLICIT Context
	}

FIG. 27 Assignment Syntax Diagram

```

ContextList      ::= SEQUENCE OF Context
Context          ::= SEQUENCE OF Subcontext
SubContext
  sub-context-type
  subcontext-value
    ::= SEQUENCE {
      [0] SubContextType,
      [1] ValueData
    }
SubContextType
  standard-subcontext-type [0] IMPLICIT INTEGER {
    network-subcontext(1),
    execution-domain-subcontext(2),
    login-domain-subcontext(3),
    node-subcontext(4),
    process-family-subcontext(5),
    process-id-subcontext(6),
    user-name-subcontext(7),
    product-name-subcontext(8),
    operating-system-subcontext(9),
    platform-id-subcontext(10)
  }
  private-subcontext      [1] IMPLICIT INTEGER {first(0),last(255)}
    }

```

FIG. 28 Context Syntax Diagram

SUBSTITUTE SHEET

FOOBAR V4.1 Allocated Units			
Units	Context Template		Full Context Specifications
	Node	User_Name	
10	BLUE	WYMAN	ENET, AA_Cluster, BLUE, PID-1..., WYMAN
10	RED	OLSEN	ENET, BB_Cluster, RED, PID-1..., OLSEN
10	RED	WYMAN	ENET, BB_Cluster, RED, PID-2..., WYMAN
10	GREEN	WYMAN	ENET, AA_Cluster, GREEN, PID-1..., WYMAN
	GREEN	WYMAN	ENET, AA_Cluster, GREEN, PID-2..., WYMAN

FIG. 29 Only unique contexts require explicit unit allocations.

FOOBAR V4.1 Allocated Units		
Units	Context Template	Full Context Specifications
	Node	
10	BLUE	ENET, AA_Cluster, BLUE, PID-1..., WYMAN
10	RED	ENET, BB_Cluster, RED, PID-1..., OLSEN
	RED	ENET, BB_Cluster, RED, PID-2..., WYMAN
10	GREEN	ENET, AA_Cluster, GREEN, PID-1..., WYMAN
	GREEN	ENET, AA_Cluster, GREEN, PID-2..., WYMAN

FIG. 30 Modification of Context_Template impacts units requirements.

21/32

```

DistributionControl ::= SEQUENCE {
    distribution-method [0] IMPLICIT INTEGER {
        refresh-distribution(1),
        initial-distribution-only(2),
        manual-distribution(3)
    },
    current-start-date [1] IMPLICIT UTCTime OPTIONAL,
    current-end-date [2] IMPLICIT UTCTime OPTIONAL,
    refresh-interval [3] IMPLICIT IntervalTime OPTIONAL,
    retry-interval [4] IMPLICIT IntervalTime OPTIONAL,
    maximum-retry-count [5] IMPLICIT INTEGER OPTIONAL,
    retries-attempted [6] IMPLICIT INTEGER OPTIONAL
}

```

FIG. 31 Distribution Control Syntax Diagram

SUBSTITUTE SHEET

22/32

```

ExecutionConstraints ::= SEQUENCE {
    operating-system      [0] IMPLICIT SEQUENCE OF Character-String OPTIONAL,
    execution-context     [1] IMPLICIT ContextList OPTIONAL,
    environment-list      [2] IMPLICIT SEQUENCE OF EnvironmentKind OPTIONAL
}
EnvironmentKind ::= INTEGER {
    batch(1),
    interactive(2),
    local(3),
    network(4),
    remote(5)
}

```

FIG. 32 Execution Constraints Syntax Diagram

SUBSTITUTE SHEET

23/32

```
LicenseID      ::= SEQUENCE {  
    issuer      [0] IMPLICIT Character-String,  
    serial-number [1] IMPLICIT Character-String,  
    amendment   [2] IMPLICIT INTEGER DEFAULT 0  
}
```

FIG. 33 License ID Syntax Diagram

SUBSTITUTE SHEET

```
LURDM ::= SEQUENCE {
    combination-permitted
    overdraft-limit
    overdraft-logging-required
    allocation-size
    lurdm-kind
        lurdt(1),
        constant(2),
        private-lurdm(3)
    named-lurdt-id
    lurdm-value
    default-unit-requirement
}
```

[0] IMPLICIT BOOLEAN DEFAULT TRUE,
[1] IMPLICIT INTEGER DEFAULT 0,
[2] IMPLICIT BOOLEAN DEFAULT FALSE,
[3] IMPLICIT INTEGER OPTIONAL,
[4] IMPLICIT INTEGER {
},
[5] IMPLICIT Character-String OPTIONAL,
[6] IMPLICIT INTEGER OPTIONAL,
[7] IMPLICIT INTEGER OPTIONAL

FIG. 34 License Unit Requirements Determination Method Syntax Diagram

25/32

```

ManagementConstraints ::= SEQUENCE {
    management-context      [0] IMPLICIT ContextList OPTIONAL,
    management-scope        [1] IMPLICIT INTEGER {
        single-platform(1),
        management-domain(2),
        entire-network(3)
    } OPTIONAL,
    backup-permitted         [2] IMPLICIT BOOLEAN DEFAULT TRUE,
    delegation-permitted     [3] IMPLICIT BOOLEAN DEFAULT TRUE,
    maximum-delegation-period [4] IMPLICIT IntervalTime OPTIONAL
}

```

FIG. 35 Management Constraints Syntax Diagram

SUBSTITUTE SHEET

```

ManagementPolicy ::= SEQUENCE {
    style
        allocative(1),
        consumptive(2),
        private-style(3)
    context-template
    duration
        transaction(1),
        assignment(2),
        immediate(3)
    lur-determination-method
    allocation-sharing-limit
    reassignment-constraint
}
    },
    [1] IMPLICIT SEQUENCE OF SubcontextType
    OPTIONAL,
    [2] IMPLICIT INTEGER {
        } OPTIONAL,
    [3] IMPLICIT LURDM OPTIONAL,
    [4] IMPLICIT INTEGER OPTIONAL,
    [5] IMPLICIT IntervalTime OPTIONAL
}

```

FIG. 36 Management Policy Syntax Diagram

SUBSTITUTE SHEET


```

Member
  member-product
  member-signature
  member-token
  ::= SEQUENCE {
    [0] IMPLICIT ProductID,
    [1] IMPLICIT Signature,
    [2] IMPLICIT NamedValueList OPTIONAL
  }

```

FIG. 37 Member Syntax Diagram

```

NamedValue
  value-name
  value-data
  ::= SEQUENCE {
    Character-String,
    ValueData
  }

ValueData
  value-boolean
  value-integer
  value-text
  value-general
  value-list
  ::= CHOICE {
    [0] IMPLICIT BOOLEAN,
    [1] IMPLICIT INTEGER,
    [2] IMPLICIT SEQUENCE OF Character-String
    [3] IMPLICIT OCTET STRING,
    [4] IMPLICIT SEQUENCE OF ValueData
  }

NamedValueList
  ::= SEQUENCE OF NamedValue

```

FIG. 38 Named Value, Value Data & Named Value List Syntax Diagrams

```
ExampleList NamedValueList ::= {  
  NamedValue {  
    value-name {Character-String "Purchase Order"}  
    value-data {INTEGER 154493}  
  }  
  NamedValue {  
    value-name {Character-String "Telephone Support #"}  
    value-data {Character-String { + 1 (999) 555-1234}  
  }  
}
```

FIG. 39 Named Value List Example

```
ProductID  
  producer  
  product-name  
  first-version  
  last-version  
  first-release-date  
  last-release-date  
  ::= SEQUENCE {  
    [0] IMPLICIT Character-String,  
    [1] IMPLICIT Character-String,  
    [2] IMPLICIT Version OPTIONAL,  
    [3] IMPLICIT Version OPTIONAL,  
    [4] IMPLICIT UTCTime OPTIONAL,  
    [5] IMPLICIT UTCTime OPTIONAL  
  }
```

FIG. 40 Product ID Syntax Diagram

Signature ::= SEQUENCE {
signature-algorithm [0] IMPLICIT Character-String,
signature-parameters [1] IMPLICIT NamedValueList OPTIONAL,
signature-value [2] IMPLICIT OCTET STRING
}

FIG. 41 Signature Syntax Diagram

Term ::= SEQUENCE {
start-date [0] IMPLICIT UTCTime OPTIONAL,
end-date [1] IMPLICIT UTCTime OPTIONAL,
}

FIG. 42 Term Syntax Diagram

```
Version
part-1
part-2
part-3
part-4

:: = SEQUENCE {
    [0] IMPLICIT INTEGER,
    [1] IMPLICIT INTEGER DEFAULT 0,
    [2] IMPLICIT INTEGER DEFAULT 0,
    [3] IMPLICIT INTEGER DEFAULT 0
}
```

FIG. 43

Attributes Specific to Filter				
Attribute	Value Syntax	Value Length	Value Number	Value Initially
Filter Items	Object(Filter Item)	-	0 or more	-
Filters	Object(Filter)	-	0 or more	-
Filter Type	Enum(Filter Type)	-	1	-

FIG. 44

Attributes Specific to Filter					
Attribute	Value Syntax	Value Length	Value Number	Value Initially	
Filter Item Type	Enum(Filter Item Type)	-	1	-	
Attribute Type	Type	-	1	-	
Match Value	any	-	0-1	-	
Filters	Object(Filter)	-	0-1	-	
Initial Substring	String(*)	1 or more	0-1	-	
Substring	String(*)	1 or more	0 or more	-	
Final Substring	String(*)	1 or more	0-1 or more	-	
License Request	Object(License Request)	-	0-1	-	

FIG. 45

SUBSTITUTE SHEET

32/32

```

Filter {
  Filter-Type AND
  Filter-Item {
    Filter-Item-Type SELECT
    Attribute-Type Product-Use-Authorization
    Filter {
      Filter-Type AND
      Filter-Item{
        Filter-Item-Type SELECT
        Attribute-Type Calling-Authorization
        Filter{
          Filter-Type AND
          Filter-Item {
            Filter-Item-Type EQUALITY
            Attribute-Type Producer
            Match-Value "Digital"
          }
          Filter-Item {
            Filter-Item-Type EQUALITY
            Attribute-Type Producer
            Match-Value "Amazing Database"
          }
        }
      }
    }
  }
  Filter-Item {
    Filter-Item-Type EQUALITY
    Attribute-Type Producer
    Match-Value "Digital"
  }
  Filter-Item{
    Filter-Item-Type EQUALITY
    Attribute-Type Issuer
    Match-Value "Digital"
  }
  Filter-Item {
    Filter-Item-Type EQUALITY
    Attribute-Type Product-Name
    Match-Value "Amazing Graphics System"
  }
}

```

FIG. 46 Example Filter Value Notation

FIRST ITF SHEET

INTERNATIONAL SEARCH REPORT

PCT/ISA 92/03812

International Application No.

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all) ⁶		
According to International Patent Classification (IPC) or to both National Classification and IPC		
Int.Cl. 5 G06F1/00		
II. FIELDS SEARCHED		
Minimum Documentation Searched ⁷		
Classification System	Classification Symbols	
Int.Cl. 5	G06F	
Documentation Searched other than Minimum Documentation to the extent that such Documents are included in the Fields Searched ⁸		
III. DOCUMENTS CONSIDERED TO BE RELEVANT⁹		
Category ¹⁰	Citation of Document, ¹¹ with indication, where appropriate, of the relevant passages ¹²	Relevant to Claim No. ¹³
Y	EP,A,0 332 304 (DIGITAL EQUIPMENT CORPORATION) 13 September 1989 cited in the application	1-3, 6-19,22, 24, 26-29, 31-33, 35-37,39 43-45
Y	see figure 1 cited in the application	5, 15, 21, 25, 30
A	see column 3, line 31 - column 7, line 55 --- -/-	
⁹ Special categories of cited documents : ¹⁰ "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art. "&" document member of the same patent family		
IV. CERTIFICATION		
Date of the Actual Completion of the International Search	Date of Mailing of this International Search Report	
09 SEPTEMBER 1992	17. 09. 92	
International Searching Authority	Signature of Authorized Officer	
EUROPEAN PATENT OFFICE	WEISS P.	

Form PCT/ISA/210 (second sheet) (January 1985)

III. DOCUMENTS CONSIDERED TO BE RELEVANT (CONTINUED FROM THE SECOND SHEET)		Relevant to Claims No.
Category *	Citation of Document, with indication, where appropriate, of the relevant passages	
Y	IBM TECHNICAL DISCLOSURE BULLETIN. vol. 31, no. 8, 1 January 1989, NEW YORK US pages 195 - 198; 'METHOD FOR MANAGING CLIENT/SERVER RELATIONSHIP IN THE AIX OPERATING SYSTEM'	1-3, 6-19, 22, 24, 26-29, 31-33, 35-37, 39 43-45
Y A	see the whole document ---	21

US 9203812
SA 60557

This annex lists the patent family members relating to the patent documents cited in the above-mentioned international search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information. 09/09/92

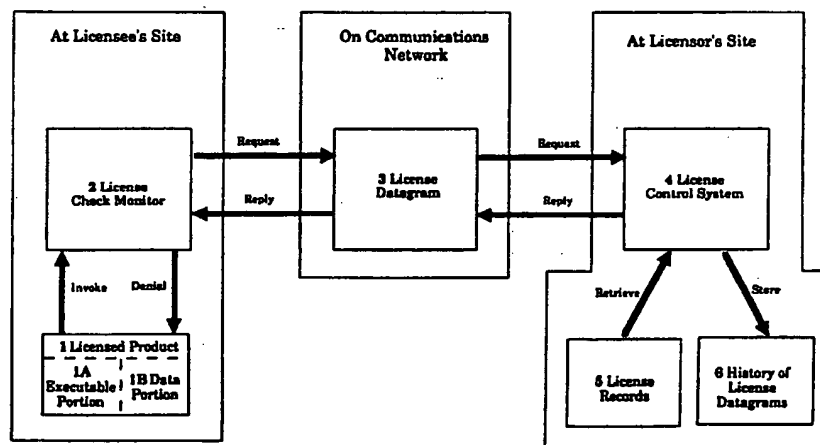
Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP-A-0332304	13-09-89	US-A- 4937863	26-06-90
		JP-A- 2014321	18-01-90

EPO FORM 0079

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁵ : G06F 11/34, H04L 9/00	A1	(11) International Publication Number: WO 93/01550 (43) International Publication Date: 21 January 1993 (21.01.93)
(21) International Application Number: PCT/US92/05387 (22) International Filing Date: 30 June 1992 (30.06.92) (30) Priority data: 724,180 1 July 1991 (01.07.91) US 907,934 29 June 1992 (29.06.92) US (71) Applicant: INFOLOGIC SOFTWARE, INC. [US/US]; 1223 Peoples Avenue, Suite 5405, Troy, NY 12180 (US). (72) Inventor: GRISWOLD, Gary, N. ; 1937 Regent Street, Schenectady, NY 12309 (US). (74) Agents: LAZAR, Dale, S. et al.; Cushman, Darby & Cush- man, Ninth Floor, 1100 New York Avenue, N.W., Wash- ington, DC 20005-3918 (US).		(81) Designated States: AT, AU, BB, BG, BR, CA, CH, CS, DE, DK, ES, FI, GB, HU, JP, KP, KR, LK, LU, MG, MN, MW, NL, NO, PL, RO, RU, SD, SE, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IT, LU, MC, NL, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, SN, TD, TG). Published <i>With international search report.</i>

(54) Title: LICENSE MANAGEMENT SYSTEM AND METHOD**(57) Abstract**

A license management system and method for recording (6) the use of licensed product (1), and for controlling (4) its use. A licensed product invokes a license check monitor (2) at regular time intervals. The monitor generates request datagrams (3) which identify the licensee and the product and sends the request datagrams over a communications facility to a license control system (4). The license control system maintains a record (6) of the received datagrams, and compares the received datagrams to data stored in its licensee database (5). Consequently, the license control system (4) transmits reply datagrams with either a denial or an approval message. The monitor (2) generates its own denial message if its request datagrams are unanswered after a pre-determined interval of time. The datagrams are counted at the control system to provide billing information.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FI	Finland	MI	Mali
AU	Australia	FR	France	MN	Mongolia
BB	Barbados	GA	Gabon	MR	Mauritania
BE	Belgium	GB	United Kingdom	MW	Malawi
BF	Burkina Faso	GN	Guinea	NL	Netherlands
BG	Bulgaria	GR	Greece	NO	Norway
BJ	Benin	HU	Hungary	PL	Poland
BR	Brazil	IE	Ireland	RO	Romania
CA	Canada	IT	Italy	RU	Russian Federation
CF	Central African Republic	JP	Japan	SD	Sudan
CG	Congo	KP	Democratic People's Republic of Korea	SE	Sweden
CH	Switzerland	KR	Republic of Korea	SN	Senegal
CI	Côte d'Ivoire	LI	Liechtenstein	SU	Soviet Union
CM	Cameroon	LK	Sri Lanka	TD	Chad
CS	Czechoslovakia	LU	Luxembourg	TC	Togo
DE	Germany	MC	Monaco	US	United States of America
DK	Denmark	MG	Madagascar		
ES	Spain				

- 1 -

LICENSE MANAGEMENT SYSTEM AND METHOD

BACKGROUNDField of the Invention

5 The present invention generally relates to
systems for managing licenses of products such as
computer software, video games, CD-ROM information,
movies and other video products, music and other audio
products, multimedia products, and other systems for
up-to-date recording of actual usage of such a
10 licensed product to enable efficient billing therefor.

Description of Related Art

Licenses for information products such as
computer software, music, video products and the like
usually provide licensees with limited rights. The
15 licenses may restrict sites of use, duration of use,
or number of concurrent uses of the products. The
licenses also may limit the use of the products
depending on currentness of licensee's payments.
However, enforcing the conditions of the licenses is
20 difficult, because, in general, the licensed products
may be easily copied or "pirated" and used without the
licensor's knowledge.

Compliance with limited license rights has been
encouraged with copy protection. Known methods of
25 computer software copy protection include putting a

SUBSTITUTE SHEET

physical hole or mark on the diskette containing a product, or placing data on the diskette in a location where no data is expected. A disk with an illegally copied software product usually would not contain the marks. At the beginning of its operation, a copy-protected, but illegally copied software product would search its own diskette for the marks. Upon failing to detect the marks, the software would abort from its normal procedures.

Most software products sold today do not have such copy protection, partly because copy protection renders legitimate duplication of copy protected software difficult, but not impossible. Copy protection frustrates the making of legitimate copies, while not eliminating unauthorized copying. Many software publishers have experienced higher sales by eliminating copy protection schemes.

Another method for enforcing limited licensing rights of computer software is described in U.S. patent No. 4,932,054 to Chou. Chou describes a "coded filter" hardware device which is plugged into a port of a computer. The "coded filter" outputs an authorization control code when a predetermined control code is sent to it. The licensed software functions properly only if the "coded filter" transmits the correct authorization control code to the software.

While devices such as described by Chou have existed for several years, they have not been well accepted by the market. Since the device is attached to the outside of a computer, it can easily be lost or stolen, preventing the use of licensed software. In addition, if a licensee purchased a number of software

products, each of which used Chou's protection scheme, the licensee would collect a stack of "coded filters."

Hershey, in U.S. patent No. 4,924,378, describes a method for limiting the number of concurrent uses of a licensed software product. Each workstation of a network has a license storage area in its local memory. License Management System (LMS) daemons are provided in the network in a number corresponding to the permissible number of concurrent uses of the software product. To use the software, a work station stores a daemon in its license storage area. If all daemons are in use, no further work stations may use the software.

Robert et al., in U.S. patent No. 4,937,863, describe a similar invention. This invention includes a license management facility which accesses a database of license information related to licensed computer software programs. When a user attempts to use a licensed program, the license management facility first checks the database. Access to the licensed product is prevented if licensing conditions related to the product are not satisfied (e.g., expiration of licensing dates, etc).

While the Robert et al. and Hershey patents show effective techniques for controlling licensed computer software, each also reveals components that cannot be easily managed by an average user. A system manager, or someone with special access privileges to the internals of a machine, must install the licensed software. This hinders the distribution of the software.

Licensable products other than computer software have not generally been copy-protected. For example,

video tapes can be easily copied by anyone with two VCR machines, and audio tapes and music CDs can be easily copied to tape. Computer CD-ROMs can be copied to magnetic disk; however, their large information storage capacity relative to that of magnetic disks makes this a very expensive proposition. The introduction of digital audio tape is being delayed, because some view its ability to easily produce very high quality copies as a threat to music royalties.

10 Hellman, in U.S. patent No. 4,658,093, describes means to bill by usage. This is accomplished via communication of an encrypted authorization code from a licensor to a base unit at the licensee's site. The encrypted authorization code contains information
15 related to an identification of the base unit, a number of uses requested, and a random or non-repeating number; however, implementation of Hellman's scheme requires a "base unit", such as a computer, video game unit, record player, video recorder, or
20 video disk player, with a unique identification number. The requirement is difficult to satisfy, because, at the present, only a fraction of such systems on the market have an internally readable serial number for identification. In addition,
25 vendors of these systems provide no guarantees for the uniqueness of any given device's serial number. Furthermore, an internal serial number can change when hardware maintenance is performed on the device. Also, Hellman's approach requires that an identical
30 copy of each software product be stored at the authorization site. These copies are used in the generation of unique keys. The unstated assumption that all copies of a specific version of a software

product are identical is unrealistic. Minor bug fixes to software are often made without generating a new version of the product. Also, some software products, such as those which run on Macintosh computers, are self-modifying.

While Hellman's invention counts each use of the software, it does not monitor the duration of use. Thus, Hellman's system would not be able to bill for extensive use of licensed software if the software were continuously operated. Finally, while Hellman suggests the inclusion of an automated communication system as part of his invention, he does not disclose how this communication system could be implemented. Instead, he mentions non-automated use of telephone and mail. In summary, Hellman's patent is an interesting discussion of cryptographic techniques, but it does not provide a practical, real-world implementation of those techniques.

Shear, in U.S. Patent No. 4,977,594, describes a system and method to meter usage of distributed databases such as CD-ROM systems. The method describes a hardware module which must be part of the computer used to access the distributed database. This module retains records of the information viewed. Once the module storage is filled, the module must be removed and delivered to someone who will charge for the usage recorded therein and set the module back to zero usage. Like Hellman's method, this method requires a hardware module which must be incorporated within the computer so the system can control user access. No database publisher will be able to use this method until there are a very large number of units containing such modules. Hardware manufacturers

will be hesitant to include the module in the design of their computers until there is sufficient demand from customers or publishers for this system. The method and apparatus according to the present invention can be implemented entirely in software and hence does not require special, dedicated computer subsystems.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide a license management system and method which can ensure that a licensed product is used only on machines under which it is licensed.

It is another object of the present invention to provide a license management system and method which may terminate access to a licensed product once its license has expired.

It is yet another object of the present invention to provide a license management system and method which may terminate access to a licensed product when payment for a license is overdue.

It is a further object of the present invention to provide a license management system and method which can limit the number of concurrent uses of a licensed product.

It is yet another object of the present invention to provide a license management system and method which can bill licensees for the duration of actual usage of a licensed product.

The present invention provides an advantageous feature of quickly and effectively implementing license agreements between a licensor and licensee.

The present invention provides another advantageous feature of allowing logic used to control licenses to be easily changed.

5 The present invention provides yet another advantageous feature of detecting, at the licensor's site, many types of attempts to alter the license management system.

10 The present invention provides a further advantageous feature of permitting anyone without special access privileges to install a licensed product.

In the present invention, a licensed product generates request "datagrams," messages transmitted over a communications network. The request datagrams
15 are sent to the licensor's site. At the licensor's site the datagram is compared to information stored in a license database. After the comparison, a reply datagram is sent to the licensee. Upon receiving the reply datagram, the licensed product reacts in
20 accordance with the instructions therewithin. For example if a reply datagram contained a "denial," the licensed product would display an appropriate message to the user and then suspend further execution of its programs.

25 In the present invention, the licensed product is implemented on a network node attached to a communications network that includes the licensor. The network node may be a computer, a CD-ROM player, a tele-computer or other multimedia machine, or any
30 other appropriate device. The node may also be an intelligent type of consumer electronic device used for presenting information, such as an intelligent television, VCR, videodisk player, music CD player,

audio tape player, telephone or other similar device. Further, the communications network may be any two-way network such as a computer network, telephone network, a cellular telephone network or other
5 wireless network, a two-way cable TV network, or any other equivalent system.

Should the user detach the node from the network, the licensed product will fail to receive reply datagrams. Upon several failures to receive reply
10 datagrams, the licensed product will generate its own denial.

After a request datagram has been sent out, a user may be permitted to use the licensed product for a limited duration. This feature may be necessary
15 because of the delays in network communications. When networks are sufficiently fast, use of a licensed product can be postponed until the reply datagram is received.

In the preferred embodiment of the present
20 invention, licensees' network addresses are used to identify the licensees. Other embodiments may use a licensed product serial number or hardware serial numbers for the identification.

A licensed product as in the present invention
25 generates a request datagram after each period of product use. The number of request datagrams received by the licensor can be used to bill the licensee. For example, if datagrams are sent after every hour of product use, the licensee will be billed for the
30 amount equal to the number of request datagrams received by the licensor multiplied by the hourly rate.

The embodiments of the present invention may incorporate a query system at a licensor's site for reporting on problem datagrams. This would allow the licensors to take appropriate actions in accordance
5 with problems associated with each datagram.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other objects and advantages of this invention will become more apparent and more readily appreciated from the following detailed description
10 of the presently preferred exemplary embodiment of the invention, taken in conjunction with the accompanying drawings, of which:

FIGURE 1 is a general block diagram of the preferred exemplary embodiment of the present
15 invention;

FIGURE 2 shows representative diagrams of the contents and formats of data at licensee's site, contained in datagrams, and at licensor's site;

FIGURE 3 illustrates a sequence of representative
20 operations executed at the licensee's site and at the licensor's site, together with required inputs for the execution of the operations and with outputs produced therefrom;

FIGURE 4 illustrates a sequence of representative
25 operations to send a request datagram, together with required inputs for the execution of the operations and with outputs produced therefrom;

FIGURE 5 illustrates a sequence of representative
30 operations when a reply datagram is overdue, together with required inputs for the execution of the operations and with outputs produced therefrom;

FIGURE 6 shows a sequence of representative operations to process a reply datagram, together with required inputs for the execution of the operations and with outputs produced therefrom;

5 FIGURE 7 shows a sequence of representative operations to generate an authorization code, together with required inputs for the execution of the operations and with outputs produced therefrom; and

10 FIGURE 8 shows a sequence of representative operations to send a reply datagram, together with required inputs for the execution of the operations and with outputs produced therefrom.

DETAILED DESCRIPTION OF THE
PRESENTLY PREFERRED EXEMPLARY EMBODIMENT

15 As shown in FIGURE 1, a licensed product 1 is located at a licensee's site. Product 1 may include a data portion 1B and a functional portion 1A such as computer software product or any other kind of information product used to control use of data
20 portion 1B. If data portion 1B is CD-ROM database information, functional portion 1A should enable the licensee to search indexes and display text. If data portion 1B is video information, functional portion 1A should control the display of the video information.
25 For audio information, functional portion 1A should play the audio information. If data portion 1B is an electronic book, functional portion 1A should display and turn pages. The above examples show some of the ways functional portion 1A can control data portion
30 1B; however, they are hardly exhaustive.

By including in product 1 both information and software which controls the information, product 1 is

an executable product. Non-software information in product 1 is preferably encrypted so that it cannot be easily extracted from the product.

License check monitor 2 sends license datagrams 3 to the licensor and also receives license datagrams 3 from the licensor. License check monitor 2 also prevents further use of product 1 when a datagram 3 containing a "denial" message is received.

License datagrams 3 are messages that describe information related to the use of licensed product 1. Datagrams 3 are sent over a communications network between the licensee and licensor. Initially, the licensee sends a request datagram 3 over the network to the licensor. The licensor then returns a reply datagram containing either an approval or denial. It is also possible to implement the present invention by having the licensor transmit a reply datagram only for approvals.

At the licensor's site, license control system 4 makes licensing decisions by comparing request datagram 3 with license records 5. After the comparison, control system 4 stores information related to request datagram 3 into history of license datagram record 6. It is noted that request datagrams 3 are periodically sent while product 1 is in use. Thus, the history of license datagrams in record 6 provides means for measuring the duration of use of product 1.

Representations of data and records stored at the licensee's site, contained in datagrams, and stored at the licensor's site are illustrated in FIGURE 2. At the licensee's site, network service 7, which handles delivery and transmission of datagrams 3, supplies

network address 8. It is by this address that license control system 4 identifies a location of use of product 1.

5 Licensed product record 9 is contained within monitor 2. Within the license product record 9 is an identification record 10, which contains the following two items: licensor's network address 11, and product model number 12 that identifies product 1. When a
10 licensor has only one product, or uses different licensor network addresses 11 for each product, product model number 12 may not be needed.

Datagram sent record 13 stores information about the last sent datagram 3. It includes a datagram number 14, which uniquely identifies the last
15 transmitted datagram 3, and the date and time 15 when the last datagram 3 was sent from the licensee's site.

Licensed product record 9 also contains control parameters record 16, which is used for controlling the timing of key events in the communication of
20 license check monitor 2 with license control system 4. Send interval 17 specifies a time interval between each transmission of a new datagram 3 from the licensee to the licensor.

Wait interval 18 is the length of time that
25 monitor 2 waits to receive a reply datagram 3 before resending the same request datagram 3. The duration of this interval depends on the speed of the communications network being used to deliver datagrams 3.

30 Disconnect allowed interval 19 is the duration of time that monitor 2 allows product 1 to be used without a reply datagram 3 from the licensor. The duration of this interval depends on the reliability

of the communications network. The interval must be long enough to take into consideration network downtime. For example, suppose a message was sent from the licensor and the network went down just afterwards. Disconnect allowed interval 19 should be long enough to allow the network to resume its normal operation and successfully deliver datagrams 3 from the licensor; otherwise, the licensee would be forced to stop using product 1 until the network was operational.

License datagram 3 contains header 20. Header 20 is used during execution of low level communication protocols within the network. Source network address 21 is the network address from where datagram 3 is sent. Destination network address 22 is the network address to where datagram 3 is sent. Additional data may be included in header 20 if required by low level protocols used in delivering datagrams 3.

Data 23, a part of datagram 3, conveys a message, and contains a number of fields. Product model number 24 and datagram number 25 identify product 1 and datagram 3, respectively. It is noted that retransmitted datagrams have an identical datagram number. Duplicate datagrams must be identified at a licensor's site so that they do not all contribute in billing a licensee.

Each datagram number 25 is unique for each request datagram 3 transmitted from the licensee, except for retransmitted datagrams. This allows a reply datagram 3 received by a licensee to be verified as an actual reply to a request datagram 3 from that licensee, as explained below.

Number of processes running 26 is the number of concurrent uses of product 1 at the time datagram 3 is sent. Authorization code 27 is used on reply datagrams 3 to indicate an approval or a denial.
5 Message text 28 contains a message which will be displayed to the user upon a denial.

License database 29 at the licensor's site holds records of information about customers, licenses, and license usage. The types of information within
10 license database 29 of the present embodiment are shown in FIGURE 2. However, a specific license management system may require its license database to hold types of information other than those in FIGURE 2. For example, licensee name and address may be
15 incorporated as a part of a license database 29.

License record 5 contains information on licenses. Licensee network address 30 identifies a precise network node which is licensed to use product 1. If request datagrams are received which do not
20 originate from known licensee network addresses 30, reply datagrams containing denial messages are transmitted. Product model number 31 is the model number of a licensed product. Termination date 32 is the expiration date of a license. When the license of
25 a product is issued for an unlimited duration, termination date 32 should reflect a date very far into the future, relative to the licensing date.

The present embodiment allows licenses to be paid for in a lease-like or rental fashion. If a licensee
30 were to rent or lease product 1, paid through date 33 would reflect the date through which the licensee has paid for using the product. Grace period 34 is the time interval for which the licensee is allowed to be

delinquent before services are disconnected. Grace period 34 would reflect a very large time interval if the license is not of a lease-like or rental type. When the license provides for a limit on the number of concurrent uses of a product 1, number of processes licensed 35 contains the limiting number. When the license does not provide for such a limit, number of processes 35 should be a very large number.

History of license datagrams 6 is an archive of datagrams 3 received from the licensee.

FIGURE 3 illustrates operations executed at the licensee's site and at the licensor's site. An overview of the processing at the licensee's site is described by steps 101.0 to 106.0, and an overview of the processing at the licensor's site is described by steps 107.0 to 110.0.

At the licensee's site, at step 101.0, product 1 invokes monitor 2. This is accomplished by first establishing monitor 2 as a handler for a timer expiration interrupt signal and for received datagrams 3. Next, a timer is set with a very short time to cause an initial call to monitor 2. At step 102.0, monitor 2 computes a time 36 since the last datagram was sent by determining the difference between the current date and sent time and date and time 15 that a datagram was last sent from the licensee's site. When product 1 commences execution, datagram sent date and time 15 is set to "null." Thus, time since send 36 is very large at the beginning of the monitor's execution. At step 103.0, time since send 36 is compared to send interval 17. If time since send 36 is greater than send interval 17, then a request datagram is transmitted, per the steps described in

FIGURE 4. Step 104.0 first checks if a reply to the last datagram has arrived and if wait interval 18 has expired. If a reply has not arrived and the wait interval has expired, steps 104.1-104.3 (FIGURE 5) are executed. Step 105.0 processes authorization code 27 in a reply when the reply is received, in accordance with steps 105.1 to 105.5 (FIGURE 6). At step 106.0, product 1 resumes normal execution of its programs until the next interrupt signal is generated.

10 At the licensor's site, license control system 4 receives and processes datagram 3, in accordance with steps 107.0 to 110.0. Step 107.0 receives request datagram 3. Step 108.0 generates authorization code 27, per steps 108.1 to 108.8 (FIGURE 7). Step 109.0
15 creates reply datagram 3 and transmits the datagram to the licensee via steps 109.1 to 109.5 (FIGURE 8).

FIGURE 4 shows the procedure which monitor 2 follows for sending request datagram 3 to the licensor. Step 103.1 sets source network address 21
20 in datagram 3 to the network address 8 of the licensee's location on the network. Step 103.2 sets destination network address 22 to licensor's network address 11. Step 103.3 encrypts product model number 12 for datagram 3. Step 103.4 assigns a unique number
25 to datagram 3, encrypts the number, and stores it as datagram number 14. This number is altered when an entirely new datagram 3 is sent. Datagrams which are retransmitted have the same datagram number 25 as the original. As already explained, this allows license
30 control system 4 to identify duplicate datagrams.

Step 103.5 counts the number of processes using product 1, currently running, encrypts the count, and stores the encryption as the number of processes

- 17 -

running 26. In the UNIX operating system, this procedure could be performed using the command "ps" to obtain a list of current processes, the command "grep" to extract the processes of product 1, and "wc" to
5 count the number of processes. Step 103.6 sets authorization code 27 to number 255 and encrypts the number.

Number 255 indicates that datagram 3 is a request for authorization. Such an indication is needed to
10 guard the present system against the following steps for circumventing the present invention: intercepting outgoing datagrams; and inputting the intercepted datagrams to monitor 2.

Step 103.7 stores the current date and time as
15 sent date & time 15. This date is needed to compute when to send the next datagram 3. Step 103.8 assigns a value to send interval 17, which sets an alarm for invoking monitor 2 to send the next datagram 3. Step 103.9 sends datagram 3.

20 In the present embodiment a datagram is transmitted via a connectionless datagram service. Methods for transmission are well documented for some networking systems. For example, TCP/IP (Transport Control Protocol/Internet Protocol) includes a con-
25 nectionless protocol called UDP (User Datagram Protocol). A method for sending a datagram using UDP protocol from a SUN Microsystem computer is documented in a SUN manual titled, Network Programming Guide, in section 9 titled "Transport Level Interface
30 Programming."

Step 103.10 sets another alarm using wait interval 18 for retransmitting datagram 3, if no reply datagram has been received. The alarm causes monitor

- 18 -

2 to be invoked for checking whether a reply datagram
3 has been received. Monitor 2 will transmit a
duplicate of the previously transmitted datagram, if
no reply has been received. After the execution of
5 step 103.10, "Send License Datagram" procedure returns
system control to step 104.0 in FIGURE 3.

FIGURE 5 shows the operation of the "Reply
Datagram is Overdue" procedure. Step 104.1 compares
time since the last datagram was sent 36 to disconnect
10 allowed interval 19, which, as described above, is the
interval that product 1 is allowed to operate even if
a reply is overdue. If time since send 36 is smaller
than disconnect allowed interval 19, datagram 3 is
retransmitted via executing step 103.9 in FIGURE 4.
15 Step 104.2 "disconnects" product 1 from further
service, if time since send 36 is greater than
disconnect allowed interval 19.

Step 104.2 comprises a sequence of sub-steps
104.2.1-104.2.3. Step 104.2.1 assigns number 5 to
20 authorization code 27 in the current datagram being
processed. Value 5 is interpreted by monitor 2 as a
denial. Step 104.2.2 sets message text 28 to the
following: "A reply from licensor to numerous
authorization requests was never received. This
25 product must be connected to a communications network
in order to function." Step 104.2.3 transfers system
control to step 105.3 in FIGURE 6. Step 105.3
processes the current denial datagram 3 as if it were
just received.

30 Through the execution of steps 104.1-104.3, the
present system permits the use of product 1 for a
prescribed period of time. After the prescribed

- 19 -

period of time has elapsed, the present system generates a denial.

FIGURE 6 illustrates the steps which monitor 2 follows in processing a reply datagram 3. Step 105.1
5 decrypts all encrypted data in the received datagram. Step 105.2 compares datagram number 25 with datagram number 14 associated with the last datagram. If datagram number 25 is not equal to datagram number 14, step 105.2 ignores the current datagram and transfers
10 procedural control to step 103.9 (FIGURE 4) in order to resend the last transmitted datagram. After disconnect allowed interval 19 elapses, monitor 2 generates a denial.

In essence, step 105.2 guards against the
15 circumvention of the present invention via: (1) intercepting a reply datagram 3 (from the licensor) containing an approval (2) storing the reply datagram 3; and (3) inputting the stored datagram to monitor 2.

If the execution of step 105.2 does not transfer
20 its procedural control to step 105.3, and if authorization control 27 is not zero (indicating an unqualified authorization has not been received), step 105.3 processes authorization code 27 via steps 105.3.1 to 105.3.3. Step 105.3.1 retrieves message
25 text 28 from datagram 3. If message text 28 is null, then the current datagram 3 is ignored, and monitor 2 resends the last transmitted datagram 3. Step 105.3.1 further protects the present system from attempts to generate fake datagrams and to feed the fake datagrams
30 to monitor 2 by checking for a proper authorization code of zero.

If message text 28 is not null, step 105.3.2 presents the message 28 to the user on an output

- 20 -

device such as a CRT screen. Step 105.3.3 terminates the current use of product 1. This step may be implemented by subroutine or function call to a simple exit that saves any current user data to a file.

5 Alternatively, product 1 may be designed so that, upon being directed to terminate further execution, it first gives the user an opportunity to save their data.

10 If authorization code 27 is zero, step 105.4 allows further use of product 1. Step 105.5 returns procedural control to 106.0 on FIGURE 3.

FIGURE 7 shows a sequence of operations within the "Generate Authorization Code" procedure. The procedure produces appropriate authorization code 27

15 when a request datagram 3 is received at the licensor's site.

Step 108.1 decrypts all encrypted data in the received datagram 3. Using source network address 21 and product model number 24 in the datagram 3, step

20 108.2 searches the license database 29 for matching licensee network address 30 and product model number 31. If license database 29 does not contain a record of product model number 24 of the product 1 being licensed to the licensee, step 108.3 sets

25 authorization code 27 of its reply datagram 3 to 1 (i.e., the sending node is not a registered address) and authorization is denied.

Step 108.3 prevents copies of product 1 from being installed on multiple nodes independently of

30 whether they are within or outside the licensee's organization. Step 108.3 also prevents the licensee from transporting product 1 from one node to another node without the licensor's approval. This is

important because the two nodes may have different processing capacities, and they may be billable at different rates.

5 If the date a request datagram is received is later than license termination date 32, step 108.4 sets authorization code 27 to number 2 (i. e., license has expired). Step 108.4 allows the licensor to fix licensing periods, or to determine free trial periods for the use of the product. The licensing period may
10 be extended by resetting license termination date 32 at the licensor's site.

If the date when the datagram is received is later than the paid through date 33 as extended by the grace period 34, step 108.5 sets authorization code 27
15 to 3 (i.e., payment is past due).

If the number of processes running 26 exceeds a licensed number of concurrent uses of product 1 (at a particular node), then step 108.6 sets authorization code 27 to 4 (i.e. concurrent process usage limit is
20 exceeded).

Step 108.7 sets authorization code 27 to 0 indicating processing can continue. It is noted that steps 108.3-108.7 are a part of a

25 IF (x1) then (y1)
ELSE if (x2) then (y2)
ELSE if (x3) then (y3) ...

statement of a procedure (e. g., FORTRAN, PASCAL, C, etc). Thus, only one of the steps 108.3-108.7 is executed. Step 108.7 sets authorization code 27 to 0
30 (indicating approval of further use) only if steps 108.3-108.6 do not execute the THEN portion of each step. Step 108.7 also stores the received datagram 3 in history of license datagrams 6.

- 22 -

Step 108.8 is the last of authorization processing rules 108.1-108.7. After the execution of steps 108.3-108.7, step 108.8 returns procedural control to step 109.0 in FIGURE 3.

5 FIGURE 8 illustrates the steps which license control system 4 follows to send reply datagram 3 to the licensee.

Step 109.1 encrypts authorization code 27 and writes the encrypted code into datagram 3. Next, step
10 109.2 writes message text 28 corresponding to authorization code 27 into datagram 3.

Step 109.2 may be replaced with the following method for relaying proper messages to a product user. Proper messages corresponding to each authorization
15 code is stored in monitor 2 at each licensee's site. Upon reception of a reply datagram 3, monitor 2 would locate within itself the proper message corresponding to the authorization code, and use the message for various purposes. This method would reduce the size
20 of reply datagrams 3. However, if the licensor wanted to implement new denial codes, each product would need to somehow incorporate the new message associated with the new denial code into itself. The list of messages, one of which may be written as message text
25 28, are as follows:

AUTHORIZATION CODE

TEXT MESSAGE

30	1 This product is not licensed to run at this location. Please contact the licensor to either license this product, or move an existing license of your organization to this location. Use of this product at this

- 23 -

location is discontinued until this problem is resolved.

2
5 Your license on this product has expired. Please contact licensor in order to have your license extended. Use of this product is discontinued until this problem is resolved.

10 3 Payment on this licensed product is over due and past your grace period. Please have your accounting department send payment in order to continue your license. Use of this product is discontinued until this problem is resolved.

20 4 Your current use of this licensed product exceeds limits for the number of uses your organization has licensed. Please try again later.

25 5 A reply from licensor to numerous authorization requests was never received. This product must be connected to a communications network in order to function.

0 Authorization is OK. There is no message.

30 Step 109.3 swaps source network address 21 and destination network address 22. Step 109.4 transmits datagram 3 back to monitor 2.

35 At step 109.5, a communications network delivers datagram 3 to monitor 2. Subsequently, procedural control returns to step 107.0 in FIGURE 3 to process the next datagram 3.

Although only a few exemplary embodiments of this invention have been described in detail above, those skilled in the art will readily appreciate that many

modifications are possible in the preferred embodiments without materially departing from the novel teachings and advantages of this invention. For example, product 1 was described as sometimes
5 consisting of information as well as software which controls the information. This approach provides the greatest flexibility, but it is also possible to include the software which controls the information in the networked machine at the licensee's site. In this
10 case, product 1 is split, with part of it on media and part on the licensee's machine. By doing this, some space can be saved on the media containing product 1, but the capabilities of these products will be limited by the standard functions available on these machines.

15 Also, the presently described embodiment includes a product 1 which is at the licensee's site. This implies that product 1 is on some physical media such as diskette, tape, or CD. However, product 1 can be electronically delivered over communications lines to
20 the licensee and therefore might exist in the memory of the licensee's machine, rather than any physical media. In the case of a product such as music, radio programs and the like, product 1 may even be broadcast to the licensee's site for playback; thus, the product
25 1 would not even be "resident" in the licensee's machine.

The presently described embodiment allows the licensee to access the licensed product concurrent with the sending and receiving of datagram 3. In this
30 way, the present invention does not inconvenience the legitimate licensee; however, for sensitive licensed products such as confidential information, the license

check monitor 2 can prevent access to the product 1 until an authorization reply datagram 3 is received.

Further, monitor 2 could be realized as an integral part of product 1. Monitor 2 could also be implemented as: 1) a separate process which is the parent process of product 1 (Such a parent process would have the authority to cancel the use of product 1); 2) a single system level task which controls license checking of all products at the licensee's site; and 3) custom logic in a digital integrated circuit (the present invention could be implemented as hardware instead of software).

Also, though the above embodiment has been described as being implemented on a computer system network where operator messages are provided on a CRT monitor or the like, the invention may be practiced on other hardware platforms by incorporating appropriate changes known to those of ordinary skill in the art. For example, in an alternative hardware embodiment such as a music or video playback device, monitor 2 is invoked by the licensee's action of pushing the "play" or similar button, and in a broadcast music application or similar system, the monitor may be invoked simply by turning the device on. The processing of monitor 2 is as described in the presently described embodiment. However, when a denial message is received or generated, monitor 2 must be able to switch "play" to "off".

The presently described embodiment is designed to be used in conjunction with a connectionless UDP (User Datagram Protocol) in the TCP/IP protocol suite as an underlying protocol. However, the present invention could also be realized using a slower,

connectionless protocol such as electronic mail or a variety of connection protocols (e. g., File Transfer Protocols (FTP), Telnet).

5 It is noted that protocol suites quite different from TCP/IP could be used, such as ISO (International Standards Organization) protocol. In addition, datagrams 3 could be sent over telephone systems with communications protocols such as those specified by CCITT (Consultative Committee on International
10 Telephony and Telegraphy). In this case, telephone numbers could serve as network addresses 21, 22. Communications protocols for wireless communications such as cellular telephone can also be used to send the datagram 3.

15 Accordingly, all such modifications are intended to be included within the scope of this invention as defined by the following claims.

WHAT IS CLAIMED IS:

1. A method for monitoring the use of a licensed product, comprising the steps of:
generating, at regular time intervals,
5 datagrams including an address in a communications facility, said facility address identifying a licensee;
automatically sending said datagrams from at least one licensee's site over said facility to a
10 licensor's site while said licensed product is in use;
receiving said datagrams at said licensor's site;
storing an indication of receipt of each of said datagrams; and
15 counting said datagrams from each licensee as an indication of the use by the licensee of said licensed product.
2. A method as in claim 1 further wherein:
said generating step includes the step of
20 incorporating a model number of said product in said datagrams; and
said counting step includes the step of separately counting datagrams for each product model number for each licensee.
- 25 3. A method as in claim 1, wherein said generating step includes the step of automatically obtaining said facility address that identifies said licensee from said facility without any data being provided by said licensee.

- 28 -

4. A method for controlling use of a licensed product comprising the steps of:

generating a request datagram including an address in a communications facility, said facility
5 address identifying a licensee;

automatically sending said request datagram from at least one licensee's site over said facility to a licensor's site while said licensed product is in use;

10 receiving said request datagram at said licensor's site;

comparing said received request datagram with rules and license data at said licensor's site to determine if use of said licensed product is
15 authorized;

sending a reply authorizing datagram to said licensee's site if use of said licensed product is approved; and

receiving said reply authorizing datagram at
20 said licensee's site and denying the use of said product when no reply authorizing datagram is received.

5. A method as in claim 4, wherein:

said generating step includes the step of
25 incorporating a model number of said product in said datagram;

said comparing step includes the step of comparing said rules and license data for a particular model number; and

30 said sending step includes the step of transmitting said reply datagram for each product model number.

SUBSTITUTE SHEET

- 29 -

6. A method as in claim 4, wherein said generating step includes the step of automatically obtaining said facility address that identifies said licensee from said facility without any data being
5 provided by said licensee.

7. A method as in claim 4 further comprising the step of sending a reply denial datagram if use of said licensed product is not approved as determined in said comparing step, said step of automatically
10 sending said request datagram from a licensee's site including the step of resending said request datagram if neither a reply authorizing datagram nor a reply denial datagram is received from said licensor's site within a predetermined time from sending said request
15 datagram from said licensee's site.

8. A method as in claim 4, wherein said step of automatically sending said request datagram from said licensee's site includes the step of sending a request datagram at regular time intervals.

20 9. A method as in claim 4, wherein:
said generating step includes the step of providing a datagram identification code within said datagram;
said reply datagram sending step includes
25 the step of inserting the same datagram identification code in said reply datagram; and
said reply receiving step rejects said reply authorizing datagram if the datagram identification code included in said reply authorizing datagram does

SUBSTITUTE SHEET

- 30 -

not match the datagram identification code included in said request datagram.

10. A method as in claim 4, wherein:

5 said comparing step includes the step of
comparing said facility address that identifies said
licensee with a list of valid licensee addresses to
determine if said facility address is a valid address;
and

10 said reply authorizing datagram is not sent
if said facility address that identifies said licensee
is not valid.

11. A method as in claim 10 further comprising
the step of sending a reply denial datagram if said
facility address that identifies said licensee is not
15 valid.

12. A method as in claim 4, wherein:

 said comparing step includes the step of
comparing a license expiration date with a date at
which said datagram is received; and

20 said reply authorizing datagram is not sent
if the license expiration date is later than the date
at which said datagram is received.

13. A method as in claim 12, further comprising
the step of sending a reply denial datagram if the
25 license expiration date is later than the date at
which said datagram is received.

14. A method as in claim 4, wherein:

SUBSTITUTE SHEET

said comparing step includes the step of checking currentness of payments from said license; and

5 said reply authorizing datagram is not sent if payment is overdue.

15. A method as in claim 14, further comprising the step of sending a reply denial datagram if payment is overdue.

16. A method as in claim 4, wherein:

10 said generating step includes the step of incorporating in said datagram data indicative of the number of processes currently using said product at said licensee's site;

15 said comparing step includes the step of comparing the number of processes using said product at the licensee's site to an authorized number; and

 said reply authorizing datagram is not sent if said number of processes using said product exceeds said authorized number.

20 17. A method as in claim 16, further comprising the step of sending a reply denial datagram if said number of processes using said product exceeds said authorized number.

25 18. A method as in claim 4, wherein said sending step includes the steps of sending said reply authorizing datagram when use of said product is approved and sending a reply denial datagram when use of said product is not approved, said receiving step

denying use of said product when said reply denial datagram is received.

19. A method as in claim 18, wherein said receiving and denying step denies use of said product
5 when neither a reply authorizing datagram nor a reply denial datagram is received within a predetermined time after said request datagram is sent.

20. A method as in claim 18, further comprising the step of indicating, at a licensee's site, a reason
10 for denial when said reply denial datagram is received.

21. A method as in claim 4, wherein:
said licensed product comprises an executable portion and a data portion; and
15 said method further comprises a step of controlling use of said data portion with said executable portion.

22. A method as in claim 4 further comprising a step of allowing use of said licensed product before
20 a reply datagram is received.

23. A system for controlling licensed product comprising:
a communications facility to which at least one licensee having a license for operating a licensed
25 product from the licensor is connected;
monitoring means, connected to said facility at a site of each said licensee, for generating a request datagram including an address of said licensee

- 33 -

on said facility and transmitting said request datagram over said facility to a site of said licensor, and for receiving and processing a reply datagram; and

5 controlling means, connected to said facility at said licensor's site, for receiving said request datagram, comparing said request datagram with rules and license data to determine if use of said licensed product is authorized and sending a reply
10 authorizing datagram to said licensee's site if use of said product is approved; and

 said monitoring means including means for denying use of said licensed product when no reply authorizing datagram is received.

15 24. A system as in claim 23, wherein:

 said monitoring means sends request datagrams at regular time intervals during use of said licensed product; and

 said controlling means further comprises
20 means for counting said request datagrams received at said controlling means and means for computing an amount to be billed to said licensee in response to said counting.

 25. A system as in claim 23 wherein:

25 said monitoring means incorporates a model number for said product in said request datagram; and

 said controlling means comprises means for counting datagrams for each product model number for each licensee, in order to compute an amount to be
30 billed to each licensee.

- 34 -

26. A system as in claim 23, wherein said monitoring means automatically obtains said facility address of said licensee from said facility without any input from said licensee.

5 27. A system as in claim 23, wherein:
 said controlling means sends a reply denial
 datagram to said licensee's site if use of said
 product is not approved; and
 said monitoring means resends said request
10 datagram if no reply authorizing datagram and no reply
 denial datagram is received within a predetermined
 period of time after said requesting datagram is sent.

 28. A system as in claim 23, wherein said
 monitoring means transmits request datagrams at
15 predetermined time intervals.

 29. A system as in claim 23, wherein:
 said monitoring means incorporates a unique
 identification code in said request datagram;
 said controlling means incorporates the same
20 request datagram identification code in said reply
 authorizing datagram; and
 said monitoring means rejects any reply
 authorizing datagram which does not include the same
 identification code as included in said request
25 datagram.

 30. A system as in claim 23, wherein said
 controlling means compares said facility address of
 said licensee with a list of valid licensee facility
 addresses and does not generate a reply authorizing

- 35 -

datagram if said facility address of said licensee is not valid.

31. A system as in claim 30, wherein said controlling means sends a reply denial datagram when
5 said facility address is not valid.

32. A system as in claim 23, wherein said controlling means compares an expiration date of a license of said product with a date at which said request datagram is received by said controlling
10 means, and does not generate a reply authorizing datagram, thus denying use of said product, if the license expiration date is earlier than the date at which said request datagram is received.

33. A system as in claim 32, wherein said
15 controlling means sends a reply denial datagram if the license expiration date is earlier than the date at which said request datagram is received.

34. A system as in claim 23, wherein said controlling means generates a reply authorizing
20 datagram, thus denying use of said product, if a payment for the use of said product is overdue.

35. A system as in claim 34, wherein said controlling means sends a reply denial datagram if payment for the use of said product is overdue.

25 36. A system as in claim 23, wherein:
said monitoring means includes in said request datagram data indicative of the number of

processes, at a licensee's site, currently using said product; and

5 said controlling means does not generate a reply authorizing datagram, thus denying a use of said product, if more than a predetermined number of processes using said product are running at the licensee's site.

10 37. A system as in claim 36, wherein said controlling means sends a reply denial datagram if more than said predetermined number of processes using said product are running at the licensee's site.

38. A system as in claim 23, wherein said controlling means sends a reply denial datagram if use of said product is not approved.

15 39. A system as in claim 38, wherein said monitoring means denies use of said licensed product when no reply authorizing datagram and no reply denial datagram is received within a predetermined time from the sending of said request datagram.

20 40. A system as in claim 38, further comprising means for indicating, at a licensee's site, a reason for denial when said reply denial datagram is received.

25 41. A system as in claim 23, wherein:
 said licensed product comprises an executable portion and a data portion; and

- 37 -

said system further comprises means for controlling use of said data portion with said executable portion.

42. A system as in claim 41, wherein said data
5 portion controlling means is disposed within said executable portion.

43. A system as in claim 41, wherein said data
portion controlling means comprises a first partial
controlling means disposed within said executable
10 portion and a second partial controlling means
disposed within said monitoring means.

44. A system as in claim 23, wherein said
monitoring means includes means for permitting use of
said licensed product before a reply datagram is
15 received.

45. A system for monitoring product comprising:
a communications facility to which at least
one licensee having a license for operating a licensed
product from a licensor is connected;
20 monitoring means, connected to said facility
at a site of each said licensee, for generating
datagrams including an address of said licensee on
said facility and transmitting said datagrams at
periodic intervals over said facility to a site of
25 said licensor; and
control means, connected to said facility at
said licensor's site, for receiving said request
datagrams, storing an indication of receipt of each of
said datagrams and counting said datagrams from each

- 38 -

licensee as an indication of the use by the licensee of said licensed product.

46. A system as in claim 45, wherein said monitoring means automatically obtains said facility address of said licensee from said facility without
5 any input from said licensee.

47. A system as in claim 45, wherein:
said monitoring means incorporates a product model number in said request datagrams; and
10 said controlling means separately counts request datagrams for each product model number for each licensee.

48. A method for monitoring the use of a licensed product comprising the steps of:
15 generating, at regular time intervals, datagrams including an address in a communications facility, said facility address identifying a licensee; and
automatically sending said datagrams from at
20 least one licensee's site over said communications facility to a licensor's site while said licensed product is in use.

49. A method as in claim 48 further wherein:
said generating step includes the step of
25 incorporating a model number of said product in said datagrams.

50. A method as in claim 48, wherein said generating step includes the step of automatically

obtaining said facility address that identifies said licensee from said communications facility without any data being provided by said licensee.

51. A method for controlling use of a licensed
5 product comprising the steps of:

generating a request datagram including a facility address that identifies a licensee in a communications facility;

10 automatically sending said request datagram from a licensee's site over said communications facility to a licensor's site while said licensed product is in use; and

receiving a reply authorizing datagram at said licensee's site and denying the use of said
15 product when no reply authorizing datagram is received.

52. A method as in claim 51 wherein:

said generating step includes the step of incorporating a model number of said product in said
20 datagram.

53. A method as in claim 51, wherein said generating step includes the step of automatically obtaining said facility address that identifies said licensee from said communications facility without any
25 data being provided by said licensee.

54. A method as in claim 51, wherein:

said reply datagram is one of at least a reply authorization datagram and a reply denial datagram; and

- 40 -

said step of automatically sending said request datagram from a licensee's site includes a step of resending said request datagram if neither a reply authorizing datagram nor a reply denial datagram is received within a predetermined time from sending said request datagram from said licensee's site.

55. A method as in claim 51, wherein said step of automatically sending said request datagram from said licensee's site includes the step of sending a request datagram at regular time intervals.

56. A method as in claim 51, wherein:
said generating step includes the step of providing a datagram identification code within said datagram; and
said reply receiving step rejects said reply authorizing datagram if the datagram identification code included in said reply authorizing datagram does not match the datagram identification code included in said request datagram.

57. A method as in claim 51, wherein:
said generating step includes the step of incorporating in said datagram data indicative of the number of processes currently using said product at said licensee's site.

58. A method as in claim 51, further comprising the steps of:
receiving a reply denial datagram; and
displaying, at a licensee's site, a reason for denial when said reply denial datagram is received.

- 41 -

59. A method as in claim 51, wherein:

said licensed product comprises an executable portion and a data portion; and

5 said method further comprises a step of controlling use of said data portion with said executable portion.

60. A method as in claim 51 further comprising a step of allowing use of said licensed product before a reply datagram is received.

10 61. A system for controlling a licensed product comprising:

a communications facility to which at least one licensee is connected;

15 monitoring means, connected to said communications facility at a site of each said licensee, for generating a request datagram including an address of said licensee on said communications facility and transmitting said request datagram over said communications facility, and for receiving and
20 processing a reply authorizing datagram; and

means for denying use of said product when no reply authorizing datagram is received.

62. A system as in claim 61, wherein:

25 said monitoring means sends request datagrams at regular time intervals during use of said licensed product.

63. A system as in claim 61 wherein:

- 42 -

said monitoring means incorporates a model number for said product in said request datagram.

64. A system as in claim 61, wherein said monitoring means automatically obtains said facility address of said licensee from said communications facility without any input from said licensee.

65. A system as in claim 61, wherein:
said monitoring means resends said request datagram if no reply authorizing datagram and no reply denial datagram is received within a predetermined period of time after said requesting datagram is sent.

66. A system as in claim 61, wherein said monitoring means transmits request datagrams at predetermined time intervals.

67. A system as in claim 61, wherein:
said monitoring means incorporates a unique identification code in said request datagram; and
said monitoring means rejects any reply authorizing datagram which does not include the same identification code as included in said request datagram.

68. A system as in claim 61, wherein:
said monitoring means includes in said request datagram data indicative of the number of processes, at a licensee's site, currently using said product.

69. A system as in claim 61, wherein:

5 said monitoring means denies use of said
licensed product when no reply authorizing datagram
and no reply denial datagram is received within a
predetermined time from the sending of said request
datagram.

70. A system as in claim 61, further comprising
means for indicating, at a licensee's site, a reason
for denial when a reply denial datagram is received.

10 71. A system as in claim 61, wherein:
said licensed product comprises an executable
portion and a data portion; and
said system further comprises means for
controlling use of said data portion with said
executable portion.

15 72. A system as in claim 71, wherein said data
portion controlling means is disposed within said
executable portion.

20 73. A system as in claim 71, wherein said data
portion controlling means comprises a first partial
controlling means disposed within said executable
portion and a second partial controlling means
disposed within said monitoring means.

25 74. A system as in claim 61, wherein said
monitoring means includes means for permitting use of
said licensed product before a reply datagram is
received.

75. A system for monitoring a licensed product comprising:

a communications facility to which at least one licensee is connected;

5 monitoring means, connected to said communications facility at a site of each said licensee, for generating datagrams including an address of said licensee on said communications facility and transmitting said datagrams at periodic
10 intervals over said communications facility.

76. A system as in claim 75, wherein said monitoring means automatically obtains said communications facility address of said licensee from said communications facility without any input from
15 said licensee.

77. A system as in claim 75, wherein:
said monitoring means incorporates a product model number in said request datagrams.

78. A method for monitoring the use of a
20 licensed product comprising the steps of:

receiving datagrams at a licensor's site on a communications facility having at least one licensee's site thereon, said datagrams being generated at regular time intervals and including a
25 facility address that identifies a licensee in said communications facility;

storing an indication of receipt of each of said datagrams; and

counting said datagrams as an indication of
30 the use of said licensed product.

- 45 -

79. A method as in claim 78 further wherein:
said datagrams include a model number of
each product; and
said counting step includes the step of
5 separately counting datagrams for each product model
number for each licensee.

80. A method for controlling use of a licensed
product comprising the steps of:
receiving a request datagram at a licensor's
10 site on a communications facility having at least one
licensee's site thereon, said request datagram
including a facility address identifying a licensee
and being automatically sent over said communications
facility to said licensor's site while said licensed
15 product is in use;
comparing said received request datagram
with rules and license data at said licensor's site to
determine if use of said licensed product is
authorized; and
20 sending a reply authorizing datagram if use
of said licensed product is approved.

81. A method as in claim 80 wherein:
said datagrams include a model number of
said product;
25 said comparing step includes the step of
comparing said rules and license data for a particular
model number; and
said sending step includes the step of
transmitting said reply datagram for each product
30 model number.

- 46 -

82. A method as in claim 80 further comprising the step of sending a reply denial datagram if use of said licensed product is not approved as determined in said comparing step.

5 83. A method as in claim 80, wherein:
 said datagrams include a datagram
 identification code; and
 said reply datagram sending step includes
 the step of inserting the same datagram identification
10 code in said reply datagram.

 84. A method as in claim 80, wherein:
 said comparing step includes the step of
 comparing said facility address that identifies said
 licensee with a list of valid licensee addresses to
15 determine if said facility address is a valid address;
 and
 said reply authorizing datagram is not sent
 if said facility address that identifies said licensee
 is not valid.

20 85. A method as in claim 84 further comprising
 the step of sending a reply denial datagram if said
 facility address that identifies said licensee is not
 valid.

 86. A method as in claim 80, wherein:
25 said comparing step includes the step of
 comparing a license expiration date with a date at
 which said datagram is received; and

- 47 -

said reply authorizing datagram is not sent if the license expiration date is later than the date at which said datagram is received.

87. A method as in claim 86, further comprising
5 the step of sending a reply denial datagram if the license expiration date is later than the date at which said datagram is received.

88. A method as in claim 80, wherein:
said comparing step includes the step of
10 checking currentness of payments from said license;
and

said reply authorizing datagram is not sent if payment is overdue.

89. A method as in claim 88, further comprising
15 the step of sending a reply denial datagram if payment is overdue.

90. A method as in claim 80, wherein:
said datagrams include data indicative of
the number of processes currently using said product
20 at said licensee's site;

said comparing step includes the step of comparing a number of processes using said product to an authorized number; and

said reply authorizing datagram is not sent
25 if said number of processes using said product exceeds said authorized number.

91. A method as in claim 90, further comprising the step of sending a reply denial datagram if said

number of processes using said product exceeds said authorized number.

92. A method as in claim 80, wherein said sending step includes the steps of sending said reply
5 authorizing datagram when use of said product is approved and sending a reply denial datagram when use of said product is not approved.

93. A system for controlling a licensed product comprising:

10 a communications facility to which at least one licensee and a licensor are connected at a licensee's site and at a licensor's site, respectively; and

15 controlling means, connected to said communications facility at said licensor's site, for: receiving a request datagram, said request datagram including an address of said licensee on said communications facility and being transmitted over
20 said communications facility to a site of said licensor; comparing said request datagram with rules and license data to determine if use of said licensed product is authorized; and sending a reply authorizing datagram to said licensee's site if use of said product is approved.

25 94. A system as in claim 93, wherein:

said request datagrams are sent at regular time intervals during use of said licensed product; and

30 said controlling means comprises means for counting said request datagrams received at said

controlling means and means for computing an amount to be billed to said licensee in response to said counting.

95. A system as in claim 93 wherein:

5 said datagrams include a model number for said product; and

 said controlling means comprises means for counting datagrams for each product model number for each licensee, in order to compute an amount to be
10 billed to each licensee.

96. A system as in claim 93, wherein:

 said controlling means sends a reply denial datagram to said licensee's site if use of said product is not approved.

15 97. A system as in claim 93, wherein:

 said datagrams include a unique identification code; and

 said controlling means incorporates the same request datagram identification code in said reply
20 authorizing datagram.

98. A system as in claim 93, wherein said controlling means compares said facility address of said licensee with a list of valid licensee facility addresses and does not generate a reply authorizing
25 datagram if said facility address of said licensee is not valid.

- 50 -

99. A system as in claim 98, wherein said controlling means sends a reply denial datagram when said facility address is not valid.

5 100. A system as in claim 93, wherein said controlling means compares an expiration date of a license of said product with a date at which said request datagram is received by said controlling means, and does not generate a reply authorizing datagram, thus denying use of said product, if the
10 license expiration date is earlier than the date at which said request datagram is received.

101. A system as in claim 100, wherein said controlling means sends a reply denial datagram if the license expiration date is earlier than the date at
15 which said request datagram is received.

102. A system as in claim 93, wherein said controlling means generate a reply authorizing datagram, thus denying use of said product, if a payment for the use of said product is overdue.

20 103. A system as in claim 102, wherein said controlling means sends a reply denial datagram if payment for the use of said product is overdue.

104. A system as in claim 93, wherein:
said datagrams include data indicative of
25 the number of processes, at a licensee's site, currently using said product; and
said controlling means does not generate a reply authorizing datagram, thus denying a use of said

- 51 -

product, if more than a predetermined number of processes using said product are running at the licensee's site.

5 105. A system as in claim 104, wherein said controlling means sends a reply denial datagram if more than said predetermined number of processes using said product are running at the licensee's site.

10 106. A system as in claim 93, wherein said controlling means sends a reply denial datagram if use of said product is not approved.

107. A system as in claim 93, wherein:
said licensed product comprises an executable portion and a data portion; and
said system further comprises means for
15 controlling use of said data portion with said executable portion.

108. A system as in claim 107, wherein said data portion controlling means is disposed within said executable portion.

20 109. A system for monitoring a licensed product comprising:

a communications facility to which at least one licensee and a licensor are connected at a licensee's site and at a licensor's site,
25 respectively; and

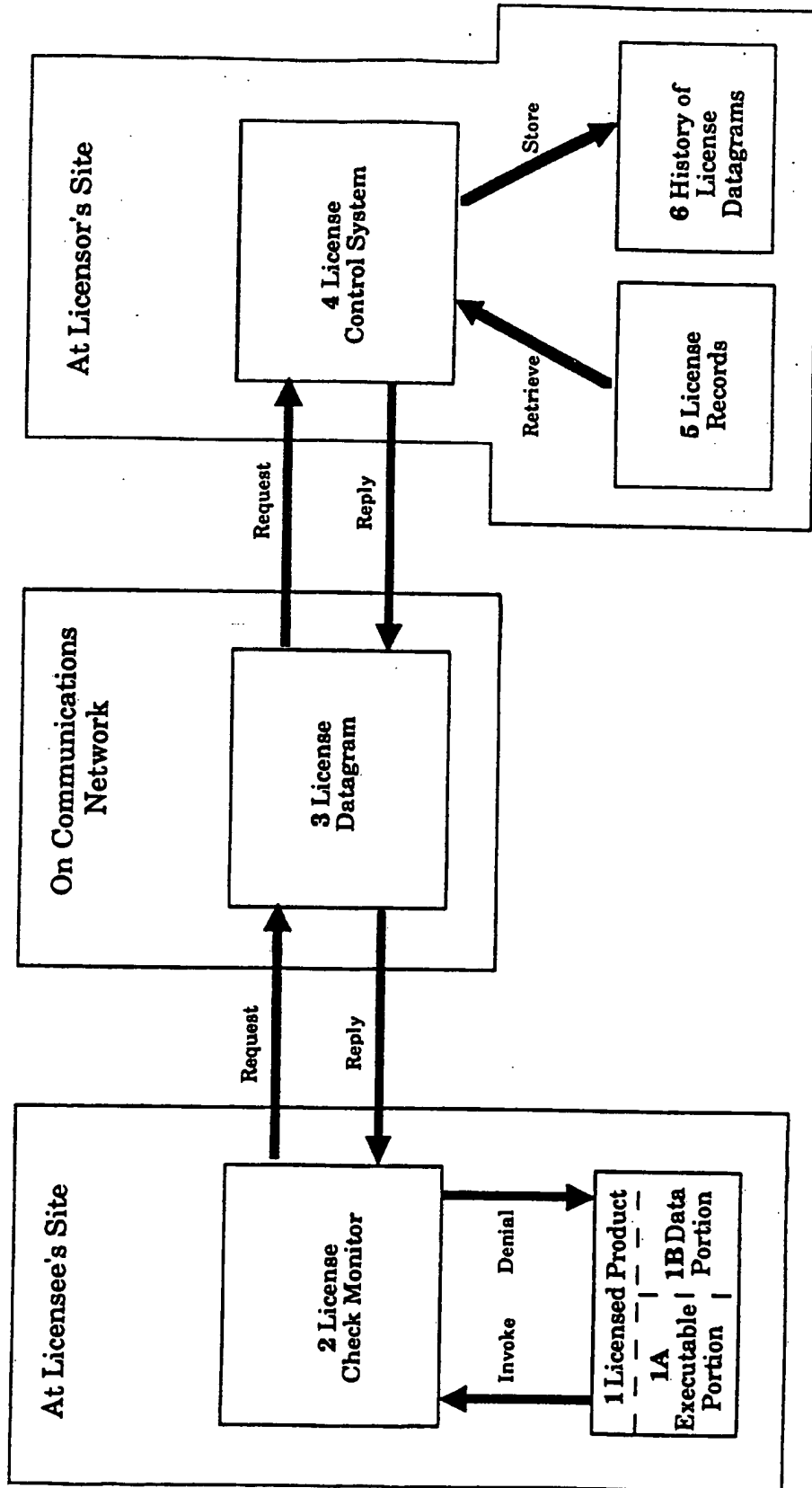
control means, connected to said communications facility at a licensor's site, for: receiving request datagrams, said request datagrams

including an address of said licensee on said communications facility and being transmitted at periodic intervals over said communications facility to said licensor's site; storing an indication of receipt of each of said datagrams; and counting said datagrams from each licensee as an indication of the use by the licensee of said licensed product.

110. A system as in claim 110, wherein:
said request datagrams include a product model number; and
said controlling means separately counts request datagrams for each product model number for each licensee.

1 / 8

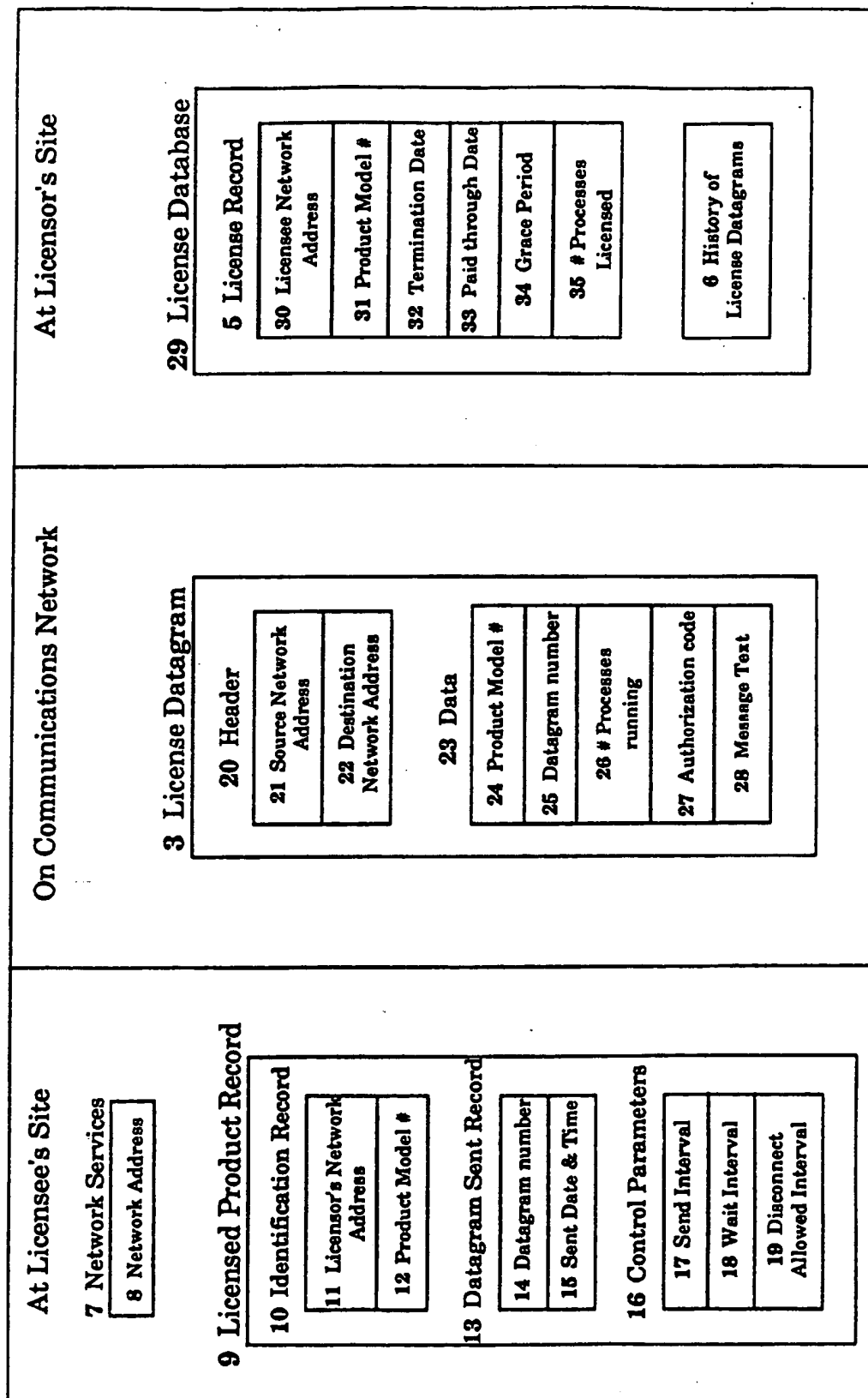
FIG. 1



SUBSTITUTE SHEET

2 / 8

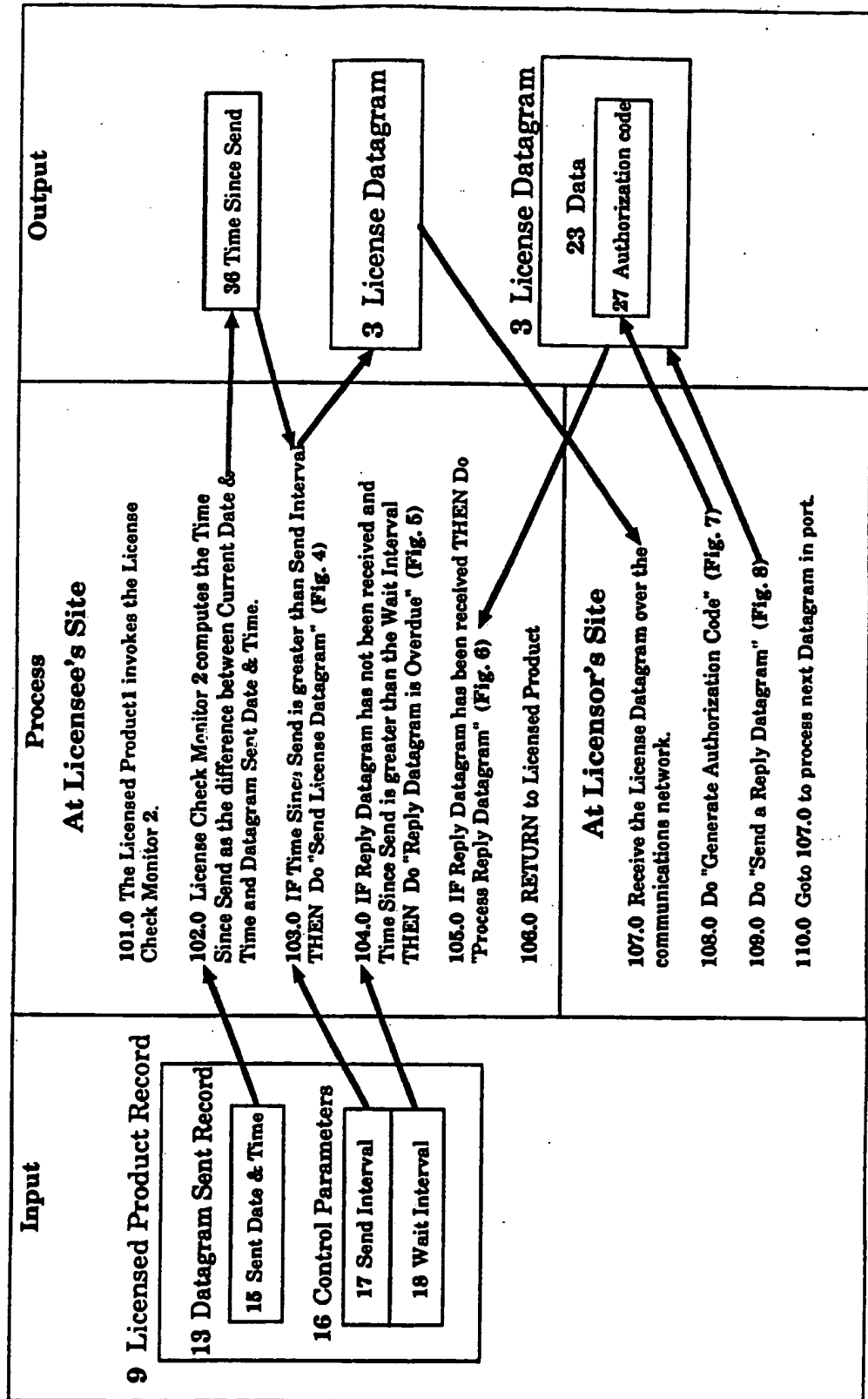
FIG. 2



SUBSTITUTE SHEET

33 / 8

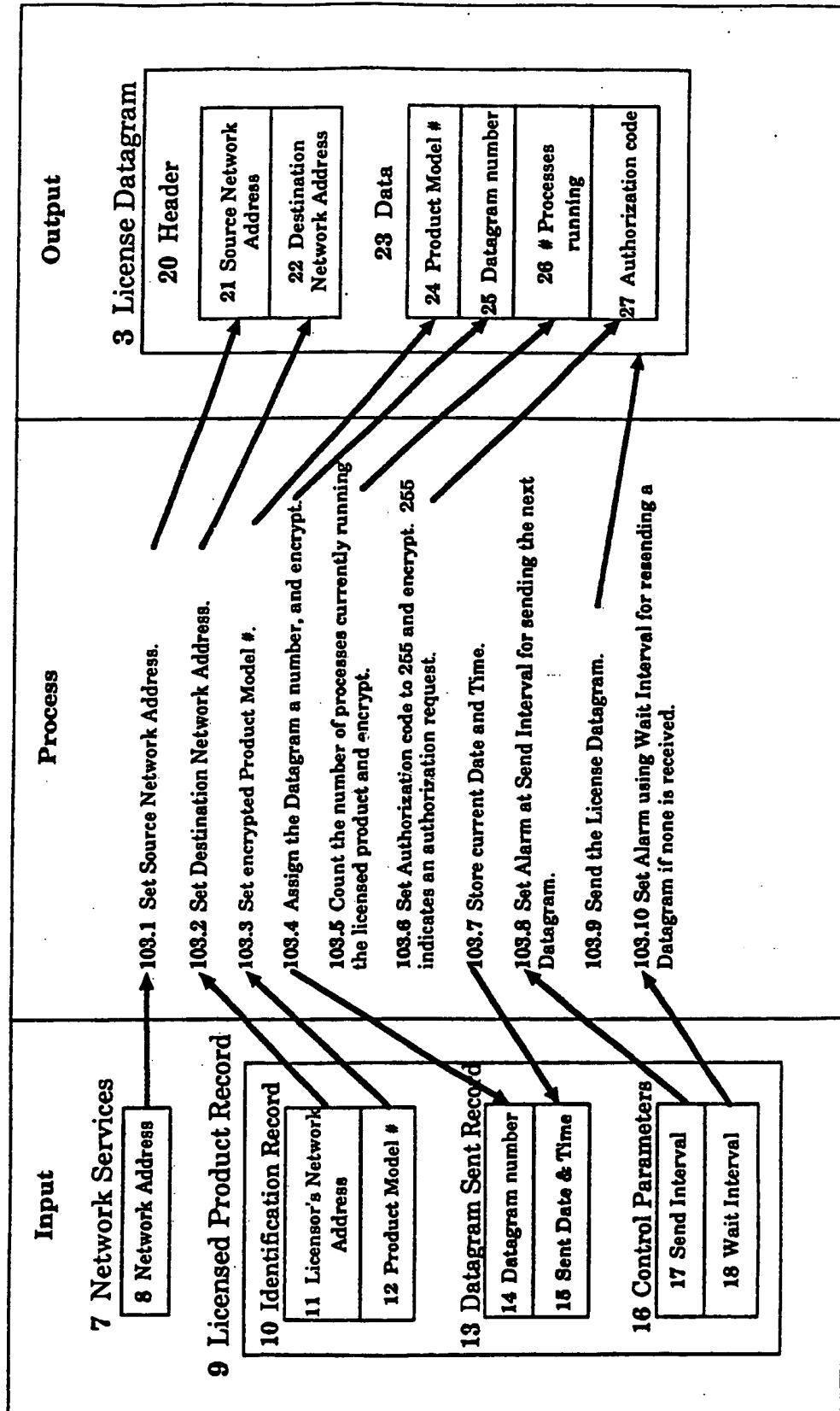
FIG. 3



SUBSTITUTE SHEET

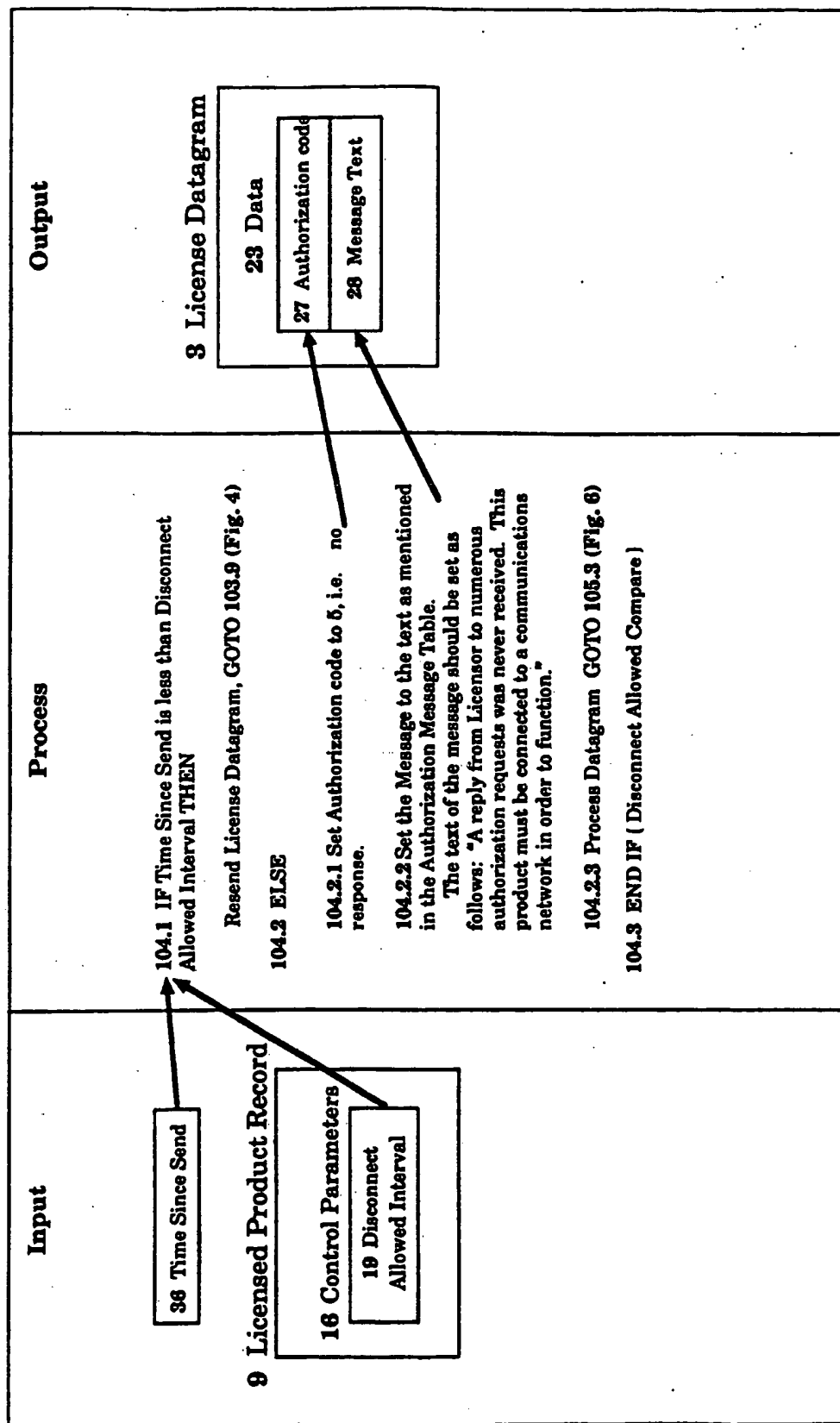
4 / 8

FIG. 4



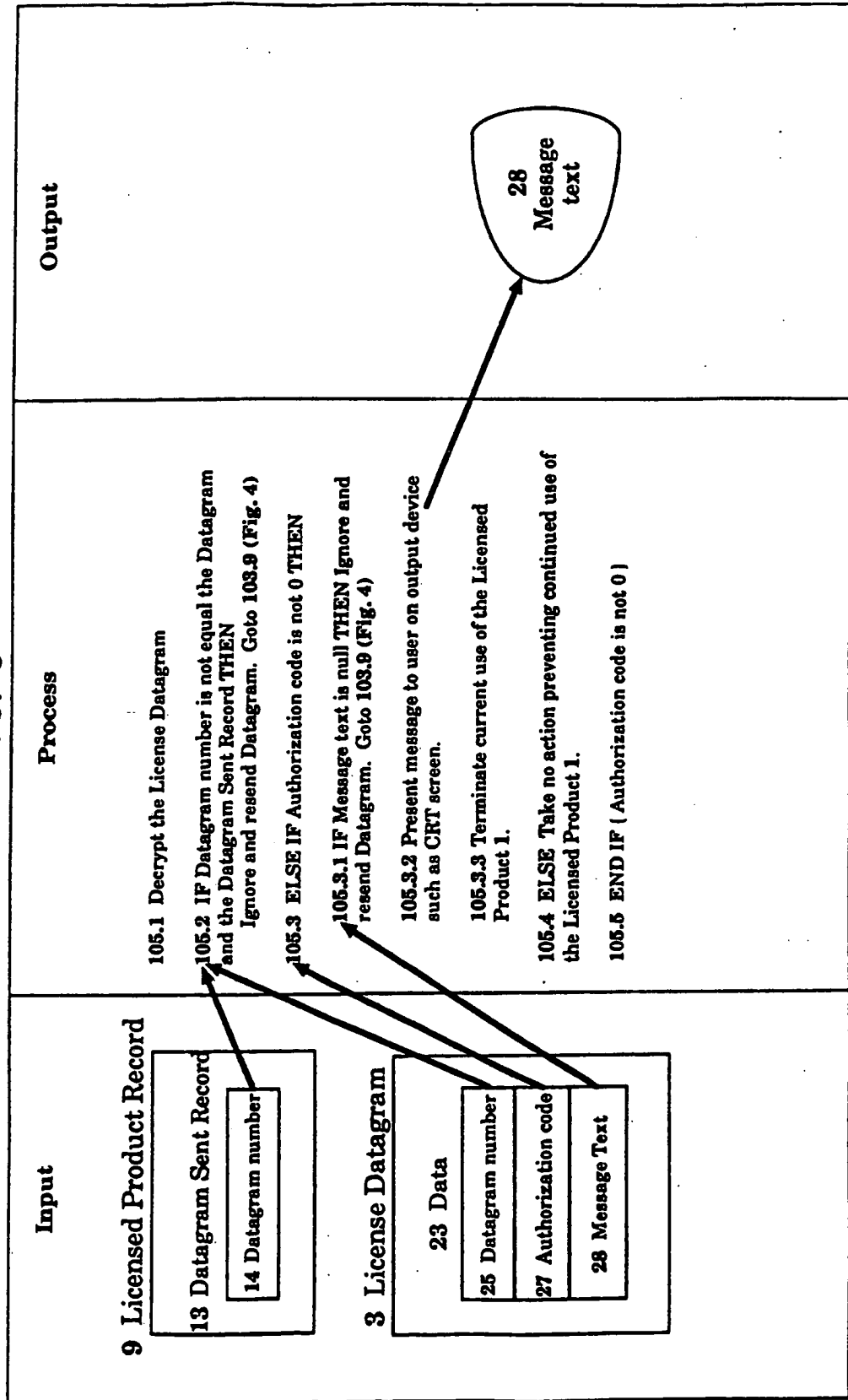
SUBSTITUTE SHEET

FIG. 5



6 / 8

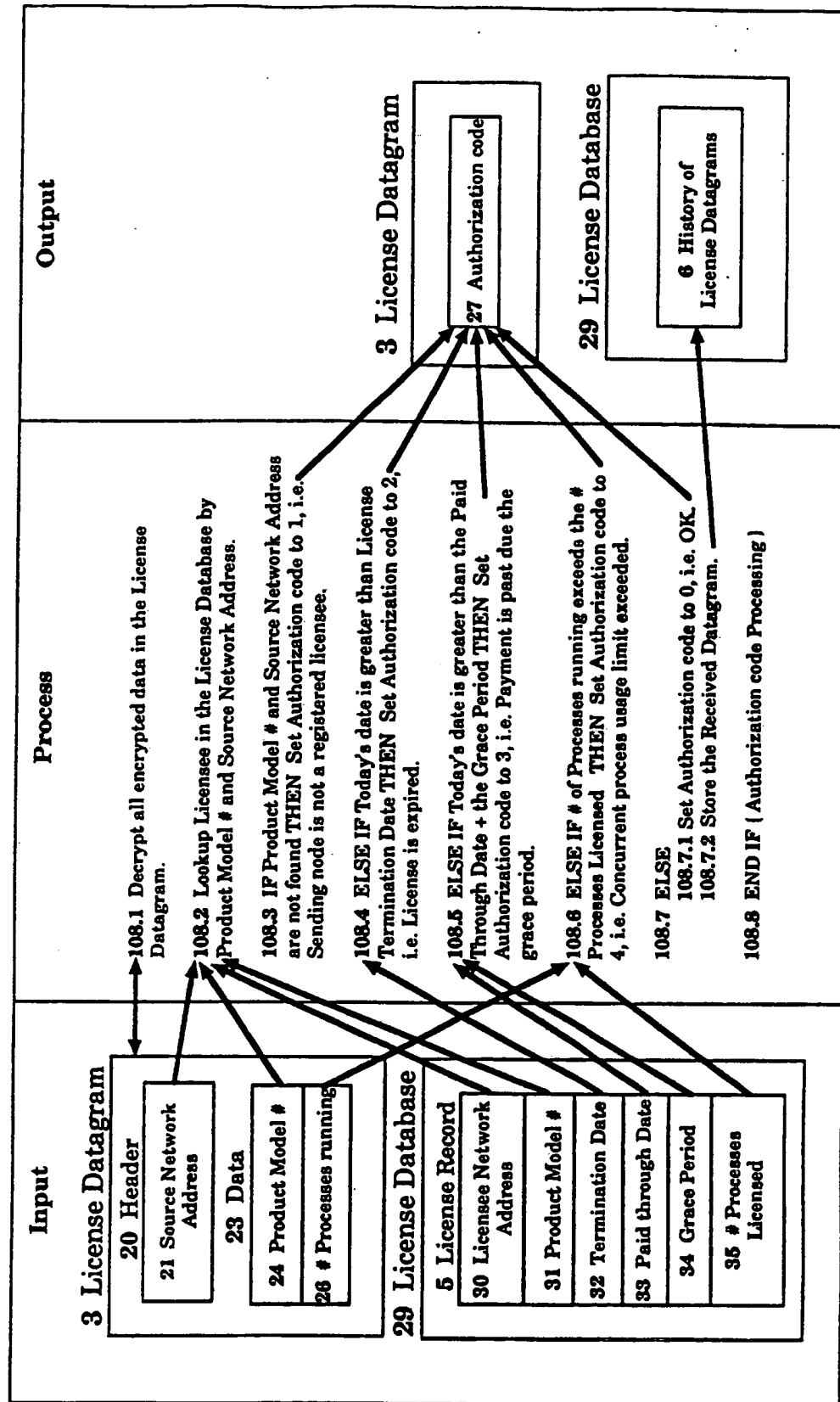
FIG. 6



SUBSTITUTE SHEET

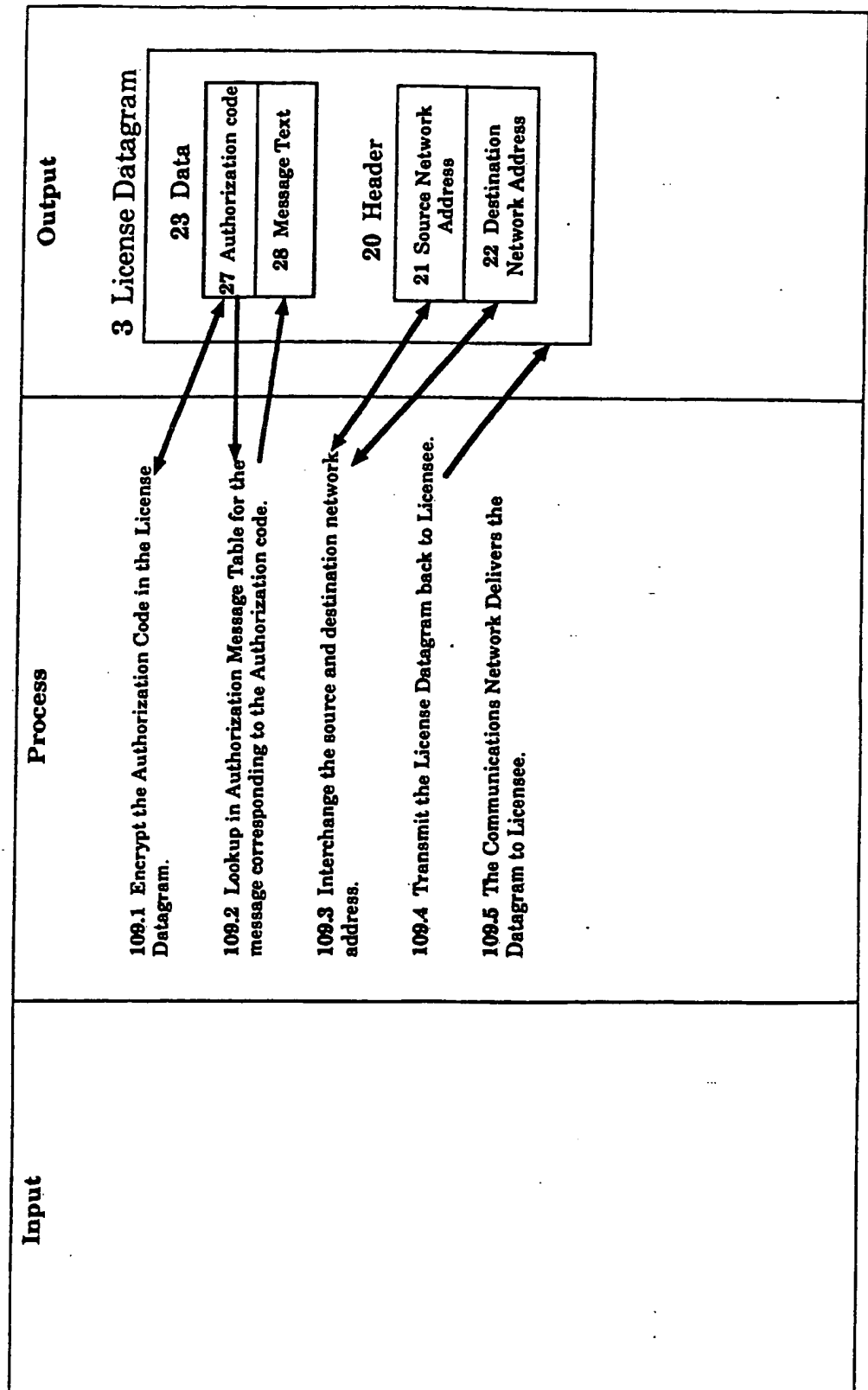
7 / 8

FIG. 7



SUBSTITUTE SHEET

FIG. 8



SUBSTITUTE SHEET

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US92/05387

A. CLASSIFICATION OF SUBJECT MATTER

IPC(5) :G06F 11/34; H04L 9/00

US CL :395/725; 380/4

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 364/406; 380/25

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS DATABASE: Software#, information, usage, monitor?, Licens?

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y,P	US,A, 5,103,476 (WAITE ET AL) 07 APRIL 1992 See entire text.	1-110
Y,P	US,A, 5,050,213 (SHEAR) 17 SEPTEMBER 1991 See column 6, lines 27-51.	1-6,9-21,23-26,29-43,45-53,56-59,61-64,67-73,75-110
Y,P	US,A, 5,047,928 (WIEDEMER) 10 SEPTEMBER 1991 See col. 6, lines 16-54.	1-6,9-21,23-36,29-43,45-53,56-59,61-64,67-73,75-110
Y	US,A, 5,023,907 (JOHNSON ET AL) 11 JUNE 1991 See entire document.	1-110

☒ Further documents are listed in the continuation of Box C.
 ☐ See patent family annex.

* Special categories of cited documents:	T	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
A document defining the general state of the art which is not considered to be part of particular relevance	X	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
E earlier document published on or after the international filing date	Y	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	A*	document member of the same patent family
O document referring to an oral disclosure, use, exhibition or other means		
P document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 05 AUGUST 1992	Date of mailing of the international search report 04 NOV 1992
Name and mailing address of the ISA/ Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	Authorized officer <i>Jeffrey Williams</i> KENNETH S. KIM
Facsimile No. NOT APPLICABLE	Telephone No. (703) 308-1634

Form PCT/ISA/210 (second sheet)(July 1992)*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US92/05387

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US,A, 5,014,234 (EDWARDS, JR.) 07 MAY 1991 See col. 3, lines 4-16.	1-110
Y	US,A, 5,010,571 (KATZNELSON) 23 APRIL 1991 See entire document.	1-6,9-21,23-26,29- 43,45-53,56-59,61- 64,67-73,75-110.
Y	MACMILLAN Publishing Company, 1985, WILLIAM STALINGS, Data and Computer Communications. p199-203.	1-110
Y,P	US,A, 5,113,519 (JOHNSON ET AL) 12 MAY 1992 See col. 6, lines 36-68.	1-110
Y	US,A, 4,937,863 (ROBERT ET AL) 26 JUNE 1990 See col. 3, lines 25-40.	1-6,9-21,23-26,29- 43,45-53,56-59,61- 64,67-73,75-110

Form PCT/ISA/210 (continuation of second sheet)(July 1992)*

BITS: *A Smartcard Protected Operating System*

Secure processors as a technology offer a great deal for applications that require enhanced computer security. In general, these processors contain not only computational capability, but memory capacity as well [17]. This self-containment makes secure processors resistant to attack, as they need not depend upon potentially vulnerable external resources.

S

ecure processors may appear in many forms. For instance, a dedicated host computer acting as a database server may be considered a form of secure processing. Another example is an add-in board with hardware for performing cryptographic operations. As the cost of computing resources continues to decrease and networks continue to proliferate, it is conceivable to commit secure hosts as servers to individual applications. The continued downsizing of computing resources may also make it possible for users to physically carry these servers.

Intelligent tokens are a class of pocket-sized secure processors that consist of an integrated circuit (IC) mounted upon a transport medium, such as plastic. They may also include downsized peripherals necessary for the token's application. Examples of such peripherals are keypads, displays and biometric devices (e.g., thumbprint scanners). The portability of these tokens lends itself to a myriad of security-sensitive applications, some of which are described later.

A subclass of intelligent tokens are IC cards known as smartcards. The physical characteristics of smartcards are specified by The International Standards Organization (ISO) [8]. In brief, the standard defines a smartcard as a credit-card-sized piece of flexible plastic with an IC embedded in the upper left-hand side. Communication with the smartcard is accomplished through contacts that overlay the IC [9]. Further, ISO also defines multiple communication protocols for issuing commands to a smartcard [10]. All references to smartcards in this article refer to ISO standard smartcards. However, the concepts and applications are valid for intelligent tokens in general.

The capability of a smartcard is defined by its IC. As the name implies, an integrated circuit consists of multiple components combined within a single chip. Some possible components are a microprocessor, nonstatic random access memory (RAM), read only memory (ROM), erasable programmable read only memory (EPROM), other nonvolatile memory (memory that retains its state when current is removed) such as electrically erasable programmable read only memory (EEPROM), and special purpose coprocessor(s) (see Figure 1). The chip designer selects the components as needed and designs the chip mask. The chip mask is burned onto the substrate material,

filled with a conductive material, and sealed with contacts protruding.

The current substrate of choice is silicon. Unfortunately, silicon, like glass, is not particularly flexible; thus to avoid breakage when the smartcard is bent, the IC is limited to only a few millimeters on a side. The size of the chip correspondingly limits the memory and processing resources that may be placed on it. For example, EEPROM occupies twice the space of ROM, while RAM requires twice the space of EEPROM. Another factor is the mortality of the EEPROM used for data storage, which is generally rated for 10,000 write cycles and deemed unreliable after 100,000 write cycles.

Several chip vendors provide ICs for use in smartcards. In general, these vendors have adapted 8-bit microcontrollers with clock rates of approximately 4 MHz for use in smartcards. However, higher performance chips are under development. Hitachi's H8/310 is representative of the capabilities of today's basic smartcard chips. It provides 256 bytes of RAM, 10 KB of ROM, and 8 KB of EEPROM. The successor to the H8/310, not yet released, claims a 16-bit, 10-MHz processor and twice the memory of the H8/310. It is likely that the other vendors have similar chips in various stages of development.

Because of these and other limits imposed by current technology, smartcards (and smart-tokens in general) are often built to application-specific standards. For example, while

BEST AVAILABLE COPY

there is increased security in incorporating peripherals with the token, the resulting expense and dimensions of self-contained tokens is often prohibitive. Because of the downsizing required for token-based peripherals, usability issues are also involved.

From a practical perspective, peripherals may be externally provided as long as there is reasonable assurance of the integrity of the hardware and software interface provided. The thickness and bend requirements for smartcards do not currently allow the incorporation of such peripherals, nor is it currently feasible to provide a constant power supply. Thus current smartcards must depend upon externally provided peripherals to supply user input as well as time and date information, and a means to display output.

Even if such devices existed for smartcards, it is likely that cost would prohibit their use. For most applications it is more cost-effective to provide a single set of relatively high cost input/output (I/O) devices for multiple cards than to increase smartcard cost by orders of magnitude. This approach has the added benefit of encouraging the proliferation of cardholders.

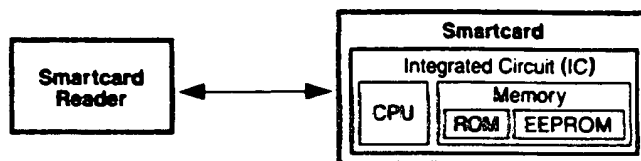
Smartcard Applications in Computer Security

Despite the limitations of existing smartcard chips, smartcards are more than adequate for a variety of applications in computer security. The National Institute of Standards and Technology (NIST) has developed the Advanced Secure Access Control System (ASACS), which provides both symmetric (secret key) and asymmetric (public key) cryptographic algorithms on a smartcard [6]. The ASACS utilizes the Data Encryption Standard (DES) for log-in authentication using the ANSI X9.26 standard authentication protocol [1]. It further offers a choice of RSA [16] or DSA [15] for digital signatures.

The ASACS card provides strong security because all secret information is utilized solely within the confines of the card. It is never necessary for a secret or private key to be transferred; all cryptographic operations are performed in their entirety on the

card. Although the current H8/310 equipped card requires up to 15 seconds to perform, sign, or verify operations, a new card being developed by Gemplus International is capable of performing the same operations in less than 0.1 seconds.¹ The Gemplus card is equipped with an SGS Thompson ST16C853 IC, which has a hardware modular exponentiator (coprocessor), 352 bytes of RAM, and 2.5 KB of EEPROM for data storage. Since both the RSA and DSA algorithms are based on modular exponentiation, it is the coprocessor that accounts for the Gemplus card's greatly enhanced performance.

Another application suitable for smartcard technology is Privacy Enhanced Mail (PEM) [2,11-13], which



provides authentication, confidentiality, and integrity services for electronic mail. Implementations are currently available internationally to the Internet community. Although currently utilized to provide end-to-end cryptographic services for electronic mail, the technology used in PEM has potential for a variety of uses requiring digital signatures and encryption.

Some implementers are currently integrating smartcards for use in sending privacy enhanced mail. A user-supplied smartcard stores the user's private key and in addition provides service calls for digital signatures and encryption so that all operations involving the private key are performed on the card. In this way the private key need never leave the card and risk compromise. Thus PEM users can sit down at any terminal that has access to the application software (and a smartcard acceptor) and read encrypted mail and send signed messages without fear of compromising their private keys.

Current smartcards, even with

¹Personal conversation with Gilles Lisimaque, Vice President, Gemplus International.

their limited capabilities, lend themselves to a variety of applications that extend beyond cryptography. The secure computing characteristics of smartcards make them ideal for access control and enhanced computer security applications.

Operating System Integrity

Since its introduction in the early 1980s, the personal computer (PC) has come to enjoy a popularity and acceptance its designers could never have foreseen. In a world that was dominated by the multiuser mainframe, the PC represented the first step in a revolution that has yet to reach its conclusion. Unfortunately, the single-user nature of the PC left little incentive for designers (hard-

Figure 1. Smartcard architecture

ware and software) to introduce security mechanisms. One result, over a decade later, has been the proliferation of computer viruses [5]—by some accounts almost 4,000 as of March 1994 [3].

A large number of PC viruses are of a class known as *boot infectors*. These viruses infect the boot sector of either a floppy or hard disk so that when the computer is subsequently powered on, and the boot sequence is initiated, the virus code is loaded into memory [7]. Because the boot sequence precedes the execution of application programs, and even the operating system itself, antiviral software is ineffective at preventing the execution of a boot-sector virus. Thus for software defenses to be effective it is vital that a "clean boot" be assured.

Access Control

To prevent deliberate sabotage as well as inadvertent catastrophe, some means of access control is desirable in a shared computer or network envi-

ronment. A typical solution to the access-control problem is to initiate a log-in sequence. This password-based scheme relies upon "something the user knows," while a token-based system uses "something the user has," and a biometric authentication (e.g., thumbprint scan) requires "something the user is." Generally, the more of these mechanisms that are used, the stronger the level of authentication.

A Smartcard Protected Operating System

The Boot Integrity Token System (BITS) is being developed to provide computer boot integrity and enforce access control [4]. A prototype exists for an IBM or compatible system (PC-BITS), although the technology is applicable to a variety of other computer systems.

The basic idea behind BITS is that the host computer will actually boot from a smartcard. Since the smartcard can be readily configured to require user authentication prior to data access, it provides an ideal mechanism to secure a host computer. That is, if critical information required to complete the boot sequence is retrieved from a smartcard, boot integrity may be reasonably assured. The security of the system assumes the physical security of the host with either a tamper-proof or tamper-evident casing, and the security of the smartcard by its design and configuration. If an attacker can gain physical access to the hardware, it is impossible to guarantee system integrity.

The PC-BITS prototype consists of an 8-bit add-in board, a smartcard acceptor that mounts in a floppy disk bay, configuration as well as file validation software, and a supply of smartcards. The board contains a special boot PROM, which is loaded with a program that interfaces to the smartcard acceptor. Further, the board is configurable to set an identifier for the host. Installation and configuration of the host can be accomplished in minutes. The process involves insertion of an add-in board and the equivalent of a floppy drive installation. Once installed, the computer will not complete the boot sequence without a valid user authentication

to a properly configured smartcard.

During system start-up, two authentications must be successfully performed to complete the boot sequence. First, the user is authenticated to the smartcard by means of the user password. If successful, the smartcard allows the PC to read the boot sector and other information from the smartcard memory. To authenticate the smartcard to the host, the card must also make available a secret shared with the host, in this case the configurable host identifier. Table 1 illustrates these transactions. If both the user and card authentication are successful, the boot sequence completes, and control is given to the PC operating system—which has been retrieved from or verified with the smartcard. The user may then proceed to utilize the PC in the usual fashion, uploading additional information (i.e., applications or application integrity information) from the smartcard as needed.

The card is expected to contain critical data such as checksums for system executables and the user's cryptographic keys. Comparing executable checksums with those stored on the smartcard provides a virus-detection mechanism that is difficult to defeat. This approach is consistent with a recent trend to validate file integrity rather than scan for known virus signatures.

In the BITS system, cards are configured and issued by a security officer using the card management software provided—the current prototype is written in C to improve portability.

Configuration entails the loading onto the smartcard of the host's boot sector as well as checksums for boot files stored on the host. At the time of issue, it is necessary to specify the machine or set of machines the user to whom the card is being issued will be granted access, so that a host key may be loaded. File integrity information is also loaded onto the smartcard at this time. All data is read-protected by the user's authentication (e.g., cannot be read unless the user password is presented correctly), and write-protected by the security-officer authentication. This arrange-

ment (depicted in Table 2) prevents users from inadvertently or deliberately corrupting critical data on the smartcard.

A concern is the ability of malicious users to create counterfeit tokens to access the host and potentially change the operating system it is intended to boot. Because the host will disallow access to users without a properly configured token as well as the password to access the token, only authorized users have the potential to create a counterfeit token. By definition, these users have access to the host; therefore, the only additional threat posed by this class of user is the booting of an unintended operating system. In the case where the intended operating system is a multilevel secure (or other multiuser system), a malicious authorized user may wish to boot an untrusted operating system to allow access to data that would be unavailable under the intended operating system.

To effectively defend against the booting of an unintended operating system, all boot relevant information must be stored on the host, with only checksums for this information stored on the token. The host boot program, modified for BITS, can then ensure that the proper operating system is loaded from the host—after being integrity checked with the values on the token.

Smartcards may be issued on a per host, per group, or per site basis, depending on the level of security desired. Because the secret shared by the host and card is configurable on the host, it is possible to issue smartcards in a one-to-one, many-to-one, or many-to-many fashion. A one-to-one mapping of users to hosts corresponds to securing a machine for a single user. Analogously, many-to-one allows the sharing of a single machine, and many-to-many allows for the sharing of multiple machines among an explicit set of users. One-to-many is a possible, but usually wasteful, mapping of computer resources.

Although accurate timing information is not yet available, it is not expected that BITS will degrade overall system performance. BITS is primarily intended for system start-up,

which is an inherently time-consuming process. It is expected that users may be willing to accept some initial performance degradation in return for greatly enhanced security. Further, after completing a clean boot, BITS in no way impedes system usability.

During system start-up, BITS will degrade performance in two ways. First, the I/O bandwidth of the smartcard and its interface are expected to be inferior to a modern hard drive. Thus for the information stored on the card the transfer speed will be slower. For the current prototype, less than 1 KB of information is stored on the card, which at 9,600 baud requires less than one second to upload. The second area of degradation occurs in the verification of the operating-system files. The severity of this degradation depends upon the processing capacity of the host computer, together with size of the operating system files to be verified, and the complexity of the hash and/or signature algorithm that is chosen.

The effectiveness of BITS is limited by the feasibility of storing of all boot-relevant information on a smartcard. To the extent this is possible, boot integrity will be preserved. BITS is not a virus checker; however, for those files whose checksums are stored on the smartcard, it is possible to detect the modification of the file on the host system. The user may be notified that an executable is suspect before it is run. In general, BITS will provide enhanced computer security by utilizing the secure storage and processing capabilities inherent in the smartcard.

BITS vs Other Technologies

Here we compare BITS with other boot integrity and access control solutions. In the process of examining the advantages and disadvantages of the BITS system, many frequently asked questions are addressed.

Because of the mode in which BITS is used, it is often compared with a boot from a floppy disk. While it is true that inserting a smartcard is similar to inserting a floppy, the interaction during the boot sequence is entirely different. The smartcard-based system incorporates two sepa-

Table 1. PC-BITS system startup

Step	Action	Implementation
0	Insert card, and power up the host	Apply power, and reset card
1	Authenticate user, and present data to the smartcard	Host requests the user password and presents it to the smartcard
2	Authenticate the card to the host	Host reads shared secret from the smartcard
3	Upload boot information	Host reads boot sector from the smartcard
4	Integrity check host-resident boot files, and complete boot sequence if successful	Host computes file-checksum; this value is compared with the checksum value stored on the card

Table 2. BITS smartcard configuration

Step	Action	Implementation
0	Security officer creates user and security officer accounts on card	Present manufacturer password and load user-specified secret codes for accounts
1	Load boot sector onto card	Create a file readable under the user password and writable under the security officer password, and write the partition boot record
2	Compute and load checksums for selected files	For each file compute a hash; this checksum together with the file name is stored on the card
3	Load host authentication information	Create a file readable under the user password and writable under the security officer password, and write a secret to be shared with the host

rate authentications, user-to-card and card-to-host, which are entirely absent from the floppy boot. Further, the integrity of the boot information on a floppy is protected by an easily removed write-protect-tab, while the smartcard requires the authentication of the security officer in order to update boot information. One may also wish to consider the ease of carrying a smartcard as compared with a floppy disk.

Another solution to the boot-integrity problem is to place the operating system in ROM. This approach will prevent modifications to boot information, but at the cost of updatability. Any upgrades to the system require

physical access to the hardware and replacement of ROM chips. If user authentication is added to the boot program, passwords may be difficult to change, and exist on a per-machine rather than a per-user basis.

If the user wishes to rely upon antiviral software, new issues are raised. The host machine must complete its boot sequence prior to executing the antiviral program. This means that a boot-infector virus will already have been executed by the time the defenses are in place. This situation may be analogized to locking the door to your house after the burglar is inside.

Log-in programs, whether stand-alone or integrated with an antiviral

program, suffer from the same timing issues as previously mentioned. Also, since most PCs provide a means of booting from floppy, log-in programs can often be trivially defeated.

While it is clear that software defenses are subvertible, hardware-based systems like BITS are not without their drawbacks. Installation of BITS requires the installation of an add-in board and smartcard acceptor. Also, individual cards must be configured. The time for setup, particularly for a first-time user, is certain to be non-nominal. The enhanced security provided by BITS requires physical access to the hardware to circumvent. While this situation is consistent with the BITS system objectives, it does imply a severe penalty to the user for a lost smartcard or forgotten password. Suitable administrative procedures, such as configuring multiple cards, become important to minimize the risk of a lost or damaged card.

The BITS system provides extremely powerful security at relatively low cost, measured both in terms of purchase price and setup time. The additional hardware required is nominal, initial setup is one time only, and upgrades require no hardware access—provided the user has the proper authentication. As with most computer systems, there are trade-offs between security and usability. It is ultimately incumbent upon the system administrators to evaluate their own application and weigh the costs.

The Future

Systems like the BITS represent the next step in a trend toward smaller personal-computing resources. From a security perspective, the less that a user depends upon from a shared environment, the better. Any shared writable executable may potentially contain malicious code. Fortunately, advances in technology are likely to permit the storage of entire operating systems as well as utilities on a smartcard, thus obviating the necessity of sharing most executables altogether.

The smartcards themselves may also be made more secure. Currently, authentication to the smartcard is limited to user-supplied passwords.

In most systems, three consecutive false presentations result in the smartcard account being disabled. However, if biometric authentication is incorporated into the card, it will be possible to achieve higher assurance in user authentication.

To date, the size requirements of smartcards have imposed the greatest limitation upon their utility; the current state of the art is a 0.7-micron resolution in the burning of chip masks. However, chip developers have recently announced the development of 0.5-micron technology as well as plans for even higher resolutions. Regardless of these advances, the chips are still currently limited to a few millimeters on a side due to the brittle nature of the silicon substrate from which they are made. A flexible substrate might allow chips that occupy the entire surface of the smartcard, resulting in an exponential gain in computing resources.

A smartcard with these enhanced capabilities could result in a truly portable (wallet-sized) personal computer, which could be made widely available at relatively low cost. In this type of computing environment only the bulky human interface need be shared. A computing station might eventually consist of a monitor, a keyboard, a network connection, and a smartcard interface. A user could walk up, supply the CPU and memory, and begin work. It would be as if you were able to supply your own engine, with a known reliability and maintenance history, for a rental car.

The implications of this technology are impressive. The existence of instant PC access for millions regardless of location would greatly enhance the utility of computers. The ability to use the same environment wherever one chooses to work would eliminate time spent customizing, and increase productivity. The security provided by smartcards would also result in increased security for sensitive data by decreasing the likelihood of compromise or loss. ■

References

1. American Bankers Association. *Financial institution sign-on authentication for wholesale financial systems (DES-based user authentication protocols)*. ANSI

- X9.26, X9 Secretariat, Washington, D.C.
2. Balenson, D.M. Privacy enhancement for Internet electronic mail. Algorithms, modes, and identifiers. Internet RFC 1423 (Part 3).
3. Brunnstem, K. *CFBASE—the computer virus catalog*. University of Hamburg, Germany.
4. Clark, P.C. *System For Protecting Computers Via Intelligent Tokens Or Smartcards* (Patent pending 07 841814).
5. Denning, P.J. *Computers Under Attack: Intruders, Worms and Viruses*. Addison-Wesley, New York, 1990.
6. Drav, J., and Balenson, D. *An overview of the advanced smartcard access control system*. Computer Security Division Computer Systems Laboratory, National Institute of Standards and Technology, Gaithersburg, Md.
7. Feudo, C.V. *The Computer Virus Desk Reference*. Business One Irwin, pp 411f.
8. International Standards Organization. *Identification cards—integrated circuits with contacts—physical characteristics*. International Standard 7816-1, 1987.
9. International Standards Organization. *Identification cards—integrated circuits with contacts—dimensions and location of the contacts*. International Standard 7816-2, 1988.
10. International Standards Organization. *Identification cards—integrated circuits with contacts—electronic signals and transmission protocols*. International Standard 7816-3, 1989.
11. Kaliski, B.S. Privacy enhancement for Internet electronic mail. *Key certification and related services*. Internet RFC 1424 (Part 4).
12. Kent, S. Privacy enhancement for Internet electronic mail. *Certificate based key management*. Internet RFC 1422 (Part 2).
13. Linn, J. Privacy enhancement for Internet electronic mail. *Message encipherment and authentication procedures*. Internet RFC 1421 (Part 1).
14. National Institute of Standards and Technology (formerly NBS). *Data encryption standard*. FIPS publication 46-1, Gaithersburg, Md.
15. National Institute of Standards and Technology. *The digital signature standard proposed by NIST*. *Commun. ACM* 35 (July 1992), 36–40.
16. Rivest, R.L., Shamir, A., and Adleman, L.M. A method for obtaining digital signatures and public key cryp-

CONTINUED ON PAGE 94

were to start all over again, knowing what we know today, what would we do differently? Clearly, my historical review shows that today's most popular approaches are refinements of concepts initiated in the 1970s. (Redwine and Riddle suggest that technology transfer takes 17 ± 3 years.) Yet I believe that our present orientation has been biased by approaching software design as if it were analogous to hardware development. We do not take advantage of the special features of software. When we do exploit the unique properties of software, however, I believe it will be possible to build powerful, flexible Quadrant III environments. This is an important message to researchers.

The article also has some lessons for practitioners. Often what a method was designed to do differs from the way it is used (i.e., the abstraction differs from its application). By understanding the goals of a

method, one can better predict its efficacy in a concrete situation. For example, a method based on a top-down philosophy will probably be inappropriate for an open problem (which has no well-defined "top"). There also is a difference between a method—which is intended to establish a systematic process for achieving some end—and the tools it uses. The fact that a method's products are created is not evidence that the method itself has been used. Moreover, if the objective of a method was to overcome the burden of excessive textual descriptions, then using a CASE tool to support that method may do little more than avoid that textual documentation. Today we are blessed with a multitude of methods and supporting tools, and each organization can accommodate only a small subset of what is available. Perhaps this taxonomy will help practitioners identify the methods best suited to their particular settings. ■

CONTINUED FROM PAGE 70

tosystems. *Commun. ACM* 21, 2 (February 1978), 120–126.

17. Tygar, J.D., Yee, B. *DYAD: A system for using physically secure coprocessors*. Carnegie Mellon University, School of Computer Science, Dec. 1992.

About the Authors:

PAUL C. CLARK is a computer scientist with Trusted Information Systems, where he works on a variety of security-related projects. His specialties include cryptography, networks, graphics, and database systems. **Author's Present Address:** Trusted Information Systems, 3060 Washington Rd., Glenwood, MD 21738. paul@tis.com

LANCE J. HOFFMAN is professor of computer science and director of the Institute for Computer and Telecommunication Systems Policy at The George Washington University. **Author's Present Address:** Department of Electrical Engineering and Computer Science, The George Washington University, Washington, D.C. 20052. hoffman@seas.gwu.edu

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

OACM 0002-0782/94/01100 \$3.50

References

1. Balzer, R., Cheatham, T.E., Jr., and Green, C. Software technology in the 1990's: Using a new paradigm. *IEEE Comput.* 16, 11 (1983), 39–45.
2. Barstow, D.R. Domain-specific automatic programming. *IEEE Trans. Soft. Eng. SE-11* (1985), 1321–1336.
3. Barwise, J. Mathematical proofs of computer system correctness. *Notices Am. Math. Soc.* 36 (1989), 844–851.
4. Blum, B.I. *TEDUM and the Software Process*. MIT Press, Cambridge, Mass. 1990.
5. Blum, B.I. *Software Engineering: A Holistic View*. Oxford University Press, New York, 1992.
6. Blum, B.I. Representing open requirements with a fragment-based specification. *IEEE Trans. Sys., Man, and Cyber.* 23 (1993), 724–736.
7. Booch, G. Object-oriented development. *IEEE Trans. Soft. Eng. SE-12* (1986), 211–221.
8. Brooks, F.P., Jr. No silver bullet. *IEEE Comput.* 20, 4 (1987), 10–19.
9. Cameron, J.R. An overview of JSD. *IEEE Trans. Soft. Eng. SE-12* (1986), 222–240.
10. Cohen, B., Harwood, W.T., and Jackson, M.I. *The Specification of Complex Systems*. Addison-Wesley, Reading, Mass. 1986.
11. DeMarco, T. *Structured Analysis and System Specification*. Yourdon, New York, 1978.

12. Dijkstra, E.W. The structure of "THE" multiprogramming system. *Commun. ACM* 11 (1968), 341–346.
13. Liskov, B.H. and Zilles, S.N. Specification techniques for data abstraction. *IEEE Trans. Soft. Eng. SE-1* (1975), 7–19.
14. Meyer, B. *Object-Oriented Software Construction*. Prentice-Hall, New York, 1988.
15. Parnas, D.L. A technique for software module specification with examples. *Commun. ACM* 15 (1972), 330–336.
16. Parnas, D.L. On criteria to be used in decomposing systems into modules. *Commun. ACM* 15 (1972), 1053–1058.
17. Prieto-Diaz, R. and Arango, G. *Tutorial: Domain Analysis and Software Systems Modeling*. IEEE Computer Society Press, Silver Spring, Md. 1991.
18. Stevens, W., Myers, G., and Constantine, L. Structured design. *IBM Sys. J.* 13 (1974), 115–139.
19. Teichrow, D. and Hershey, E.A., III. PSL-PSA: A computer-aided technique for structured documentation and analysis of information processing systems. *IEEE Trans. Soft. Eng. SE-3* (1977), 41–48.
20. Warnier, J.D. *Logical Construction of Systems*. Van Nostrand, New York, 1981.
21. Zave, P. The operational versus the conventional approach to software development. *Commun. ACM* 27 (1984), 104–118.

About the Author:

BRUCE I. BLUM is currently principal professional staff at Johns Hopkins University Applied Physics Laboratory and associate professor of biomedical engineering at Johns Hopkins School of Medicine. His research interests include knowledge representation for the software process, software engineering, and medical informatics. **Author's Present Address:** Applied Physics Laboratory, Johns Hopkins University, Laurel, MD 20723-6099. bib@aplcomm.jhuapl.edu

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

OACM 0002-0782/94/01100 \$3.50

This work was supported in part by the U.S. Navy, Space and Naval Warfare Systems Command (SPAWAR) under contract N00039-91-C-0001, task VMAR9 with the Office of Naval Research (ONR) and the Naval Surface Warfare Center (NSWC).

Compact disks are an excellent medium for publishing data, but the fear of piracy keeps many databases on-line. Infosafe thinks it has the answer.

Data guard

By Philip E. Ross

NINETEENTH-CENTURY American publishers waited at the docks for early shipments so they could pirate the latest work of Sir Walter Scott and other British literary lions. Copyright conventions later solved the problem.

But now that words, pictures and music can be reduced to digital signals, copyright conventions are much harder to enforce. That's one reason publishers hesitate to put their works on compact disks. The disks are cheap to manufacture and distribute but easy to copy—much easier than copying a printed book. For a data service priced at, say, \$20,000, the temptation to make illegal copies is great.

There are two other drawbacks. There is no flat fee suitable for all users. A customer who will access the database once a week isn't going to pay \$20,000 a year for it. And a database published monthly on CDs is not up-to-the-minute.

The alternative to the compact disk is an on-line service offered via passwords with a fee for each use. This solves the piracy problem, the pricing problem and the freshness problem. But on-line services have a stumbling block of their own: communications. Long distance costs money and modems are slow.

Infosafe Systems, a New York City startup, says it has a solution to this dilemma, combining the best of the CD and the on-line system.

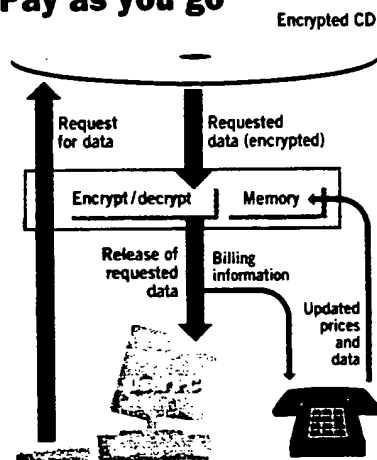
The company markets a box that database vendors send to their customers. Attached to the customer's computer, the box decodes data encrypted on a CD-ROM, protecting the publisher from piracy and allowing him to charge by the use if he wants to. The box also has up to 1 gigabyte of memory and a modem. That gives the publisher a chance to update the database—and perhaps reprise the in-

formation—between CD mailings. The updating goes on automatically, generally at night.

The Infosafe box releases valuable data in stages. Suppose, for example, that Infosafe's client is the owner of a photograph library, and that this client's customers are, in turn, advertising agencies. An ad designer might order up a low-resolution, digitized image of San Francisco at a price of \$5. After a few such selections, he

Andy Christoforides

Pay as you go



Infosafe's box decrypts a CD-ROM and supplements it with phoned-in updates.

would be in a position to decide whether to buy the high-resolution form for \$350. "That's our striptease tactic," says Thomas Lipscomb, 55, cofounder and chief executive.

By providing free searches, querying each expenditure and displaying the bill to the user, the Infosafe system prevents users from burning up company budgets (or from blaming anyone but themselves if they do).

Rather than sell its box, Infosafe leases it. The company takes a per-

centage of its client publisher's royalties, while leaving the potentially very flexible pricing structure entirely up to the client.

For example, a user looking for biographical information wouldn't need a whole set of, say, *Who's Who*; he could be billed for just a couple of individual bios. This sort of ordering can provide valuable marketing information for vendors of information. They can offer optional databases, where the client's prior purchases indicate a likely interest in them.

Suppose you don't want someone tracking your use of the database. A law firm, for example, might not want people to know which cases it was researching. "We can blind the thing so we don't know exactly what you bought—just the price class," says Lipscomb.

As a former publisher who headed the New York Times Book Co. in the late 1970s, Lipscomb knows what his potential clients want. He got into data security in 1988 by cofounding a firm now called Wave Systems Corp. Wave competes with Infosafe but uses a slightly different approach. It installs a decoder clip inside each of a database buyer's personal computers. Yet another antipiracy scheme is from Rainbow Technologies, which attaches a plug-in security device to the computer, giving access only to the authorized user.

Infosafe has six customers so far. Among them are the American Trucking Associations, which plans to use the system to sell training and federal regulatory information to its 40,000 members, and the International Typeface Corp. in New York, which licenses fonts to those honest enough to pay for them.

For International Typeface the Infosafe system offers a welcome relief from illegal copying. Once upon a time, typefaces were incorporated in typesetting machinery that was easy to police. Nowadays the characters are just blips on a 5.25 diskette. "Piracy has brought the typeface industry to its knees," says Charles Wilhelm, a spokesman for International Typeface. "You can punch a button, duplicate the information and transmit it to somebody else." His firm aims to start putting all its customers on an Infosafe system this fall.

BOOKS

While U Wait

BY PAUL HILTS, JOHN MUTTER
and SALLY TAYLOR

Here's one scenario for a bookstore of the not-so-distant future: in a small space, proprietors stock one copy of each book they can supply for the customer. Or stores might forgo sample copies altogether and have only a catalogue. When a customer wants a book, the bookseller asks the customer's preferences: What typeface and size, what color and grade of paper, plain or fancy binding? Once these and other matters are decided, the bookseller presses a few buttons on the in-store printer. Within minutes, the book, whose digitally encoded text is transferred via modem from the publisher or is stored in the machine after having been scanned earlier, is printed out and bound in the store. The finished book can look as handsome as anything currently available through traditional book manufacturing and distribution—and this includes books with color pictures and illustrations.

Seem far-fetched? A look at on-demand printing, which has been sweeping college campuses in '93, proves that the technology is nearly in place. As this movement continues into '94, it will raise some very interesting

Digital transmission
and on-demand
printing promise
to change the
bookseller's world
forever

questions for trade publishers, wholesalers, general booksellers and even writers.

Coursepacks—customized textbooks that are electronically assembled and printed at or near the campus—are now the fastest growing area in the college store world, according to *Campus Marketplace*, the National Association of College Stores' weekly newsletter. NACS's most recent sales data for the industry show that sales in October 1993 of "custom-published/locally produced materials" rose 148.1% over October 1992, the 10th month in a row that total sales dollars in this category have risen.

Many campus stores and nearby

copying shops are doing coursepacks: Stanford and Southern Cal are among the largest campus custom publishing operations. One example of what a store can do with the new technology is offered by the Cornell Campus Store at Cornell University, where sales of coursepacks grew 30% during the fall semester (3.5 million impressions in course packets from 80,000 different originals). Cornell is distinguished because it is one of the few doing coursepacks electronically. It is also doing them completely legally.

Cornell uses a Documents on Demand system manufactured by Xerox and prints on a Xerox DocuTech Network Publisher Model 90. The material is scanned and stored electronically. The system has been in use since fall 1992. "About three-quarters of the material is data used in previous semesters," explains Rod Perkins, Cornell's digital imaging specialist. "Each semester it is fine-tuned, and the reprints can save time in copyright clearance and electronic scanning. As professors become more aware of our services, more are using it."

Cornell's coursepacks generally include articles from newspapers and periodicals, sections of books and papers by faculty members. Heavy paper stock is used for covers and most of the jobs are three-hole punched and shrinkwrapped, although 40% are perfect-bound and 1% are plastic spiral-bound. In the bookstore, coursepacks are shelved with other titles required for a particular course. The Cornell Business School is the single-largest customer.

The store has been able to reduce waste to 7%. In cases where the store has printed too few copies, reprints can be made easily and quickly, drawing on the data stored in the system's optical disc "jukebox."

The Cornell Campus Store charges \$.05 a page for production services and a binding fee. Even though copy-

right fees can raise the cost of course-packs to the range of conventional textbooks, they are preferred by professors, according to Jim Lawrence, manager of electronics publishing.

Cornell is aboveboard about copyrights, says Lawrence: "All the material is analyzed for copyright needs and the data is entered into the system's management computer, which automatically generates letters for copyright approvals." The store uses the Copyright Clearance Center as much as possible. (CCC charges \$.0075 per page.) Otherwise, the store goes directly to publishers for copyright clearances, and the store usually pays whatever the publisher asks. However, "if the publisher wants 15¢ a page," Lawrence says, "the professor will usually opt to put the book on reserve at the library."

Broadening Markets

The market for instant books is now broadening to scholarly publishers. John Paeglow formed Integrated Book Technology with two partners three years ago to fulfill a new need for electronic printing.

"Publishers were constantly requesting lower quantities [for reprints]," Paeglow tells *PW*. "At these low quantities, it wouldn't pay to set up a conventional binding line. The only machine we found to be satisfactory for the quality we wanted was a Xerox DocuTech. Then Jim Ryan, one of my partners, came up with a binding solution incorporating library repair techniques. We've tested the binding and found it is stronger than Smyth-sewn."

Some of the most highly respected scholarly publishers—including Cambridge University Press, John Wiley and Marcel-Dekker—are now using IBT for reprints and very short run originals.

According to Cambridge University Press production director Richard

Hollick, "We'll reprint 30 copies of a book, just to have stock on hand. The unit cost is higher than printing 1000 copies conventionally, but by reducing the minimum run we can keep books in print indefinitely. In 1991, for example, we had orders for 24,000 units we could not fill because the books were out of print.

"While we have only done about 50 titles so far with IBT, the potential of never having a book out of print and never having overstock would make a tremendous impact on our total economic picture," Hollick says.

Bill Clockel at John Wiley & Sons reports a 50% increase in these short-run high-quality reprints since last year. The company is now doing 15 titles a month.

"One of the big applications of this will be global utilization," says Clockel. "Instead of sending books to Asia and Europe, we'll be able to send files. The DocuTech is good at reproducing exactly the same image every time from a file." IBT's Paeglow is also beginning to work with Xerox to develop other production sites around the country.

Brave New Bookstore

Which leads us back to the scenario of the on-demand general trade bookstore. While the technology will soon be available, tradition and customers' love of the ambiance of bookstores could inhibit the growth of such modernistic retailing.

Imagine what would happen if all bookstores converted to on-demand, in-store printing. Printing plants wouldn't be necessary. Wholesalers would have no books to stock and distribute. Publishers' warehouses would not be needed. Reps might still call on stores, but not so much to sell to stores as help stores sell. Publishers would still edit and market titles, but they might lose the tight control they have now over what the book looks like and

how they reach the ultimate customer.

Among the many advantages of this brave new book world: a sale would be a sale—returns would no longer exist; publishers wouldn't need to tie up money in inventory; paper would not be wasted; freight costs would vanish; a book would never have to go out of print or out of stock. In addition, the new technology could be a boon for independents. With on-demand printing, they could claim to offer as many titles as chain superstores currently offer today. A shop the size of the smallest of today's bookstores could "stock" a million titles.

Some high hurdles stand in the way of these new bookstores: redefining the relationships among the various elements in the bookselling chain of publisher, printer, distributor and bookseller; attitudes of the bookseller toward being involved with printing; and consumers' feelings about the bookstore and their demand for "instant books." The music industry is dealing with these issues, too, and its experience might be instructive for booksellers.

Much like on-demand printing, on-demand "pressing" of music CDs has already been proposed. Earlier this year IBM and Blockbuster Entertainment announced two joint ventures to make and dispense music CDs on demand, both in Blockbuster stores and in other retail outlets. (Making the CD would take about six minutes, IBM and Blockbuster estimate.)

The joint ventures were received coolly by music publishers, according to an article in December's *Musician* magazine, more because they challenged a well-defined business structure from which the publishers and distributors profited, rather than significant fears about the new technology. The publishers say that this new system could make their current manufacturing and distribution capabilities redundant—perhaps useless.

OUTLOOK 94

They don't object to the idea of better access to obscure titles, they just want to be the ones controlling the dollars.

One crucial issue concerning the acceptance of the custom CDs as well as custom books is, who decides what goes into the custom product, and who will profit? Even though IBM and Blockbuster announced that they intend the system to copy whole CDs intact, the music publishers are wary of technology that would allow songs to be mixed and matched from various albums. McGraw-Hill's Primis electronic custom publishing system was similarly hobbled by other publishers' desire to keep control over their copyrights, and not releasing their materials to be included in a McGraw-Hill product, even though they knew they would be paid proper royalties, even on items that never sold well before.

This limited availability of titles might also be a major hurdle for consumers' acceptance of custom electronic products. In the late 1980s they showed little enthusiasm for an experiment by which they could create audiotapes in booths supplied by Personics Corp. in some Tower Records shops. The company offered a limited selection because of difficulties getting copyrights from publishers, and customers seemed to prefer to buy the prepackaged tapes.

Crossing Over

Of course, even assuming there is a wholesale changeover among retailers to on-demand printing, there are many areas where the new technology wouldn't completely wash away the old forms. Certainly on-demand printing would make sense for many backlist and midlist titles, for which demand is steady but without a concentrated rush at any given time. For

blockbuster titles, however, huge demand concentrated within a relatively short period would make printing in-store less attractive, mainly because of the amount of time required.

An alternate view is that the crossover from collegiate bookstores to trade stores might first come in mass market paperbacks. There has been a shift in attitude in recent years among bookstores' clients from being "book readers" to being "book consumers." The distinction is that consumers buy more, but want to do so at the lowest possible price, regardless of quality of production or point of purchase—from book clubs, chain stores or drug-stores.

If the largest single piece of a book's cost is distribution, then eliminating the cost of warehousing and distribution by truck and rail will lead to greater profits for the publisher, and lower cost to the buyer—which might be just the market impetus necessary to force the issue.

Wholesalers on the Hot Seat

Overall, the effect of new technology would be to rationalize the business, increase competition, improve service and allow for more flexible marketing decisions. But revolutionary change doesn't come easily. Of all segments of the book industry, traditional printing and distribution companies are the most aware of the new technology and have the most to lose.

Wholesalers are expanding their traditional bounds: nearly every wholesaler is now a marketer as well, offering a slew of buying programs, catalogues, recommendations and more to booksellers. Ingram Book Co., for example, is putting more emphasis on its Publisher Resources Inc. subsidiary, which acts as a distributor in the Publishers Group West mode. In addition, two weeks ago Linwood Lacy, former head of Ingram's microcomputer subsidiary, was appointed presi-

dent of Ingram Industries (see News, p. 25), vaulting over the long-time No. 2 person, Phil Pfeffer, with the rationale that Lacy has more experience with the microcomputer business than Pfeffer. The corporate parent clearly sees a shift in the marketplace.

Wholesalers who want to fill their traditional role could specialize, cultivating nonbookstore outlets such as cooking stores, gift shops and other places outside the traditional trade sales channels (the costs of a printer wouldn't be justified for a few shelves of garden books at the garden center, for example). Wholesalers could print books themselves, too. Mail-order books, particularly from book clubs, by definition, would require some use of traditional printing.

Even the world's largest printer is adapting. R. R. Donnelley has a Books on Demand software program that operates much like Cornell's coursepack program. Donnelley also has taken other measures, including the purchase of printer Haddon Craftsman, in large part because Haddon adds a computerized fulfillment system that can provide a range of services, from taking orders to shipping books directly to bookstores.

Last year saw the first real impact of on-demand delivery in collegiate coursepacks. Because the campus bookstore and the university press have interlocking spheres of influence, look in 1994 for the further entrenchment of coursepacks, but also the expansion of demand into scholarly publishing, and perhaps even into some trade shops that serve local campus needs.

The wave of the future is hitting shore already, and the biggest push may come from end-users. When readers understand what it means to be able to get any title in a matter of minutes at their local "bookstore," those in the industry who are prepared will fare the best. □

Copyright 1994 Information Access Company;
Copyright Coastal Associates Publishing LP 1994
Computer Shopper

April, 1994

SECTION: Vol. 14 ; No. 4 ; Pg. 62; ISSN: 0886-0556

LENGTH: 267 words

HEADLINE: "Cash register on a chip" may revolutionize software pricing and distribution; Wave Systems Corp.

BYLINE: Strattner, Anthony

BODY:

A little-known company called Wave Systems Corp. is poised to radically change the way software is sold. The New York City-based startup has developed a chip--or more precisely, an encryption algorithm--that works as a kind of cash register, unwrapping software application features as needed and charging users accordingly. For example, you could use the financial functions of a spreadsheet without paying for its trigonometric functions.

Call it digital vending, said Jim Kollegger, Wave Systems' vice president of marketing. Untitled as yet, the company's product can even establish credit, not unlike a Pitney Bowes postage meter, by phoning your credit-card company via

modem and ensuring that the software publisher gets paid (with presumably smaller cuts for the hardware vendor and Wave Systems).

Any type of pricing scheme can be supported, Kollegger claims. He naturally wants to cash in on the coming torrent of CD-ROM-based applications, and is ardently courting major software publishers with CD-ROM distribution plans.

Wave's information-metering scheme is currently undergoing beta testing at corporate sites that use large financial and legal databases. The company also needs to push its technology into the installed PC base, and for that, Wave is relying on its first licensee, **National Semiconductor Corp. (NSC)**.

NSC is using Wave Systems' algorithm combined with its own encryption technology called iPower--a hardware-based access card, known as a token, for securing and processing secret keys and data.

The shirt-pocket-sized card will ship later this year.

SIC: 7372 Prepackaged software ; 3674 Semiconductors and related devices ;
3577 Computer peripheral equipment, not elsewhere classified

IAC-NUMBER: IAC 15040695

IAC-CLASS: Computer; Health

LANGUAGE: ENGLISH

LOAD-DATE-MDC: March 16, 1994

THIS PAGE BLANK (USPTO)

Copyright 1994 Information Access Co., a division of Ziff Communications Co.;

Copyright Pemberton Press Inc. 1994

CD-ROM Professional

March, 1994

SECTION: Vol. 7 ; No. 2 ; Pg. 134; ISSN: 1049-0833

LENGTH: 1106 words

HEADLINE: New distribution option for electronic publishers; iOpener data encryption and metering system for CD-ROM use; Column

BYLINE: O'Conner, Mary Ann

BODY:

I recently attended a presentation on CD-ROM and multimedia which was sponsored by Dataware, Adobe, Emerging Technology Applicants (ETA), IDG, Sony, and Hitachi. A particularly interesting portion of the program featured a presentation by Mr. Sandy Sanderson, Director of Business Development for the iPower Strategic Business Unit of National Semiconductor.

Sanderson introduced a new technology for information distribution that is currently under development at National Semiconductor. This innovative concept promises to open new doors into the world of electronic publishing for many information providers and software developers. First, a little background.

Software developers currently spend a great deal of time and money duplicating floppy disks and documentation. Distribution channels often eat into profits in exchange for increased competition for shelf space and distribution muscle. At the same time, software prices are dropping, further shrinking revenues.

Information providers are faced with selling customers large databases of information, much of which they cannot use, at relatively high prices. This tends to limit their potential markets for the information contained within these databases.

For both environments, the existing methods for distribution of information or software are proving to be inadequate for the growing information age and the way people need to access information.

A REAL EYE OPENER

National Semiconductor, in conjunction with Wave Systems Corp., is developing the iOpener system to address the questions of how to continue to be profitable while selling products at lower prices and how to streamline the distribution of information products. The iOpener technology is based upon the concept that whatever the customer uses is paid for and that the customer pays only for what he/she uses. The result is a type of information meter.

While the iOpener system will not be specific to any delivery system, its first products will use existing CD-ROM technology.

By combining encryption technology and metering capability, Wave Systems has developed an information distribution system which enables information

providers and users to track and monitor information usage on a pay-per-view basis. Specifically, CD-ROM discs are used as the storage media, encryption protects the data on the CDs from illegal access; a chip or board installed in a PC or LAN server functions as the meter; and the specific usage is recorded and billed at Wave's processing center. The underlying principles are that the consumer pays only for the specific data used, and the publisher is protected from illegal distribution and gains specific market feedback regarding what information is being purchased.

Wave Systems Corp. and National Semiconductor are building partnerships with industry leaders in order to promote this distribution system as the new industry standard. National Semiconductor has been given the license to manufacture the meter or chip; National Data Corp. is providing the centralized processing system which will track all purchases made at the desktop; and Personal Library Software will provide one of the search interfaces for the CD-ROMs. Wave Systems will determine all encryption algorithms, maintain the management system, handle billing functions, develop relationships with publishers, and market the system to various vertical industry segments.

A TWO STEP PROCESS

Wave's metering system is PC-based and involves two steps: encryption of data which protects information from unauthorized use, and metering which records retrieval of information by the consumer. Users are charged only for the information they use, not for online search time, printing, or for the entire contents of a CD!

ENCRYPTION OF DATA

Encryption of data is the first step in the process. Wave uses an encryption algorithm known as the Data Encryption Standard (DES) to encrypt all information before it is provided to consumers on CDs. Wave encrypts the data according to the publisher's instructions. For example, the publisher can decide how to package the information--whether to sell it by the paragraph, the page, the article, or the picture--and Waves can then encrypt it accordingly. Once encrypted, the information cannot be retrieved from the CD with being "decrypted" and recorded on the Wave meter. This process protects the data from illegal or unauthorized use.

METERING USAGE

The Wave System uses a small computer board or chip that incorporates the decryption technology with a metering capability. The meter is placed within the user's desktop PC or LAN server and is activated as soon as an end-user wants to retrieve a "package" of information. The meter records the user's statistics (name, billing code, etc.), checks the password, decrypts the information for the user, and creates a billing record of what information was purchased. Periodically, the meter automatically dials Wave's operations center using its built-in modem and uploads all transaction records. These records are then used to generate monthly bills and usage reports. The bills can be set up according to projects, departments, clients or end-users, so that customers can create any bill-back procedures that are helpful. Publishers, in turn, can obtain direct feedback on what parts of their information products are being used by the

market.

INFORMATION RETRIEVAL BY THE USER

The Wave system enables consumers to have immediate desktop access to virtually any information related to their professional or personal needs. For example, a consumer can maintain a desktop library of encrypted CDs at little or no cost; the CDs sit unused until the user needs and retrieves a specific piece of information or software. The user is not charged for search time or for browsing titles and abstracts that come up in response to a search. A cost is incurred only when the consumer decides to purchase the full text of an article or document, or perhaps buy a piece of clip art, or rent or buy a software package or video game. At that point, the user is informed of the price, and the data is immediately decrypted as the purchase is recorded by the Wave meter.

COMMERCIAL AVAILABILITY

Wave System is currently conducting a beta test of its metering system within the legal and accounting markets. The first version of the system being tested is a standalone version; a network version will be tested shortly. Wave is simultaneously building partnerships to promote the new system. Commercial introduction of the Wave system is planned for early 1994.

SIC: 3674 Semiconductors and related devices ; 3577 Computer peripheral equipment, not elsewhere classified ; 7372 Prepackaged software ; 2700 PRINTING AND PUBLISHING ; 2741 Miscellaneous publishing

IAC-NUMBER: IAC 15186610

IAC-CLASS: Trade & Industry

LANGUAGE: ENGLISH

LOAD-DATE-MDC: June 09, 1994

Intellectual Property Header Descriptors: A Dynamic Approach

by Luella Upthegrove and Tom Roberts

ABSTRACT The global electronic library will need standards that facilitate the controlled distribution and protection of digitized intellectual properties, and that encourage library expansion and access. This paper describes a system based on intellectual property distribution and protection that is currently being tested at Case Western Reserve University, and defines a global header descriptor applicable to the electronic distribution of intellectual properties.

INTRODUCTION The mission of the Library Collections Services Project (LCS) at Case Western Reserve University (CWRU) is to establish an online multimedia repository to serve the academic and research needs of the CWRU community. To this end, LCS has created a number of prototype applications that demonstrate the potential of a networked multimedia repository. These prototypes address the interests of the providers and consumers of intellectual property (IP) resident in the repository.

Early on, the LCS project team recognized the responsibility it had to maintain and protect the electronic IP. The team collected IP management requirements by meeting with members of the publishing and legal communities, reproduction rights organizations, librarians, online information service providers, and academicians.

The issues that resulted from these meetings fall under the general headings of: IP protection, IP use management, and royalty compensation.

DEVELOPING THE LOCAL ENVIRONMENT

The LCS team began its system design by defining end-to-end system components based on the requirements gathered. The requirements fell logically into the broad categories of: Ownership, Compensation, Permissioning, User Access, Privacy/Confidentiality, and Permitted Uses. Prior to building the prototype repository and applications, appropriate permissions were obtained from participating rightsholders.

Applications were designed that verify user authorization, access the IP, and manage IP use before and during repository access. These applications compare user information and usage data coupled to the IP to determine access and use. Comparison of user information and usage

data satisfies the protection and use requirements detailed in license agreements negotiated with IP rightsholders.

These applications, all of which adhere to a set of data and protocol standards, are called *compliant applications*. Only compliant applications can access the IP in the repository.

EXPANSION TO THE GLOBAL ENVIRONMENT

To expand this model, consider that all IP consumers are part of the local environment. They access repositories on which they have registered accounts, and information passed between user and repository is managed at the local level. Users maintain the ability to query information contained in remote repositories; however, the request for the IP transacts between the user's local repository and the remote repository in the global environment.

The LCS model can be expanded to this global environment. To accomplish this the following assumptions, significant issues in themselves, are made:

1. Permissions for storage, access, use, and compensation have been negotiated and agreed upon.
2. Compliant applications are resident and in use on all participating systems.
3. Economic structures for billing and compensation have been established.
4. Technical strategies for locating IP are in place.

A typical global transaction may develop as follows.

A user locates and requests IP on a remote repository. The request is routed through their local repository to the remote repository where the requested IP resides. The protocol of this communication contains a standard request that includes information identifying the IP, the requesting repository, the user environment specifications, and the intended use. The remote repository verifies the request, constructs a header descriptor based on that request, and replies to the requesting repository. This header descriptor is in the form of a standardized global header descriptor.

Using the local environment presented earlier as a base, data common across the global environment can be identified. The following elements are proposed for inclusion in a global header descriptor.

Ownership

to include rightsholder identification and contact information for use in compensation, special permissioning, and copyright code compliance.

Permitted Uses

to include uses as negotiated with the rightsholder detailing authorized users, display resolutions, print capability, etc.

Royalty Compensation

to include the compensation framework as it relates to the permitted uses.

IP Attributes

to include the physical attributes, and component parts comprising the IP.

To accommodate this information, each descriptor element would contain a variable length data string preceded by a standard ID. These elements would be mapped to the local repository for use by functionally compliant applications.

In the form of a dynamically generated global header descriptor, information common, particular, and primary to IP providers and purveyors can be developed to enable global and local protection and use management.

CONCLUSION

The challenge of developing standards for the global electronic library may seem overwhelming; however, inaction will render the vision vain. Opportunities are afforded to those who begin now to define the framework of the new environment.

This paper presents a prototype system designed for intellectual property protection and use management in a local electronic environment. It then begins to describe a global header descriptor based on the two premises that: the electronic environment is comprised of local users connected to primary global repositories, and that intellectual property access is mediated by applications compliant to established protection and use monitoring requirements.

To this end, it is proposed that a global header descriptor contain a set of data elements that identify intellectual property: Ownership, Permitted Uses, Royalty Compensation, and IP Attributes.

Local and global standards must cooperate to provide access and use controls such that IP providers, purveyors, and consumers are confident that their interests are protected. Properly designed standards will enable repositories to fulfill their responsibilities, and encourage the use and expansion of the global electronic library.

BIOGRAPHY

Thomas Roberts, DBA Communication Media and Documentation Services, is consulting with CWRU's LCS Project. CM&D specializes in knowledge transfer using established and emerging technologies. With CWRU, CM&D is identifying and analyzing copyright, permissioning, and royalty compensation issues as they apply to electronic intellectual property distribution.

Tom Roberts
Library Collections Services
Case Western Reserve University
10900 Euclid Ave., Baker 6
Cleveland, OH 44106-7033
tfr4@po.cwru.edu

Luella Upthegrove, Database Administrator for CWRU's LCS project, helped design and develop the prototype electronic library, and is currently involved in planning for the second version of the system.

Luella Upthegrove
Library Collections Services
Case Western Reserve University
10900 Euclid Avenue
Cleveland, OH 44106-7033
(216) 368-8921
FAX: (216) 368-8880
lru@po.cwru.edu

Copyright 1994 IDG Communications, Inc.

InfoWorld

May 2, 1994

SECTION: ENTERPRISE COMPUTING/MANAGEMENT; Pg. 84

LENGTH: 1233 words

HEADLINE: Metered PCs: Is your system watching you?;

Wave Systems beta tests new technology

BYLINE: By Shawn Willett

BODY:

Utility companies meter your gas and water usage. Nielsen meters what TV programs their families watch. Some employers even meter their employees' phone conversations. It was probably inevitable that someone would come up with a metered PC to keep track of what software you use.

A chip set to be introduced this fall from Wave Systems Corp., a 4-year-old start-up, can do just that. The WaveMeter records exactly what software you use, when you use it, and for how long. Analysts say the technology could solve the persistent problem of how to charge fairly for software and prevent piracy.

For some IS managers the chip could be a dream come true -- providing

flexible and accurate licensing arrangements, not to mention a wealth of usage and inventory management information. For others the scheme could be a Big Brother nightmare with your vendors essentially monitoring every move your users make.

TERRIFIC TECHNOLOGY. Wave Systems, which was founded by Peter Sprague, chairman

of National Semiconductor Corp., has been beta testing the technology, now in board form, for several months. Software needs to be prepped by the publisher or a reseller in order to be tracked by the monitoring scheme. The system consists of a chip, called the WaveMeter, that collects information on what software is in use, encrypts it and then, using a built-in modem, sends the information to the transaction center. The chip not only keeps track of what is being used, it also creates a bill and sends it to the company.

A database at the transaction processing site also keeps customer credit and debit information for automatic billing. WaveNet, in turn, forwards royalties to software resellers or software providers to complete the last link in the financial chain.

Wave hopes to eventually offer software electronically to users and even allow them to rent some software for a period of weeks or months.

Observers say that regardless of its uses, the Wave Systems platform is ready for prime time. "The technology is terrific and usable," says Karen Cone, program director for software asset management at the Gartner Group in Stamford, Conn. Gartner has hosted several demonstrations of the technology at

conferences for software publishers and others. "It could eventually get rid of the overhead associated with the distribution and tracking of software."

Wave officials say they are building interfaces so the information collected by the chip can be passed on to license servers being developed by Novell Inc. and Gradient Technologies Inc. Wave also plans to develop hooks into asset management software to give IS managers a better handle on what software is on their network.

Novell demonstrated the capabilities of the chip with its NetWare Navigator software distribution system and its NetWare licensing server in March. The licensing server is scheduled to ship later this year.

PCS WITH METERS. Wave is offering the chip to PC manufacturers, at a price of \$ 8 to \$ 10 per chip in OEM quantities. Wave company officials claimed that other big names are backing the project as well. Without naming vendors, Camille LeBlanc, manager of software licensing at Wave, says two of the top five PC manufacturers have agreed to pre-install the chip on their systems this fall.

According to LeBlanc, software providers and resellers have also shown interest in prepping their software so it will work with Wave's chip. So far Personal Library Software Inc., Dataware Technologies Inc., and a few other providers of on-line, broadcast, and other high-priced content information, have agreed to modify their software so it can be metered by the WaveMeter chip.

"Software publishers are interested because it means they can move to more flexible licensing options," LeBlanc says. "In the past they had no reasonable

mechanism to secure their intellectual property to do that."

Not coincidentally, publishers realize the technology could cut down on the illegal use of software if the WaveMeter and meter-enabled software becomes pervasive, analysts say.

Beta testers are using the Wave hardware now, for the most obvious applications -- metering of access to expensive content material such as financial information. Customers in the vertical, financial, and law firm markets say the technology is saving them money.

According to Tom Flemming, library director at the Piper & Marbury law firm in Baltimore, his company is using WaveNet to view cases that are on CD-ROM. "I'd rather pay \$ 2 a case than \$ 20,000 for the whole CD-ROM," Flemming says.

SOFTWARE RENTALS. Analysts say the renting of expensive software programs will be the second biggest use of the new technology.

"This will be particularly useful if you are running some kind of temporary event," Cone says. She added that companies will also find it expedient to rent expensive CAD or computer-aided software engineering (CASE) software to do specific projects, then return the software and pay a bill.

Usage-based pricing for specific time periods makes sense on high-end applications, agrees Kathryn Roy, vice president of strategic marketing at Lotus Development Corp. Desktop software pricing, on the other hand, may be another matter: "We are talking about products that cost about \$ 100," she says. It probably wouldn't make sense for someone to pay a \$ 10- or \$ 20-per-day rental

charge, Roy says.

ASSET MANAGEMENT. The actual metering and billing of applications, especially off-the-shelf software, will follow eventually, analysts say, if only because IS managers are desperate for an accurate way to measure usage in their organizations.

The scheme makes sense in sites with tens of thousands of PCs, where global site licenses are often based on educated guesses of usage, analysts say.

"We can tell [software companies] exactly what we are using. If we have 200 copies of a program and only 160 people are using it, then the 160 is debited against our account," says Flemming, who is enthusiastic about the metering.

However, analysts and IS managers are less enthusiastic about whether WaveNet's proposal to automatically bill users will fly. First, there are security concerns about sending a company's usage data outside the company that no amount of encryption will alleviate.

The Big Brother aspect can also loom large. Having an outsider monitor users' every move can give employees the distinct impression that they are not trusted, analysts note.

"There are a lot of questions. For example, would it charge you every time you execute a program and leave the image up on the screen or when you actually use it," says Ken DiGuilio, senior PC specialist at Peterson Consulting LP in

Chicago and member of Chicago Association of Microcomputer Professional's licensing committee. "It seems like the least practical way of doing things right now," he adds.

IS managers also need to be able to ease into an external monitoring system.

"The user has to be in control of the data for this to work," Cone says. She adds that the user and IS manager need to agree if the user wants to take the WaveMeter out. She predicts many users will initially use the information gathered by the WaveMeter as a bargaining chip to negotiate a deal with the software provider.

"Otherwise," Cone says, "It could be pretty intrusive."

GRAPHIC: Diagram 1, no caption, LYNNE FOY; Diagram 2, How much is that program

being used? The WaveMeter system can keep track of how much an application is used or whether unlicensed software is loaded on the metered PCs., G.

BOREN/INFOWORLD.

LANGUAGE: ENGLISH

LOAD-DATE-MDC: May 10, 1994

Copyright and Information Services in the Context of the National Research and Education Network¹

by R. J. (Jerry) Linn

ABSTRACT

The High Performance Computing Act (HPCA) of 1991 (P.L. 102-194) places unenforceable requirements to protect copyrights and intellectual property rights on the National Research and Education Network (NREN). This paper discusses the roles and responsibilities of the NREN and associated information services; technical approaches to authentication, redistribution and authorization of use of electronic documents over the NREN; and an amendment to the High Performance Computing Act.

INTRODUCTION

It is clear that when the High Performance Computing Act of 1991 was written the notion of digital libraries was a consideration of the authors. It is also clear that the Congress intended that copyrighted materials be distributed over the Network. The legislative history of the Act affirms this position. The Act, as drafted in the 100th Congress (S.1067 and H.R. 3131, 1990), included provisions for authorization of appropriations for the National Science Foundation to establish digital libraries. Other bills introduced into Congress which provide for similar authorization of appropriations include S.2937, introduced in 1992, and S.4, introduced in 1993. Indeed, prior to 1991, digital libraries were integral to the thinking related to "information services" and were the stimulus for language incorporated into the HPC Act of 1991 with respect to protection of copyright.² The language employed in the Act of 1991 assumes information services are embedded in the Network, as part of a network infrastructure. However, the term "network" has very specific and narrow connotations when used by professionals in the computer and communications communities versus the broad definition of the term in the Act. Furthermore, recent papers and reports from a workshop focused on the National Research and Education Network (NREN) reflect the common understanding that the NREN is *only an access medium* to application services.³ Therein lies the weakness of the legislation: the definition of the "Network" is too broad to assign responsibility for protection of copyright and intellectual property rights. Furthermore, the professional community to whom the courts would turn for expert witnesses to aid in interpretation of the law is not likely to agree with the reasonableness of the requirements that the Act places on operators of the Network or the

ability to enforce its provisions except in the computers attached to the Network which offer information services.

Specifically, the Act defines the "Network" in Sec. 4 as follows:

(4) 'Network' means a computer network referred to as the National Research and Education Network established under section 102; and Sec. 102 (c) "Network Characteristics" states:

The Network shall —

....

- (5) be designed and operated so as to ensure the continued application of laws that provide network and information resources security measures, including those that protect copyright and other intellectual property rights, and those that control access to data bases and protect national security;
- (6) have accounting mechanisms which allow users or groups of users to be charged for their usage of copyrighted materials available over the Network and, where appropriate and technically feasible, for their usage of the Network;

There are several important things to note because they become "first premises" for a discussion. First, the NREN is a concept (the Act never defines who owns and operates it; the Act authorizes appropriation of Federal funding to agencies to implement the concept). Second, the NREN is a logical entity derived from a network of networks (an internet). And third, the NREN is a part of the *Internet*—that network of networks whose span is global and whose common denominator is a shared name and address space.

The Network established under Sec. 102(a) does not imply that the Federal government installs or owns the physical assets of the NREN (e.g., optical fiber cables, routers) nor does it preclude the NREN from being derived from commercial, private sector sources and services. *This ambiguity is important.* The definition and ownership of the NREN are not cast in concrete (like highways); this omission allows the NREN (or parts of it) to transition from government provided and/or subsidized services to commercial for-profit services, or an evolving combination of both. Evolving Federal policy supports transition to commercial services as required services become commercial commodities.

Which networks comprise the NREN and who owns/subsidizes them are not as important as understanding that "ownership" of subnetworks, levels of subsidy and recipients of subsidy are all subject to change over time. Therefore, defensible answers for issues related to copyright, intellectual property rights and the NREN must take into account the diversity of the technology base in component subnetworks, of ownership, of agency missions and goals, and of those services accessed by the NREN versus common services provided by subnetworks comprising the NREN/Internet. This complexity suggests that it will be beneficial to partition the problem into smaller components for analysis and discussion.

Subsequent subsections present the "Network" as a set of services, establish both technical and pragmatic reasons for doing so, and discuss protection mechanisms appropriate for the decomposed services. Specific

technical mechanisms are outlined which may be employed to distribute and protect copyrighted materials by an information service. Finally, an argument is presented that the HPC Act of 1991 should be amended such that the protection of copyrights and intellectual property is properly the responsibility of information service providers and users. An amendment is offered which would realize the position presented.

DELINEATION OF SERVICES

A delineation of network services aligned with widely recognized technical boundaries and terms will aid in a dialogue because functions and responsibilities can be discussed within an established framework. Professionals familiar with network architectures associate specific functions and services with well-known named layers of a network architecture. The terms and concepts used below are recognized by an international community of computer and communications professionals.⁴ Thus, it is unnecessary to define new terms and concepts in order to establish a framework to discuss issues.

The functions associated with the two lowest layers of a network architecture are *physical*, point-to-point connectivity and signaling, and data transmission via *data links* which interconnect computers or routers. Next in the hierarchy are *network-layer* functions which select routes and relay data packets enroute to their destination. These functions are the least common denominator of a "computer network" and are often implemented by routers which comprise or interconnect wide area subnetworks.

The *transport layer* establishes end-to-end connectivity and may provide for retransmission of data packets lost or corrupted by lower layers. Thus, the transport layer provides a reliable end-to-end communications medium for application programs and services. Note that the public switched network may also be used to provide an end-to-end communications path between computers; however, end-to-end communications is achieved by different technical means.

Information services are provided by *application-layer* programs and supporting protocols at the end points of a communications path. Examples of application-layer services are electronic mail and file transfer, which are implemented by application-layer protocols (e.g., Simple Mail Transfer Protocol (SMTP) and X.400 are electronic mail protocols).

Connectivity of subnetworks in the NREN/Internet functions at the network layer (see Figure 1). Each subnetwork serves as a switching fabric for a set of computers; i.e., the network layer software receives and relays packets of data from one node in the network to another node based only on its destination address. Note that the routing and relay (switching) functions assigned to the network layer are the least common denominator of the Internet (NREN). Specifically, a subnetwork (e.g., college campus, midlevel network or the NSFnet of the National Science Foundation) may use one set of technologies and another subnetwork may use another. However, the "glue" that interconnects them is a **common, minimal set of protocols necessary to provide the routing and relay functions**. Any additional set of functions is optional in the network layer and is only likely to be incorporated if actually required in a given environment (e.g., security, network management). Therefore,

any assumption that the "Network" is a uniform, ubiquitous environment is erroneous—particularly when the NREN is viewed as a set of interconnected autonomous subnetworks.

This is a greatly simplified sketch of a multi-layered network architecture. The sketch highlights crucial networking design concepts; i.e., specific functions are assigned to layers in a network to accommodate an array of lower-layer communications technologies and for design and maintenance purposes. However, we have sufficient information and a set of terms which is rich enough to pose questions about how and where the requirements of the Act might be implemented and to explain why they might or might not be reasonable requirements in the first place. We are also prepared to identify and discuss conflicting objectives, if proper design and engineering principles are not followed.

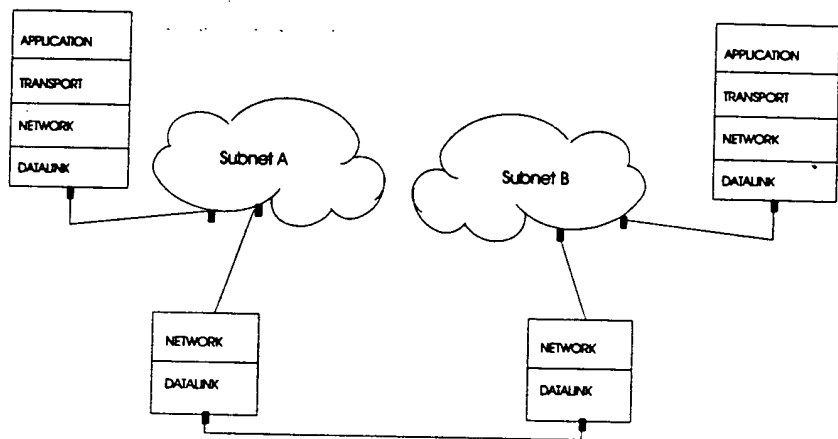


Figure 1: An Internet -- a network and networks

Thus, when the Act states:

the Network shall —

....

- (5) be designed and operated so as to ensure the continued application of laws that provide network and information resources security measures, including those that protect copyright and other intellectual property rights, and those that control access to data bases and protect national security;

we can ask: "What does this mean? What protection is required? How may required protection be achieved in the context of existing network architectures? And, who should be responsible?"

Clearly, the routing and relay functions of the network layer will not protect copyrighted materials. In fact, they do not even assure delivery of data packets. Therefore, the "Network" described in the Act requires more functions than those described for the network layer. So it is appropriate to ask: "What protection is inherent in a network; what additional protection is required; and where is it most appropriately offered?"

Under normal circumstances, network-layer software *does not* inspect the contents of data packets. There are at least two good reasons not to do so. First, inspection of packets for any purpose introduces unnecessary overhead and degrades the throughput of the network (a serious consideration in high-speed networks). Second, inspection of packets (or streams of data) jeopardizes the privacy of the information being transmitted. Also, recall that the network-layer software was described earlier as "least common denominator," with the implication that any additional functions were optional. Consequently, the network layer is not a viable candidate for uniform protection of copyrighted materials.

Data integrity protection against accidental changes is assured if specific transport protocols are employed at the end points of a connection. Specifically, the network can protect against accidental loss or corruption of data during transmission from one point to another. This is true for the Transmission Control Protocol (TCP) and the Organization for International Standardization (ISO) Transport Class 4 (TP4); both detect and retransmit lost and corrupted data. However, the transport protocols cannot protect against redistribution of materials obtained from a legitimate source, nor can they assure the authenticity of the materials transmitted over the network. The means to assure authenticity of materials and achieve protection from deliberate abuse by end users is to implement the required protection mechanisms in computer systems as part of the application programs which deliver services to users.

TECHNICAL MEANS FOR PROTECTION OF COPYRIGHTED MATERIALS

New protective services can be created for information dissemination which can also be applied to those materials that have a copyright. However, requirements for protection must be defined before describing how protection might be achieved. Below is a set of requirements which serve as a starting point for a discussion.

Protections and Features Required

Authentication: A mechanism is required to certify that any material received is a bona fide copy of the original (data authentication) and possibly who it came from (origin authentication). If the copy is not authentic, then this fact should be detectable and the copy discarded. Recall that the transport layer *may* provide for integrity protection against accidental changes, but authentication provides a means for protection against both accidental and intentional changes.

Limited redistribution: Publishers want to control distribution to those who have paid a fee for the use of copyrighted materials. Mechanisms should be implemented to restrict the number of copies printed to those paid for and to the individual who paid for them.

Protection against plagiarism and change: Authors and publishers do not want their materials used without appropriate attribution, nor do they want the materials excised, edited, or modified such that authenticity is jeopardized. Information should be stored in a form which makes it difficult, if not impossible, to remove the copyright mark, or excise or modify text.

Object form: Information should be stored and exchanged in standardized but device-independent forms. Processing software employed by a

user should display or print the materials in an appropriate form given the constraints of the user's video display and/or printer.⁵

To discourage plagiarism, excising parts of the text and other unauthorized uses of the information, an object could be put in a "sealed envelope" and distributed in one of several forms which are not easily read and modified by humans. These forms could include SGML, G4Fax, and PostScript or other useful forms. SGML denotes the Standard Graphics Markup Language. SGML text would require processing of the input text to render meaningful output on either a video display or printer. G4Fax denotes Group 4 Facsimile which is a compressed bit stream using an international standard for scanning and compressing facsimile images. It may be displayed or printed on raster scan output devices (video display or printer). G4Fax could readily be used for interlibrary exchange to avoid document handling and scanning. PostScript denotes the form used by PostScript printers. It is a page description language that is widely implemented, is useful for printing purposes only, and would not require significant processing if directed to a printer.

Appropriate remuneration: Remuneration could take the form of a subscription fee, license fee, contract, or fee for services rendered, as appropriate. Dissemination may be by an author, original publisher, information service, or library (hereafter called an authorized distribution source).

It is assumed that interlibrary loan and electronic redistribution of single copies of papers to individuals by libraries who have a subscription, license or contract with a publisher constitutes "fair use." It is also assumed that fees for services will be established (commercial, for-profit; and not-for-profit) and public access could be via public libraries. Specifically, an individual could ask for and get a copy of a paper or article as easily as he or she can reproduce it on a copier in a library (and at a comparable price). Remuneration by an individual patron could be at the time the material was obtained, if there was a fee.

TECHNICAL MECHANISMS

A set of mechanisms may be combined to address the requirements outlined above. For discussion purposes, we consider a body of material (information) as an "object" with certain components and attributes. One attribute is an electronic "copyright" mark; the object forms noted earlier are another attribute. Object-oriented technology associates processing of objects with their attributes. For simplicity, however, we describe an object as an envelope and its contents. The information on the envelope is visible and the contents hidden and sealed with a digital signature. Examples of information on the envelope could include title, author(s), abstract, keywords (e.g. full bibliographic record) and attributes describing the form of the object, a digital signature, copyright status (yes/no), and date and timestamp associated with an authorized copy. Visibility of information on the envelope has other obvious advantages related to search and retrieval of information stored in digital libraries, but they are outside the scope of this paper. Figure 2 presents a graphic perspective of the concepts.

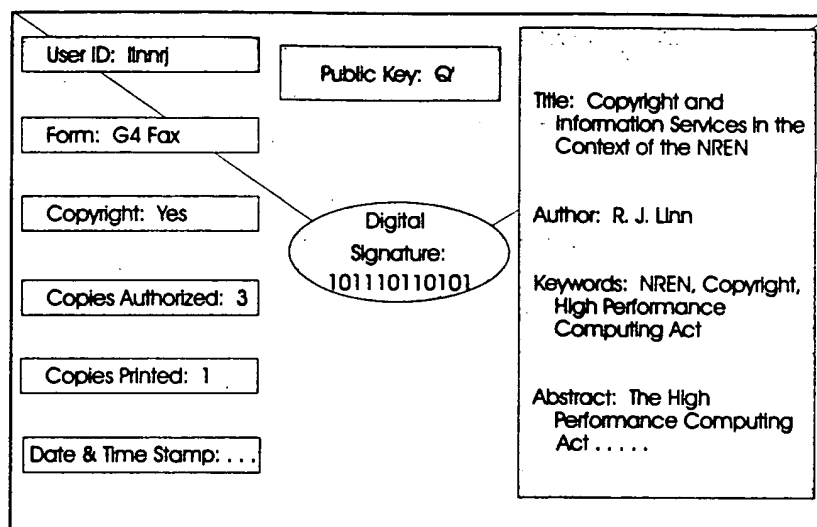


Figure 2: An Information Object -- Envelope plus Content

For our purposes we assume:

- an object is processed by standardized software (hereafter called *rendering software*);
- creating an original information object, file transfer over a network, and rendering of the information on a video display or printer are built-in functions of the rendering software;
- the rendering software is
 - inexpensive or free because it is in the interest of the public, and of publishers and authors to protect their intellectual property, and
 - widely available; e.g., distributed by publishers, information service providers, computer manufacturers;
- copies of objects are exchanged using the rendering software—a copy is obtained from an authorized distribution source (may be an individual if there is no fee for use); and
- the structure and exchange formats of objects are standardized (either de facto or de jure).

Active Protection Mechanisms

Two active mechanisms *implemented in the rendering software* will achieve the requirements for protection outlined in the previous section.

Authentication: Confirmation of authenticity of the source and contents of the envelope can be achieved by use of a public key, digital signature algorithm. The public key is provided by the author or publisher and is written on the envelope. The public key is used to verify the digital signature of the information written on the envelope and its contents. If either is changed, the digital signature verification algorithm detects and reports failure. If verification failure is detected when an object is being obtained from an information service, its retransmission

should be requested. (This might occur if data were lost or corrupted.) If a failure is detected when displaying or printing an object, further processing should be inhibited. This might indicate a bootleg copy, or a mismatch of user identification with that on the envelope, or it might indicate that the authorized number of copies have been printed. Optionally, the object could be destroyed by the rendering software when verification fails.

Limited redistribution: Identifying the holder of the copy on the envelope (e.g., user identification) and a copy counter can be employed to limit electronic redistribution. The user identification and initial value of the copy counter stored on the envelope are established when a copy is obtained from an authorized distribution source. The number of printed copies allowed is a function of the fee paid. The copy counter is used to restrict the number of copies rendered on a printer. As the copy counter is decremented, a residual copy count and new digital signature is computed and affixed to the envelope to prevent an unlimited number of copies from being printed.

Note that sending a copy of an object via electronic mail, redistribution by a bulletin board and other simple copying mechanisms will not update the contents of the envelope which contains the date and timestamp of an authorized copy. If the date and timestamp in the directory entry for a file containing an object do not match those in the envelope of the object, the rendering software considers the copy to be unauthorized. Consequently, the information contained in the envelope will not be presented to a user by the rendering software and unauthorized copies are useless.

Materials may be displayed on a video display an unlimited number of times by the user identified on the envelope. Other users are prohibited from displaying an object with the "copyright" attribute. However, unlimited rendering and redistribution is permitted if an authorized distribution source omits the "copyright" attribute on the envelope, or enters "unrestricted" in either the user identification or copies authorized fields.

Passive Protection Mechanisms

Object form: The object forms described above are not human interpretable forms (SGML, G4Fax, PostScript). Furthermore, an object is stored in a form which may not be displayed or printed without the rendering software unless it is extracted from within its envelope. Although this is a passive protection mechanism, significant technical information and expertise are required to defeat it.

Note that all the forms described above prevent easy redistribution by simply making a copy and mailing or printing it with utility software because the rendering software is required to display or print an object. These forms also inhibit using a simple editor to "cut and paste" text into another document because no form is human readable, and direct user access into the contents of the envelope is not allowed by the rendering software.

Write protection: Write protection is the first line of defense required to protect the authenticity of information disseminated by an information service. It restricts the privileges to create or modify stored information

to the rightful owner(s); these are called "write privileges" associated with a file. Restrictions are essential for any information service and must be implemented within the computer system offering the information service. Write protection is not a function of the "Network" but is a responsibility of the parties operating an information service.

In summary, two active forms of protection are proposed for intellectual property: *authentication* and *limited redistribution*. Two complementary, passive mechanisms are also identified, but are inadequate on their own (*object form* and *write protection*). All the mechanisms suggested are implementable on computers accessed by a network, and are completely independent of the networking technology used to access an information service. All mechanisms are applicable to any information distributed over a computer network whether or not the information carries a copyright mark.

SUMMARY ARGUMENTS

Separation of the roles and responsibilities of the "Network" and "information service providers" provides a logical and pragmatic framework for disentangling and discussing the legal and technical issues related to the NREN and copyright.

First, the NREN is a concept (or logical entity) rather than something physical with fixed boundaries. The present and future NREN will be part of the global Internet. As such, its owners are both public and private entities, and it is not uniform in the underlying technology deployed. Pragmatically, it is impossible to require any owner of part of the Internet (a subnetwork) to add new, optional network functions which do not serve the owner's immediate needs. Consequently, the Network as a whole can only provide the "least common denominator" services with respect to networking functions. These common functions are selection of routes and forwarding packets enroute to their destination; this is called "packet switching." Often, technical people think of the "Network" in terms of these limited functions; e.g., NSFnet provides the packet switching and routing functions to interconnect other networks.

Second, the language of Sec. 102 (c)(5) implies that operators of subnetworks which are part of the Network could be liable for the illegal actions of both the providers of information services accessed via the Network and the users of these information services; i.e., "must be designed and operated to ensure ... including those that protect copyright ..."

These requirements to protect copyrights and intellectual property rights are at odds with established protection for common carriers who also provide networks capable of providing access to information services which distribute copyrighted materials. Carriers are not liable for the illegal activities of their users. Surely, a telephone company would not be held legally liable if an information service used facsimile machines to illegally sell and distribute journal articles. Note that it is technically feasible for the NREN to become integrated with the public switched network in the near future (e.g., narrowband ISDN services (Integrated Services Digital Network) could be used to access the Internet). Using this situation as an example, there could be a dichotomy in terms of requirements and liabilities related to operators of subnet-

works with respect to a single illegal act; e.g., if part of the access path was via the public switched network and part via a midlevel network.

Third, consider that the "operator of the Network" is responsible for collecting and redistributing fees to the "appropriate entity" for use of copyrighted materials (c.f. Sec. 102 (c)(6) in the introduction). Is it likely that private sector providers of information services (e.g., a publisher) want an intermediary (Uncle Sam/Federal agencies) to collect and redistribute funds for services rendered? Even if an information service did want this service, which "network operator" is responsible (or would accept the responsibility)? Federal agencies operating a subnetwork do not want the responsibility of collecting and redistributing fees for private sector parties. Note that definitions of "operator of the Network" and "appropriate entity" (author, publisher, ...) are open questions. Particularly, when user access is granted via a sequence of subnetworks, who is the network operator? Is it the "operator" who provides the "user" access to the network, the operator who connects the information service, both, or some more complex combination?

Finally, a number of network-independent mechanisms may be employed by information service providers to limit redistribution and assure that copies remain unmodified. These include data compression, authorized use meter (copy counter), and public-key, digital-signature techniques. Digital signature can be employed as a tool to "seal an envelope" and verify the authenticity of copyrighted materials distributed over the Network. These mechanisms can be implemented to protect copyrights and the interests of publishers and authors completely independent of the network technology used to access the materials.

Definition of standardized technical practices to achieve the desired results and inexpensive software to distribute, protect and render copyrighted materials are all that are needed to protect the interests of publishers and achieve the intent of the High Performance Computing Act.

CONCLUSIONS

Sections 102 (c)(5) and (6) of the Act place unrealistic and unenforceable requirements on the "Network" and its operators (Federal, State or private sector parties) to (1) protect copyrights and intellectual property rights; and (2) account for use, collect fees and remunerate copyright holders. These should be the responsibility of the information service providers and users of information services. These are unrealistic burdens to place on Federal agencies or private sector operators of subnetworks which are part of the NREN (Internet).

While it is impossible to assure complete protection against malicious individuals, the appropriate remedy is to develop and deploy technical protections in the appropriate places, and apply the law in the same manner it is used to prevent bootleg copies of paper documents being reproduced on copiers.

The rationale developed in this paper could be used to interpret the existing law and develop regulations and rules aligned with the proposed amendment. If regulations and rules with the same intent were written, they would not clarify the intent of Congress⁶ and would be more readily challenged in the courts. An amendment would clarify the intent of Congress and make the law enforceable. The author believes that clarity

on this issue is in the public interest as well as that of authors and publishers. To this end, an amendment is proposed as an appendix.

APPENDIX Proposed Amendment to the HPC Act of 1991

Insert the following definition at the end of Sec. 4.

“(6) “Information Service Provider” means an entity or individual who disseminates information, data, or copyrighted materials to others, for free or for fee as appropriate.”

(Note that this definition is broad enough to include libraries, for-profit publishers, or individuals who want to participate in an “electronic press”—and is not restricted to the dissemination of copyrighted materials).

Substitute the following for Sec. 102 (c)(5) and (6):

The Network shall —

....

“(5) be designed and operated so as to enable the continued application of laws, regulations, directives and standards that prescribe security measures for network and information resources and those that control access to data bases and protect national security;

“(6) have accounting mechanisms which allow users or groups of users to be charged for their usage of the Network, where appropriate;”

and insert after Sec. 102 (e) —

“(f) Information services which distribute copyrighted information shall be designed and operated so as to enable the continued application of laws which protect copyrights and other intellectual property rights, including appropriate remuneration of copyright holders, while allowing for the ‘fair use’ provisions of the copyright law.”

and renumber Sec. 102 “(f)” and “(g)” as “(g)” and “(h)”.

NOTES

1. This paper is a contribution of the National Institute of Standards and Technology. As such, it is not subject to copyright. The opinions expressed in this paper have not been endorsed by the Federal Networking Council, or any other federal working group.

2. These provisions were first specified in a draft of H.R. 3131, Title III, “Information Services,” Sec. 302, “Copyrighted Materials,” 1990.

3. Proceedings of the NREN Workshop, Monterey, CA, Sept. 16-18, 1992, EDUCOM.

4. The terminology employed is based upon the “Open Systems Interconnection—Basic Reference Model,” published by the Organization for Interna-

tional Standardization in 1984. A similar delineation of functions and terminology is used in the Internet architecture defined by the Internet Architecture Board/Internet Engineering Task Force.

5. The techniques described in this paper are equally applicable to output media other than video displays and printers. Thus, "object form" is intended to denote some machine-processable form of digital information which requires "rendering software" to present the content of the information in a human interpretable form—video, audio, printed text or some combination thereof.

6. In a conversation with the author, Mike Nelson, who was on then Senator Gore's staff and now is in the Office of Science and Technology Policy in the White House, said, "Yes, we knew headers were required, but protection of copyright by the 'Network' is essential. Thus, the law reflects the intent of Congress."

BIOGRAPHY

R.J. (Jerry) Linn, a computer scientist, is Associate Director of the Computer Systems Laboratory at the National Institute of Standards and Technology (NIST) in Maryland. As a Commerce-Science Fellow in the U.S. House of Representatives, he worked on the High Performance Computing Act of 1990. His research activities include formal protocol design, specification and testing.

R.J. Linn

Associate Director for Program Implementation
Computer Systems Laboratory
B164 Technology Bldg.
National Institute of Standards and Technology
Gaithersburg, MD 20899
linnrj@osi.ncsl.nist.gov



FOR IMMEDIATE RELEASE: March 4, 1994

FOR MORE INFORMATION:

EPR Electronic Commerce Technologies
David Van Wie
(408) 774-6146

National Semiconductor
Lori Sinton
(408) 721-2448

National Semiconductor and EPR Partner for Information Metering/Data Security Cards

iPower(tm) Business Unit Adopts VDE Technology for Digital Rights Protection

Sunnyvale, CA - March 4, 1994 - EPR Electronic Commerce Technologies announced today that it has licensed its Virtual Distribution Environment (VDE) technology to the iPower Business Unit of National Semiconductor. VDE is a security environment for data, programs, and content, and is based upon breakthrough research in the areas of Information Metering and Digital Rights Protection. National iPower data security cards are the first impenetrable intelligent cards capable of performing sophisticated security functions such as securing and processing secret keys and data.

The National/EPR partnership will integrate VDE technology into the absolutely tamper proof environment of the iPower card. Products resulting from this partnership will make the highest level of data security and information access widely available in a personal form for all kinds of applications including electronic funds transfer, electronic commerce, electronic information vending, and access, authorization, and identification systems.

"VDE delivers a rich array of secure transaction control, auditing, reporting, budgeting, privacy, and related capabilities" said Victor Shear, President and CEO of EPR. "These capabilities are used for managing and analyzing the usage of electronic content and protecting the rights of electronic community members. They address the transaction management needs of all electronic highway participants, from government intelligence agencies, to creators of entertainment products, to providers of telecommunications infrastructure," Shear said. "We believe that delivering these capabilities in a highly secure, personal form, such as the iPower Data Security card, will meet the most demanding needs of emerging Electronic Commerce systems."

"The nice thing about the technology is it's portability. It lets you take your purchasing and access rights with you wherever you go" said Walt Curtis, Vice President of the iPower business unit at National Semiconductor. "We are working to identify how we can implement VDE technology in various chips and subsystems we build within the iPower business," Curtis said. "The technology could go into a variety of hardware tokens, PCMCIA cards, and Smart cards. It will likely appear in hardware meters or on a PC motherboard or set top converter."

BEST AVAILABLE COPY

EPR has designed the Virtual Distribution Environment to provide metering, security, administration, and privacy filtering technologies for electronic content distribution and data security applications. VDE is scaleable and will operate in portable and desktop personal computers, servers, mini-computers, mainframes, set-top controllers, cable head end systems, personal digital assistants (PDAs), telephone switching networks, and digital consumer appliances. VDE supports a wide variety of control mechanisms. For example, VDE enforces:

- charging users of information databases different rates for viewing or printing information;
- charging advertisers each time a user sees an ad in an electronic newspaper (and charging advertisers greater fees if a user requests more detailed information);
- allowing a movie to be viewed a certain number of times;
- and allowing users to extract fragments from audio, video, or multimedia products and basing charges according to the nature and quantity of content taken.

iPower technology centers around a specialized semiconductor called a Secure Processing Unit. This chip allows applications builders to "hide" one or more encryption engines, encryption keys, application data, and the CPU that processes them. All this electronic information is protected from unauthorized read-out or reverse engineering. iPower technology puts encryption keys and secret application structures and data inside a portable token. If lost or stolen, the token is resistant to the efforts of the most resourceful and determined thieves.

EPR Electronic Commerce Technologies was founded in 1990, and is located in Sunnyvale, California. EPR holds four issued Patents and several Trade Secrets on information metering and digital rights protection based upon pioneering work started in the mid 80's. EPR is a privately held corporation dedicated to the development of OEM technologies relating to Electronic Commerce.

National Semiconductor Corporation provides semiconductor technologies for moving and shaping information. The company focuses on three key areas - communications, analog-intensive applications, and personal systems. Principal market segments are computing, communications, automotive, military-aerospace, and industrial systems. National Semiconductor is headquartered in Santa Clara, California, and has 23,400 employees worldwide. Principal manufacturing sites are in Arlington, Texas; South Portland, Maine; and Greenock, Scotland. In fiscal 1993 the company surpassed the \$2.0 billion mark and achieved the highest earnings in its history.

###

[Back to EPR Home Page](#)

Find out more about EPR at info@epr.com

Copyright © 1994,1995 Electronic Publishing Resources, Incorporated. All rights reserved.

InterTrust, DigiBox, NetTrust, EPR, and the EPR logo are trademarks of Electronic Publishing Resources, Inc. Other trademarks and tradenames are the property of their respective holders.

The technology described herein may be covered by one or more of the following EPR U.S. patents: 4,827,508; 4,977,594; 5,050,213; and 5,410,598. Additional patents are pending.

BEST AVAILABLE COPY

Petitioner Apple Inc. - Exhibit 1012, p. 1000