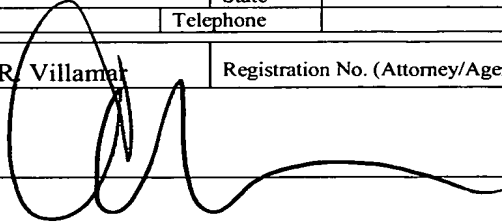


17712 U.S. PTO
121605

UTILITY PATENT APPLICATION TRANSMITTAL <i>(Only for new nonprovisional applications under 37 CFR 1.53(b))</i>		Attorney Docket No. 111325-028700																					
		First Inventor Mark J. STEFIK, et al.																					
		Title SYSTEM FOR CONTROLLING THE DISTRIBUTION AND USE OF DIGITAL WORKS																					
		Express Mail Label No. _____																					
APPLICATION ELEMENTS <i>See MPEP chapter 600 concerning utility patent application contents.</i>		ADDRESS TO: Commissioner for Patents P.O. Box 1450 Alexandria, VA 22313-1450																					
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. <i>See 37 CFR 1.27.</i> 3. ☒ Specification [Total Pages 87] <i>Both the claims and abstract must start on a new page (For information on the preferred arrangement, see MPEP 608.01(a))</i> 4. ☒ Drawing(s) (35 U.S.C. 113) [Total Sheets 13] 5. Oath or Declaration [Total Pages 2] a. ☐ Newly executed (original or copy) b. ☒ A 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) ☐ Landscape Table on CD 8. Nucleotide and/or Amino Acid Sequence Submission <i>(if applicable, items a. - c. are required)</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		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="2" style="text-align: center;">ACCOMPANYING APPLICATION PARTS</th> </tr> <tr> <td colspan="2"> 9. ☐ Assignment Papers (cover sheet & document(s)) Name of Assignee _____ </td> </tr> <tr> <td> 10. ☐ 37 CFR 3.73(b) Statement <i>(when there is an assignee)</i> </td> <td> ☐ Power of Attorney </td> </tr> <tr> <td colspan="2"> 11. ☐ English Translation Document (if applicable) </td> </tr> <tr> <td colspan="2"> 12. ☒ Information Disclosure Statement (PTO/SB/08 or PTO-1449) ☐ Copies of citations attached </td> </tr> <tr> <td colspan="2"> 13. ☐ Preliminary Amendment </td> </tr> <tr> <td colspan="2"> 14. ☒ Return Receipt Postcard (MPEP 503) <i>(Should be specifically itemized)</i> </td> </tr> <tr> <td colspan="2"> 15. ☐ Certified Copy of Priority Document(s) <i>(if foreign priority is claimed)</i> </td> </tr> <tr> <td colspan="2"> 16. ☐ Nonpublication Request under 35 U.S.C. 122(b)(2)(B)(i). <i>Applicant must attach form PTO/SB/35 or its equivalent.</i> </td> </tr> <tr> <td colspan="2"> 17. ☐ Other: _____ </td> </tr> </table>		ACCOMPANYING APPLICATION PARTS		9. ☐ Assignment Papers (cover sheet & document(s)) Name of Assignee _____		10. ☐ 37 CFR 3.73(b) Statement <i>(when there is an assignee)</i>	☐ Power of Attorney	11. ☐ English Translation Document (if applicable)		12. ☒ Information Disclosure Statement (PTO/SB/08 or PTO-1449) ☐ Copies of citations attached		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). <i>Applicant must attach form PTO/SB/35 or its equivalent.</i>		17. ☐ Other: _____	
ACCOMPANYING APPLICATION PARTS																							
9. ☐ Assignment Papers (cover sheet & document(s)) Name of Assignee _____																							
10. ☐ 37 CFR 3.73(b) Statement <i>(when there is an assignee)</i>	☐ Power of Attorney																						
11. ☐ English Translation Document (if applicable)																							
12. ☒ Information Disclosure Statement (PTO/SB/08 or PTO-1449) ☐ Copies of citations attached																							
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). <i>Applicant must attach form PTO/SB/35 or its equivalent.</i>																							
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.: 11/198,216 Filed August 8, 2005 Prior application information: Examiner James Reagan Group / Art Unit: 3621 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 _____																							
Address _____																							
City _____		State _____ Zip Code _____																					
Country _____		Telephone _____ Fax _____																					
Name (Print/Type) Carlos R. Villamar		Registration No. (Attorney/Agent) 43,224																					
Signature 		Date December 16, 2005																					

FEE TRANSMITTAL FOR FY 2005

Applicant claims small entity status. See 37 CFR 1.27

AL AMOUNT OF PAYMENT \$1,850.00

Complete if Known

Application Number	Not Yet Assigned
Filing Date	December 16, 2005
First Named Inventor	Mark J. STEFIK, et al.
Examiner	James Reagan
Art Unit	3621
Attorney Docket No.	111325-028700

METHOD OF PAYMENT (check all that apply)

☐ Check ☐ Credit Card ☐ Money Order ☐ None ☐ Other (please identify): _____

☒ Deposit Account Deposit Account Number: 19-2380 Deposit Account Name: Nixon Peabody LLP

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

☒ Charge fee(s) indicated below ☐ Charge fee(s) indicated below, except for the filing fee

☒ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 ☒ Credit any overpayments

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-20238.

FEE CALCULATION

1. BASIC FILING, SEARCH AND EXAMINATION FEES

Application Type	FILING FEES		SEARCH FEES		EXAMINATION FEES		Fees Paid (\$)
	Fee (\$)	Small Entity Fee (\$)	Fee (\$)	Small Entity Fee (\$)	Fee (\$)	Small Entity Fee (\$)	
Utility	300	150	500	250	200	100	\$1000.00
Design	200	100	100	50	130	65	
Plant	200	100	300	150	160	80	
Reissue	300	150	500	250	600	300	
Provisional	200	100	0	0	0	0	

2. EXCESS CLAIM FEES

Fee Description	Fee (\$)	Small Entity Fee (\$)
Each claim over 20 or, for Reissues, each claim over 20 and more than in the original patent	50	25
Each independent claim over 3 or, for Reissues, each independent claim more than in the original patent	200	100
Multiple document claims	360	180

Total Claims 37 - 20 or HP = 17 **Extra Claims** **Fee (\$)** \$50 **Fee Paid (\$)** \$850.00 **Multiple Dependent Claims** **Fee (\$)** \$360 **Fee Paid (\$)** 0

Indep. Claims 3 - 3 or HP = 0 **Extra Claims** **Fee (\$)** \$200 **Fee Paid (\$)** 0

HP = highest number of independent claims paid for, if greater than 3

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$250 (\$125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

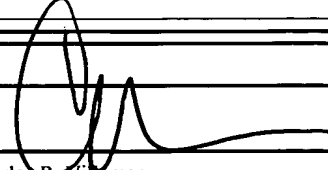
Total Sheets 100 - 100 = 0 **Extra Sheets** **Number of each additional 50 or fraction thereof** 0 (round up to a whole number) **Fee (\$)** \$250 **Fee Paid (\$)** 0

4. OTHER FEE(S)

Non-English Specification, \$130 fee (no small entity discount)

Other: _____

SUBMITTED BY

Signature		Registration No. 43,224 (Attorney/Agent)	Telephone (202) 585-8000
Name (Print/Type)	Carlos R. Villamar	Date	December 16, 2005

SEND TO: Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SYSTEM FOR CONTROLLING THE DISTRIBUTION AND USE OF DIGITAL WORKS

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation of United States Patent Application Serial No. 11/198,216 of *STEFIK, et al.*, filed on August 8, 2005, entitled "SYSTEM FOR CONTROLLING THE DISTRIBUTION AND USE OF DIGITAL WORKS," now pending, which is a continuation of United States Patent Application Serial No. 10/176,608 of *STEFIK, et al.*, filed on June 24, 2002, entitled "SYSTEM FOR CONTROLLING THE DISTRIBUTION AND USE OF DIGITAL WORKS," now United States Patent No. 6,934,693, which is a continuation of United States Patent Application Serial No. 09/777,845, filed on February 7, 2001, now pending, which is a divisional of United States Patent Application Serial No. 08/967,084, filed on November 10, 1997, now United States Patent No. 6,236,971, which is a continuation of United States Patent Application Serial No. 08/344,760, filed on November 23, 1994, now abandoned, the disclosures of all of which are hereby incorporated by reference herein.

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 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 its 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 schemes 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 usage 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, usage 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, Miss. 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 user's cartridge after a certain check-out time has expired.

However, individuals cannot loan books because the cartridges may only be used with the owner's 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 how the owner of a digital work may allow it to be distributed.

[0012] While flexibility in distribution 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 media. 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 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 for billing do not provide for billing of copies made of the work. For example, if data is copied from the CD-ROM described in Shear, any subsequent use of the copy of the information cannot be metered or billed. In other words, the means for billing runs with the media rather than the underlying work. It would be desirable to have a distribution system where the means for billing is always transported with the work.

SUMMARY OF THE INVENTION

[0016] A system for controlling the distribution and use of digital works using digital tickets is disclosed. A ticket is an indicator that the ticket holder has already paid for or is otherwise entitled to some specified right, product or service. In the present invention, a "digital ticket" is used to enable the ticket holder to exercise usage rights specifying the requirement of the digital ticket. Usage rights are used to define how a digital work may be used or distributed. Specific instances of usage rights are used to indicate a particular manner of use or distribution. A usage right may specify a digital ticket which must be present before the right may be exercised. For example, a digital ticket may be specified in a Copy right of a digital work, so that exercise of the Copy right requires the party that desires a copy of the digital work be in possession of the necessary digital ticket. After a copy of the digital work is successfully sent to the requesting party, the digital ticket is "punched" to indicate that a copy of the digital work has been made. When the ticket is "punched" a predetermined number of times, it may no longer be used.

[0017] Digital works are stored in repositories. Repositories enforce the usage rights for digital works. Each repository has a "generic ticket agent" which punches tickets. In some instances only the generic ticket agent is necessary. In other instances, punching by a "special ticket agent" residing on another repository may be desired. Punching by a "special ticket

agent” enables greater security and control of the digital work. For example, it can help prevent digital ticket forgery. Special ticket agents are also useful in situations where an external database needs to be updated or checked.

[0018] A digital ticket is merely an instance of a digital work. Thus, a digital ticket may be distributed among repositories in the same fashion as other digital works.

[0019] A digital ticket may be used in many commercial scenarios such as in the purchase of software and prepaid upgrades. A digital ticket may also be used to limit the number of times that a right may be exercised. For example, a user may purchase a copy of a digital work, along with the right to make up to 5 Copies. In this case, the Copy right would have associated therewith a digital ticket that can be punched up to 5 times. Other such commercial scenarios will become apparent from the detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

[0020] FIG. 1 is a flowchart illustrating a simple instantiation of the operation of the currently preferred embodiment of the present invention.

[0021] 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.

[0022] FIG. 3 is a block diagram of a repository coupled with a credit server in the currently preferred embodiment of the present invention.

[0023] FIGS. 4a and 4b are examples of rendering systems as may be utilized in the currently preferred embodiment of the present invention.

[0024] FIG. 5 illustrates a contents file layout for a digital work as may be utilized in the currently preferred embodiment of the present invention.

[0025] 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.

[0026] FIG. 7 illustrates the components of a description block of the currently preferred embodiment of the present invention.

[0027] FIG. 8 illustrates a description tree for the contents file layout of the digital work illustrated in FIG. 5.

[0028] FIG. 9 illustrates a portion of a description tree corresponding to the individual digital work illustrated in FIG. 6.

[0029] 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.

[0030] 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.

[0031] 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.

[0032] 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.

[0033] FIG. 14 is diagram illustrating the basic components of a usage right in the currently preferred embodiment of the present invention.

[0034] FIG. 15 lists the usage rights grammar of the currently preferred embodiment of the present invention.

[0035] 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.

[0036] 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.

[0037] 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.

[0038] 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

Overview

[0039] A system for controlling use and distribution of digital works is disclosed. The present invention 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 hand requires essentially no more space than keeping one copy on hand. The digital medium also lowers the costs of alteration, reuse and billing.

[0040] 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.

[0041] Herein the terms “digital work”, “work” and “content” refer 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 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 how a digital work can be used and if it can be further distributed. Each usage right may have one or more specified conditions which must be satisfied before the right may be exercised. Appendix 1 provides a Glossary of the terms used herein.

[0042] A key feature of the present invention is that usage rights are permanently “attached” to the digital work. Copies made of a digital work will also have usage rights

attached. Thus, the usage rights and any associated fees assigned by a creator and subsequent distributor will always remain with a digital work.

[0043] The enforcement elements of the present invention are embodied in 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.

[0044] The combination of attached 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.

[0045] 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, attach them to 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 help 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, 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 attached to 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.

[0046] 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 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.

[0047] 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.

[0048] Communication with a rendering repository 203 occurs in connection with the rendering 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.

[0049] 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.

[0050] 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.

RENDERING SYSTEMS

[0051] A rendering system is generally defined as a system comprising a repository and a rendering device which can render 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.

[0052] 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.

[0053] 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.

[0054] 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 its 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 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.

STRUCTURE OF DIGITAL WORKS

[0055] Usage rights are attached directly to 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.

[0056] 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 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.

[0057] 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 60,001-85K. 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 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).

[0058] 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 are 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.

[0059] 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.

[0060] 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).

[0061] 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.

[0062] 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 1001 and status information field 1002. The right code field 1001 will contain a unique code assigned to a right. The status information field 1002 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.

[0063] 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.

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 repostories and dates for operations that copy, transfer, backup, or restore a digital work.

TABLE 1
DIGITAL WORK STATE INFORMATION

[0064] 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 attached to 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 attached to the folder which defines how folder contents can be managed.

ATTACHING USAGE RIGHTS TO A DIGITAL WORK

[0065] It is fundamental to the present invention that 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.

[0066] The attachment of usage rights into a digital work may occur in a variety of ways. If the usage rights will be the same for an entire digital work, they could be attached 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.

[0067] 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 attached to the digital work as it is transported.

Resolving Conflicting Rights

[0068] 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.

[0069] 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.

[0070] 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.

[0071] 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.

[0072] 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.

REPOSITORIES

[0073] 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.

[0074] 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.

[0075] Communications integrity refers to the integrity of the communications 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.

[0076] 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.

[0077] 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.

[0078] 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. 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.

[0079] 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.

[0080] 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.

[0081] 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.

[0082] 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.

[0083] 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 uninterruptible 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.

[0084] 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.

[0085] 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.

[0086] 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.

[0087] 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.

Repository Security Classes

[0088] 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.

[0089] 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.

[0090] 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.

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	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	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.

TABLE 2
REPOSITORY SECURITY LEVELS

Level	Description of Security
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.

TABLE 2
REPOSITORY SECURITY LEVELS

[0091] 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.

Repository User Interface

[0092] 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.

[0093] 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.

[0094] 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.

CREDIT SERVERS

[0095] In the present invention, 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.

[0096] 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.

[0097] 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.

[0098] 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 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.

USAGE RIGHTS LANGUAGE

[0099] The present invention 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 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.

[00100] 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 attach multiple versions of a right to a work. So a creator may attach a 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.

[00101] 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 use or distribution privileges 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 use or distribution privileges for the right. The 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.

[00102] 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 | b | 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.

[00103] 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.

[00104] 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.

[00105] 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.

[00106] 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.”

[00107] The Usage Rights Grammar is listed in its entirety in FIG. 15 and is described below.

[00108] Grammar element 1501 “Digital Work Rights:=(Rights*)” define the digital work rights as a set of rights. The set of rights attached to 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.

[00109] 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.

[00110] It is important to note that a digital work may have multiple versions of a right, each having the same right code. The multiple versions would provide alternative conditions and fees for accessing the digital work.

[00111] A Grammar element 1503 “Right-Code:=Render-Code | Transport-- Code | File-Management-Code | 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.

[00112] Grammar element 1504 “Render-Code:=[Play: {Player: Player-ID} | Print: {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.

- 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.

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

[00113] Grammar element 1505 “Transport-Code:=[Copy | Transfer | 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.

- Copy: Make a new copy of a work
- Transfer: Moving a work from one repository to another.
- Loan: Temporarily loaning a copy to another repository for a specified period of time.

[00114] Grammar element 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}” 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.

[00115] 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.

[00116] 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.

- Backup: To make a backup copy of a digital work as protection against media failure.

- Restore: To restore a backup copy of a digital work.
- Delete: To delete or erase a copy of a digital work.
- Folder: To create and name folders, and to move files and folders between folders.
- Directory: To hide a folder or its contents.

[00117] Grammar element 1507 “Derivative-Works-Code: [Extract | Embed | 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.

- Extract: To remove a portion of a work, for the purposes of creating a new work.
- Embed: To include a work in an existing work.
- Edit: To alter a digital work by copying, selecting and modifying portions of an existing digital work.

[00118] Grammar element 1508 “Configuration-Code:=Install | 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.

- Install: To install new software on a repository.
- Uninstall: To remove existing software from a repository.

[00119] 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.

[00120] 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.

Copy Count Specification

[00121] For various transactions, it may be desirable to provide some limit as to the number of “copies” of the work which may be exercised 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.

[00122] Grammar element 1510 “Copy-Count:=(Copies: positive-integer | 0 | 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.

Control Specification

[00123] 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.

[00124] Grammar element 1511 “Control-Spec:=(Control: {Restrictable | Unrestrictable} {Unchargeable | 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.

Time Specification

[00125] 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 | Sliding-Interval | Meter-Tim- e} 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 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.

[00126] 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.

[00127] 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.

[00128] 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.

[00129] 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

[00130] All of the time specifications include time-unit specifications in their ultimate instantiation.

Security Class and Authorization Specification

[00131] 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.

[00132] 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.

[00133] 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.

[00134] 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.

[00135] 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 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.

[00136] 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.

[00137] 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.

[00138] 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 attached thereto.

[00139] The capability to unpunch a ticket is important in the following cases:

- A digital work is circulated at low cost with a limitation that it can be used only once.
- A digital work is circulated with a ticket that can be used once to give discounts on purchases of other works.
- 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.

[00140] 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.

Usage Fees and Incentives Specification

[00141] The billing for use of a digital work is fundamental to a commercial distribution system. Grammar Element 1517 “Fee-Spec:={Scheduled-Discount} Regular-Fee-Spec | Scheduled-Fee-Spec | Markup-Spec” provides a range of options for billing for the use of digital works.

[00142] 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.

[00143] 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.

[00144] 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.

[00145] Grammar Element 1518 “Scheduled-Discount:=(Scheduled-Discount: (Time-Spec Percentage)*)” A Scheduled-Discount is 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.

[00146] Grammar Element 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)” provides for several kinds of fee specifications.

[00147] 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.

[00148] 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.

[00149] 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.

[00150] 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.

[00151] 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.

[00152] 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.

[00153] 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.

Examples of Sets of Usage Rights

- ((Play) (Transfer (SC: 3)) (Delete))

[00154] 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.

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

[00155] 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.

- ((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)))

[00156] 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.

- ((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)))

[00157] 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.

- ((Play) (Transfer (SC: 3))

(Copy Authorization: License-123-ID (SC: 3)))

[00158] 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.

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

[00159] 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.

- ((Play Player: Player-876-ID) (From: Feb. 2, 1994 Until: Feb. 15, 1995) (Fee: Metered: \$0.01 Per: 0:1:0 Min: \$0.25 Per: 0/1/0 To: Account-ID-567))

[00160] 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.

- ((Play) (Transfer) (Delete)(Loan 2 (Delete: Transfer Loan)))

[00161] 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.

- [00162] • ((Play) (Transfer) (Delete) (Backup) (Restore (SC:3))

(Loan 2 Remaining-Copy-Rights: (Delete: Play Transfer)

Next-Set-of-Rights: (Delete: Transfer Loan)))

[00163] 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.

- ((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: Feb. 14, 1994 Until: Feb. 15, 1995)

(Fee: Metered: \$0.01 Per: 0:1:0 Min: \$0.25 Per: 0/1/0

To: Account-ID-567))))

[00164] 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 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.

- ((Play Player: Player-Small-Screen-123-ID)

(Embed (Fee: Per-Use \$0.01 To: Account-678-ID))

(Copy (Fee: Per-Use \$1.00 To: Account-678-ID)))

[00165] 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.

REPOSITORY TRANSACTIONS

[00166] 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.

[00167] 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.

Message Transmission

[00168] 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 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.

[00169] 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.

[00170] 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.

[00171] 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.

[00172] The respective public keys for the repositories to be used for encryption are obtained in the registration transaction described below.

Session Initiation Transactions

[00173] 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 others identities. As it is assumed that the communication channel between the repositories is reliable but not secure, there is a risk that a non-repository may mimic the protocol in order to gain illegitimate access to a repository.

[00174] 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.

[00175] 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.

[00176] 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.

[00177] 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.

[00178] 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 1618. If they are the same, the registration transaction has successfully completed.

[00179] 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.

[00180] 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.

[00181] To achieve greater accuracy, repository-2 can request the time again up to a fixed number of times (e.g. five times), repeat the clock synchronization steps, and average the results.

[00182] 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.

Billing Transactions

[00183] 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:

- 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.

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

- 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.

- A Begin-charges transaction to assign a charge. This transaction is much the same as an assign-fee transaction except that it is used for metered use. It includes the same information as the assign-fee 4, ii transaction as well as the usage fee information. The credit-server is then responsible for running a clock.

- 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.)

- 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.

[00184] 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.

Usage Transactions

[00185] 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.

[00186] 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.

[00187] 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.

[00188] 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.

[00189] 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.

[00190] 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.

[00191] 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.

[00192] 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 are 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.

[00193] 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 are determined from the remaining rights specifications from the versions of the 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.

[00194] 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.

[00195] It should be noted that the order in which the conditions are checked need not follow the order of steps 1806-1815.

[00196] 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.

[00197] 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.

Transmission Protocol

[00198] 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 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.

[00199] 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.

[00200] 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.

[00201] 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.

[00202] 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.

[00203] 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.

[00204] 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.

[00205] 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.

[00206] 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

[00207] 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.

[00208] 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.

[00209] 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.

[00210] 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.

[00211] 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.

[00212] 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.

[00213] 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 variation adds a level of encryption for no real gain in accountability.

[00214] The transactions for specific usage rights are now discussed.

The Copy Transaction

[00215] 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.

- 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.

- 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.

- 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.

- The repositories perform the common closing transaction steps.

The Transfer Transaction

[00216] 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.

- 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.

- 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 is set to the number-of-copies requested.
- The requester records the work contents, data, and usage rights and stores the work.
- The server decrements its copy count by the number of copies involved in the transaction.
- The repositories perform the common closing transaction steps.
- If the number of copies remaining in the server is now zero, it erases the digital work from its memory.

The Loan Transaction

[00217] 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.

- 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.
- 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 loan to the server.
- 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.
- The requester records the digital work contents, data, usage rights, and loan period and stores the work.

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

- 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.

[00218] Case 1. If the time of the loan period is not yet exhausted and the requester sends the repository a Return message.

- 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.

- The requester deactivates its copies and removes the contents from its memory.

[00219] 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 works that were borrowed.

- 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.

The Play Transaction

[00220] 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.

[00221] 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.

- 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.

- 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.

- 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.

- When the player is finished, the player and the requester remove the contents from their memory.

- The repositories perform the common closing transaction steps.

The Print Transaction

[00222] 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.

[00223] 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.

- 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.

- 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.

- The repositories perform the common opening transaction steps.
- The server transmits blocks of data according to the transmission protocol.
- The requester prints the work contents, using the printer.
- When the printer is finished, the printer and the requester remove the contents from their memory.

- The repositories perform the common closing transaction steps.

The Backup Transaction

[00224] 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.

[00225] 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.

[00226] 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.

- 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.

- 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.

- 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.

- The repositories perform the common closing transaction steps.

[00227] 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.

The Restore Transaction

[00228] 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.

- 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.

- 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.

- The repositories perform the common opening transaction steps.
- The server retrieves the key from the restoration file. It decrypts the work contents, data, and usage rights.

- 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.

- The requester stores the digital work.
- The repositories perform the common closing transaction steps.

The Delete Transaction

[00229] 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.

- 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.

- The repositories perform the common opening transaction steps.
- The server deletes the file, erasing it from the file system.
- The repositories perform the common closing transaction steps.

The Directory Transaction

[00230] 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.

[00231] 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 are exercised as part of the normal process of exercising all rights.

- 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.

- 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.

- The repositories perform the common opening transaction steps.

- The server sends the requested data to the requester according to the transmission protocol.

- The requester records the data.

- The repositories perform the common closing transaction steps.

The Folder Transaction

[00232] 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.

- 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.

- 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.

- The repositories perform the common closing transaction steps.

The Extract Transaction

[00233] 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.

- 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.

- 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.

- 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.

- The repositories perform the common closing transaction steps.

The Embed Transaction

[00234] 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.

- 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.

- The server checks the control specifications for all of the rights in the part and the destination. If they are incompatible, the server ends the transaction with an error.

- 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.

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

- The repositories perform the common closing transaction steps.

The Edit Transaction

[00235] An Edit transaction is a request to make a new 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.

- 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.

- 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.

- 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.)

- The repositories perform the common closing transaction steps.

[00236] 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.

[00237] 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.

Authorization Transactions

[00238] 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.

[00239] 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 must process. The authorization is given to the generic authorization (or ticket) server of the repository which begins to interpret the authorization.

[00240] 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.

[00241] If a remote authorization server is required, then the authorization process first performs the following steps:

- The generic authorization server attempts to set up the communications channel. (If the channel cannot be set up, then authorization fails with an error.)
- 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.)
- 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.)
- 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.
- If all of the required steps are completed satisfactorily, the authorization server completes the transaction normally, signaling that authorization is granted.

The Install Transaction .

[00242] 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.

- 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).

- 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.
- 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.)
- 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.)
- 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.)
- 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.

- The repositories perform the common closing transaction steps.

The Uninstall Transaction

[00243] 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.

- 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).

- 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.
- The requester checks whether the software is installed. If the software is not installed, the transaction ends with an error.
- 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.)
- 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.)
- 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.
- The repositories perform the common closing transaction steps.

DISTRIBUTION AND USE SCENARIOS

[00244] 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.

Consumers as Unpaid Distributors

[00245] 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.

[00246] 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.

[00247] This scenario is performed as follows:

[00248] 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.

Paid Distributors

[00249] 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.

[00250] 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.

[00251] This scenario is performed as follows:

[00252] 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.

[00253] 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.

[00254] 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.

Licensed Distribution

[00255] 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.

[00256] 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.

[00257] This scenario is performed as follows:

[00258] 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.

[00259] 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.

[00260] A distributor obtains the distribution license and a number of copies of the work. He makes copies for his customers, using his distribution license.

[00261] A customer buys and uses the work. He cannot make new copies because he lacks a distribution license.

Super Distributors

[00262] 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.

[00263] 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.

[00264] 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. 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.

1-Level Distribution Fees

[00265] 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.

[00266] This scenario pays a distributor only for use of copies that he actually sold.

[00267] 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.

Distribution Trees

[00268] 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.

[00269] 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.

[00270] 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.

Weighted Distribution Trees

[00271] 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.

[00272] 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.

[00273] 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.

Fees for Reuse

[00274] 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.

[00275] 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.

[00276] The mechanisms for derived works are the same as those for distribution.

Limited Reuse

[00277] 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.

[00278] This scenario deals with fine grained control of the rights and fees for reuse.

[00279] This scenario is performed as follows:

[00280] The first creators create their original 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.

Commercial Libraries

[00281] 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.

[00282] 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.

Demo Versions

[00283] 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. This scenario deals with fees for loaning rather than fees for making copies. The fees are collected by the same automatic mechanisms.

[00284] This scenario is performed as follows:

[00285] 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.

Upgrading a Digital Work with a Vendor

[00286] 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.

[00287] This scenario deals with a common situation in computer software. It shows how 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.

[00288] This scenario is performed as follows:

[00289] 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.

[00290] The distributor sells the old bundled work and ticket to customers.

[00291] The customer extracts the work and the ticket. He uses the work according to the agreements until the new version becomes available.

[00292] 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.

[00293] 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 consumer's repository.

[00294] The consumer can now use the new version of the work.

Distributed Upgrading of Digital Works

[00295] 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.

[00296] 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.

[00297] This scenario is similar to the previous one except that the Copy right on the new 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.

Limited Printing

[00298] 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.

[00299] 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.

[00300] This scenario is performed as follows:

[00301] 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.

Demand Publishing

[00302] 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.

[00303] 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.

[00304] 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.

Metered Use and Multiple Price Packages

[00305] 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.

[00306] 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.

Fees for Font Usage

[00307] 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.

[00308] This scenario is performed as follows: the font designer creates a font as a digital 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.

Rational Database Usage Charges

[00309] 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.

[00310] This scenario is performed as follows:

[00311] 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.

Print Spooling with Rights

[00312] 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.

[00313] 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 is about to be printed.

[00314] Thus, a system for enforcing the usage rights of digital works is disclosed. While the embodiments disclosed herein are preferred, it will be appreciate 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 following claims.

APPENDIX A

GLOSSARY

AUTHORIZATION REPOSITORY:

[00315] 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 the right may be exercised.

BILLING CLEARINGHOUSE:

[00316] 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.

BILLING TRANSACTIONS:

[00317] The protocol used by which a repository reports billing information to a credit server.

CLEARINGHOUSE TRANSACTIONS:

[00318] The protocol used between a credit server and a clearinghouse.

COMPOSITE DIGITAL WORK:

[00319] A digital work comprised of distinguishable parts. Each of the distinguishable parts is itself a digital work which has usage rights attached.

CONTENT:

[00320] The digital information (i.e. raw bits) representing a digital work.

COPY OWNER:

[00321] 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 printing, viewing, transferring, or other specific uses.

CREATOR:

[00322] A term which refers to a party who produces a digital work.

CREDIT SERVER:

[00323] 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.

DESCRIPTION TREE:

[00324] 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.

DIGITAL WORK (WORK):

[00325] Any encapsulated digital information. Such digital information may represent music, a magazine or book, or a multimedia composition. Usage rights and fees are attached to the digital work.

DISTRIBUTOR:

[00326] A term which refers to a party who legitimately obtains a copy of a digital work and offers it for sale.

IDENTIFICATION (DIGITAL) CERTIFICATE:

[00327] 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.

MASTER REPOSITORY:

[00328] 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".)

PUBLIC KEY ENCRYPTION:

[00329] 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.

REGISTRATION TRANSACTIONS:

[00330] The protocol used between repositories to establish a trusted session.

RENDERING REPOSITORY:

[00331] 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.

RENDERING SYSTEM:

[00332] 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.

REPOSITORY:

[00333] 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.

REQUESTER MODE:

[00334] A mode of a repository where it is requesting access to a digital work.

REVENUE OWNERS:

[00335] 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.

SERVER MODE:

[00336] A mode of a repository where it is processing an incoming request to access a digital work.

SHELL DESCRIPTION BLOCK:

[00337] 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:

[00338] A term used to refer to the protocols by which repositories communicate.

USAGE FEES:

[00339] A fee charged to a requester for access to a digital work. Usage fees are specified within the usage rights language.

USAGE RIGHTS:

[00340] A language for defining the manner in which a digital work may be used or distributed, as well as any conditions on which use or distribution is premised.

USAGE TRANSACTIONS:

[00341] A set of protocols by which repositories communicate in the exercise of a usage rights. Each usage right has it's own transaction steps.

WHAT IS CLAIMED:

1. A method for securely rendering digital documents, comprising:
storing a digital document in a document platform;
storing at least one usage right associated with the digital document, the at least one usage right specifying a manner of use indicating the manner in which the digital document can be rendered by the document platform;
determining whether the digital document may be rendered by the document platform based on at least the at least one usage right; and
if the at least one usage right allows the digital document to be rendered on the document platform, rendering the digital document by the document platform.
2. The method of claim 1, wherein the manner of use includes the number of times the digital document can be rendered.
3. The method as recited in claim 1, wherein at least one part of the content and at least one usage right are included in different files.
4. The method as recited in claim 1, wherein at least one part of the content and at least one usage right are stored in the same file.
5. The method as recited in claim 1, wherein at least a portion of the digital content is a software program.
6. The method as recited in claim 1, wherein at least one usage right comprises a revenue identifier for identifying a revenue owner of the digital content.
7. The method as recited in claim 1, wherein at least one usage right also specifies one or more conditions which must be satisfied before the manner of rendering may be exercised.

8. The method as recited in claim 7, wherein at least one condition includes determining the presence of a digital ticket.

9. The method as recited in claim 1, wherein at least one of the usage rights or a part of the content are stored on a removable storage device.

10. The method as recited in claim 1, wherein at least one part of the content and at least one usage right are stored on the same device.

11. The method as recited in claim 1, wherein at least one of the content and at least one usage right are stored on different devices.

12. The method as recited in claim 1, wherein the method is implemented with a computer system including one or more hardware and/or software components.

13. The method as recited in claim 1, wherein the method is implemented via one or more computer readable instructions embedded on a computer readable medium and configured to cause one or more computer processors to perform the steps of the method.

14. A method for securely rendering digital documents, comprising:
storing a digital document in a document repository;
storing at least one usage right associated with the digital document;
requesting access to the digital document from a document platform;
determining whether the request may be granted based on the usage right, the determining step including authenticating the document platform and determining whether the usage right includes a manner of use that allows transfer of the digital document to the document platform;
if the usage right allows the transfer of the digital document to the document platform, transferring the digital document to the document platform; and
rendering the digital document by the document platform.

15. The method as recited in claim 14, wherein at least one part of the content and at least one usage right are included in different files.

16. The method as recited in claim 14, wherein at least one part of the content and at least one usage right are stored in the same file.

17. The method as recited in claim 14, wherein at least a portion of the digital content is a software program.

18. The method as recited in claim 14, wherein at least one usage right comprises a revenue identifier for identifying a revenue owner of the digital content.

19. The method as recited in claim 14, wherein at least one usage right also specifies one or more conditions which must be satisfied before the manner of rendering may be exercised.

20. The method as recited in claim 19, wherein at least one condition includes determining the presence of a digital ticket.

21. The method as recited in claim 14, wherein at least one of the usage rights or a part of the content are stored on a removable storage device.

22. The method as recited in claim 14, wherein at least one part of the content and at least one usage right are stored on the same device.

23. The method as recited in claim 14, wherein at least one of the content and at least one usage right are stored on different devices.

24. The method as recited in claim 14, wherein the method is implemented with a computer system including one or more hardware and/or software components.

25. The method as recited in claim 14, wherein the method is implemented via one or more computer readable instructions embedded on a computer readable medium and configured to cause one or more computer processors to perform the steps of the method.

26. A method for securely rendering digital documents, comprising:

receiving a request to transfer a digital document from a document repository to a document platform the digital document having at least one usage right associated therewith;

authenticating the document platform by accessing at least one identifier associated with the document platform or with a user of the document platform and determining whether the identifier is associated with at least one of a document platform and user authorized to use the digital document;

if the authenticating step is successful, determining whether the digital document may be transferred and stored on a document platform based on a manner of use included in the at least one usage right;

if the at least one usage right allows the digital document to be transferred to a document platform, transferring the digital document to the document platform; and

rendering the digital document by the document platform.

27. The method as recited in claim 26, wherein at least one part of the content and at least one usage right are included in different files.

28. The method as recited in claim 26, wherein at least one part of the content and at least one usage right are stored in the same file.

29. The method as recited in claim 26, wherein at least a portion of the digital content is a software program.

30. The method as recited in claim 26, wherein at least one usage right comprises a revenue identifier for identifying a revenue owner of the digital content.

31. The method as recited in claim 26, wherein at least one usage right also specifies one or more conditions which must be satisfied before the manner of rendering may be exercised.

32. The method as recited in claim 31, wherein at least one condition includes determining the presence of a digital ticket.

33. The method as recited in claim 26, wherein at least one of the usage rights or a part of the content are stored on a removable storage device.

34. The method as recited in claim 26, wherein at least one part of the content and at least one usage right are stored on the same device.

35. The method as recited in claim 26, wherein at least one of the content and at least one usage right are stored on different devices.

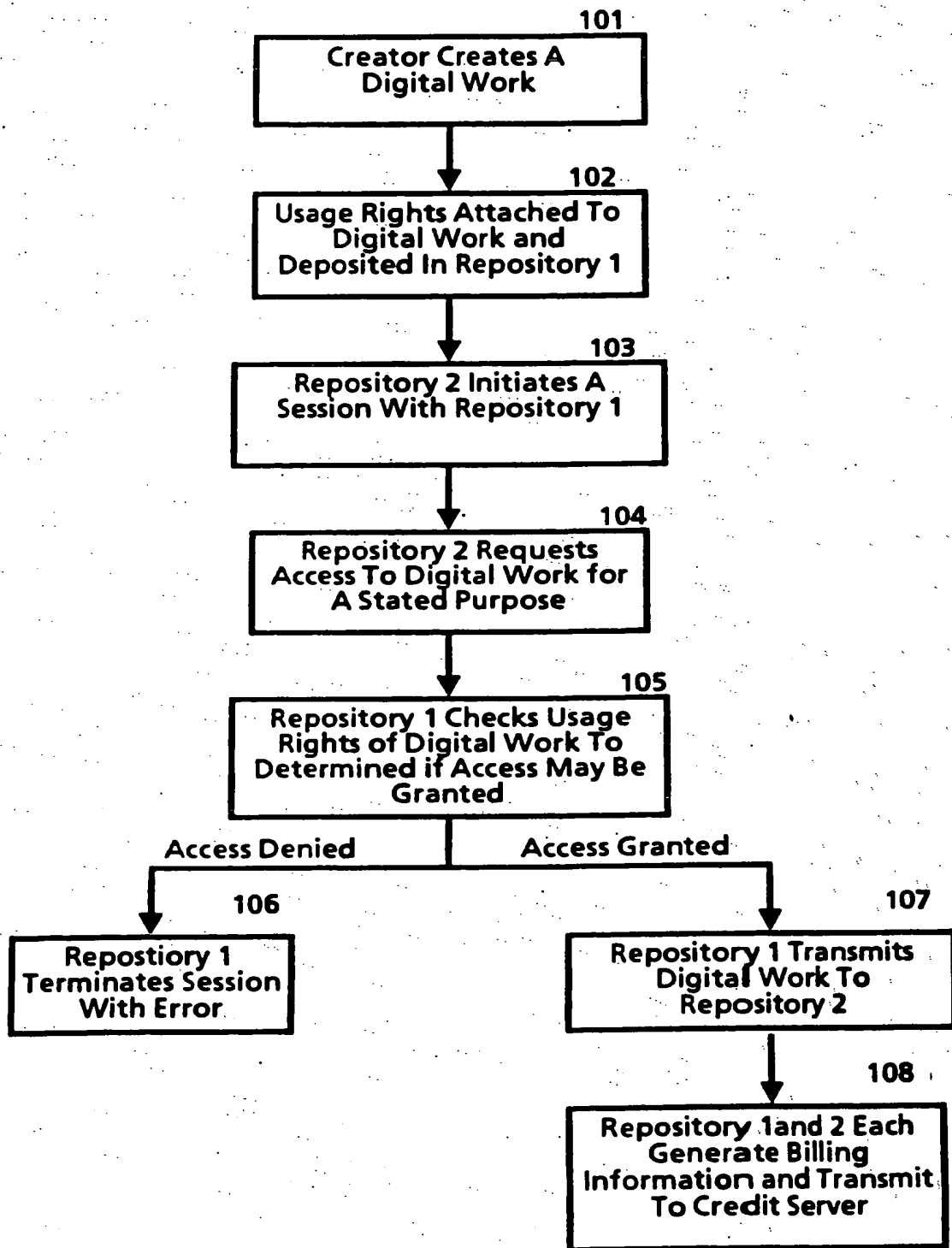
36. The method as recited in claim 26, wherein the method is implemented with a computer system including one or more hardware and/or software components.

37. The method as recited in claim 26, wherein the method is implemented via one or more computer readable instructions embedded on a computer readable medium and configured to cause one or more computer processors to perform the steps of the method.

ABSTRACT OF THE DISCLOSURE

A method, system and software for securely rendering digital documents, including storing a digital document in a document platform; and storing a usage right associated with the digital document. The usage right specifies a manner of use indicating the manner in which the digital document can be rendered by the document platform. The digital document comprises plural parts of digital content. The usage right includes plural usage rights respectively associated with each of the plural parts of digital content. Whether one of the parts of the digital document may be rendered by the document platform is determined based a respective usage right. If the respective usage right allows the digital document to be rendered on the document platform, the corresponding part of the digital document is rendered by the document platform.

Figure 1



BEST AVAILABLE COPY

Figure 2

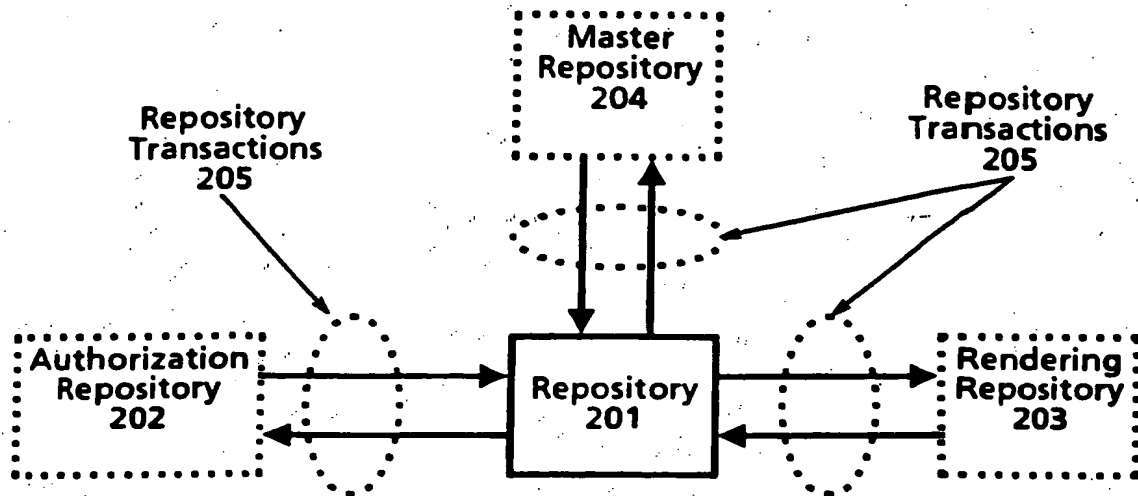


Figure 3

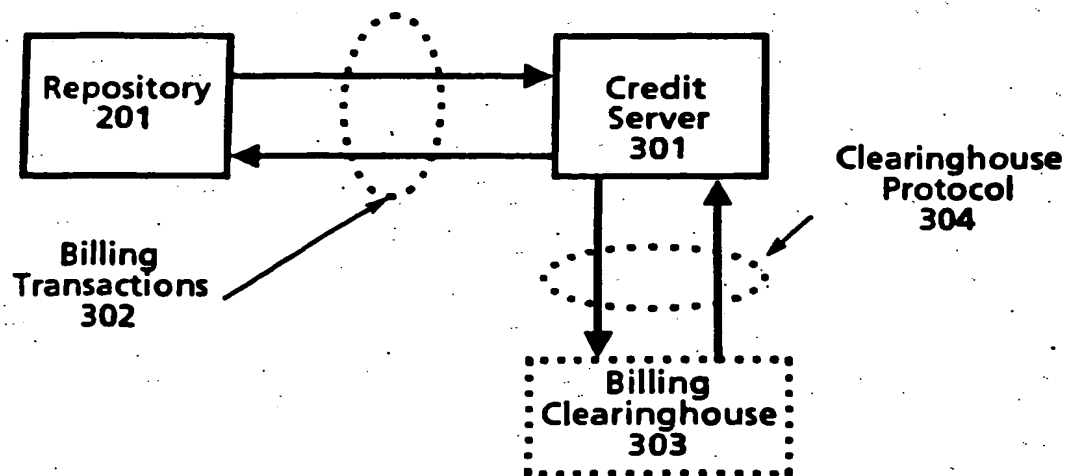


Figure 4a

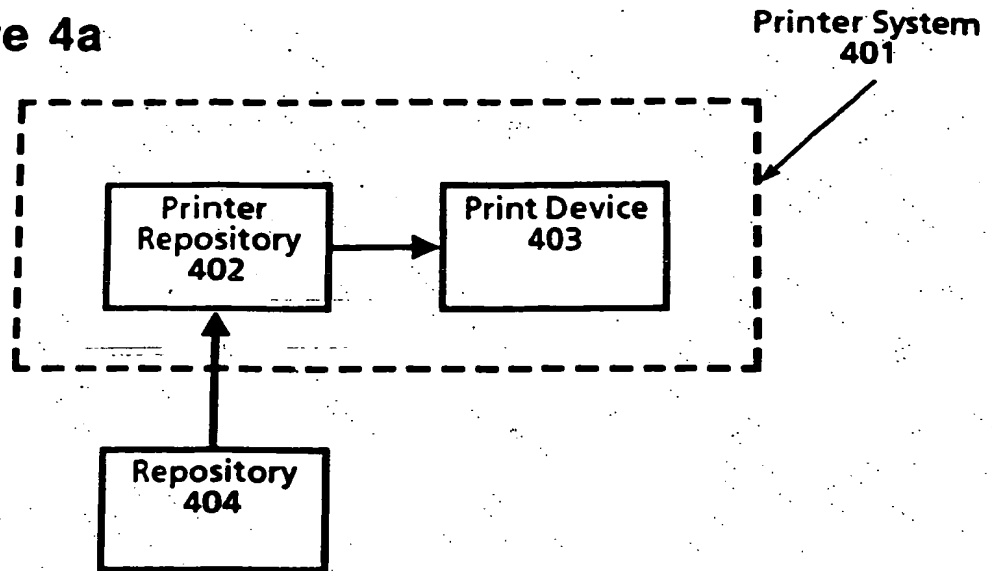
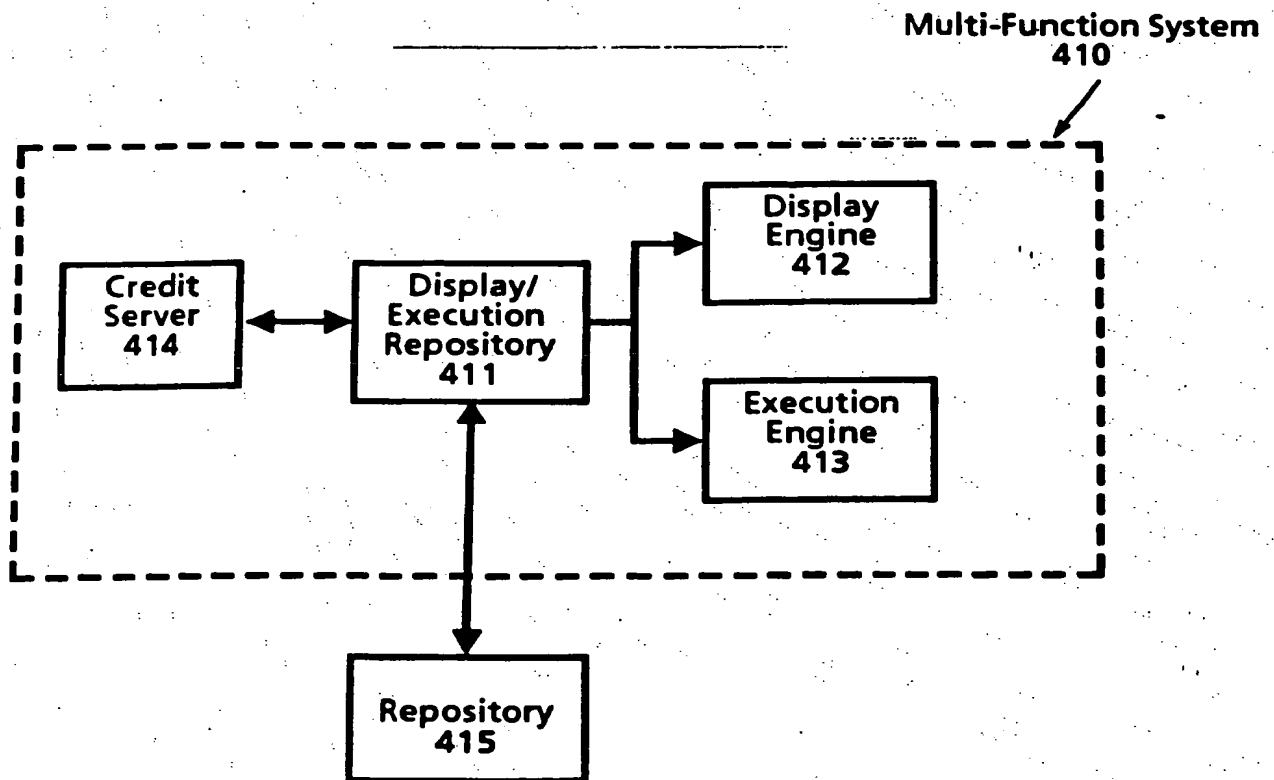


Figure 4b



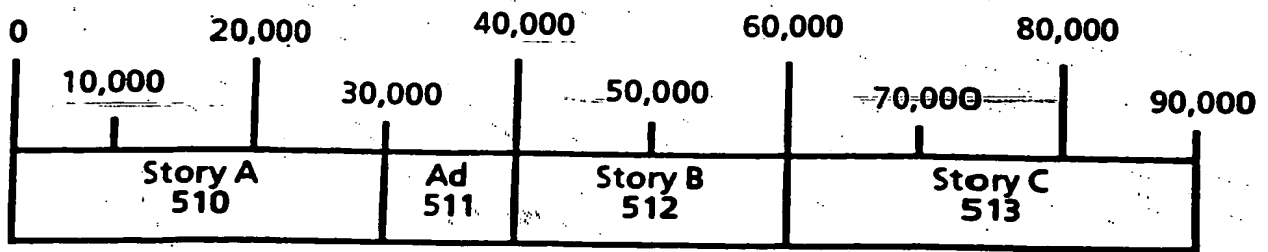


Figure 5

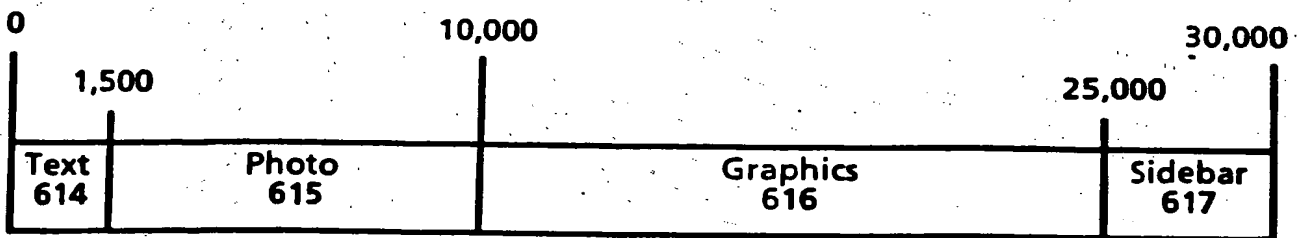


Figure 6

Figure 7

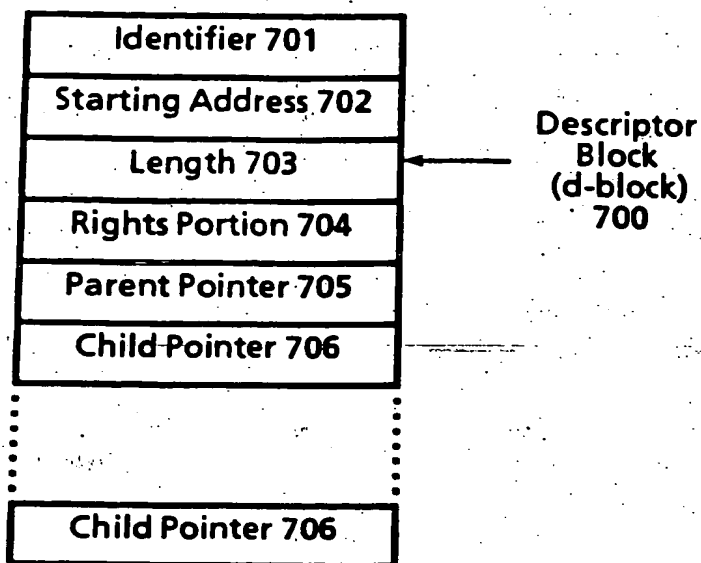


Figure 8

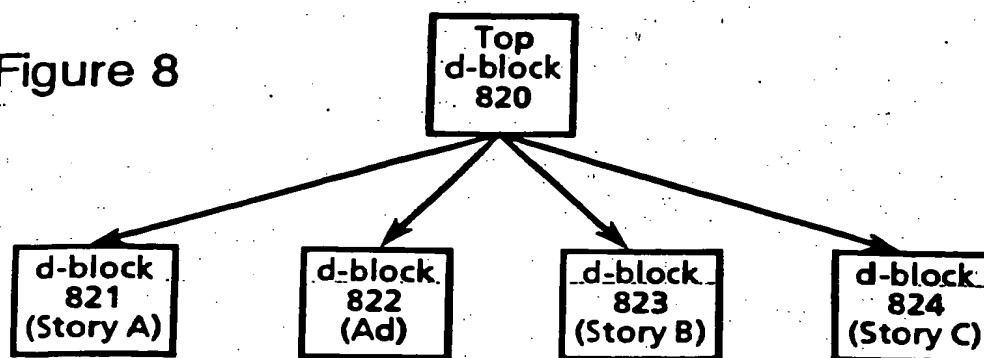


Figure 9

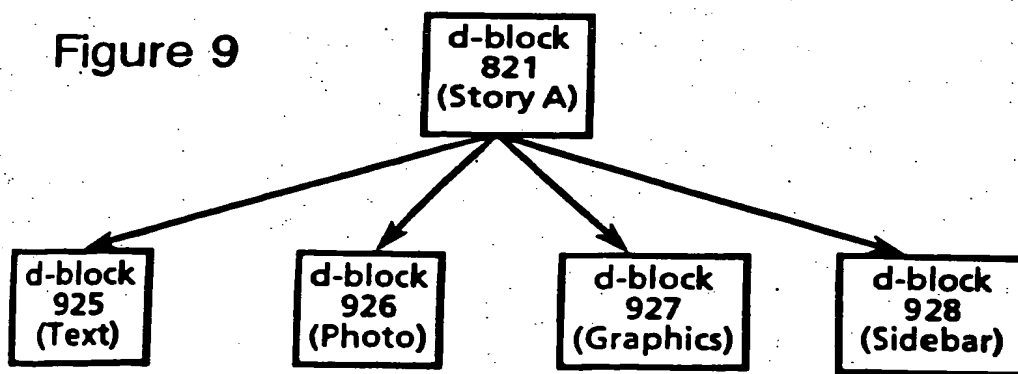


Figure 10

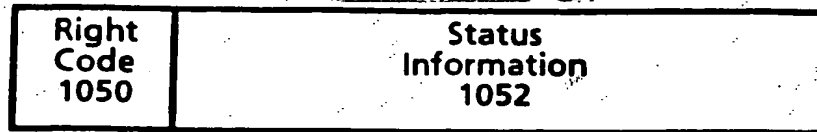


Figure 14

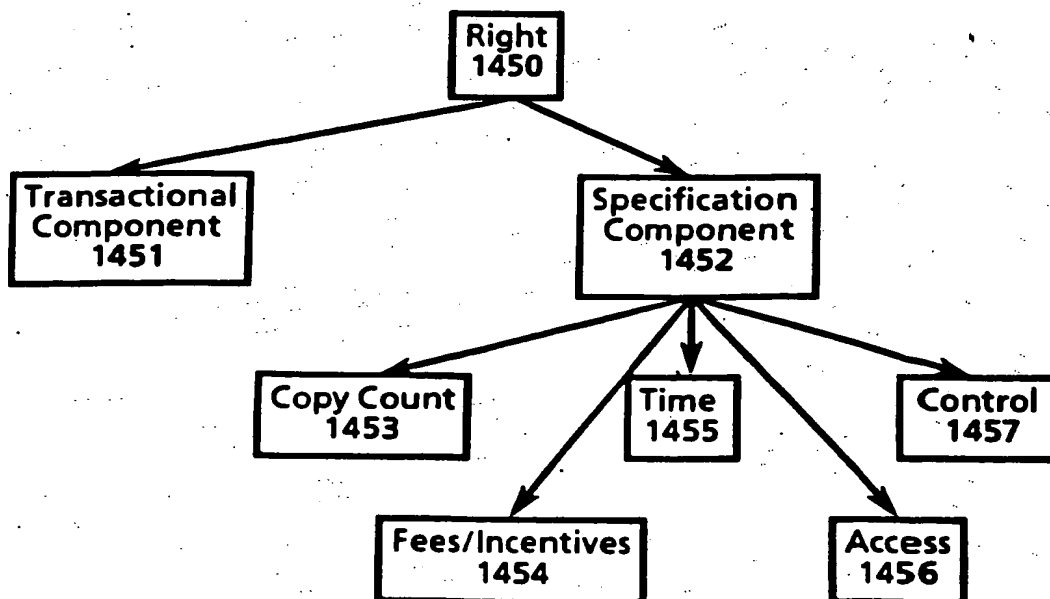


Figure 11

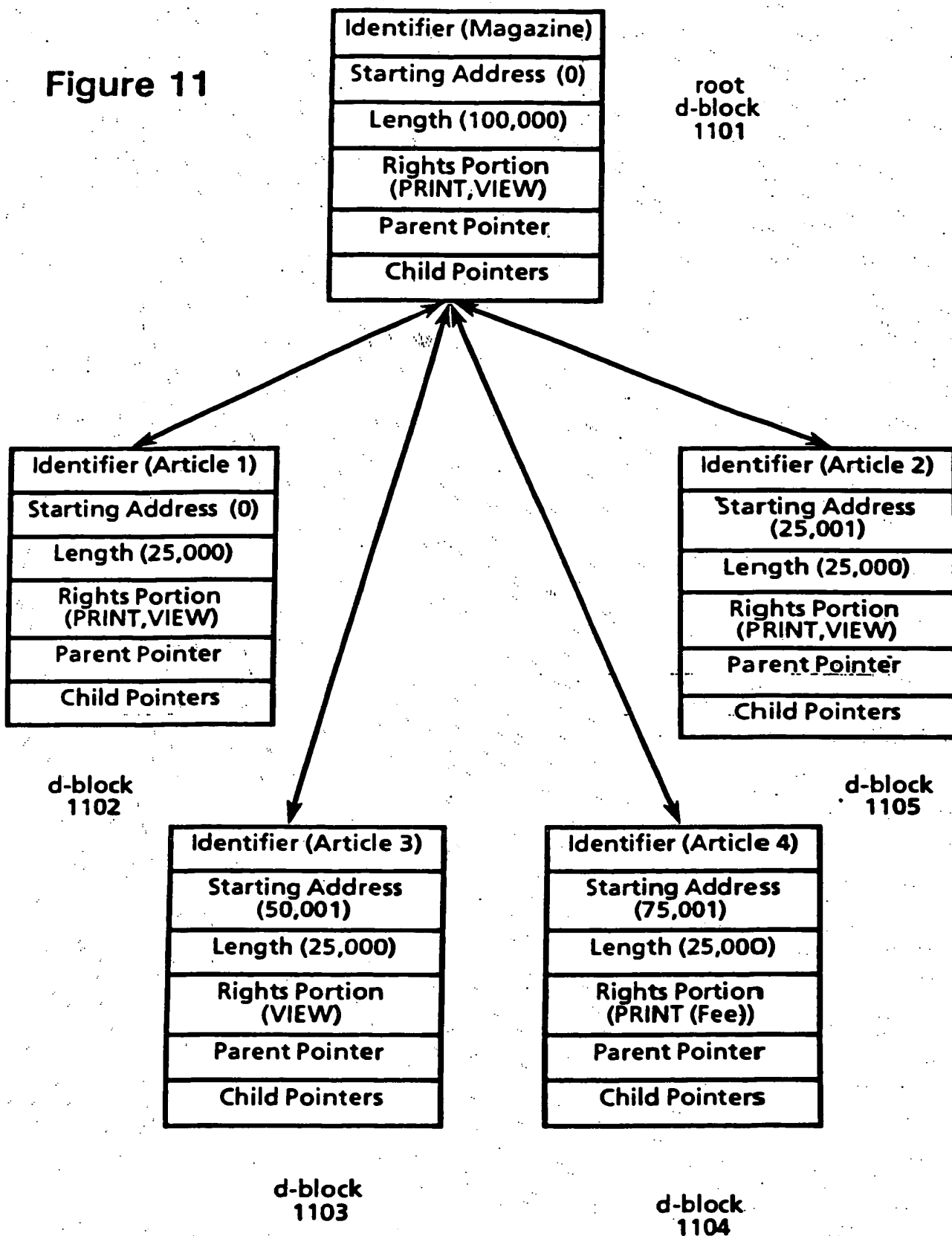


Figure 12

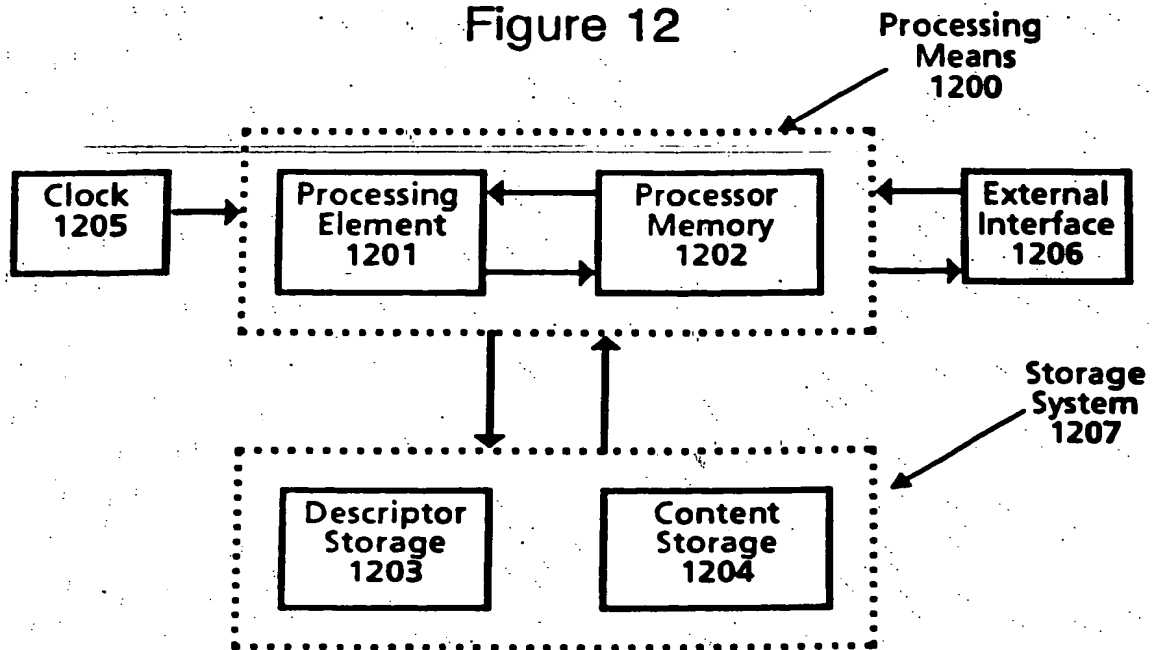


Figure 13

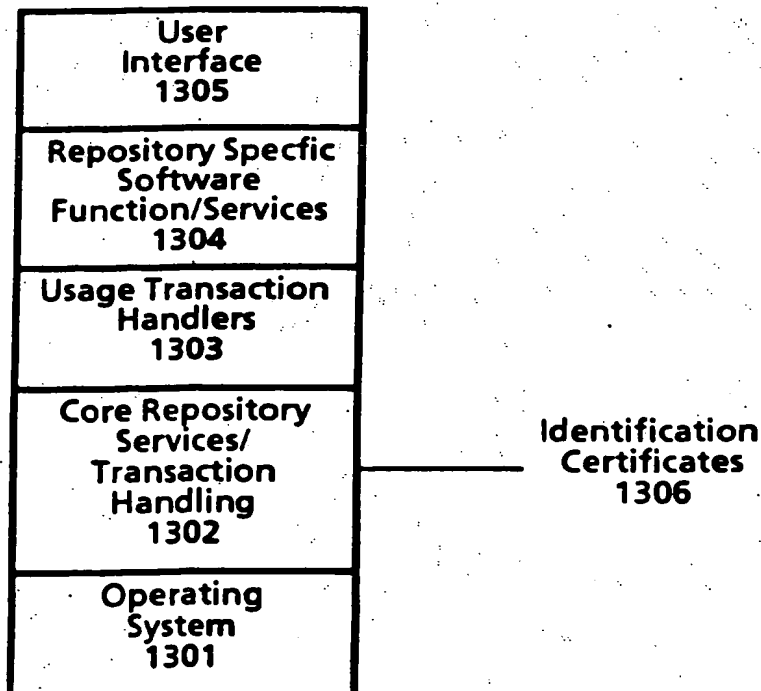


FIGURE 15

```

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

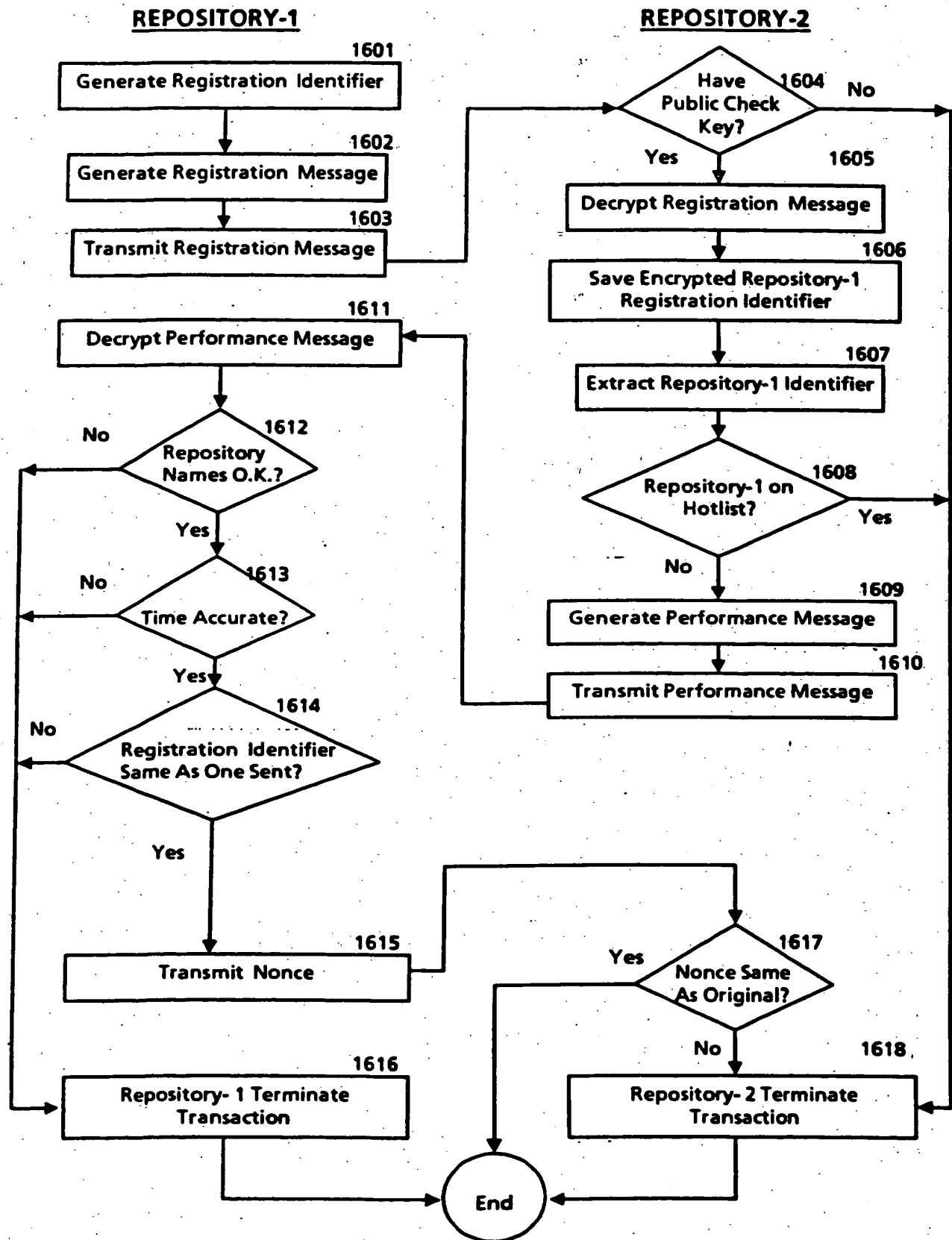


Figure 17

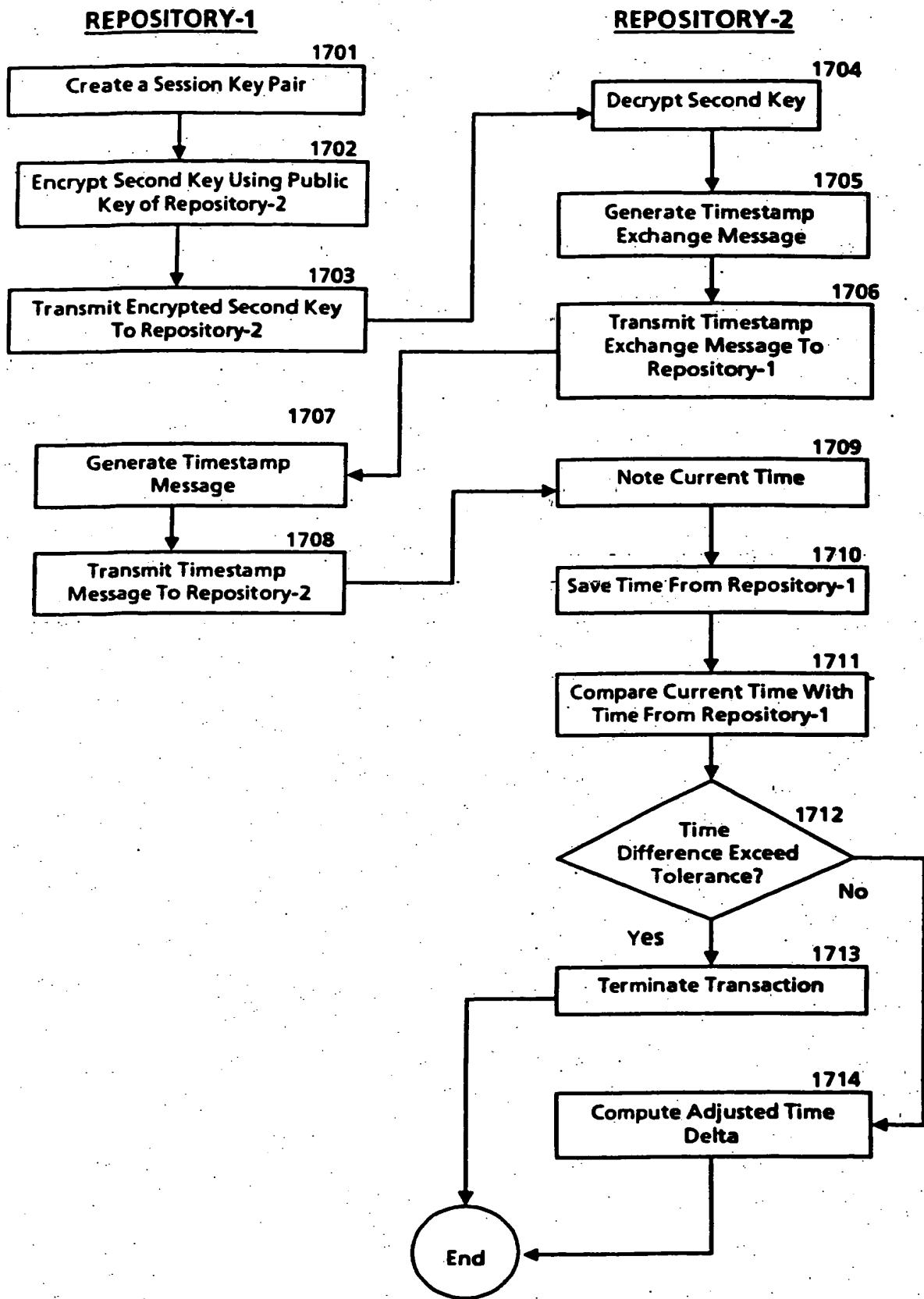


Figure 18

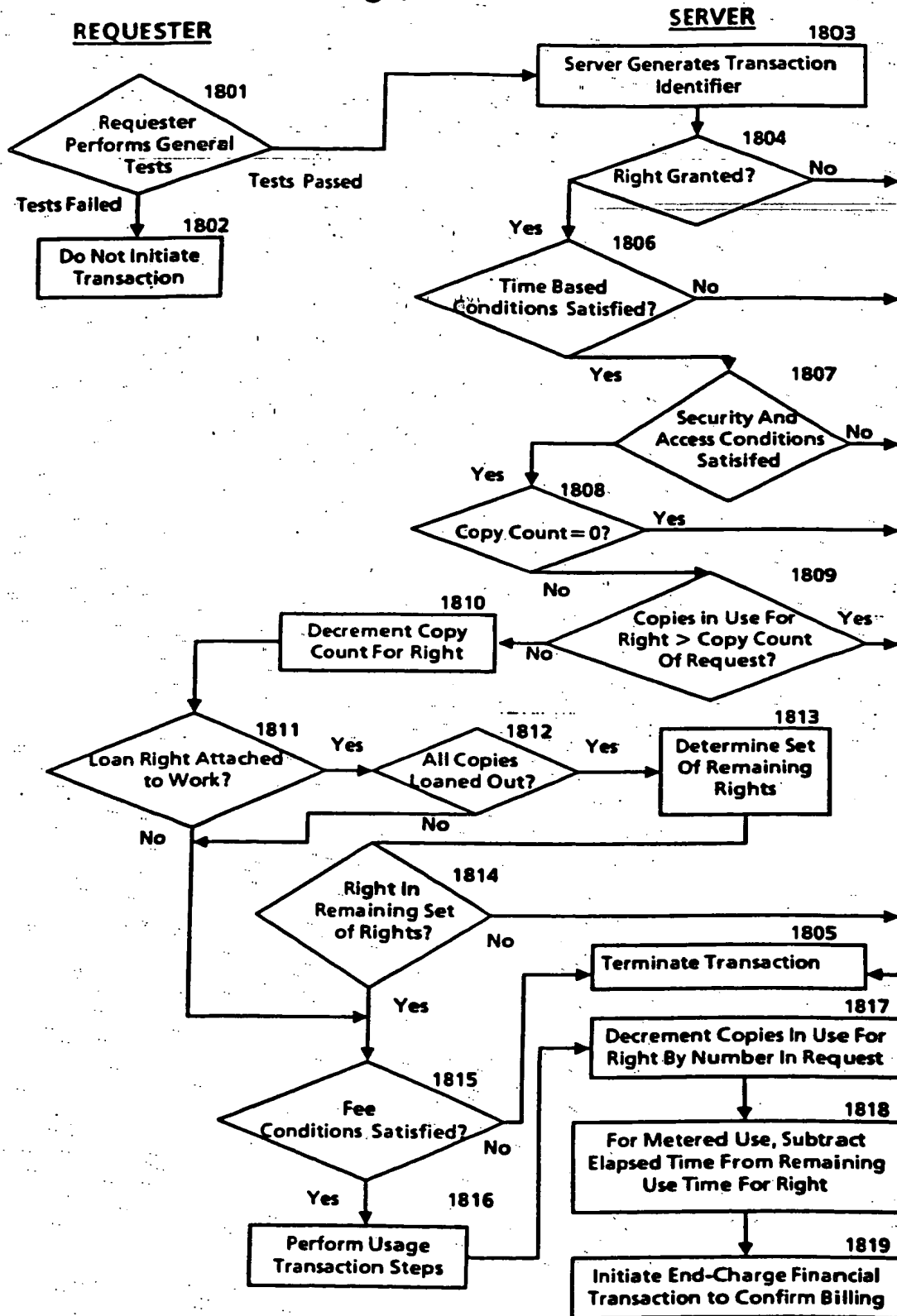
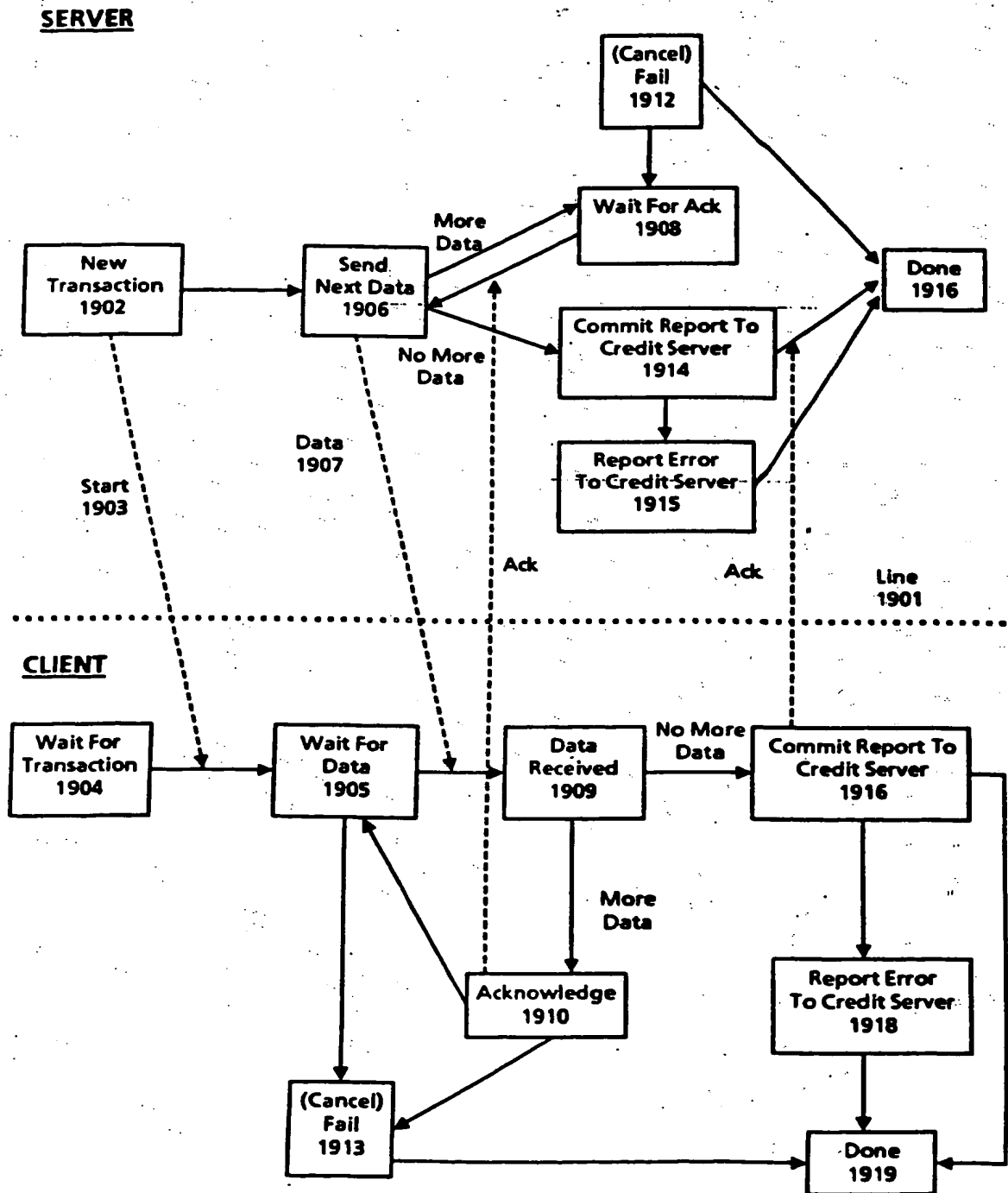


Figure 19



**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.

Declaration and Power of Attorney for Patent Application

As a below inventor, I hereby declare that:

Attorney's Docket No.

D/94789

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: **SYSTEM FOR CONTROLLING THE DISTRIBUTION AND USE OF DIGITAL WORKS UTILIZING DIGITAL TICKETS**

the specification of which



is attached hereto



was filed on as
Application Serial No.
and was amended on (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 referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56(a).

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application

(Number)

(Country)

(Day/Month/Year Filed)

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose material information as defined in Title 37, of Federal Regulations Code, §1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Application Serial No.)

(Filing Date)

(Status)
(patented, pending, abandoned)

(Application Serial No.)

(Filing Date)

(Status)
(patented, pending, abandoned)

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.
Ronald Zibelli, Reg. No. 20,484; Serge Abend, Reg. No. 24,152;
John E. Beck, Reg. No. 22,833; Richard B. Domingo, Reg. No. 36,784

SEND CORRESPONDENCE TO:
Ronald Zibelli
Xerox Corporation
Xerox Square 020
Rochester, New York 14644

DIRECT TELEPHONE CALLS TO:
(name and telephone number)
Richard B. Domingo
415-812-4269

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 §1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

BEST AVAILABLE COPY

Full name of sole or first inventor: **Mark J. Stefik**Inventor's Signature: *Mark J. Stefik* Date: 22 Nov 94Residence: **55 Big Tree Way, Woodside, California 94062**Citizenship: **U.S.A.** Post Office Address: same as aboveFull name of second joint inventor, if any: **Peter L. T. Pirolli**Inventor's Signature: *Peter L. T. Pirolli* Date: 22 Nov 1994Residence: **44 Wildwood Place, El Cerrito, California 94530**Citizenship: **Canadian** Post Office Address: same as above

Full name of third joint inventor, if any:

Inventor's Signature: _____ Date: _____

Residence:

Citizenship: _____ Post Office Address: _____

Full name of fourth joint inventor, if any:

Inventor's Signature: _____ Date: _____

Residence:

Citizenship: _____ Post Office Address: _____

Full name of fifth joint inventor, if any:

Inventor's Signature: _____ Date: _____

Residence:

Citizenship: _____ Post Office Address: _____

Full name of sixth joint inventor, if any:

Inventor's Signature: _____ Date: _____

Residence:

Citizenship: _____ Post Office Address: _____

Full name of seventh joint inventor, if any:

Inventor's Signature: _____ Date: _____

Residence:

Citizenship: _____ Post Office Address: _____

Full name of eighth joint inventor, if any:

Inventor's Signature: _____ Date: _____

Residence:

Citizenship: _____ Post Office Address: _____

Full name of ninth joint inventor, if any:

Inventor's Signature: _____ Date: _____

Residence:

Citizenship: _____ Post Office Address: _____

Rev: 5/15/89

BEST AVAILABLE COPY

PATENT APPLICATION SERIAL NO. _____

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

12/19/2005 SDENB081 00000076 192380 11304794

01 FC:1011	300.00 DA
02 FC:1111	500.00 DA
03 FC:1311	200.00 DA
04 FC:1202	850.00 DA

PTO-1556
(5/87)

U.S. Government Printing Office: 2002 — 496-267/88033

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re New Continuation Patent Application of)
Mark J. STEFIK, et al.)
Based on U.S. Application No. 11/198,216) Attn: Applications
Filed: August 8, 2005) Branch
For: SYSTEM FOR CONTROLLING THE)
DISTRIBUTION AND USE OF)
DIGITAL WORKS) Date: December 16, 2005

INFORMATION DISCLOSURE STATEMENT

United States Patent and Trademark Office
Customer Window, Mail Stop
Randolph Building
401 Dulany Street
Alexandria, VA 22314

Sir:

In accordance with the provisions of 37 C.F.R. 1.56 and 37 C.F.R. 1.97-1.99, it is requested that the references listed on the attached Form PTO-1449 be made of record in the above-identified application.

The references listed on the attached Form PTO-1449 were cited in parent Application No. 11/198,216 filed August 8, 2005, and its predecessors Application Nos. 10/176,608 and 09/777,845 and 08/967,084 and 08/344,760, from which priority is claimed under 35 U.S.C. 120.

It is requested that the accompanying PTO-1449 be considered and made of record in the above-identified 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.

Respectfully submitted,

By: _____

Carlos R. Villanar
Registration No. 43,224

NIXON PEABODY LLP
401 9th Street, N.W.
Suite 900
Washington, D.C. 20004-2128

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.

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Complete if Known			
				Application Number		Not Yet Assigned	
				Filing Date		December 16, 2005	
				First Named Inventor		Mark J. STEFIK, <i>et al.</i>	
				Art Unit		3621	
				Examiner Name		James A. Reagan	
Sheet	1	of	10	Attorney Docket Number		111325-028700	
U.S. PATENT DOCUMENTS							
Examiner Initials*	Cite No. ¹	U.S. Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	
		Number - Kind Code ² (if known)					
		US-3,263,158		07/26/1966	D.W. Bargaen et al.		
		US-3,609,697		09/28/1971	Blevins et al.		
		US-3,790,700		02/05/1974	Callais et al.		
		US-3,798,605		03/19/1974	Feistel		
		US-4,159,468		06/26/1979	Barnes et al.		
		US-4,220,991		09/02/1980	Hamano et al.		
		US-4,278,837		07/14/1981	Best		
		US-4,323,921		04/06/1982	Guillou		
		US-4,442,486		04/10/1984	Mayer		
		US-4,529,870		07/16/1985	Chaum		
		US-4,558,176		12/10/1985	Arnold et al.		
		US-4,593,376		06/03/1986	Volk		
		US-4,614,861		09/30/1986	Pavlov et al.		
		US-4,644,493		02/17/1987	Chandra et al.		
		US-4,658,093		04/14/1987	Hellman		
		US-4,713,753		12/15/1987	Beobert et al.		
		US-4,796,220		01/03/1989	Wolfe		
		US-4,817,140		03/28/1989	Chandra et al.		
		US-4,827,508		05/02/1989	Shear		
		US-4,868,376		09/19/1989	Lessin et al.		
FOREIGN PATENT DOCUMENTS							
Examiner Initials*	Cite No. ¹	Foreign Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T ⁶
		Country Code ³	Number ⁴ Kind Code ⁵ (if known)				
		0 084 441	EP	07/27/1983	TABS LIMITED		
		0 180 460	EP	05/07/1986	SONY CORPORATION		
		0 332 707	EP	09/20/1989	HONDA GIKEN KOGYO KABUSHIKI KAISHA		
		0 651 554	EP	05/03/1995	EASTMAN KODAK CO.		
		0 668 695	EP	08/23/1995	VICTOR COMPANY OF JAPAN LIMITED		
		0 725 376	EP	08/07/1996	SONY CORP.		
		2 136 175	GB	09/12/1984	ATALLA CORP.		
		2 236 604	GB	04/10/1991	SUN MICROSYSTEMS INC		
		0 715 241	JP	06/05/1996	MITSUBISHI CORP.		
		04-369068	JP	12/21/1992	CHIYUUBU NIHON DENKI SOFUTOEUA KK		Abst
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 Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at 222.uspto.gov or MPEP 901.04. 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.

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.

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.

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Complete if Known			
				Application Number		Not Yet Assigned	
				Filing Date		December 16, 2005	
				First Named Inventor		Mark J. STEFIK, <i>et al.</i>	
				Art Unit		3621	
				Examiner Name		James A. Reagan	
Sheet	2	of	10	Attorney Docket Number		111325-028700	

U.S. PATENT DOCUMENTS					
Examiner Initials*	Cite No. ¹	U.S. Patent Document	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Number - Kind Code ² (if known)			
		US-4,891,838	01/02/1990	Faber	
		US-4,924,378	05/08/1990	Hershey et al.	
		US-4,932,054	06/05/1990	Chou et al.	
		US-4,937,863	06/26/1990	Robert et al.	
		US-4,949,187	08/14/1990	Cohen	
		US-4,953,209	08/28/1990	Ryder, Sr. et al.	
		US-4,961,142	10/02/1990	Elliott et al.	
		US-4,975,647	12/04/1990	Downer et al.	
		US-4,977,594	12/11/1990	Shear	
		US-4,999,806	03/12/1991	Chernow et al.	
		US-5,010,571	04/23/1991	Katznelson	
		US-5,014,234	05/07/1991	Edwards, Jr.	
		US-5,023,907	06/11/1991	Johnson et al.	
		US-5,047,928	09/10/1991	Wiedemer	
		US-5,050,213	09/17/1991	Shear	
		US-5,052,040	09/24/1991	Preston et al.	
		US-5,058,164	10/15/1991	Elmer et al.	
		US-5,103,476	04/07/1992	Waite et al.	
		US-5,113,519	05/12/1992	Johnson et al.	
		US-5,136,643	08/04/1992	Fischer	

FOREIGN PATENT DOCUMENTS							
Examiner Initials [*]	Cite No. ¹	Foreign Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T ⁶
		Country Code ³	Number ⁴	Kind Code ⁵ (if known)			
		05-268415	JP	10/15/1993	RICOH CO LTD		Abst
		06-175794	JP	06/24/1994	FUJI XEROX CO LTD		Abst
		06-215010	JP	08/05/1994	SONY CORP.		Abst
		07-084852	JP	03/31/1995	HITACHI LTD.		Abst
		07-200317	JP	08/04/1995	TOSHIBA CORP.		Abst
		07-244639	JP	09/19/1995	FUJITSU LTD		Abst
		62-241061	JP	10/21/1987	NEC CORP.		Abst
		64-068835	JP	03/14/1989	RYOICHI MORI		Abst
		WO 01/63528	PCT	08/30/2001	IPDN COPR.		
		WO 92/20022	PCT	11/12/1992	DIGITAL EQUIPMENT CORP.		

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 Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at 222.uspto.gov or MPEP 901.04. 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.

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.

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.

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Complete if Known	
				Application Number	Not Yet Assigned
				Filing Date	December 16, 2005
				First Named Inventor	Mark J. STEFIK, <i>et al.</i>
				Art Unit	3621
				Examiner Name	James A. Reagan
Sheet	3	of	10	Attorney Docket Number	111325-028700

U.S. PATENT DOCUMENTS							
Examiner Initials*	Cite No. ¹	U.S. Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	
		Number - Kind Code ² (if known)					
		US-5,138,712		08/11/1992	Corbin		
		US-5,146,499		09/08/1992	Geffroin		
		US-5,148,481		09/15/1992	Abraham et al.		
		US-5,159,182		10/27/1992	Eisele		
		US-5,183,404		02/02/1993	Aldous et al.		
		US-5,191,193		03/02/1993	Le Roux		
		US-5,204,897		04/20/1993	Wyman		
		US-5,222,134		06/22/1993	Waite et al.		
		US-5,235,642		08/10/1993	Wobber et al.		
		US-5,247,575		09/21/1993	Sprague et al.		
		US-5,255,106		10/19/1993	Castro		
		US-5,260,999		11/09/1993	Wyman		
		US-5,263,157		11/16/1993	Janis		
		US-5,263,158		11/16/1993	Janis		
		US-5,276,444		01/04/1994	McNair		
		US-5,276,735		01/04/1994	Boebert et al.		
		US-5,291,596		03/01/1994	Mita		
		US-5,301,231		04/05/1994	Abraham et al.		
		US-5,311,591		05/10/1994	Fischer		
		US-5,319,705		06/07/1994	Halter et al.		
FOREIGN PATENT DOCUMENTS							
Examiner Initials*	Cite No. ¹	Foreign Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T ⁶
		Country Code ³ Number ⁴	Kind Code ⁵ (if known)				
		WO 93/01550	PCT	01/21/1993	INFOLOGIC SOFTWARE, INC		
		WO 94/01821	PCT	01/20/1994	SECURE COMPUTING CORP.		
		WO 96/24092	PCT	08/08/1996	BENSON, Greg		
		WO 97/48203	PCT	12/18/1997	INTEL CORP.		
		WO 98/11690	PCT	03/19/1998	GLOVER, John J.		
		WO 98/42098	PCT	09/24/1998	CRYPTOWORKS, INC.		
		WO 99/49615	PCT	09/30/1999	MICROTOME		
		EP 0 538 216 A1		04/21/1993	TELEVERKET		
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 Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at 222.uspto.gov or MPEP 901.04. 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.

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.

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.

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Complete if Known	
				Application Number	Not Yet Assigned
				Filing Date	December 16, 2005
				First Named Inventor	Mark J. STEFIK, <i>et al.</i>
				Art Unit	3621
				Examiner Name	James A. Reagan
Sheet	4	of	10	Attorney Docket Number	111325-028700

U.S. PATENT DOCUMENTS						
Examiner Initials*	Cite No. ¹	U.S. Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Number - Kind Code ² (if known)				
		US-5,337,357		08/09/1994	Chou et al.	
		US-5,339,091		08/16/1994	Yamazaki et al.	
		US-5,341,429		08/23/1994	Stringer et al.	
		US-5,347,579		09/13/1994	Blandford	
		US-5,381,526		01/10/1995	Ellson	
		US-5,394,469		02/28/1995	Nagel et al.	
		US-5,410,598		04/25/1995	Shear	
		US-5,412,717		05/02/1995	Fischer	
		US-5,428,606		06/27/1995	Moskowitz	
		US-5,432,849		07/11/1995	Johnson et al.	
		US-5,438,508		08/01/1995	Wyman	
		US-5,444,779		08/22/1995	Daniele	
		US-5,453,601		09/26/1995	Rosen	
		US-5,455,953		10/03/1995	Russell	
		US-5,457,746		10/10/1995	Dolphin	
		US-5,473,687		12/05/1995	Lipscomb et al.	
		US-5,473,692		12/05/1995	Davis	
		US-5,499,298		03/12/1996	Narasimhalu et al.	
		US-5,502,766		03/26/1996	Boebert et al.	
		US-5,504,814		04/02/1996	Miyahara	

FOREIGN PATENT DOCUMENTS						
Examiner Initials [*]	Cite No. ¹	Foreign Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Country Code ³	Number ⁴			

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 Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at 222.uspto.gov or MPEP 901.04. 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.

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.

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.

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Complete if Known			
				Application Number		Not Yet Assigned	
				Filing Date		December 16, 2005	
				First Named Inventor		Mark J. STEFIK, <i>et al.</i>	
				Art Unit		3621	
				Examiner Name		James A. Reagan	
Sheet	5	of	10	Attorney Docket Number		111325-028700	

U.S. PATENT DOCUMENTS						
Examiner Initials*	Cite No. ¹	U.S. Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Number - Kind Code ² (if known)				
		US-5,504,818		04/02/1996	Okano	
		US-5,504,837		04/02/1996	Griffeth et al.	
		US-5,509,070		04/16/1996	Schull	
		US-5,530,235		06/25/1996	Stefik et al.	
		US-5,532,920		07/02/1996	Hartrick et al.	
		US-5,534,975		07/09/1996	Stefik et al.	
		US-5,539,735		07/23/1996	Moskowitz	
		US-5,563,946		10/08/1996	Cooper et al.	
		US-5,568,552		10/22/1996	Davis	
		US-5,621,797		04/15/1997	Rosen	
		US-5,629,980		05/13/1997	Stefik et al.	
		US-5,633,932		05/27/1997	Davis et al.	
		US-5,634,012		05/27/1997	Stefik et al.	
		US-5,638,443		06/10/1997	Stefik et al.	
		US-5,649,013		07/15/1997	Stuckey et al.	
		US-5,655,077		08/05/1997	Jones et al.	
		US-5,708,717		01/13/1998	Alasia	
		US-5,734,823		03/31/1998	Saigh et al.	
		US-5,734,891		03/31/1998	Saigh	
		US-5,737,413		04/07/1998	Akiyama et al.	

FOREIGN PATENT DOCUMENTS							
Examiner Initials*	Cite No. ¹	Foreign Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T ⁶
		Country Code ³ Number ⁴	Kind Code ⁵ (if known)				
		</					

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 Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at 222.uspto.gov or MPEP 901.04. 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.

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.

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.

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Complete if Known			
				Application Number		Not Yet Assigned	
				Filing Date		December 16, 2005	
				First Named Inventor		Mark J. STEFIK, <i>et al.</i>	
				Art Unit		3621	
				Examiner Name		James A. Reagan	
Sheet	6	of	10	Attorney Docket Number		111325-028700	

U.S. PATENT DOCUMENTS					
Examiner Initials*	Cite No. ¹	U.S. Patent Document	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Number - Kind Code ² (if known)			
		US-5,737,416	04/07/1998	Cooper et al.	
		US-5,745,569	04/28/1998	Moskowitz et al.	
		US-5,748,783	05/05/1998	Rhoads	
		US-5,757,907	05/26/1998	Cooper et al.	
		US-5,761,686	06/02/1998	Bloomberg	
		US-5,765,152	06/09/1998	Erickson	
		US-5,768,426	06/16/1998	Rhoads	
		US-5,825,892	10/20/1998	Braudaway et al.	
		US-5,892,900	04/06/1999	Ginter et al.	
		US-5,910,987	06/08/1999	Ginter et al.	
		US-5,915,019	06/22/1999	Ginter et al.	
		US-5,917,912	06/29/1999	Ginter et al.	
		US-5,920,861	07/06/1999	Hall et al.	
		US-5,940,504	08/17/1999	Griswold	
		US-5,943,422	08/24/1999	Van Wie et al.	
		US-5,949,876	09/07/1999	Ginter et al.	
		US-5,982,891	11/09/1999	Ginter et al.	
		US-5,999,949	12/07/1999	Crandall	
		US-6,047,067	04/04/2000	Rosen	
		US-6,112,181	08/29/2000	Shear et al.	

FOREIGN PATENT DOCUMENTS							
Examiner Initials [*]	Cite No. ¹	Foreign Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T ⁶
		Country Code ³ Number ⁴	Kind Code ⁵ (if known)				

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 Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at 222.uspto.gov or MPEP 901.04. 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.

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.

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.

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Complete if Known			
				Application Number		Not Yet Assigned	
				Filing Date		December 16, 2005	
				First Named Inventor		Mark J. STEFIK, <i>et al.</i>	
				Group Art Unit		3621	
				Examiner Name		James A. Reagan	
Sheet	7	of	10	Attorney Docket Number		111325-028700	

U.S. PATENT DOCUMENTS					
Examiner Initials*	Cite No. ¹	U.S. Patent Document	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Number - Kind Code ² (if known)			
		US-6,115,471	09/05/2000	Oki et al.	
		US-6,138,119	10/24/2000	Hall et al.	
		US-6,157,721	12/05/2000	Shear et al.	
		US-6,185,683	02/06/2001	Ginter et al.	
		US-6,226,618	05/01/2001	Downs et al.	
		US-6,233,684	05/15/2001	Stefik et al.	
		US-6,237,786	05/29/2001	Ginter et al.	
		US-6,240,185	05/29/2001	Van Wie et al.	
		US-6,253,193	06/26/2001	Ginter et al.	
		US-6,266,618	07/24/2001	Ye et al.	
		US-6,292,569	09/18/2001	Shear et al.	
		US-6,301,660	10/09/2001	Benson	
		US-6,327,652	12/04/2001	England et al.	
		US-6,330,670	12/11/2001	England et al.	
		US-6,345,256	02/05/2002	Milsted et al.	
		US-6,363,488	03/26/2002	Ginter et al.	
		US-6,389,402	05/14/2002	Ginter et al.	
		US-4,882,752	11/21/1989	Lindman et al.	
		US-5,224,163	06/29/1993	Gasser et al.	
		US-5,299,263	03/29/1994	Beller et al.	

FOREIGN PATENT DOCUMENTS							
Examiner Initials [*]	Cite No. ¹	Foreign Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T ⁶
		Country Code ³	Number ⁴ Kind Code ² (if known)				

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 Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at 222.uspto.gov or MPEP 901.04. 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.

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.

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.

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Complete if Known	
				Application Number	Not Yet Assigned
				Filing Date	December 16, 2005
				First Named Inventor	Mark J. STEFIK, <i>et al.</i>
				Art Unit	3621
				Examiner Name	James A. Reagan
Sheet	8	of	10	Attorney Docket Number	111325-028700

OTHER PRIOR ART – NON PATENT LITERATURE DOCUMENTS			
Examiner Initials*	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
		"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 the IEICE, Vo. E 73, No. 7, Tokyo, JP	
		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. 1-3, April 1994, Computer Shopper, Vol. 14, No. 4, ISSN 0886-0556	

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 Applicant's unique citation designation number (optional). 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.

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.

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Complete if Known			
				Application Number		Not Yet Assigned	
				Filing Date		December 16, 2005	
				First Named Inventor		Mark J. STEFIK, <i>et al.</i>	
				Art Unit		3621	
				Examiner Name		James A. Reagan	
Sheet	9	of	10	Attorney Docket Number		111325-028700	

OTHER PRIOR ART - NON PATENT LITERATURE DOCUMENTS			
Examiner Initials*	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
		O'Conner, M., "New Distribution Option for Electronic Publishers; iOpener Data Encryption and Metering System for CD-ROM use; Column", pp. 1-6, 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	
		Simmell, 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	

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 Applicant's unique citation designation number (optional). 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.

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.

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Complete if Known	
				Application Number	Not Yet Assigned
				Filing Date	December 16, 2005
				First Named Inventor	Mark J. STEFIK, <i>et al.</i>
				Art Unit	3621
				Examiner Name	James A. Reagan
Sheet	10	of	10	Attorney Docket Number	111325-028700

OTHER PRIOR ART - NON PATENT LITERATURE DOCUMENTS			
Examiner Initials*	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
		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", PP. 1-24, 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	
		Henry H. Perritt, Jr., "Technological Strategies for Protecting Intellectual Property in the Networked Multimedia Environment", April 2-3, 1993, Knowbots, Permissions Headers & Contract Law	
		Perritt, Henry H.; Knowbots, Permissions Headers and Contract Law paper for conference on <i>Technological Strategies for Protecting Intellectual Property in the Networked Multimedia Environment</i> , April 2-3, 1993, pages 1-22	
		European Search Report dated September 8, 2004 in corresponding European Application No. EP 03 015 128.6	
		European Search Report dated September 13, 2004 in corresponding European Application No. EP 03 015 127.8	
		European Search Report for Corresponding European Application 95308422.5	

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 Applicant's unique citation designation number (optional). 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.

APPLICATION DATA SHEET

Electronic Version v14

Stylesheet Version v14.1

*Applicant Information:*Inventor 1:

Applicant Authority Type:	Inventor
Citizenship:	US
Given Name:	Mark
Middle Name:	J.
Family Name:	STEFIK
City of Residence:	Portola Valley
State of Residence:	CA
Country of Residence:	US
Address-1 of Mailing Address:	10 Portola Green Circle
Address-2 of Mailing Address:	
City of Mailing Address:	Portola Valley
State of Mailing Address:	CA
Postal Code of Mailing Address:	94028
Country of Mailing Address:	US
Phone:	
Fax:	
E-mail:	

Inventor 2:

Applicant Authority Type:	Inventor
Citizenship:	CA
Given Name:	Peter
Middle Name:	L.T.
Family Name:	PIROLI
City of Residence:	San Francisco
State of Residence:	CA
Country of Residence:	US
Address-1 of Mailing Address:	2958 Sloat Blvd.
Address-2 of Mailing Address:	

City of Mailing Address: San Francisco
State of Mailing Address: CA
Postal Code of Mailing Address: 94116
Country of Mailing Address: US
Phone:
Fax:
E-mail:

Correspondence Information:

Customer Number: 22204 *22204*

Application Information:

Title of Invention: SYSTEM FOR CONTROLLING THE DISTRIBUTION AND
USE OF DIGITAL WORKS

Application Type: regular, utility

Attorney Docket Number: 111325-028700

Botanic Information:

Publication Information:

Suggested Figure for Publication -

Suggested Classification -

Suggested Technology Center -

Total Number of Drawing Sheets -

Representative Information:

practitioner(s) at Customer Number:

22204 *22204*

as our attorney(s) or agent(s) to prosecute the application identified above, and to transact all business in the United States Patent and Trademark Office connected therewith.

Domestic Priority Information:

This is a Continuation of US application number 11/198,216, filed 2005-08-08, now pending which is a continuation of United States Patent Application Serial No. 10/176,608 filed on June 24, 2002 now United States Patent No. 6,934,693, which is a continuation of United States Patent Application Serial No. 09/777,845, filed on February 7, 2001, now pending, which is a divisional of United States Patent Application Serial No. 08/967,084, filed on November 10, 1997, now United States Patent No. 6,236,971, which is a continuation of United States Patent Application Serial No. 08/344,760, filed on November 23, 1994, now abandoned..

Foreign Priority Information:

Assignee Information:

Assignee 1:

Organization Name: CONTENTGUARD HOLDINGS, INC.

Address-1 of Mailing Address: 103 Foulk Road, Suite 200-M

Address-2 of Mailing Address:

City of Mailing Address: Wilmington,

State of Mailing Address: DE

Postal Code of Mailing Address: 19803

Country of Mailing Address: US

Phone:

Fax:

E-mail:

14, 26

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.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875 Effective December 8, 2004

Application or Docket Number

11304794

APPLICATION AS FILED - PART I

(Column 1)

(Column 2)

SMALL ENTITY

OR

OTHER THAN
SMALL ENTITY

FOR	NUMBER FILED	NUMBER EXTRA
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A
SEARCH FEE (37 CFR 1.16(k), (l), or (m))	N/A	N/A
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A
TOTAL CLAIMS (37 CFR 1.16(i))	37 minus 20 =	17
INDEPENDENT CLAIMS (37 CFR 1.16(h))	3 minus 3 =	0
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$250 (\$125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).	
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))		

RATE (\$)	FEE (\$)
N/A	150.00
N/A	\$250
N/A	\$100
X\$ 25 =	
X100 =	
+180=	
TOTAL	

RATE (\$)	FEE (\$)
N/A	300.00
N/A	\$500
N/A	\$200
X\$50 =	80
X200 =	
+360=	
TOTAL	1850

* If the difference in column 1 is less than zero, enter "0" in column 2.

APPLICATION AS AMENDED - PART II

(Column 1)

(Column 2)

(Column 3)

SMALL ENTITY

OR

OTHER THAN
SMALL ENTITY

AMENDMENT A	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total (37 CFR 1.16(i))	Minus	**	=
Independent (37 CFR 1.16(h))	Minus	***	=
Application Size Fee (37 CFR 1.16(s))			
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))			

RATE (\$)	ADDITIONAL FEE (\$)
X\$ 25 =	
X100 =	
+180=	
TOTAL ADD'L FEE	

RATE (\$)	ADDITIONAL FEE (\$)
X\$50 =	
X200 =	
+360=	
TOTAL ADD'L FEE	

(Column 1)

(Column 2)

(Column 3)

AMENDMENT B	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total (37 CFR 1.16(i))	Minus	**	=
Independent (37 CFR 1.16(h))	Minus	***	=
Application Size Fee (37 CFR 1.16(s))			
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))			

RATE (\$)	ADDITIONAL FEE (\$)
X\$ 25 =	
X100 =	
+180=	
TOTAL ADD'L FEE	

RATE (\$)	ADDITIONAL FEE (\$)
X\$50 =	
X200 =	
+360=	
TOTAL ADD'L 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.

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371 (c) DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NUMBER
11/304,794	12/16/2005	Mark J. Stefik	111325-028700

22204
NIXON PEABODY, LLP
401 9TH STREET, NW
SUITE 900
WASHINGTON, DC 20004-2128

CONFIRMATION NO. 6096
FORMALITIES
LETTER

Date Mailed: 01/27/2006

NOTICE TO FILE CORRECTED APPLICATION PAPERS

Filing Date Granted

An application number and filing date have been accorded to this application. The application is informal since it does not comply with the regulations for the reason(s) indicated below. Applicant is given TWO MONTHS from the date of this Notice within which to correct the informalities indicated below. 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 required item(s) identified below must be timely submitted to avoid abandonment:

- Replacement drawings in compliance with 37 CFR 1.84 and 37 CFR 1.121(d) are required. The drawings submitted are not acceptable because:
 - The drawings must be reasonably free from erasures and must be free from alterations, overwriting, interlineations, folds, and copy marks. See Figure(s) 1-3, 4a-4b, and 5-19.

Applicant is cautioned that correction of the above items may cause the specification and drawings page count to exceed 100 pages. If the specification and drawings exceed 100 pages, applicant will need to submit the required application size fee.

Replies should be mailed to: Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

*A copy of this notice **MUST** be returned with the reply.*

Office of Initial Patent Examination (571) 272-4000, or 1-800-PTO-9199, or 1-800-972-6382
PART 3 - OFFICE COPY



ZFW

TRANSMITTAL FORM (to be used for all correspondence after initial filing)		Application Number	11/304,794
		Filing Date	December 16, 2005
		First Named Inventor	Mark J. STEFIK, et al.
		Group Art Unit	3621
		Confirmation No.	6096
Total Number of Pages in This Submission		Attorney Docket Number	111325-028700

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 Notice to File Corrected Application Papers	☐ 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 ☐ Appeal Communication to Board of Appeals and Interferences ☐ Appeal Communication to Group (Appeal Notice, Brief, Reply Brief) ☐ Proprietary Information ☐ Status Letter ☐ Application Data Sheet ☐ Request for Corrected Filing Receipt with Enclosures ☐ A self-addressed prepaid postcard for acknowledging receipt ☐ Other Enclosure(s) (please identify below):
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	Carlos R. Villamar-Reg. No. 43,224 Nixon Peabody LLP 401 9 th Street, N.W. Suite 900 Washington, D.C. 20004-2128
Signature	
Date	March 13, 2006

CERTIFICATE OF MAILING OR TRANSMISSION [37 CFR 1.8(a)]	
I hereby certify that this correspondence is being:	
☐ deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop _____, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450	
☐ transmitted by facsimile on the date shown below to the United States Patent and Trademark Office at (703) _____.	
Date	Signature
	Typed or printed name



Docket No. 111325-028700

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of)
Mark J. STEFIK, et al.)
Application No. 11/304,794) Art Unit: 3621
Filed: December 16, 2005) Confirmation No. 6096
For: SYSTEM FOR CONTROLLING THE)
DISTRIBUTION AND USE OF DIGITAL)
WORKS) Date: March 13, 2006

NOTICE TO FILE CORRECTED APPLICATION PAPERS AND
SUBMISSION OF REPLACEMENT DRAWINGS

United States Patent and Trademark Office
Customer Service Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

Sir:

In response to the Notice to File Corrected Application Papers, mail January 27, 2006, enclosed herewith are 13 sheets of replacement drawings consisting of Figures 1-13 for the above-identified application.

It is requested by the undersigned attorney that these drawings be made of record in the above-identified application. Acknowledgment of receipt of these drawings is requested.

Respectfully submitted,

By: _____

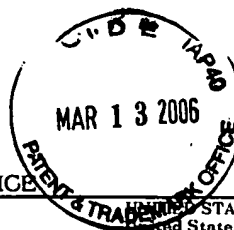
Carlos R. Villamar
Registration No. 43,224

NIXON PEABODY LLP
401 9th Street, N.W.
Suite 900
Washington, DC 20004-2128
Telephone: (202) 585-8000

W709332.1



UNITED STATES PATENT AND TRADEMARK OFFICE



UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371 (c) DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NUMBER
11/304,794	12/16/2005	Mark J. Stefik	111325-028700

22204
NIXON PEABODY, LLP
401 9TH STREET, NW
SUITE 900
WASHINGTON, DC 20004-2128

CONFIRMATION NO. 6096
FORMALITIES
LETTER

Date Mailed: 01/27/2006

NOTICE TO FILE CORRECTED APPLICATION PAPERS

Filing Date Granted

An application number and filing date have been accorded to this application. The application is informal since it does not comply with the regulations for the reason(s) indicated below. Applicant is given TWO MONTHS from the date of this Notice within which to correct the informalities indicated below. 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 required item(s) identified below must be timely submitted to avoid abandonment:

- Replacement drawings in compliance with 37 CFR 1.84 and 37 CFR 1.121(d) are required. The drawings submitted are not acceptable because:
 - The drawings must be reasonably free from erasures and must be free from alterations, overwriting, interlineations, folds, and copy marks. See Figure(s) 1-3, 4a-4b, and 5-19.

Applicant is cautioned that correction of the above items may cause the specification and drawings page count to exceed 100 pages. If the specification and drawings exceed 100 pages, applicant will need to submit the required application size fee.

Replies should be mailed to: Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

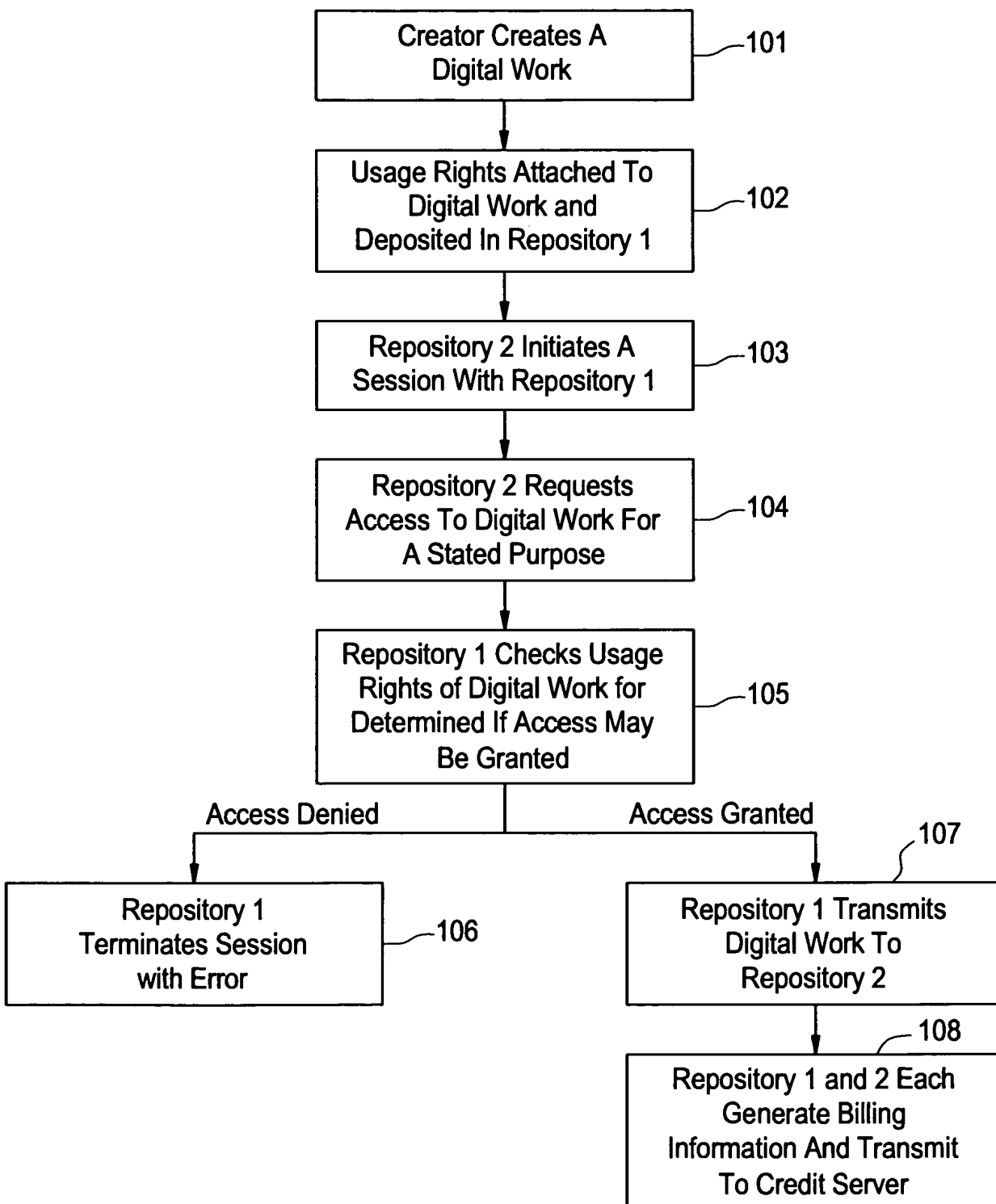
*A copy of this notice **MUST** be returned with the reply.*

Office of Initial Patent Examination (571) 272-4000, or 1-800-PTO-9199, or 1-800-972-6382
PART 2 - COPY TO BE RETURNED WITH RESPONSE



1/13

FIG. 1



2/13

FIG. 2

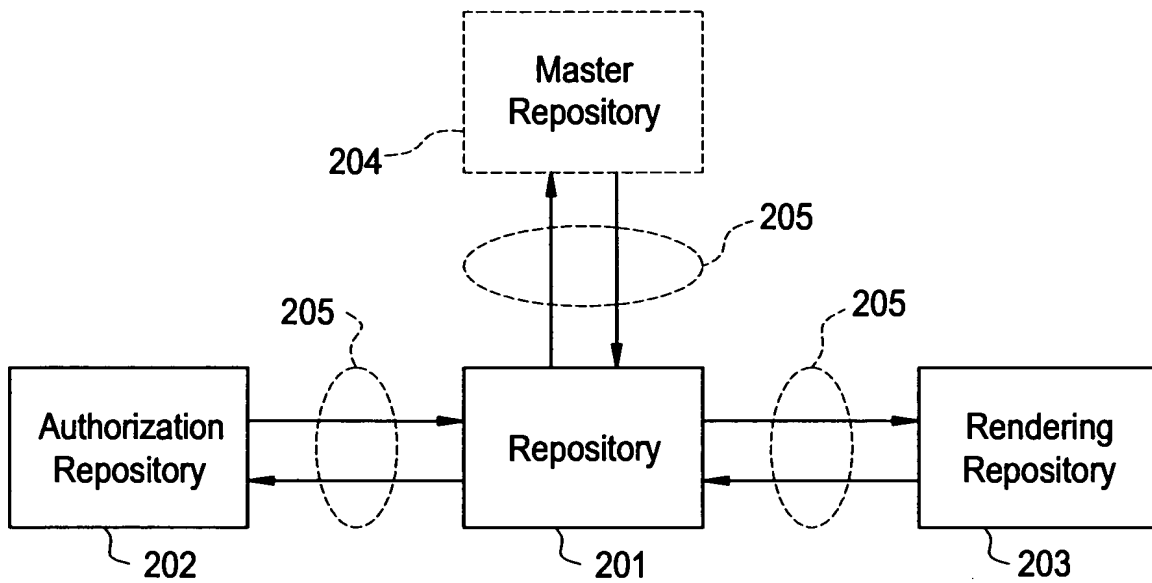
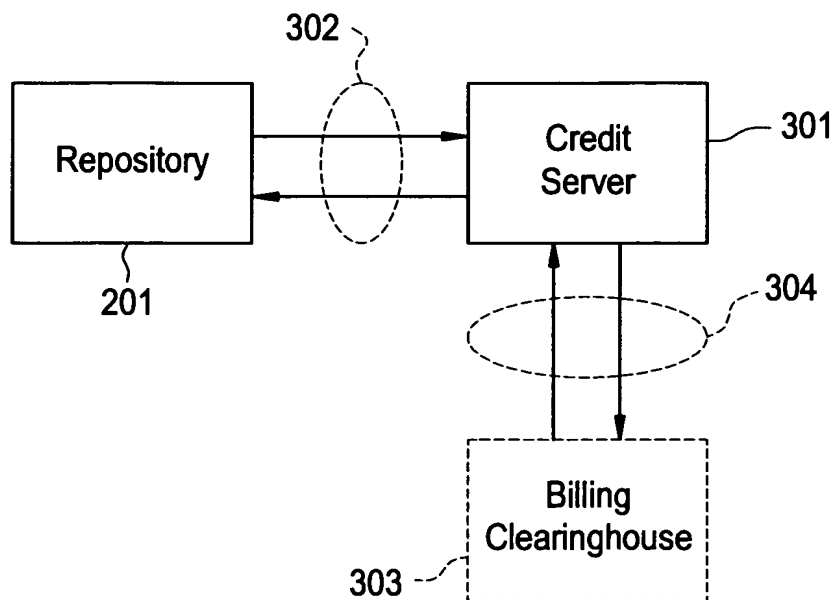


FIG. 3



3/13

FIG. 4A

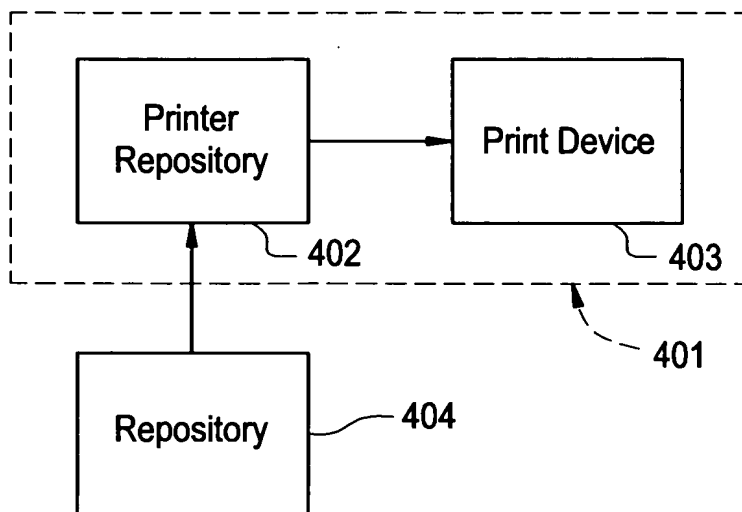
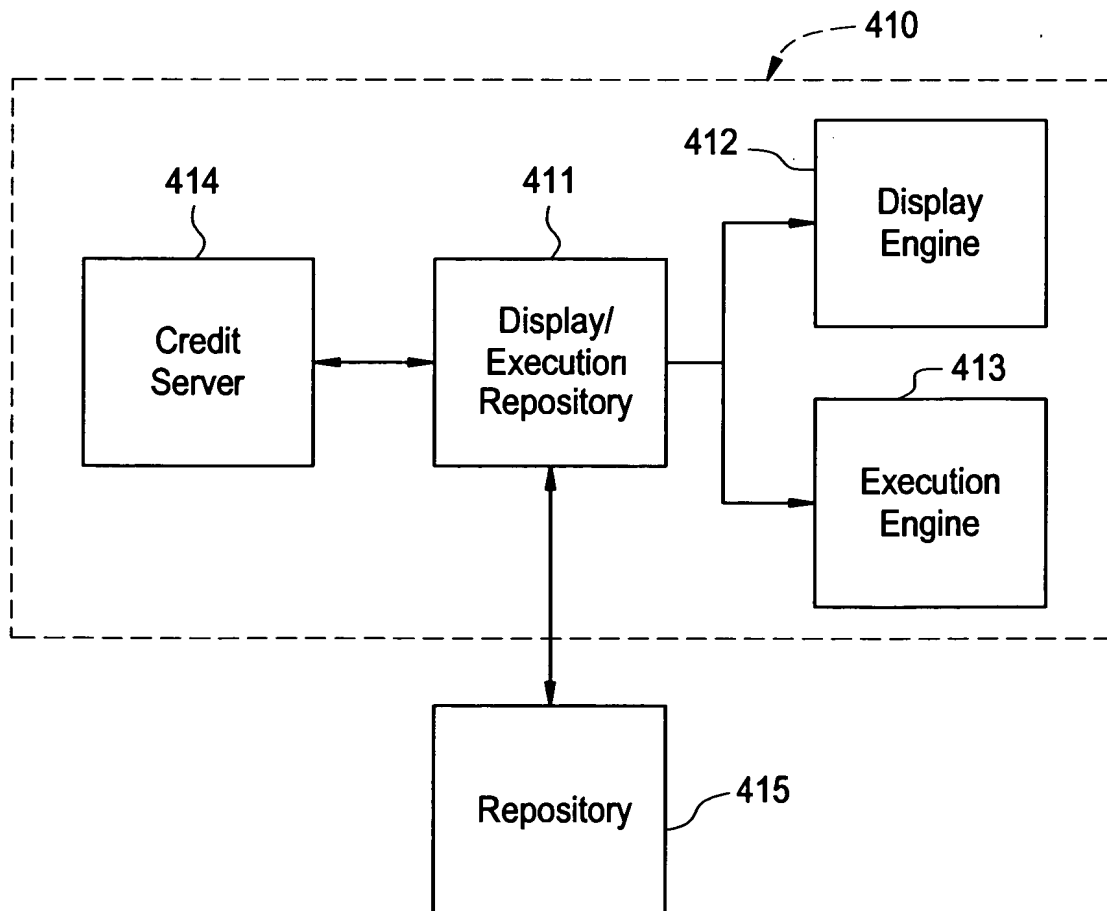


FIG. 4B



4/13

FIG. 5

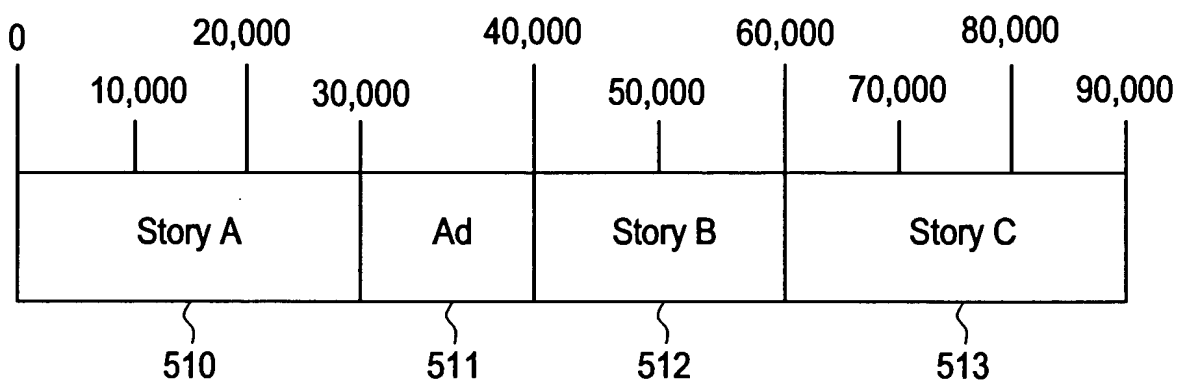


FIG. 6

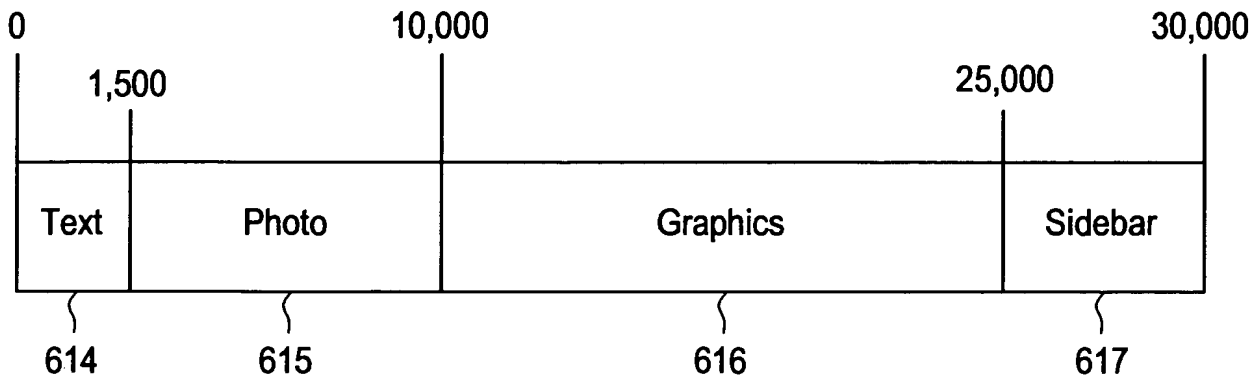


FIG. 7

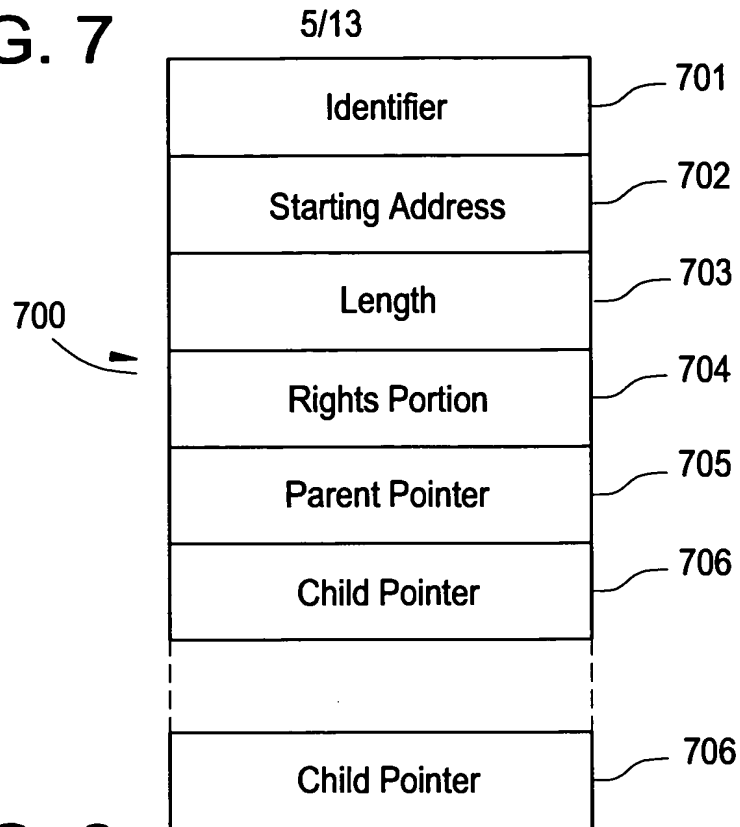


FIG. 8

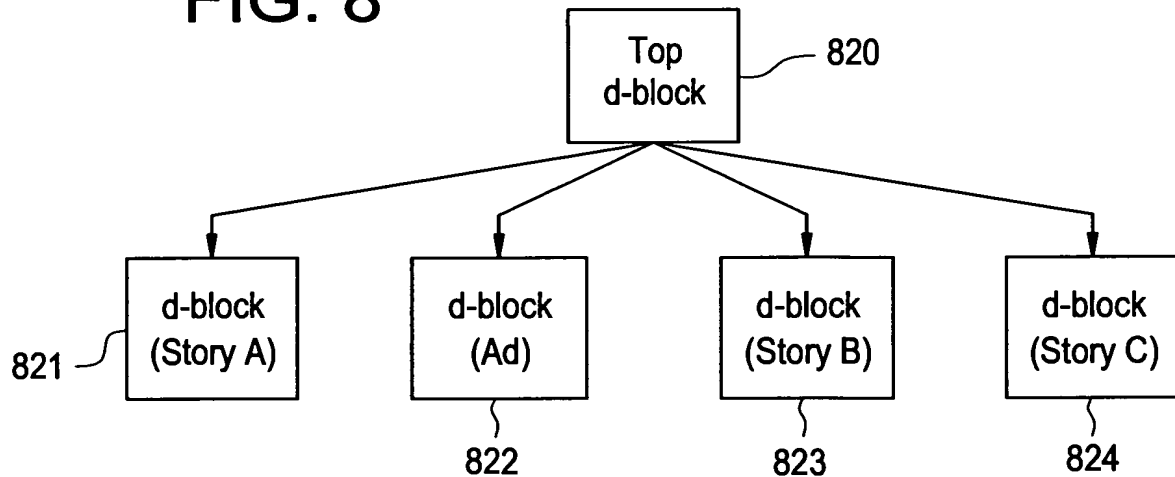
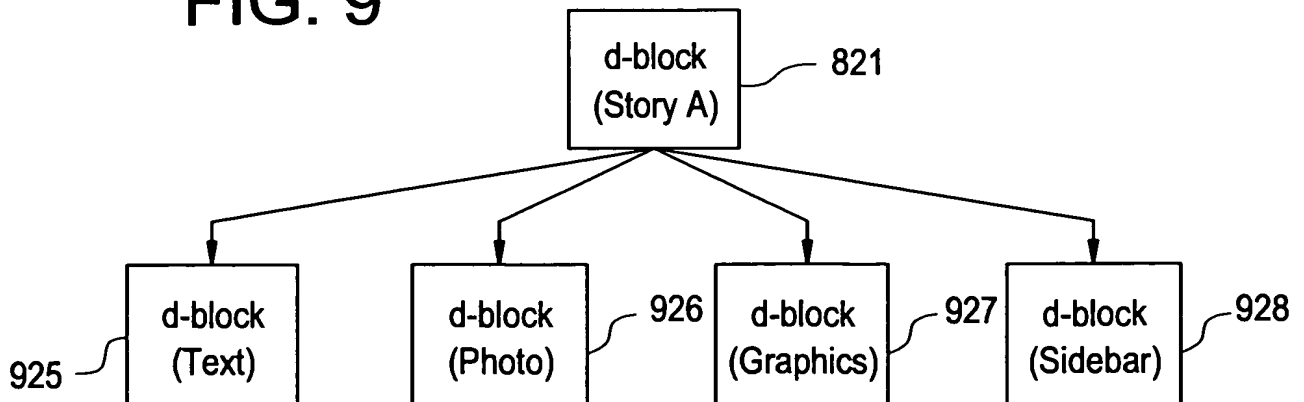


FIG. 9



6/13

FIG. 10

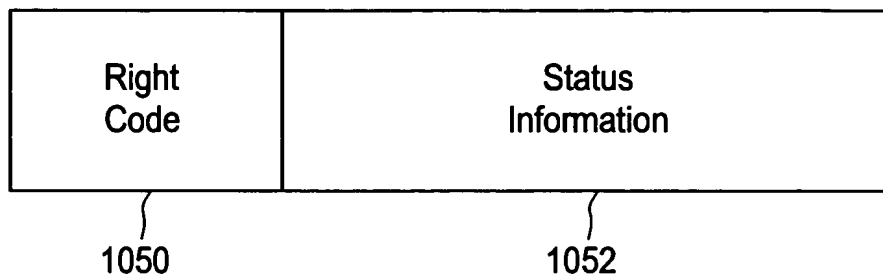
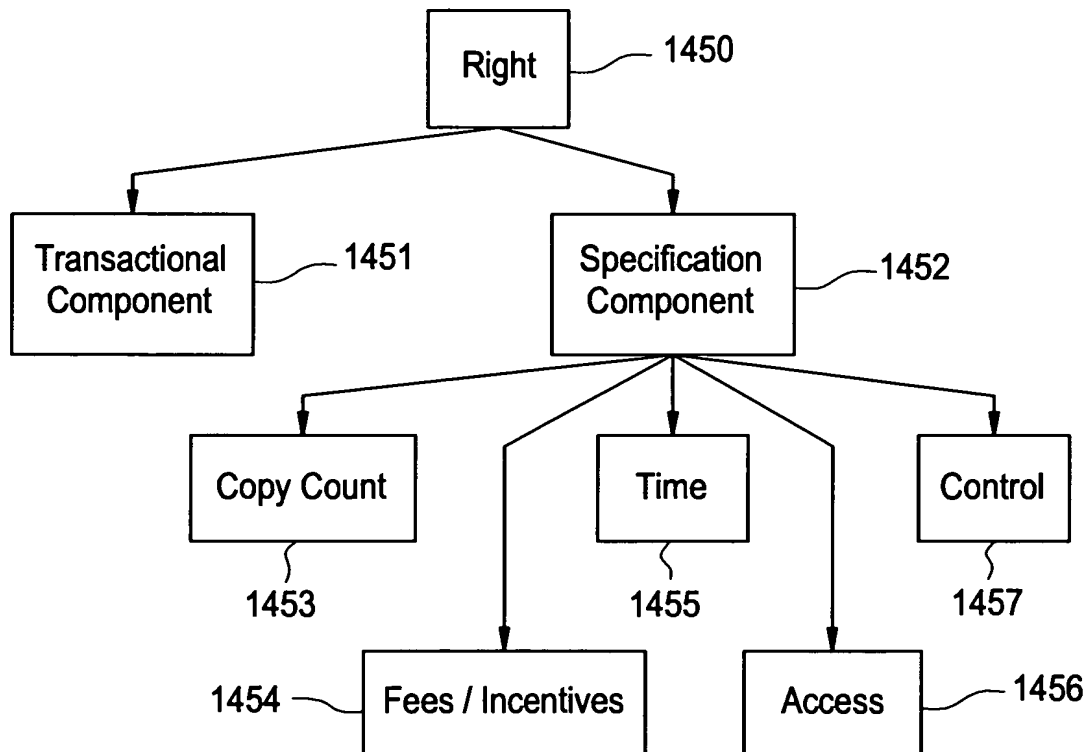
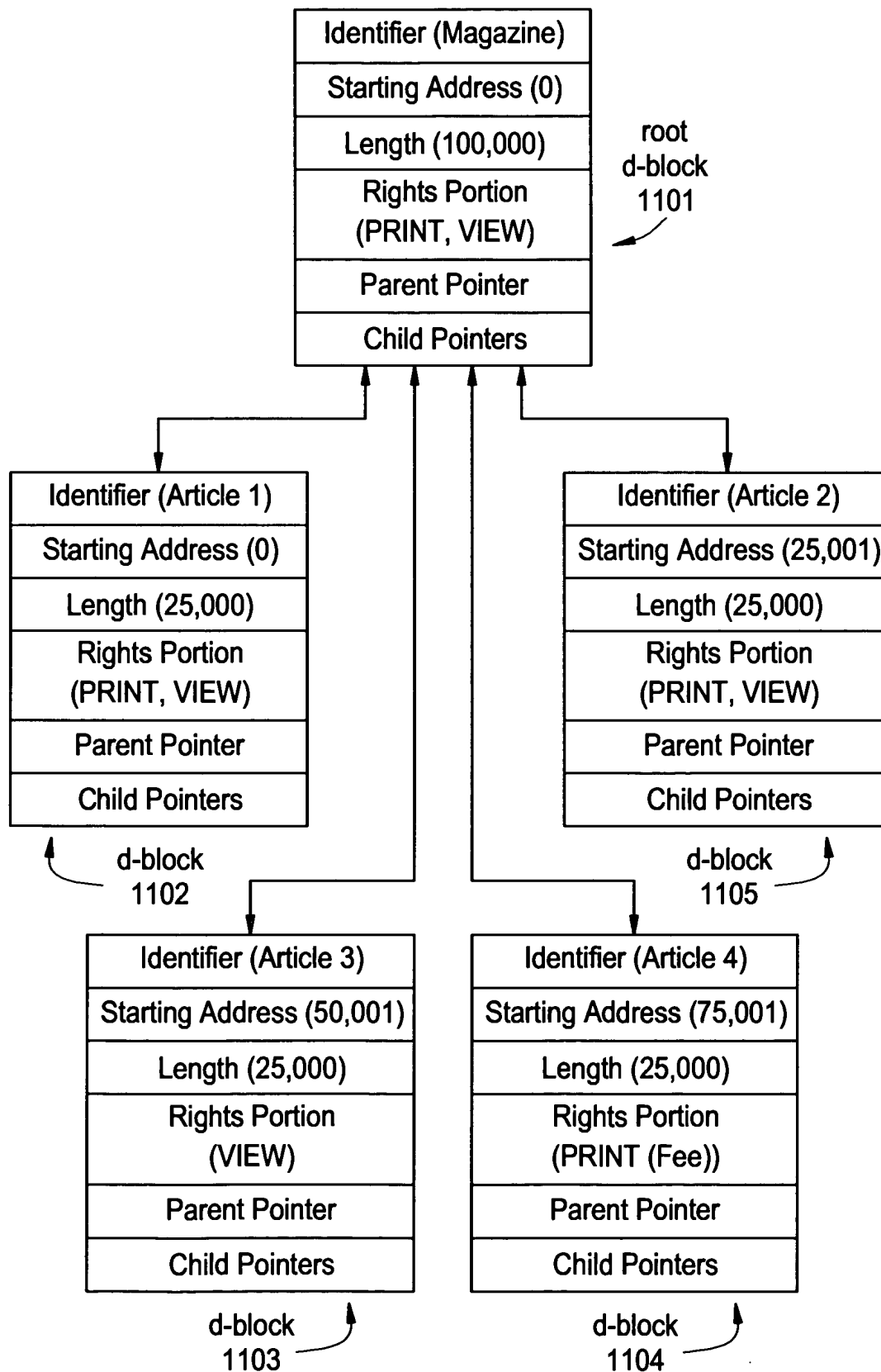


FIG. 14



7/13

FIG. 11

8/13

FIG. 12

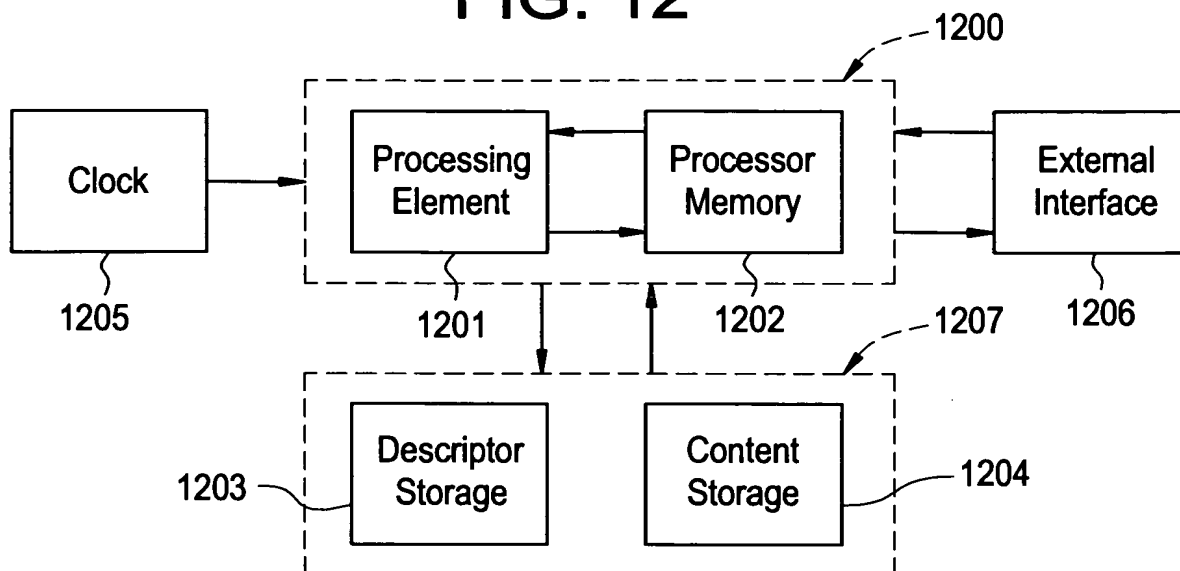


FIG. 13

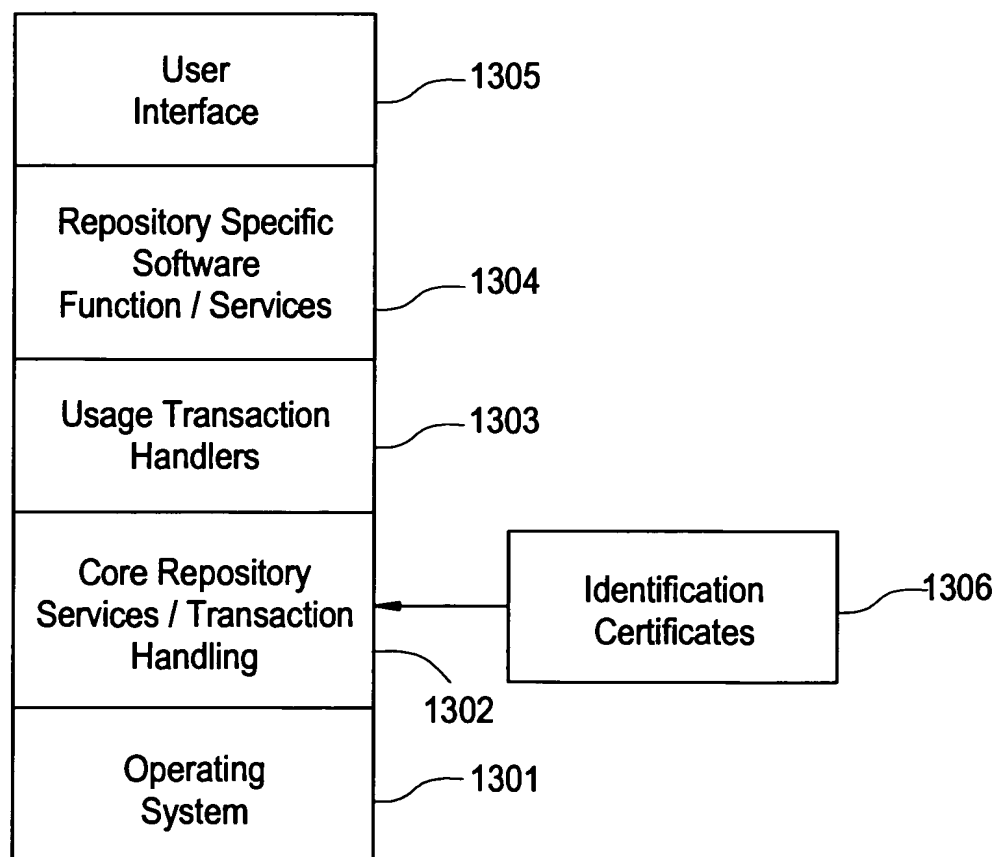


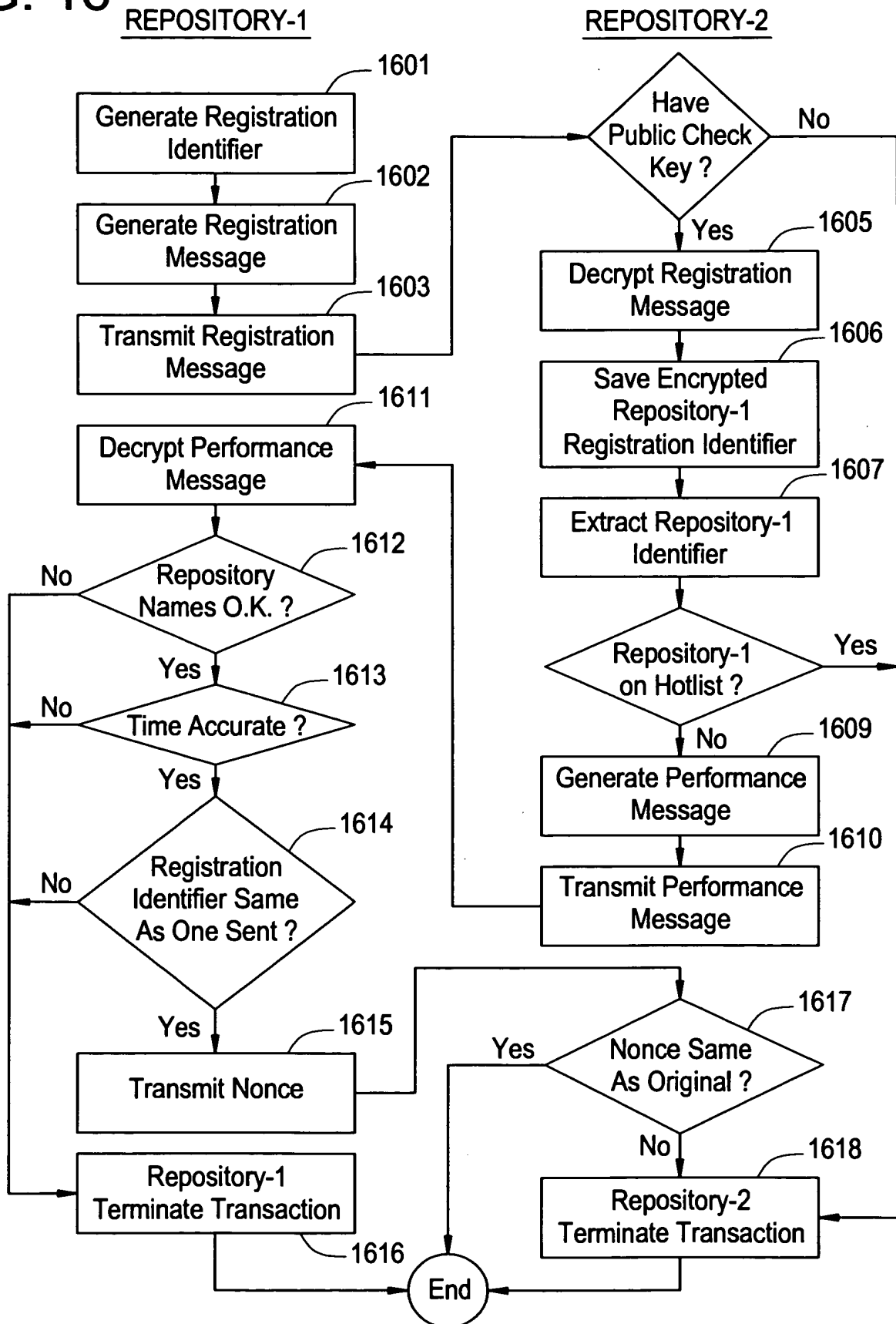
FIG. 15

9/13

- 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

FIG. 16

10/13



11/13

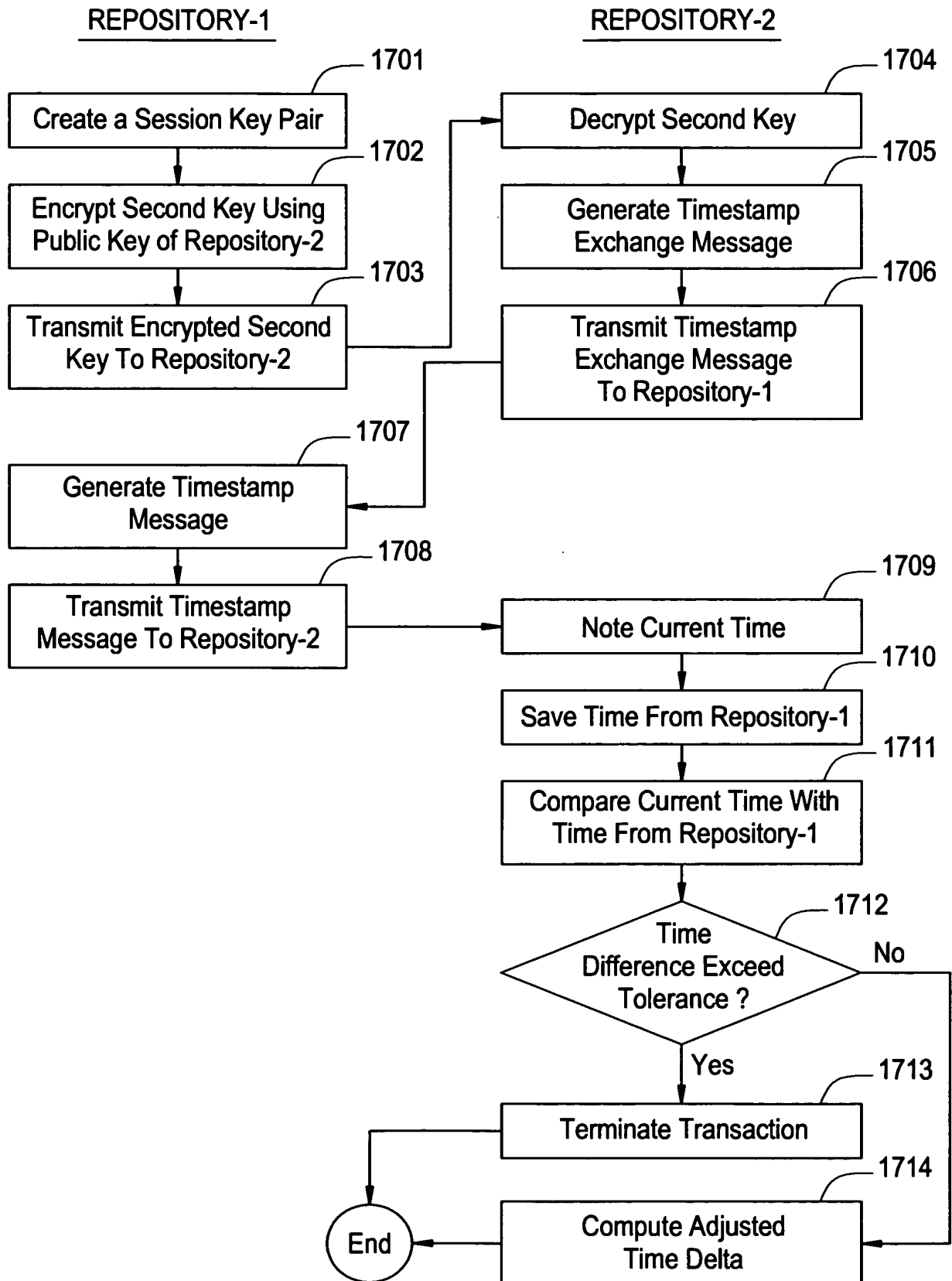
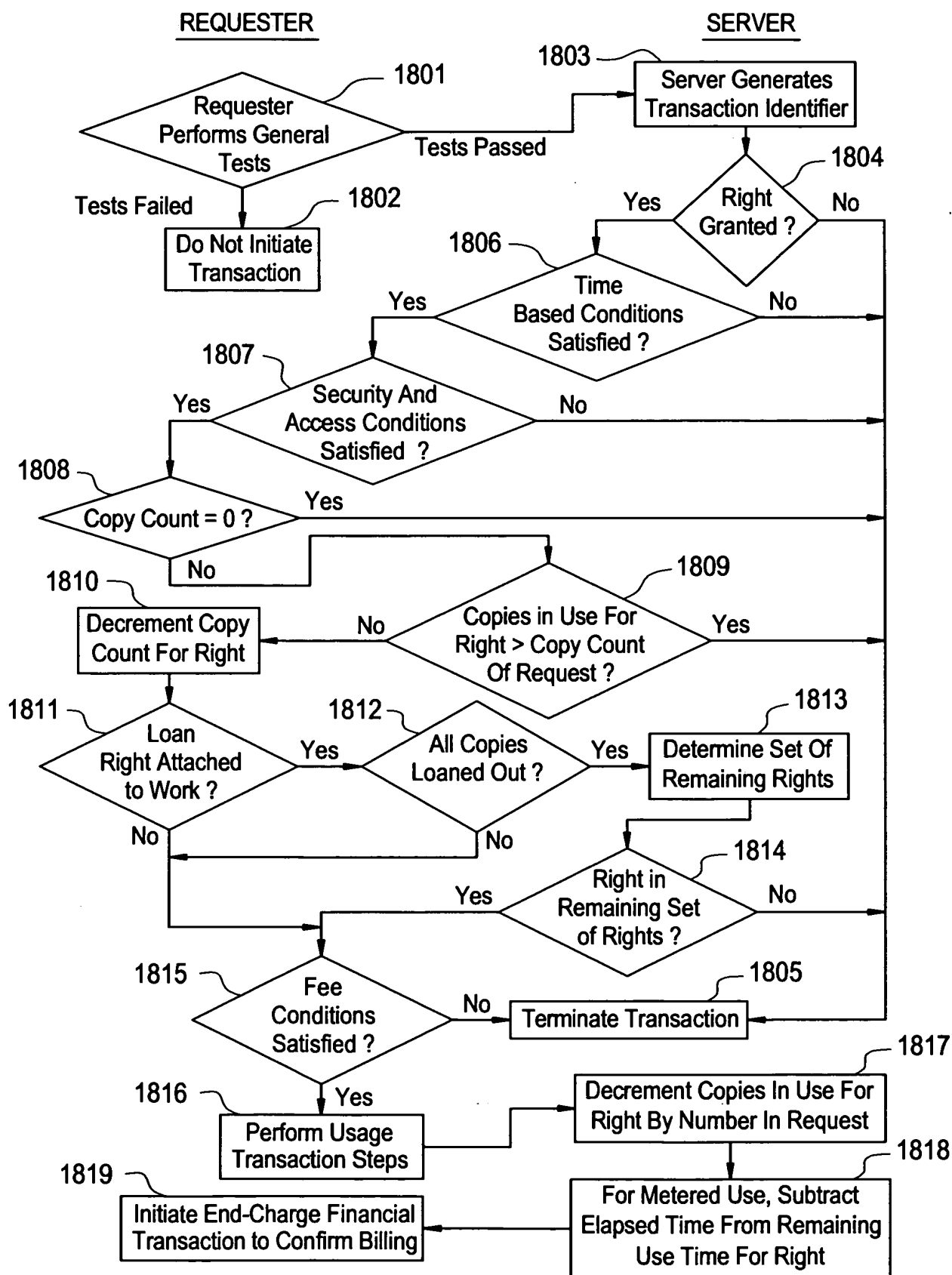
FIG. 17

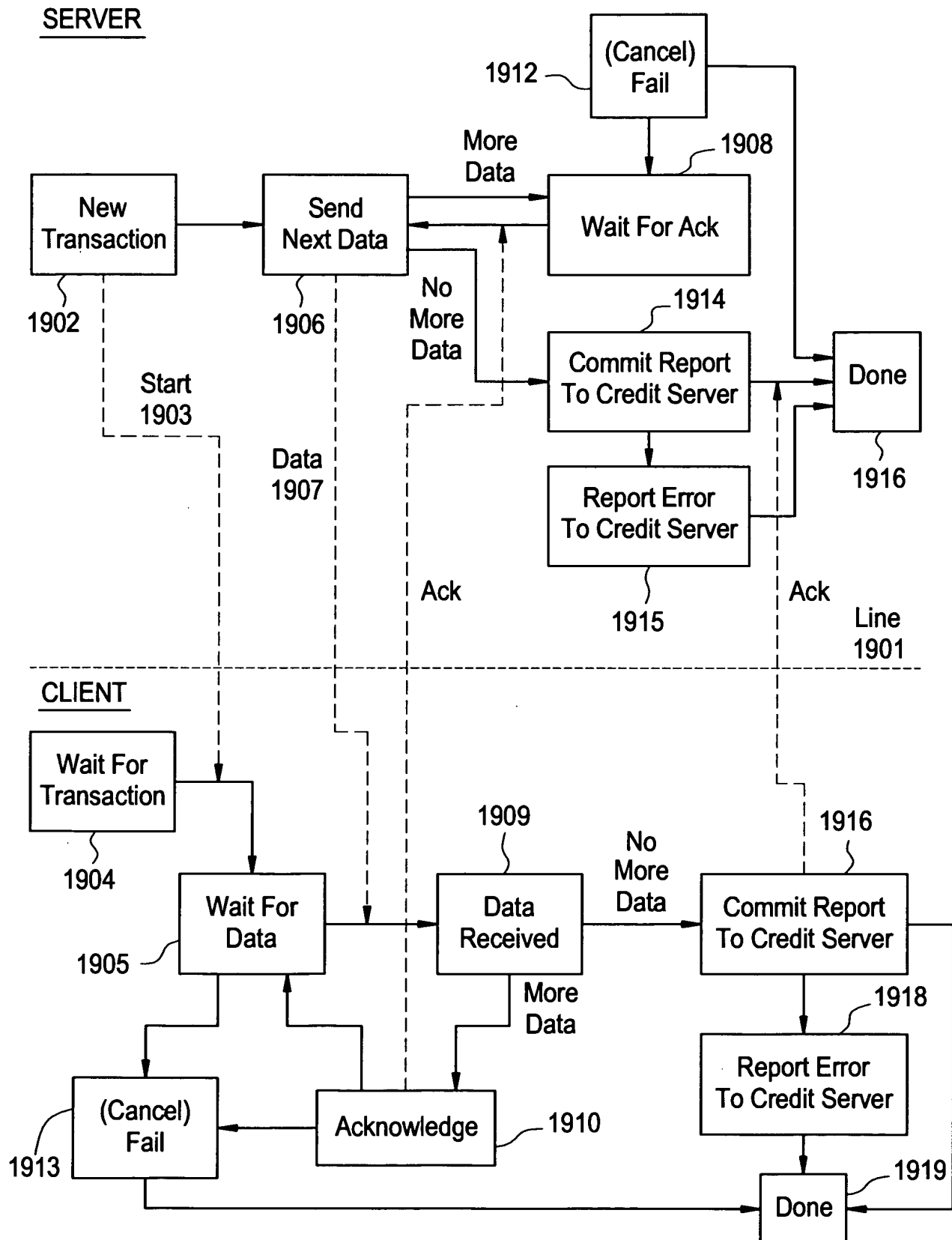
FIG. 18

12/13



13/13

FIG. 19



PATENT ABSTRACTS OF JAPAN

(D)

(11)Publication number : 06-230847
(43)Date of publication of application : 19.08.1994

(51)Int.Cl.

G06F 1/00
G06F 9/445
G06F 13/00

(21)Application number : 05-041922
(22)Date of filing : 04.02.1993

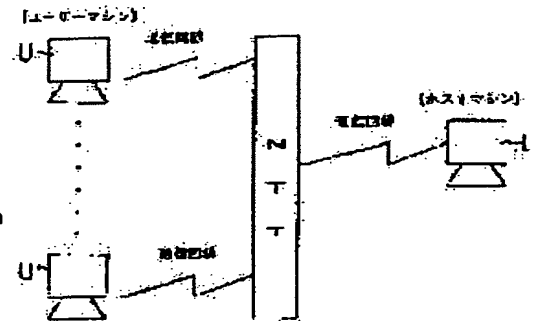
(71)Applicant : RAND SYST:KK
(72)Inventor : MASUI CHIYOICHI

(54) METHOD FOR LENDING EXPENSIVE COMPUTER SOFTWARE

(57)Abstract:

PURPOSE: To provide an expensive computer software lending method capable of inexpensively providing expensive computer software represented by an integrated business information system or the like to a user without reducing reliability or an advanced sense and easily recovering the development cost of a system developing company.

CONSTITUTION: The lending method is constituted of hardware consisting of a user machine and a host machine which are mutually connected through a communication line and software provided with a program file for processing various work, a means for measuring the real using time of a data file and the program file and a means for inhibiting the illegal use of the program file so that a file including user managing information is transmitted to check the validity of the file at the start time and the end time of use of work program file, the start time of use and the end time of use are written in the user managing information file and the real using time of a working program file is calculated based upon the time information to calculate an changing amount.



LEGAL STATUS

[Date of request for examination] 04.02.1993
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number] 2073327
[Date of registration] 25.07.1996
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平 6 - 2 3 0 8 4 7

(43) 公開日 平成 6 年 (1994) 8 月 19 日

(51) Int. Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G06F 1/00	370	F 7165-5B		
9/445				
13/00	351	Z 7368-5B		
		9367-5B	G06F 9/06	420 J

審査請求 有 請求項の数 6 F D (全 11 頁)

(21) 出願番号 特願平 5 - 4 1 9 2 2

(22) 出願日 平成 5 年 (1993) 2 月 4 日

(71) 出願人 3 9 3 0 0 3 9 7 4

株式会社ランドシステム

大阪府吹田市豊津町 9 番 4 4 号

(72) 発明者 増井 千代一

大阪府吹田市豊津町 9 番 4 4 号 株式会社
ランドシステム内

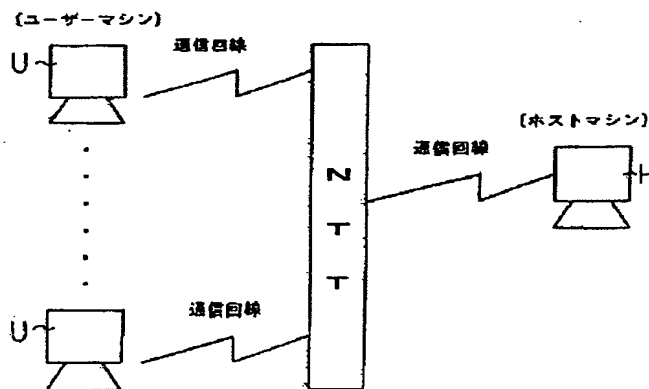
(74) 代理人 弁理士 柳野 隆生

(54) 【発明の名称】 高額コンピュータソフトレンタル方法

(57) 【要約】

【目的】 統合経営情報システム等に代表される高額コンピュータソフトを信頼感や高級感を落とすことなく、ユーザーに低価格で提供でき、しかもシステム開発会社の開発経費の回収も容易にした高額コンピュータソフトレンタル方法を提供せんとするものである。

【構成】 通信回線によって結ばれたユーザーマシンとホストマシンとからなるハードウェアと、各種業務処理するプログラムファイルと、データファイル並びに前記プログラムの実利用時間を計測する手段及び前記プログラムファイルの不正利用を禁止する手段を具備したソフトウェアとから構成され、業務プログラムファイルの利用開始時と利用終了時に、ユーザー管理情報を含むファイルを送信して当該ファイルの妥当性チェックを行うとともに、同時にユーザー管理情報ファイルに利用開始時刻及び利用終了時刻を書き込んでおき、この時刻情報に基づいて業務プログラムファイルの実利用時間を算出して課金額を計算する。



【特許請求の範囲】

【請求項 1】 ユーザーオフィスに設置された通信機能を装備したパソコンよりなるユーザーマシンと、コンピュータソフトの利用管理センターに設置された通信機能を装備したパソコン又は汎用コンピュータよりなるホストマシン、並びに前記ユーザーマシンと前記ホストマシンを繋ぐ通信回線とから構成されたハードウェアと、各種業務を処理するプログラムファイルと、データファイル並びに前記プログラムファイルの実利用時間を計測する手段及び前記プログラムファイルの不正利用を禁止する手段を具備したソフトウェアと、から構成され、その運用手順が、ユーザーが通信回線をオープンしたうえ、ユーザーマシンからホストマシンに対してユーザー管理情報を含む利用開始宣言ファイルを送信し、前記利用開始宣言ファイルを受信したホストマシンが、利用開始宣言ファイルの妥当性チェックを行い、妥当である場合にはユーザーマシンに対して利用許可指令を返信するとともに、ユーザーマシン又はホストマシンがこの利用開始時刻を記録した後、通信回線を遮断してなる利用開始手続工程と、利用許可指令を受信したユーザーがプログラムファイルの不正利用禁止手段を解除してユーザーマシンを用いて所定の業務を行う業務処理工程と、ユーザーが業務終了と同時に通信回線を再オープンし、ユーザーマシンからホストマシンに対して、利用終了宣言ファイルを所定手順に従って送信し、利用終了宣言ファイルを受信したホストマシン又はユーザーマシンがこの利用終了時刻を記録した後、通信回線を遮断してなる利用終了手続工程と、ホストマシン又はユーザーマシンが前記利用開始時刻と前記利用終了時刻とからプログラムファイルの実利用時間を計算し、当該実利用時間に基づいてユーザーに対する課金額を算出する課金額算出工程と、から構成される高額コンピュータソフトレンタル方法。

【請求項 2】 利用開始宣言ファイルに含まれるユーザー管理情報は、前回の利用終了宣言ファイルの内容が反映される構成となし、ホストマシンに対してユーザー管理情報を含む利用開始宣言ファイルを送信する前に予備処理として、ユーザーマシン内部で利用開始宣言ファイルの妥当性をチェックしてなる請求項 1 記載の高額コンピュータソフトレンタル方法。

【請求項 3】 ユーザー管理情報は暗号化してなる請求項 1 又は 2 記載の高額コンピュータソフトレンタル方法。

【請求項 4】 プログラムファイル及びデータファイルは全てユーザーマシンの記録媒体に格納させてなる請求項 1、2 又は 3 記載の高額コンピュータソフトレンタル方法。

【請求項 5】 処理内容に対応する複数のサブプログラムファイルを統合して構成される大規模プログラムファ

イルのうち、ユーザーマシンがその都度処理する該当業務に対応するジョブ別サブプログラムファイルを、必要に応じてホストマシンから通信回線を通じてユーザーマシンに送信してユーザーマシンに格納し、該当処理後はユーザーマシン上の該当サブプログラムファイルを消去してなる請求項 1、2 又は 3 記載の高額コンピュータソフトレンタル方法。

【請求項 6】 データファイルはユーザーマシンの記録媒体に格納し、他方、プログラムファイルのうち業務処理を担う本体プログラムファイルの全てをホストマシンの記録媒体に格納しておき、ユーザーマシンは通信回線を通じてホストマシンの記録媒体内の本体プログラムファイルにアクセスながら、業務をユーザーマシンの CPU によって処理してなる請求項 1、2 又は 3 記載の高額コンピュータソフトレンタル方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は統合経営情報システム等に代表される高額コンピュータソフトの通信回線を用いたレンタル方法に関する。

【0002】

【従来の技術】 例えば、財務分析、利益計画、予算編成、財務予想、業績評価等を行うことができる、いわゆる統合経営情報システムは、経営計画を立案するうえで強力な武器となることは経営者に広く認識されている。しかしながら、このような統合経営情報システムは膨大なプログラム量と高度な解析内容を有していることから極めて高価であり、例えば買取方式であると、通常 200 万円～500 万円程度の入手費用を要するのが普通である。

【0003】

【発明が解決しようとする課題】 200 万円～500 万円の金額は大企業にとっては負担不可能な金額ではないものの、中小企業にとっては極めて大きな負担であり、このことが経営計画立案上、極めて有益な本システムの導入を阻害する大きな要因となっている。また、統合経営情報システムの稼働は通常、一カ月に 1 回～2 回程度であり、常時稼働しないシステムに対して多額の費用を投入することに対する抵抗感も本システム導入の阻害要因となっている。

【0004】 また買い取り方式に代えてリース方式も考慮されるが、リース方式では月々の支払い金額の低減ははかれるものの、トータル費用においては買い取り方式と大差なく、しかもリース方式の場合はパソコン等のハードウェアも抱き合わせ購入することを前提にしてシステムが構築されている場合が多く、ユーザーが所持する既存のハードウェアが無駄になる問題がある。

【0005】 統合経営情報システムの導入を促進するには、統合経営情報システムの販売価格を下げることも選択枝の一つとして残っているが、導入企業がよほど多く

ない限り、膨大な知識と労力を投入して開発した本統合経営情報システムの開発経費を回収することができない問題があり、更に、表面価格が低く設定されるとシステム全体に対する信頼感及び高級感が低下する問題もある。

【 0 0 0 6 】このような問題は統合経営情報システムに限らず、利用頻度の少ない高額コンピュータソフト、即ち日常的に利用しない高額コンピュータソフト一般に共通する問題である。本発明はかかる現況に鑑みてなされたものであり、統合経営情報システム等に代表される高額コンピュータソフトを信頼感及び高級感を落とすことなく、ユーザーに低価格で提供でき、しかもシステム開発会社の開発経費の回収も容易にした高額コンピュータソフトレンタル方法を提供せんとするものである。

【 0 0 0 7 】

【課題を解決するための手段】本発明者は上記課題を解決するにあたって、鋭意検討した結果、コンピュータソフトを実利用した時間分だけ費用請求するようにすれば、ユーザーの費用負担は軽減し、またこのようにすれば費用負担が軽減する結果、当該システムの導入企業数を飛躍的に増やすことが可能となり、システムの開発費用の回収も容易となる筈であるとの着想を得た。また実利用時間分だけの費用負担であるから、ユーザーの立場から見たときに無駄がなく、また高級感や信頼感を阻害する懸念もないと判断される。

【 0 0 0 8 】しかしながら、このような実利用分だけ費用請求する制度を可能にするためには、「実利用時間を如何に監視するか」という問題を解決しなければならない。本発明者は、この解決手段として、公衆通信回線又は総合デジタル通信回線網（以下、I S D N と称す）を用いてユーザーマシンの稼働状況を把握することを思いつき、この着想を更に具体化することによって本発明を完成させた。

【 0 0 0 9 】上記課題を解決した本発明は、ユーザーオフィスに設置された通信機能を装備したパソコンよりなるユーザーマシンと、コンピュータソフトの利用管理センターに設置された通信機能を装備したパソコン又は汎用コンピュータよりなるホストマシン、並びに前記ユーザーマシンと前記ホストマシンとを繋ぐ通信回線とから構成された「ハードウェア」と、各種業務を処理するプログラムファイルと、データファイル並びに前記プログラムファイルの実利用時間を計測する手段及び前記プログラムファイルの不正利用を禁止する手段を具備した「ソフトウェア」とから構成されている。そして、その運用手順は、利用開始手続工程、業務処理工程、利用終了手続工程、課金額算出工程とから構成され、それぞれの工程は次の内容を有している。

＜利用開始手続工程＞：ユーザーが通信回線をオープンし、ユーザーマシンからホストマシンに対して、ユーザー管理情報を含む利用開始宣言ファイルを送信し、前記

利用開始宣言ファイルを受信したホストマシンが、利用開始宣言ファイルの妥当性チェックを行い、妥当である場合にはユーザーマシンに対して利用許可指令を返信するとともに、ユーザーマシン又はホストマシンがこの利用開始時刻を記録した後、通信回線を遮断する。

＜業務処理工程＞：利用許可指令を受信したユーザーがプログラムファイルの不正利用禁止手段を解除してユーザーマシンを用いて所定の業務を処理する。

＜利用終了手続工程＞：ユーザーが業務終了と同時に通信回線を再オープンし、ユーザーマシンからホストマシンに対して、利用終了宣言ファイルを送信し、利用終了宣言ファイルを受信したホストマシン又はユーザーマシンがこの利用終了時刻を記録した後、通信回線を遮断する。

＜課金額算出工程＞：ホストマシン又はユーザーマシンが前記利用開始時刻と前記利用終了時刻とからプログラムファイルの実利用時間を計算し、当該実利用時間に基づいてユーザーに対する課金額を算出する。

【 0 0 1 0 】利用開始宣言ファイルに含まれるユーザー管理情報は、前回の利用終了宣言ファイルの内容に連動して規定される構成となして、ユーザー管理情報を不正利用防止手段として利用することも可能であり、この場合は、ユーザーマシンは、利用開始時にホストマシンに対してユーザー管理情報を含む利用開始宣言ファイルを送信する前段階処理として、ユーザーマシン内部で利用開始宣言ファイルの妥当性を予備チェックすることが好ましい。また、このユーザー管理情報は暗号化することがより好ましい。

【 0 0 1 1 】プログラムファイル及びデータファイルの格納態様は様々であるが、例えば、次の①～⑤の態様が考慮される。

①プログラムファイル及びデータファイルの全てをユーザーマシンの記録媒体に格納させる態様。

②処理内容に対応する複数のサブプログラムファイルを統合して構成される大規模プログラムファイルのうち、ユーザーマシンが処理する該当業務に対応するサブプログラムファイルを、必要に応じてその都度ホストマシンから通信回線を通じてユーザーマシンに送信してユーザーマシンに格納し、該当処理後はユーザーマシン上の該当サブプログラムファイルを消去する態様。

③プログラムファイルのうち、業務処理を担う本体プログラムファイルの全てをホストマシンの記録媒体に格納しておき、ユーザーマシンは通信回線を通じてホストマシンの記録媒体内の本体プログラムファイルに常時アクセスながら、業務をユーザーマシンのCPUによって処理する態様。

④プログラムファイルの全てをユーザーマシンに格納しておきデータファイルの一部又は全部をホストマシンに格納したり、あるいはプログラムファイルの全てをホストマシンに格納しておきデータファイルの一部又は全部

をホストマシンに格納する態様。

【 0 0 1 2 】

【作用】このような構成の高額コンピュータソフトレ
タル方法の運用手順の概略は次の如くである。ここで
は、業務プログラムファイル及び業務データファイルの
全てがユーザーマシンに格納されている態様を中心にし
て説明する。先ず業務開始に先立って、ユーザーは業務
プログラムファイル及び業務データファイルが格納され
たハードディスク又はフロッピーディスクを読み書き可
能な状態にセッティングした状態で、通信モデムを介し
てホストマシンとの通信回線を自動ダイヤリングによ
ってオープンして利用管理センターに設置されたホストマ
シンを呼び出す。

【 0 0 1 3 】次いでユーザーマシンからユーザー管理情
報を含む利用開始宣言ファイルを、ユーザーマシン内部
でユーザー管理情報の妥当性チェックを経たのち、ある
いは経ることなく所定手順に従って利用管理センターの
ホストマシンに送信する。ユーザーマシン自身がユーザ
ー管理情報の妥当性の予備チェックを行い、その結果に
不整合があると判断された場合には、この段階で警告メ
ッセージを画面表示したうえ回線遮断処理を行う。

【 0 0 1 4 】ユーザーマシン自身が行う予備妥当性チ
ェックを行わない場合や、予備妥当性チェックを問題なく
通過した場合は、次いでホストマシンによるユーザー管
理情報の妥当性チェックが行われる。このホストマシン
によるユーザー管理情報の妥当性チェックは、事前にユ
ーザーマシンサイドでユーザー管理情報の予備妥当性チ
ェックが行われている場合には簡易処理で充分である。

【 0 0 1 5 】ユーザー管理情報の妥当性チェックの結
果、その結果に不整合がない場合にはホストマシンから
利用許可指令をユーザーマシンに送信すると同時に、こ
の時刻を利用開始時刻として記録する。

【 0 0 1 6 】次いで利用許可指令を受信したユーザー
は、この利用許可指令に基づいて、ユーザーマシン上の
業務プログラムファイルの不正利用禁止手段を解除する
とともに利用管理センターとの通信回線を遮断したう
え、業務プログラムファイルを用いて所定の業務処理を
実行する。業務処理中はユーザーマシンとホストマシン
との間の通信回線は遮断されている。

【 0 0 1 7 】業務が終了したならば、ユーザーは通信回
線を再オープンし、ホストマシンに対して利用終了宣言
ファイルを所定手順に従って送信した後、回線を遮断す
る。他方、利用終了宣言ファイルを受信した利用管理セ
ンターは当該利用終了宣言ファイルを受信した時刻を利
用終了時刻として記録したうえ、前記利用開始時刻と前
記利用終了時刻とから業務プログラムファイルの実利用
時間をホストマシンによって計算し、当該実利用時間に
基づいてユーザーに対する課金金額を算出するものであ
る。

【 0 0 1 8 】尚、上述した説明では利用開始時刻及び利

用終了時刻の記録及び業務プログラムファイルの実利用
時間の計算をホストマシンによって処理しているが、こ
の処理はユーザーマシン側で行う場合もある。更に、回
線遮断動作もユーザーマシン側から行うのではなくホス
トマシン側から行うことも考慮される。

【 0 0 1 9 】このように本発明の高額コンピュータソフ
トレタル方法では、業務プログラムの利用開始時にホス
トマシンに通信回線を通じてアクセスして、ホストマシ
ンから受ける利用許可指令によって、ユーザーマシン上
の業務プログラムの利用が可能となり、その後は回線遮
断して、ユーザーマシンを用いて業務処理を行い、利用
終了時に再度、回線オープンしてホストマシンに対して
業務プログラムの利用終了を連絡するという一連の手順
を経るものである。そして、利用開始時刻と利用終了時
刻を記録することで業務プログラムの実利用時間を計算
し、この実利用時間に基づいて課金額を算出するもので
ある。

【 0 0 2 0 】

【実施例】次に本発明の詳細を図例の実施例に基づき説
明する。図 1 は本発明のシステム概念を示すネットワ
ーク図を示している。本ネットワークは、業務プログラ
ムを利用する企業のオフィスに設置されたユーザーマシ
ン U、U……と、業務プログラムの提供者である利用管
理センターに設置されたホストマシン H を、通信回線を
経由して接続した構成である。通信回線としては公衆回
線及び I S D N が利用可能であるが、サイズの大きいプ
ログラムファイルやデータファイルの授受を行わない限
り、通常は公衆回線の利用で充分対応できる。ユーザー
マシンは通信モデム機能を有するパソコンであって、業
務プログラムの運用が可能な能力を有するものであれば
任意のものが利用可能である。また、ホストマシンはい
わゆる汎用コンピュータ、ミニコンピュータ、ワークス
テーション、パソコン等が採用可能であり、具体的な処
理能力や規模は、ネットワークの規模や利用管理センタ
ーが担う業務内容によって適宜選択される。

【 0 0 2 1 】図 2 として示すものは、上記ハードウェア
構成に格納されるソフトウェアの概略構成を示すもので
あり、その構成は通常のアプリケーションソフトと同
様、プログラムファイル P とデータファイル D とから構
成されている。また、データファイル D 内には、業務デ
ータファイル以外に利用状況を管理するファイルを含む
ユーザー管理情報ファイルが内蔵されている。また、プ
ログラムファイルには不正利用を防止するための手段
（プロテクト）を具備させている。プログラムファイル
P 及びデータファイル D の格納場所をユーザーマシン U
のハードディスク上とするか、あるいはホストマシンの
ハードディスク又は磁気テープとするかは、本システム
の運用態様によって様々な形態が考えられる。

【 0 0 2 2 】例えば、これらの具体的な態様は、「表
1」で示され、パターン(1)～(12)の 12 種類の態様が

考えられる。
【0023】

【表1】

プログラムファイルの 格納態様 データファイルの 格納態様	全て ユーザ ーマシン に格納	基本 的にホ ストマ シンに 格納し てお く	基本 的にホ ストマ シンに 格納し てお く
全てユーザマシンで管理	パターン (1)	パターン (2)	パターン (3)
全てホストマシンで管理	パターン (4)	パターン (5)	パターン (6)
ユーザマシンとホストマシンで 分散管理	パターン (7)	パターン (8)	パターン (9)
ユーザマシンとホストマシンで 同時管理	パターン (10)	パターン (11)	パターン (12)

【0024】ここでは、データファイルの全てをユーザ
ーマシンに格納しながら、プログラムファイルの格納形
態として種々の形態を採用したもの（パターン(1)～
(3)）を説明対象とし、この中でも特に、プログラムフ
ァイル及びデータファイルの全てをユーザマシンに格
納した場合（パターン(1)）を中心にして説明する。

【0025】パターン(1)のシステム構成は図3に示す
ように、ユーザマシン側に設置したハードディスク等
の記憶媒体内に、プログラムファイルとデータファイル
の両方が格納されている。プログラムファイルは、財務
分析プログラム、利益計画プログラム、予算編成プログ
ラム、財務予想プログラム、業績評価プログラム等のジ
ョブ別サブプログラム群から構成され、それぞれの該当
ジョブを処理する度に、各サブプログラムを主メモリ
上に常駐させるようにしている。このシステムでは、ユ
ーザマシンとホストマシン間のデータ授受は利用開始
時と利用終了時の2回だけであり、プログラムファイル
の全てが予めユーザマシンに格納されているか、ある
いはユーザ側が保管していることが前提である。

【0026】データファイルには、図4で示される項目
内容を有する「利用状況ファイル」が組み込まれてお
り、この「利用状況ファイル」を業務プログラムの利用

開始時と利用終了時に編集することによって、ホストマ
シンとの通信内容を反映させ、この通信内容が反映され
た「利用状況ファイル」をチェックすることにより業務
プログラムファイルの不正利用防止がはかれるように工
夫している。そして、この「利用状況ファイル」の編集
及びチェックを伴いながら行われるソフトレンタルの具
体的手順は、図5で示されるメニュー構成図によって表
現される。

【0027】次に「利用状況ファイル」の主要項目につ
いて説明する。「ユーザーID」はユーザー識別用のID
であり、「商品名」は利用対象となる業務プログラム
ファイル名である。電送区分は、利用開始手続きを正常
終了したならば「1」に変更し、他方、利用終了手続き
を正常終了したならば「2」に変更するものであり、利
用開始手続きを行うときに、この内容をチェックしてそ
の内容が最初から「1」となっていれば、前回の利用終
了手続きが正常終了していないと判断し、また利用終了
手続きを行う際に、その内容が既に「2」となっていれ
ば、この場合は利用開始手続きが正常終了していないと
判断するものであり、いずれの場合もエラーメッセージ
を発して回線を遮断するものである。

【0028】「利用開始日付」、「利用開始時間」はユ

ユーザーマシンからホストマシンにアクセスして利用開始
手続を行う際に書き込まれ、他方、「利用終了日付」、
「利用終了時間」はユーザーマシンからホストマシンに
再アクセスして利用終了手続を行う際に書き込まれるも
ので、いずれもユーザーマシンのシステム日付・時間を
参照して自動的に書き込まれる。そして、「利用開始日
付・時間」と「利用終了日付・時間」との前後関係のチ
ェックを行い、「利用終了日付・時間」<「利用開始日
付・時間」の場合又は「利用終了日付・時間」>「シス
テム日付・時間」の場合には、いずれの場合も論理的な
妥当性に欠くとの判断から不正利用のおそれがあるとし
て回線遮断を行うようにしている。

【0029】「チケット終了日付」、「チケット終了時
間」、「チケット利用回数」、「チケット利用時間」
は、業務プログラムファイルの利用権をチケット購入と
いう形態で予め確保しておく手法を採用した場合に機能
する項目であり、この場合、チケット1枚で利用できる
時間が、例えば「チケット1枚で5時間の連続利用可
能」というように予め定められており、この換算率に基
づいて前記各項目に該当する時間が書き込まれる。

【0030】前記各項目内容はユーザーが、その内容を
容易に改変できないようにするために、暗号化処理を施
した上で書き込まれるように工夫されている。暗号化処
理の手法は公知の手法が採用可能であり、全ての項目内
容を暗号化することや主要項目のみを暗号化すること等
が採用できる。上記した「利用状況ファイル」の内容は
一例に過ぎず、他の態様も勿論考慮されるが、重要な
のは「利用状況ファイル」の一部に実質的に不正利用禁
止手段の機能を担うプロテクト情報が含まれていること
であり、且つこの情報が平文ではなく暗号化された状態
で含まれていることである。尚、ここで説明した不正利
用禁止手段に代えて他の不正利用禁止手段を採用すること
や他の不正利用禁止手段を併用することもできる。

【0031】本発明の高額コンピュータソフトレンタル
方法の運用手順の一例を、当該レンタルシステムを管理
する制御ソフトのメニュー画面（図5参照）の表示内容
に照らして説明すると次の如くである。

【0032】メニューは大別して、「1. 利用開始宣言
ジョブ」、「2. 利用終了宣言ジョブ」と「チケット関
連ジョブ」（「3. チケットによる利用宣言ジョブ」、
「チケット購入ジョブ」）及び「5. ファイル整理ジョ
ブ」から構成されている。このメニュー画面は特定操作
によってハードディスクから立ち上げることも可能であ
るが、不正利用をより確実に防止する観点からは、これ
ら起動メニューはフロッピーディスクベースで供給し、
フロッピーをセッティングして電源ON状態とすること
により、自動的に起動メニューが立ち上がるようにして
おき、且つ起動メニューが立ち上がったならば、所定の
キー操作以外は受け付けられないようにして汎用オペレー
ションシステム（OS）におけるプロンプトモードに復帰

しないようにしておくことが望まれる。

【0033】「チケット関連ジョブ」は本システムの利
用権をチケット形式で購入しておき、実利用時間をチケ
ット消費枚数に換算して課金する方式を採用する場合に
必要となるジョブであるが、ここではチケット方式に依
らず、実利用時間に対応して算出される課金額をユーザ
ーに対して直接請求する方式を例にして説明する。ユー
ザーからの支払いの具体的形式は様々であり、例えば予
め保証金を供託しておいた上で毎月の使用料を銀行振込
したり、あるいは自動引き落としすること、あるいはク
レジット会社との提携により処理すること等が考慮され
る。

【0034】業務プログラムの利用は次の手順に従って
進められる。先ず業務開始に先立って、ユーザーは業務
プログラムファイル及び業務データファイルが格納され
たハードディスク又はフロッピーディスクを読み書き可
能な状態で用意しておく。この状態で、起動プログラム
を実行して、ディスプレイに起動メニューを呼び出し、
次いで「1. 利用開始宣言ジョブ」を選択し、図6で示
されるような「利用開始宣言画面」を表示させる。

【0035】画面指示に従って、パスワード入力、利用
を希望する業務プログラム（図例のものは、商品名「F
O C A S」のみを表示した状態を示している。）等を入
力する。必要項目の入力が完了すると、ユーザーマシン
は図4で示された内容の「利用状況ファイル」の仮作成
を行う。この仮作成は、ホストマシンへのアクセスが初
アクセスの場合には、新規ファイルの作成によって行わ
れ、他方、初アクセスでない場合には、前回の利用終了
時に編集更新した「利用状況ファイル」を参照して「利
用状況ファイル」の仮作成を行う。この仮作成は「利用
状況ファイル」における日時管理項目等、各項目相互間
に矛盾がないか否かの妥当性チェックを行いながら実行
し、妥当性チェックをクリアーした場合のみ「利用状況
ファイル」の仮作成が完了するように工夫されている。

「利用状況ファイル」の主要項目の内容は暗号化された
形態で記録されているので、前妥当性チェックは暗号情
報を復号しながら行われる。暗号化された「利用状況フ
ァイル」の主要項目はユーザーにとっては解読できない
から、ユーザーが「利用状況ファイル」を改竄すること
は不可能であり、業務プログラムファイルを不正利用す
ることはできない。

【0036】「利用状況ファイル」の仮作成が無事終了
すると、ホストマシンとの通信回線（公衆回線）を自動
ダイヤリングによってオープンして利用管理センターに
設置されたホストマシンを呼び出す。

【0037】次いで仮作成した「利用状況ファイル」を
ホストマシンに送信する。ホストマシンは受信した「仮
の利用状況ファイル」の妥当性チェックを行い、その内
容が正常であると判断したならば、その旨をユーザーマ
シンに通知し、ユーザーマシンは「仮の利用状況ファ

ル」を正式の「利用状況ファイル」として記録する。ホストマシンによる妥当性チェックは簡易な内容で充分であり、省略することも可能である。またこれとは反対にホストマシンによる妥当性チェックをより厳格にすることも可能であり、例えば、前回の利用終了時にユーザーマシンが作成した「利用状況ファイル」のコピーをホストマシン内部にも保存しておき、この情報を参照して、

10 次回の利用開始時にユーザーマシンから送信されてくる「仮の利用状況ファイル」の妥当性チェックを行ってもよい。

【0038】ユーザーマシンに「正式の利用状況ファイル」が記録されたならば、ユーザーマシンは利用管理センターから業務プログラムの利用許可が下りたものと判断して、業務プログラムにかけられていたプロテクトを解除するとともに、その時刻を利用開始時刻として「利用状況ファイル」に書き込み、その直後に通信回線を遮断する。尚、ユーザーマシン側で行われる「利用状況ファイル」の妥当性チェックで不正利用と判断された場合は、利用管理センターへの自動ダイヤリングは行われ

20 ず、また、ホストマシン側で行われる「利用状況ファイル」の妥当性チェック工程において不正利用と判断された場合には、その段階で通信回線が遮断される仕組みになっている。

【0039】業務プログラムファイルのプロテクトが解除されたならば、ユーザーは当該業務プログラムを用いて自由に業務処理を行う。この業務処理時間は様々であるが、統合経営情報システム等の場合は4時間～6時間程度である。業務処理中はユーザーマシンとホストマシン間の通信回線は遮断されているので業務プログラム稼働中の通信費は不要である。

30 【0040】業務処理が完了したならば、起動時のメニューに復帰し、「2. 利用終了宣言ジョブ」を選択する。「2. 利用終了宣言ジョブ」が選択されたならば、「利用終了宣言画面」が表示されるので、必要項目を画面指示にしたがって入力し、「利用状況ファイル」に利用終了時刻を書き込むとともに、「利用状況ファイル」の編集更新を行った後、利用開始時と同様、「利用状況ファイル」の妥当性チェックを行い、次いで利用管理センターに対して自動ダイヤリングを行って回線オープンし、ホストマシンに「利用状況ファイル」を送信する。

40 「利用状況ファイル」を受信したホストマシンは「利用状況ファイル」の妥当性チェックを行ったうえ、その結果をユーザーマシンに通知し、ユーザーマシンは当該通知を受信したことを確認したうえ回線遮断する。

【0041】このようにして、ユーザーマシンとホストマシン間で業務プログラムの利用開始時及び利用終了時に「利用状況ファイル」の授受を行い、そして「利用状況ファイル」に含まれる時間情報に基づいて実利用時間を算出し、この実利用時間を基礎として課金額を算出するものである。

【0042】本発明の代表的実施例であるパターン(1)の運用手順は上記の如くであるが、他の実施形態も考慮される。例えば、図7で示したものは「表1」中のパターン(2)の実施形態を示すシステム概念図である。この実施形態では、データファイルはユーザーマシンUに格納するが、プログラムファイルはホストマシンHに格納しておく。そして、処理内容に対応する複数のサブプログラムファイルを統合して構成される大規模プログラムファイルのうち、ユーザーマシンがその都度処理する該

10

20

30

40

50

50 当業務に対応するサブプログラムファイルを、必要に応じてホストマシンから通信回線を通じてユーザーマシンに送信してユーザーマシンのハードディスクに格納し、該当処理後はユーザーマシン上の該当サブプログラムファイルを消去するという方式を採用するものである。この方式によれば、本来、社外秘とすべき財務データ等は、企業内部で完全管理しながら、汎用の業務プログラムファイルのみを利用管理センターから、その都度入手して運用することができる。したがって、ユーザーマシンでは動かすことが困難な大規模プログラムでも、ジョブ別サブプログラム単位で動かすことができるとともに、各ジョブ別サブプログラムはその都度ホストマシンからダウンロードするものであるから、最新のプログラムを利用することができ、バージョンアップ等のメンテナンスも極めて容易となる。尚、この場合、通信回線を介して授受されるプログラムファイルのサイズが小さい場合は公衆回線でも対応可能であるが、プログラムサイズが大きい場合には、通信回線としてISDNを用いることが好ましいことはいうまでもない。

【0043】図8はパターン(3)の実施形態を示すシステム概念図である。この実施形態でも、パターン(2)と同様、データファイルはユーザーマシンに格納するが、プログラムファイルはホストマシンに格納しておく。パターン(3)がパターン(2)と異なるのは、ユーザーマシンがジョブ別サブプログラムを実行する際に、パターン(2)のように該当サブプログラムをユーザーマシン側のハードディスクにダウンロードすることなく、ホストマシンのハードディスクに直接アクセスするようにしたことであり、ホストマシンにユーザーマシンのリモートコンピュータとしての機能を担わせたことである。このような方式を採用すれば、ユーザーマシンのハードディスクでは格納できないような大規模なサブプログラムでも実行することが可能となり、より高度な業務プログラムの提供が可能となる。

【0044】図9はパターン(6)の実施形態を示すシステム概念図である。この実施形態でも、プログラムファイル及びデータファイルの全てをホストマシンのハードディスクに格納する方式である。この場合、ユーザーマシンはホストマシンの端末機としての機能を担うことになり、より高度な業務処理が可能となるとともに、プログラムファイル及びデータファイルの全てがホストマシ

に蓄積されていることから、利用管理センターが各種の解析サービスや経営上の指導サービスを行うことも可能である。但し、この場合、ユーザー企業と利用管理センターとの間でデータファイルの利用に関する契約を締結しておく必要があることはいうまでもない。

【0045】図10はパターン(12)の実施形態を示すシステム概念図である。この実施形態では、プログラムファイル及びデータファイルの全てをユーザーマシンのハードディスクに格納しておくとともに、同じファイルをホストマシンのハードディスクにも格納しておき、一方のファイルを更新したときには、他方のファイルも更新する方式である。この方式を採用した場合は前記パターン(6)で述べた作用効果に加えて、何らかの理由でユーザーマシン上のデータファイルが消失した場合に、そのデータファイルの修復作業を担うバックアップセンターとしての機能を利用管理センターに担わせることもできる。

【0046】

【発明の効果】本発明によれば、ユーザーは実利用時間に対する利用代金を支払うだけで有益なコンピュータソフトを利用できるので、統合経営情報システムに代表されるような高額コンピュータソフトを、ソフトの信頼感及び高級感を低下させることなく、多数の企業に普及させることができ、ユーザーに多くの恩恵を与えるとともに、他方、システム開発会社にとっては開発経費の回収が容易となる。

【0047】また、実利用時間を管理する手法として、利用開始時と利用終了時に利用管理センターとの通信回線をオープンしてその時刻を記録する方式を採用したので、実利用時間の把握が正確に行える。また、ソフトウェアには不正利用禁止手段が具備されているから不正使用も防止できる。そして、特にユーザーマシンにプログラムファイル及びデータファイルの全てを格納した場合は、ユーザーマシンとホストマシン間の通信は、利用開始時と利用終了時において極めて短い時間だけ行われるものに過ぎないから、通信費用は極めて安価である。

【0048】また、利用開始宣言ファイルに含まれるユーザー管理情報は、前回の利用終了宣言ファイルの内容に連動して規定される構成となし、ユーザーマシンは、利用開始時にホストマシンに対してユーザー管理情報を含む利用開始宣言ファイルを送信する前段階処理として、ユーザーマシン内部で利用開始宣言ファイルの妥当性をチェックするようにした場合は、ホストマシン側で行う妥当性チェックは極めて簡易なものとしたり、あるいは省略することも可能となり通信時間が短くなって利用管理センターの負担が軽減される。

【0049】処理内容に対応する複数のサブプログラムファイルを統合して構成される大規模プログラムファイルのうち、ユーザーマシンが処理する該当業務に対応するサブプログラムファイルを、必要に応じてホストマシンから通信回線を通じてユーザーマシンに送信してユーザーマシンに格納し、該当処理後はユーザーマシン上の該当サブプログラムファイルを消去するようにしたときには、本来、企業外部に洩れることが好ましくない財務データ等を、企業内部で管理しながら、汎用の業務プログラムファイルのみを利用管理センターから、その都度入手して運用することができる。したがって、ユーザーマシンでは動かすことが困難な大規模プログラムでも、ジョブ別サブプログラム単位で動かすことができるとともに、各ジョブ別サブプログラムはその都度ホストマシンからダウンロードするものであるから、最新のプログラムを利用することができ、バージョンアップ等のメンテナンスも極めて容易となる。

【0050】また、プログラムファイルのうち、業務処理を担う本体プログラムファイルの全てをホストマシンの記録媒体に格納しておき、ユーザーマシンは通信回線を通じてホストマシンの記録媒体内の本体プログラムファイルにアクセスながら、業務をユーザーマシンのCPUによって処理した場合は、ユーザーマシンのハードディスクでは格納できないような大規模なサブプログラムでも実行することが可能となり、より高度な業務プログラムの提供が可能となる。

【図面の簡単な説明】

【図1】 本発明のシステムの概念を示す説明図

【図2】 ソフトウェア構成の概念を示す説明図

【図3】 本発明の代表的実施例であるパターン(1)のシステムの概念を示す説明図

【図4】 利用状況ファイルの主要項目を示す説明図

【図5】 起動メニューの主要内容を示す説明図

【図6】 利用開始宣言ジョブを実行画面を示す説明図

【図7】 本発明の他の実施例であるパターン(2)のシステムの概念を示す説明図

【図8】 本発明の他の実施例であるパターン(3)のシステムの概念を示す説明図

【図9】 本発明の他の実施例であるパターン(6)のシステムの概念を示す説明図

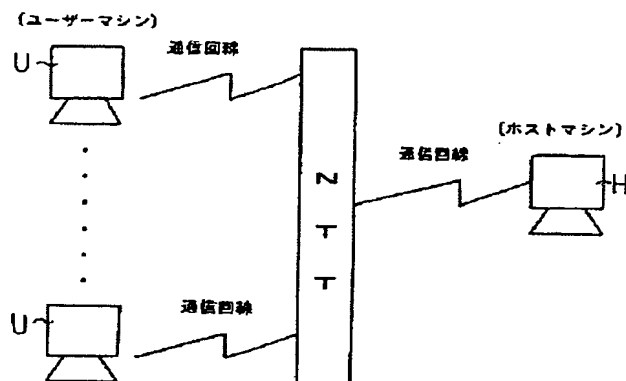
【図10】 本発明の他の実施例であるパターン(12)のシステムの概念を示す説明図

【符号の説明】

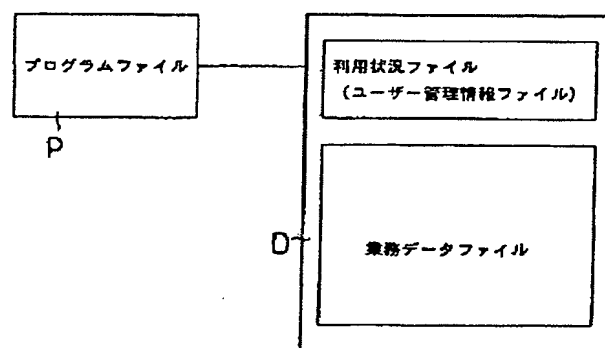
U ユーザーマシン

H ホストマシン

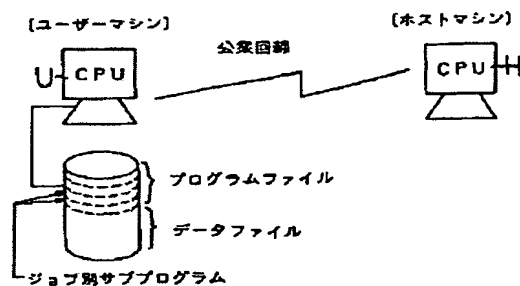
【図 1】



【図 2】



【図 3】



【図 4】

<利用状況ファイルの主要項目>

項 目	内 容
・ユーザー ID	初期設定したユーザー ID
・商品名	画面より選択された商品名
・現在使用仮票 No.	西暦の下 2 桁 + 利用区分ファイルの自動発番用最終 No. + 1
・電送区分	1
・利用区分	1
・使用開始日付	システム日付
・使用開始時間	システム時間
・使用終了日付	システム日付
・主要終了時間	システム時間
・チケット終了日付	Z E R O
・チケット終了時間	Z E R O
・チケット利用回数	Z E R O
・チケット利用時間	Z E R O
・アクセス回数	Z E R O
・アクセス回数累計	未編集
・有効期限	未編集
・プロテクトコード	未編集
・再送 SW	0

【図 6】

(画面イメージ)

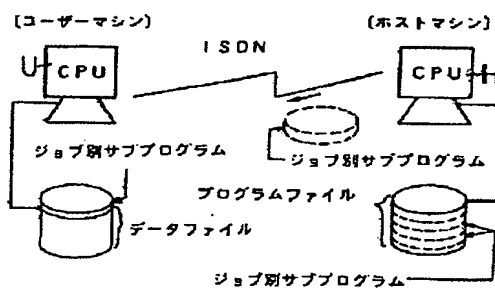
利用開始宣言画面 1992.12.20

PASSWORD XXXXXXXX

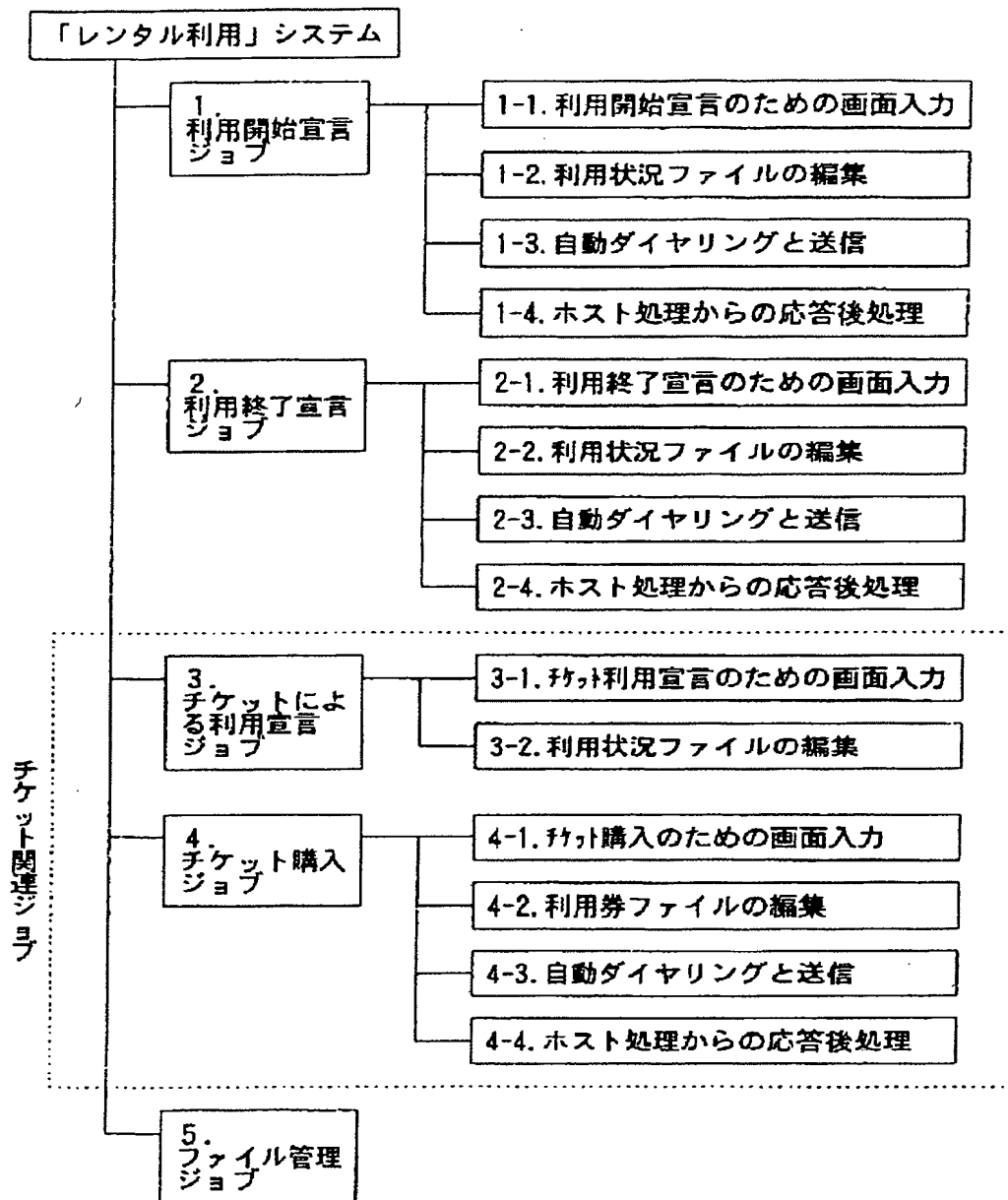
	商品名	チケット
()	1 FOCAS	0 回
	...	...
	...	...
(_)	10	

ソフトレンタルするソフトウェアに (Y) を入力 確認 _

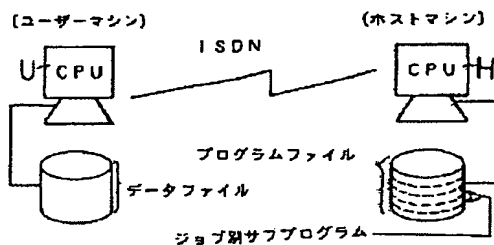
【図 7】



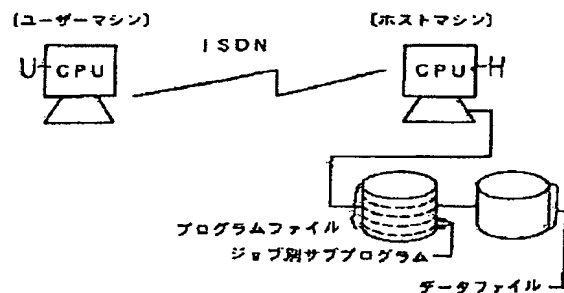
【図 5】



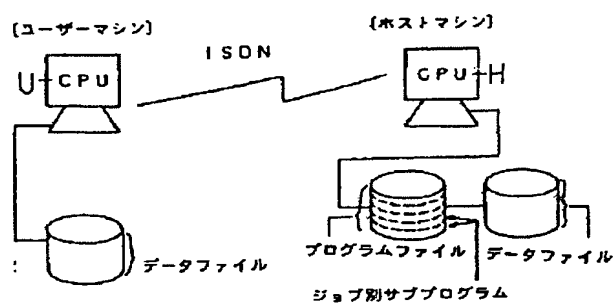
【図 8】



【図 9】



【図 10】



PATENT ABSTRACTS OF JAPAN

(11)Publication number : 03-014109
 (43)Date of publication of application : 22.01.1991

(51)Int. Cl. G06F 1/00
 G06F 9/06

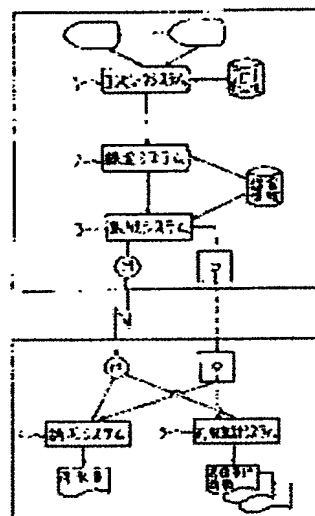
(21)Application number : 01-151569 (71)Applicant : NEC SOFTWARE LTD
 (22)Date of filing : 13.06.1989 (72)Inventor : NARITA KIYOTAKE

(54) CALCULATING SYSTEM FOR EXPENSE OF SOFTWARE

(57) Abstract:

PURPOSE: To expand the utilization of software and to protect the copyright by providing a charging/collecting means for counting up the frequency of the utilized software at every computer system of a user, and charging it based on charging information, and a claiming system counting up this information in a user unit for claiming.

CONSTITUTION: Software which can be utilized under the environment of a computer system 1 is fully provided as a standard and released. A charging system 2 counts up the frequency of the software utilized by the computer system, transfers the information to a collecting system 3, and counts up and outputs charging information. A claiming system 4 counts up the information outputted from the collecting system 3 to a user unit, and executes a claim. In such a way, the generation of appropriation which disregards the copyright can be prevented.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's
 decision of rejection]

[Kind of final disposal of application
 other than the examiner's decision of
 rejection or application converted
 registration]

[Date of final disposal for
 application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's
 decision of rejection]

[Date of requesting appeal against
examiner's decision of rejection]

[Date of extinction of right]

⑫ 公開特許公報(A) 平3-14109

⑬ Int. Cl.⁴G 06 F 1/00
9/06

識別記号

3 7 0 F
4 1 0 Z

庁内整理番号

7459-5B
7361-5B

⑭ 公開 平成3年(1991)1月22日

審査請求 未請求 請求項の数 1 (全2頁)

⑮ 発明の名称 ソフトウェア費用算定方式

⑯ 特 願 平1-151569

⑰ 出 願 平1(1989)6月13日

⑱ 発 明 者 成 田 清 威 東京都港区芝5丁目7番15号 日本電気ソフトウェア株式会社内

⑲ 出 願 人 日本電気ソフトウェア株式会社 東京都港区高輪2丁目17番11号

⑳ 代 理 人 弁理士 内 原 晋

明 細 書

1. 発明の名称

ソフトウェア費用算定方式

2. 特許請求の範囲

ユーザのコンピュータシステム毎に、該コンピュータシステムにて利用されたソフトウェアの度数を集計し、課金情報に基づいて課金する課金・集収手段と、

該課金・集収システムから出力された情報をユーザ単位に集計して請求を行なう請求システムとを有することを特徴とするソフトウェア費用算定方式。

3. 発明の詳細な説明

〔産業上の利用分野〕

本発明は、ソフトウェア費用算定方式に関する。

〔従来の技術〕

従来、ソフトウェア費用は、それぞれのソフト

ウェアごとに価格が設定されており、それらの利用の有無にかかわらず購入時に費用が発生した。また、VAN(付加価値通信網)等TSS(タイムシェアリングシステム)サービスによるソフトウェアのリモート処理サービスがある。

〔発明が解決しようとする課題〕

上述した従来のソフトウェア費用は、購入時に決定され、その時点でのソフトウェアリリースとなる為、以下のような欠点がある。

1. ソフトウェアリリースとなる為著作権を無視した流用が発生する。
2. ソフトウェアの利用可否の為に購入前の十分なテストが出来ない。
3. 機能アップを行うには再購入が必要である。
4. 利用の有無にかかわらず固定費用が発生する。
5. 購入されているソフトウェアの把握は出来るが、利用率は把握出来ない。
6. ユーザごとにリリースする為、毎回セットアップが必要である。

〔課題を解決するための手段〕

本発明のソフトウェア費用算定方式は、ユーザのコンピュータシステム毎に、該コンピュータシステムにて利用されたソフトウェアの度数を集計し、課金情報に基づいて課金する課金・集収手段と、

該課金・集収システムから出力された情報をユーザ単位に集計して請求を行なう請求システムとを有することを特徴とする。

〔実施例〕

次に、本発明について図面を参照して説明する。

第1図は、本発明の一実施例の構成図である。

図中1はユーザに於けるコンピュータシステムであり、この環境下に利用可能なソフトウェアが標準的に完備されリリースされる。2は、課金システムであり、コンピュータシステムにて利用されたソフトウェアを度数集計し、集収システム3へ情報を引き継ぎ、課金情報を集計出力する。

4は、集収システム3より出力された情報をユーザ単位に集計し、請求を行うシステムである。

5は、集収システム3より出力された情報により、利用状況等による市場把握のための資料出力を行い、今後の戦略のベースとなる情報提供を行う利用集計システムである。

〔発明の効果〕

以上説明したように本発明は、ソフトウェア費用の度数課金を行う事により、システムセットアップが不要となり、ソフトウェア利用の拡大、著作権の保護また市場の把握が出来る効果がある。

4. 図面の簡単な説明

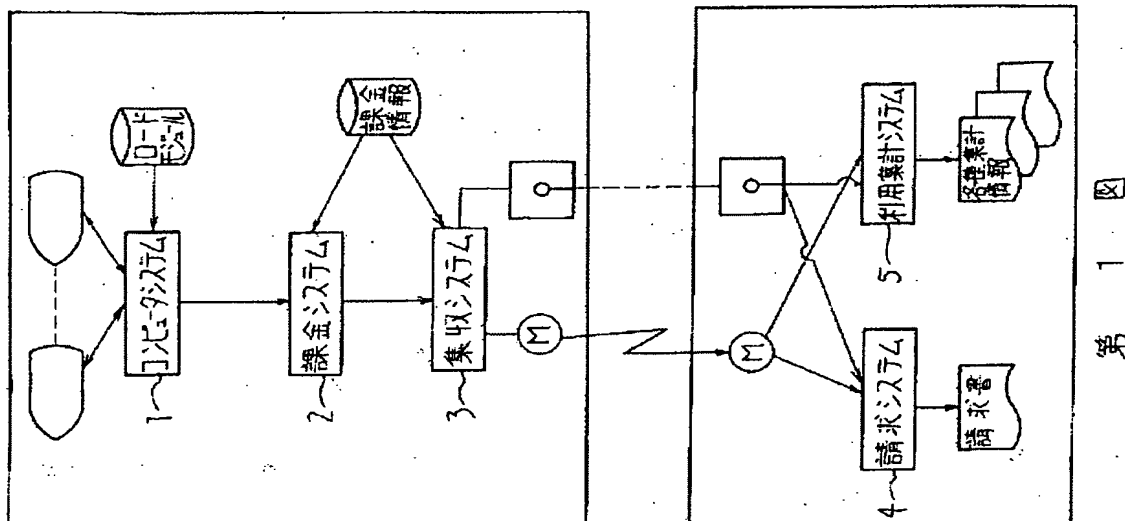
第1図は本発明の一実施例の構成図である。

1……コンピュータシステム、2……課金システム、3……集収システム、4……請求システム、5……利用集計システム。

代理人 弁理士 内 原 晋

- 3 -

- 4 -



216-940

信学技報 Vol. 94 No. 240

電子情報通信学会技術研究報告

ISEC 94-13~22

〔情報セキュリティ〕

1994年9月21日



EIC 電子情報通信学会

BEST AVAILABLE COPY

超流通のためのシステムアーキテクチャ

鳥居 直哉* 木島 裕二* 勝山 恒男* 小森 宰次**

* (株) 富士通研究所

** (株) 富士通

674 明石市大久保町西脇64

211 川崎市中原区上小田中1015

あらまし

現在のCD-ROMによるソフトウェア販売から、森らの提案した理想的な超流通システムまでを視野におき、種々の超流通システムを開発するための共通的なシステムの枠組み、及びその機能を実現する機能部品についての検討結果を示す。超流通システムに必要な基本的な暗号関数の組み合わせを実現するセキュリティ部品を第一層とし、それらを用いて超流通システムの技術単位を実現する超流通部品を第二層とした。第二層の超流通部品を使用することにより、コンテンツの売り切りから従量課金まで、多様な機能を有する第三層の超流通応用システムを見通しよく構築することが可能となる。

和文キーワード 超流通、ソフトウェア、従量課金、試用

System Architecture for Superdistribution

Naoya Torii* Yuji Kijima* Tsuneo Katsuyama* Saiji Komori*

*Fujitsu Laboratories Ltd.

**Fujitsu Ltd.

64, Nishiwaki, Okubo-cho,
Akashi, 674

1015, Kamikodanaka Nakahara-ku,
Kawasaki, 211

Abstract

We proposed a new architecture for developing the superdistribution systems which were originally proposed by Professor Mori of Tsukuba University in 1983. This architecture includes three level function sets. The low level functions consist of security functions. The middle level functions consist of the low level functions and are the basic functions for developing the superdistribution system. We call various types of superdistribution systems as the top level functions which can be constructed by combining the middle level functions.

英文 key words superdistribution, software protection, pay per use, try and buy

1. はじめに

1983年に筑波大学の森らにより情報化社会での電子情報の健全な流通をめざして超流通システムが提案された[1][2]。このシステムは、ソフトウェアのコピーを禁止するのではなく、ソフトウェアの利用に対して従量課金を可能にすることにより情報提供者の利益を保証し、ユーザは、いつでも、どこでもソフトウェアを入手し、利用できる仕組みを提供するものである。

超流通システムの基本的な仕組みは、まず情報提供者がソフトウェアに使用条件を付加し、それに従った課金処理を行わなければ利用できない仕組みを組み込んで広く配布し、コピーも自由にさせる。課金の仕組みをもつ専用計算機(S計算機)を持つユーザは流通するソフトウェアの中から必要なものを自由に選んで利用する。利用に際しては、ユーザのS計算機で課金処理が行われ、利用ログがとられる。ユーザの利用に関しては予め利用上限金額が設定されており、利用上限に達したユーザはソフトウェアの利用が出来なくなる。更にソフトウェアを利用したいユーザは、センタに利用料金を支払い利用ログ情報を通知することで、再び利用できる様になる。

この超流通システムの特徴は、利用者の視点からは、あらゆる情報がいつでも手の届くところにあること、又その中から内容を確認し、場合によっては試用もして、必要な情報を選択することができることである。更に、利用に対して課金するということは、必要な部分だけを購入したり、時間限定、利用回数限定など、好みに応じ、必要な分だけの情報をそれなりの対価で利用できることである。一方、情報提供者の視点からは、情報をころがしておくだけでコストをかけずに確実に利用対価が徴収できるため、自身の所持する情報資産を安心してどんどん公開できることである。

最近、CD-ROMによるソフトウェア販売が開始されている[3]。これは、ユーザが低価格で入手したCD-ROMに格納してある複数のソフトウェアの内、気に入ったソフトウェアの利用権をセンタから購入すれば、対応するソフトウェアがインストールできるシステムである。ソフトウェア本体を販売するのではなく、利用権を販売するという意味で、超流通システムの兆しと

して注目されている。

本稿では、現在のCD-ROMによるソフトウェア販売から、森らの提案した理想的な超流通システムまでを視野におき、種々の超流通システムを開発するための共通的なシステムの枠組み、及びその機能を実現する機能部品についての検討結果を示す。

以下、説明の簡単のためにプログラム及びデータの電子情報を総称してコンテンツと呼び、プログラムという場合はデータを含まず、データという場合はプログラムを含まないものとする。

2. 超流通システムモデル

以下のモデルの上で超流通システムを検討する(図1-図3参照)。

(1)構成要素

情報を提供するコンテンツ提供者、利用するユーザがあり、その間に超流通機能を提供するセンタが介在する。また、CD-ROMなどでコンテンツを流通させるには販売店が必要である。更に、情報の存在を利用者に知らせることも重要となるため「こんな情報がある」という紹介業者も輩出するであろう。本稿では、これらの5種の構成要素を想定する。

〔コンテンツ提供者〕センタに登録し、コンテンツをセンタに提供する。また、ユーザのコンテンツ利用に応じたコンテンツ利用料をセンタから得る。

〔センタ〕超流通の実現のための核になる機能を提供する。まずコンテンツを超流通コンテンツに変換する。これは、課金処理なしでは利用出来ない仕組みをコンテンツに組み込むことを意味する。そして超流通コンテンツを広く流通させる。また、登録したユーザに対し、要求に応じてコンテンツ利用のためのライセンス情報の発行、コンテンツ利用に対する料金の徴収及び分配をおこなう。従量課金のためには、ユーザ側で利用制御や課金処理を行う課金モジュールを提供する。

〔販売店〕センタに登録し、超流通コンテンツの流通、販売を行う。又、ユーザのコンテンツ利用に応じた利益分配をセンタから得る。

〔紹介業者〕超流通コンテンツの内容の紹介、評価を行い、ユーザにコンテンツ選択のための

情報を提供する。また、ユーザの超流通コンテンツ利用に応じた利益分配をセンタから得る。

〔ユーザ〕超流通コンテンツを利用し、代金を支払う。

(2)コンテンツの流れ

コンテンツの流れを図1に示す。コンテンツ提供者から提供されたコンテンツは、センタで超流通コンテンツに変換され、販売店を経由して、或いはセンタから直接ユーザに流れる。紹介業者は、コンテンツ提供者、或いはセンタから提供される超流通コンテンツを基に紹介情報を作成しユーザに提供する。

ユーザは、超流通コンテンツを低廉な価格で入手する。超流通コンテンツは、制限付で試用可能である。

(3)コンテンツの利用

超流通コンテンツを利用したい場合、ユーザはライセンス情報と残高情報(利用限度額)をセンタから入手し、課金モジュールに設定する。これにより超流通コンテンツの利用が可能となる。ライセンス情報には、コンテンツ利用のための秘密鍵と利用条件が含まれている。超流通コンテンツを起動すると、起動時、或いは決められたタイミングで課金モジュールへライセンス有無の確認依頼が出力される。課金モジュールでは、ライセンスがある場合は利用条件に従って残高情報が更新され、利用ログがとられる。さらに、ライセンス情報の秘密鍵を用いて課金モジュールから超流通コンテンツに利用許可が通知される。以下、簡単のために課金モジュール内部のライセンス情報、残高情報、及びログ情報をまとめて利用情報と呼ぶことにする。超流通コンテンツ利用代金の支払い方法については、最初の残高情報設定時に、前払いか、後払いかを設定する(図2参照)。

課金モジュールがないユーザでも超流通コンテンツを買い取ることは可能である。この場合、代金と引換えにセンタからライセンス情報を受け取る。

(4)関連業者への利益分配

超流通コンテンツは、利用されて初めて利用料が発生する。この利用料の内の幾分かは販売努力や紹介努力の見返りとして販売店や紹介業者に分配されてしかるべきである。

超流通コンテンツの利用に応じた利益分配の

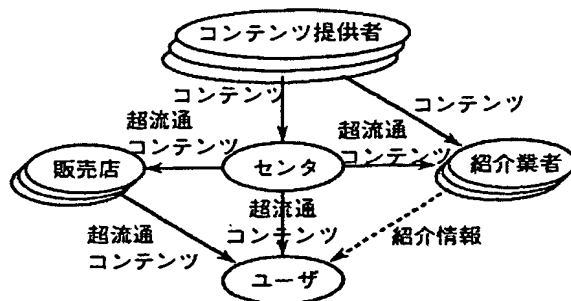


図1 コンテンツの流れ

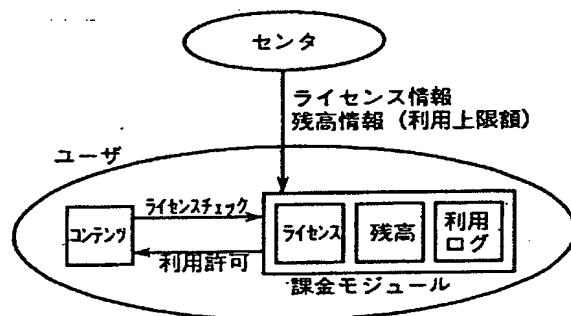


図2 コンテンツの利用

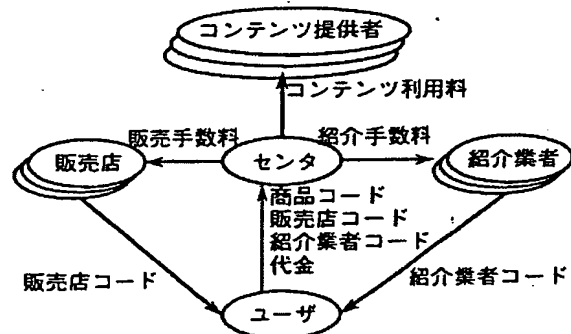


図3 関連業者への利益分配

ためには、超流通コンテンツの商品コード、入手した販売店コード、紹介情報を参考にした紹介業者コードをセンタに通知する必要がある。

従量課金の場合は、残高情報を課金モジュール

ルに入力する時点で、これらの業者コードを含んだログ情報をセンタに通知する。買い取りの場合は、ライセンス購入時にこれらの業者コードをセンタに通知する(図3参照)。

センタでは、その情報にもとづいて利益分配を行う。即ち、コンテンツ提供者にはコンテンツ利用料として、販売店には販売手数料として、紹介業者には紹介手数料として分配する。

3. 超流通システムの検討

3.1 超流通システムの基本機能

超流通システムが提供すべき基本機能は、課金、利益分配、コンテンツの試用、情報の保護である。特に課金に関しては、売り切りや各種従量課金の機能を提供しなければならない。従量課金は、課金単位、課金集計方法、ユーザの代金支払い方法の組み合わせにより多様な実現形態があるが、それらを柔軟に実現可能にする。

(1)課金

(a)売り切り

一回の支払いでコンテンツを無制限に利用できる。

(b)従量課金

①課金単位

起動回数によるもの、利用期間によるもの、及び利用時間による従量課金機能を提供する。

②課金集計方法

コンテンツ毎に課金する個別集計と、複数のコンテンツの利用に対してまとめて課金する一括集計の機能を提供する。一括集計の場合は、センタでの利益分配のためにログ情報の通知が必要であるが、個別集計の場合は、不要である。

③代金支払い方法

ユーザの代金支払い方法については、前払いと後払いを可能とする。両者の違いは課金モジュールの利用情報の削除機能の違いとして表れる。即ち、前払いならユーザがローカルに利用情報を削除可能であるが、後払いの場合はセンタがユーザの代金の支払いに応じて削除する。

(2)利益の分配

販売店、紹介業者のコードをセンタに通知することにより利益分配を可能とする。

(3)試用

コンテンツを購入する前の試用を可能にする。試用は、制約付きでコンテンツの利用を可能と

する形をとる。ライセンス購入により制約が解除される。

(4)保護機能

(a)コンテンツの保護

ユーザにコンテンツ利用のためのライセンス、あるいは課金モジュールがないと利用できない機能を提供する。

(b)利用情報の保護

ユーザが不正に課金モジュール内の利用情報を更新できないようにする。

3.2 超流通技術の部品化

超流通システムの実現には、暗号、認証、アクセス制御技術等のセキュリティ技術が重要な役割を果たすが、これらの技術の適用に当たっては、システム全体にわたる安全性を考慮したシステム構築が必要である。そこで、超流通システムの構築にあたって見通しをよくし、安全性を保証するために、超流通システムの実現技術を部品の形で提供し、その中にセキュリティ技術を織り込むことにした。

このために、セキュリティ技術および超流通技術の整理を行った。超流通システムに必要な基本的な暗号関数の組み合わせをセキュリティ部品として第一層に据え、超流通システムの技術単位に分類した第二層の超流通部品の中に組み込む。この第二層の超流通部品を使用して第

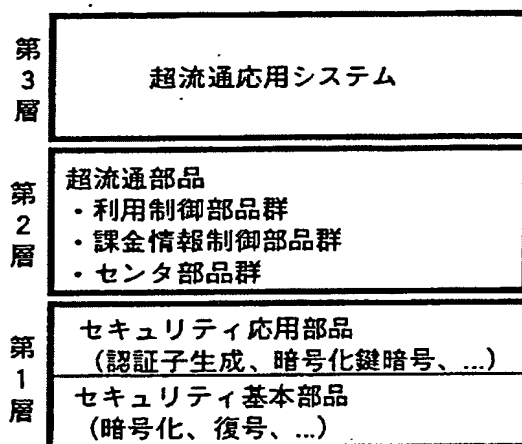


図4 超流通部品の階層化

表1 第一層 基本部品

部品名	内 容	利用基本部品
暗号化	———	———
復号	———	———
乱数発生	———	———
比較	———	———
アクセス制御	———	———
秘密鍵	———	———
認証子生成	受信した乱数を秘密鍵で暗号化	暗号化、秘密鍵
認証子確認	乱数を秘密鍵で暗号化し、受信した認証子と比較	乱数発生、暗号化、秘密鍵、比較
暗号化鍵暗号	暗号化鍵を秘密鍵で暗号化	暗号化、秘密鍵
暗号化鍵復号	秘密鍵で暗号化された暗号化鍵を復号	復号、秘密鍵
連番付署名作成	秘密鍵を用いて連番付メッセージの圧縮変換	暗号化、秘密鍵
連番付署名確認	連番付署名されたメッセージの連番、署名確認	暗号化、秘密鍵、比較

三層の超流通応用システムを構築する(図4参照)
超流通部品の検討結果を以下に示す。

(1)第一層(セキュリティ部品)

表1に基本部品と応用部品を示す。基本部品は、超流通システムの基本機能を実現するために必要なセキュリティの基本技術を実現する部品である。応用部品は、次に説明する第二層の超流通部品を構築するにあたって、共通に使えるようなセキュリティ応用技術を実現する部品であり、基本部品の組み合わせで構築する。

(2)第二層(超流通部品)

超流通部品は、コンテンツの利用時の制御に関連する利用制御部品群、課金モジュール内部に格納される利用情報(ライセンス情報、残高情報、及びログ情報)を管理する利用情報ファイル管理部品群、センタ部品群に分類した。図5に超流通部品の配置例を示す。

(a)利用制御部品群

①利用権チェック

プログラムの起動時、あるいは定められたタイミングで、ユーザがライセンスを所有しているかどうかをチェックする。この際に後述の利用許可発行・課金集計との通信を行う。送信情報は乱数であり、受信情報はライセンス情報に含まれる認証用の鍵で送信乱数を暗号化したデータである。利用権チェックは、内部の認証用の鍵を用いて同様の処理を行い同じ暗号化乱数であればライセンスを所持しているとする。この方法は、従量課金の課金単位、課金集計方

法、代金支払い方法によらず共通に利用する。

②利用権チェック・復号

コンテンツがデータの場合は、利用権チェックの代わりに利用権チェック・復号がライセンスの所有をチェックする。更に、データは暗号化されているのでライセンス情報に含まれる鍵で暗号化データを復号する。データの復号処理を利用許可発行・課金集計に任せることもできる。

③利用許可発行・課金集計

利用権チェック、あるいは利用権チェック・復号からの要求に応じてライセンス情報の所持を示す情報、或いはコンテンツの暗号鍵、あるいは暗号化データの復号結果を出力する。売り切りの場合は、ライセンスの有無を確認して利用を許可する。従量課金の場合は、ライセンスを確認後、残高を削減し、利用履歴を追加する。残高情報は、前払いでも後払いでも、コンテンツ利用に対して削減する。すなわち、前払いの場合は、前払い額に応じた残高から削減され、後払いの場合は、利用上限が設定され、その額から削減される。

(b)利用情報ファイル管理部品群

課金モジュール内部には、各コンテンツのライセンス情報を格納するライセンスファイル、残高情報を格納する残高ファイル、コンテンツ利用ログを格納するログファイルがある。この3種のファイルを総称して利用情報ファイルと呼ぶ。課金モジュール側にある、ライセンス入

力、残高入力、ログ通知、及びファイル削除（センタ）の各部品は、課金モジュールとセンタが各ファイルの情報を通信する場合に、署名と連番を付加するための部品である。但し、ライセンス情報に含まれるプログラム認証用の鍵やデータの暗号は、モジュールの秘密鍵で暗号化する。

又、利用情報ファイル参照部品は、ユーザが利用情報ファイルを見るための部品である。ファイル削除（ローカル）は、前払いの場合に利用情報ファイルの項目を削除するための部品である。ファイル削除（センタ）は、後払いの場合に、センタからの連番、署名付のコマンドにより利用情報ファイルを削除するための部品である。

(c) センタ部品群

センタ部品群は、ユーザの利用情報ファイル

更新要求に応じて処理を行うセンタ側利用情報ファイル管理部品群と、販売店、紹介業者の管理を行う業者管理部品群、コンテンツ提供者からのコンテンツに課金、及び保護機構を組み込み、流通の管理も行うコンテンツ管理部品群、ユーザ認証、ID発行、利用料金徴収等のユーザ管理部品群の4種に分類した。

センタ側利用情報ファイル管理部品群は、ライセンス管理、残高管理、及びログ管理からなる。

ライセンス管理は、コンテンツ提供者の要求に応じてコンテンツ利用条件を設定する（ユーザは、その設定条件内で利用条件を選択する）。

又、ユーザの要求に応じてコンテンツ利用のための鍵を発行する。但し、ライセンスに含まれる秘密鍵は、課金モジュールがある場合は、ユーザの課金モジュール鍵で、無い場合は、

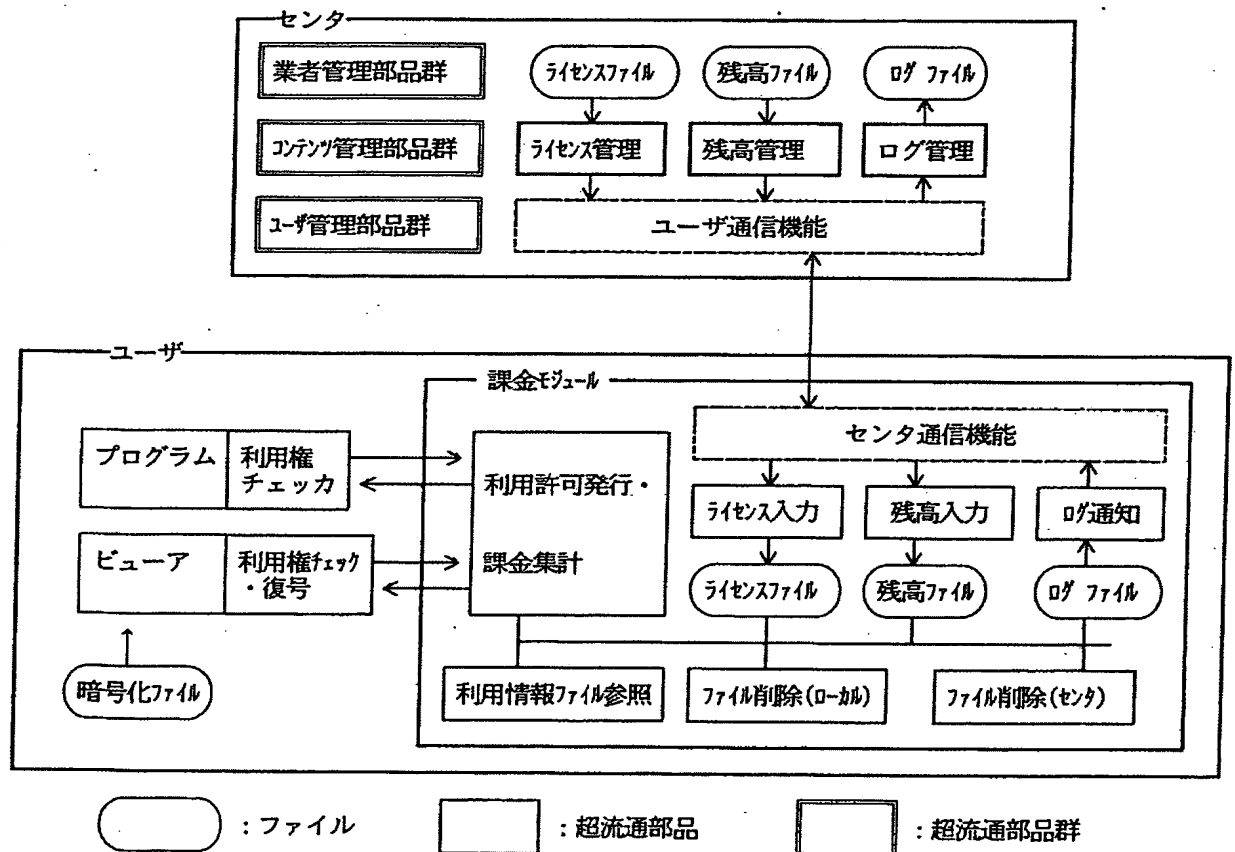


図5 超流通部品配置例

表2 第二層 超流通部品（利用制御部品群、利用情報ファイル管理部品群）

部 品 名		内 容	利用部品
利用権チェック	1	課金モジュールの無い場合に対応	暗号化鍵復号、比較
	2	課金モジュールの有る場合に対応	認証子確認、暗号化鍵復号、乱数発生
利用権チェック・復号	1	課金モジュールの無い場合（部品内部で復号）	暗号化鍵復号、復号
	2	課金モジュールの有る場合（部品内部で復号）	暗号化鍵復号、復号、乱数発生
	3	課金モジュールの有る場合（課金モジュールで復号）	なし
利用許可発行・課金集計	1	プログラムに対応、売りきりの場合	暗号化鍵復号、比較
	2	プログラムに対応、従量課金の場合	認証子生成
	3	データに対応、従量課金の場合（利用権チェック・復号で復号）	暗号化
	4	データに対応、従量課金の場合（部品内部で復号）	復号
ライセンス入力		セクタからのライセンスを残高ファイルに格納	暗号化鍵復号
残高入力		セクタからの残高を残高ファイルに格納	連番付署名確認
ログ通知		課金モジュールからセクタにログ（利用履歴）を通知	連番付署名作成
ファイル削除（センタ）		セクタからの署名付ライセンス削除（後払いのライセンス）	連番付署名確認
ファイル削除（ローカル）		ユーザ要求によるライセンス削除（前払いのライセンス）	なし
利用情報ファイル参照		ユーザが、利用情報ファイル（残高、利用属性、ログ情報）を参照	アクセス制御

ユーザの登録したユーザ固有情報で暗号化される。

残高管理は、ユーザからの要求に応じてユーザ側の課金モジュール内部の残高情報の更新情報を送る。

ログ管理は、ユーザのコンテンツ利用履歴を吸い上げて管理する。この情報は、業者管理部品群での利益分配のために用いられる。

業者管理部品群、コンテンツ管理部品群、及びユーザ管理部品群の詳細については割愛する。

表2に第二層（超流通部品）のうち、利用制御部品群と利用情報ファイル管理部品群に含まれる部品について示す。

3.3 超流通部品による機能実現

表3に超流通コンテンツがプログラムの場合の各種課金形態とそのために必要な部品を示す。利用権チェック、利用許可発行・課金集計の項目の番号は、表2における該当部品の番号である。課金単位の違いは部品構成に影響しない。

表4に、超流通コンテンツがデータの場合の各種課金形態とそのために必要な部品を示す。利用権チェック・復号、利用許可発行・課金集計の項目の番号は、表2における該当部品の番号である。課金方法の各項目に対して2つずつ

の部品構成を記載したが、これはデータの復号を利用権チェック・復号で行う場合と、利用許可発行・課金集計で行う場合のそれぞれに相当する。

4. むすび

本稿では、超流通システムの開発に向けて、超流通技術の部品化を検討した。超流通システムに必要な基本的な暗号関数の組み合わせを実現するセキュリティ部品を第一層とし、それらを用いて超流通システムの技術単位を実現する超流通部品を第二層とした。第二層の超流通部品を使用することにより、コンテンツの売り切りから従量課金まで、多様な機能を有する第三層の超流通応用システムを見通しよく構成できる。今後は、これらの部品を適用した応用システムの構築をすすめ、有効性の検証を行う予定である。

表3 超流通部品を用いた各種課金の実現（プログラムの場合）

課金方法		課金 モジュール	利用権 チェック	利用許可発行 課金集計	ライセンス 入力	残高 入力	ログ 通知	ファイル削除		ファイル 参照
								ローカル	センタ	
売り切り		なし	1	×	×	×	×	×	×	○
			2	1	○	×	×	×	×	○
個別集計	前払い	あり	2	2	○	○	×	○	×	○
	後払い		2	2	○	○	○	×	○	○
一括集計	前払い		2	2	○	○	○	○	×	○
	後払い		2	2	○	○	○	×	○	○

○：利用 ×：不要

表4 超流通部品を用いた各種課金の実現（データの場合）

課金方法		課金 モジュール	利用権 番号	利用許可発行 課金集計	ライセンス 入力	残高 入力	ログ 通知	ファイル削除		ファイル 参照
								ローカル	センタ	
売り切り		なし	1	×	×	×	×	×	×	○
			2	3	○	×	×	×	×	○
			3	4						
個別集計	前払い	あり	2	3	○	○	×	○	×	○
			3	4						
	後払い		2	3	○	○	○	×	○	○
			3	4						
一括集計	前払い		2	3	○	○	○	○	×	○
			3	4						
	後払い		2	3	○	○	○	×	○	○
			3	4						

○：利用 ×：不要

[参考文献]

[1] 森 亮一、田代 秀一“ソフトウェア・サービス・システム(SSS)の提案”，電子通信学会 論文誌，Vol. J70-D, No1, pp70-81(1987)

[2] (社)日本電子業振興協会“マイクロコンピュータに関する調査報告書[III] ソフトウェア技術” 91-パ-3(平成3年3月)

[3] “ソフト流通、「物」から「使用権」へ暗号化してCD-ROMで配布”，日経エレクトロニクス 1993. 9. 13(No. 590) pp. 143-150

複写をされる方に

本誌に掲載された著作物は、政令が指定した図書館で行うコピーサービスや、教育機関で教授者が講義に利用する複写をする場合等、著作権法で認められた例外を除き、著作権者に無断で複写すると違法になります。そこで、本著作物を合法的に複写するには、著作権者から複写に関する権利の委託を受けている次の団体と、複写をする人またはその人が所属する企業・団体等との間で、包括的な許諾契約を結ぶようにして下さい。

日本複写権センター 〒107 東京都港区北青山3-3-7 第一青山ビル
Phone 03-3401-2382, 03-3475-5618 Fax 03-3401-2386

Notice about photocopying

In order to photocopy any work from this publication legally, you or your organization needs to obtain permission from the following organization that has been delegated for the copyright clearance by the copyright owner of this publication.

[Japan] Japan Reprographic Rights Center
Daiichi Aoyama Bldg. 3F,
3-3-7, Kita-Aoyama, Minato-ku, Tokyo, 107 Japan
Phone +81-3-3401-2382 Telefax +81-3-3401-2386

[U.S.A.] Copyright Clearance Center, Inc.
222 Rosewood Drive, Danvers, MA 01923, USA
Phone (508) 750-8400 Telefax (508) 750-4744

電子情報通信学会技術研究報告

信学技報 Vol. 94 No. 240

1994年9月21日 発行

IEICE Technical Report

©電子情報通信学会 1994

Copyright : © 1994 by the Institute of Electronics, Information and Communication Engineers (IEICE)

発行人 東京都港区芝公園3丁目5番8号 機械振興会館内

社団法人 電子情報通信学会 事務局長 田村 恵一

発行所 東京都港区芝公園3丁目5番8号

社団法人 電子情報通信学会 電話 (03) 3433-6691
郵便振替口座 東京2-35300番

The Institute of Electronics, Information and Communication Engineers,
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome., Minato-ku,
TOKYO, 105 JAPAN

本技術研究報告に掲載された論文の著作権は(社)電子情報通信学会に帰属します。

Copyright and reproduction permission: All rights are reserved and no part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission in writing from the publisher. Notwithstanding, instructors are permitted to photocopy isolated articles for noncommercial classroom use without fee.

**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.

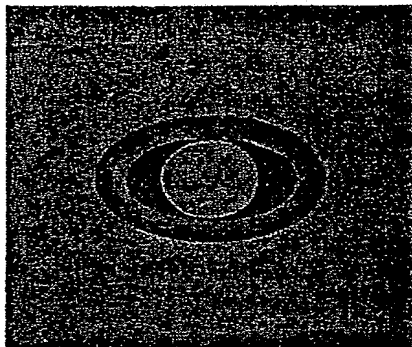
Defects in the images include but are not limited to the items checked:

- ☐ 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.

*Capability-Based
Computer Systems*



Henry M. Levy

Capability-Based Computer Systems

Henry M. Levy

This book was published by Digital Press in 1984. It is still the most thorough survey and description of early capability-based and object-based hardware and software systems. The book is now out of print and the copyright belongs to the author, who makes the material available here for viewing or downloading, in Adobe Acrobat PDF format (free Acrobat reader available [here](#)).

BEST AVAILABLE COPY

Contents

1	<i>Capability- and Object-Based System Concepts</i>	1
	<i>Capability-Based Systems</i>	3
	<i>Memory Addressing in Computer Systems</i>	5
	<i>The Context of an Address</i>	9
	<i>Protection in Computer Systems</i>	10
	<i>The Object-Based Approach</i>	13
	<i>Capabilities and Object-Based Systems</i>	15
	Summary	17
	For Further Reading	18
2	<i>Early Descriptor Architectures</i>	21
	Introduction	21
	The Burroughs B5000	22
	The Rice University Computer	25
	The Basic Language Machine	30
	Discussion	34
	For Further Reading	38
3	<i>Early Capability Architectures</i>	41
	Introduction	41
	Dennis and Van Horn's Supervisor	41
	The MIT PDP-1 Timesharing System	47
	The Chicago Magic Number Machine	48
	The CAL-TSS System	52
	Discussion	57
	For Further Reading	61
4	<i>The Plessey System 250</i>	65
	Introduction	65

xi

Contents

	System Overview	66
	Capability Addressing	66
	Capability Register Usage	69
	Inform and Outform Capabilities	69
	Instructions and Addressing	71
	Protected Procedure Calls	72
	Operating System Resource Management	73
	Input and Output	74
	Discussion	75
	For Further Reading	77
5	<i>The Cambridge CAP Computer</i>	79
	Introduction	79
	Hardware Overview	79
	CAP Process Structure	80
	CAP Addressing Overview	81
	Capabilities and Virtual Addresses	83
	Process Data Structures	85
	Memory Address Evaluation	86
	Subprocess Creation	87
	The Capability Unit	89
	Protected Procedures	90
	Long-Term Storage and Long-Term Names	95
	Discussion	96
	For Further Reading	99
6	<i>The Hydra System</i>	103
	Introduction	103
	Hydra Overview	103
	Hydra Objects and Types	105
	Processes, Procedures, and Local Name Spaces	107
	Hydra Operations	109
	Capabilities and Rights	111
	Supporting Protected Subsystems	113
	<i>Templates</i>	113
	<i>Typecalls</i>	116
	Hydra Object Storage System	116
	Capability Representation	120
	Reference Counts and Garbage Collection	121
	Discussion	122
	For Further Reading	125
7	<i>The STAROS System</i>	127
	Overview of STAROS	127
	STAROS Object Support	129

STAROS Capabilities	130
Object Addressing	131
STAROS Abstract Type Management	133
Discussion	134
For Further Reading	135
8 The IBM System/38	137
Introduction	137
System Objects	139
Object Addressing	141
<i>Virtual Memory</i>	141
<i>Pointers</i>	142
<i>Contexts</i>	144
<i>Physical Address Mapping</i>	145
Profiles and Authority	147
<i>Authority/Pointer Resolution</i>	148
Programs/Procedures	150
<i>The Instruction Stream</i>	151
<i>Program Activation and Invocation</i>	152
<i>Protected Procedures</i>	153
Special Privileges	154
Discussion	154
For Further Reading	157
9 The Intel iAPX 432	159
Introduction	159
Segments and Objects	161
Object Addressing	163
<i>Object Descriptors</i>	163
<i>Access Descriptors</i>	165
Program Execution	167
<i>Domains and Instruction Objects</i>	168
<i>Procedure Call and Context Objects</i>	169
<i>Instruction Operand Addressing</i>	171
<i>Context Allocation</i>	172
<i>Parameter Passing</i>	173
Abstraction Support	173
<i>Domains and Refinements</i>	174
<i>Creation of Typed Objects</i>	176
<i>Programmer-Defined Types</i>	177
Storage Resources	179
Instructions	182
Discussion	184
For Further Reading	186

10 Issues in Capability-Based Architectures	187
Introduction	187
Segmentation	188
Storage of Capabilities	189
Capability Representation	191
Objects	195
Protected Procedures and Type Extension	196
Object Lifetimes and Garbage Collection	197
Object Locking	201
Revocation	202
Conclusions	203
<i>Capability and Object System Bibliography</i>	205
<i>Index</i>	217

Figures

1-1.	A Capability	3
1-2.	Conventional Segment Address Translation	6
1-3.	Capability Register Addressing	7
1-4.	System Object Access Matrix	11
1-5.	Access Control and Capability Lists	11
2-1.	B5000 Program Reference Table	23
2-2.	B5000 Descriptor Formats	24
2-3.	Rice University Computer Codeword Format	27
2-4.	Rice University Computer Memory Organization	29
2-5.	Example of BLM Numeric Formats	32
2-6.	Basic Language Machine Addressing	33
2-7.	BLM Address and Codeword Formats	34
3-1.	Processes, Computations, and C-lists	43
3-2.	Protected Procedure Protection Spheres	46
3-3.	Chicago Magic Number Machine Linkage Segment	52
3-4.	CAL-TSS Object Addressing	55
4-1.	Plessey System 250 Capability Formats	67
4-2.	Plessey System 250 Capability Loading	68
4-3.	System 250 Instruction Formats	71
4-4.	Protected Procedure Resource Subsystem	74
5-1.	CAP Process Hierarchy	81
5-2.	CAP Process Addressing	82
5-3.	CAP Capability and Access Rights Formats	83
5-4.	CAP Virtual Address	84
5-5.	CAP PRL Entry	85
5-6.	CAP Process Base	86

5-7.	Capability Unit Register Format	89
5-8.	CAP Capability Unit	91
5-9.	CAP Enter Capability and Enter PRL Formats	92
5-10.	CAP Protected Object Implementation	94
6-1.	Hydra Object and Type Object	106
6-2.	Hydra Type Hierarchy	108
6-3.	Hydra Capability	111
6-4.	Hydra Procedure Call	115
6-5.	Hydra TypeCall	117
6-6.	Active Fixed Part Directory	119
6-7.	Hydra Capability Formats	121
7-1.	A CM* Cluster	128
7-2.	STAROS Capability and Capability Rights Word	130
7-3.	STAROS Object Descriptor Format	132
7-4.	STAROS Directory Structure	133
8-1.	System/38 Implementation Layers	138
8-2.	IBM System/38 System Object	140
8-3.	System/38 Virtual Address	142
8-4.	System/38 Virtual Address Translation	146
8-5.	System/38 Example High-level Instruction	152
9-1.	Intel iAPX 432 Structure	160
9-2.	Intel 432 Segment	161
9-3.	Intel 432 Storage Segment Descriptor	163
9-4.	Intel 432 Access Descriptor	165
9-5.	Intel 432 Address Translation	167
9-6.	Intel 432 Domain and Instruction Objects	168
9-7.	Intel 432 Context Object Representation	170
9-8.	Intel 432 Access Selector Formats	172
9-9.	Intel 432 Parameter Passing	174
9-10.	Intel 432 Domain Refinement	175
9-11.	Intel 432 Type Control Object Data Part	176
9-12.	Intel 432 Dynamic Object Addressing	178
9-13.	Intel 432 Storage Resource Object	179
9-14.	Intel 432 Instruction Format	182
9-15.	Intel 432 Reference Format	183

Tables

1-1.	Major Descriptor and Capability Systems	2
3-1.	Dennis and Van Horn Supervisor Capability Operations	44
3-2.	Chicago Magic Number Supervisor Capability Operations	51

Contents

6-1.	Hydra Kernel-Implemented Types	107
6-2.	Generic Object and Capability Operations	110
6-3.	Capability and Generic Object Access Rights	111
6-4.	Hydra Active and Passive Fixed Parts	118
7-1.	STAROS Representation Types	129
7-2.	STAROS Capability Types	131
8-1.	System/38 System Object Types	139
8-2.	System/38 Pointer Instructions	144
8-3.	System/38 Context Instructions	145
8-4.	System/38 Authority Management Instructions	149
9-1.	Intel 432 System Object Types	162
9-2.	Intel 432 Storage Segment Descriptor Fields	164
9-3.	Intel 432 Access Descriptor Instructions	166

Capability-Based Computer Systems

Capability- and Object-Based System Concepts

Although the complexity of computer applications increases yearly, the underlying hardware architecture for applications has remained unchanged for decades. It is, therefore, not surprising that the demands of modern applications have exposed limitations in conventional architectures. For example, many conventional systems lack support in:

1. *Information sharing and communications.* An essential system function is the dynamic sharing and exchange of information, whether on a timesharing system or across a network. Fundamental to the sharing of storage is the addressing or naming of objects. Sharing is difficult on conventional systems because addressing is local to a single process. Sharing would be simplified if addresses could be transmitted between processes and used to access the shared data.
2. *Protection and security.* As information sharing becomes easier, users require access controls on their private data. It must also be possible to share information with, or run programs written by, other users without compromising confidential data. On conventional systems, all of a user's objects are accessible to any program which the user runs. Protection would be enhanced if a user could restrict access to only those objects a program requires for its execution.
3. *Reliable construction and maintenance of complex systems.* Conventional architectures support a single privileged mode of operation. This structure leads to monolithic design; any module needing protection must be part of the single operating system kernel. If, instead, any module could execute within a protected domain, systems could be built as a collection of independent modules extensible by any user.

Over the last several decades, computer industry and university scientists have been searching for alternative architectures that better support these essential functions. One alternative architectural structure is *capability-based* addressing. Capability-based systems support the *object-based* approach to computing.

This book explains the capability/object-based approach and its implications, and examines the features, advantages, and disadvantages of many existing designs. Each chapter presents details of one or more capability-based systems. Table 1-1 lists the systems described, where they were developed, and when they were designed or introduced.

System	Developer	Year	Attributes
Rice University Computer	Rice University	1959	segmented memory with "codeword" addressing
Burroughs B5000	Burroughs Corp.	1961	stack machine with descriptor addressing
Basic Language Machine	International Computers Ltd., U.K.	1964	high-level machine with codeword addressing
Dennis and Van Horn Supervisor	MIT	1966	conceptual design for capability supervisor
PDP-1 Time-sharing System	MIT	1967	capability supervisor
Multicomputer/Magic Number Machine	University of Chicago Institute for Computer Research	1967	first capability hardware system design
CAL-TSS	U.C. Berkeley Computer Center	1968	capability operating system for CDC 6400
System 250	Plessey Corp., U.K.	1969	first industrial capability hardware and software system
CAP Computer	University of Cambridge, U.K.	1970	capability hardware with microcode support
Hydra	Carnegie-Mellon University	1971	object-based multi-processor O.S.
STAROS	Carnegie-Mellon University	1975	object-based multi-processor O.S.
System/38	IBM, Rochester, MN.	1978	first major commercial capability system, tagged capabilities
iAPX 432	Intel, Aloha, OR.	1981	highly-integrated object-based micro-processor system

Table 1-1: Major Descriptor and Capability Systems

Before surveying these systems at a detailed architectural level, it is useful to introduce the concepts of capabilities and object-based systems. This chapter defines the concept of capability, describes the use of capabilities in memory addressing and protection, introduces the object-based programming approach, and relates object-based systems to capability-based addressing.

Simplified examples of capability-based and conventional computer systems are presented throughout this chapter. These examples are meant to introduce the capability model by contrasting it with more traditional addressing mechanisms. In fact, many design choices are possible in both domains, and many conventional systems exhibit some of the properties of capability systems. No one of the following models is representative of all capability or conventional systems.

1.1 Capability-Based Systems

Capability-based systems differ significantly from conventional computer systems. Capabilities provide (1) a single mechanism to address both primary and secondary memory, and (2) a single mechanism to address both hardware and software resources. While solving many difficult problems in complex system design, capability systems introduce new challenges of their own.

Conceptually, a capability is a token, ticket, or key that gives the possessor permission to access an entity or object in a computer system. A capability is implemented as a data structure that contains two items of information: a *unique object identifier* and *access rights*, as shown in Figure 1-1.

The identifier *addresses* or *names* a single object in the computer system. An object, in this context, can be any logical or physical entity, such as a segment of memory, an array, a file, a

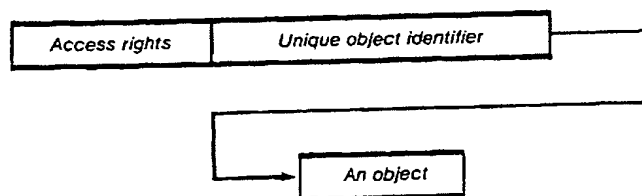


Figure 1-1: A Capability

line printer, or a message port. The access rights define the *operations* that can be performed on that object. For example, the access rights can permit read-only access to a memory segment or send-and-receive access to a message port.

Each user, program, or procedure in a capability system has access to a *list of capabilities*. These capabilities identify all of the objects which that user, program, or procedure is permitted to access. To specify an object, the user provides the *index* of a capability in the list. For example, to output a record to a file, the user might call the file system as follows:

```
PUT( file_capability , "this is a record" );
```

The capability specified in the call serves two purposes. First, it identifies the file to be written. Second, it indicates whether the operation to be performed (PUT in this case) is permitted.

A capability thus provides addressing and access rights to an object. Capabilities are the basis for object protection; a program cannot access an object unless its capability list contains a suitably privileged capability for the object. Therefore, the system must prohibit a program from directly modifying the bits in a capability. If a program could modify the bits in a capability, it could forge access to any object in the system by changing the identifier and access rights fields.

Capability system integrity is usually maintained by prohibiting direct program modification of the capability list. The capability list is modified only by the operating system or the hardware. However, programs can obtain new capabilities by executing operating system or hardware operations. For example, when a program calls an operating system routine to create a new file, the operating system stores a capability for that file in the program's capability list. A capability system also provides other capability operations. Examples include operations to:

1. Move capabilities to different locations in a capability list.
2. Delete a capability.
3. Restrict the rights in a capability, producing a less-privileged version.
4. Pass a capability as a parameter to a procedure.
5. Transmit a capability to another user in the system.

Thus, a program can execute direct control over the movement of capabilities and can share capabilities, and therefore, objects, with other programs and users.

It is possible for a user to have several capability lists. One list will generally be the master capability list containing capabilities for secondary lists, and so on. This structure is similar to a multi-level directory system, but, while directories address only files, capabilities address objects of many types.

1.1.1 Memory Addressing in Computer Systems

This section presents simplified models for both conventional and capability-based memory addressing systems. Although capabilities can control access to many object types, early capability-based systems concentrated on using capabilities for primary memory addressing. The first use of capabilities for memory protection was in the Chicago Magic Number Machine [Fabry 67, Yyngve 68], and an early description of capability-based memory protection appeared in Wilkes' book on timesharing systems [Wilkes 68]. Later, [Fabry 74] described the advantages of capabilities for generalized addressing and sharing.

For purposes of a simplified model, consider a conventional computer supporting a multiprogramming system in which each program executes within a single process. A program is divided into a collection of segments, where a segment is a contiguous section of memory that represents some logical entity, such as a procedure or array. A process defines a program's address space: that is, the memory segments it can access. The process also contains data structures that describe the user, and a directory that contains the names of a set of files. These files represent the user's long-term storage.

When a program is run, the operating system creates a process-local segment table that defines the memory segments available to the program. The segment table is a list of *descriptors* that contain physical information about each segment. Figure 1-2 shows example formats for a process virtual address and segment table descriptor. The operating system loads various segments needed by the program into primary memory, and loads the segment table descriptors with the physical address and length of each segment. A process can then access segments by reading from or writing to virtual addresses.

Each virtual address contains two fields: the segment number and the offset of a memory element within that specified segment. On each virtual address reference the hardware uses the segment number field as an index to locate an entry in the

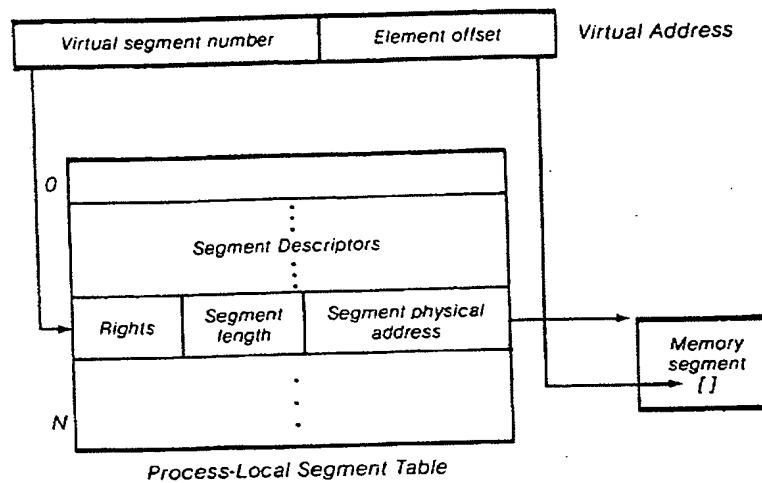


Figure 1-2: Conventional Segment Address Translation

process segment table. This descriptor contains the physical location of the segment. The length field in the descriptor is used to check that the offset in the virtual address is within the segment bounds. The rights field in the segment table entry indicates the type of access permitted to that segment (for example, read or write).

The model shown in Figure 1-2 has the following properties:

1. The system supports a segmented process virtual address space. A virtual address is local to the process and is translated through the process-local segment table.
2. A program can construct any virtual address and can attempt to read or write that address. On each reference, the hardware ensures that (a) the segment exists, (b) the offset is valid, and (c) the attempted operation is permitted. Otherwise, an error is signaled.
3. Loading of segment table entries is a privileged operation and can be accomplished only by the operating system. In general, a segment table is created at the time a program is loaded. The program then executes in a static addressing environment.
4. Sharing of segments between processes requires that the operating system arrange for both process-local segment tables to address the shared segments. If two processes wish to use the same virtual address to access a shared segment,

the segment descriptors must be in the same locations in both segment tables.

5. Any dynamic sharing of segments requires operating system intervention to load segment descriptors.

A *capability-based system* also supports the concept of a process that defines a program's execution environment. In the capability system, each process has a capability list that defines the segments it can access. Instead of the segment table descriptors available to the conventional system hardware, the capability addressing system consists of a set of *capability registers*. The *program* can execute hardware instructions to transfer capabilities between the capability list and the capability registers. The number of capability registers is generally small compared to the size of the capability list. Thus, at any time, the capability registers define a subset of the potentially accessible segments that can be physically addressed by the hardware. A simplified hardware model for this system is shown in Figure 1-3.

The model shown in Figure 1-3 has the following properties:

1. The system has a segmented virtual address space. A segment of memory can only be addressed by an instruction if a capability for that segment has been loaded into a capability register.

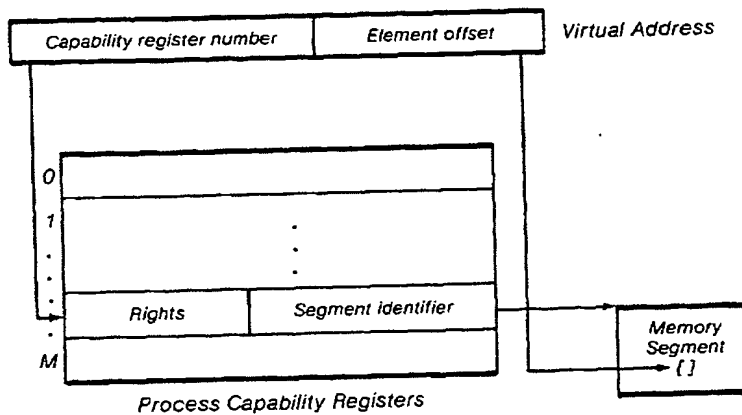


Figure 1-3: Capability Register Addressing

2. While loading of a segment descriptor in the conventional system is privileged, loading of a capability register is not. Instead of controlling the loading of the register, the capability system controls *the pattern of bits* that can be loaded. Only a valid capability can be loaded into a capability register.
3. The capability system provides a dynamically changing address space. The address space changes whenever the program changes one of the capability registers.
4. A virtual address identifies a process-local capability register. In this sense, a virtual address has similar properties to a virtual address in the conventional system. Sharing a virtual address does not in itself give access to the same segment.
5. A capability, however, is *not* process-local. Capabilities are *context independent*; that is, the segment addressed by a capability is independent of the process using that capability. A process can share a segment by copying or sending a capability from its capability list to the capability list of a cooperating process. Each of the processes can then access the segment.

One important difference between the conventional and capability approaches involves the ability of a program to affect system-wide or process-local objects. In the conventional system, a program executes within a virtual address space defined by a process. Every procedure called by that program has access to the process address space, including segments and files. Every procedure executes within an identical protection environment.

In the capability system, a procedure can only affect objects for which capability registers have been loaded. It is possible, therefore, for different procedures called by the same program to have access to different segments. Although all procedures may have the potential to load capability registers from the capability list, some procedures may choose to execute within a very small addressing sphere.

The ability to restrict the execution or addressing environment of a procedure has several benefits. First, if a procedure is allowed access only to those segments absolutely needed, the hardware can detect any erroneous references. For example, a reference past the end of an array might be caught before it destroys another variable. Second, if a procedure is found to be in error, it is easy to determine what segments might have been affected. If the segments that could have been modified were local to the procedure, recovery might be substantially easier.

Most capability systems go a step further by allowing each procedure to have a *private* capability list. A procedure can

thus protect its objects from accidental or malicious access by its callers, and a program can protect its objects from access by called procedures. Every procedure can have, in effect, its own address space. To permit a procedure access to a local object, a program can pass a capability for the object as a parameter when the procedure is called. Therefore, in a capability system, every procedure can be protected from every other procedure because each has a private capability list. When one procedure calls another, it knows that the called procedure can access only local objects for which capabilities are passed.

1.1.2 The Context of an Address

Each object in a capability system has a unique identifier. Conceptually, each object's identifier is unique for all time. That is, an identifier is assigned when an object is created and that identifier is never reused, even after the object is deleted. During the object's lifetime, its unique identifier is used within capabilities to specify the object. An attempt to use a capability with an identifier for a deleted object causes an error.

In practice, the object identifier field of a capability must be used by hardware to locate the object. From the hardware viewpoint, the identifier is an address—either the address of a segment or perhaps the address of a central descriptor that contains physical information about the segment. The need to handle addresses efficiently in hardware typically causes addresses to be small—16 or 32 bits, for example. For this reason, identifiers tend to have too few bits to be unique for all time. However, the choice of the number of bits in an identifier is an important system design decision that dictates the way in which capabilities can be used.

In conventional systems, an address is meaningful only within a single process. In a capability system, addresses (capabilities and their identifiers) are context-independent. That is, the interpretation of a capability is independent of the process using it. The unique identifier within a capability must have a system-wide interpretation. Unique identifiers must be large enough to address all of the segments likely to be in use by all executing processes at any time. This allows capabilities to be freely passed between processes and used to access shared data.

Addressing on most conventional systems is restricted in terms of time as well as context. An address is meaningful only within the lifetime of a single process. Therefore, addresses cannot be used to name objects whose lifetimes are greater than

the process creating the objects. If a process wishes to create a long-term storage object, such as a file, it must interface to the file system. Files typically require different naming, protection, and storage mechanisms than memory segments.

A significant advance made possible by capabilities is the naming and protection of both long-term and short-term objects with a single mechanism. If the identifier field is very large, it may be possible to implement identifiers unique for all time. Each object is addressed by capabilities containing its unique identifier, independent of whether it is stored in primary or secondary memory. The operating system or hardware can maintain data structures that indicate the location of each object. If a program attempts to access an object in secondary memory, the hardware or operating system can bring the object into primary memory so that the operation can be completed. From the program's point of view, however, there is a single-level address space. Capabilities, as well as data, can be saved for long periods of time and stored in secondary memory.

There are, therefore, several contexts in which an address can have meaning. For example, for:

1. Primary memory segments of a single process.
2. Primary memory segments of all existing processes.
3. All existing segments in both primary and secondary memory.

Most conventional systems support only type 1, while capabilities allow for any of the listed addressing types. More importantly, while conventional systems are concerned only with the protection of *data*, capability systems are concerned also with the protection of *addresses*. A process on a capability system cannot fabricate new addresses. As systems become more general in their addressing structure as in types 2 and 3, the protection of addresses becomes crucial to the integrity of the system.

1.1.3 Protection in Computer Systems

Lampson contrasts the capability approach with the traditional approach by showing the structure of protection information needed in a traditional operating system [Lampson 71]. Figure 1-4 depicts an access matrix showing the privileges that each system user is permitted with respect to each system object. For example, user Fred has read and write privileges to File1 and no privileges to File2, while user Sandy is allowed to read both files.

		System Objects					
		File1	File2	File3	ProcessJ	Mailbox10	...
System Users	Fred	Read Write		Read	Delete Suspend Wakeup	Send	
	Sandy	Read	Read			Send Receive	
	Molly			Read Write		Send	
	.						
	.						

Figure 1-4: System Object Access Matrix

One conventional approach to the maintenance of protection information is *access control lists*, in which the operating system keeps an *access list* for each object in the system. Each object's list contains the names of users permitted access to the object and the privileges they may exercise. When a user attempts to access an object, the operating system checks the access list associated with that object to see if the operation is authorized. Each of the columns of Figure 1-4 represents an access control list.

The capability system offers an alternative structure in which the operating system arranges protection information by user instead of by object. A *capability list* is associated with each user in the system. Each capability contains the name of an object in the system and the user's permitted privileges for accessing the object. To access an object, the user specifies a capability in the local capability list. Each of the rows of Figure 1-4 represents a capability list. Figure 1-5 shows an access list

<u>Access List for Mailbox10</u>	<u>Capability list for Fred</u>
Fred(send)	File1(read,write)
Sandy(send,receive)	File3(read)
Molly(send)	ProcessJ(delete,suspend,wakeup)
.	Mailbox10(send)
.	.
.	.

Figure 1-5: Access Control and Capability Lists

and a capability list derived from the protection matrix in Figure 1-4.

One important difference between the capability list and access list is the user's ability to *name* objects. In the access list approach, a user can attempt to name any object in the system as the target of an operation. The system then checks that object's access list. In the capability system, however, a user can only name those objects for which a capability is held: that is, to which some access is permitted.

In either case, the integrity of the system is only as good as the integrity of the data structures used to maintain the protection information. Both access control list and capability list mechanisms must be carefully controlled so that users cannot gain unauthorized access to an object.

Similar protection options exist outside the computer world. A useful analogy is the control of a safe deposit box. Suppose, for example, that Carla wishes to keep all of her valuables in a safe deposit box in the bank. On occasion, she would like one or more trustworthy friends to make deposits or withdrawals. There are basically two ways that the bank can control access to the box. First, the bank can maintain a list of people authorized to access the box. To make a transaction, Carla or any of her friends must prove their identity to the bank's satisfaction. The bank checks the (access control) list for Carla's safe deposit box and allows the transaction if the person is authorized. Or, instead of maintaining a list, the bank can issue Carla one or more keys to her safe deposit box. If Carla needs to have a friend access the box, she simply gives a key to the friend.

A number of observations can be made about these two alternative protection systems. The properties of the access list scheme are:

1. The bank must maintain a list for each safe deposit box.
2. The bank must ensure the validity of the list at all times (e.g., it cannot allow the night watchman to add a name).
3. The bank must be able to verify the identity of those asking to use a box.
4. To allow a new person to use the box, the owner must visit the bank, verify that he or she is the owner of the box, and have the new name added to the list.
5. A friend cannot extend his or her privilege to someone else.
6. If a friend becomes untrustworthy, the owner can visit the bank and have that person's name removed from the list.

The alternative scheme involving keys has the following properties:

1. The bank need not be involved in any transactions once the keys are given, except to allow a valid keyholder into the vault.
2. The physical lock and key system must be relatively secure; that is, it must be extremely difficult to forge a key or to pick the lock on a safe deposit box.
3. The owner of a box can simply pass a key to anyone who needs to access the box.
4. Once a key has been passed to a friend, it is difficult to keep them from giving the key to someone else.
5. Once a friend has made a transaction, the owner can ask for the key back, although it may not be possible to know whether or not the friend has made a copy.

The advantage of the key-based system is ease of use for both the bank and customer. However, if today's friends are likely to become tomorrow's enemies, the access list has the advantage of simple guaranteed access removal. Of course, the access control list and the key (or capability) systems are not mutually exclusive, and can be combined in either the computer or banking world to provide the advantages of both systems for increased protection.

1.2 The Object-Based Approach

Over the last few decades, several areas of computer science have converged on a single approach to system design. This approach, known as *object-based computing*, seeks to raise the level of abstraction in system design. The events that have encouraged object-based design include:

1. Advances in computer architecture, including capability systems and hardware support for operating systems concepts.
2. Advances in programming languages, as demonstrated in Simula [Dahl 66], Pascal [Jensen 75], Smalltalk [Ingalls 78], CLU [Liskov 77], and Ada [DOD 80].
3. Advances in programming methodology, including modularization and information hiding [Parnas 72] and monitors [Hoare 74].

This section introduces the object approach and discusses its relationship to capability-based computer systems.

What is object-based computing? Simply stated, the object approach is a method of structuring systems that supports *ab-*

straction. It is a philosophy of system design that decomposes a problem into (1) a set of *abstract object types*, or resources in the system, and (2) a set of *operations* that manipulate *instances* of each object type.

To make this idea more concrete, consider the following simplified example. Imagine that we are programming a traffic simulation for a city. First, define a set of objects that represent, abstractly, the fundamental entities that make up the traffic system. Some of the object *types* for the traffic simulation might be:

- passenger
- bus
- bus stop
- taxi
- car

Then, for each object type, define the operations that can be performed. Bus objects, for example, might support the operations:

- PUT_BUS_INTO_SERVICE(bus_number)
- MOVE_BUS(bus_number, bus_stop)
- LOAD_PASSENGERS(bus_number, passenger_list)
- UNLOAD_PASSENGERS(bus_number, passenger_list)
- GET_PASSENGER_COUNT(bus_number)
- GET_POSITION(bus_number)
- REMOVE_BUS_FROM_SERVICE(bus_number)

Each bus operation accepts a bus number as a parameter. At any time there may be many bus objects in the system, and we identify each bus by a unique number. Each of these bus objects is an *instance* of the *type* bus. The *type* of an object identifies it as a member of a class of objects that share some behavioral properties, such as the set of operations that can be performed on them.

What has been gained by defining the system in this way? First, there now exist a fundamental set of objects and operations for the simulation. We can now implement the procedures to perform the operations on each type of object. Since only a limited number of procedures operate on each object type, access to the internal data structures used to maintain the state of each type can be restricted. This isolation of the knowledge of those data structures should simplify any future

changes to one of the object abstractions because only a limited set of procedures is affected.

Second, and more importantly, we have raised the *level of abstraction* in the simulation program. That is, we can now program the simulation using buses, passengers, and bus stops as the fundamental objects, instead of bits, bytes, and words, which are normally provided by the underlying hardware. The buses and passengers are our data types just as bits and bytes are the data types supported in hardware. The simulation program will consist mainly of control structures plus procedure calls to perform operations on instances of our fundamental objects.

Of course, in this example, the procedures implementing the operations are programmed using lower-level objects, such as bytes, words, and so on. Or, they may be further decomposed into simpler abstract objects that are then implemented at a low level. Object-based systems provide a fundamental set of objects that can be used for computing. From this basis, the programmer constructs new higher-level object types using combinations of the fundamental objects. In this way the system is extended to provide new features by creating more sophisticated abstractions.

This methodology aims to increase productivity, improve reliability, and ease system modification. Through the use of well-defined and well-controlled object interfaces, systems designers hope to simplify the construction of complex computer systems.

1.2.1 Capabilities and Object-Based Systems

In the simulation example, each object is identified by a unique number. To move a bus from one stop to another, we call the MOVE_BUS operation with the unique number of the bus to move. For purposes of the simple simulation, a small set of integers suffices to identify the buses or other objects. No protection is needed because these objects are implemented and used by a single program.

The use of the object approach to build operating system facilities presents different requirements. For example, suppose we wish to build a calendar system to keep track of scheduled meetings, deadlines, reminders, and so on. The fundamental object of the calendar system, from the user's point of view, is a calendar object. Our calendar management system provides routines that create a new calendar, and modify,

query, or display an existing calendar. Many users in the system will, of course, want to use this facility.

Several familiar issues now arise: (1) how does a user name a calendar object, (2) how is that calendar protected from access by other users, and (3) how can calendars be shared under controlled circumstances? Only the owner of a calendar should be able to make changes, and the annotations in each calendar must be protected from other users, since they might contain confidential information. However, a user might permit selected other users to check if he or she is busy during a certain time, in order to automate the scheduling of meetings.

Capabilities provide a solution to these problems. When a user creates a new calendar, the calendar creation routine allocates a segment of memory for which it receives a capability. This segment is used to store data structures that will hold the calendar's state. The create routine uses this capability to initialize the data structures, and then returns it to the caller as proof of ownership of the calendar. In order to later modify or query the calendar, the user specifies the returned capability; the capability identifies the calendar and allows the modify or query procedure to gain access to the data structures. Only a user with a valid capability can access a calendar.

A weakness with this scenario is that the calendar system cannot prevent the calendar owner from using its capability to access the data structures directly. The calendar system would like to protect its data structures both to ensure consistency and to guarantee that future changes in data format are invisible outside of the subsystem. In addition, if a user passes a calendar capability to another user, the second user can then modify the data structures or read confidential information.

These problems exist because the calendar system returns a fully-privileged calendar capability to the user. Instead, what is needed is a capability that identifies a specific calendar and is proof of ownership, but does not allow direct access to the underlying data structures. In other words, the calendar system would like to return only *restricted* capabilities to its clients. However, the calendar system must retain the ability to later *amplify* the privileges in one of its restricted capabilities so that it can access the data structures for a calendar.

There are several ways of providing type managers with this special ability. (These mechanisms are examined in detail throughout the book.) However, given this power over capabilities for its objects, a type manager can ensure that its clients operate only through the well-defined object operation interface. A client can pass a capability parameter to the type man-

ager when requesting a service, but cannot otherwise use the capability to read or write the object it addresses. This facility is fundamental to any system that allows creation and protection of new system types.

1.3 Summary

The capability concept can be applied in hardware and software to many problems in computer system design. Capabilities provide a different way of thinking about addressing, protection, and sharing of objects. Some of the properties of capabilities illustrated in this chapter include their use in:

1. Addressing primary memory in a computer system.
2. Sharing objects.
3. Providing a uniform means of addressing short- and long-term storage.
4. Support for a dynamic addressing environment.
5. Support for data abstraction and information hiding.

These, of course, are advantages of capability-based systems. The most important advantage is support for object-based programming. Object-based programming methodology seeks to simplify the design, implementation, debugging, and maintenance of sophisticated applications. While capabilities solve a number of system problems, their use raises a whole new set of concerns. And, as is often the case in computer system design, the concept is much simpler than the implementation.

The remainder of this book is devoted to examining many different capability-based and object-based designs. The characteristics of each system are described with emphasis on addressing, protection, and object management. Each system represents a different set of tradeoffs and presents different advantages and disadvantages. When comparing the systems, consider the differences in goals, technologies, and resources available to the system developers.

The final chapter of this book considers issues in capability system design common to all of the systems described. A few of the questions to be considered follow. It may be useful to remember these questions when examining each system design.

1. What is the structure of an address?
2. How is a capability represented? How is a capability used to locate an object?

3. How are capabilities protected?
4. What is the lifetime of a capability?
5. What types of objects are supported by the hardware and software?
6. What is the lifetime of an object?
7. How can users extend the primitive set of objects provided by the base hardware and software?

1.4 For Further Reading

The concept of capability is formally defined in the 1966 paper by Dennis and Van Horn [Dennis 66]. Chapter 3 examines this paper in some detail. The paper by Fabry [Fabry 74] compares capability addressing and conventional segmented addressing of primary memory, while Redell [Redell 74a] describes issues in capability systems and the use of sealing mechanisms that support the addition of new object types to a system. These papers are a fundamental part of capability literature.

Capability systems have been discussed in various contexts. Two papers by Lampson [Lampson 69 and Lampson 71] describe the requirements for protection in operating systems and the capability protection model. The surveys by Linden [Linden 76] and Denning [Denning 76], which appeared in a special issue of *ACM Computing Surveys*, describe capability systems and their relationship to security and fault tolerance in operating systems.

The architecture books by Myers [Myers 82] and Iliffe [Iliffe 82] also discuss some of the systems described in this book. Myers' book contains details of Sward [Myers 80], a capability-based research system built at IBM that is omitted here. A capability system model, as well as discussion of some existing capability systems, appears in the book by Gehringer [Gehringer 82]. Jones [Jones 78a] provides a good introduction to the concepts of object-based programming.



The Burroughs B5000 computer. (Courtesy Burroughs Corporation.)

**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.



JFW

TRANSMITTAL FORM <i>(to be used for all correspondence after initial filing)</i>	Application Number	11/304,794	
	Filing Date	December 16, 2005	
	First Named Inventor	Mark J. STEFIK, <i>et al.</i>	
	Group Art Unit	3621	
	Examiner Name	Not Yet Assigned	
Total Number of Pages in This Submission		Attorney Docket Number	111325-028700

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 ☐ Appeal Communication to Board of Appeals and Interferences ☐ Appeal Communication to Group (Appeal Notice, Brief, Reply Brief) ☐ Proprietary Information ☐ Status Letter ☐ Application Data Sheet ☐ Request for Corrected Filing Receipt with Enclosures ☐ A self-addressed prepaid postcard for acknowledging receipt ☒ Other Enclosure(s) (please identify below): 1. PTO Form 1449 2. Fourteen Cited References
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	Carlos R. Villamar Registration No. 43,224 Nixon Peabody LLP 401 9 th Street, N.W., Suite 900 Washington, D.C. 20004-2128
Signature	/Carlos R. Villamar, Reg. # 43,224/
Date	August 25, 2006

CERTIFICATE OF MAILING OR TRANSMISSION [37 CFR 1.8(a)]	
I hereby certify that this correspondence is being:	
☐ deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop _____, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450	
☐ transmitted by facsimile on the date shown below to the United States Patent and Trademark Office at (703) _____.	
Date	Signature
	Typed or printed name



Attorney Docket No: 111325-028700
Application No. 11/304,794

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of	)	
Mark J. STEFIK, <i>et al.</i>	)	
Application No. 11/304,794	)	Group Art Unit: 3621
Filed: December 16, 2005	)	Confirmation No. 6096
For: SYSTEM FOR CONTROLLING	)	
THE DISTRIBUTION AND USE OF	)	Examiner: Not Yet Assigned
DIGITAL WORKS	)	

INFORMATION DISCLOSURE STATEMENT

United States Patent and Trademark Office
Customer Service Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

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.

This document is being submitted before the first Office Action on the merits, therefore no fee or certification is required under 37 C.F.R. § 1.97 (b).

It is requested that the accompanying PTO-1449 be considered and made of record in the above-identified application. To assist the Examiner, the documents are listed on the attached form PTO-1449. It is respectfully requested that an Examiner return an initial copy of this form to the undersigned.

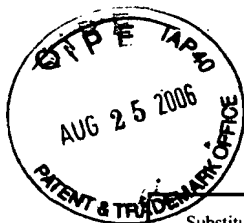
The Commissioner is hereby authorized to charge any fees connected with this filing which may be required now, or credit any overpayment to Deposit Account No. 19-2380 (111325-028700).

Respectfully submitted,
NIXON PEABODY, LLP

/Carlos R. Villamar, Reg. # 43,224/
Carlos R. Villamar
Registration No. 43,224

Date: **August 25, 2006**

CUSTOMER NO. 22204
NIXON PEABODY LLP
401 9th Street, N.W., Suite 900
Washington, DC 20004-2128
Telephone: (202) 585-8000



Substitute for form 1449A/PTO				Complete if Known		
INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Application Number	11/304,794	
				Filing Date	December 16, 2005	
				First Named Inventor	Mark J. STEFIK, <i>et al.</i>	
				Art Unit	3621	
				Examiner Name	Not Yet Assigned	
Sheet		of		Attorney Docket Number	111325-028700	
U.S. PATENT DOCUMENTS						
Examiner Initials*	Cite No. ¹	U.S. Patent Document Number - Kind Code ² (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	
		US-				
		US-				
FOREIGN PATENT DOCUMENTS						
Examiner Initials*	Cite No. ¹	Foreign Patent Document Country Code ³ Number ⁴ Kind Code ⁵ (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T ⁶
		JP 06-214862	08/05/1994	SATOMI et al.	English Abstract	
		JP 05-298174	11/12/1993	NUKUI	English Abstract	
		JP 06-230847	08/19/1994	MASUI	English Abstract	
		JP 03-014109	01/22/1991	NARITA	English Abstract	
		JP 06-318167	11/15/1994	TAKAHASI	English Abstract	
		JP 06-103286	04/15/1994	HARTRICK et al.	English Abstract	
OTHER PRIOR ART - NON PATENT LITERATURE DOCUMENTS						
Examiner Initials*	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
		Henry M. Levy, "Capability-Based Computer Systems", Digital Press, 1984, P1-18 USA				
		Kawahara Masaharu, "Consideration of a Method of Charging for Electronic Objects in Superdistribution", Technical Report of the Institute of Electronics, Information and Communication Engineers, EIC Institute of Electronics, Information and Communication Engineers, September 21, 1994, Shingaku Giho Vol. 94 No. 240, P17-24 Japan - English Translation				
		Ryoichi Mori et al., "The Inevitable Way Toward Superdistribution", Information Symposium Papers, Information Processing Society of Japan, February 17, 1994, Vol. 94, No. 1, P67-76 Japan				
		Shinichi Ueki et al., "The Accounting Process in Software Usage Monitor for Superdistribution", Information Processing Society of Japan Technical Report, 90-Is-27, January 16, 1990, Vol. 90, No. 1, P1-10 Japan English Translation				
		Naoya Torii et al., "System Architecture of Superdistribution", Technical Report of the Institute of Electronics, Information and Communication Engineers, EIC Institute of Electronics, Information and Communication Engineers, September 21, 1994, Shingaku Giho Vol. 94 No. 240, P59-66 Japan				
		Yasushi Kuho et al., "Information Technology System Security for Open Distributed Computing Environment", The Hitachi Hyoron, November 1994, Vol. 76, P49-52 Japan				
		Hiroyoshi Takeda, "Function and Use of Workstation" The Office Management, Nikkan Kogyo Shinbun, August 1, 1989, Vol. 28, No. 8, P8-17 Japan				
		Atsushi Iizawa, "Document Image Database System", Information Processing Society, May 15, 1992, Vol. 33, No. 5, P497-504 Japan				
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 Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. 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 Applicant's unique citation designation number (optional). ² Applicant is to place a check mark here if English language Translation is attached.

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-103286
 (43)Date of publication of application : 15.04.1994

(51) Int. Cl. G06F 15/21
 G06F 15/20
 G06F 15/20

(21)Application number : 05-082568 (71)Applicant : INTERNATL BUSINESS MACH
 CORP <IBM>
 (22)Date of filing : 16.03.1993 (72)Inventor : HARTRICK THOMAS V
 STEVENS JEFFREY N
 SABIA NICHOLAS J

(30)Priority

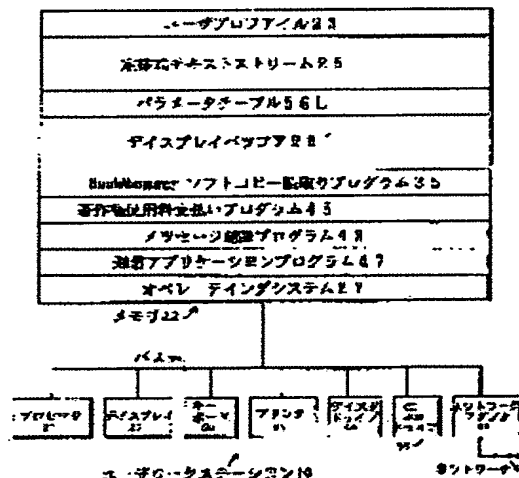
Priority number : 92 875919 Priority date : 29.04.1992 Priority country : US

(54) METHOD FOR MANAGING COPY OF STRUCTURED DOCUMENT AND DEVICE FOR THE SAME

(57)Abstract:

PURPOSE: To improve a method for managing the copy of the soft copy book of a structured document.

CONSTITUTION: The publishing company of a soft copy book includes copyright fee payment information in either the structured document text of the book or a copyright fee payment information file accompanied with the book. A user uses a soft copy book reading program 35 in a work station 10 in order to read the book from a storage disk. When the user inputs a command for copying the book to a writing enabling recording medium 46 such as a magnetic disk, inputs a command for printing the hard copy of the book by a printer 44, or inputs command for transmitting the copy of the book through a modem, a copyright fee payment program 45 intercepts those copy commands, and interrupts those copying operations. The copyright fee payment program 45 presents the copyright fee payment information stored in the book text or the file accompanied with the book to the user.



LEGAL STATUS

[Date of request for examination] 19.05.1993
 [Date of sending the examiner's decision of rejection]
 [Kind of final disposal of application]

✓ 

other than the examiner's decision of
rejection or application converted
registration]

[Date of final disposal for
application]

[Patent number] 2659896

[Date of registration] 06. 06. 1997

[Number of appeal against examiner's
decision of rejection]

[Date of requesting appeal against
examiner's decision of rejection]

[Date of extinction of right] 06. 06. 2003

全項目

- (19) 【発行国】日本国特許庁 (JP)
 (12) 【公報種別】公開特許公報 (A)
 (11) 【公開番号】特開平6-103286
 (43) 【公開日】平成6年(1994)4月15日
 (54) 【発明の名称】構造化文書複製管理方法、構造化文書複製管理装置及び構造化文書複製管理プログラム
 (51) 【国際特許分類第5版】

G06F 15/21 350 7052-5L
 15/20 566 D 9288-5L
 570 M 9288-5L

【審査請求】有

【請求項の数】9

【全頁数】31

- (21) 【出願番号】特願平5-82568
 (22) 【出願日】平成5年(1993)3月16日
 (31) 【優先権主張番号】07/875919
 (32) 【優先日】1992年4月29日
 (33) 【優先権主張国】米国 (US)
 (71) 【出願人】
 【識別番号】390009531
 【氏名又は名称】インターナショナル・ビジネス・マシーンズ・コーポレイション
 【氏名又は名称原語表記】INTERNATIONAL BUSINESS MACHINES CORPORATION
 【住所又は居所】アメリカ合衆国10504、ニューヨーク州 アーモンク (番地なし)
 (72) 【発明者】
 【氏名】トーマス・ブイ・ハートリック
 【住所又は居所】アメリカ合衆国、メリーランド州20878、ゲイサースバーグ、シャープステッド・レーン 170番地
 (72) 【発明者】
 【氏名】ジェフリー・エヌ・ステイブンス
 【住所又は居所】アメリカ合衆国、メリーランド州20878、ゲイサースバーグ、サンドパイパー・レーン 18637番地
 (72) 【発明者】
 【氏名】ニコラス・ジェイ・サビア
 【住所又は居所】アメリカ合衆国、メリーランド州20906、シルバー・スプリング、ドウルリー・ロード 2213番地
 (74) 【代理人】
 【弁理士】
 【氏名又は名称】頓宮 孝一 (外3名)

(57) 【要約】

【目的】構造化文書のソフトコピーブックの複製を管理する方法を改善する。

【構成】ソフトコピーブックの出版社は当該ブックの構造化文書テキスト又は当該ブックに付随する著作権使用料支払い情報ファイルのいずれかに著作権使用料支払い情報を含む。ブックを記憶ディスクから読み取るためにユーザはワークステーション10でソフトコピーブック読取りプログラム35を用いる。ユーザが磁気ディスクのような書込み可能記憶媒体46にブックを複製するコマンドを入力するか、プリンタ44でブックのハードコピーを印刷するコマンドを入力するか又はモデムを介してブックの複製物を伝送するコ

ル

マンドを入力すると、著作権使用料支払いプログラム４５がこれらの複製コマンドをインターセプトしてこれらの複製動作を中断させる。著作権使用料支払いプログラム４５はブックテキスト又はブックに付随するファイルに格納された著作権使用料支払い情報をユーザに提示する。

【特許請求の範囲】

【請求項１】構造化文書の著作権使用料支払い要件に従うように、ネットワークのプリンタ装置における上記構造化文書のページの印刷を管理する方法において、特殊タグを有する著作権使用料支払いエレメントを含む構造化文書の定様式テキストストリームを上記ネットワークの第１のデータプロセッサに入力するステップと、上記定様式テキストストリームを探索することにより上記著作権使用料支払いエレメントを識別し、著作権使用料支払いフラグを上記第１のデータプロセッサのメモリに格納するステップと、ユーザから印刷コマンドを受信することにより、上記第１のデータプロセッサによつて制御されるプリンタ装置において上記構造化文書の一部を印刷するステップと、上記印刷コマンド及び上記著作権使用料支払いフラグに応答して、上記第１のデータプロセッサから上記ネットワークの第２のデータプロセッサに上記構造化文書の上記一部を印刷するための許可についての要求を伝送するステップと、上記第２のデータプロセッサにおいて上記要求を受け取り、これに回答して上記ユーザに著作権使用料支払い総額を請求するステップと、上記第２のデータプロセッサから上記第１のデータプロセッサに許可メッセージを伝送することにより上記構造化文書の上記一部を印刷するステップと、上記第１のデータプロセッサにおいて上記許可メッセージを受け取り、これに回答して上記第１のデータプロセッサから上記プリンタ装置に印刷制御信号を出力することにより、上記プリンタ装置において上記構造化文書の上記一部を印刷するステップとを含むことを特徴とする構造化文書複製管理方法。

【請求項２】構造化文書の著作権使用料支払い要件に従うように、ネットワークの複写装置における上記構造化文書のページの複写を管理する方法において、特殊タグを有する著作権使用料支払いエレメントを含む構造化文書の定様式テキストストリームを上記ネットワークの第１のデータプロセッサに入力するステップと、上記定様式テキストストリームを探索することにより上記著作権使用料支払いエレメントを識別し、著作権使用料支払いフラグを上記第１のデータプロセッサのメモリに格納するステップと、ユーザから複写コマンドを受信することにより、上記第１のデータプロセッサによつて制御される複写装置において上記構造化文書の一部を複写するステップと、上記複写コマンド及び上記著作権使用料支払いフラグに応答して、上記第１のデータプロセッサから上記ネットワークの第２のデータプロセッサに上記構造化文書の上記一部を複写するための許可についての要求を伝送するステップと、上記第２のデータプロセッサにおいて上記要求を受け取り、これに回答して上記ユーザに著作権使用料支払い総額を請求するステップと、上記第２のデータプロセッサから上記第１のデータプロセッサに許可メッセージを伝送することにより上記構造化文書の上記一部を複写するステップと、上記第１のデータプロセッサにおいて上記許可メッセージを受け取り、これに回答して上記第１のデータプロセッサから上記複写装置に複写制御信号を出力することにより、上記複写装置において上記構造化文書の上記一部を複写するステップとを含むことを特徴とする構造化文書複製管理方法。

【請求項３】構造化文書の著作権使用料支払い要件に従うように、ネットワークの通信装置における上記構造化文書のページの通信を管理する方法において、特殊タグを有する著作権使用料支払いエレメントを含む構造化文書の定様式テキストストリームを上記ネットワークの第１のデータプロセッサに入力するステップと、上記定様式テキストストリームを探索することにより上記著作権使用料支払いエレメントを識別し、著作権使用料支払いフラグを上記第１のデータプロセッサのメモリに格納するステップと、ユーザから通信コマンドを受信することにより、上記第１のデータプロセッサによつて制御される通信装置において上記構造化文書の一部を通信するステップと、上記通信コマンド及び上記著作権使用料支払いフラグに応答して、上記第１のデータプロセッサから上記ネットワークの第２のデータプロセッサに上記構造化文書の上記一部を通信するための許可についての要求を伝送するステップと、上記第２のデータプロセッサにおいて上記要求を受け取り、これ

に、上記第2のデータプロセッサから上記第1のデータプロセッサに許可メッセージを送信することにより上記構造化文書の上記一部を通信するステップと、上記第1のデータプロセッサにおいて上記許可メッセージを受け取り、これに応答して上記第1のデータプロセッサから上記通信装置に通信制御信号を出力することにより、上記通信装置において上記構造化文書の上記一部を通信するステップとを含むことを特徴とする構造化文書複製管理方法。

【請求項4】実行されたとき、構造化文書の著作権使用料支払い要件に従うように、ネットワークのプリンタ装置における上記構造化文書のページの印刷を管理する方法を実行するコンピュータプログラムにおいて、特殊タグを有する著作権使用料支払いエレメントを含む構造化文書の定様式テキストストリームを上記ネットワークの第1のデータプロセッサに入力するステップと、上記定様式テキストストリームを探索することにより上記著作権使用料支払いエレメントを識別し、著作権使用料支払いフラグを上記第1のデータプロセッサのメモリに格納するステップと、ユーザから印刷コマンドを受信することにより、上記第1のデータプロセッサによつて制御されるプリンタ装置において上記構造化文書の一部を印刷するステップと、上記印刷コマンド及び上記著作権使用料支払いフラグに応答して、上記第1のデータプロセッサから上記ネットワークの第2のデータプロセッサに上記構造化文書の上記一部を印刷するための許可についての要求を送信するステップと、上記第2のデータプロセッサにおいて上記要求を受け取り、これに応答して上記ユーザに著作権使用料支払い総額を請求するステップと、上記第2のデータプロセッサから上記第1のデータプロセッサに許可メッセージを送信することにより上記構造化文書の上記一部を印刷するステップと、上記第1のデータプロセッサにおいて上記許可メッセージを受け取り、これに応答して上記第1のデータプロセッサから上記プリンタ装置に印刷制御信号を出力することにより、上記プリンタ装置において上記構造化文書の上記一部を印刷するステップとを含むことを特徴とする構造化文書複製管理プログラム。

【請求項5】実行されたとき、構造化文書の著作権使用料支払い要件に従うように、ネットワークの複写装置における上記構造化文書のページの複写を管理する方法を実行するコンピュータプログラムにおいて、特殊タグを有する著作権使用料支払いエレメントを含む構造化文書の定様式テキストストリームを上記ネットワークの第1のデータプロセッサに入力するステップと、上記定様式テキストストリームを探索することにより上記著作権使用料支払いエレメントを識別し、著作権使用料支払いフラグを上記第1のデータプロセッサのメモリに格納するステップと、ユーザから複写コマンドを受信することにより、上記第1のデータプロセッサによつて制御される複写装置において上記構造化文書の一部を複写するステップと、上記複写コマンド及び上記著作権使用料支払いフラグに応答して、上記第1のデータプロセッサから上記ネットワークの第2のデータプロセッサに上記構造化文書の上記一部を複写するための許可についての要求を送信するステップと、上記第2のデータプロセッサにおいて上記要求を受け取り、これに応答して上記ユーザに著作権使用料支払い総額を請求するステップと、上記第2のデータプロセッサから上記第1のデータプロセッサに許可メッセージを送信することにより上記構造化文書の上記一部を複写するステップと、上記第1のデータプロセッサにおいて上記許可メッセージを受け取り、これに応答して上記第1のデータプロセッサから上記複写装置に複写制御信号を出力することにより、上記複写装置において上記構造化文書の上記一部を複写するステップとを含むことを特徴とする構造化文書複製管理プログラム。

【請求項6】実行されたとき、構造化文書の著作権使用料支払い要件に従うように、ネットワークの通信装置における上記構造化文書のページの通信を管理する方法を実行するコンピュータプログラムにおいて、特殊タグを有する著作権使用料支払いエレメントを含む構造化文書の定様式テキストストリームを上記ネットワークの第1のデータプロセッサに入力するステップと、上記定様式テキストストリームを探索することにより上記著作権使用料支払いエレメントを識別し、著作権使用料支払いフラグを上記第1のデータプロセッサのメモリに格納するステップと、ユーザから通信コマンドを受信することにより、上記第1のデータプロセッサによつて制御される通信装置において上記構造化文書の一部を通信するステップと、上記通信コマンド及び上記著作権使用料支払いフラグに応答して、上記第1のデータプロセッサから上記ネットワークの第2のデータプロセッサに上記構造化文書の上記一部を通信するための許可についての要求を送信するステップと、上記第2のデータプロセッサにおいて上記要求を受け取り、これに応答して上記ユーザに著作権使用

料支払い総額を請求するステップと、上記第2のデータプロセッサから上記第1のデータプロセッサに許可メッセージを送送することにより上記構造化文書の上記一部を通信するステップと、上記第1のデータプロセッサにおいて上記許可メッセージを受け取り、これに回答して上記第1のデータプロセッサから上記通信装置に通信制御信号を出力することにより、上記通信装置において上記構造化文書の上記一部を通信するステップとを含むことを特徴とする構造化文書複製管理プログラム。

【請求項7】構造化文書の著作権使用料支払い要件に従うように、ネットワークのプリンタ装置における構造化文書のページの印刷を管理するデータ処理システムにおいて、上記システムは、特殊タグを有する著作権使用料支払いエレメントを含む構造化文書の定様式テキストストリームを上記ネットワークの第1のデータプロセッサに入力する入力手段と、上記入力手段に結合され、上記定様式テキストストリームを探索することにより上記著作権使用料支払いエレメントを識別し、著作権使用料支払いフラグを上記第1のデータプロセッサのメモリに格納する探索手段と、上記メモリに結合され、ユーザから印刷コマンドを受信することにより、上記第1のデータプロセッサによつて制御されるプリンタ装置において上記構造化文書の一部を印刷する第1の受信手段と、上記第1の受信手段に結合され、上記印刷コマンド及び上記著作権使用料支払いフラグに回答して、上記第1のデータプロセッサから上記ネットワークの第2のデータプロセッサに上記構造化文書の上記一部を印刷するための許可についての要求を送送する第1の伝送手段と、上記第2のデータプロセッサにおいて上記要求を受け取り、これに回答して上記ユーザに著作権使用料支払い総額を請求する、上記第2のデータプロセッサにおける第2の受信手段と、上記第2の受信手段に結合され、上記第2のデータプロセッサから上記第1のデータプロセッサに許可メッセージを送送することにより上記構造化文書の上記一部を印刷する第2の伝送手段と、上記第1のデータプロセッサにおいて上記許可メッセージを受け取り、これに回答して上記第1のデータプロセッサから上記プリンタ装置に印刷制御信号を出力することにより、上記プリンタ装置において上記構造化文書の上記一部を印刷する、上記第1のデータプロセッサにおける許可メッセージ受信手段とを具えることを特徴とする構造化文書複製管理装置。

【請求項8】構造化文書の著作権使用料支払い要件に従うように、ネットワークの複写装置における上記構造化文書のページの複写を管理するデータ処理システムにおいて、上記システムは、特殊タグを有する著作権使用料支払いエレメントを含む構造化文書の定様式テキストストリームを上記ネットワークの第1のデータプロセッサに入力する入力手段と、上記入力手段に結合され、上記定様式テキストストリームを探索することにより上記著作権使用料支払いエレメントを識別し、著作権使用料支払いフラグを上記第1のデータプロセッサのメモリに格納する探索手段と、上記メモリに結合され、ユーザから複写コマンドを受信することにより、上記第1のデータプロセッサによつて制御される複写装置において上記構造化文書の一部を複写する第1の受信手段と、上記第1の受信手段に結合され、上記複写コマンド及び上記著作権使用料支払いフラグに回答して、上記第1のデータプロセッサから上記ネットワークの第2のデータプロセッサに上記構造化文書の上記一部を複写するための許可についての要求を送送する第1の伝送手段と、上記第2のデータプロセッサにおいて上記要求を受け取り、これに回答して上記ユーザの著作権使用料支払い総額を請求する、上記第2のデータプロセッサにおける第2の受信手段と、上記第2の受信手段に結合され、上記第2のデータプロセッサから上記第1のデータプロセッサに許可メッセージを送送することにより上記構造化文書の上記一部を複写する第2の伝送手段と、上記第1のデータプロセッサにおいて上記許可メッセージを受け取り、これに回答して上記第1のデータプロセッサから上記複写装置に複写制御信号を出力することにより、上記複写装置において上記構造化文書の上記一部を複写する、上記第1のデータプロセッサにおける許可メッセージ受信手段とを具えることを特徴とする構造化文書複製管理装置。

【請求項9】構造化文書の著作権使用料支払い要件に従うように、ネットワークの通信装置における上記構造化文書のページの通信を管理するデータ処理システムにおいて、上記システムは、特殊タグを有する著作権使用料支払いエレメントを含む構造化文書の定様式テキストストリームを上記ネットワークの第1のデータプロセッサに入力する入力手段と、上記入力手段に結合され、上記定様式テキストストリームを探索することにより上記著作権使用料支払いエレメントを識別し、著作権使用料支払いフラグを上記第1のデータ

プロセッサのメモリに格納する探索手段と、上記メモリに結合され、ユーザから通信コマンドを受信することにより、上記第1のデータプロセッサによつて制御される通信装置において上記構造化文書の一部を通信する第1の受信手段と、上記第1の受信手段に結合され、上記通信コマンド及び上記著作権使用料支払いフラグに応答して、上記第1のデータプロセッサから上記ネットワークの第2のデータプロセッサに上記構造化文書の上記一部を通信するための許可についての要求を伝送する第1の伝送手段と、上記第2のデータプロセッサにおいて上記要求を受け取り、これに応答して上記ユーザの著作権使用料支払い総額を請求する、上記第2のデータプロセッサにおける第2の受信手段と、上記第2の受信手段に結合され、上記第2のデータプロセッサから上記第1のデータプロセッサに許可メッセージを伝送することにより上記構造化文書の上記一部を通信する第2の伝送手段と、上記第1のデータプロセッサにおいて上記許可メッセージを受け取り、これに応答して上記第1のデータプロセッサから上記通信装置に通信制御信号を出力することにより、上記通信装置において上記構造化文書の上記一部を通信する、上記第1のデータプロセッサにおける許可メッセージ受信手段とを具えることを特徴とする構造化文書複製管理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は構造化文書複製管理方法、構造化文書複製管理装置及び構造化文書複製管理プログラムに関し、特にデータ処理について、ソフトコピー文書を複製するための著作権使用料支払い要件に従うように、データプロセッサによるソフトコピー文書についての複製動作及び印刷動作を管理する際に適用して好適なものである。

【0002】

【従来の技術】現在のワード処理プログラム及びテキスト編集プログラムは構造化文書アーキテクチャを用いることにより、これらのプログラムと共に準備された文書の表示形態及び印刷形態を一段と制御して一段と柔軟性をもたせる。構造化文書アーキテクチャについては、例えば1989年4月26日出願、米国特許出願第344,332号「アクティブな意図解釈を用いて構造化文書内のエレメントを操作する方法」に記述されている。構造化文書は国際標準化機構の規格8879-1986版に説明されているような標準化された汎用マークアップ言語に従つて作成することができる。この標準化された汎用マークアップ言語に従つてマークアップされたテキストのデータストリームにおいて、そのテキストは開始タグ及びその内容でなるエレメントに分割され、必要なときは終了タグで終了する。WYSIWYG (what you see is what you get) エディタにおいては、テキストの構造がテキストの各エレメントごとに開始タグ及び終了タグによつて定義されるときでさえもテキストはそれが印刷されたときに得られるものと同じようにユーザに表示される。構造化文書がユーザに表示されるときに構造化文書内のエレメントが様式化される。通常用いられるエレメントのカテゴリに分類されるエレメントはパラグラフ、単純リスト、番号付きリスト、黒丸リスト及びリスト項目を含む。

【0003】1990年2月28日出願、米国特許出願第486,461号「注釈を電子出版された資料と関連付ける方法」に説明されている方法によつて、構造化文書アーキテクチャタグを用いてマークアップされた大型文書をソフトコピーブックとして操作することができる。米国特許出願第486,461号は、マークアップされた同じ原始資料からオンラインソフトコピーブックを作成し、これを用いて例えばテキストフォーマッタの下で使用されるマークアップ言語ソース又はワードプロセッサのような印刷された情報を作成する。ブックデータストリームはオンライン情報を格納するように中間フォーマットで形式化され、特にブック表示プログラムが使用し得るように設計される。このデータストリームは構造化文書タグを用いることによつてオンラインソフトコピーブックについての構造上の情報を捕獲して保存する。この方法は使用可能な手法により種々のディスプレイ装置にオンラインソフトコピーブックを表示するのに適している。ブック表示プログラムはブックデータストリームを操作し、このブック表示プログラムによりユーザはオンライン情報と対話することができる。電子ブックは、それが表示画面に表示されることを除けばハードコピーブックのように用いられてもよい。この方法により、読者はオンラインソフトコ

ピーブックがあたかもハードコピーブックであるかのようにこのオンラインソフトコピーブックを取り扱うことができ、ブックデータストリームをその入力として使用する。

【0004】ブックデータストリームはソフトコピーブック構築専用プログラムによつて作成され、このソフトコピーブック構築専用プログラムはワード頻度分析を実行することによつて、マークアップされた原始資料を処理することによりソフトコピーブック内のワードを表す短縮データトークンを形成する。次に、この短縮されトークン化されたテキストは例えばコンパクト光ディスクのような記憶ディスクにブックデータストリームとして記憶される。ブックデータストリームを記憶ディスクから読み取るためには、ソフトコピーブック読取り専用プログラムがこの短縮されトークン化されたテキストをマークアップされた原始資料に再変換し、その後これをディスプレイ装置に表示するか、ASCIIファイルとして他の記憶ディスクに複写するか、ハードコピープリンタにおいて印刷するか又はモデムを介して他のデータプロセッサに伝送する。マークアップされた原始資料をブックデータストリームに変換するプロセスについては、米国特許第 5,099,426号「情報検索用に用いられる相互参照キーワードに対して形態素情報を用いる方法」に記述されている。

【0005】ソフトコピーブック読取り専用プログラムはIBM BookManager (TM) READプログラムにおいて実施され、このIBM BookManager (TM) READプログラムはユーザによるオンラインブックの管理、探索及び検査を援助する。2つの相補的なBookManagerプロダクトが存在し、BookManager BUILD はソフトコピーブック構築専用プログラムであり、汎用マークアップ言語を用いてマークアップされたファイルからオンラインブックを作成する。BookManager READプロダクトはBookManager BUILD によつて作成されたオンラインブックを管理、探索及び表示することができる。BookManager READプログラムプロダクトについては、IBM編、1989年2月、出版番号 SC23-0449-0号「オンラインブックを表示するBookManager (TM) READ」に記述されている。BookManager BUILD プログラムプロダクトについては、IBM編、1989年2月、出版番号 SC23-0450-0号「オンラインブックを作成する BookManager (TM) BUILD」に記述されている。これらの出版物はIBMの支社から入手することができる。

【0006】BookManager BUILDプログラムプロダクト及びBookManager READプログラムプロダクトはオンラインソフトコピーブックを使用し、このオンラインソフトコピーブックは次の2つのIBMの出版物、すなわち1990年9月、出版番号SC34-5012-02号「IBM BookMaster - テキスト・プログラマ・ノートブック - 発行3.0」及び1990年9月、出版番号SC34-5009-03号「IBM BookMaster - ユーザガイド - 発行3.0」に説明されている汎用マークアップ言語 (GML) を用いて様式化される。これらのIBM BookMaster 出版物はIBMの支社から入手することができる。

【0007】ソフトコピーブックはそのソフトコピー形式で頻繁に通信され、格納され、表示され、修正され、ハードコピーがこのようになされることはあまりない。このような電子ブック又は電子文書の著作者はこのように生成された電子ブックが適正な形式で機密保護されて取り扱われることを強く望み、その著作物の原作者の一切の権利が尊重されることを望んでいる。このことは1990年6月29日出願、米国特許出願第 546,334号「特殊化された機能を起動する構造化文書タグ」に説明されている。米国特許出願第 546,334号は、著作者の電子ブック又は電子文書の取扱いについて著作者の意向を強要する手段について記述している。著作者は著作権表示を自身のソフトコピー文書に載せてもよく、この方法は著作権表示を一段と目立たせる。著作者が文書のソフトコピーに載せる、例えば「社外秘」、「複製厳禁」又は「配布厳禁」のような機密保護ラベルは複製を制限又は防止するために実施される。

【0008】米国特許出願第 546,334号を用いることにより、特殊な著作権表示がディスプレイ装置に表示されるようにこの特殊な著作権表示を、表示された文書の最初のページに載せる。著作権表示の文字ストリングに付随する、著作権のための特殊な構造化文書タグを構造化文書の定様式テキストストリームのどこにでも配置することができる。これをグローバルタグと呼ぶ。プリンタがシステムに接続されて構造化文書を印刷するように指令されると、プリンタは著作権表示を含むように表紙を選択的に印刷し、その後構造化文書の残りを印刷する。

【0009】米国特許出願第 546,334号が開示する他の特徴は、「複製厳禁」のような特殊なラベルが表示されるようにこのラベルを文書のページに載せることである。特殊な構

造化文書タグは構造化文書に「複製厳禁」ラベルを伴う。「複製厳禁」ラベルが存在することにより、構造化文書がディスプレイ装置に表示されるときに「複製厳禁」ストリングがすべてのページに選択的に表示される。さらに、システムに接続されたプリンタが構造化文書を印刷するように指令されると、「複製厳禁」ラベルの特殊な構造化文書タグの存在にตอบสนองして印刷動作は中止される。さらに、ディスクドライブ記憶装置がシステムに接続されると、「複製厳禁」ラベルの特殊な構造化文書タグにตอบสนองして構造化文書をディスクに書き込むコマンドが中止される。さらに、通信アダプタがシステムに接続されると、「複製厳禁」ラベルの特殊な構造化文書タグにตอบสนองしてこの構造化文書を伝送するコマンドの機能が中止される。

【0010】もちろん、著作者は生計をたてるために著作を業としている。このような動機は著作者もその出版社も同じであり、著作者及びその出版人が出版物用としてのソフトコピーブックの媒体を選択して来た。必要なことは、読者がソフトコピーブックの一部又はすべてを複製したいときにソフトコピーブックの出版者及び著作者への著作権使用料の支払いを読者に強要する手段である。

【0011】構造化文書アーキテクチャの概念と関連して、以下の用語を本発明の説明に用いる。

【0012】SGMLとは標準化された汎用マークアップ言語のことである。タグでなるマークアップ言語を用いて構造化文書を作成する。定義及び詳細については国際標準化機構の規格 8879-1986版を参照。

【0013】構造化文書とはSGMLに応じた形式の定義に従って作成された文書のことである。

【0014】エレメントとは開始タグ及びその英数字ストリングの内容でなり、必要ときには終了タグを含む、SGMLにより定義されたエンティティのことである。

【0015】空エレメントとはその内容が空であり、テキストも他のエレメントも含まないエレメントのことである。

【0016】現在の文書位置とはディスプレイの現在のページを決定する文書位置のことである。

【0017】通常用いられるエレメントとはパラグラフ、単純リスト、番号付きリスト、黒丸リスト及びリスト項目を含む通常用いられるエレメントのカテゴリに分類されるエレメントのことである。

【0018】タグ名とはすべて同じ規則に従って様式化されたエレメント群に与えられる名前のことである。例えばタグ名にはPARAGRAPH、SIMPLE LIST 及びLIST ITEM がある。

【0019】定様式テキストストリームとは英数字の順序のことであり、この英数字は構造化文書を有するエレメントについて構造化文書タグ及びそれに関連したテキストを含む。

【0020】

【発明が解決しようとする課題】従って本発明の目的は構造化文書の著作権使用料支払い要件に従うように、データ処理システムにおいて構造化文書のソフトコピーテキストを管理する改善された方法を提供することである。

【0021】本発明の他の目的は構造化文書の著作権使用料支払い要件に従うように、構造化文書のページの印刷を管理する改善された方法を提供することである。

【0022】本発明のさらに他の目的は構造化文書の著作権使用料支払い要件に従うように、大容量記憶媒体への構造化文書の書き込みを管理する改善された方法を提供することである。

【0023】本発明のさらに他の目的は構造化文書の著作権使用料支払い要件に従うように、構造化文書のソフトコピーの遠隔通信を管理する改善された方法を提供することである。

【0024】

【課題を解決するための手段】かかる課題を解決するため本発明においては、構造化文書の著作権使用料支払い要件に従うように、ネットワーク49のプリンタ装置44における構造化文書のページの印刷を管理する方法において、特殊タグ360を有する著作権使用料支払いエレメントを含む構造化文書の定様式テキストストリーム25をネットワーク49の第1のデータプロセッサ20に入力するステップと、定様式テキストストリーム25を探索することにより著作権使用料支払いエレメントを識別し、著作権使用料支払いフラ

グを第1のデータプロセッサ20のメモリに格納するステップと、ユーザ10から印刷コマンドを受信することにより、第1のデータプロセッサ20によつて制御されるプリンタ装置44において構造化文書の一部を印刷するステップと、印刷コマンド及び著作権使用料支払いフラグに応答して、第1のデータプロセッサ20からネットワーク49の第2のデータプロセッサ15に構造化文書の一部を印刷するための許可についての要求を送送するステップと、第2のデータプロセッサ15において当該要求を受け取り、これに回答してユーザに著作権使用料支払い総額を請求するステップと、第2のデータプロセッサ15から第1のデータプロセッサ20に許可メッセージを送送することにより構造化文書の一部を印刷するステップと、第1のデータプロセッサ20において許可メッセージを受け取り、これに回答して第1のデータプロセッサ20からプリンタ装置44に印刷制御信号を出力することにより、プリンタ装置44において構造化文書の一部を印刷するステップとを含むようにする。

【0025】また本発明においては、構造化文書の著作権使用料支払い要件に従うように、ネットワーク49の複写装置46における構造化文書のページの複写を管理する方法において、特殊タグ360を有する著作権使用料支払いエレメントを含む構造化文書の定様式テキストストリーム25をネットワーク49の第1のデータプロセッサ20に入力するステップと、定様式テキストストリーム25を探索することにより著作権使用料支払いエレメントを識別し、著作権使用料支払いフラグを第1のデータプロセッサ20のメモリに格納するステップと、ユーザ10から複写コマンドを受信することにより、第1のデータプロセッサ20によつて制御される複写装置46において構造化文書の一部を複写するステップと、複写コマンド及び著作権使用料支払いフラグに応答して、第1のデータプロセッサ20からネットワーク49の第2のデータプロセッサ15に構造化文書の一部を複写するための許可についての要求を送送するステップと、第2のデータプロセッサ15において当該要求を受け取り、これに回答してユーザ10に著作権使用料支払い総額を請求するステップと、第2のデータプロセッサ15から第1のデータプロセッサ20に許可メッセージを送送することにより構造化文書の一部を複写するステップと、第1のデータプロセッサ20において許可メッセージを受け取り、これに回答して第1のデータプロセッサ20から複写装置46に複写制御信号を出力することにより、複写装置46において構造化文書の一部を複写するステップとを含むようにする。

【0026】また本発明においては、構造化文書の著作権使用料支払い要件に従うように、ネットワーク49の通信装置48における構造化文書のページの通信を管理する方法において、特殊タグ360を有する著作権使用料支払いエレメントを含む構造化文書の定様式テキストストリーム25をネットワーク49の第1のデータプロセッサ20に入力するステップと、定様式テキストストリーム25を探索することにより著作権使用料支払いエレメントを識別し、著作権使用料支払いフラグを第1のデータプロセッサ20のメモリに格納するステップと、ユーザ10から通信コマンドを受信することにより、第1のデータプロセッサ20によつて制御される通信装置48において構造化文書の一部を通信するステップと、通信コマンド及び著作権使用料支払いフラグに応答して、第1のデータプロセッサ20からネットワーク49の第2のデータプロセッサ15に構造化文書の一部を通信するための許可についての要求を送送するステップと、第2のデータプロセッサ15において当該要求を受け取り、これに回答してユーザ10に著作権使用料支払い総額を請求するステップと、第2のデータプロセッサ15から第1のデータプロセッサ20に許可メッセージを送送することにより構造化文書の一部を通信するステップと、第1のデータプロセッサ20において許可メッセージを受け取り、これに回答して第1のデータプロセッサ20から通信装置に通信制御信号を出力することにより、通信装置48において構造化文書の一部を通信するステップとを含むようにする。

【0027】また本発明においては、実行されたとき、構造化文書の著作権使用料支払い要件に従うように、ネットワーク49のプリンタ装置44における構造化文書のページの印刷を管理する方法を実行するコンピュータプログラムにおいて、特殊タグ360を有する著作権使用料支払いエレメントを含む構造化文書の定様式テキストストリーム25をネットワーク49の第1のデータプロセッサ20に入力するステップと、定様式テキストストリーム25を探索することにより著作権使用料支払いエレメントを識別し、著作権使用料支払いフラグを第1のデータプロセッサ20のメモリに格納するステップと、ユーザ10から印刷コマンドを受信することにより、第1のデータプロセッサ20によつて制御さ

れるプリンタ装置 44 において構造化文書の一部を印刷するステップと、印刷コマンド及び著作権使用料支払いフラグに応答して、第 1 のデータプロセッサ 20 からネットワーク 49 の第 2 のデータプロセッサ 15 に構造化文書の一部を印刷するための許可についての要求を伝送するステップと、第 2 のデータプロセッサ 15 において当該要求を受け取り、これに応答してユーザに著作権使用料支払い総額を請求するステップと、第 2 のデータプロセッサ 15 から第 1 のデータプロセッサ 20 に許可メッセージを伝送することにより構造化文書の一部を印刷するステップと、第 1 のデータプロセッサ 20 において許可メッセージを受け取り、これに応答して第 1 のデータプロセッサ 20 からプリンタ装置 44 に印刷制御信号を出力することにより、プリンタ装置 44 において構造化文書の一部を印刷するステップとを含むようにする。

【0028】また本発明においては、実行されたとき、構造化文書の著作権使用料支払い要件に従うように、ネットワーク 49 の複写装置 46 における構造化文書のページの複写を管理する方法を実行するコンピュータプログラムにおいて、特殊タグ 360 を有する著作権使用料支払いエレメントを含む構造化文書の定様式テキストストリーム 25 をネットワーク 49 の第 1 のデータプロセッサ 20 に入力するステップと、定様式テキストストリーム 25 を探索することにより著作権使用料支払いエレメントを識別し、著作権使用料支払いフラグを第 1 のデータプロセッサ 20 のメモリに格納するステップと、ユーザ 10 から複写コマンドを受信することにより、第 1 のデータプロセッサ 20 によつて制御される複写装置 46 において構造化文書の一部を複写するステップと、複写コマンド及び著作権使用料支払いフラグに応答して、第 1 のデータプロセッサ 20 からネットワーク 49 の第 2 のデータプロセッサ 15 に構造化文書の一部を複写するための許可についての要求を伝送するステップと、第 2 のデータプロセッサ 15 において当該要求を受け取り、これに応答してユーザ 10 に著作権使用料支払い総額を請求するステップと、第 2 のデータプロセッサ 15 から第 1 のデータプロセッサ 20 に許可メッセージを伝送することにより構造化文書の一部を複写するステップと、第 1 のデータプロセッサ 20 において許可メッセージを受け取り、これに応答して第 1 のデータプロセッサ 20 から複写装置 46 に複写制御信号を出力することにより、複写装置 46 において構造化文書の一部を複写するステップとを含むようにする。

【0029】また本発明においては、実行されたとき、構造化文書の著作権使用料支払い要件に従うように、ネットワーク 49 の通信装置 48 における構造化文書のページの通信を管理する方法を実行するコンピュータプログラムにおいて、特殊タグ 360 を有する著作権使用料支払いエレメントを含む構造化文書の定様式テキストストリーム 25 をネットワーク 49 の第 1 のデータプロセッサ 20 に入力するステップと、定様式テキストストリーム 25 を探索することにより著作権使用料支払いエレメントを識別し、著作権使用料支払いフラグを第 1 のデータプロセッサ 20 のメモリに格納するステップと、ユーザ 10 から通信コマンドを受信することにより、第 1 のデータプロセッサ 20 によつて制御される通信装置 48 において構造化文書の一部を通信するステップと、通信コマンド及び著作権使用料支払いフラグに応答して、第 1 のデータプロセッサ 20 からネットワーク 49 の第 2 のデータプロセッサ 15 に構造化文書の一部を通信するための許可についての要求を伝送するステップと、第 2 のデータプロセッサ 15 において当該要求を受け取り、これに応答してユーザ 10 に著作権使用料支払い総額を請求するステップと、第 2 のデータプロセッサ 15 から第 1 のデータプロセッサ 20 に許可メッセージを伝送することにより構造化文書の一部を通信するステップと、第 1 のデータプロセッサ 20 において許可メッセージを受け取り、これに応答して第 1 のデータプロセッサ 20 から通信装置に通信制御信号を出力することにより、通信装置 48 において構造化文書の一部を通信するステップとを含むようにする。

【0030】また本発明においては、構造化文書の著作権使用料支払い要件に従うように、ネットワーク 49 のプリンタ装置 44 における構造化文書のページの印刷を管理するデータ処理システムにおいて、当該システムは、特殊タグ 360 を有する著作権使用料支払いエレメントを含む構造化文書の定様式テキストストリーム 25 をネットワーク 49 の第 1 のデータプロセッサ 20 に入力する入力手段と、入力手段に結合され、定様式テキストストリーム 25 を探索することにより著作権使用料支払いエレメントを識別し、著作権使用料支払いフラグを第 1 のデータプロセッサ 20 のメモリに格納する探索手段と、メモリに結合され、ユーザ 10 から印刷コマンドを受信することにより、第 1 のデータプロセ

ツサ 20 によつて制御されるプリンタ装置 44 において構造化文書の一部を印刷する第 1 の受信手段と、第 1 の受信手段に結合され、印刷コマンド及び著作権使用料支払いフラグに回答して、第 1 のデータプロセッサ 20 からネットワーク 49 の第 2 のデータプロセッサ 15 に構造化文書の一部を印刷するための許可についての要求を伝送する第 1 の伝送手段と、第 2 のデータプロセッサ 15 において要求を受け取り、これに回答してユーザ 10 に著作権使用料支払い総額を請求する、第 2 のデータプロセッサ 15 における第 2 の受信手段と、第 2 の受信手段に結合され、第 2 のデータプロセッサ 15 から第 1 のデータプロセッサ 20 に許可メッセージを伝送することにより構造化文書の一部を印刷する第 2 の伝送手段と、第 1 のデータプロセッサ 20 において許可メッセージを受け取り、これに回答して第 1 のデータプロセッサ 20 からプリンタ装置 44 に印刷制御信号を出力することにより、プリンタ装置 44 において構造化文書の一部を印刷する、第 1 のデータプロセッサ 20 における許可メッセージ受信手段とを設けるようにする。

【0031】また本発明においては、構造化文書の著作権使用料支払い要件に従うように、ネットワーク 49 の複写装置 46 における構造化文書のページの複写を管理するデータ処理システムにおいて、当該システムは、特殊タグ 360 を有する著作権使用料支払いエレメントを含む構造化文書の定様式テキストストリーム 25 をネットワーク 49 の第 1 のデータプロセッサ 20 に入力する入力手段と、入力手段に結合され、定様式テキストストリーム 25 を探索することにより著作権使用料支払いエレメントを識別し、著作権使用料支払いフラグを第 1 のデータプロセッサ 20 のメモリに格納する探索手段と、メモリに結合され、ユーザ 10 から複写コマンドを受信することにより、第 1 のデータプロセッサ 20 によつて制御される複写装置 46 において構造化文書の一部を複写する第 1 の受信手段と、第 1 の受信手段に結合され、複写コマンド及び著作権使用料支払いフラグに回答して、第 1 のデータプロセッサ 20 からネットワーク 49 の第 2 のデータプロセッサ 15 に構造化文書の一部を複写するための許可についての要求を伝送する第 1 の伝送手段と、第 2 のデータプロセッサ 15 において要求を受け取り、これに回答してユーザ 10 の著作権使用料支払い総額を請求する、第 2 のデータプロセッサ 15 における第 2 の受信手段と、第 2 の受信手段に結合され、第 2 のデータプロセッサ 15 から第 1 のデータプロセッサ 20 に許可メッセージを伝送することにより構造化文書の一部を複写する第 2 の伝送手段と、第 1 のデータプロセッサ 20 において許可メッセージを受け取り、これに回答して第 1 のデータプロセッサ 20 から複写装置 46 に複写制御信号を出力することにより、複写装置 46 において構造化文書の一部を複写する、第 1 のデータプロセッサ 20 における許可メッセージ受信手段とを設けるようにする。

【0032】また本発明においては、構造化文書の著作権使用料支払い要件に従うように、ネットワーク 49 の通信装置 48 における構造化文書のページの通信を管理するデータ処理システムにおいて、当該システムは、特殊タグ 360 を有する著作権使用料支払いエレメントを含む構造化文書の定様式テキストストリーム 25 をネットワーク 49 の第 1 のデータプロセッサ 20 に入力する入力手段と、入力手段に結合され、定様式テキストストリーム 25 を探索することにより著作権使用料支払いエレメントを識別し、著作権使用料支払いフラグを第 1 のデータプロセッサ 20 のメモリに格納する探索手段と、メモリに結合され、ユーザ 10 から通信コマンドを受信することにより、第 1 のデータプロセッサ 20 によつて制御される通信装置 48 において構造化文書の一部を通信する第 1 の受信手段と、第 1 の受信手段に結合され、通信コマンド及び著作権使用料支払いフラグに回答して、第 1 のデータプロセッサ 20 からネットワーク 49 の第 2 のデータプロセッサ 15 に構造化文書の一部を通信するための許可についての要求を伝送する第 1 の伝送手段と、第 2 のデータプロセッサ 15 において要求を受け取り、これに回答してユーザの著作権使用料支払い総額を請求する、第 2 のデータプロセッサ 15 における第 2 の受信手段と、第 2 の受信手段に結合され、第 2 のデータプロセッサ 15 から第 1 のデータプロセッサ 20 に許可メッセージを伝送することにより構造化文書の一部を通信する第 2 の伝送手段と、第 1 のデータプロセッサ 20 において許可メッセージを受け取り、これに回答して第 1 のデータプロセッサ 20 から通信装置 48 に通信制御信号を出力することにより、通信装置 48 において構造化文書の一部を通信する、第 1 のデータプロセッサ 20 における許可メッセージ受信手段とを設けるようにする。

【0033】

【作用】 ソフトコピーブックの著作者又は出版社は、ユーザがソフトコピーブック又はソ

フトコピーブックの個別の章を複製する際に支払うべき著作権使用料の総額を特定する。ソフトコピーブックのソーステキストは構造化文書の定様式テキストストリームの形式を有する。

【0034】本発明によると、出版社はソフトコピーブックの構造化文書テキスト又はソフトコピーブックに付随する著作権使用料支払い情報ファイルのいずれかに著作権使用料支払い情報を含む。次に出版社はコンパクトディスク光記憶媒体、フロッピーディスク磁気記憶媒体又は他の適正な記憶媒体にソフトコピーブックを複製する。ソフトコピーブックは、短縮されトークン化されたテキストとして記憶媒体に格納される。

【0035】ソフトコピーブックを記憶ディスクから読み出すために、ユーザは自身のワークステーションにおいてソフトコピーブック読取り専用プログラムを適用する。このプログラムは短縮されトークン化されたテキストを構造化文書のソーステキストに再変換し、その後構造化文書のソーステキストをディスプレイ装置に表示することができる。

【0036】本発明によると、ユーザはまたソフトコピーブック読取りプログラムと共に著作権使用料支払いプログラムを適用する。ユーザが例えば磁気ディスクのような書込み可能記憶媒体にソフトコピーブックを複製するコマンドを入力するか、プリンタを用いてソフトコピーブックのハードコピーを印刷するコマンドを入力するか又はモデムを介してソフトコピーブックの複製物を伝送するコマンドを入力すると、著作権使用料支払いプログラムがこれらの複製コマンドをインターセプトしてこれらの複製動作を中断させる。その代わり著作権使用料支払いプログラムはソフトコピーブックに付随するファイル又はブックテキストに格納された著作権使用料支払い情報をユーザに提示する。ユーザは著作権使用料支払いプログラムがソフトコピーブックの複製を許可する前に出版社に著作権使用料を支払うかのオプションを選択しなければならない。

【0037】ソフトコピーブックの構造化文書テキスト又はソフトコピーブックに付随する著作権使用料情報ファイルに含まれる著作権使用料支払い情報は、ソフトコピーブック全体に関する著作権使用料支払い情報と共にソフトコピーブックの個別の章に関する著作権使用料支払い情報を含むことができる。ユーザが複製コマンドを入力するとき、ユーザが自身のワークステーションにソフトコピーブックの特定の章を表示している場合、著作権使用料支払いプログラムはソフトコピーブック全体についての著作権使用料支払い情報と共に、現在表示されている章についての著作権使用料支払い情報を表示する。

【0038】ユーザが著作権使用料を支払うことを選択すると、例えばモデムを介して公衆電話網に呼を配置することによって、ユーザのワークステーション及び出版社のデータプロセッサ間に通信セッションが確立される。著作権使用料支払いプログラムは出版社のデータプロセッサにメッセージを送り、ユーザがソフトコピーブックの特定部分又はソフトコピーブック全体を複製し得るように要求する。この要求メッセージはユーザの名前、ユーザのメールアドレス及びユーザのクレジットカード番号又は他の請求情報を含む。またこの要求メッセージは、ユーザが支払うべき著作権使用料の総額を出版社のデータプロセッサが計算し得るようにソフトコピーブックについての十分な情報を含む。

【0039】ユーザが申し出た支払い方式が有効であることをチェックした後、出版社のデータプロセッサはユーザの著作権使用料支払いプログラムに、ユーザが申し込んだソフトコピーブックの複製を許可する許可メッセージを送り返す。次にユーザのワークステーションの著作権使用料支払いプログラムが、中断されていた複製動作を推奨し、これによりユーザが申し込んだソフトコピーブックを複製することができる。

【0040】出版社のデータプロセッサはユーザによる要求を記録してソフトコピーブックを複製し、ユーザのクレジットカードの口座に対する請求書を作成する。

【0041】ユーザの要求に回答して出版社のデータプロセッサが送った許可メッセージは認証データを含むことができ、ユーザのワークステーションの著作権使用料支払いプログラムがこの認証データを用いることにより出版社からの許可メッセージを認証する。これによりユーザ又は他の者が著作権使用料支払いプロセスを免れることを防止する。

【0042】本発明によると、認証メッセージは開始ページデータをも含み、ユーザはこの開始ページデータを用いて、中止されていた複製セッションを回復させる。ユーザが申し込んだソフトコピーブックを複製すると、ユーザのワークステーションから出版社のデータプロセッサに周期的に確認信号が送られる。これはユーザが上手く複製した各ページ又は各章ごとになされる。出版社のデータプロセッサは確認信号の数を数える。ユーザが申し込んだソフトコピーブックの複製が完了する前に複製が失敗すると、出版社のデータ

プロセッサはソフトコピーブックのうち上手く複製された最後のページ又は最後の章の番号を用いて障害記録を記憶する。これは出版社が受け取った確認信号の数によって示される。その後ユーザは自身のワークステーションを出版社のデータプロセッサと再接続してソフトコピーブックを複製するように新たに要求すると、上手く複製された最後のページ又は最後の章を一致させた状態で、出版社のデータプロセッサはユーザに新しい許可メッセージを送る。ユーザのワークステーションが出版社のデータプロセッサに再接続されないと、出版社のデータプロセッサは上手く複製されたソフトコピーブックの一部だけについてユーザに請求する。

【0043】従つて本発明は、読者がソフトコピーブックの一部又はすべてを複製したいとき、ソフトコピーブックの出版社及び著作者への著作権使用料の支払いを読者に強要する改善された手段を提供する。

【0044】

【実施例】以下図面について、本発明の一実施例を詳述する。

【0045】図1は著作権使用料支払いプログラム45を含む、本発明によるユーザのワークステーションのシステムブロック図である。ワークステーション10はバス29によつてメモリ22に接続されるプロセッサ20を含む。プロセッサ20はメモリ22に格納されたプログラムについてのプログラム命令を実行する。またバス29にはディスプレイ26、キーボード50、プリンタ44、磁気ディスクドライブ46、コンパクト光ディスクドライブ55及びネットワークアダプタ48も接続される。ネットワークアダプタ48は通信ネットワーク49に接続され、通信ネットワーク49は公衆交換電話網である場合がある。

【0046】メモリ22はユーザプロファイル23、構造化文書定様式テキストストリーム25（図6）、ロード済みパラメータテーブル56L（図9）、ディスプレイバッファ26'（図7）、BookManager ソフトコピー読取りプログラム35、著作権使用料支払いプログラム45（図10）、通信アプリケーションプログラム47及びオペレーティングシステム27を格納する。

【0047】図2に示すようにユーザのワークステーション10は通信ネットワーク49を介して出版社のデータプロセッサ15に接続される。出版社のデータプロセッサ15はネットワーク49及びバス98に接続されるネットワークアダプタ92を含む。またバス98にはプロセッサ94及びメモリ96も接続される。プロセッサ94はメモリ96に格納されたプログラムについてのプログラム命令を実行する。

【0048】メモリ96は第1のブックデータ70を格納し、第1のブックデータ70は第1のブックについての総ページ数、第1章のページ数及び第2章のページ数を含む。またメモリ96は第2のブックデータ72を格納し、第2のブックデータ72は第2のブックについての総ページ数、第1章のページ数、第2章のページ数及び第3章のページ数を含む。またメモリ96は著作権使用料請求プログラム74（図12）、出版社のプライベート鍵76、公開鍵暗号化プログラム78、通信アプリケーションプログラム79及びオペレーティングシステム80を格納する。

【0049】ソフトコピーブックの著作者又は出版社は、ユーザがソフトコピーブック又はソフトコピーブックの個別の章を複製する際に支払うべき著作権使用料の総額を特定する。ソフトコピーブックのソーステキストは構造化文書の定様式テキストストリーム25の形式を有する。本発明によると、出版社はソフトコピーブックの構造化文書テキスト又はソフトコピーブックに付随する著作権使用料支払い情報ファイルのいずれかに著作権使用料支払い情報を含む。著作権使用料支払い情報を含む、構造化文書の定様式テキストストリームについて2つの例を示し、その第1の例を図3、図4及び図5に示す。

【0050】プロセッサ20に結合されたディスプレイ装置26は例えばパラグラフ28及び38並びに図5のディスプレイ装置26に示すリスト項目32、34及び36を含むリスト30のような複数の文書エレメントを WYSIWYG形式で表示する。またカーソル54も示されている。文書エレメントは番号を付されたリスト又は番号を付されていないリストのいずれかを含むことができ、またグラフィックス及びテキストを含むこともできる。特に注意すべきは、図5のディスプレイ装置26に示す構造化文書のページについて、著作権表示40及び「ブック複製料金」著作権使用料メッセージ306がWYSIWYG形式で表示されていることである。文書エレメント28～40は図3に見ることができるよう構造化文書表記法を用いて、順序付けられたシーケンスを有する定様式データストリームに

編成される。この順序付けられたシーケンスは複数のエレメントタグの対応する順序付けられたシーケンスによつて特定化される。図3に示すパラグラフ28は開始タグ〔p〕及び終了タグ〔／p〕を用いて構造化文書表記法で示される。例えば図3に示すパラグラフ28のような各エレメントは開始タグ〔p〕、この例においてはパラグラフのテキスト及び終了タグ〔／p〕を有する構造化文書エレメントである。開始タグ及び終了タグはエレメントのタイプを識別する際に役に立ち、この例ではパラグラフを示す。パラグラフ28がディスプレイ26に WYSIWYG形式で表示される発生順序は、例えばリスト30及びパラグラフ38のような他の構造化文書エレメントの発生順序の前後関係の範囲内において、図3に示すそのパラグラフ28を表している構造化文書エレメントの定様式テキストストリームにおける発生順序によつて決定される。

【0051】本発明によると、著作権使用料メッセージ306のような特殊なストリングは構造化文書内に、特殊な構造化文書タグを有する特殊なエレメントに組み込まれる。特に、著作権使用料メッセージエレメント306は開始タグ〔royalty〕及び終了タグ〔／royalty〕間に組み込まれたストリングである。プロセッサ20は著作権使用料開始タグのような特殊タグの存在を検出すると、メモリ22にその存在を特別に書き留めることにより、特定プリンタ44、ディスクドライブ記憶装置46又は通信アダプタ48が動作するように要求されたときに複製動作を禁止させ、著作権使用料支払いプロセスを呼び出す。例えば著作権使用料エレメント306が文書テキスト25内で識別されると、著作権使用料エレメントが識別された構造化文書を印刷するようにユーザが要求したときにはこのプリンタ44は使えない。

【0052】例えばパラグラフ28、リスト30及びパラグラフ38のような構造化文書エレメントの発生順序は図4に示すようにメモリ22に格納された定様式テキストストリームにおけるその発生順序によつて決定し得る。図4はタグをもつ図3の構造化文書テキストが文字列型の順序オーダでメモリ22に格納され、構造化文書テキストは定様式テキストストリーム25であることを示す。パラグラフエレメント28はタグ28A及び28B並びにテキスト28Cを含む。リストエレメント30はタグ30A及び30Bを含む。リスト項目エレメント32はタグ32A及び32B並びにテキスト32Cを含む。リスト項目エレメント34はタグ34A及び34B並びにテキスト34Cを含む。リスト項目エレメント36はタグ36A及び36B並びにテキスト36Cを含む。リスト項目エレメント38はタグ38A及び38B並びにテキスト38Cを含む。

【0053】特殊な著作権表示エレメントは著作権ストリング40Cを囲む開始タグ40A及び終了タグ40Bによつて表される。同様に、特殊な著作権使用料メッセージである「ブック複製料金」エレメントは「ブック複製料金」ストリング306Cを囲む開始タグ306A及び終了タグ306Bによつて表される。本発明によると、プロセッサ20がメモリ22に格納された定様式テキストストリーム25全体を探索して著作権使用料メッセージストリング306Cの開始タグ306A又は終了タグ306Bが存在することを検出すると、プロセッサ20は著作権使用料メッセージストリング306Cを図8に示すメモリ22内のパラメータテーブル56の特定区画306Iにロードし、図9に示すようにフラグ366又は368をセットすることにより著作権使用料メッセージストリングの存在を示す。

【0054】さらに本発明によると、著作権使用料支払いプログラム45を図1に示すデータ処理システムに初期プログラムローディングするとき、デフォルトパラメータテーブル56は事前に特定した区画においてメモリにロードされる。図8に示すデフォルトパラメータテーブル56はその中に入力された一組のデフォルト値を有し、これはメモリ22に記憶される。またユーザは値のプロファイルのパラメータテーブル56に入力することができる。初期プログラムローディングでロードされると、パラメータテーブル56は図8及び図9に示すように著作権使用料タグ「〔royalty〕」等の特殊タグをそれぞれ表す文字を含む。また図8のパラメータテーブル56には、タグを付された特定のストリングがメモリ22にロードされた文書テキスト内で識別されたときにデータ処理システムが応答したことを示す値を含む。例えば著作権使用料タグ「〔royalty〕」が構造化文書内で検出されると、この構造化文書をディスプレイ26に表示すべきとき、表示されたページが著作権使用料メッセージを含むということをデフォルトパラメータテーブル56が示す。

【0055】図6は定様式テキストストリーム25の第2の例についてのメモリーメージ

を一段と詳細に示した図であり、各特定のパラグラフと関連したパラグラフ座標 1 5 6、見出し及び他のエレメントを定様式テキストストリーム 2 5 に特に示す。図 6 に示す定様式テキストストリーム 2 5 は参照した上述のタグと同様の構造化文書タグを有する。図 6 の構造化文書は図 7 のディスプレイ 2 6 に WYSIWYG 形式で表示される。図 6 の構造化文書 2 5 は 3 つの主要な部分、すなわちエレメント 3 0 0 ~ 3 1 4 となるブックタイトル部分、エレメント 3 1 6 ~ 3 4 0 となる第 1 章の部分及びエレメント 3 4 2 ~ 3 5 0 となる第 2 章の部分を含む。

【0056】図 6 の第 2 の例の構造化文書 2 5 についてのブックタイトル部分は以下のエレメント、すなわち「[bk] ブックタイトル[/bk]」3 0 0、「[ed] 第 2 版[ed]」3 0 2、「[cpr] (C) ABC CO 1990[/cpr]」3 0 4、「[royalty] ブック複製料金[/royalty]」3 0 6、「[amount] \$20.00[/amount]」3 0 8、「[phone] 1-800-123-1234[/phone]」3 1 0、「[public key] 13A723F9...6[/public key]」3 1 2 及び「[validation] ブック複製料金が支払われる[/validation]」3 1 4 を有する。

【0057】著作権使用料メッセージ 3 0 6 は特殊タグ [royalty] の後に著作権使用料メッセージを表す特殊なストリング「ブック複製料金」と終了タグ[/royalty] とを続けることによって表される。図 9 のデフォルトテーブル 5 6 のカラム 3 6 4 は、このストリングがディスプレイ 2 6 に表示されることを示す。著作権使用料メッセージ 3 0 6 のような著作権使用料エレメントが構造化文書 2 5 に最初に発生し、[h1] のような章の見出しに発生しないとき、著作権使用料エレメントはソフトコピーブック全体に影響を与えるグローバルエレメントと考えられる。これは図 9 のデフォルトテーブル 5 6 のブック著作権使用料フラグのカラム 3 6 6 に伝えられる。著作権使用料エレメント 3 0 6 は構造化文書 2 5 に最初に発生する著作権使用料エレメントであるので、著作権使用料エレメント 3 0 6 はソフトコピーブック全体を複製するための著作権使用料メッセージを表す。また著作権使用料メッセージ 3 0 6 に続いて即座に発生する特殊なエレメント内の関連した著作権使用料情報もソフトコピーブック全体を複製すべきときに適用されるグローバルエレメントと考えられる。著作権使用料エレメント 3 0 6 に関連するエレメントは「[amount] \$20.00[/amount]」3 0 8、「[phone] 1-800-123-1234[/phone]」3 1 0、「[public key] 13A723F9...6[/public key]」3 1 2 及び「[validation] ブック複製料金が支払われる[/validation]」3 1 4 である。

【0058】エレメント「[amount] \$20.00[/amount]」3 0 8 はソフトコピーブック全体を複製する際の費用を表す。図 9 のデフォルトテーブル 5 6 のカラム 3 6 4 はこのストリングがディスプレイ 2 6 に表示されることを示す。

【0059】エレメント「[phone] 1-800-123-1234[/phone]」3 1 0 は出版社のデータプロセッサ 1 5 の電話番号を表し、この電話番号は電話網 4 9 を介して呼び出される。図 9 のデフォルトテーブル 5 6 のカラム 3 6 4 はこのストリングがディスプレイ 2 6 に表示されることを示す。

【0060】エレメント「[public key] 13A723F9...6[/public key]」3 1 2 は出版社の公開鍵であり、ユーザの著作権使用料支払プログラム 4 5 はこの出版者の公開鍵を用いて出版社から受け取った許可メッセージが正当であることを検証する。図 9 のデフォルトテーブル 5 6 のカラム 3 6 4 はこのストリングがディスプレイ 2 6 に表示されないことを示す。

【0061】エレメント「[validation] ブック複製料金が支払われる[/validation]」3 1 4 は、出版社の許可メッセージが正当であることを保証するためにユーザの著作権使用料支払いプログラム 4 5 を出版社の許可メッセージと比較するストリングである。図 9 のデフォルトテーブル 5 6 のカラム 3 6 4 はこのストリングがディスプレイに表示されないことを示す。

【0062】図 6 の第 2 の例の構造化文書 2 5 における第 1 章の部分は以下のエレメント、すなわち「[h1] 第 1 章の見出し[/h1]」3 1 6、「[royalty] 章複製料金[/royalty]」3 1 8、「[amount] \$1.00[/amount]」3 2 0、「[validation] 第 1 章の料金が支払われる[/validation]」3 2 2、「[p] テキストのパラグラフ[/p]」3 2 4、「[h2] 第 1 のトピック見出し[/h2]」3 2 6、「[p] テキストのパラグラフ[/p]」3 2 8、「[p] テキストのパラグラフ[/p]」3 3 0、「[h2] 第 2 のトピック見出し[/h2]」3 3 2、「[p] テキストのパラグラフ[/p]」3 3

4、「[h3] 第1のサブトピック見出し[/h3]」336、「[p] テキストのパラグラフ[/p]」338及び「[p] テキストのパラグラフ[/p]」340を有する。

【0063】第1章の見出しについての構造化文書タグは開始タグ[h1]及び終了タグ[/h1]である。章の見出しの下には4つのエレメントがある。第1のエレメントは特殊なタグ[royalty]の後に著作権使用料メッセージを表す特殊なストリング「章複製料金」と終了タグ[/royalty]とを続けることによつて表された著作権使用料メッセージ318である。図9のデフォルトテーブル56のカラム364はこのストリングがディスプレイ26に表示されることを示す。著作権使用料エレメントが例えば[h1]のような章見出しに発生すると、この著作権使用料エレメントは章h1だけに影響を与えるローカルエレメントと考えられる。著作権使用料エレメントは図9のデフォルトテーブル56の章著作権使用料フラグのカラム368に伝えられる。著作権使用料エレメント318は著作権使用料エレメント306の後に構造化文書25に発生する第2番目の著作権使用料エレメントであるので、この著作権使用料エレメントはブック著作権使用料エレメント306に代用されない。

【0064】また著作権使用料メッセージ318に続いて即座に発生する、特殊なエレメント内の関連する著作権使用料情報も、第1章を複製すべきときにこの第1章だけに適用されるローカルエレメントと考えられる。著作権使用料エレメント318に関連するエレメントは「[amount] 1.00\$[/amount]」320及び「[validation] 第1章の料金が支払われる[/validation]」322である。

【0065】第2のエレメントは特殊なタグ[amount]の後に著作権使用料総額を表す特殊なストリング「1.00\$」と終了タグ[/amount]とを続けることによつて表される。図9のデフォルトテーブル56のカラム364はこのストリングがディスプレイ26に表示されることを示す。

【0066】第3のエレメントは特殊なタグ[validation]の後に検証を表す特殊なストリング「第1章の料金が支払われる」と終了タグ[/validation]とを続けることによつて表される。図9のデフォルトテーブル56のカラム364はこのストリングがディスプレイ26に表示されないことを示す。

【0067】章見出しの下にある第4のエレメントはパラグラフ開始タグ[p]及び終了タグ[/p]によつて表示されたパラグラフエレメント324である。

【0068】定様式テキストストリーム25は階層編成によつて特徴付けられ、その開始タグ[h2]及び終了タグ[/h2]によつて示された第1のトピック見出しエレメント326はこの階層内の章見出しエレメント316に従属するという点に特徴がある。定様式テキストストリーム25についてこの階層構造を用いることにより、座標システム156を確立する。

【0069】図6に示す座標システム156はこれがメモリ22に格納し得るように示され、座標システム156に結合した特定の見出しすなわち特定のエレメントエントリの階層レベルを表す6つの数字をもっている。例えばパラグラフエレメント330はbk=「1」、h1=「1」、h2=「1」、h3=「0」、p=「2」及びel=「1」の座標を有する。これは、エレメント330が第1のブックbk=「1」の一部、第1のh1章見出しの一部、第1のh2トピック見出しの一部であるが、第1のトピック見出しに従属するトピック見出しの一部ではないということを意味する。p=「2」座標は、エレメント330が第1のトピック見出しの下にある第2番目のパラグラフエレメントであることを表す。el=「1」座標はエレメント330がパラグラフ内の第1のエレメントであることを表す。

【0070】図6の第2の例の構造化文書25における第2章の部分は以下のエレメント、すなわち「[h1] 第2章の見出し[/h1]」342、「[royalty] 章複製料金[/royalty]」344、「[amount] 2.00\$[/amount]」346、「[validation] 第2章の料金が支払われる[/validation]」348及び「[p] テキストのパラグラフ[/p]」350を有する。

【0071】第2章の見出し342の構造化文書のタグは開始タグ[h1]及び終了タグ[/h1]である。第2章の見出し342の下には4つのエレメントがある。第1のエレメントは特殊なタグ[royalty]の後に著作権使用料メッセージを表す特殊なストリング「章複製料金」と終了タグ[/royalty]とを続けることによつて表される著作権使用料メッセージ344である。図9のデフォルトテーブル56のカラム364はこのストリングがディスプレイ26に表示されることを示す。著作権使用料エレメントが第2章の見出

し342のような章の見出しに発生すると、この著作権使用料エレメントは第2章の見出し342だけに影響を与えるローカルエレメントと考えられる。この著作権使用料エレメントは図9のデフォルトテーブル56の章著作権使用料フラグのカラム368に伝えられる。著作権使用料エレメント344は著作権使用料エレメント306及び318の後に構造化文書25に発生する第3番目の著作権使用料エレメントであるので、この著作権使用料エレメント344はブック著作権使用料エレメント306に代用されない。

【0072】また著作権使用料メッセージ344に続いて即座に発生する、特殊なエレメント内の関連する著作権使用料情報は第2章を複製すべきときにこの第2章だけに適用されるローカルエレメントと考えられる。著作権使用料エレメント344に関連するエレメントは「[amount] \$2.00 [/amount]」346及び「[validation] 第2章の料金が支払われる [/validation]」348である。

【0073】第2のエレメントは特殊なタグ[amount]の後に著作権使用料総額を表す特殊なストリング「\$2.00」と終了タグ「[/amount]」とを続けることによつて表される。図9のデフォルトテーブル56のカラム364はこのストリングがディスプレイ26に表示されることを示す。

【0074】第3のエレメントは特殊なタグ[validation]の後に検証を表す特殊なストリング「第2章の料金が支払われる」と終了タグ「[/validation]」とを続けることによつて表された検証データ348である。図9のデフォルトテーブル56のカラム364はこのストリングがディスプレイ26に表示されないことを示す。

【0075】第2章の見出しの下にある第4のエレメントはパラグラフ開始タグ[p]及び終了タグ[/p]によつて表示されたパラグラフエレメント350である。

【0076】ソフトコピーブックの著作者又は出版社はソフトコピーブック又はソフトコピーブックの個々の章を複製するユーザが支払うべき著作権使用料の総額を特定する。ソフトコピーブックのソーステキストは構造化文書の定様式テキストストリーム25の形式を有する。本発明によると、出版社はソフトコピーブックの構造化文書テキスト又はソフトコピーブックに付随する著作権使用料支払い情報ファイルのいずれかに著作権使用料支払い情報を含む。

【0077】本発明によると、出版社は図6に示すようなソフトコピーブックの構造化文書テキスト又は図13に示すようにソフトコピーブックに付随する著作権使用料支払い情報ファイル58のいずれかに著作権使用料支払い情報を含む。次に出版社はコンパクトディスク光記憶媒体、フロッピーディスク磁気記憶媒体又は他の適正な記憶媒体55'にソフトコピーブックを複製する。このソフトコピーブックは短縮されトークン化されたテキストとしてこれらの記憶媒体に格納される。

【0078】ソフトコピーブックを記憶ディスクから読み出すためには、ユーザは自身のワークステーション10においてソフトコピーブック読取り専用プログラム35を適用する。このプログラム35は短縮されトークン化されたテキストを構造化文書のソーステキストに再変換し、その後図7に示すように構造化文書のソーステキストをディスプレイ装置26に表示することができる。

【0079】本発明によると、ユーザはまたソフトコピーブック読取りプログラム35に従つて図10の著作権使用料支払いプログラム45を適用する。ユーザがディスクドライブ46上の磁気ディスクのような書込み可能記憶媒体にソフトコピーブックを複製するコマンドを入力するか、プリンタ44を用いてソフトコピーブックのハードコピーを印刷するコマンドを入力するか又はモデム及びネットワークアダプタ48を介してソフトコピーブックの複製物を伝送するコマンドを入力すると、著作権使用料支払いプログラム45はこれらの複製コマンドをインターセプトしてこれらの複製動作を中断させる。

【0080】その代わり著作権使用料支払いプログラム45は図6のソフトコピーブックテキスト又はソフトコピーブックに付随する著作権使用料支払い情報ファイル58に格納された著作権使用料支払い情報の表示画面をユーザに提示する。ユーザは著作権使用料支払いプログラムがソフトコピーブックの複製を許可する前に出版社に著作権使用料を支払うかのオプションを選択しなければならない。

【0081】図6に示すソフトコピーブックの構造化文書テキスト25又はソフトコピーブックに付随する著作権使用料情報ファイル58に含まれる著作権使用料支払い情報はソフトコピーブックの個々の章に関する著作権使用料支払い情報及びソフトコピーブック全体に関する著作権使用料支払い情報を含むことができる。ユーザが複製コマンドを入力す

るとき自身のワークステーションにソフトコピーブックの特定の章を表示している場合、図 1 4に示すように著作権使用料支払いプログラム 4 5 は現在表示されている章についての著作権使用料支払い情報及びソフトコピーブック全体についての著作権使用料支払い情報を表示する。

【0082】ユーザが著作権使用料を支払うことを選択すると、例えばモデム及びネットワークアダプタ 4 8 を介して公衆電話網 4 9 に呼を配置することによつてユーザのワークステーション 1 0 及び出版社のデータプロセッサ 1 5 間に通信セッションが確立される。著作権使用料支払いプログラム 4 5 は出版社のデータプロセッサ 1 5 にメッセージを送り、ユーザがソフトコピーブックの特定の部分又はソフトコピーブック全体を複製し得るように要求する。この要求メッセージはユーザの名前、ユーザのメールアドレス及びユーザのクレジットカード番号又は他の請求情報を含む。またこの要求メッセージは、ユーザが支払う著作権使用料の総額を出版社のデータプロセッサが計算し得るようにソフトコピーブックについての十分な情報を含む。

【0083】出版社のデータプロセッサ 1 5 は図 1 2に示す著作権使用料請求プログラム 7 4 を有し、このプログラム 7 4 はユーザの要求メッセージを受信して処理する。ユーザが申し出た支払い形式が有効であることをチェックした後、出版社のデータプロセッサ 1 5 の著作権使用料請求プログラム 7 4 はユーザの著作権使用料支払いプログラム 4 5 に、ユーザが申し込んだソフトコピーブックの複製を許可する許可メッセージを送り返す。その後ユーザのワークステーション 1 0 の著作権使用料支払いプログラム 4 5 は、中断されていた複製動作を推奨し、これにより申し込んだソフトコピーブックを複製することができる。

【0084】出版社のデータプロセッサ 1 5 の著作権使用料請求プログラム 7 4 はユーザによる要求を記録することによりソフトコピーブックを複製し、ユーザのクレジットカードの口座に対する請求書を作成する。

【0085】ユーザの要求に応答して出版社のデータプロセッサ 1 5 の著作権使用料請求プログラム 7 4 が送った許可メッセージは認証データを含むことができ、ユーザのワークステーション 1 0 の著作権使用料支払いプログラム 4 5 がこの認証データを用いて出版社からの許可メッセージを認証する。これによりユーザ又は他の者が著作権使用料支払いプロセスから免れることを防止する。

【0086】本発明によると、許可メッセージはまたユーザが中止した複製セッションを回復させる際に用いる開始ページデータを含んでもよい。ユーザが申し込んだソフトコピーブックを複製すると、図 1 0に示すようにユーザのワークステーション 1 0 から出版社のデータプロセッサ 1 5 に周期的に応答信号が送られる。これはユーザが上手く複製した各ページ又は各章についてなされる。出版社のデータプロセッサ 1 5 の著作権使用料請求プログラム 7 4 は図 1 2に示すように応答信号の数を数える。ユーザが申し込んだソフトコピーブックの複製が完了する前に複製が失敗すると、出版社のデータプロセッサ 1 5 の著作権使用料請求プログラム 7 4 はソフトコピーブックのうち上手く複製された最後のページ又は最後の章の番号を用いて障害レコードを記憶する。これは出版社の著作権使用料請求プログラム 7 4 が受け取った応答信号の数によつて示される。その後、ユーザは自身のワークステーション 1 0 を出版社のデータプロセッサ 1 5 と再接続し、ソフトコピーブックを複製するように新たに要求すると、上手く複製された最後のページ又は最後の章を一致させた状態で、出版社の著作権使用料請求プログラム 7 4 はユーザの著作権使用料支払いプログラム 4 5 に新しい許可メッセージを送る。ユーザが自身のワークステーション 1 0 を出版社のデータプロセッサ 1 5 に再接続しないと、出版社のデータプロセッサ 1 5 の著作権使用料請求プログラム 7 4 はソフトコピーブックのうち上手く複製された部分だけについてユーザに請求する。

【0087】上述したように、ユーザの要求に応答して出版社のデータプロセッサ 1 5 の著作権使用料請求プログラム 7 4 が送った許可メッセージは認証データを含むことができ、ユーザのワークステーション 1 0 の著作権使用料支払いプログラム 4 5 はこの認証データを用いることにより出版社から受信した許可メッセージを認証する。この認証データは出版社の著作権使用料請求プログラム 7 4 が送り返すデータワードと同じぐらい単純であり、このデータワードはワークステーション 1 0 においてユーザの著作権使用料支払いプログラム 4 5 が保存する同様のデータワードと比較される。また例えば出版社の著作権使用料請求プログラム 7 4 がデジタル署名を許可メッセージに適用するような一段と高

性能の認証技術を用いることができ、ユーザの著作権使用料支払いプログラム45はこの許可メッセージを認証することができる。デジタル署名を適用してメッセージを認証する標準的な技術は公開鍵暗号アルゴリズムを用いる技術である。

【0088】公開鍵暗号アルゴリズムについては、1979年3月、IEEE会報、第67巻「プライバシー及び認証：暗号技術入門」397頁～427頁に説明されている。公開鍵システムは、秘密鍵配送チャネルが十分なレベルの保安全性をもっているかぎり、このチャネルをもつ必要はない。公開鍵暗号系においては2つの鍵が用いられ、1つは暗号化のための鍵であり、他の1つは暗号解読のための鍵である。公開鍵アルゴリズムシステムは、暗号化のためのPU鍵及び暗号解読のためのPR鍵という一対のランダムな逆方向鍵を生成することが容易で、かつPU鍵及びPR鍵で動作させることが容易であるように設計されるが、PU鍵からPR鍵を起算するのはコンピュータではできない。通信の各当事者は一対の逆方向鍵、すなわちPU鍵及びPR鍵を生成する。通信の当事者は暗号解読変換PR機密鍵を保持し、これを公開ディレクトリに配置することによつて暗号化変換PU公開鍵を作成する。メッセージを暗号化してこれらを通信当事者に送ることができるが、自分へのメッセージを解読することはできない。PU鍵を用いて暗号化し、PR鍵を用いて解読し得ることが望ましい。このような理由から通常PUを公開鍵と呼び、PRをプライベート鍵と呼ぶ。

【0089】公開鍵暗号系の特徴はメッセージの送り手を独自に識別するデジタル署名を提供することである。当事者Aは符号付きメッセージMを当事者Bに送りたいとき、当事者Aは自分のプライベート鍵PRを用いてこのメッセージを操作することにより符号付きメッセージSを作成する。秘密にしたいときはPR鍵を当事者Aの暗号解読鍵として用いたが、現在はPR鍵を自分の「暗号化」鍵として用いることによりデジタル署名を送る。当事者BはメッセージSを受信すると、当事者Aの公開PU鍵を用いて暗号テキストSを操作することによつてメッセージMを回復させることができる。当事者Aのメッセージを上手く解読することによつて、受信者Bはこのメッセージが送り手Aから来たことが分かる。符号を付すべきデータをデータが短いときに良く動作するプライベート鍵を用いて暗号化するか又はまず符号を付すべきデータを強力な一方向性暗号関数を用いてハッシュし、このように作成されハッシュされた値をプライベート鍵を用いて暗号化するかのいずれかによつてデジタル署名を作成することができる。いずれの方法でも作成し得る。

【0090】符号を付すべきデータのハッシュに基づいてデジタル署名を作成する方法については、1991年8月22日出願、米国特許出願第07/748,407号「制御ベクトルに基づいたハイブリッド公開鍵アルゴリズム／データ暗号化アルゴリズム鍵配送方法」に教示されている。

【0091】公開鍵暗号装置の例としては、米国特許第4,218,582号「公開鍵暗号装置及び方法」、米国特許第4,200,770号「暗号化装置及び方法」並びに米国特許第4,405,829号「暗号通信システム及び方法」に開示されている。

【0092】公開鍵暗号化技術は出版社の著作権使用料請求プログラム74からユーザの著作権使用料支払いプログラム45に送られた許可メッセージに出版社のデジタル署名を付加する際に使用される。上述のように出版社は公開鍵及びそれに対応するプライベート鍵76を最初に作成する。出版社の公開鍵は公開鍵エレメント312として図6のソフトコピーブックの定様式テキストストリーム25に挿入される。次に出版社はCD-ROM媒体形式でこのソフトコピーブックを出版し、ユーザはこのCD-ROMを購入する。他の実施例においては出版社の公開鍵は図13のCD-ROM55'の著作権使用料情報ファイル58に含まれる。

【0093】ユーザがソフトコピーブックを複製するコマンドを入力すると、ユーザの著作権使用料支払いプログラム45はソフトコピーブックの複製要求と共に、検証ストリング314の複写物を出版社に送る。この要求が第1章というような1つの章を複製する要求であるとき、その章についての検証ストリング322がこの要求と共に出版社に送られる。次に出版社の著作権使用料請求プログラム74は公開鍵暗号化プログラム78を用いて出版社のプライベート鍵76で暗号化することによつて、このプログラム74がユーザから受け取った検証ストリングを操作する。その後出版社の著作権使用料請求プログラム74は暗号化された検証ストリングを許可メッセージの一部としてユーザに送り返す。

【0094】次にユーザの著作権使用料支払いプログラム45は、ソフトコピーブックに載っている公開鍵エレメント312から得られた出版社の公開鍵を用いて、プログラム

45が出版社から受け取った暗号化された検証ストリングを復号化することによってこの暗号化された検証ストリングを操作する。ユーザの著作権使用料支払いプログラム45は上述した公開鍵暗号アルゴリズムを含むメッセージ認証プログラム43を用いることにより、復号化を実行する。次にこの復号化された検証ストリングをソフトコピーブックに載っている検証ストリング314又は322と比較する。この比較が満足すべきものであれば出版社の著作権使用料請求プログラム74だけが許可メッセージを送ることができる。

【0095】図10及び図11はユーザのワークステーション10において実行される著作権使用料支払いプログラム45のフローチャートを示す。

【0096】図10はステップ400において、BookManager ソフトコピーブック読取りプログラム35を開始する。

【0097】次にステップ402において、著作権使用料支払いプログラム45を開始する。

【0098】次にステップ404において、定様式テキストストリーム25を入力してメモリ22に格納する。

【0099】次にステップ406において、特殊タグ(図8)を用いてエレメントを探索する。

【0100】次にステップ408において、特殊タグ360を用いてエレメントについてのテキストストリング362及びエレメント座標156をパラメータテーブル56L(図9)にロードする。

【0101】次にステップ410において、ブック著作権使用料フラグ366をパラメータテーブル56L(図9)にセットする。

【0102】次にステップ412において、章著作権使用料フラグ368をパラメータテーブル56L(図9)にセットする。

【0103】次にステップ414において、BookManager ソフトコピー読取りプログラム35を継続する。

【0104】次にステップ416において、ユーザの入力要求を検出することにより印刷又は複写する。

【0105】次にステップ418において、表示されるブックの章を識別する。

【0106】次にステップ420において、検査されている章の見出し342、章著作権使用料支払いメッセージ344及び章支払い総額346を表示する(図14)。

【0107】次にステップ422において、ブックタイトル300、ブック著作権使用料支払いメッセージ306及びブック支払い総額308を表示する(図14)。

【0108】次にステップ424において、ユーザをプロモートすることにより選択肢を入力する(図14)。

【0109】次にステップ426において、ユーザの名前、メールアドレス及びクレジットカード情報を含むユーザプロフィール23を得る。

【0110】次にステップ428において、出版社のデータプロセッサ15の電話番号310を得る。

【0111】次にステップ430において、通信アプリケーションプログラム47を用いて出版社のデータプロセッサ15に電話コールを配置する。

【0112】次にステップ432において、ユーザプロフィール23、ブックタイトル300、版302及びブック支払い総額308、ユーザの章又はブックの選択並びに検証ストリング348を含む要求を出版社に送ることにより印刷又は複写する。

【0113】次にステップ434において、出版社のプライベート鍵を用いて暗号化された検証ストリング及び START_PAGE 値を含む許可メッセージを出版社のプロセッサ15から受信する。

【0114】次にステップ436において、メッセージ認証プログラム43の公開鍵暗号アルゴリズムを用いて、出版社の公開鍵312で、暗号化された検証ストリングを復号化する。

【0115】次にステップ438において、出版社からのこの復号化された検証ストリングが検証ストリング348と比較されると、プリンタ44又はディスクドライブ46は START_PAGE 値で始まる選択された章又はソフトコピーブックを印刷又は複写することができる。

【0116】次にステップ440において、選択された各ページを上手く印刷又は複写す

るとACK応答を出版社のプロセッサ15に送る。

【0117】次にステップ442において、出版社の許可メッセージをこの複製物に任意に印刷する。

【0118】次にステップ444において、BookManager ソフトコピー読取りプログラム35に戻る。

【0119】図12は出版社のデータプロセッサ15において実行される著作権使用料請求プログラム74のフローチャートである。

【0120】図12はステップ500において、著作権使用料請求プログラム74を開始する。

【0121】次にステップ501において、ユーザプロファイル23、ブックタイトル300、版302及びブック支払い総額308、ユーザの章又はソフトコピーブックの選択並びに検証ストリング348を含む要求をユーザワークステーション10から受信して印刷又は複写する。

【0122】次にステップ502において、ユーザのクレジットを検査する。

【0123】次にステップ504において、ブックデータ70を得る。

【0124】次にステップ505において、このユーザ及びソフトコピーブックについて障害レコードがあるとき、停止したページ= START_PAGE を障害レコードにセットするか又は START_PAGE =1をセットする。

【0125】次にステップ506において、出版社のプライベート鍵76を用いて検証ストリングを暗号化する。

【0126】次にステップ508において、出版社の公開鍵76を用いて暗号化された検証ストリング及び START_PAGE 値を含む許可メッセージをアSEMBルする。

【0127】次にステップ510において、許可メッセージをユーザのワークステーション10に送る。

【0128】次にステップ512において、複製すべきページ数 MAX_PAGEをブックデータ70から得る。

【0129】次にステップ514において、 PER_PAGE費用を計算し、BILL=0及びPAGE=START_PAGE をセットする。

【0130】次にステップ516において、ユーザワークステーション10からACKを受信する。

【0131】次にステップ518において、BILL=BILL+ PER_PAGE : PAGE=PAGE+1 【0132】次にステップ520において、一段と多くのACKがあるとき、ステップ516に進むか又はステップ522に進む。

【0133】次にステップ522において、PAGE< MAX_PAGEであるとき、ユーザプロファイル23、ブックタイトル300、版302及びブック支払い総額308、ユーザによる章又はソフトコピーブックの選択並びに複製を失敗した場所のPAGEの値を含む障害レコードを作成する。

【0134】次にステップ524において、請求された支払い総額=BILLを用いてユーザへの送り状を印刷する。

【0135】次にステップ526において、メインプログラムに戻る。

【0136】図13はソフトコピーブックの定様式テキストストリーム25及び著作権使用料情報ファイル58を含む、本発明の第2の実施例におけるCD-ROM55'を示す。

【0137】図14はユーザのワークステーションに表示される著作権使用料支払い情報を示す。この例において2つのオプションが選択肢としてユーザに提示されている。第1のオプションは「現在表示されている章についての複製料金」である。図示の例では「第2章の見出し」についてである。ユーザはボックスに「X」を入力することによってこのオプションを選択してもよい。またユーザはマウスのような指示装置を用いることによりこれを選択してもよい。第2のオプションは「表示されているブック全体についての複製料金」についてである。このオプションの場合、「ブックタイトル」、「第2版」及び「ブック複製料金」が表示され、上述の手法と同じ手法によりこれらを選択することができる。

【0138】その結果、本発明は、読者がソフトコピーブックの一部又はすべてを複製したいとき、ソフトコピーブックの出版社及び著作者への著作権使用料の支払いを読者に強

要する改善された手段を提供する。

【0139】上述の通り本発明をその最適な実施例に基づいて図示、説明したが、本発明の精神及び範囲から脱することなく詳細構成について種々の変更を加えてもよい。

【0140】

【発明の効果】上述のように本発明によれば、読者がソフトコピーブックの一部又はすべてを複製するときに、ソフトコピーブックの出版社及び著作権者への著作権使用料の支払いを読者に強要する手段を設けることにより、著作権及び出版社はその著作物における著作権を簡易かつ確実に保護することができる。

【図面の簡単な説明】

【図1】図1は著作権使用料支払いプログラムを含む、本発明によるユーザのワークステーションのシステムブロック図である。

【図2】図2は著作権使用料請求プログラムを含む、本発明による出版社のデータプロセッサのシステムブロック図である。

【図3】図3はエレメントタグ及びそれに関連したテキストについての第1の例を示す略線図である。

【図4】図4は図3の第1の例のエレメントタグ及びそれに関連したテキストのメモリ編成を示す略線図である。

【図5】図5は図3及び図4の第1の例についてのテキストを表示した略線図である。

【図6】図6は定様式テキストストリーム25の第2の例についてのメモリイメージを一段と詳細に示したものであり、特に、定様式テキストストリームにおいて、各特定のエレメントと結合したエレメント座標156を示す略線図である。

【図7】図7は図6の第2の例のテキストを表示した略線図である。

【図8】図8はロードされていない初期のパラメータテーブル56を示す表である。

【図9】図9は専用タグを用いてエレメントからの情報をロードされた後のパラメータテーブル56を示す表である。

【図10】図10は著作権使用料支払いプログラム45のフローチャートである。

【図11】図11は著作権使用料支払いプログラム45のフローチャートである。

【図12】図12は著作権使用料請求プログラム74のフローチャートである。

【図13】図13はソフトコピーブック及び著作権使用料情報ファイル58を含む、本発明の第2の実施例におけるCD-ROM55'を示す略線図である。

【図14】図14はユーザのワークステーションに著作権使用料支払い情報を表示した略線図である。

【符号の説明】

10……ワークステーション、15……出版社のデータプロセッサ、20、94……プロセッサ、22、96……メモリ、23……ユーザプロファイル、25……定様式テキストストリーム、26……ディスプレイ、26'……ディスプレイパツファ、27……オペレーティングシステム、29、98……バス、35……BookManager ソフトコピー読取りプログラム、43……メッセージ認証プログラム、44……プリンタ、45……著作権使用料支払いプログラム、46……磁気ディスクドライブ、47……通信アプリケーションプログラム、48、92……通信アダプタ、49……通信ネットワーク、50……キーボード、55、55'……CD-ROMドライブ、56……パラメータテーブル、56L……ロード済みパラメータテーブル、58……著作権使用料支払い情報ファイル、70……第1のブックデータ、72……第2のブックデータ、74……著作権使用料請求プログラム、76……出版社のプライベート鍵、78……公開鍵暗号化プログラム、79……通信アプリケーションプログラム、80……オペレーティングシステム、156……座標システム。

【図1】

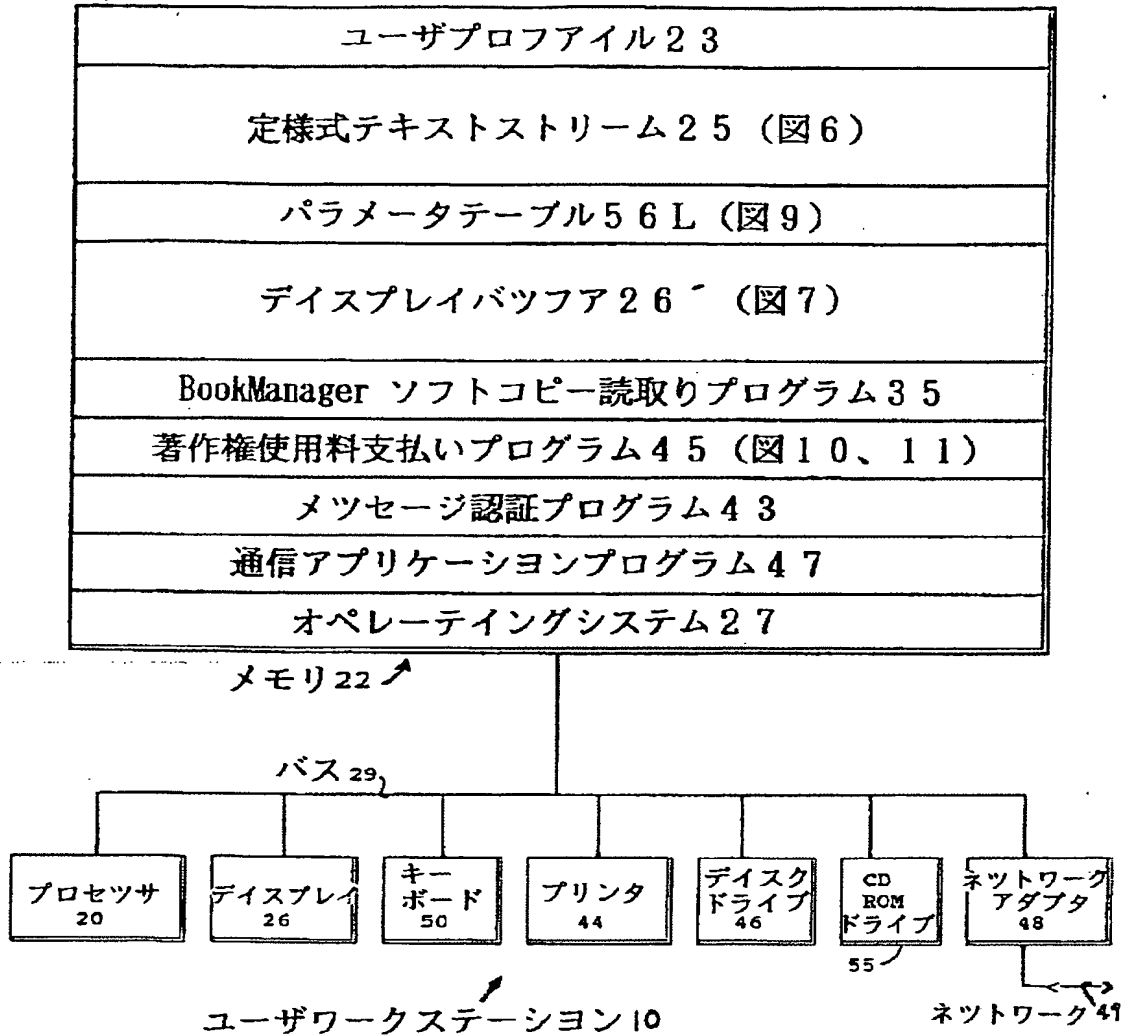


図1 ユーザワークステーション

【図2】

出版社のデータプロセッサ15

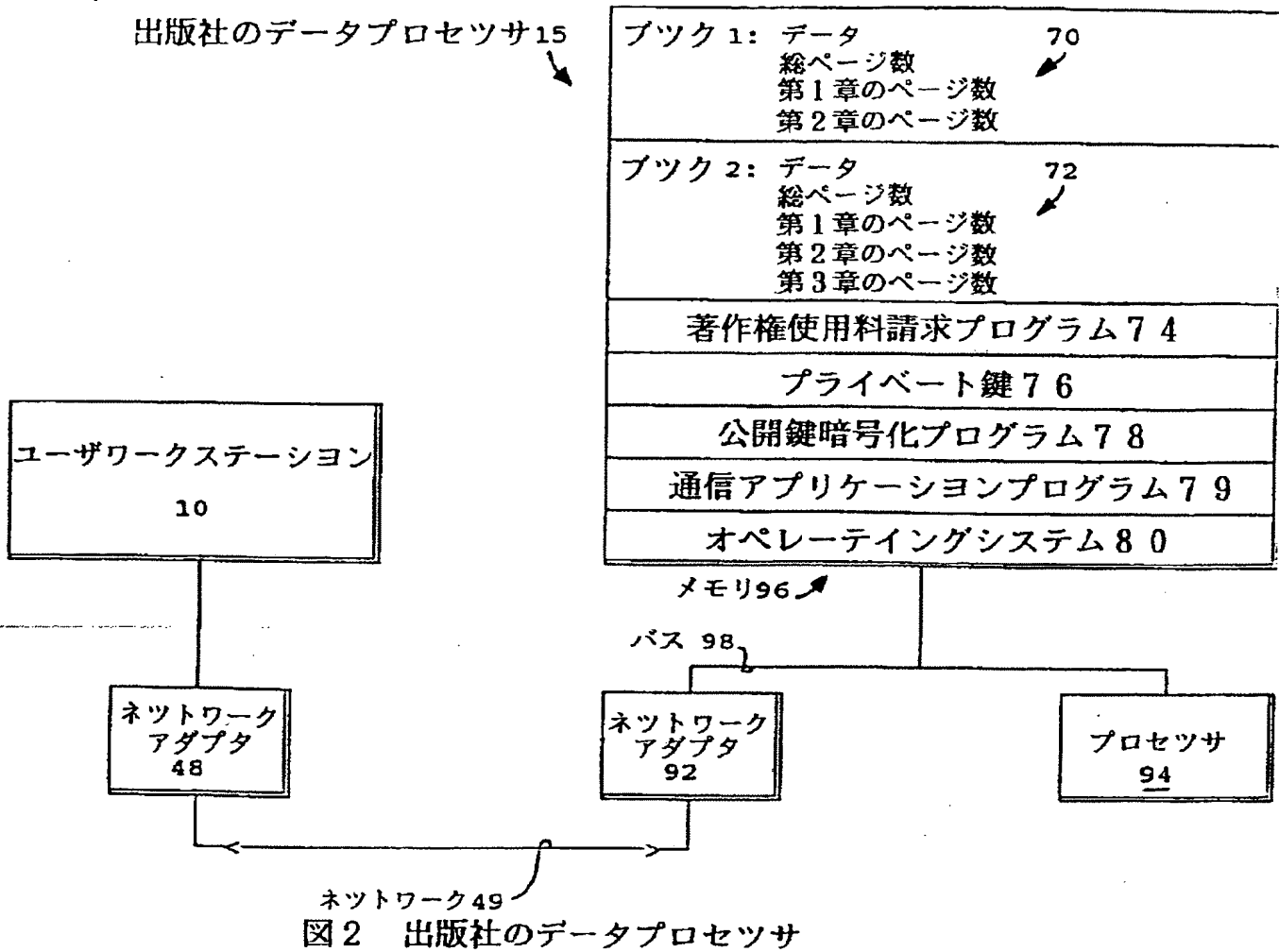


図2 出版社のデータプロセッサ

【図3】

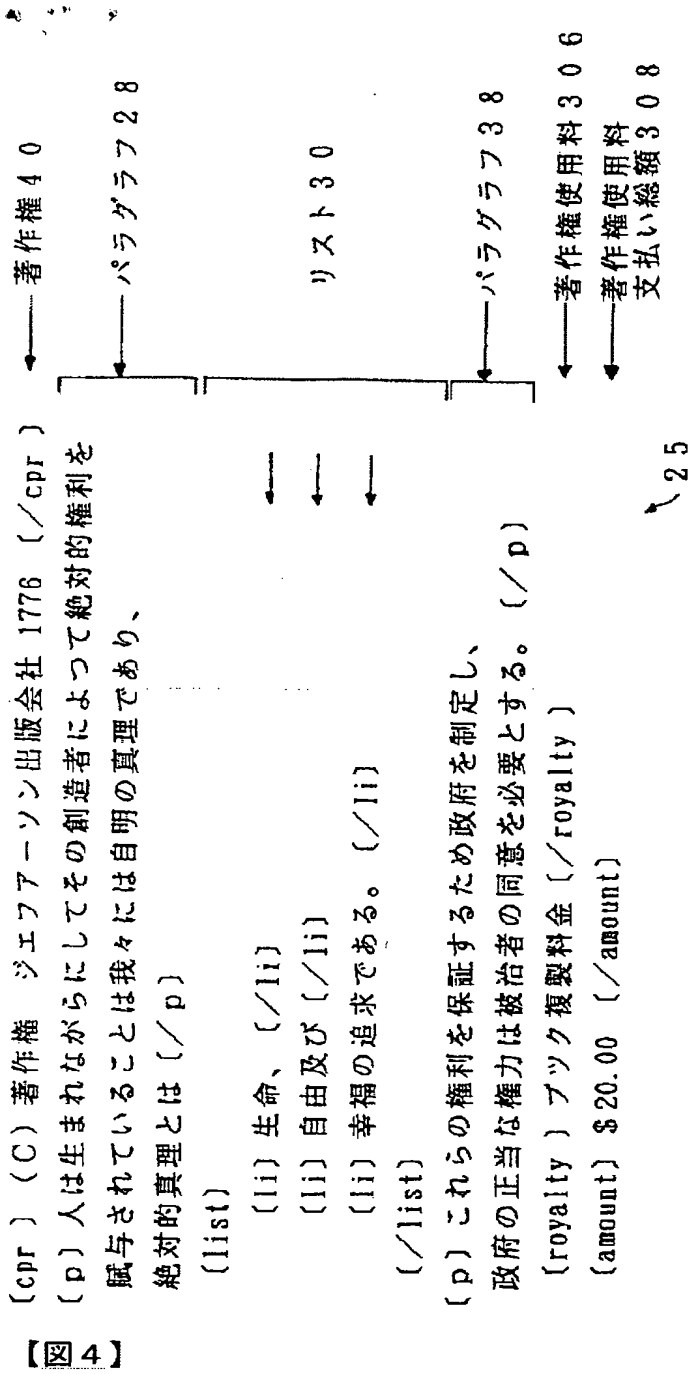


図3 エレメントタグ及びそれに関連したテキストの制御例

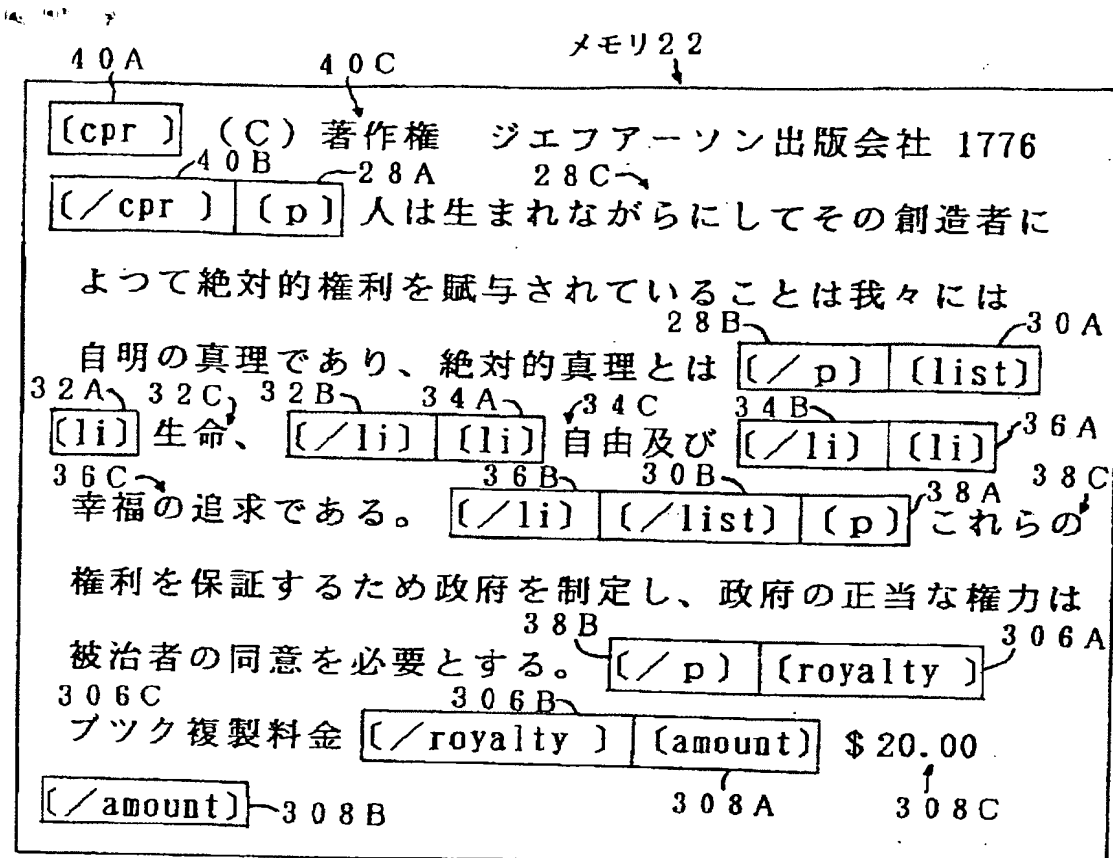


図4 エLEMENTタグ及びそれに関連したテキストの
制御についてのメモリ編成

【図5】

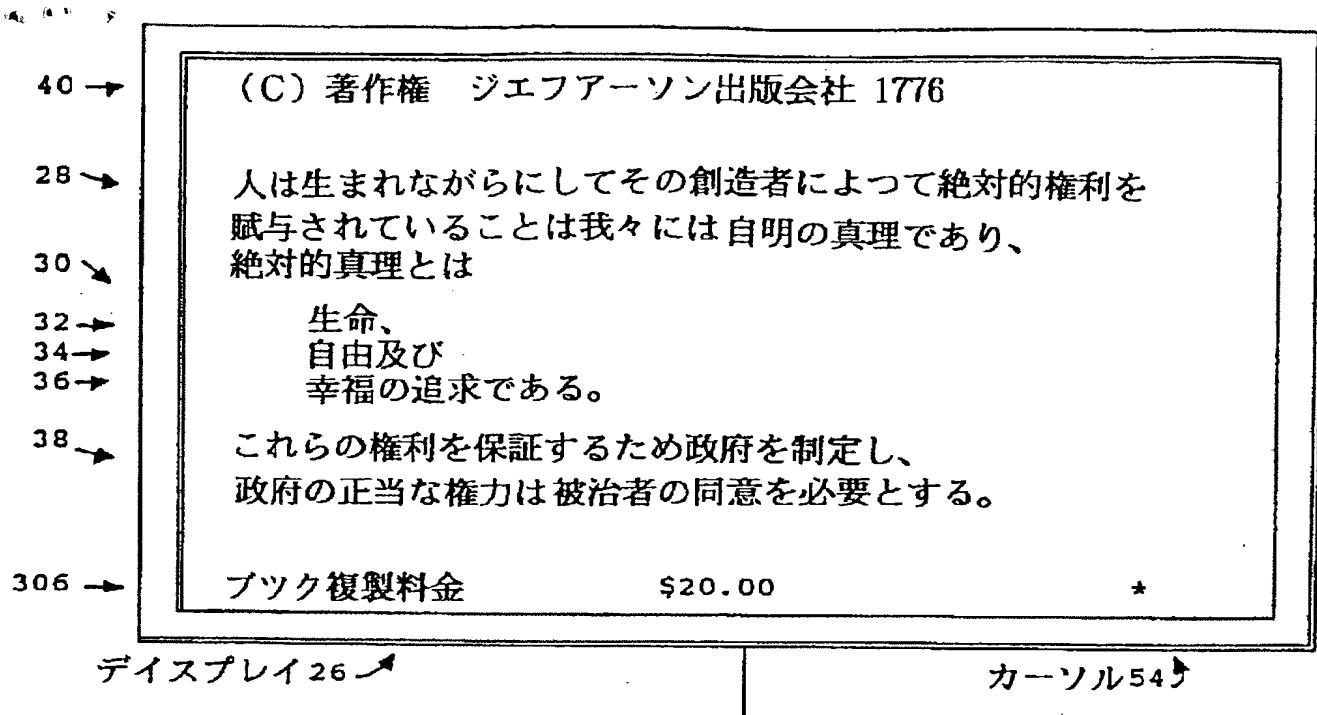


図5 エLEMENTタグ及びそれに関連したテキストの表示

【図13】

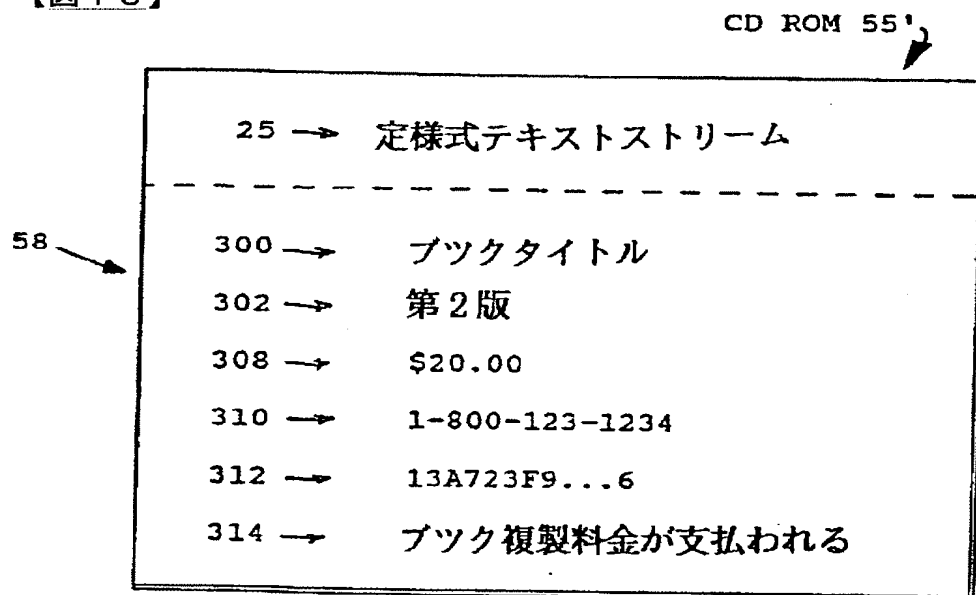


図13 CD-ROM 55

【図6】

定様式テキストストリーム²⁵

座標 156

bk h1 h2 h3 p el

300	[bk]ブックタイトル[/bk]	1	0	0	0	0	1
302	[ed] 第2版 [/ed]	1	0	0	0	1	1
304	[cpr] (C)ABC Co 1990. [/cpr]	1	0	0	0	2	1
306	[royalty] ブック複製料金 [/royalty]	1	0	0	0	3	1
308	[amount] \$20.00 [/amount]	1	0	0	0	3	2
310	[phone] 1-800-123-1234 [/phone]	1	0	0	0	3	3
312	[public key] 13A723F9...6 [/public key]	1	0	0	0	3	4
314	[validation] ブック複製料金が支払われる[/validation]	1	0	0	0	3	5
316	[h1] 第1章の見出し [/h1]	1	1	0	0	0	1
318	[royalty] 章複製料金 [/royalty]	1	1	0	0	1	1
320	[amount] \$ 1.00 [/amount]	1	1	0	0	1	2
322	[validation] 第1の章の料金が支払われる[/validation]	1	1	0	0	1	3
324	[p]テキストのパラグラフ[/p]	1	1	0	0	2	1
326	[h2] 第1のトピック見出し[/h2]	1	1	1	0	0	1
329	[p]テキストのパラグラフ[/p]	1	1	1	0	1	1
330	[p]テキストのパラグラフ[/p]	1	1	1	0	2	1
332	[h2] 第2のトピック見出し [/h2]	1	1	2	0	0	1
334	[p]テキストのパラグラフ[/p]	1	1	2	0	1	1
336	[h3] 第1のサブトピック見出し[/h3]	1	1	2	1	0	1
338	[p]テキストのパラグラフ[/p]	1	1	2	1	1	1
340	[p]テキストのパラグラフ[/p]	1	1	2	1	2	1
342	[h1] 第2章の見出し [/h1]	1	2	0	0	0	1
344	[royalty] 章複製料金 [/royalty]	1	2	0	0	1	1
346	[amount] \$ 2.00 [/amount]	1	2	0	0	1	2
348	[validation] 第2の章の料金が支払われる[/validation]	1	2	0	0	1	3
350	[p]テキストのパラグラフ[/p]	1	2	0	0	2	1

22

図6 定様式テキスト及びエレメント座標のメモリーイメージメモリ

【図7】

300 →	ブックタイトル
302 →	第 2 版
304 →	(C)ABC Co 1990
306 →	ブック複製料金
308 →	\$20.00
310 →	1-800-123-1234
	:
316 →	第 1 章の見出し
318 →	章複製料金
320 →	\$ 1.00
324 →	テキストのパラグラフ
326 →	第 1 のトピック見出し
328 →	テキストのパラグラフ
330 →	テキストのパラグラフ
332 →	第 2 のトピック見出し
334 →	テキストのパラグラフ
336 →	第 1 のサブトピック見出し
338 →	テキストのパラグラフ
340 →	テキストのパラグラフ
	:
342 →	第 2 章の見出し
344 →	章複製料金
346 →	\$ 2.00
350 →	テキストのパラグラフ

ディスプレイ 26 ↗

図 7 図 6 のテキストの表示

【図 8】

ロードされていない初期のパラメータテーブル56

【図9】

360	362	156	364	366	368
特殊タグ	特殊なタグを用いてエレメントからロードされたテキスト	エレメント座標 b k h1 h2 h3 p e l	ディスプレイテキスト	ブック著作権 使用料フラグ	著作権 使用料フラグ
300I bk			yes		
302I ed			yes		
304I cpr			yes		
306I royalty			yes		
308I amount			yes		
310I phone			yes		
312I public key			no		
314I validation			no		

図8 ロードされていない初期のパラメータテーブル

【図10】

360	362	ロード済みパラメータテーブル56L	156	364	366	368
特殊タグ	特殊なタグを用いて エレメントからロードされた テキストストリング	エレメント座標 bk h1 h2 h3 p el	ディスプレイ テキスト	著作権 使用料 フラグ	著作権 使用料 フラグ	著作権 使用料 フラグ
300L bk	ブックタイトル	1 0 0 0 0 1	yes			
302L ed	第2版	1 0 0 0 1 1	yes			
304L cpr	(C)ABC Co.1990	1 0 0 0 2 1	yes			
306L royalty	ブック複製料金	1 0 0 0 3 1	yes	x		
308L amount	\$20.00	1 0 0 0 3 2	yes			
310L phone	1-800-123-1234	1 0 0 0 3 3	yes			
312L public key	13A723F9...6	1 0 0 0 3 4	no			
314L validation	ブック複製料金が支払われる	1 0 0 0 3 5	no			
318L royalty	章複製料金	1 1 0 0 1 1	yes		x	
320L amount	\$ 1.00	1 1 0 0 1 2	yes			
322L validation	第1の章の料金が支払われる	1 1 0 0 1 3	no			
344L royalty	章複製料金	1 2 0 0 1 1	yes			x
346L amount	\$ 2.00	1 2 0 0 1 2	yes			
348L validation	第2の章の料金が支払われる	1 2 0 0 1 3	no			

図9 ロード済みパラメータテーブル

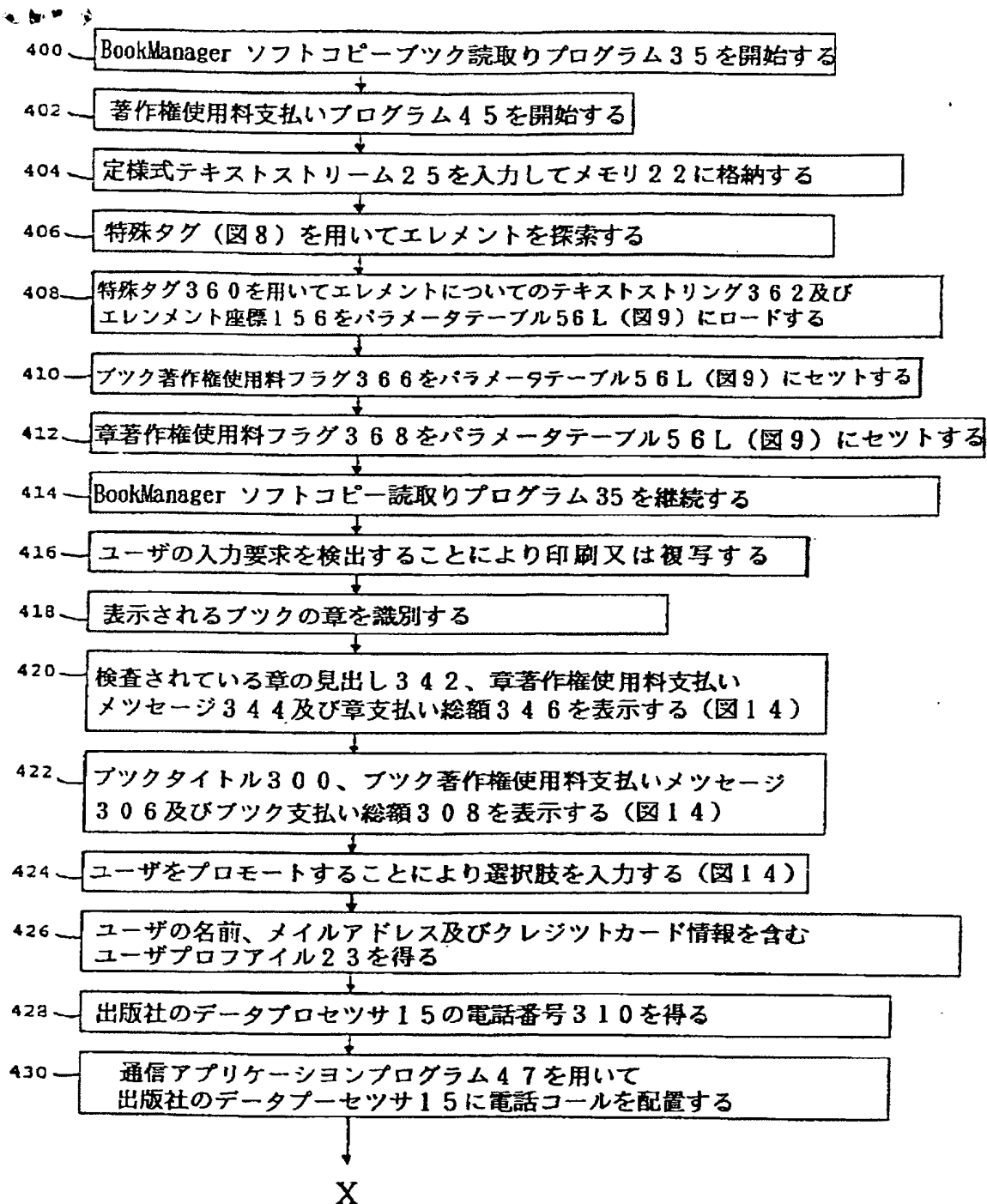


図 10 著作権使用料支払いプログラム

【図 11】

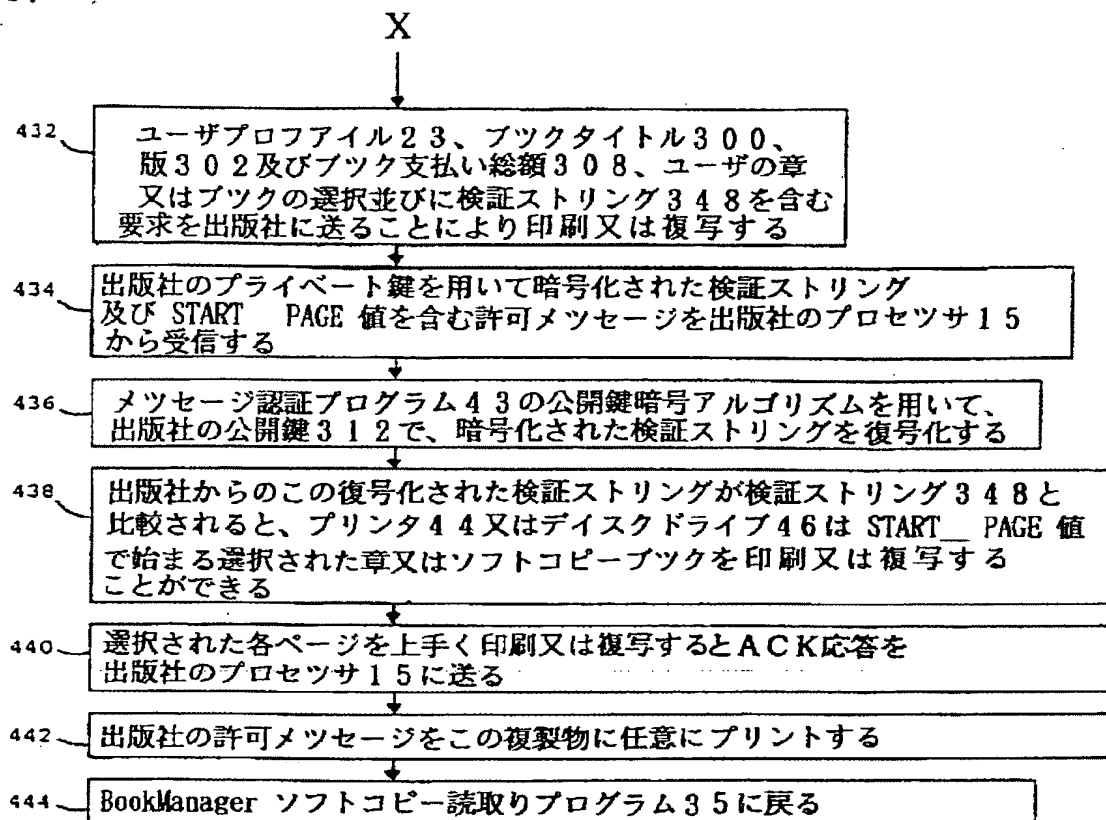


図11 著作権使用料支払いプログラム

【図12】

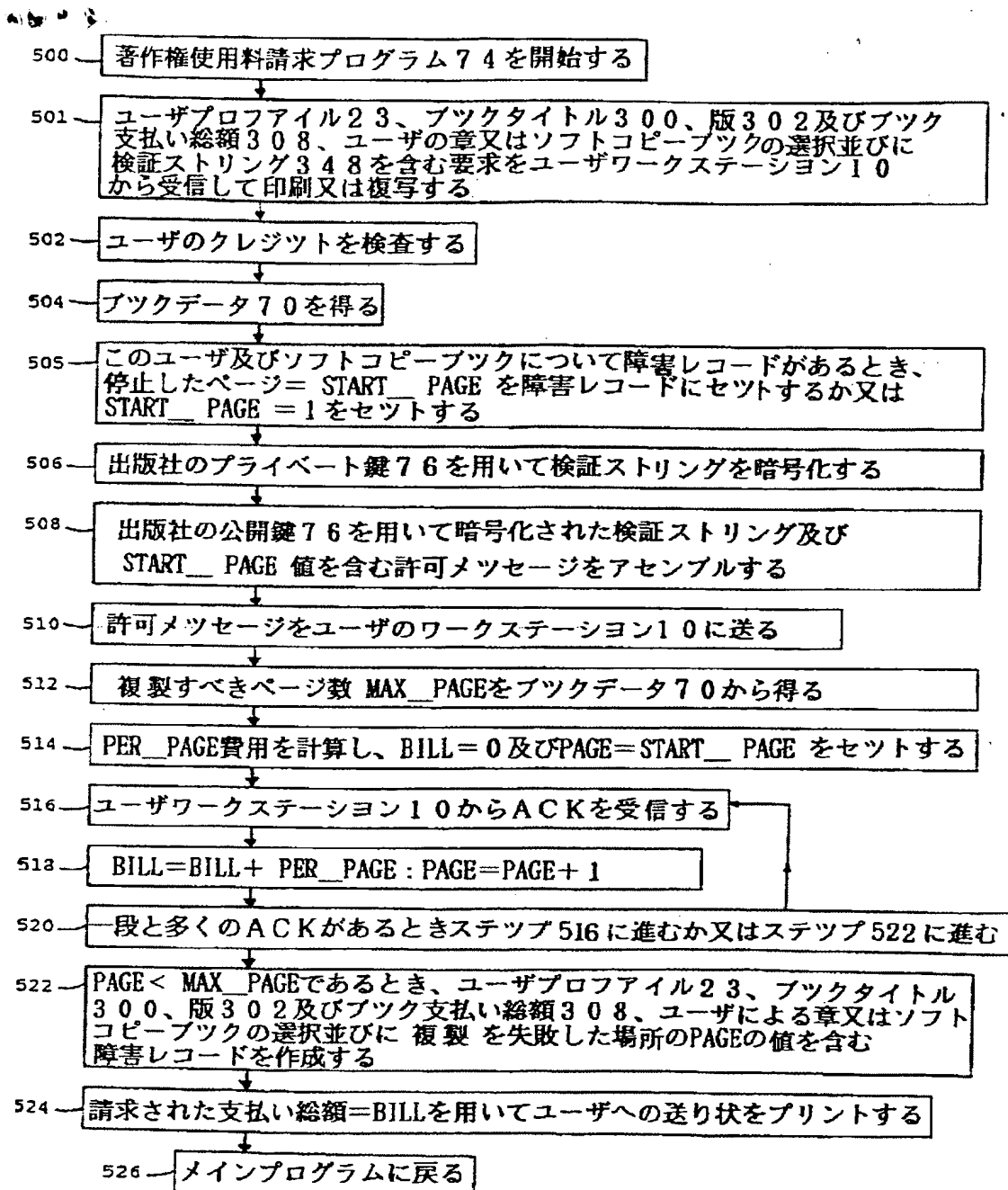


図12 著作権使用料請求プログラム

【図14】

4. 2. 2.

342 →

344 →

346 →

300 →

302 →

306 →

308 →

310 →

現在表示されている章
についての複製料金

第2章の見出し ボックスに「X」を入力する

章複製料金

☐

\$ 2.00

表示されているブック全体
についての複製料金

ブックタイトル ボックスに「X」を入力する

第2版

☐

ブック複製料金

\$20.00

選択した項目を入力して
1-800-123-1234をコール

選択した項目の料金の総額をあなたの
クレジットカードの番号に請求します

ディスプレイ26

図14 ユーザへの指示メッセージの表示

PATENT ABSTRACTS OF JAPAN

(11) Publication number : 06-318167
 (43) Date of publication of application : 15.11.1994

(51) Int. Cl. G06F 12/00

(21) Application number : 06-038681 (71) Applicant : TOSHIBA CORP
 (22) Date of filing : 09.03.1994 (72) Inventor : TAKAHASHI TOSHINARI

(30) Priority

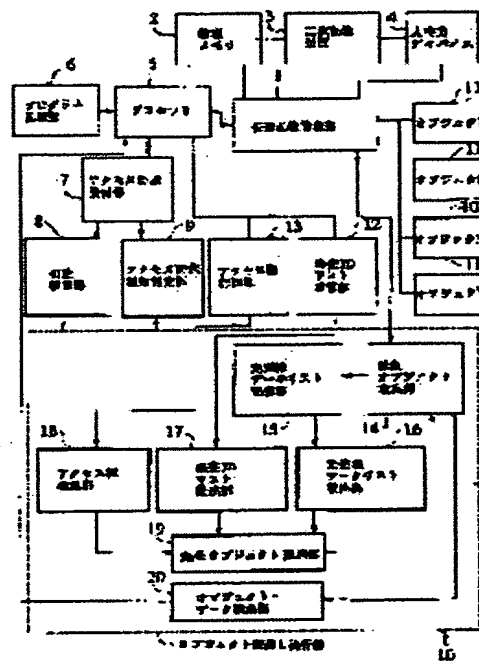
Priority number : 05 48159 Priority date : 09.03.1993 Priority country : JP

(54) DEVICE AND METHOD FOR SCANNING OBJECT

(57) Abstract:

PURPOSE: To provide an object scanning procedure and a scanning right flexibly by rewriting a scan ID list corresponding to the selection of an object to be scanned when the scan ID list is executed by collating with a scan right data list defined on the object and the object to be scanned is selected.

CONSTITUTION: A scan right ID list detecting part 17 takes out the scan ID list from a scan ID list storage part 12, and an access right verification part 18 verifies whether or not the access right of the object exists on a process making access based on data in an access right storage part 13. A scan object selection part 19 appropriately judges the object made to be a scan object appropriately based on information from the access right verification part 18, the scan ID list detecting part 17, and a scan right data list detecting part 16, and requests the take-out of the object required furthermore to a target object take-out part 14. An object read-out part 20 obtains the obtained object data, and returns the content of the data to a processor 5.



LEGAL STATUS

[Date of request for examination] 19.09.2000

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application, other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for
application]

[Patent number]

3623979

[Date of registration]

03. 12. 2004

[Number of appeal against examiner's
decision of rejection]

[Date of requesting appeal against
examiner's decision of rejection]

[Date of extinction of right]

全項目

- (19) 【発行国】日本国特許庁 (JP)
 (12) 【公報種別】公開特許公報 (A)
 (11) 【公開番号】特開平6-318167
 (43) 【公開日】平成6年(1994)11月15日
 (54) 【発明の名称】オブジェクト走査装置及び方法
 (51) 【国際特許分類第5版】

G06F 12/00 537 A 8944-5B

【審査請求】未請求

【請求項の数】13

【出願形態】OL

【全頁数】24

- (21) 【出願番号】特願平6-38681
 (22) 【出願日】平成6年(1994)3月9日
 (31) 【優先権主張番号】特願平5-48159
 (32) 【優先日】平5(1993)3月9日
 (33) 【優先権主張国】日本 (JP)
 (71) 【出願人】

【識別番号】000003078

【氏名又は名称】株式会社東芝

【住所又は居所】神奈川県川崎市幸区堀川町72番地

(72) 【発明者】

【氏名】高橋 俊成

【住所又は居所】神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内

(74) 【代理人】

【弁理士】

【氏名又は名称】三好 秀和 (外3名)

(57) 【要約】

【目的】 本発明は、アプリケーションの動作と独立に一般ユーザが作成した各々のオブジェクトに対して柔軟なアクセス手順およびアクセス権を与えることができるオブジェクト走査装置を提供する。

【構成】 リンク103で関連付けられた複数のオブジェクト100の走査に際し、オブジェクトを走査する際の条件となる走査IDリスト104をオブジェクトに定義された走査権データ・リストと照合して実行すると共に、走査すべきオブジェクトを選択する際に、この選択方法に応じて走査IDリスト104を走査IDリスト変更部105によって書き換える。

【特許請求の範囲】

【請求項1】 複数のサブオブジェクトによって形成されるオブジェクトを走査するオブジェクト走査装置にあって、誰が各サブオブジェクトを走査出来るかを示す各サブオブジェクトの走査権と共にオブジェクトを記憶するメモリ手段と、該オブジェクトに対するアクセスを要求しているユーザを同定する少なくとも1つの走査IDを記憶する走査IDリスト手段と、該オブジェクトの走査時に、該走査IDリスト手段に記憶された走査IDと、当該オブジェクトに対して前記記憶手段が記憶する走査権に基づいて、走査すべきサブオブジェクトを選択する走査オブジェクト選択手段と、上記走査オブジェクト選択手段の選択したサブオブジェクトを当該オブジェクトに対するアクセスの結果として出力する出力

手段どからなることを特徴とするオブジェクト走査装置。

【請求項2】 前記記憶手段は、誰が該オブジェクトをアクセス出来るかを示す各オブジェクトのアクセス権と共に前記オブジェクトを記憶し、前記走査オブジェクト選択手段は前記走査IDで同定されたユーザに対するアクセス権が前記記憶手段に記憶されている時にサブオブジェクトを選択することを特徴とする請求項1記載のオブジェクト走査装置。

【請求項3】 ユーザのアクセス権により前記オブジェクトにアクセスし、前記出力手段が該オブジェクトに対するアクセスの結果として出力したサブオブジェクトを使って該オブジェクトを操作するプログラムを実行するプロセッサ手段を有することを特徴とする請求項2記載のオブジェクト走査装置。

【請求項4】 前記記憶手段は、誰が該オブジェクトをアクセス出来るかを示す各オブジェクトのアクセス権と共に前記オブジェクトを記憶し、更に前記走査オブジェクト選択手段が選択したサブオブジェクトについて前記走査IDで同定されたユーザに対するアクセス権の有無を検証するアクセス権検証手段を有することを特徴とする請求項1記載のオブジェクト走査装置。

【請求項5】 ユーザのアクセス権により前記オブジェクトにアクセスし、前記出力手段が該オブジェクトに対するアクセスの結果として出力したサブオブジェクトを使って該オブジェクトを操作するプログラムを実行するプロセッサ手段を有することを特徴とする請求項4記載のオブジェクト走査装置。

【請求項6】 前記記憶手段は前記オブジェクトを、リンクで関連付けられた複数のサブオブジェクトから形成されるオブジェクトとして記憶することを特徴とする請求項1記載のオブジェクト走査装置。

【請求項7】 前記記憶手段は前記オブジェクトを、該オブジェクトのサブオブジェクトを走査する手順を示す走査制御データと共に記憶することを特徴とする請求項1記載のオブジェクト走査装置。

【請求項8】 前記走査IDリスト手段が記憶する走査IDを変更する走査IDリスト変更手段を有することを特徴とする請求項1記載のオブジェクト走査装置。

【請求項9】 前記走査IDリスト変更手段は前記走査IDリスト手段が記憶する走査IDの中で、どのサブオブジェクトの走査権も無いものを削除することを特徴とする請求項8記載のオブジェクト走査装置。

【請求項10】 前記走査オブジェクト選択手段が選択したサブオブジェクトの一つは更なるサブオブジェクトを含むもので、前記記憶手段は誰が該更なるサブオブジェクトを走査出来るかを示す該更なるサブオブジェクトの走査権と共に該更なるサブオブジェクトを記憶し、前記走査IDリスト変更手段は前記走査IDリスト手段が記憶する走査IDの中で、どの該更なるサブオブジェクトの走査権も無いものを、該サブオブジェクトの一つを走査する時に削除することを特徴とする請求項8記載のオブジェクト走査装置。

【請求項11】 前記各サブオブジェクトの走査権は該各サブオブジェクトを走査出来るとされる者に該各サブオブジェクトの走査が許可されるようになることの該各サブオブジェクトへのアクセスの種類も示すことを特徴とする請求項1記載のオブジェクト走査装置。

【請求項12】 前記オブジェクトを各サブオブジェクトの走査権を指定してメールとして作成し、作成したメールをメールアドレスを指定して転送するメール作成手段と、指定されたメールアドレスに応じて該メール作成手段から転送されたメールを受信し、前記出力手段が出力するサブオブジェクトを受信したメールの内容として取り出すメール受信手段と、からなるメールツールを有することを特徴とする請求項1記載のオブジェクト走査装置。

【請求項13】 複数のサブオブジェクトによって形成されるオブジェクトを走査するオブジェクト走査方法であって、誰が各サブオブジェクトを走査出来るかを示す各サブオブジェクトの走査権の付与と共にオブジェクトを記憶するステップと、該オブジェクトに対するアクセスを要求しているユーザを同定する少なくとも1つの走査IDを記憶するステップと、該オブジェクトの走査時に、当該オブジェクトにおいて該走査IDで同定されたユーザに対して付与された走査権に基づいて、走査すべきサブオブジェクトを選択するステップと、前記選択ステップにおいて選択されたサブオブジェクトを当該オブジェクトに対するアクセスの結果として出力するステップとからなることを特徴とするオブジェクト走査方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、計算機のファイル・システム、仮想記憶空間、データベースなどの構造化された記憶手段のオブジェクト走査装置に関するものである。

【0002】

【従来の技術】従来、リンクで関連付けられた複数のオブジェクトを走査する方法には、ハイパー・テキストのように順次リンクをたどる方法や、計算機のファイル・システムのように与えられた名前から該当するオブジェクト群を検索して呈示する方法がある。また、ユーザによってアクセス可能なオブジェクトを限定する方法や、一時的にアクセス権を変更する方法なども知られている。

【0003】また、これらの方法を組み合わせることによってアクセス可能なオブジェクトを一般的に表現することやリンクで関連付けられた複数のオブジェクトを連続した一つのデータとして読み出すことも従来から知られている。

【0004】一方、最近では、これらの従来技術を組み合わせ、複数のオブジェクトを組み合わせ階層的に大きなオブジェクトを構成する際に、アクセス権の情報を各サブ・オブジェクトに持たせることによって、パラメータを変えて走査の手順を変え、一つの組み合わせオブジェクトを見かけ上何種類もの組み合わせオブジェクトのように表現することが試みられている。

【0005】以下に、このような考えのものについて説明する。

【0006】今、高橋さんがグループAに属する名簿ファイルを作成しているとする。

【0007】図20は名簿本体を一つのオブジェクトとして作成した状態を示し、121は名簿本体を示すオブジェクトである。この例ではファイル名が無いものが示されているが名前が付いていてもかまわない。

【0008】1211は、オブジェクト121のオーナーのIDを示している。ここに示された番号が高橋さんであることは、図21のID表131の対応する欄1311を参照することにより知ることができる。1212はオブジェクト121のグループであり、これがグループAであることは同様にID表131の対応する欄1312から知ることができる。1213はオブジェクト121オーナーに対するアクセス権である。RD、WR、EXの各フィールドはそれぞれ、読み出し、書き込み、実行の権限を示している。このオブジェクト121は単なるデータで、プログラムではないから実行はできない。ファイルのオーナーは当然読み出しも書き込みも出来るからRDとWRは1であり、EXは0になっている。1214はオブジェクト121のグループに対するアクセス権である。グループのデータであるから、読み出しは可能でRDが1となっているが、データの改ざんを防ぐために書き込みは不可能でWRが0となっている。1215はオブジェクト121のその他の人に対するアクセス権である。グループの名簿は機密性があり、読み出しも許可しないので、RDが0になっている。そして、1216はオブジェクト121のデータ本体であり、名簿のデータが記入されている。現実にはここにデータ本体を記入するのではなく、データ本体へのポインタやデータ本体の記入されたブロックのIDなどを記入する場合もあるがその例はここでは特に示さない。

【0009】次に、このような名簿ファイルを管理するために図22に示すようなコメントを記入したオブジェクト141を作成する。ここで、1411はオーナーのID、1412はグループのIDである。このファイルのグループはグループID1412に登録された“320”からグループAとなる。このことはID表131の対応するフィールド1312からわかる。また、1413はオブジェクト141のオーナーに対するアクセス権であり、上述の1214と同様である。1414、1415はそれぞれこのオブジェクトのグループおよびその他の人に対するアクセス権である。このオブジェクトの内容は、他人にはもちろんのこと、グループAのメンバーにもみられてはならないから、RDが共に0になっていることが図20のオブジェクト121の場合とは異なる。

【0010】このようにして各オブジェクトごとに異なるアクセス許可を与えることは、従来から広く行われてきた。近年では、この例のような複数のオブジェクトを一つのファイルのように扱うことが試みられている。それを例示したのが図23である。ここで、オ

オブジェクト151はオブジェクト121とオブジェクト141とを関連付け、1つの組み合わせオブジェクトを表現するためのオブジェクトである。

【0011】このオブジェクト151にはオブジェクト名1511が付いている。これは一般の計算機のファイル・システムにおけるファイル名に相当するものである。また、1512はオーナーのID、1513はグループのID、1514はこのオブジェクトのオーナーに対するアクセス権をそれぞれ示している。このアクセス権については、今後名簿に書かれたデータやコメントを変更することはあるが、この「グループA名簿」という名簿自体の構造を変更する必要はないから、オーナーに対しても書き込みを不可としてあるためWRは0である。1515はオブジェクト151のグループに対するアクセス権である。1516はオブジェクト151のその他の人に対するアクセス権である。その他の人に名簿を見られてはならないが、この「グループA名簿」というファイルがグループAの名簿本体とそれに対するコメントから成っているということを隠す理由は特になく、読み出しを許可している。1517はオブジェクト151がサブ・オブジェクトを持つことを示すビットであり、1になっている。サブ・オブジェクトをいくつ持つかは、サブ・オブジェクト数フィールド1518の内容を読むと2個であることがわかる。2個のサブ・オブジェクトが何であるかは、1番目のサブ・オブジェクトのID1519および2番目のサブ・オブジェクトのID1520を見ればわかる。これらのIDはそれぞれサブ・オブジェクト121及び141を示している。

【0012】このようにして複数のオブジェクトを構成することにより、高橋さん（ID128番）が「グループA名簿」を表現しているオブジェクト151を読むと、グループA名簿本体のデータの格納されたサブ・オブジェクト121および、コメントの記入されたサブ・オブジェクト141があることがわかり、また、それらサブ・オブジェクトはいずれも高橋さん（ID128番）にアクセス権があるから、アクセスすることができる。

【0013】一方、グループA（ID320番）に属する他の人が「グループA名簿」というファイルを読むと、グループAの名簿本体のデータの格納されたサブ・オブジェクト121の内容は読むことができるが、コメントの記入されたサブ・オブジェクトを読もうとしても、アクセス権がないから、読むことはできない。

【0014】なお、オブジェクト151の他の人に対するアクセス権1516のRDは1であるから、このオブジェクト151をたどってオブジェクト121およびオブジェクト141を走査することはグループ以外の人にもできることになるが、実際にはオブジェクト121およびオブジェクト141において、その他の人に対する読み出し権を出していないから、名簿のデータが読まれてしまうということは起こらない。

【0015】このような方法を用いることにより、ユーザごとにオブジェクトの全部または一部分だけをアクセス可能とすることが実現できる。

【0016】しかし、高橋さん（ID128番）が読むとグループA名簿はコメント付きのひとつのファイルであるように見え、その他のグループAの人（ID320番）が読むと、コメント無しのファイルであるように見える、といった柔軟な表現をすることはできない。

【0017】また、仮にそういった表現が可能になる機構を用意しても、次のような問題点が生じる。

【0018】いま、名簿のデータを用い、それを整形して表形式で印刷するようなプログラムがある場合、プログラムは、名簿にはコメントがないものと考えて作成されているものとする。そして、このプログラムをグループAの人が使えるように考えて作成した場合、高橋さん（ID128番）がこのプログラムを実行すると、プログラムが必要としていないコメントに対してもアクセス権があるため、読み出すことができ、このプログラムが正常に動作しないことがある。すなわち、プログラムは「グループA」という権限で名簿ファイルを読み出そうとしているにもかかわらず、それよりも権利の強い「高橋さん」という権限で名簿ファイルを読み出してしまうことがある。

【0019】そこで、こういった状況を回避するのに最もよく使われる方法の一つとして、プログラムの実行と同時にアクセス権を変更するSUID（USP 4,135,240）がある。このSUIDはOWNER IDとしての権限を変更するためのものである。そのままではこの問題の解決にはならないが、SUIDに多少の応用を加えることによって実現することができる。以下その解決例を示す。

【0020】図24に示すオブジェクト161は名簿整形用の一般プログラムを格納したオブジェクトであり、図25に示すオブジェクト171は、同様に名簿整形用のプログラ

仏である。これら図2.4と図2.5では、それぞれ同一部分には同一符号を付している。

【0021】この場合、オブジェクト161において、SAID (set access id) 1611は何も定義されていないので、オブジェクト161に記述されたプログラムは通常動作することを意味している。ここで1613はオーナーID、1614はグループのIDである。また、1615と1616はそれぞれこのオブジェクトのオーナーおよびグループに対するアクセス権である。さらに1617はその他の人のアクセス権で、グループA以外の人には実行することができない。そして、1618は実行プログラムの本体である。この実行プログラム1618は名簿データとしてオブジェクト151を用いている。

【0022】このようにしても、実行プログラム1618を実行した場合、高橋さん(ID128)以外の人を実行すると期待した結果が得られるが、高橋さんが実行すると、このプログラムが読むはずでなかったオブジェクト141のコメントも読んでしまい、正しく動作しない場合がある。

【0023】これに対し、図2.5のオブジェクト171はオブジェクト161とよく似ているが、SAID 1612の設定が異なる。SAID 1612には320番というIDがセットされている。これにより、このオブジェクト171を高橋さんが実行すると、高橋さんの本来のアクセス権である実行者としての権利を放棄し、320番という権利のみが使えるようになる(この動作はSUIDの仕組みを応用したものである)。

【0024】このようなオブジェクト171の実行プログラム1619では、データとして名簿のオブジェクト151およびそのサブ・オブジェクトとしてオブジェクト121、オブジェクト141をアクセスする場合のアクセス権としては320番が使われるので、プログラム1619を誰が実行してもアクセス可能なデータは同一になり、プログラム1619は期待通りの動作をする。

【0025】このような従来の方法では、プログラムと、それが使うデータとを関連付けることにより、特殊なアクセス権によるオブジェクト走査を可能としているが、この方法によると、プログラムにSAIDを付ける権利が不明確であることに問題がある。すなわち、容易にSAIDを付けることが出来ると、そのプログラムを使うことにより任意のアクセス権を得ることができ、オブジェクトの持っているアクセス権が意味をなさなくなってしまうことである。

【0026】したがって、SAIDを付ける権限を別の方法で管理しなければならないのであるが、従来それを解決する優れた方法は行われていなかった。例えば、ルートという特権ユーザにのみSAIDを付ける権限を与えるといった方法でこの問題を回避していた。

【0027】本来、オブジェクトのアクセス権を設定する権利は原則としてそのオブジェクトのオーナーにあるのであるから、その機構を特定のユーザによってしか実現できないというのは大きな制約となる。また、オブジェクト(データ)自体に持たせるはずのアクセス権が、他のプログラムの設定によってのみ制御できるという制約は非常に不便である。オブジェクトに対して当初期待したはずの保護機構が実現されない。

【0028】

【発明が解決しようとする課題】このように、従来のオブジェクト走査装置にあっては、オブジェクトに対する走査の条件としてアクセス権そのものを使わざるを得なかったために、オブジェクトの走査方法をその利用方法に応じて変化させることが困難であった。

【0029】そこで、この問題を解決するために、プログラムごとに特別のアクセス権を与える方法によりアクセス権を持つ部分を変更することが試みられているが、この方法においても実際にオブジェクトのどの部分をアクセスするかということについては、アプリケーション・プログラムのみに依存するため、作成したオブジェクトに対して、走査方法を規定することが非常に困難であった。また、オブジェクト全体を1つのファイルとみなしてアクセスする場合においても、アクセス権のみを用いたオブジェクトでは、実際にどのようなアクセスが行われるかということをおブジェクト自体に持たせることができなかった。

【0030】本発明は、上記事情に鑑みてなされたもので、アプリケーションの動作とは独立に一般ユーザが作成した各々のオブジェクトに対して柔軟なオブジェクト走査手順および走査権を与えることができるオブジェクト走査装置を提供することを目的とする。

【0031】

【課題を解決するための手段】本発明は、複数のサブオブジェクトによって形成されるオブジェクトを走査するオブジェクト走査装置にあって、誰が各サブオブジェクトを走査出

示す各サブオブジェクトの走査権と共にオブジェクトを記憶するメモリ手段と、該オブジェクトに対するアクセスを要求しているユーザを同定する少なくとも1つの走査IDを記憶する走査IDリスト手段と、該オブジェクトの走査時に、該走査IDリスト手段に記憶された走査IDと、当該オブジェクトに対して前記記憶手段が記憶する走査権に基づいて、走査すべきサブオブジェクトを選択する走査オブジェクト選択手段と、上記走査オブジェクト選択手段の選択したサブオブジェクトを当該オブジェクトに対するアクセスの結果として出力する出力手段とにより構成されている。

【0032】又、本発明は、複数のサブオブジェクトによって形成されるオブジェクトを走査するオブジェクト走査方法であって、誰が各サブオブジェクトを走査出来るかを示す各サブオブジェクトの走査権の付与と共にオブジェクトを記憶するステップと、該オブジェクトに対するアクセスを要求しているユーザを同定する少なくとも1つの走査IDを記憶するステップと、該オブジェクトの走査時に、当該オブジェクトにおいて該走査IDで同定されたユーザに対して付与された走査権に基づいて、走査すべきサブオブジェクトを選択するステップと、前記選択ステップにおいて選択されたサブオブジェクトを当該オブジェクトに対するアクセスの結果として出力するステップとにより構成されている。

【0033】

【作用】この結果、本発明によれば、リンクで関連付けられた複数のオブジェクトの走査に際し、オブジェクトを走査する際の条件となる走査IDリストをオブジェクトに定義された走査権データ・リストと照合して実行する。さらには、走査すべきオブジェクトを選択する際に、この選択方法に応じて走査IDリストを書き替えるようになる。

【0034】これにより、それらオブジェクトの走査時の走査手順を定義して、走査すべきオブジェクトの選択方法に応じて走査IDリストを定義し、またそれを走査時に動的に書き換えることもできるようになり、走査方法を柔軟に定義しつつ、オブジェクトの保護も容易に行なうことができる。又、複数のオブジェクトにより構成されたデータに対し、オブジェクト単位でのアクセス権利が定義できる。

【0035】

【実施例】以下、本発明の実施例を図面に従い説明する。

【0036】図1は第一の実施例の概略構成を示している。図において、1は仮想記憶管理部で、物理的な記憶装置への入出力を管理を容易にするための管理部である。

【0037】仮想記憶管理部1は、この装置の仮想記憶スペースに仮想的に格納されたオブジェクト11を管理する。但し、仮想的に格納されるオブジェクトを持つことは本発明に必須のことではなく、公知の記憶管理手段で代用しても良い。又、オブジェクト11は、リンク103で互いに関連づけたものであってもよい。

【0038】仮想記憶管理部1の管理するデータは、実際には物理メモリ2や二次記憶装置3に配置される。二次記憶装置3はハードディスク・ドライブやCD-ROM、ICカードなどである。

【0039】ここで、二次記憶装置3に配置される理由は、パーシステント・オブジェクトとして半永久的に記憶するのが目的である場合と物理メモリ2のスワップ領域として利用される場合などがある。また、オブジェクト情報を提携して持ち歩くために配置する場合もある。

【0040】物理メモリ2はさらにキャッシュ・メモリなどを用いて階層化することもできるが、それについては一般的な手法を用いることとし、本実施例では説明を省略する。

【0041】また、仮想記憶管理部1は、入出力デバイス4にも接続されている。入出力デバイス4は、例えば、利用するオブジェクトのデータを実際に計算機に記憶することなく、必要に応じて電話回線や無線、ネットワークなどを介してあたかも仮想記憶空間内に常駐するかのように扱うためのものである。これについては単一仮想記憶の管理に関する一般的な手法を用いることとし、本実施例では説明を省略する。

【0042】さらに、プログラム記憶部6に蓄えられているプログラムを解釈実行し、仮想記憶管理部1に管理されるオブジェクトデータを処理する実行主体であるプロセッサ5がある。プロセッサ5内には、命令の実行途中に一時的に利用するレジスタ類、現在実行している命令のプログラム記憶部6での位置を示すためのプログラムカウンタ、スタックデータの先頭位置を示すスタックポインタなどが存在し、これら一連のデータを入れ替えることによってプログラムを実行し、必要に応じて仮想記憶管理部1に対し、必要なデータの読み出しおよび書き込みを要求する。

【0043】アクセス要求受付部7はプロセッサ5からのアクセス要求を受け付ける。これはデータの読み出し、書き込み、消去などあらゆるデータ処理要求を含む。引数検出部8およびアクセス要求の種類判定部9は、アクセス要求受付部7の受け付けた要求のそれぞれ引数および種類を判定する。受け付けたアクセス要求がオブジェクト読みだしの要求である場合には、その要求をオブジェクト読み出し実行部10へ送る。その際、実行の引数は引数検出部8から送られる。

【0044】オブジェクト11は一般に仮想記憶空間に存在するものであり、仮想記憶管理部1によって仮想的に記憶されている。なお、このオブジェクト11は仮想記憶以外のいかなる記憶管理方法によって蓄えられていても、本特許の主旨は変わらない。

【0045】ここで言うオブジェクト11とはあらゆるデータの集まりのことであり、データ・ファイルやプログラムはもちろん、プログラムが一時的に利用する作業用の構造体なども含まれる。また、各オブジェクト11同士の中には、リンク103で関連付けられているものがある。

【0046】これにより複数のオブジェクト11をまとめて一つのオブジェクトとして扱うこともでき、これもオブジェクトと呼ぶ。特に、複数のオブジェクトを一つのオブジェクトとして考えるとき、それを構成するオブジェクトをサブ・オブジェクトと呼ぶこともできる。

【0047】走査IDリスト記憶部12は走査IDリストを管理している。走査IDリストは実行主体（スレッド）ごとにあってもよいし、複数の実行主体で同一の走査IDリストを共有していてもよい。同様にアクセス権記憶部13はユーザのアクセス権を管理している。

【0048】ここに記述した各構成要素は、それを管理する特定のサーバーを持たせるなどの方法を使えば、必ずしも仮想記憶空間1に置く必要はない。

【0049】オブジェクト読み出し実行部10においては、対象オブジェクト取出部14が、指定されたオブジェクトを仮想記憶管理部1から読み出すことを指定し、得られた走査権データ・リストを記憶部15に記憶し、また読み出したデータをオブジェクトとデータ読出部20に送る。

【0050】走査権データ・リスト検出部16は、走査権データ・リスト記憶部15のデータをもとに走査権データ・リストを取り出し、走査オブジェクト選択部19へ送る。

【0051】走査権IDリスト検出部17は、現在アクセスしているプロセスに関する走査IDリストを走査IDリスト記憶部12より取り出し、アクセス権検証部18は、現在アクセスしているプロセスにオブジェクトのアクセス権があるか否かをアクセス権記憶部13のデータを基に検証する。

【0052】走査オブジェクト選択部19は、アクセス権検証部18、走査IDリスト検出部17、走査権データ・リスト検出部16からの情報を基に、後に詳述する手順により、走査対象とするオブジェクトを適切に判断し、さらに必要となるオブジェクトの取り出しを対象オブジェクト取出部14に要求する。

【0053】オブジェクト・データ読出部20は、これら一連の流れの繰り返しによって得られたオブジェクト・データの必要部分を得、プロセッサ5に対し結果としてその内容を返す。

【0054】次に、図2は、複数のオブジェクトをリンクで関連付け、一つの意味のあるファイルとして扱った場合のオブジェクトの構成例を示している。

【0055】この場合、グループAの名簿本体のオブジェクト201には、名簿作成者のために用意されたコメントのオブジェクト202が付随している。そして、両者はリンク203で結合され、グループAの名簿を表現するオブジェクト204によって結ばれている。

【0056】したがって、これらのオブジェクト201、202、204およびリンク203の全部をあわせてグループA名簿205として捉えることができる。

【0057】一方、グループBの名簿にはコメントが付いていないので、グループBの名簿本体206がグループBの名簿そのものになっている。

【0058】さらに、グループCの名簿には、やはりコメントが付いていないので、グループBと同様の構造でも構わないが、図2の例では、グループCの名簿本体のオブジェクト207がリンク203でグループCの名簿を表現するノードであるオブジェクト208に結ばれている。そして、これらは、グループC名簿209として捉えることができ

る。

【0059】グループCの名簿209は、二つのオブジェクトを組み合わせる必要はなく、一つのオブジェクトとして表現してもよい。ここではコメントの追加などを行う可能性を考えて二つのオブジェクトで表現している。

【0060】そして、これらグループA名簿のオブジェクト204、グループB名簿のオブジェクト206およびグループC名簿のオブジェクト208は、それぞれリンク203によりグループ名簿オブジェクト210に結ばれ、これら全てのオブジェクトおよびリンクの集合をグループ名簿200として捉らえている。

【0061】この場合、以下の説明を簡単にするために、各オブジェクトにはユニークなID番号が付いているものとする。例えば、ID番号として、グループ名簿オブジェクト210をIDM、グループA、B、Cの名簿オブジェクト204、206、208のそれぞれをIDGA、IDGB、IDGC、グループA、Cの名簿本体オブジェクト201、207のそれぞれをIDGA0、IDGC0、グループAの名簿コメントオブジェクト202をIDGACとしている。

【0062】図3は、グループ名簿オブジェクト210の構造を示したものである。

【0063】ここで、301はオブジェクト210のID、302はこのオブジェクト210を作成した人のID番号である。このID番号は従来のファイル・システムのオーナーのIDに相当するが、このIDはユーザのIDには限られず、グループのIDでも構わない。また、プログラムまたはスレッドに番号を振り、その番号を用いても良い。

【0064】また、303はオブジェクトのアクセス権を持っているものを定義するアクセス権データ・リストである。これはデータ一つでもよいが一般にはリストとなっている。この例でのアクセス権データ・リスト303は、図4に示すように3つのアクセス権データ304、305、306のリストからなっている。

【0065】ここで、オブジェクトを作成した人のユーザIDを128番とし、その人には書き込みおよび読み出しの権利があるものとする。アクセス権データ304のフラグ

“R”及び“W”は、ユーザIDが128番の人は読み出しと書き込みはできることを意味している。但し、実行のためのアクセス権はまだないので他のアクセス権データで許可しない限りできない。同様にアクセス権データ305は、グループID320番から322番の人は読み出しの権利を持つことを示すフラグ“R”を有し、アクセス権データ306は、アクセス権の設定をユーザIDやグループIDではなく任意のプログラムで指定できることを示している。ここで指定したプログラムIDのプログラムを実行した結果が真(true)であれば、このプログラムが読み出しの権利を持つことを示す。

【0066】なお、プログラムあるいはその実行主体(スレッド)にもユーザIDと等価な番号を振ることにすれば、この例で左端に書いたIDまたはprogramという区別のフィールドは不要である。

【0067】一方、このオブジェクト210へのアクセスは、アクセス権データ・リスト303を構成する3つのアクセス権データ304、305、306のいずれかによって指定されている場合に認められる。この場合、アクセス権データの論理的な組み合わせにより、さらに複雑なアクセス権を指定することもできるが、ここでは最も単純な例のひとつを示している。

【0068】図3の307は走査制御データで、この走査制御データ307は、オブジェクト210を走査する際に、サブ・オブジェクトをどのように走査するかを規定するものである。ここに走査制御プログラムのプログラムIDを記入すれば一般にどのような走査制御も実現することができるが、ここでは説明を理解しやすくするために最も典型的な例を示す。ここで、EVERYTHINGというフラグは走査可能なサブ・オブジェクト全てをたどることを意味している。すなわち、走査の権利のある全てのサブ・オブジェクトを順次たどって読み出す。複数のグループに属している人がグループ名簿を読んだときには複数のグループ名簿が読めるようにするため、このフラグを設定している。そして、ひとつも走査可能なサブ・オブジェクトがなければ permission denied というエラーになる。これに対し、EVERYTHING-SILENT というフラグにしておけば、走査可能なサブ・オブジェクトが一つもなければあたかもグループ名簿は中身の空のファイルであるように見え、エラーにはならない。EVERYTHINGとEVERYTHING-SILENTの意味はエラー処理を除いて同一であるため、以下の実施例においては、これらを特に区別せずに用いる。又、EVERYTHINGを用いた場合に本来行わなければならないエラー処理部分についても説明を省略する。

・【0069】他の走査制御プログラムの例としては、ONLYONE フラグにしておく、走査可能なサブ・オブジェクトが一つも見つければそのオブジェクトを走査し、他のサブ・オブジェクトは無視する。ただし、走査可能なサブ・オブジェクトが一つもなければエラーになる。ONLYONE-SILENTフラグにするとエラーにはならない。

【0070】308はサブ・オブジェクトの数を示し、ここではサブ・オブジェクト数308は3個になっているので、オブジェクト210にはサブ・オブジェクトが3個あることがわかり、N○1～N○3のサブ・オブジェクトのIDがそれぞれサブ・オブジェクトID309に定義される。N○1～N○3のサブ・オブジェクトには、それぞれ走査権データ・リスト310が付けられている。ここでの走査権データ・リスト310は、いずれも要素は一つだけで、図5に示すように走査権データ311として、それぞれグループIDが320番、321番、322番の者に走査権を認めている。また、各走査権データ311に記述したR、W、Xの各フラグはそれぞれ読み出し、書き込み、実行に伴う走査が起こった場合の走査権を表示している。この例では全ての場合に走査権を認めている。

【0071】このような走査権データ・リスト310は、前に説明したアクセス権データ・リスト303と同様の手順で走査権の有無を調べるものであるが、機能は全く別個のものである。すなわち、アクセス権データ・リスト303は、そのオブジェクトのアクセスができるか否かを示すものであるのに対して、走査権データ・リスト310は、サブ・オブジェクトを自動的に走査すべきか否かを示すものである。例えば、あるサブ・オブジェクトにアクセス権はあるが走査権がない場合、サブ・オブジェクトの組み合わせで出来たオブジェクトを単純に走査すると、該当のサブ・オブジェクトは走査されない（見えない）が、サブ・オブジェクトを明示的に指定して読み出そうとした場合にはアクセス権があるので読み出すことができる。

【0072】なお、各走査権データ311は、グループのメンバに対してWの権利を出しており通常のアクセス権機構の感覚では違和感があるかもしれない。このWのフラグは単に、走査する際の手順を示したものであるから、オブジェクトの保護とは必ずしも関係がない。オブジェクトの保護は各オブジェクトのアクセス権リストで行うものである。仮に、このWフラグを悪用して、名簿のデータを書き込もうとしても実際にはサブ・オブジェクトに書き込みの権限が無いので、書き込むことはできない。この場合Wフラグがどちらになっても意味は無いと勘違いしやすいが、それは誤りである。Wフラグがもし無ければ、書き込みのために走査した際にサブ・オブジェクトに到達できないから、単に書き込みができないだけのことであるが、きちんとWフラグを立てておけば、書き込みのために走査した際にサブ・オブジェクトまで走査に行き、サブ・オブジェクトのアクセス権がチェックされ、適当なエラー処理が行われるのである。

【0073】次に、図6はグループAの名簿を表現するオブジェクト204の構造を示したものである。ここでは、グループ名簿オブジェクト210と同一部分には同符号を付して説明を省略する。グループB、Cの名簿のオブジェクト206、208も同様の構造であるので説明は省略する。

【0074】この場合、アクセス権データ・リスト303は、2つのアクセス権データ312を有している。これらは、それぞれ、ユーザIDが128番の人は読み出しと書き込みの権限があること、グループIDが320番の人は読み出しの権限があることを示している。

【0075】そして、2つあるN○1とN○2のサブ・オブジェクトのうち1番目のものはグループAの名簿コメント・オブジェクト202を示し、2番目のものはグループAの名簿本体オブジェクト201を示している。そして、それぞれのサブ・オブジェクト202、201は共に1つの走査権データを持つ走査権データ・リスト310を持っている。これら走査権データ・リスト310の走査権データ313は、それぞれユーザIDが128番の人およびグループIDが320番の人にかかるサブ・オブジェクトの走査権があることを示している。

【0076】次に、このようなオブジェクトを例として、オブジェクト走査が実施される手順を、いろいろな人がいろいろなオブジェクトを走査しながら読み出すことを例にして説明する。

【0077】まず、オブジェクト走査に先立ち、走査IDリスト記憶部12に走査IDリストを設定する。これはアクセス権が複数あるのと同様に、複数の走査IDを設定することができる。しかし、もともと自分に権利の無いIDについて走査権を持っていたとしても意味

の無いことが多いので、一般にはデフォルトとして現在自分の持っている全ての権利IDを用いることが多い。

【0078】例えば、ユーザID128番の人がグループID320番のグループとグループID340番のグループに属していたとすると図7のように3つの走査ID611をつないだ走査IDリストの初期値61を設定する。もちろん、明示的にこれ以外のデータを走査IDリストの初期値としても構わない。

【0079】この状態で、グループAの名簿オブジェクト204の走査を開始すると、図6において、まず、アクセス権データ・リスト303にユーザのID128が登録されているから、このオブジェクト204をアクセスすることができる。グループAの名簿オブジェクト204は2つのサブ・オブジェクトを持っているので、最初のサブ・オブジェクトを走査できるかどうか調べる。ここで、走査権データ・リスト310の走査権データ313を見ると、読み出しについてユーザID128に対し走査権があるから、走査IDリスト中の走査ID611とマッチし、走査権が認められ、走査する。

【0080】ここで、走査制御データ307はEVERYTHINGであるから、さらに次のサブ・オブジェクトも走査しようと試る。すると走査権データによるとグループID320に走査権があるから、同様に走査する。すなわち、この人にとっては、グループAの名簿オブジェクト204はコメントと本文のつながったものとして走査されることになる。

【0081】同じ操作を今度は同じグループに属するユーザID129の人が行ったとすると、走査IDリストの初期値61は、例えば図8のようになるから、グループAの名簿オブジェクト204を走査する際に、1番目のサブ・オブジェクトについては走査権データ・リスト310と走査IDリストがマッチせず、走査の対象にならない。すなわち、この人にとっては、グループAの名簿オブジェクト204は名簿の本文だけのものとして走査されることになる。

【0082】次に、同じ走査を今度はグループAに属さない人が行ったとすると、走査IDリストの初期値61で走査ID612は、例えば図9のようになるから、グループAの名簿オブジェクト204を走査する際に、1番目のサブ・オブジェクトは前回同様走査対象とならず、2番目のサブ・オブジェクトも走査の対象にならない。すなわち、この人にとっては、グループAの名簿オブジェクト204はまったく中身の無いファイルと同じものになる。

【0083】ここで、仮に、この人にも走査権を出すようにデータを定義しておけば、今度は走査をし、サブ・オブジェクトのアクセス権が出ていないとすると、そこでエラーとなる。このようにアクセス権と走査権の微妙な性質の違いを本発明のオブジェクト走査方法においては容易に実現することができる。

【0084】なお、以上説明した走査の手順は、各オブジェクトに対して再帰的に行うことが可能である。

【0085】すなわち、前の説明と同じユーザID128番の人がグループ名簿オブジェクト210をアクセスすると、まず図3のオブジェクト210のアクセス権データ・リスト303にはID128が書かれているからアクセス可能であり、3つのサブ・オブジェクトがあることがわかる。

【0086】次に、それぞれのサブ・オブジェクトが走査可能かどうか調べるが、それぞれの走査権データ・リスト310のうち、ユーザID128番の人が属しているグループID320番の書かれているサブ・オブジェクトN01だけが走査可能であることがわかるので、そのID#IDGAを持つサブ・オブジェクト204を次に走査する。以下前に説明した通りである。

【0087】このような手順により、ユーザID128番の人がグループ名簿オブジェクト210をアクセスすると、グループAの名簿のコメントと本文のつながったものとして走査される。

【0088】また、前と同じユーザID129番の人がグループ名簿オブジェクト210をアクセスすると、やはりその3つのサブ・オブジェクトのうちN01だけが走査可能であることがわかり、以下前に説明した通りの手順により、グループAの名簿本文として走査される。

【0089】ここで、上述のオブジェクト走査において、サブ・オブジェクトを選択する手順について図10により説明する。

【0090】即ち、あるユーザによってあるオブジェクトの走査が開始されたとすると、

まず、ステップ1101において、このオブジェクトに対するアクセス権がこのユーザにあるかどうか決定される。もしなければ、エラー処理がなされ、ある場合には、ステップ1102に進み、以下に説明するように操作IDリストの初期値61が設定される。

【0091】次に、このオブジェクトのサブ・オブジェクト数をステップ1103において可変数Nに読み込み、ステップ1104で数Nがゼロであるかどうか判断する。数Nがゼロであれば、走査するサブ・オブジェクトはなく、操作は終了する。そうでなければ、次のステップ1105において、N番目のサブ・オブジェクトがこのユーザにとって走査可能かどうかをN番目のサブ・オブジェクトの走査権データリストを検索して判断する。もしこのN番目のサブ・オブジェクトが走査可能でなければ、変数NをN-1に更新し、上記ステップ1104に戻る。

【0092】そうでない場合は、次にステップ1109において、このN番目のサブ・オブジェクトの走査を開始する。この時、このN番目のサブ・オブジェクトが更にサブ・オブジェクトを含んでいる場合は、このN番目のサブ・オブジェクトについて再帰的に呼び出しを行い、操作はスタートに戻って図10の操作をN番目のサブ・オブジェクトについて再帰的に繰り返される。

【0093】このN番目のサブ・オブジェクトの走査が終了した後、このオブジェクトデータの走査管理データ307がEVERYTHINGであるかどうかをステップ1111において決定する。もしそうでなければ、操作は終了し、EVERYTHINGであれば、操作は上記ステップ1106に戻る。

【0094】以上詳述したとおり、オブジェクトの持つ走査権データ・リストの情報を利用し、アクセスする際にどのオブジェクトを走査すれば良いかを判断し、適切に必要なオブジェクトだけを取り出す機構を用意したことにより、従来のアクセス権管理の方式では実現不可能であった柔軟なオブジェクト表現が可能となる。

【0095】また、個々のプログラムに特別な定義を行うことなく、オブジェクト自体に走査の手順を記述することができるので、オブジェクトの柔軟な表現を自然な形でプログラムから利用することができる。従って、近年盛んになった共同作業支援のシステムのように、アクセス権やユーザからの見え方などを複雑に記述しなければならない新しいタイプのプログラムの開発も非常に容易に行うことが可能となる。

【0096】又、第1の実施例においては、単一のオブジェクトを形成する異なったオブジェクトに異なったアクセス権を設定する際に、プログラムの操作が1つのスレッドから他のスレッドに変更され、プログラム操作がこのような不規則になることによってデバッグを行うのが難しくなるということから生じていた従来の問題を解決することができる。

【0097】故に、この実施例によれば、機能的に異なったアクセス権及び走査権を利用することによって、アプリケーションプログラムの操作とは関係なく、一般ユーザが作成した各オブジェクトについてオブジェクト走査手順及び走査権を柔軟に指定することが可能なオブジェクト走査装置を提供することができる。

【0098】以下、本発明の第2の実施例について説明する。

【0099】図11は第二の実施例の概略構成を図1と同一部分には同一符号を用いて示している。第一の実施例との違いは、オブジェクト読出し実行部10に、走査IDリストを動的に変更する走査IDリスト変更部21を持つことにある。

【0100】走査IDリスト変更部21は、走査オブジェクト選択部19と走査IDリスト記憶部12との間に接続されている。

【0101】走査IDリスト21は、走査するオブジェクトの選択に従い走査オブジェクト選択部19によって走査IDリスト記憶部12に格納されている走査IDリストに走査IDの追加および削除を行うものである。

【0102】走査オブジェクト選択部19は実施例1において説明した動作に加え、オブジェクトを走査する動作と同時に、走査したオブジェクトをいかなる権限でアクセスしたかを判断し、適切な権限によりアクセスすべく、走査IDリストの変更を行う。

【0103】以下、オブジェクトの走査の手順を図面を参照しながら、詳しく説明する。なお、特に断りの無い限り、第1実施例と同一の名称、符号は、同一の意味であるものとする。

【0104】グループ名簿オブジェクト210を走査する場合を考える。この場合、例えばユーザID128番の人がグループID320番のグループとグループID340番のグループに属していたとすると、第1実施例の場合と同様、図7のように3つの走査ID611を

つないだ走査IDリストの初期値61が設定される。

【0105】この場合、図3に示すように、グループ名簿オブジェクト210は3つのサブ・オブジェクトを持っている。そして、これら3つのサブ・オブジェクトの走査権データリスト310を見ると、1番目のサブ・オブジェクトだけが走査可能であることがわかる。したがって、サブ・オブジェクトであるグループAの名簿オブジェクト204だけを走査する。その際、1番目のサブ・オブジェクトに関しては走査権データリストにはIDが320であるユーザの権利しか定義されておらず、それ以外の走査IDは不要となるので、サブ・オブジェクトを走査する際にこれらの不要な走査IDは削除する。

【0106】したがって、図7のような初期値61を持っていた走査IDリストはこの時点で図12に示すように変更される。走査ID128、340は切り離されてID320のみが有効となる。

【0107】つまり、この走査は、1番目のサブ・オブジェクトはグループAの権限でアクセスされるべきであるという走査権データリスト310からの情報を利用して、本来持っていたユーザID128のアクセス権を行使せず、より弱いID320の権利を走査権として用いることを意味している。

【0108】次に、サブ・オブジェクトであるグループAの名簿オブジェクト204を走査する。

【0109】まず、1番目のサブ・オブジェクトの走査権データ・リスト310を見るとID128番の走査権313が定義されている。今走査を行っているユーザのIDは128番であるが、前述の変動によって現在の走査IDリストは図12の様にユーザID128番という走査IDを持っていないから、このサブ・オブジェクト201の走査は行われない。また、2番目のサブ・オブジェクトの走査権データ・リスト310はグループID320番に対して走査権を出しているから、走査が行われる。

【0110】このような手順により、ユーザID128番の人がグループ名簿オブジェクト210をアクセスすると、グループA名簿の本体オブジェクト201の内容と同一のデータが読み出され、ユーザに元来与えられている強い走査権に基づくオブジェクト202のコメントの内容の読み出しは行わない。

【0111】ユーザIDが128番のユーザにとってはグループA名簿の本体オブジェクト201もコメントオブジェクト202も共にアクセス権があるのであるから、従来技術においてはそのオブジェクトをアクセスするプログラムに特別の権限を与えない限り、常に両方のオブジェクトが参照されることになっていた。

【0112】他のユーザがグループ名簿オブジェクト210を走査した場合の動作は、容易に理解できるので説明を省略する。

【0113】ここで、上述のオブジェクト走査において、サブ・オブジェクトを選択し、また走査IDリストを変更する手順について図13のフローチャートを用いて説明する。

【0114】この場合、オブジェクト走査開始によりアクセス権があるかどうかを判断し（ステップ1001）、アクセス権がなければエラーとし、アクセス権があれば走査IDリストの初期値61を設定し（ステップ1002）、走査IDリストの先頭要素をポインタPに、サブ・オブジェクトの個数をNに設定する（ステップ1003、1004）。そして、N=0かどうかを判断し（ステップ1005）、N=0ならば、走査IDリストからPの走査IDを削除して（ステップ1009）、ステップ1010に進む。N≠0ならば、N番目のサブ・オブジェクトの走査権データリストをチェックしてN番目のサブ・オブジェクトはポインタPで示される走査IDで走査可能かどうかを調べる（ステップ1006）。ここで、走査不可ならばNをN-1にして（ステップ1007）、ステップ1005に戻り、走査可能ならば次の走査IDがあるか調べ（ステップ1010）、ある場合は次の要素をPに設定し（ステップ1008）、ステップ1004に戻り、なければステップ1011に進む。

【0115】サブオブジェクトの個数を再度Nに設定する（ステップ1011）。そして、N=0かどうかを判断し（ステップ1012）、YESならば処理を終了する。また、NOならばN番目のサブオブジェクトは走査可能かどうかを判断し（ステップ1013）、ここでNOならばNをN-1にして（ステップ1014）、ステップ1012に戻り、YESならば現在の走査IDリストをスタックにセーブし（ステップ1015）、N番目のサブオブジェクトのオブジェクト走査を開始する（ステップ1016）。この時N番目のサブ・オブジェクトはさらにサブ・オブジェクトがあれば、再帰的に呼び出しが行

われる。)サブ・オブジェクトの走査が全て終了したらスタックから走査IDを取り出す(ステップ1017)。そして、走査制御データ307がEVERYTHINGかどうかを判断し(ステップ1018)。ここでYESならばステップ1014に戻り、NOならば処理を終了する。

【0116】従って、図13により説明したオブジェクト走査においては、走査すべきオブジェクトを選択した際に、走査権のあるサブオブジェクトが1つも存在しない走査IDがあれば、それを削除するようになっている。

【0117】次に、走査すべきオブジェクトを選択した際に、選択したオブジェクトを走査することの不可能な走査IDがあれば、それを削除するようにする手順について図14により説明する。

【0118】この場合、オブジェクト走査開始によりアクセス権がユーザにあるかどうかを判断し(ステップ1101)、アクセス権がなければエラーとし、アクセス権があれば走査IDリストの初期値61を設定する(ステップ1102)。そして、サブ・オブジェクトの個数をNに設定し(ステップ1103)、 $N=0$ かどうかを判断し(ステップ1104)、YESならば処理を終了する。また、NOならばN番目のサブ・オブジェクトは走査可能かどうか判断し、NOならば、Nを $N-1$ にして(ステップ1106)、ステップ1104に戻り、YESならば、現在の走査IDリストをスタックにセーブし(ステップ1107)、N番目のサブオブジェクトの走査権データリストに含まれない走査IDを走査IDリストから削除し(ステップ1108)、N番目のサブオブジェクトのオブジェクト走査を開始し(ステップ1109)、この時、N番目のサブ・オブジェクトが更にサブ・オブジェクトを含んでいたらN番目のサブ・オブジェクトを再帰的に呼び出し、操作はスタートに戻り、図14のこの操作をN番目のサブ・オブジェクトについて再帰的に繰り返す。N番目のサブ・オブジェクトの走査が全て終了したらスタックから走査IDを取り出す(ステップ1110)。そして、このオブジェクトの走査制御データ307がEVERYTHINGフラグかどうかを判断し(ステップ1111)、ここでYESならばステップ1106に戻り、NOならば処理を終了する。

【0119】従って、このようにすれば、例えば、ユーザIDが128番のユーザが明示的にグループA名簿204を参照した場合には、ユーザID128番の権限を使って両方走査が行われ、単純にグループ名簿210を参照すると、サブ・オブジェクト204を走査した時点で走査権が変更され、グループ320の権限で名簿ファイルをアクセスすることができるようになる。

【0120】これにより、例えば、自分の編集した名簿ファイルを名簿整形プログラムに与えて実行した時に、他のメンバは正しく動作するのに、自分だけはコメントの入った誤ったデータを使って実行するために正しく動作しないというようなことを防止できる。

【0121】つまり、このような構成によれば、複数のオブジェクトにより構成されたデータに対し、オブジェクト単位でアクセス権を定義できるのみならず、それらオブジェクトの走査時の走査手順を定義し、また走査すべきオブジェクトを選択するとともに、この選択に従って走査IDリストを書き換えることができるので、走査方法を柔軟に定義しつつ、オブジェクトの保護も容易に行うことができる。

【0122】すなわち、複数のオブジェクトを一つのファイルとみなして走査する際に、各单位オブジェクトごとに記述された走査権を別個に組み合わせるのではなく、全オブジェクトを統一した権利の基で走査するので、オブジェクトの作成者が意図しなかった走査方法が採用されることを未然に防ぐことができる。

【0123】また、走査すべきオブジェクトを選択した際に、特定の走査IDリストに変更し、または特定の一つまたは複数の走査IDを走査IDリストに追加し、また、特定の走査IDを走査IDリストから削除する走査IDリスト変更部を備えることにより、特定のオブジェクト走査時に限り特別の方法を与えることができる。又、ここで説明する電子メールとは、従来から用いられている方式のものに限らず、ある人からある人へ伝達されるあらゆるデータ一般、例えば電子会議において相手に示す提示資料、会議風景のビデオ・データ等も指す。

【0124】また、走査IDリストの変更時にはオブジェクトの作成者が持っていない走査IDを追加することのできないという制約を設けることにより、従来一般に用いられていたアクセス権管理を廃止し、走査権で代用することとしても、前記特別の走査IDを与える機構を悪用してオブジェクトを不正にアクセスすることを防ぐことができる。

【0125】また、走査IDリストの変更時に、特定の実行プログラムによって指定された走査IDを追加することを認めるように前記制約を緩和すると、より柔軟なオブジェクトの走査を可能にすることができる。

【0126】以下に、図15を参照して本発明の第3の実施例を説明する。この実施例はオブジェクト走査方法及び装置を電子メール作成ツールに応用したものである。

【0127】図15は、電子メール作成ツールを組み込んだオブジェクト走査装置の構成図である。図11と重複する部分は同一とし、説明を省略する。

【0128】まず、電子メール作成部30について説明する。これは、電子メールを作成、送信しようとするものが利用する部分である。まず、文書入力部31によって入力された文章はオブジェクト作成部32に送られ、文章オブジェクトの形で格納するよう仮想記憶管理部1に要求する。ここでいう文章とは、文字で書かれた通常文章はもちろん、画像、音声、データベース等の一般データであって良い。

【0129】文章作成者は必要に応じて作成した文章の一部または全部に走査権を設定することができる。この指示は文章入力部31から走査権データ・リスト作成部33に対して送られ、走査権データ・リスト作成部33は指示に応じて作成文書に対し走査権データ・リストを設定する。この処理にはオブジェクトのサブ・オブジェクトへの分割が伴う場合が一般的である。設定すべき内容はオブジェクト作成部22に対し適宜要求される。

【0130】以上は電子メールの作成までの説明であり、このようにして作成された文章は適当な走査権およびアクセス権を設定することによりそのまま他の人から参照することができるが、ここでは、従来からある電子メールのように送信相手に対し送り付けるといった概念の元に説明する。ただし、本発明はこの概念に限定されたものではない。

【0131】オブジェクト送信部34は作成された文章オブジェクトを送信する。具体的には、送信相手のメール・アドレス50（宛名）に従って、受信者が読めるようにメールの送信の連絡をしたり、必要であれば他の計算機に対しネットワークを介してデータ転送を行う。これら処理は従来から Sendmail と呼ばれるプログラムで行われており、詳しい説明は省略する。

【0132】次に、電子メール受信部40について説明する。これは電子メールを受信し読む者が利用する部分である。まず、オブジェクト受信部41が電子メールの到着を検出する。これは電子メール読出部42に伝えられ、読出部42はプロセッサ5に対し文章オブジェクトの読み出しを命ずる。読み出しを命じてから結果として文章オブジェクトを読み出すまでの手順は実施例2で説明したものと同一である。

【0133】電子メールを受信する者は、受け取った電子メールを読み出す際の走査IDリストを走査IDリスト指示部43に指令し、指令された内容を走査IDリスト設定部44に伝え、走査IDリスト設定部44は走査IDリスト記憶部12に情報を書き込む。ここで言う走査IDリストとは「誰の権限でこのメールを読むべきか」というものを指定するものであり、もちろん、アクセス権の無いメール（例えば他人宛てのメール）が読めるという意味ではない。

【0134】しかし、一般に電子メールを読む度にユーザに対してこの指令を走査IDリスト指示部43に送る操作は煩雑であるため、デフォルト走査IDリスト指示部45は、状況に応じてデフォルトの走査IDリストを決め、走査IDリスト設定部44にその内容を伝える。デフォルトの走査IDリストを決める方法は一般にユーザがカスタマイズできるように環境設定ファイル等で指示するが、ここでは、オブジェクト送信部34からオブジェクト受信部41に指示されたメール・アドレス50を元に決定するものとする。例えば、その電子メールがグループA全員に宛てられたものであれば、デフォルトの走査IDリストはグループAのみの走査権とする。受信者は、必要や好みに応じてこのデフォルトの変更を走査IDリスト指示部43に対して行うことができる。

【0135】電子メール読出部42がプロセッサ5から受け取った結果は電子メール表示部46に送られ、実際にその電子メールを読むことができる。また、電子メールを読んだ後に、走査IDリスト指示部43に走査IDリストの変更を要求し、異なる走査権の元で再度電子メールを読むことも可能である。

【0136】以上のような構成による電子メールの送受信に関して、実際に使われる文書を例として具体的にデータ構造がどのように表現されるのかを以下に例示する。

【0137】図16は、リーダーがプロジェクトAおよびプロジェクトBの今期の計画を記した文書を作成したものである。この文書作成ツールは文書の一部に走査権設定の属性を

付ける機能を持っている。

【0138】通常、文書には特に走査権を指定する必要は無いが、リーダはこの文書が機密性が高いと判断し、各プロジェクトの計画はそれぞれのメンバ以外からは解らないようにしようとした。

【0139】範囲を指定し、前半はプロジェクトA、後半はプロジェクトBに対し走査権を与える指示を行った。また、プロジェクトAに関する文章の一部は、個人のプライバシーに触れるため、自分だけが走査するように指示した。

【0140】文書作成ツールはこの指示を受け、適切な走査権データ・リストの設定を行い、図17に示すようなデータ構造を作成した。なお、図17において、本実施例の説明に直接必要とならないフィールドは記述を省略した。

【0141】リーダはプロジェクトAとプロジェクトBの両方の権限を持っているので、作成した文書全体を読むことができる。

【0142】リーダは、プロジェクトAおよびプロジェクトBのメンバ全員にこの文書を電子メールで送信した。この文書オブジェクトの走査は、電子メール作成ツールが送信する際に行われても良いし、電子メール作成ツールは文書全体を送信し、受信者が読み出す際にオブジェクトの走査が行われても良いし、その両方でも良いが、ここでは、受信者が読み出す際に行われるものとして記述する。

【0143】プロジェクトAのみに属しているメンバ1がこの電子メールを受信すると、まず、アクセスに先立ち、図18のように走査IDリストにプロジェクトAと自分のIDである MEMBER1を設定する（実際には他のIDも設定されるかもしれないが、この部分の説明には関係ないので省略する）。そしてこの文書オブジェクトを走査すると、プロジェクトAの権限に基づいて走査するため、実施例1の場合と同様の手順にて、文書1の部分だけがアクセスされる。従って、その人には文書1の内容だけが読める。

【0144】一方、プロジェクトBのみに属しているメンバは同様に文書3の内容だけが読める。

【0145】また、プロジェクトAとプロジェクトBの両方に属しているメンバ2がこの電子メールを受信すると、まず図19のように走査IDリストにプロジェクトAとプロジェクトBと自分のIDである MEMBER2を設定する。そしてこの文書オブジェクトを走査すると、プロジェクトAとプロジェクトBの両方の権限で走査し、また、TRAVARSE INFOが EVERYTHING であるため、文書1の内容と文書3の内容の連結したひとつの文書として読める。

【0146】このような手順によって電子メールの送信を行うことによるメリットは次に示すとおり、いくつもある。

【0147】まず、本来ひとつの文書として管理していたものを、メールの送信を行うために必要部分だけを切り貼りして相手に応じて送る必要がなくなる。一旦走査権データ・リストの設定された文書は、誰に送信しても読ませる必要のある部分だけが読めるように自動的に送られる。

【0148】また、誤って読まれてはいけない部分を送ってしまう心配もなくなる。例えば送信する文書の中に文書2のような個人のプライバシーに関するコメントの記述された部分が含まれていても、受信者がそれを読んでしまうということは起こらない。あるいは、プロジェクトAにもプロジェクトBにも属さない人に送ってしまうと、中身の何もないメールとなるだけである。

【0149】さらには、プロジェクトAとプロジェクトBの両方に属した人が受信した場合、従来の方法においては、プロジェクトAに関する部分はプロジェクトAのメンバに、プロジェクトBに関する部分はプロジェクトBのメンバにそれぞれ送信する必要があるため、元々ひとつの文章であった文書1と文書3とが別々の電子メールとなってしまい、整理に不便である。

【0150】さらに、近年実用化が試みられている、単一仮想記憶というオペレーティング・システムにおいては、分散計算機環境において同一のオブジェクトを単一（同一の）アドレスでアクセスすることができるが、そういったシステムでは電子メールの送信は単にそのオブジェクトへのポインタを渡すだけで済む。そのような環境において、従来の電子メールの送信方法では送り先の相手ごとにデータを加工し直すことになるので、余分な記憶領域を必要とすることになるが、本発明ではそのような場合にもやはりそのオブジェクトへのポインタを渡すだけで済むので、性能が飛躍的に向上する。

【0151】このように、本発明のオブジェクト走査装置を用いれば、ここで説明したような非常に単純な例でさえ、その有益性は明らかである。こういったオブジェクトをさらに組み合わせた複雑なオブジェクトの走査をも自動的に行うことができるのであるから、その効果は計り知れない。

【0152】なお、本発明は上記実施例のみに限定されず、要旨を変更しない範囲で適宜変形して実施できる。

【0153】

【発明の効果】本発明によれば、複数のオブジェクトにより構成されたデータに対し、オブジェクト単位でアクセス権が定義できるのみならず、それらオブジェクトの走査時の走査手順を定義することにより、柔軟に走査を行い、オブジェクトの保護を容易に行うことができる。さらには、サブオブジェクトの選択に応じて走査IDリストを書き換えることにより、走査権を動的に変更し、一層柔軟なアクセスを行うことができる。

【0154】このことから、複数のオブジェクトを組み合わせてオブジェクトを構成する場合、個々のオブジェクト毎に異なるアクセス権を設定できるようにする試みが近年行われているが、この試みの副作用として、プログラムを実行する主体によってプログラムの動作が変化してしまい、動作が一貫しないためにデバッグが非常に困難になるといった問題が生じて来ているが、アクセス権とは別に走査権を定義することができるので、このような問題点も解決できる。

【図面の簡単な説明】

【図1】本発明に係るオブジェクト走査装置の第1の実施例の概略構成を示す図。

【図2】図1の装置で取り扱うオブジェクトの構成例を示す図。

【図3】図2のオブジェクトのデータの構成例を示す図。

【図4】図3の構成におけるアクセス権データの構成例を示す図。

【図5】図3のデータの構成における走査権データの構成例を示す図。

【図6】図3に示されるオブジェクトのサブ・オブジェクトの構成例を示す図。

【図7】図1の装置における128番のユーザの走査IDリストの初期値を示す図。

【図8】図1の装置における129番のユーザの走査IDリストの初期値を示す図。

【図9】図1の装置における130番のユーザの走査IDリストの初期値を示す図。

【図10】図1の装置におけるサブ・オブジェクト選択の手順の一例を説明するためのフローチャート。

【図11】本発明に係るオブジェクト走査装置の第2の実施例の概略構成を示す図。

【図12】図11の装置におけるユーザの走査IDリストの変更後を示す図。

【図13】図11の装置で走査サブ・オブジェクトの選択及び走査IDリスト変更を行う操作のフローチャート。

【図14】図11の装置で走査サブ・オブジェクトの選択及び走査IDリスト変更を行う操作の他の場合におけるフローチャート。

【図15】電子メールツールを組み込んだ本発明に係るオブジェクト走査装置の第3の実施例を示す概略構成図。

【図16】図15の装置で扱うメール文書の一例を示す図。

【図17】図15の装置において図16の文書を表示するオブジェクトのデータ構成を示す図。

【図18】図15の装置におけるユーザの走査IDリストの初期値を示す図。

【図19】図15の装置における他のユーザの走査IDリストの初期値を示す図。

【図20】従来の名簿本体のオブジェクトを説明する図。

【図21】従来の名簿本体のIDリストを説明する図。

【図22】図20の名簿のコメントのオブジェクトを説明する図。

【図23】図20及び22のオブジェクトに関連した複数のオブジェクトを連結したオブジェクトを説明する図。

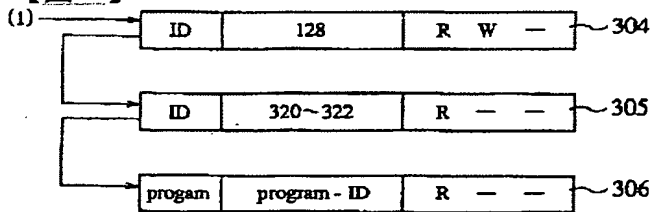
【図24】従来の名簿整形用のプログラムを登録するオブジェクトの一般的な図。

【図25】従来のプログラム登録用のオブジェクトの変形例を示す図。

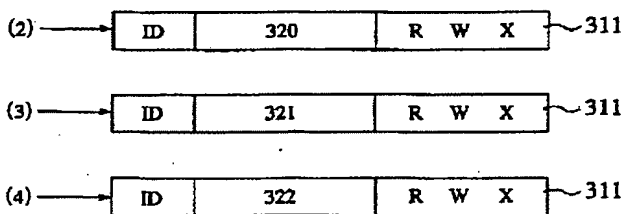
【符号の説明】

- 1 仮想記憶管理部
 - 2 物理メモリ
 - 3 二次記憶装置
 - 4 入出力デバイス
 - 5 プロセッサ
 - 6 プログラム記憶部
 - 7 アクセス要求受付部
 - 8 引数検出部
 - 9 アクセス要求種類判定部
 - 10 オブジェクト読出し実行部
 - 11 オブジェクト
 - 12 走査IDリスト記憶部
 - 13 アクセス権記憶部
 - 14 対象オブジェクト取出部
 - 15 走査権データ・リスト記憶部
 - 16 走査権データ・リスト検出部
 - 17 走査IDリスト検出部
 - 18 アクセス権検証部
 - 19 走査オブジェクト選択部
 - 20 オブジェクトデータ読出部
- 103 リンク

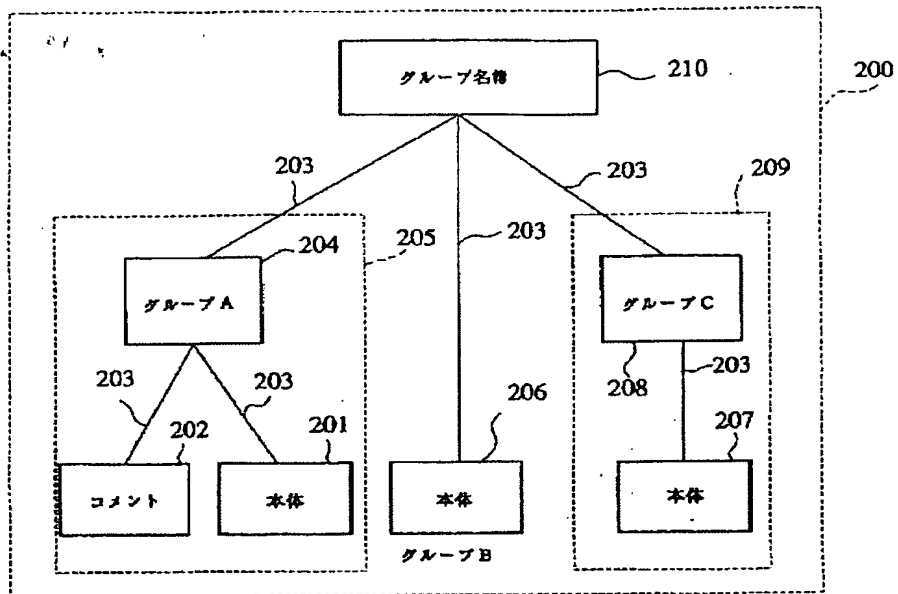
【図4】



【図5】



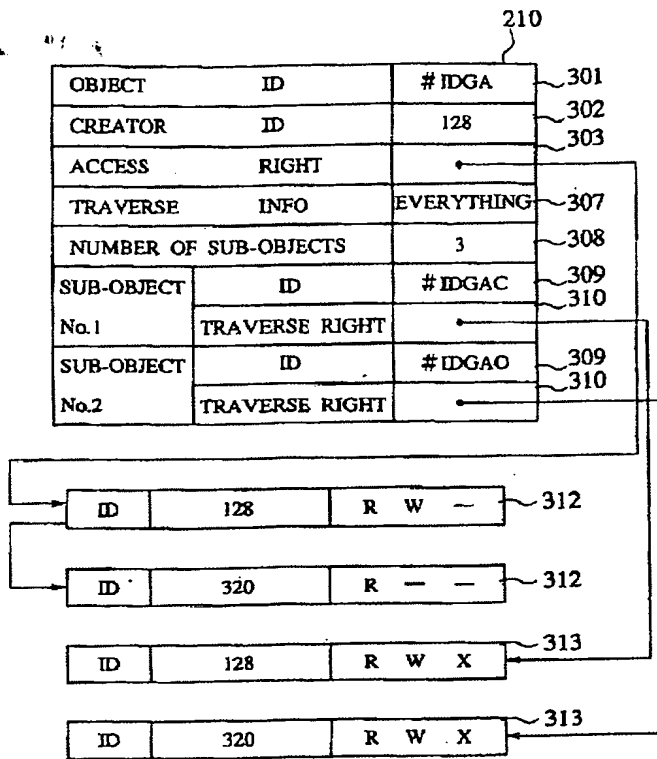
【図1】



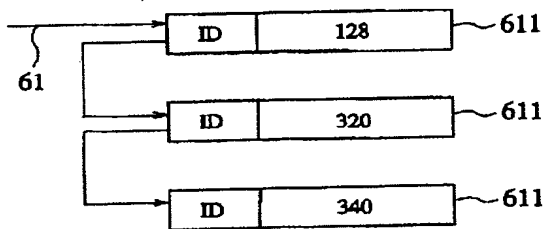
【図 3】

210			
OBJECT	ID	#IDM	301
CREATOR	ID	128	302
ACCESS RIGHT		●	303 (1)
TRAVERSE INFO		EVERYTHING	307
NUMBER OF SUB-OBJECTS		3	308
SUB-OBJECT	ID	#IDGA	309
No.1	TRAVERSE RIGHT	●	310 (2)
SUB-OBJECT	ID	#IDGB	309
No.2	TRAVERSE RIGHT	●	310 (3)
SUB-OBJECT	ID	#IDGC	309
No.3	TRAVERSE RIGHT	●	310 (4)

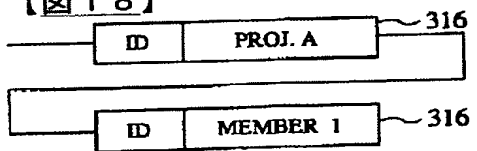
【図 6】



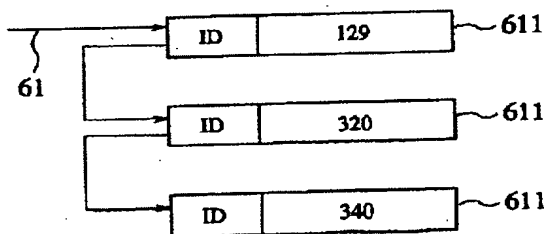
【図 7】



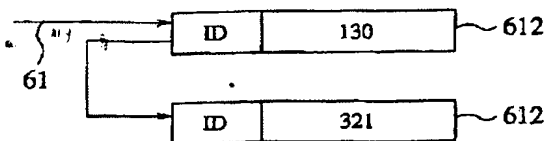
【図 18】



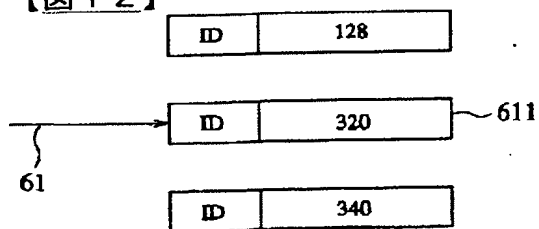
【図 8】



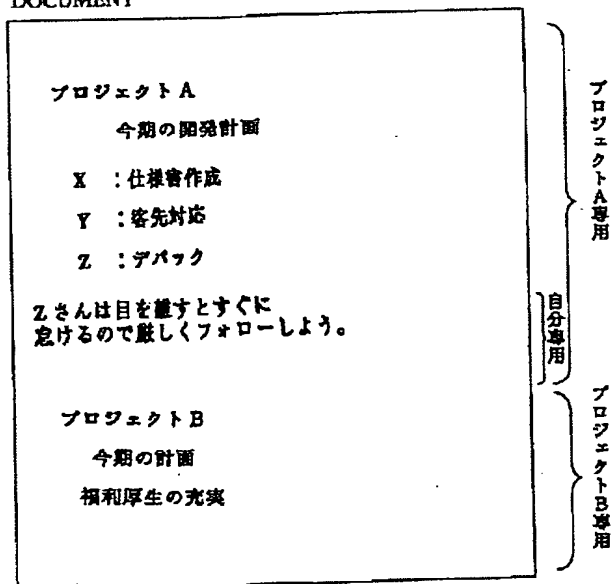
【図 9】



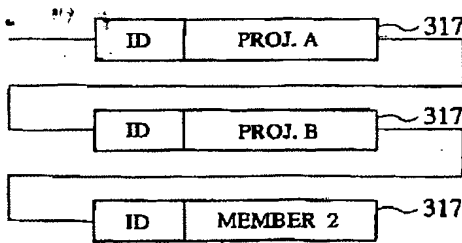
【図 1 2】



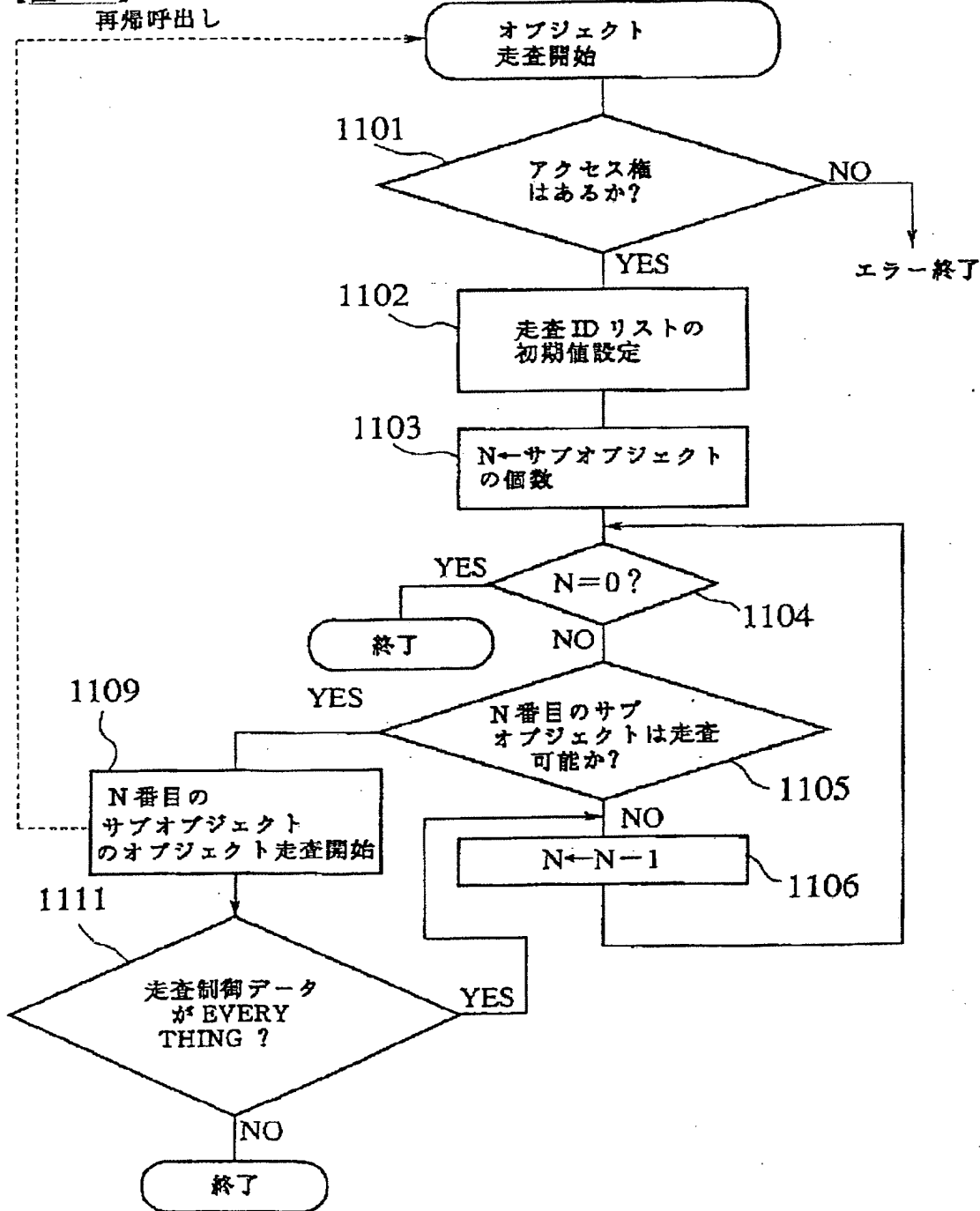
【図 1 6】
DOCUMENT



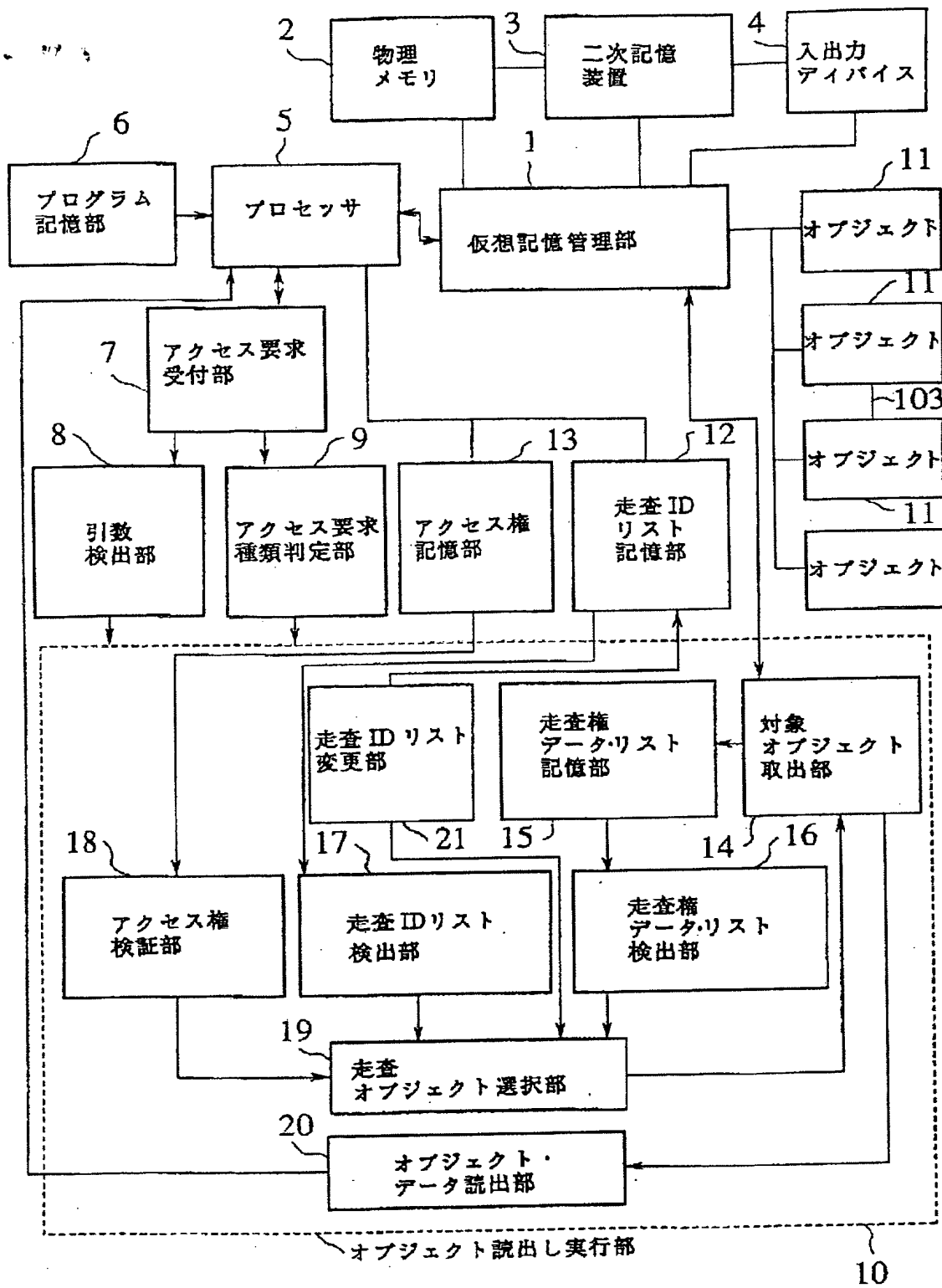
【図 1 9】



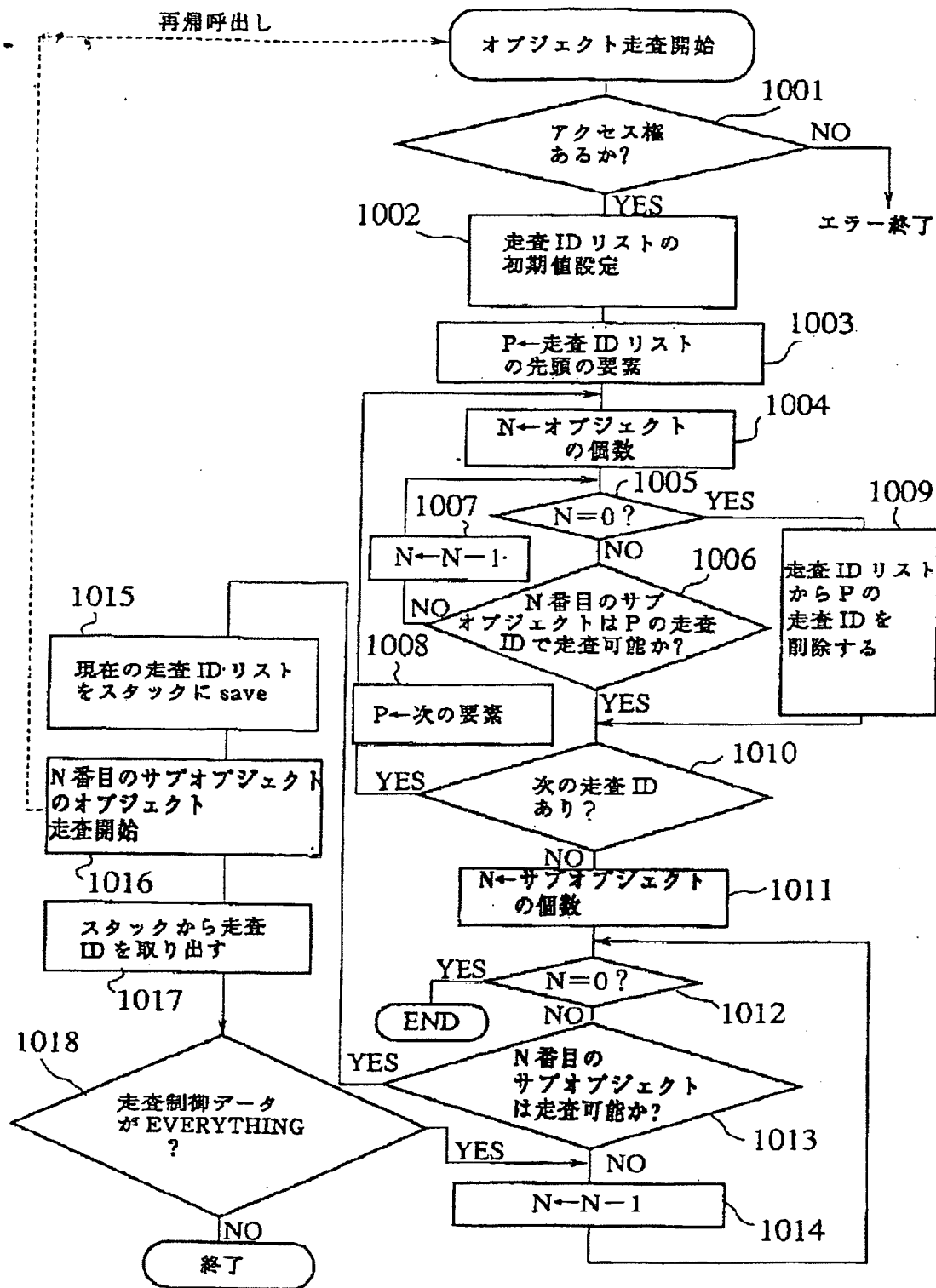
【図 10】



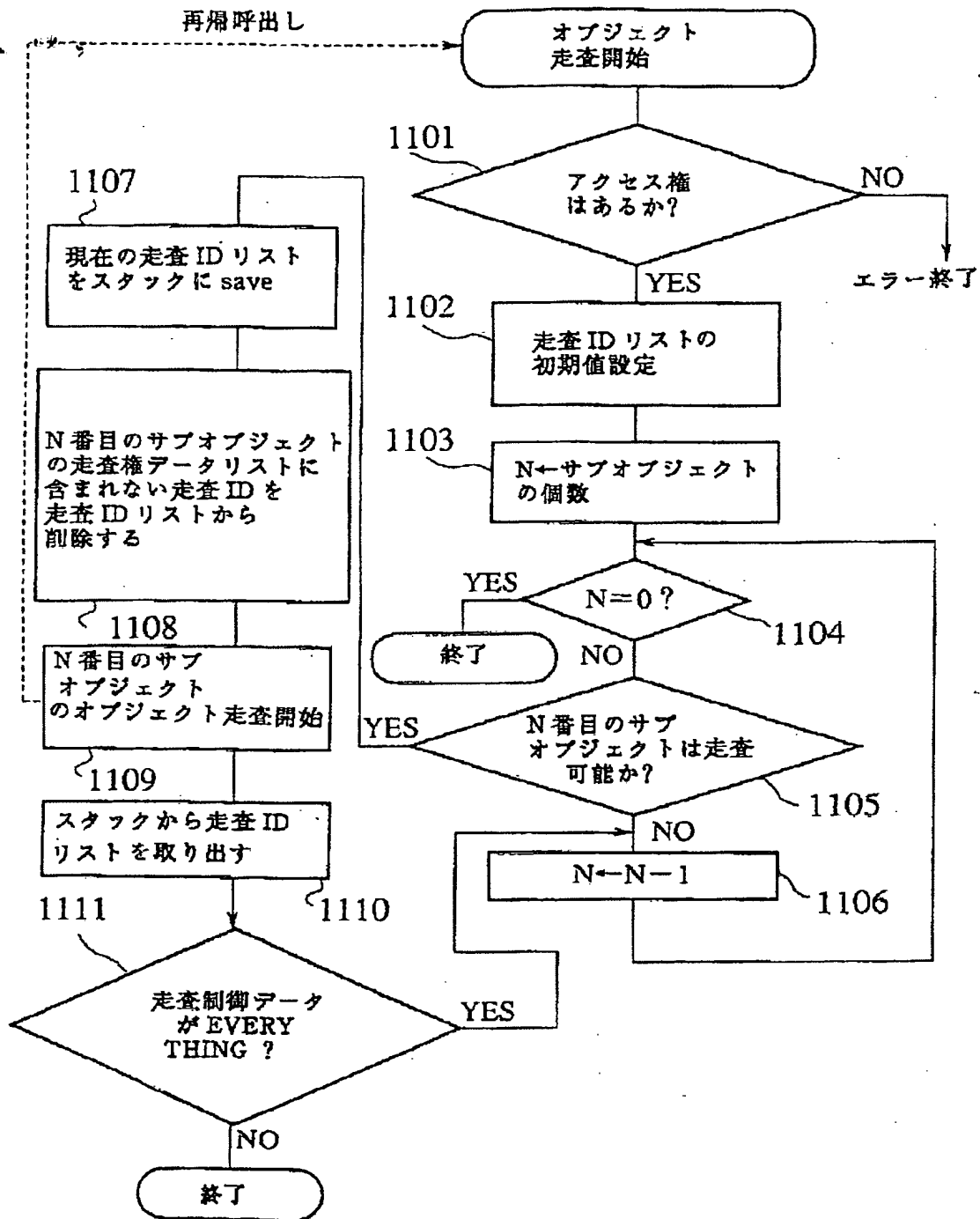
【図 11】



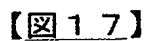
【図13】

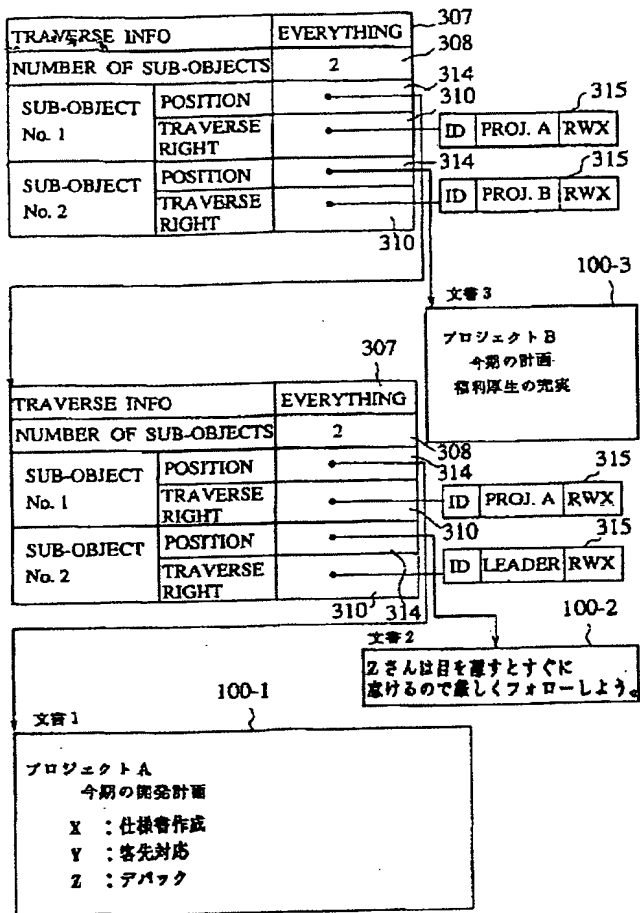


【図 14】



【図 15】





【図 2 0】

1211	OWNER	ID	128
1212	GROUP	ID	320
1213	OWNER		
	RD	WR	EX
	1	1	0
1214	GROUP		
	RD	WR	EX
	1	0	0
1215	OTHER		
	RD	WR	EX
	0	0	0
1216	伊藤	〒345	0123-45-6789
	佐藤	〒456	03-9312-1234
	高橋	〒567	045-356-0012

【図 2 1】

131

1311	NAME	PASSWORD	SHELL	ID
	高橋	ACZKL 2	lbin 1 csh	128
	佐藤	MNOQ 14	lbin 1 sh	129
	⋮	⋮	⋮	⋮
1312	グループA	—	—	320
1313	グループB	—	—	321
	グループC	—	—	322

【図22】

1411	OWNER	ID	128
1412	GROUP	ID	320
1413	OWNER		
	RD	WR	EX
	1	1	0
1414	GROUP		
	RD	WR	EX
	0	0	0
1415	OTHER		
	RD	WR	EX
	0	0	0
1416	佐藤さんは8月に転居予定なので、その頃 再確認すること。 伊藤さんの直通電話は0123-45-6800 だが内線なので名簿には記載しない		

【図23】

1511	OBJECT NAME	"グループA名称"	
1512	OWNER	ID	128
1513	GROUP	ID	320
1514	RD	WR	EX
	1	0	0
1515	RD	WR	EX
	1	0	0
1516	RD	WR	EX
	1	0	0
1517	SUB - OBJECT	1	NUMBER OF SUB-OBJECTS 2
1519	SUB - OBJECT	No. 1-ID	50
1520	SUB - OBJECT	No. 2-ID	52

【図24】

1611	SET ACCESS	ID	—
1613	OWNER	ID	129
1614	GROUP	ID	320
1615	OWNER		
	RD	WR	EX
	1	1	1
1616	GROUP		
	RD	WR	EX
	0	0	1
1617	OTHER		
	RD	WR	EX
	0	0	0
1618	実行プログラム		

【図25】

1612	SET ACCESS	ID	320
1613	OWNER	ID	129
1614	GROUP	ID	320
OWNER			
1615	RD	WR	EX
	1	1	1
GROUP			
1616	RD	WR	EX
	0	0	1
OTHER			
1617	RD	WR	EX
	0	0	0
1619	実行プログラム		

PATENT ABSTRACTS OF JAPAN

(B)

(11)Publication number : 05-298174

(43)Date of publication of application : 12.11.1993

(51)Int.Cl.

G06F 12/00

G06F 12/00

G06F 13/00

(21)Application number : 04-101355

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 21.04.1992

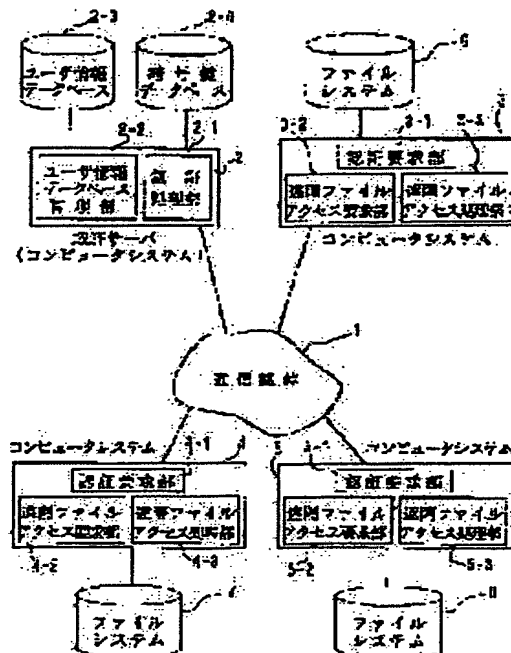
(72)Inventor : NUKUI HARUMI

(54) REMOTE FILE ACCESS SYSTEM

(57)Abstract:

PURPOSE: To improve the justification and safety of a file access.

CONSTITUTION: A remote file access request part 3-2 of a computer system 3 to be a remote file access request origin ciphers the authenticated certificate and the access request list acquired by the authentication server (computer system) 2 by a self-decoding key, prepares an access request packet by adding user identification information to the ciphered information, further ciphers it by the ciphering key of an access request destination computer system, and performs the access request of a remote file. A remote file access processing part 4-3 of a computer system 4 to be a remote file access request destination decodes the ciphered access request packet by the self-decoding key, acquires the ciphering key of a user from the authentication server 2 based on the user identification information obtained by further decoding, decodes the authenticated certificate and the access request list and performs the authentication of the user and the processing for the acceptance and rejection of the remote file access.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平 5 - 2 9 8 1 7 4

(43) 公開日 平成 5 年 (1 9 9 3) 1 1 月 1 2 日

(51) Int. Cl. ⁵	識別記号	庁内整理番号	F 1	技術表示箇所
G06F 12/00	545	F 8526-5B		
	537	H 8526-5B		
13/00	351	E 7368-5B		

審査請求 未請求 請求項の数 1 (全 6 頁)

(21) 出願番号 特願平 4 - 1 0 1 3 5 5
(22) 出願日 平成 4 年 (1 9 9 2) 4 月 2 1 日

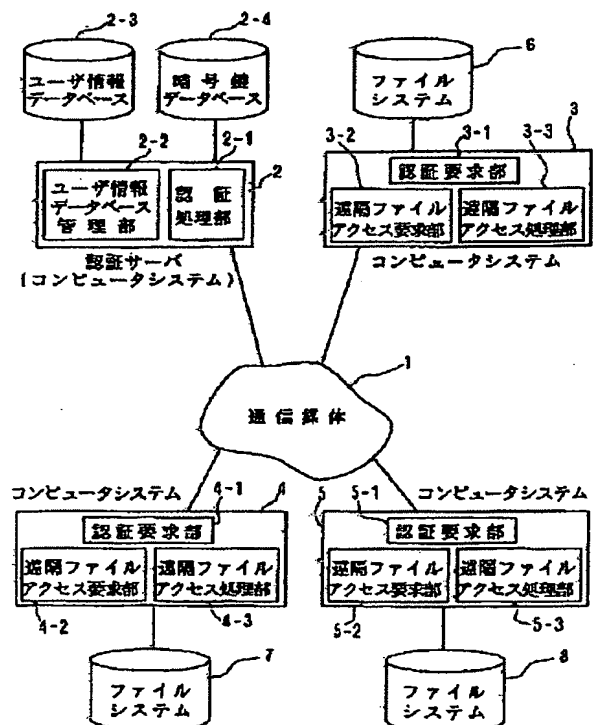
(71) 出願人 0 0 0 0 0 3 0 7 8
株式会社東芝
神奈川県川崎市幸区堀川町 7 2 番地
(72) 発明者 眞井 春美
神奈川県川崎市幸区柳町 7 0 株式会社東芝
芝柳町工場内
(74) 代理人 弁理士 木村 高久

(54) 【発明の名称】 遠隔ファイルアクセスシステム

(57) 【要約】

【目的】 ファイルアクセスの正当性と安全性を向上する。

【構成】 遠隔ファイルアクセス要求元のコンピュータシステム 3 の遠隔ファイルアクセス要求部 3 - 2 は、認証サーバ 2 から取得した認証済証明書とアクセス要求リストを自己の復号鍵で暗号化し、該暗号化された情報にユーザ識別情報を付加してアクセス要求パケットを作成し、さらにアクセス要求先コンピュータシステムの暗号鍵で暗号化し、遠隔ファイルのアクセス要求を行い、遠隔ファイルアクセス要求先のコンピュータシステム 4 の遠隔ファイルアクセス処理部 4 - 3 は、暗号化されたアクセス要求パケットを自己の復号鍵で復号化し、さらに復号して得られたユーザ識別情報をもとに認証サーバ 2 からユーザの暗号鍵を取得し、認証済証明書およびアクセス要求リストを復号し、ユーザの認証および遠隔ファイルアクセス許可の処理を行う。



【特許請求の範囲】

【請求項 1】複数のコンピュータシステムが任意の通信媒体を介して接続されたネットワーク上で、特定のコンピュータシステムが利用者情報を一括管理して発行する認証済証明書に基づいて、前記複数のコンピュータシステムに接続される遠隔ファイルシステムのファイルアクセスを行う遠隔ファイルアクセスシステムにおいて、遠隔ファイルアクセス要求元のコンピュータシステムは、前記認証済証明書とファイルアクセス要求リストを利用者固有の利用者識別情報をもとに暗号化し、該暗号化された情報と利用者識別情報をさらに前記特定のコンピュータシステムから取得したアクセス要求先コンピュータシステムの情報をもとに暗号化し、該暗号化された情報により前記遠隔ファイルのアクセス要求を行う遠隔ファイルアクセス要求手段を有し、遠隔ファイルアクセス要求先のコンピュータシステムは、前記遠隔ファイルのアクセス要求を自コンピュータシステムがもつ情報により復号して利用者識別情報を得て、さらに該利用者識別情報をもとに暗号化されている前記認証済証明書とファイルアクセス要求リストを復号して利用者の認証および遠隔ファイルアクセス許可の処理を行う遠隔ファイルアクセス処理手段を有して、相互に認証を行うことを特徴とする遠隔ファイルアクセスシステム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、任意の通信媒体を介して接続された複数のコンピュータシステムから構成されるネットワークシステム上の遠隔ファイルシステムをアクセスする遠隔ファイルアクセスシステムに関する。

【0002】

【従来の技術】従来、複数のコンピュータシステムが任意に通信媒体を介して接続された分散環境においては、特定のコンピュータである認証サーバにより利用開始手続が行われ、正しいユーザと判断された時に、認証済証明書が発行され、さらに、その後この認証済証明書を再度認証サーバに提出し、サービス利用のための証明書を得ることによって利用可能になるという手続が行われていた。

【0003】これに対し、遠隔ノードのコンピュータシステムに存在する遠隔ファイルシステムをアクセスすることができる遠隔ファイルアクセスシステムにおいては、利用者識別子とファイル保護情報の組合せにより、遠隔ファイルアクセスの可否を判断するという単一ノード内と同様な一方的ファイルアクセス形態をとっていた。

【0004】このため、遠隔ノードのコンピュータシステムからファイルアクセスを要求した利用者が、本来の利用者識別子を所有する利用者であるという保証はなく、不正にアクセスされる可能性があるという問題点が

あった。

【0005】

【発明が解決しようとする課題】前述したように、複数のコンピュータシステムが任意の通信媒体を介して接続されたネットワーク上で、特定のコンピュータシステムが利用者情報を一括管理して発行する認証済証明書に基づいて、前記コンピュータシステムに接続される遠隔ファイルシステムのファイルアクセスを行う遠隔ファイルアクセスシステムにおいて、前記認証済証明書による要求元から要求先への一方向の認証であったため、遠隔のコンピュータシステムからファイルアクセスを要求した利用者が、本来の利用者識別子を所有する利用者であるという保証はなく、不正にアクセスされる可能性があるという問題点があった。

【0006】そこで、本発明は、かかる問題点を除去し、ファイルアクセスの正当性と安全性を向上した遠隔ファイルアクセスシステムを提供することを目的とする。

【0007】

【課題を解決するための手段】本発明は、複数のコンピュータシステムが任意の通信媒体を介して接続されたネットワーク上で、特定のコンピュータシステムが利用者情報を一括管理して発行する認証済証明書に基づいて、前記複数のコンピュータシステムに接続される遠隔ファイルシステムのファイルアクセスを行う遠隔ファイルアクセスシステムにおいて、遠隔ファイルアクセス要求元のコンピュータシステムは、前記認証済証明書とファイルアクセス要求リストを利用者固有の利用者識別情報をもとに暗号化し、該暗号化された情報と利用者識別情報をさらに前記特定のコンピュータシステムから取得したアクセス要求先コンピュータシステムの情報をもとに暗号化し、該暗号化された情報により前記遠隔ファイルのアクセス要求を行う遠隔ファイルアクセス要求手段を有し、遠隔ファイルアクセス要求先のコンピュータシステムは、前記遠隔ファイルのアクセス要求を自コンピュータシステムがもつ情報により復号して利用者識別情報を得て、さらに該利用者識別情報をもとに暗号化されている前記認証済証明書とファイルアクセス要求リストを復号して利用者の認証および遠隔ファイルアクセス許可の処理を行う遠隔ファイルアクセス処理手段を有して、相互に認証を行うことを特徴とする遠隔ファイルアクセスシステム。

【0008】

【作用】本発明は、複数のコンピュータシステムが任意の通信媒体を介して接続されたネットワーク上で、特定のコンピュータシステムが利用者情報を一括管理して発行する認証済証明書に基づいて、前記複数のコンピュータシステムに接続される遠隔ファイルシステムのファイルアクセスを行う遠隔ファイルアクセスシステムにおいて、遠隔ファイルアクセス要求手段は、遠隔ファイルア

クセス要求元のコンピュータシステムは、前記認証済証明書とファイルアクセス要求リストを利用者固有の利用者識別情報をもとに暗号化し、該暗号化された情報と利用者識別情報をさらに前記特定のコンピュータシステムから取得したアクセス要求先コンピュータシステムの情報をもとに暗号化し、該暗号化された情報により前記遠隔ファイルのアクセス要求を行い、遠隔ファイルアクセス処理手段は、遠隔ファイルアクセス要求先のコンピュータシステムは、前記遠隔ファイルのアクセス要求を自コンピュータシステムがもつ情報により復号して利用者識別情報を得て、さらに該利用者識別情報をもとに暗号化されている前記認証済証明書とファイルアクセス要求リストを復号して利用者の認証および遠隔ファイルアクセス許可の処理を行い、相互に認証を行う。

【0009】

【実施例】以下、図面を参照して本発明の実施例について説明する。

【0010】図1は、本発明の一実施例である遠隔ファイルアクセスシステムの構成ブロック図である。図1において、複数のコンピュータシステム2～5は通信媒体1に接続されている。コンピュータシステム2は、認証サーバであり、ネットワーク上のユーザ情報を一括管理するユーザ情報データベース管理部2-2とユーザのサービス利用可否を判断するための認証済証明書を発行する認証処理部2-1から構成されており、ユーザ情報およびユーザがネットワーク上で利用可能なサービス情報を管理するユーザ情報データベース2-3および暗号鍵データベース2-4が接続されている。コンピュータシステム3～5は、ユーザが一般に利用するコンピュータシステムであり、ユーザ名およびパスワードのユーザ情報を入力する認証要求部3-1～5-1と、他のコンピュータシステムへサービスを要求する遠隔ファイルアクセス要求部3-2～5-2と、他のコンピュータシステムからのサービス要求を受け付ける遠隔ファイルアクセス処理部3-3～5-3から構成される。なお、遠隔ファイルアクセス処理部3-3～5-3は、ファイルシステム6～8のアクセス許可リストを有している。

【0011】図2は、遠隔ファイルアクセスの概要フローチャートである。図2において、まず、コンピュータシステム3～5の認証要求部3-1～5-1は、ユーザが入力したユーザ名・パスワードをもとに、認証サーバ2の認証処理部2-1にユーザの身元確認を要求し、正しいと判断されて認証済証明書が発行されることにより、ユーザの利用開始手続が完了する(ステップ21)。次に、この認証済証明書は、コンピュータシステム3～5に保存される。コンピュータシステム3～5の遠隔ファイルアクセス要求部3-2～5-2は、認証済証明書をもとに遠隔ファイルアクセスの要求を行う(ステップ22)。これに対し、コンピュータシステム3～5の遠隔ファイルアクセス処理部3-3～5-3は、遠

隔ファイルアクセス要求のファイルアクセス可否のチェックを行い(ステップ23)、アクセス許可である場合には、認証済証明書を確認してファイルアクセスが行われる。

【0012】図3は、認証済証明書のフォーマット図である。図3において、認証済証明書は、認証サーバのホスト名および認証サーバの位置を示すアドレスが格納されている。その他、ユーザ名、ユーザの識別を示すユーザID番号、認証済証明書の発行時刻および有効期限が格納される。この認証済証明書をもとに、コンピュータシステムの遠隔ファイルアクセス要求部は、遠隔ファイルアクセスを行うことになる。

次に、認証済証明書を発行されたコンピュータシステム3のユーザがコンピュータシステム4のファイルアクセスを要求する場合を説明する。この遠隔ファイルアクセス要求は、コンピュータシステム3の遠隔ファイルアクセス要求部3-2が、アクセス要求バケットを組み立てて、該当ノードであるコンピュータシステム4の遠隔ファイルアクセス処理部4-3に転送することによって行われる。

【0013】図4は、アクセス要求バケットの構成を示す図である。図4において、斜線部41の認証済証明書とアクセス要求リストは、コンピュータシステム3の要求ユーザの復号鍵で暗号化され、斜線部40のように、アクセス要求バケット全体は、遠隔ノードであるコンピュータシステム4の暗号鍵で暗号化されている。なお、アクセス要求リストは、対象ファイル名とアクセスの種類が含まれている。

【0014】図5は、遠隔ファイルアクセス要求部の動作フローチャートである。図5において、まず、認証済証明書を認証サーバ2から受け取ると、コンピュータシステム3の遠隔ファイルアクセス要求部3-2は、遠隔ノードのコンピュータシステム4の遠隔ファイルのアクセス要求リストを作成する(ステップ51)。次に、ファイルアクセスする遠隔ノードの暗号鍵を取得する(ステップ52)。さらに、受け取った認証済証明書と作成したアクセス要求リストを、ユーザの身元保証を行うため、ユーザのみが知る復号鍵で暗号化する(ステップ53)。さらに、ユーザ名、ユーザID、ステップ53で暗号化した情報からアクセス要求バケットを作成する(ステップ54)。その後、遠隔ノードの暗号鍵で暗号化する(ステップ55)。ここで、この暗号鍵は、認証サーバ2の暗号鍵データベース2-4から取得する。その後、遠隔ノードの遠隔ファイルアクセス処理部に送信する(ステップ56)。

【0015】図6は、遠隔ファイルアクセス処理部の動作フローチャートである。図6において、まず、コンピュータシステム4の遠隔ファイルアクセス処理部4-3は、受信したアクセス要求バケットを自己のみが知る復号鍵で復号する(ステップ61)。次に、アクセス要求

5

バケットが復号できたか否かを判定し（ステップ62）、復号できない場合は、ステップ69へ移行し、他のノードへの要求またはネットワーク上第三者によってデータが介在されたと判断して、遠隔ファイルアクセスの処理を否定し、中止する（ステップ69）。一方、復号できた場合は、アクセス要求バケットの中から、アクセス要求元のユーザ名を取り出し、このユーザ名により認証サーバ2の暗号鍵データベース2-4からユーザの暗号鍵を取得する（ステップ63）。さらに、この取得された暗号鍵により、アクセス要求リストを復号化する（ステップ64）。次に、アクセス要求リストが復号できたか否かを判定する（ステップ65）。復号できない場合は、アクセス要求リストがアクセス要求者のみが知るアクセス要求者の復号鍵で暗号化されているので、異なるユーザからのアクセス要求であるか、第三者により介在されたとしてアクセス要求を否定し、遠隔ファイルアクセス処理を中止する（ステップ69）。一方、アクセス要求リストが復号された場合は、アクセス要求リストの中から認証済証明書を取り出し、認証済証明書に含まれる認証済証明書の発行時刻と有効期限により、認証済証明書の正当性を判定する（ステップ66）。認証済証明書の正当性が確認されない場合は、ステップ55と同様に、遠隔ファイルアクセス処理を否定し、処理を中止する（ステップ69）。一方、認証済証明書が正当で

$$\text{発行時刻 } T_t \leq \text{現在時刻 } T_c \leq \text{発行時刻 } T_t + \text{有効期限 } T_l$$

(1)

次に、式(1)が成立する場合は、正当な認証済証明書として処理し（ステップ75）、式(2)が成立しない場合は、不正な認証済証明書として処理する（ステップ76）。

【0018】なお、以上の実施例においては、公開鍵暗号方式による遠隔ファイルアクセスシステムの例を示しているが、公開鍵暗号方式に限定するものではなく、システムとして最も適する暗号方式を採用すればよい。

【0019】このようにして、遠隔ファイルアクセスのファイルアクセス権のチェックは、認証済証明書を用いて、アクセス要求者が、本来使用すべきユーザIDによりアクセス要求を行っているかの判断が可能となり、これによる身元保証を行った上で、ユーザIDとファイルアクセスリストによるアクセスチェックを行うため、正確にチェックすることが可能となり、ネットワーク上で遠隔ファイルアクセスシステムの正確性・安全性を向上させることが可能となる。

【0020】

【発明の効果】以上説明したように、本発明は、複数のコンピュータシステムが任意の通信媒体を介して接続されたネットワーク上で、特定のコンピュータシステムが利用者情報を一括管理して発行する認証済証明書に基づいて、前記複数のコンピュータシステムに接続される遠隔ファイルシステムのファイルアクセスを行う遠隔ファイルアクセスシステムにおいて、遠隔ファイルアクセス

6

あると確認できた場合は、さらに、アクセス要求をユーザ識別子（ユーザID）と遠隔ファイルアクセス処理部4-3が有するアクセス許可リストにより、ファイルアクセスの許可を判断する（ステップ67）。アクセス許可リストにユーザIDでのアクセス許可が登録されていない場合には、ステップ66と同様に、ファイルアクセスが否定され、アクセス処理が中止される（ステップ69）。一方、アクセス許可が登録されている場合に、初めて、ファイルアクセスが許可される（ステップ68）。

【0016】ここで、ステップ66における認証済証明書の正当性判断について、詳細に説明する。図7は、認証済証明書の正当性判断フローチャートである。図7において、まず、認証済証明書の中からユーザ名、ユーザIDを取り出し（ステップ71）、ファイルアクセス要求バケットのユーザ名、ユーザIDと等しいか否かを判断する（ステップ72）。ここで、等しくない場合は、不正な認証済証明書として処理される（ステップ76）。一方、ユーザ名、ユーザIDが等しい場合は、現在時刻 T_c 、発行時刻 T_t 、有効期限 T_l を取り出す（ステップ73）。次に、式(1)が成立するか否かを判断する（ステップ74）。

【0017】

要求手段は、遠隔ファイルアクセス要求元のコンピュータシステムは、前記認証済証明書とファイルアクセス要求リストを利用者固有の利用者識別情報をもとに暗号化し、該暗号化された情報と利用者識別情報をさらに前記特定のコンピュータシステムから取得したアクセス要求先コンピュータシステムの情報をもとに暗号化し、該暗号化された情報により前記遠隔ファイルのアクセス要求を行い、遠隔ファイルアクセス処理手段は、遠隔ファイルアクセス要求先のコンピュータシステムは、前記遠隔ファイルのアクセス要求を自コンピュータシステムがもつ情報により復号して利用者識別情報を得て、さらに該利用者識別情報をもとに暗号化されている前記認証済証明書とファイルアクセス要求リストを復号して利用者の認証および遠隔ファイルアクセス許可の処理を行い、相互に認証を行う。すなわち、従来の一方向的なユーザIDとアクセス許可リストの組合せによる遠隔ファイルアクセス権のチェックに加えて、アクセス要求者が本来利用するユーザIDによって要求しているかを認証することにより、アクセス要求者の身元保証を相互に行うようにしているため、遠隔ファイルシステムのファイルアクセスの正当性・安全性を向上することができるという利点を有する。

【図面の簡単な説明】

【図1】本発明の一実施例である遠隔ファイルアクセスシステムの構成ブロック図。

【図 2】 遠隔ファイルアクセスの概要フローチャート。

【図 3】 認証済証明書のフォーマット図。

【図 4】 アクセス要求バケットの構成を示す図。

【図 5】 遠隔ファイルアクセス要求部の動作フローチャート。

【図 6】 遠隔ファイルアクセス処理部の動作フローチャート。

【図 7】 認証済証明書の正当性判断フローチャート。

【符号の説明】

1 通信媒体

2 認証サーバ

2-1 認証処理部

2-2 ユーザ情報データベース管理部

2-3 ユーザ情報データベース

2-4 暗号鍵データベース

3~5 コンピュータシステム

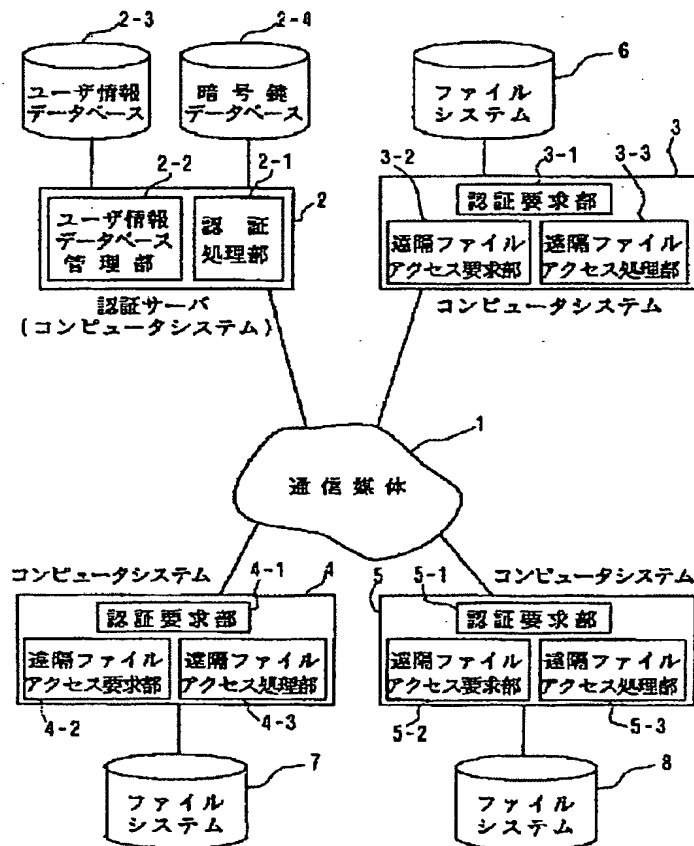
3-1~5-1 認証要求部

3-2~5-2 遠隔ファイルアクセス要求部

3-3~5-3 遠隔ファイルアクセス処理部

10 6~8 ファイルシステム

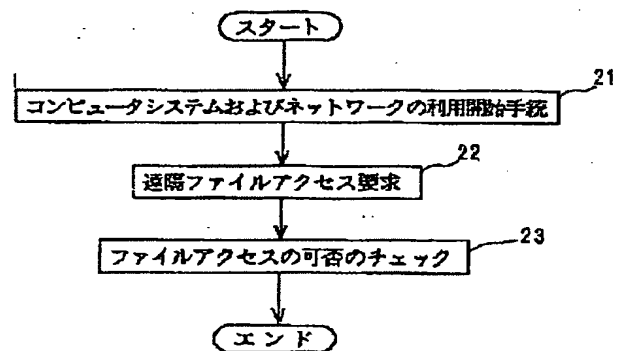
【図 1】



【図 3】

ユーザ名	ユーザID	認証サーバの ホスト名	認証サーバの アドレス	発行時刻	有効期限	

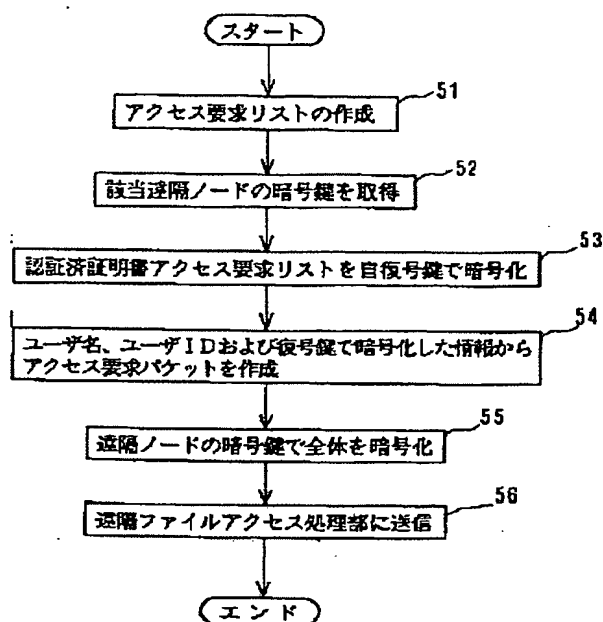
【図 2】



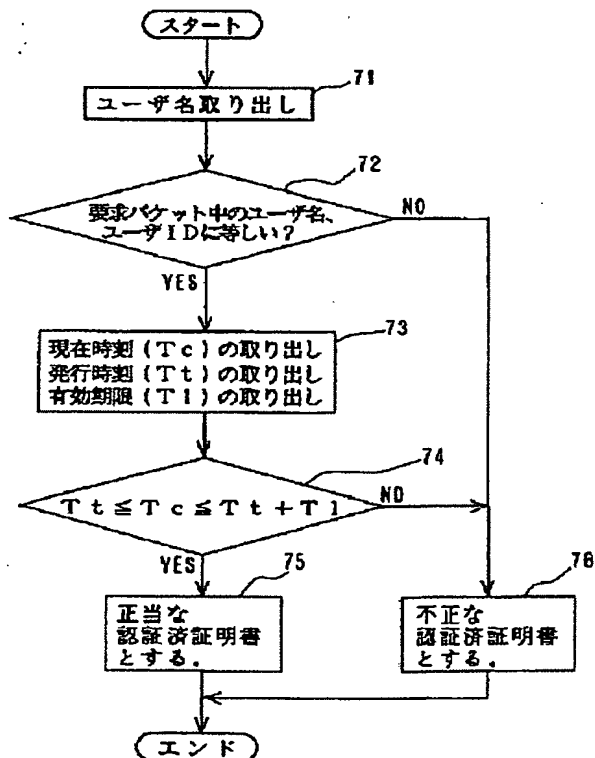
【図 4】



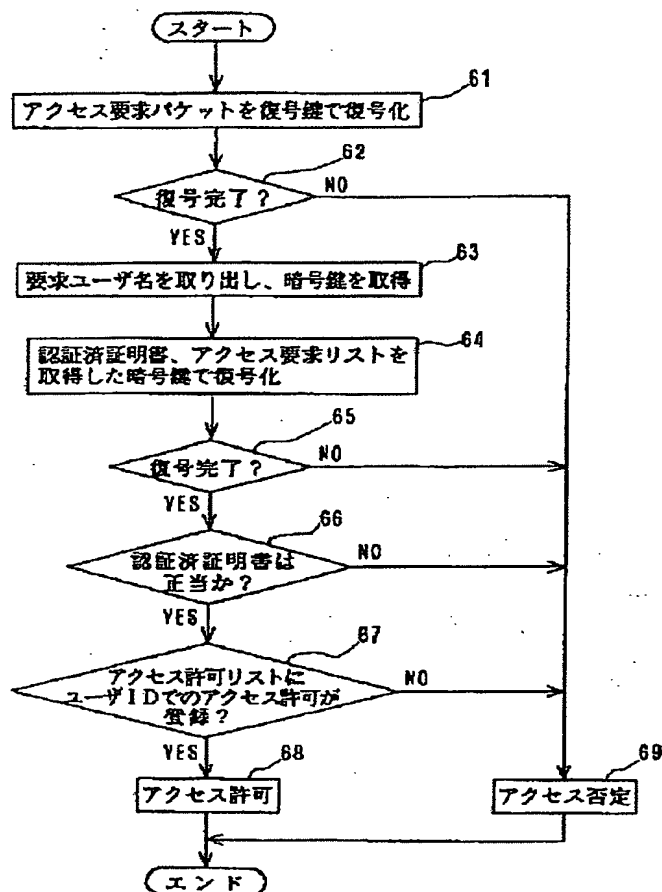
【図 5】



【図 7】



【図 6】



A

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-214862

(43)Date of publication of application : 05.08.1994

(51)Int.Cl. G06F 12/00
G06F 13/00

(21)Application number : 05-003835

(71)Applicant : HITACHI LTD

(22)Date of filing : 13.01.1993

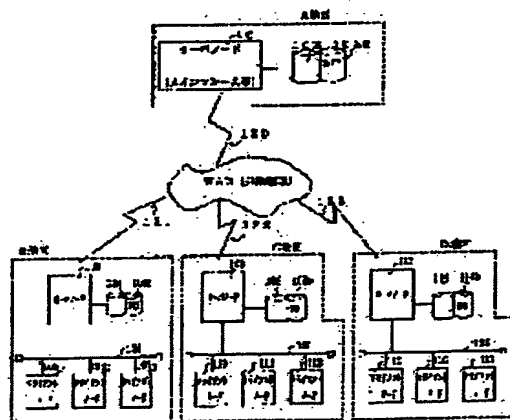
(72)Inventor : SATOMI SHIGEKI
ONO NAOO
OKA MASAMI

(54) DOCUMENT ACCESS METHOD FOR CLIENT/SERVER SYSTEM

(57)Abstract:

PURPOSE: To completely and surely protect security by performing the access control of a document file provided in a server node in the unit of a user and the unit of an area in a document.

CONSTITUTION: A server node 101 of a host center in a block A is provided with a document file 102 to be managed at the host center. A server node 103 in a block B is also provided with a document file 104. A user managing table and a document managing table are incorporated in the document file 104, and the access right for each of read, update and print is set in the unit of the user and a prescribed area in the document. When document access requests are received from client nodes 105, 106 and 107, the server node 103 judges whether the user requesting the access can access the area in the document requested with the access or not by collating the user managing table and the document managing table. Other server nodes 108 and 113 are similarly operated as well.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-214862

(43)公開日 平成6年(1994)8月5日

(51)Int.Cl.⁵G 0 6 F 12/00
13/00

識別記号

5 3 7 A 8526-5B
3 5 7 Z 7368-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数 3 O L (全 22 頁)

(21)出願番号

特願平5-3835

(22)出願日

平成5年(1993)1月13日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 里見 繁樹

神奈川県川崎市幸区鹿島田890番地の12

株式会社日立製作所情報システム事業部内

(72)発明者 小野 猶生

神奈川県川崎市幸区鹿島田890番地の12

株式会社日立製作所情報システム事業部内

(72)発明者 岡 眞美

神奈川県川崎市幸区鹿島田890番地の12

株式会社日立製作所情報システム事業部内

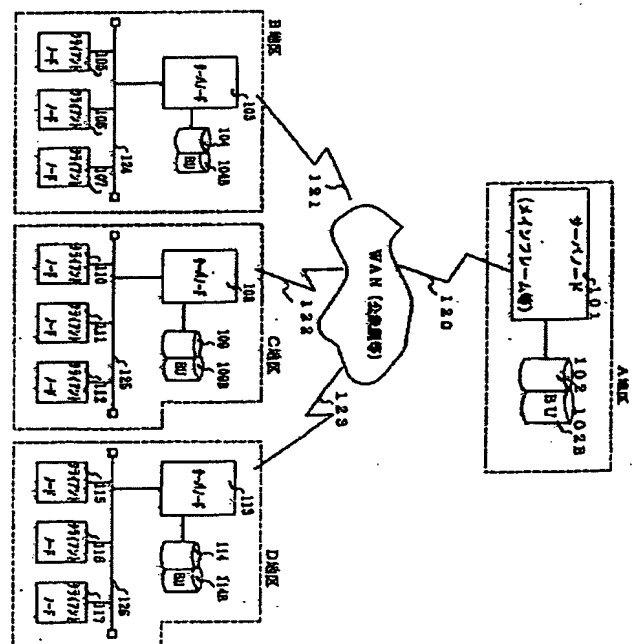
(74)代理人 弁理士 鈴木 誠

(54)【発明の名称】 クライアント・サーバシステムにおける文書アクセス方法

(57)【要約】

【目的】 サーバノードが具備する文書ファイルのアクセス制御をユーザ単位及び文書内領域単位に行って、完全・確実な機密保護を実現する。

【構成】 A地区のホストセンタのサーバノード101は、該ホストセンタで管理する文書ファイル102を有する。B地区のサーバノード103も文書ファイル104を有する。該文書ファイル104上にユーザ管理テーブルと文書管理テーブルを内蔵し、ユーザ単位及び文書の所定領域単位に読み込み、更新、印刷等別のアクセス権を設定しておく。サーバノード103は、クライアントノード105、106、107からの文書アクセス要求を受けると、ユーザ管理テーブルと文書管理テーブルを突き合わせ、アクセスを要求したユーザがアクセスを要求された文書内領域をアクセスできるか判定する。他のサーバノード108、113も同様である。



【特許請求の範囲】

【請求項1】 文書ファイルを具備し、文書の読込み、更新、印刷等のサービスを提供するサーバノードと、ユーザのサービス要求をサーバノードへ送り、サービスの提供を受けるクライアントノードから構成されるクライアント・サーバシステムにおける文書アクセス方法であって、

前記サーバノードは、ユーザごとに文書ファイルに対するサービス別のアクセス権を管理するユーザ管理テーブルと、文書ファイルの所定領域ごとにサービス別のアクセス権を管理する文書管理テーブルを持ち、クライアントノードからユーザのサービス要求を受信すると、前記ユーザ管理テーブルと文書管理テーブルを突き合せて、当該ユーザの要求するサービスを提供できるかどうか判定することを特徴とする文書アクセス方法。

【請求項2】 前記ユーザ管理テーブル及び文書管理テーブルのアクセス権をそれぞれ複数段階にレベル分けし、サーバノードは、ユーザのサービス要求に対して、当該ユーザのアクセス権レベル及び当該文書の当該領域のアクセス権レベルに応じたサービスを提供することを特徴とする請求項1記載の文書アクセス方法。

【請求項3】 前記サーバノードは、前記ユーザ管理テーブルに対するユーザごとのサービス別のアクセス権を管理するユーザ管理テーブル・アクセス権管理テーブルと、前記文書管理テーブルに対する所定領域ごとのサービス別のアクセス権を管理する文書管理テーブル・アクセス権管理テーブルを持ち、クライアントノードからユーザの前記ユーザ管理テーブルあるいは前記文書管理テーブルのサービス要求を受信すると、前記ユーザ管理テーブル・アクセス権管理テーブルあるいは前記文書管理テーブル・アクセス権管理テーブルを参照して、前記ユーザの要求するサービスを提供できるかどうか判定することを特徴とする請求項1もしくは2記載の文書アクセス方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、文書ファイルを具備して、文書の読込み、更新、印刷等のサービスを提供するサーバノードと、ユーザのサービス要求をサーバノードへ送り、サービスの提供を受けるクライアントノードから構成されるクライアント・サーバシステムにおける文書アクセス方法に関する。

【0002】

【従来の技術】従来、ワードプロセッサ等の文書編集装置においては、例えば特開平1-243172号公報に記載のように、文書内の一部分の変更を禁止するようにした文書内領域アクセス制御方式がある。この方式では、文書を構成するテキスト、図形、画像、表などの領域単位にアクセス権を設定できるようにして、複数のユ

ーザが共有する文書内に他人が変更できない情報を書込み可能にしている。アクセス権チェックは、ユーザのパスワード入力と文書内領域のアクセス権フラグにより判定し、特定データに対し部分的に書き込みができない様にしている。

【0003】

【発明が解決しようとする課題】上述の従来技術では、文書の書込み権についてはアクセス制御を行うため、特定ユーザの特定領域への書込みのみを許可し、それ以外の書込みを禁止できるが、文書の読み込み、印刷等については何も考慮されていない。このため、文書の画面表示、印刷のさいは何も規制されずに、全ユーザが全文書を読んだり印刷できることになる。これは、ワードプロセッサのように利用するユーザがある程度特定され、且つ限定される場合は左程問題はないが、クライアント・サーバシステムのように、オープンシステム環境下で多数のユーザへ文書の読込み、更新、印刷等の各種サービスを提供する場合には問題がある。

【0004】本発明の目的は、クライアント・サーバシステムにおいて、サーバノードが具備する文書ファイルに対して、ユーザ単位及び文書内所定領域単位の読込み、書込み（更新）、印刷等を許可することにより、オープンシステム環境下での不正ユーザによる不正アクセスをほぼ完全に防止し、機密保護の管理を容易に実現することにある。

【0005】

【課題を解決するための手段】上記目的を達成するために、請求項1の発明は、文書ファイルを具備し、文書の読込み、更新、印刷等のサービスを提供するサーバノードと、ユーザのサービス要求をサーバノードへ送り、サービスの提供を受けるクライアントノードから構成されるクライアント・サーバシステムにおいて、サーバノードは、ユーザごとに文書ファイルに対するサービス別のアクセス権を管理するユーザ管理テーブルと、文書ファイルの所定領域ごとにサービス別のアクセス権を管理する文書管理テーブルを持ち、クライアントノードからユーザのサービス要求を受信すると、前記ユーザ管理テーブルと文書管理テーブルを突き合せて、当該ユーザの要求するサービスを提供できるかどうか判定することを特徴とするものである。

【0006】請求項2の発明は、前記ユーザ管理テーブル及び文書管理テーブルのアクセス権をそれぞれ複数段階にレベル分けし、サーバノードは、ユーザのサービス要求に対して、当該ユーザのアクセス権レベル及び当該文書の当該領域のアクセス権レベルに応じたサービスを提供することを特徴とするものである。

【0007】請求項3の発明は、サーバノードは、さらに前記ユーザ管理テーブルに対するユーザごとのサービス別のアクセス権を管理するユーザ管理テーブル・アクセス権管理テーブルと、前記文書管理テーブルに対する

所定領域ごとのサービス別のアクセス権を管理する文書管理テーブル・アクセス権管理テーブルを持ち、クライアントノードからユーザの前記ユーザ管理テーブルあるいは前記文書管理テーブルのサービス要求を受信すると、前記ユーザ管理テーブル・アクセス権管理テーブルあるいは前記文書管理テーブル・アクセス権管理テーブルを参照して、前記ユーザ管理テーブルあるいは前記文書管理テーブルについてユーザの要求するサービスを提供できるかどうか判定することを特徴とするものである。

【0008】

【作用】サーバノードのユーザ管理テーブルと文書管理テーブルには、ユーザごと及び文書の領域ごとに読み込み、書き込み、印刷等のアクセス権が設定されている。該サーバノードは、クライアントノードからの文書アクセス要求を受け、文書管理テーブルとユーザ管理テーブルの突き合わせを行い、サービスすなわちアクセスを要求したユーザがアクセスを要求された文書内領域にアクセスできるか判定し、その結果をクライアントノードに返す。これにより、ユーザはアクセスを許可された文書内領域に限り、許可された範囲で（読み込み／書き込み／印刷等）アクセスすることができる。つまり、オープンシステム環境における文書の読み込み、書き込み、印刷等のアクセス制御を文書内領域単位及びユーザ単位に実現できる。

【0009】

【実施例】以下、本発明の一実施例について図面により詳細に説明する。

【0010】図1では本発明の一実施例を示すクライアント・サーバシステムのネットワークブロック図である。ここで、A地区にはホストセンターがあり、サーバノード101が設置される。このサーバノード101はメインフレーム・ホストコンピュータ等であり、ホストセンター管理する文書に関するファイル102と該ファイル102のバックアップ用ファイル102Bを持つ。B地区にはサーバノード103が設置され、文書ファイル104とそのバックアップ用ファイル104Bを持つ。サーバノード103は、パソコン、ワークステーション等である。このサーバノード103はLAN等の回線124でクライアントノード105、106、107と接続されている。クライアントノード105、106、107もパソコン、ワークステーション等である。C地区、D地区とB地区も同様に、サーバノード108、113が文書ファイル109、114、バックアップ用ファイル109B、114B、を持ち、LAN等の回線125、126によりクライアントノード110～112、115～117とそれぞれ接続されている。

【0011】A地区のサーバノード101は、B地区のサーバノード103と、回線120～WAN（公衆網等）～回線121で接続され、さらに、C地区、D地区

のサーバノード108、113とも同様に、回線120～WAN（公衆網等）～回線122、回線120～WAN（公衆網等）～回線123で各々接続されている。通信回線はISDN、DDX-C、DDX-P、専用回線等である。

【0012】以下では、B地区のクライアント・サーバシステムを例にして、サーバノードの文書管理及びユーザ、クライアントノード、サーバノードの動作例を詳述する。

10 【0013】図10は、B地区サーバノード103が持つ文書ファイル104上のユーザID／パスワード管理テーブル1000である。図10において、ユーザID／パスワード管理テーブル1000には、サーバノード103が管理する文書ファイルをアクセスできる全ユーザのユーザID／パスワードが登録されており、各々のユーザID／パスワードごとに、読み込み、更新、印刷のアクセス権が4段階レベルP1～P4で設定してある。例えば、P1は極秘文書もアクセスできるレベル、P2は極密以外の秘密文書はアクセスできるレベル、P3は社外秘文書はアクセスできるレベル、P4は規制無文書のみアクセスできるレベル等である。また、このユーザID／パスワード管理テーブル1000には、ユーザID／パスワードの初期登録日時と登録者及び最新変更日時と変更者を設定できる。

20 【0014】図11は、B地区サーバノード103の文書ファイル104中の文書のアクセス権管理テーブル1000であり、B地区のサーバノード103が管理する文書のアクセス権を設定している。図11においては、文書は、ファイル単位、ページ単位、文・図・文節・単位などの領域ごとに、読み込み、更新、印刷のアクセス権レベルを設定できる。アクセス権レベルは、例えば4段階あり、P1は極秘扱い、P2は秘密扱い、P3は社外秘扱い、P4は規制無し扱い等である。また、ディスプレイ画面や印刷紙上の取扱いメッセージ（例えば“極秘”“社外秘”等）の表示の有無を設定できる。さらに文書及びファイルの所有者の設定も行える。

40 【0015】図12は、B地区サーバノード103の文書ファイル104上のユーザID／パスワード管理テーブルのアクセス権管理テーブル1200である。これは、図10のユーザID／パスワード管理テーブル1000に対するアクセス（表示、変更、印字等）権をユーザID／パスワードごとに4段階レベルに設定したものである。権利レベルは、例えば、Q1がP1レベルのアクセス権を扱える、Q2がP2レベルのアクセス権を扱える、Q3がP3レベルのアクセス権を扱える、Q4がP4レベルのアクセス権を扱える、等である。これは、サーバノード103側のシステム管理者があらかじめ設定しておく。

50 【0016】図13は、B地区サーバノード103の文書ファイル104上の文書アクセス権管理テーブルのA

クセス権管理テーブル1300であり、図11の文書アクセス権管理テーブル1100に対するアクセス（表示、変更、印字等）権を4段階レベルに設定したものである。権利レベルは、例えばQ1がP1相当のアクセス権レベル、Q2がP2相当のアクセス権レベル、Q3がP3相当のアクセス権レベル、Q4がP4相当のアクセス権レベル、等である。これも、サーバノード103側のシステム管理者があらかじめ社内文書取扱規則に従い設定しておく。

【0017】A地区、C地区、D地区サーバノード101、108、113も、図10～図13と同様のユーザID/パスワード管理テーブル、文書のアクセス権管理テーブル、ユーザID/パスワード管理テーブルのアクセス権管理テーブル、文書アクセス権管理テーブルのアクセス権管理テーブルを各々文書ファイル102、109、114上に持つ。

【0018】B地区のサーバノード103は、クライアントノード105～107からユーザのアクセス要求を受け取ると、必要に応じて図10～図13の管理テーブルを文書ファイル104から読み込み、文書や一覧表や必要な指示をクライアントノードに送信する。以下の説明では、クライアントノードは105、ユーザはユーザIDがNW101のものとする。

【0019】図2乃至図4は、ユーザが文書を読み込む場合のユーザNW101、クライアントノード105、サーバノード103、101の動作例を示すフローチャートである。

【0020】まず、クライアントノード105のユーザNW101が文書を読み込む場合、ユーザNW101はクライアントノード105に対し電源を投入し（201）、初期画面を立ち上げる（202）。これにより、クライアントノード105は、ユーザID/パスワードの入力要求のメッセージを画面に表示する（203）。ユーザNW101が自ユーザID/パスワードを入力すると（204）、クライアントノード105は、入力されたユーザID/パスワードを直ちにサーバノード103に送信する（205）。

【0021】サーバノード103は、ユーザID/パスワードを受信し、文書ファイル104からユーザID/パスワード管理テーブル1000（図10）を読み込み（206）、ユーザID/パスワードの整合性チェックを行う（207、207'）。整合性がとれない時には、サーバノード103は当該ユーザID/パスワードでは、文書のアクセスができない旨をクライアントノード105に対し送信する（208）。クライアントノード105は、サーバノード103からのメッセージを受信し、これを画面に表示する（209）。一方、ユーザID/パスワードの整合性がとれた時には、対象とする文書をホストセンターで管理している可能性がある場合に対処するため、サーバノード103はユーザID/パ

スワードをA地区ホストセンターのサーバノード101に対し送信する。

【0022】A地区サーバノード101は、ユーザID/パスワードを受信すると（210）、自分の管理する文書ファイル102上に該当ユーザID/パスワードが登録されていないかチェックする（211、212）。そして、A地区サーバノード101は文書ファイル102上に登録されていなければ、その旨をB地区サーバノード103に送信し（212）、登録されていれば、当該ユーザID/パスワードがアクセスできる文書であるか判定するために、文書ファイル102より文書のアクセス権管理テーブルを読み込み、既取得のユーザID/パスワード管理テーブルとつぎあわせ、つぎあわせ結果テーブルを作成する（213）。さらにA地区サーバノード101は、つぎあわせ結果テーブルをもとに、該当ユーザがアクセスできる文書を判定し、該当する文書一覧をB地区サーバノード103に送信する（214）。なお、サーバノード101で作成される突合せ結果テーブル及び文書一覧の様式は、後述のサーバノード103で作成される突合せ結果テーブル1及び文書一覧1と基本的に同じである。

【0023】B地区サーバノード103は、A地区サーバノード101から文書一覧を受信するとともに、B地区の文書ファイル104から文書のアクセス権管理テーブル1100（図11）を読み込み、既取得のユーザID/パスワード管理テーブル1000とつぎあわせを行い、つぎあわせ結果テーブル（これを、つぎあわせ結果テーブル1と称す）を作成する（215）。さらにA地区サーバノード101からの受信結果を考慮して、該当ユーザがアクセスできる文書一覧（これを文書一覧1と称す）を作成し、クライアントノード105に送信する（216）。つぎあわせ結果テーブル1及び文書一覧1と、その作成の仕組みについては、図14及び図15により後述する。

【0024】クライアントノード105は、受信した文書一覧1を画面に表示し（217）、ユーザにアクセスしたい文書名の入力をうながすメッセージを出力する（218）。ユーザが文書名を入力すると（219）、クライアントノード105は、入力された文書名が文書一覧1上に存在するかチェックを行い（220）、存在する場合には、サーバノード103に該当文書の送信要求を出す（221）。サーバノード103は、文書ファイル104から該当する文書を読み込み（222）、既作成のつぎあわせ結果テーブル1をもとに、文書上に取扱メッセージの表示や表示できない部分のマスクがけを行い（223）、クライアントノード105に該当文書を送信する（224）。クライアントノード105は、受信した文書を画面に表示し（225）、ユーザが文書操作を開始する（226）。

【0025】一方、ユーザから入力された文書名が文書

一覧1に存在しない時、クライアントノード105は、文書が新規作成であるか問い合わせのメッセージを出力する(227)。ユーザがYESかNOを入力すると(228)、クライアントノード105は、これを判定し(229)、NO(新規作成文書でない)の場合には、該当文書が無い旨のメッセージを出力する(230)。YES(新規作成文書である)の場合には、クライアントノード105は文書の新規作成要求をサーバノード103に連絡する(231)。サーバノード103は、文書の新規作成要求を受信すると、つきあわせ結果テーブル1を参照して、該当するユーザが文書の新規作成ができるか判定し(232、233)、判定結果をクライアントノード105に送信する(234)。クライアントノード105は、受信結果を判定して(235)、新規作成ができない場合、別文書名などの入力をうながすメッセージを表示する(236)。新規作成ができる場合、クライアントノード105は新規文書画面を表示し(237)、ユーザが新規文書入力を開始する(238)。

【0026】図5は、ユーザが文書を変更して格納する場合のユーザNW101、クライアントノード105、サーバノード103、101の動作例を示すフローチャートである。

【0027】ユーザが、表示された文書を変更し格納を行う場合には、ユーザの文書入力終了(301)に続く文書格納要求(302)に対し、B地区クライアントノード105は、格納要求をサーバノード103に対して送信する(303)。B地区サーバノード103は、格納要求を受信すると、既作成のつきあわせ結果テーブル1を参照し、該当ユーザが該当文書の変更ができるか判定し(304、305)、判定結果をクライアントノード105に送信する(306)。クライアントノード105は、受信結果を判定し(307)、文書変更(格納)ができない場合、文書格納できない旨を画面表示する(308)。

【0028】一方、文書変更ができる場合、クライアントノード105は、ユーザに格納確認をうながすメッセージを表示し(309)、ユーザの格納確認(310)を受け、文書の格納要求と格納する文書をサーバノード103に送信する(311)。サーバノード103は、これを受信し、つきあわせ結果テーブル1をもとに格納文書の所有者を判定し(312、313)、所有者がA地区ホストセンターのサーバノード101の場合には、A地区サーバノード101に対して格納要求と格納文書を送信する。これに対し、A地区サーバノード101は、既作成のつきあわせ結果テーブルをもとに変更権のない箇所が変更されていないかチェックする(314、315)。チェックを通過すれば、文書を格納し(317)、変更権のない箇所が変更されている場合には、格納できない箇所と格納できない旨をB地区サーバノード

103に送信する(316)。

【0029】格納する文書の所有者がB地区サーバノード103の場合には(313)、B地区サーバノード103は、つきあわせ結果テーブル1を参照し、変更権のない箇所が変更されていないかチェックする(318、319)。変更権のない箇所が変更されていない事が確認できた場合(319)、文書を格納する(321)。変更権のない箇所が変更されている場合には、A地区サーバノード101からの送信文も考慮し、格納できない箇所と格納できない旨をクライアントノード105に送信する(320)。クライアントノード105は、受信した格納できない箇所と格納できない旨のメッセージを画面出力する(322)。

【0030】図6は、ユーザが文書を印刷する場合のユーザNW101、クライアントノード105、サーバノード103の動作例を示すフローチャートである。

【0031】ユーザNW101が文書を印刷する場合には、ユーザがアクセスできる文書一覧1から印刷したい文書名を選択し、印刷要求を入力すると(401)、B地区クライアントノード105は、印刷要求をサーバノード103に送信する(402)。B地区サーバノード103は、つきあわせ結果テーブル1を参照し、該当ユーザが該当文書を印刷できるか判定し(403、404)、判定結果をクライアントノード105に送信する(405)。クライアントノード105は、この判定結果により、印刷権が無い場合、印刷できない旨のメッセージを画面出力する(406)。

【0032】一方、印刷権がある場合、クライアントノード105は、印刷先プリンタ入力をうながすメッセージを指示し(407)、ユーザからのプリンタ名入力(408)を受け、この指定されたプリンタ名をサーバノード103に送信する(409)。サーバノード103は、該当プリンタが出力可能である事を確認した上(410)、再度つきあわせた結果テーブル1を参照し、取扱メッセージや印字できない部分のマスクが行い(411)、印刷文書をクライアントノード105に送信する(412)。クライアントノード105は、サーバノード103から印刷文書を受信すると、ユーザに印刷確認をうながすメッセージを表示し(413)、ユーザからの印刷確認の入力を受け(414)、印刷を行う(415)。

【0033】ここで、ユーザNW101が文書を読み込み/変更(更新)/印刷する場合にサーバノード103で作成・参照されるつきあわせ結果テーブル1や文書一覧1について、図14及び図15により説明する。

【0034】図14はつきあわせ結果テーブル1の一例である。これは、B地区サーバノード103がクライアントノード105からの最初の文書アクセス要求を受信した時に、文書ファイル104上の図10のユーザID/パスワード管理テーブル1000と図11の文書アク

セス権管理テーブル1100を読み、この2つのテーブルをつきあわせ、権利レベルの高低比較を行い、アクセス可/不可を表にしたものである。このつきあわせ結果テーブル1は、該当ユーザNW101の処理終了までサーバノード103内に保持しておく。権利レベルは、P1、P2、P3、P4の順に高いものとし、高低比較では、ユーザの権利レベル $\geq$ 文書の権利レベルの時、アクセス可能であり、ユーザ権利レベル $<$ 文書の権利レベルの時、アクセス不可とする。例えばユーザNW101の読み権レベルをP2、LAN・CD1051ページ図1の読み権レベルをP1とすると、 $P2 < P1$ であるから、ユーザNW101はLAN・CD1051ページ図1の読み権はない。一方、LAN・CD1052ページ読み権レベルをP2とすると、 $P2 = P2$ であるから、ユーザNW101はLAN・CD1052ページを読み込むことができる。以下同様にアクセス可/不可を判定している。又、取扱表示、取扱印刷については、文書アクセス権管理テーブル1100から取扱メッセージや表示/印刷できない箇所のマスクかけを考慮して、画面表示/印刷時のレイアウトを図示したものである。

【0035】図15はユーザがアクセスでき文書一覧1である。ここでは、B地区サーバノード103が図14のつきあわせ結果テーブル1をもとに、クライアントノード105に送信するために作成したユーザNW101がアクセスできる文書一覧を示している。これがクライアントノード105の画面に表示され、ユーザNW101はこの中から読み、書き込みあるいは印刷する文書を選択する。

【0036】次に、ユーザがユーザID/パスワード管理テーブル(ユーザID/パスワード権利一覧)をアクセス(表示、変更、印刷)する場合の動作を説明する。

【0037】図7は、ユーザNW101がユーザID/パスワード管理テーブル(図10)を一覧表示する場合のユーザNW101、クライアントノード105、サーバノード103、101の動作例を示すフローチャートである。

【0038】ユーザNW101がユーザID/パスワード管理テーブルを一覧を表示する場合、ユーザNW101からユーザID/パスワード権利一覧表示の要求(501)を受けると、B地区クライアントノード105は、該当ユーザのユーザID/パスワードと、ユーザID/パスワード権利一覧表示要求をB地区サーバノード103に送信する(502)。サーバノード103は、クライアントノード105から該要求を受信すると、文書ファイル104上のユーザID/パスワード管理テーブルのアクセス権管理テーブル1200(図12)を読み、つきあわせ結果テーブル2を作成するとともに(504)、クライアントノード105から受信したユーザID/パスワードをA地区サーバノード103に送信する(503)。A地区サーバノード101は、文書

ファイル102上のユーザID/パスワード管理テーブルを読み、B地区サーバノード103から送信されたユーザID/パスワードが登録されているか検索する(505、505')。登録されていない旨をB地区サーバノード103に送信する(506)。登録されていれば、A地区サーバノード101は、ユーザID/パスワード管理テーブルのアクセス権管理テーブル(図12と同様のテーブル)の読みを行い、該当ユーザと他ユーザの権利レベルをつきあわせ、つきあわせ結果テーブルを作成する(507)。さらに、このつきあわせ結果テーブルを参照し、該当ユーザがアクセスできるユーザID権利一覧を作成し、B地区サーバノード103に送信する(508)。B地区サーバノード103は、既作成のつきあわせ結果テーブル2と、A地区サーバノード101から受信したユーザID権利一覧をもとに、該当ユーザがアクセスできるユーザID権利一覧2を作成し、クライアントノード105に送信する(509)。クライアントノード105は、受信したユーザID権利一覧を画面表示する(510)。

【0039】なお、B地区サーバノード103で作成するつきあわせ結果テーブル2及びユーザID権利一覧2と、その作成の仕組みについては、図16、図17を用いて後述する。又、A地区サーバノード101で作成するつきあわせ結果テーブル、ユーザID権利一覧の詳細説明は省略するが、作成方法についてはB地区サーバノード103の場合と同様である。

【0040】図8は、ユーザNW101がユーザID/パスワード管理テーブル(図10)を変更するときのユーザNW101、クライアントノード105、サーバノード103の動作例を示すフローチャートである。

【0041】ユーザNW101が画面表示中のユーザID/パスワード管理テーブル(ユーザID権利一覧)を変更する場合、ユーザNW101からの入力終了後(601)、ユーザID権利一覧の変更要求(602)を受けると、B地区クライアントノード105はユーザNW101のユーザID/パスワードと変更要求をB地区サーバノード103に送信する(603)。B地区サーバノード103は、既作成のつきあわせ結果テーブル2(図16)をもとに該当ユーザに変更権があるか判定する(604、604')。該当ユーザに変更権がない場合には、サーバノード103は変更できない旨のメッセージをクライアントノード105に送信し(605)、クライアントノード105が変更できない旨を画面表示する(606)。該当ユーザに変更権がある場合には、サーバノード103は変更できる旨のメッセージと変更する一覧表の送信要求をクライアントノード105に送信する(607)。クライアントノード105が変更する一覧表を送信すると(608)、サーバノード103はつきあわせ結果テーブル2(図16)をもとに変更権

がない箇所が変更されていないかチェックし(609、609')、もし変更権のない箇所が変更されている場合には、変更できない箇所がある旨のメッセージとその箇所をクライアントノード105に送信し(610)、クライアントノード105がこれを画面表示する(611)。変更箇所にて全て変更権があることが確認された場合、B地区サーバノード103はクライアントノード105に、変更確認要求を送信し(612)、クライアントノード105が変更確認をユーザにうながすメッセージを画面表示する(613)。ユーザが変更確認を入力すると(614)、クライアントノード105はこれをサーバノード103に送信し(615)、サーバノード103がユーザID/パスワード管理テーブルの更新を行う(616)。

【0042】図9は、ユーザNW101がユーザID/パスワード管理テーブル(図10)を印刷する場合のユーザNW101、クライアントノード105、サーバノード103の動作例を示すフローチャートである。

【0043】ユーザNW101がユーザID/パスワード管理テーブル(ユーザID権利一覧)を印刷する場合、ユーザから印刷要求(701)を受けると、クライアントノード105はユーザID/パスワードと印刷要求をB地区サーバノード103に送信する(702)。B地区サーバノード103は既作成のつきあわせ結果テーブル2(図16)をもとに、該当ユーザに印刷権があるか判定し(703、703')、印刷権がない時は、印刷できない旨のメッセージをクライアントノード105に送信し(704)、クライアントノード105がこれを画面表示する(705)。印刷権がある時は、クライアントノード103は印刷できる旨のメッセージをクライアントノード105に送信し(706)、クライアントノード105はユーザに印刷先プリンタ名の入力をうながすメッセージを表示する(707)。ユーザからのプリンタ名入力(708)を受けると、クライアントノード105はこれをサーバノード103に送信する(709)。サーバノード103は指定されたプリンタが出力可能であるか確認し(710)、既作成のつきあわせ結果テーブル2(図16)を参照して印刷文書を作成し、クライアントノード105に送信する(711)。クライアントノード105は、サーバノード103からの印刷文書を受信し、ユーザに印刷確認をうながすメッセージを表示する(712)。ユーザが印刷確認を行うと(713)、クライアントノード105は、印刷を行なう(714)。

【0044】次に、図16及び図17により、つきあわせ結果テーブル2とユーザID権利一覧2の詳細と作成方法について説明する。

【0045】図16はのつきあわせ結果テーブル2の一例である。これは、ユーザID/パスワード管理テーブルをアクセスする場合に、B地区サーバノード103

が、文書ファイル104上のユーザID/パスワード管理テーブルのアクセス権管理テーブル1200(図12)を読み込み、該当ユーザID/パスワード権利と他ユーザID/パスワード権利をつきあわせ、権利レベルの高低比較を行い、アクセス可/不可を表にしたものである。このつきあわせ結果テーブル2は、該当ユーザNW101の処理終了までサーバノード103内に保持しておく。権利レベルはQ1、Q2、Q3、Q4、の順に高く、高低比較では、該当ユーザの権利レベル $\geq$ アクセス相手権利レベルの時、アクセス可であり、該当ユーザ権利レベル $<$ 相手ユーザ権利レベルの時、アクセス不可である。例えばユーザNW101のユーザID/パスワード管理テーブルの表示権はQ2、ユーザOP110の表示権をQ1とすると、 $Q2 < Q1$ であるから、ユーザNW101は、ユーザID/パスワード管理テーブル上のユーザOP101の権利を表示することはできない。一方、ユーザNW101の変更権がQ2に対し、ユーザBS320の変更権が、Q3であるとする、 $Q2 > Q3$ であるから、ユーザNW101はユーザID/パスワード管理テーブル上のユーザBS320のアクセス権を変更できることになる。以下同様にアクセス可/不可を判定できる。

【0046】図17はユーザID権利一覧2の一例である。これは、サーバノード103が図16のつきあわせ結果テーブル2をもとに、ユーザNW101からのユーザID/パスワード権利一覧表示要求に対する返信のために作成した表である。図16より、ユーザNW101はユーザOP110のアクセス権一覧を表示/変更/印刷することができないため、図17には、ユーザOP110以外のユーザNW101、ユーザBS320、ユーザBS320、ユーザOA510のユーザID/パスワード管理テーブルのユーザID権利一覧が表示されている。サーバノード103は図17のユーザID権利一覧2をクライアントノード105に送信し、クライアントノード105の画面に表示する。ユーザNW101はこの一覧表に示されるユーザID/パスワード管理テーブル内のユーザにアクセスすることになる。

【0047】ユーザが図11の文書アクセス権管理テーブル1100をアクセス(表示/変更/印刷)する場合のユーザ/クライアントノード/サーバノードの動作は、基本的にはユーザが文書を読み込み/更新/印刷する場合と同様である。ただし、サーバノードは図12及び図13のユーザ及び文書の各アクセス権管理テーブル1200、1300をつき合わせて、図14や図15のような突合せ結果テーブルや一覧表を作成することになる。

【0048】

【発明の効果】請求項1の発明によれば、サーバノード、クライアントノードから構成されるネットワークシステムにおいて、複数ユーザ、複数クライアントノード

からの文書アクセスに対し、サーバノードがユーザ単位及び文書内領域単位にアクセス権を管理し、特定ユーザに特定データの読み込み、印刷、書き込み等を許可することにより、不正ユーザによる不正アクセスを防止し、機密保護の管理が容易に実現できる。

【0049】また、請求項2の発明によれば、サーバノードがユーザ単位及び文書内領域単位にアクセス権にレベルをつけて管理することにより、ユーザのアクセス権レベルが高い場合又はアクセス権レベルが高い箇所だけアクセスを許可し、アクセス不可の文書内領域は自動的に除いて読み込み、更新、印刷等を行うことができる。

【0050】さらに、請求項3の発明によれば、サーバノードは、ユーザ権限一覧のアクセス権管理テーブルとファイル権限一覧のアクセス権管理テーブルを持ち、ユーザ権限一覧のアクセス権管理テーブルにおいて、ユーザID/パスワード管理テーブルのアクセス権をユーザIDごとに設定し管理し、ファイル権限一覧のアクセス権管理テーブルにおいて、ファイルのアクセス権管理テーブルのアクセス権をユーザIDごとにさらにファイル又は文、図、文節ごとに管理することにより、ユーザ及び文書管理テーブルに対する不正ユーザによる不正アクセスも防止することができる。

【図面の簡単な説明】

【図1】本発明の一実施例を示すクライアント・サーバシステムのネットワークブロック図である。

【図2】ユーザが文書を読み込む時のユーザ、クライアントノード、サーバノードの動作例を示すフローチャートである。

【図3】図2のフローチャートの続きである。

【図4】同じく図2のフローチャートの続きである。

【図5】ユーザが文書を変更する時のユーザ、クライアントノード、サーバノードの動作例を示すフローチャートである。

【図6】ユーザが文書を印刷する時のユーザ、クライアントノード、サーバノードの動作例を示すフローチャートである。

【図7】ユーザがユーザID/パスワード管理テーブルを表示する時のユーザ、クライアントノード、サーバノードの動作例を示すフローチャートである。

【図8】ユーザがユーザID/パスワード管理テーブルを変更する時のユーザ、クライアントノード、サーバノードの動作例を示すフローチャートである。

【図9】ユーザがユーザID/パスワード管理テーブルを印刷する時のユーザ、クライアントノード、サーバノードの動作例を示すフローチャートである。

【図10】サーバノードが持つユーザID/パスワード管理テーブルの一例である。

【図11】サーバノードが持つ文書のアクセス権管理テーブルの一例である。

【図12】サーバノードが持つユーザID/パスワード管理テーブルのアクセス権管理テーブルの一例である。

【図13】サーバノードが持つ文書アクセス権管理テーブルのアクセス権管理テーブルの一例である。

【図14】図10のユーザID/パスワード管理テーブルと図11の文書アクセス権管理テーブルのつきあわせ結果を示すテーブルである。

【図15】クライアントノードの画面に表示される特定ユーザがアクセスできる文書一覧表の一例である。

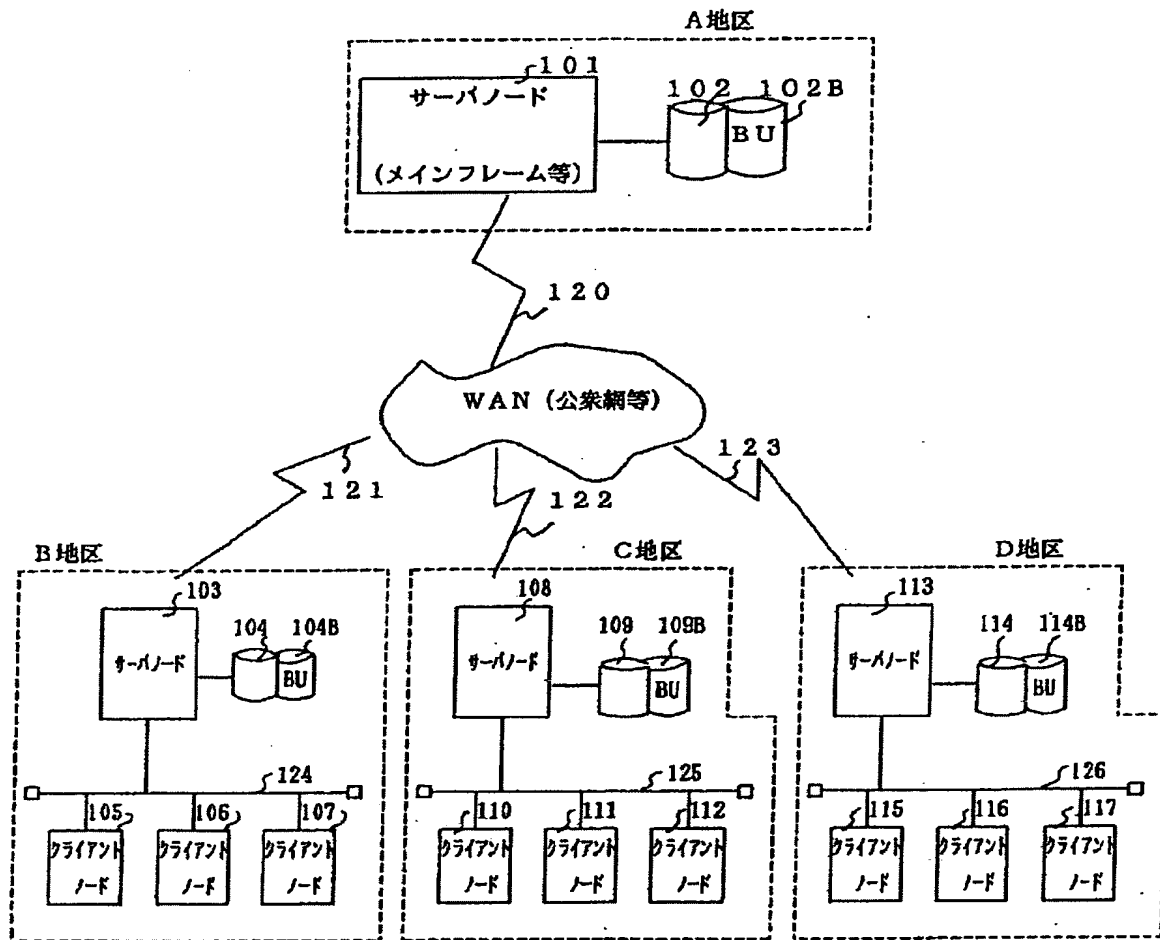
【図16】図12のユーザID/パスワード管理テーブルのアクセス権管理テーブルにおける自ユーザと他ユーザのつきあわせ結果を示すテーブルの一例である。

【図17】クライアントノードの画面に表示される特定ユーザがアクセスできるユーザID権利一覧表の一例である。

【符号の説明】

101、103、108、113 サーバノード
105、106、107、110、111、112、115、116、117 クライアントノード
102、104、109、114 ファイル
102B、104B、109B、114B バックアップ用ファイル
120、121、122、123 通信回線
1000 ユーザID/パスワード管理テーブル
1100 文書のアクセス権管理テーブル
1200 ユーザIDパスワード管理テーブルのアクセス権管理テーブル
1300 文書アクセス権管理テーブルのアクセス権管理テーブル

【図1】



【図12】

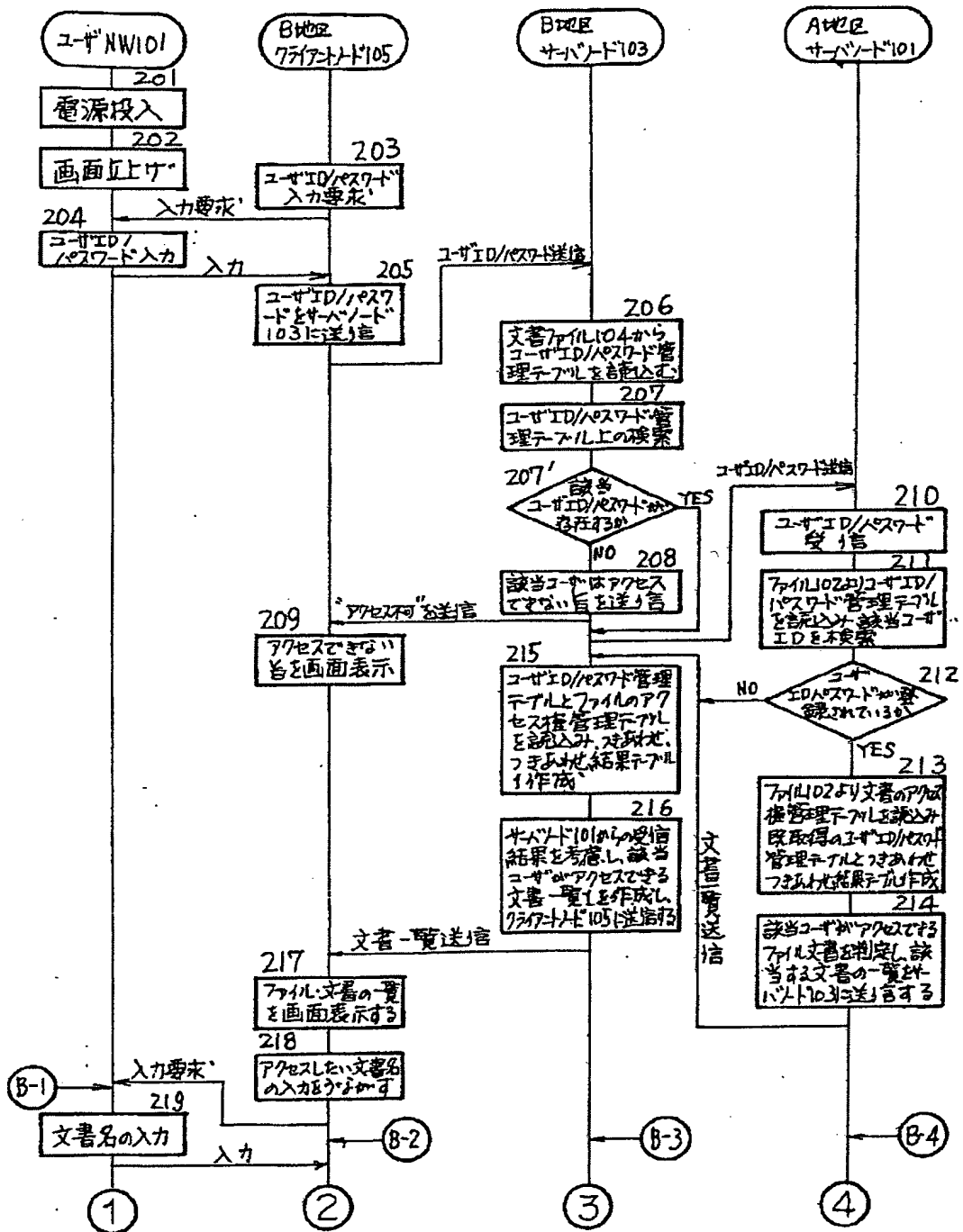
ユーザIDパスワード管理テーブルのアクセス権管理テーブル

1200～

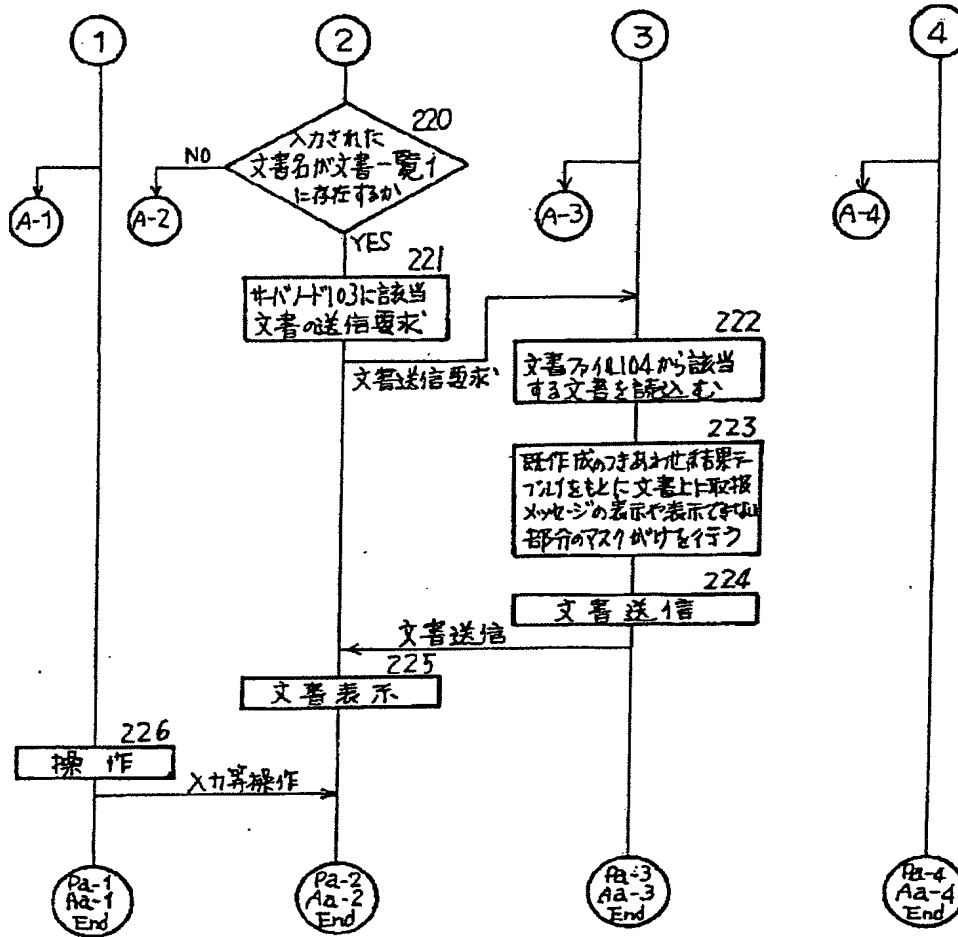
No.	ユーザID	パスワード	表示権	変更権	印刷権
1	NW101	1129	Q2	Q2	Q2
2	OP110	1003	Q1	Q1	Q1
3	BS320	0325	Q3	Q3	Q3
4	OA510	1229	Q4	Q4	Q4
—	—	—	—	—	—
—	—	—	—	—	—
—	—	—	—	—	—

凡例 Q1 : P1レベルのアクセス権を扱える
 Q2 : P2レベルのアクセス権を扱える
 Q3 : P3レベルのアクセス権を扱える
 Q4 : P4レベルのアクセス権を扱える

【図 2】



【図3】



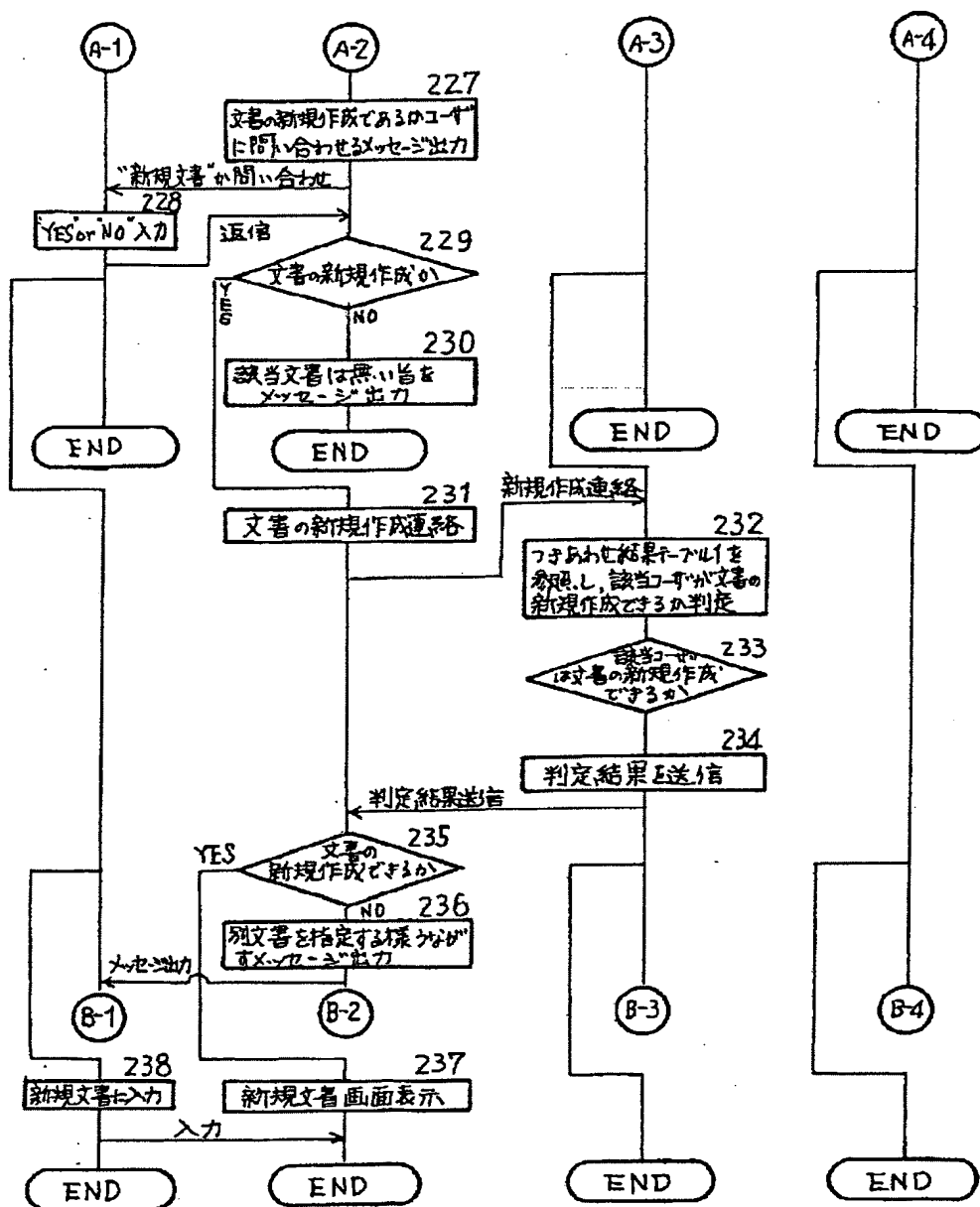
【図16】

つきあわせ結果テーブル2

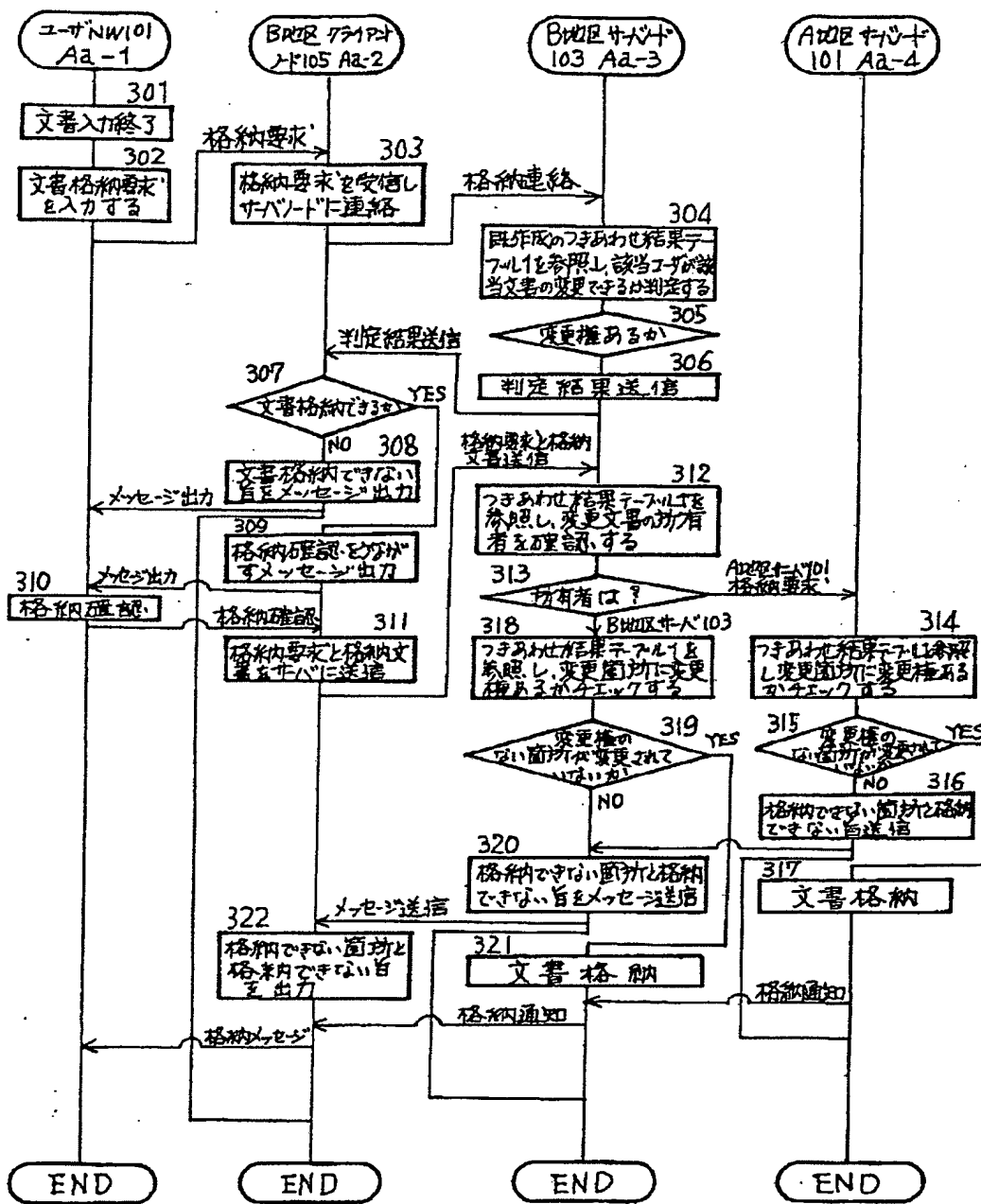
相手ユーザ	ユーザNW 101		表示権		変更権		印刷権	
	Q2		Q2		Q2		Q2	
NW101/1129	Q2	○	Q2	○	Q2	○	Q2	○
OP110/1003	Q1	×	Q1	×	Q1	×	Q1	×
BS320/0325	Q3	○	Q3	○	Q3	○	Q3	○
OA510/1229	Q4	○	Q4	○	Q4	○	Q4	○

凡例 ○ : アクセス可 × : アクセス不可

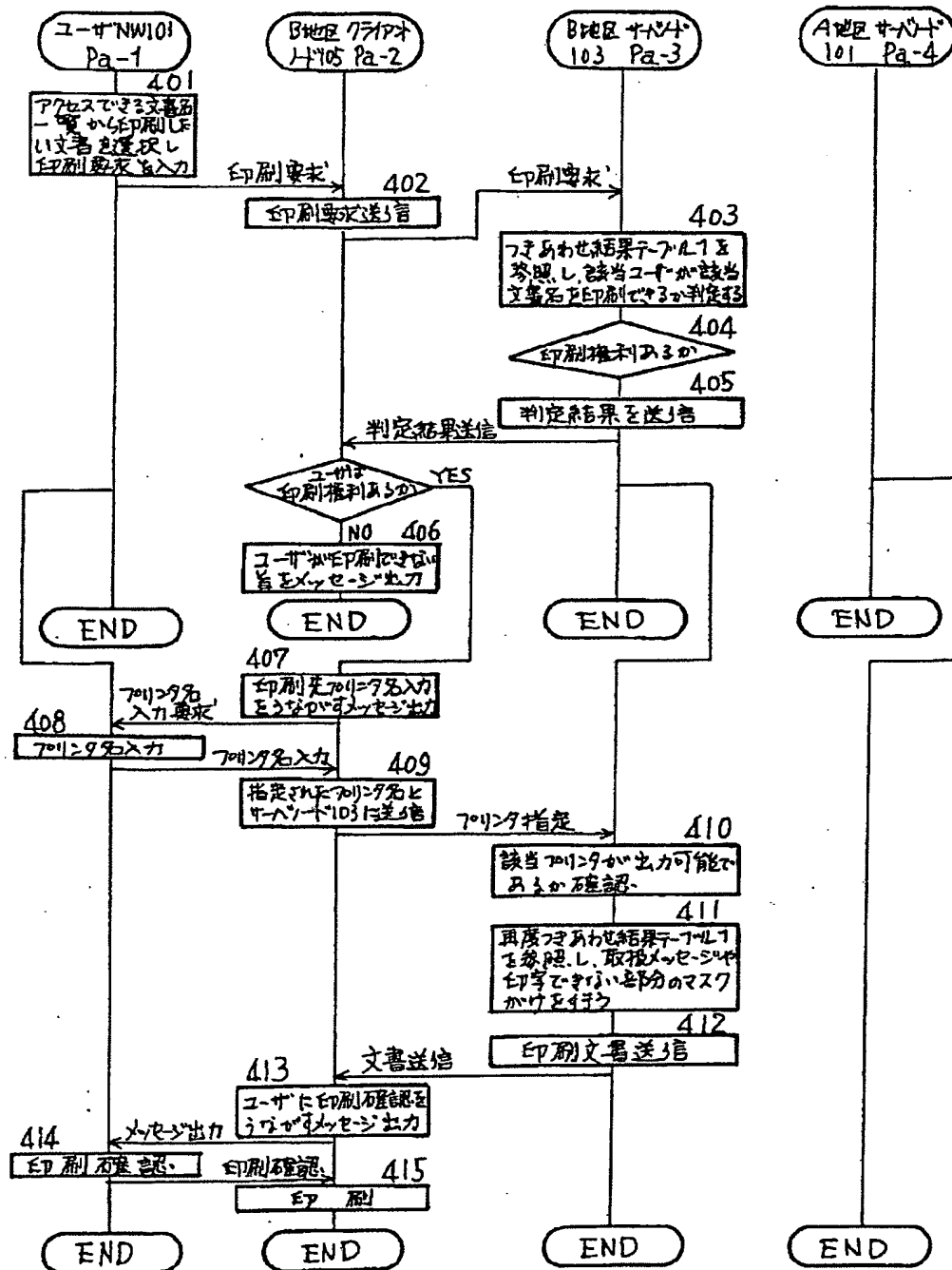
【図4】



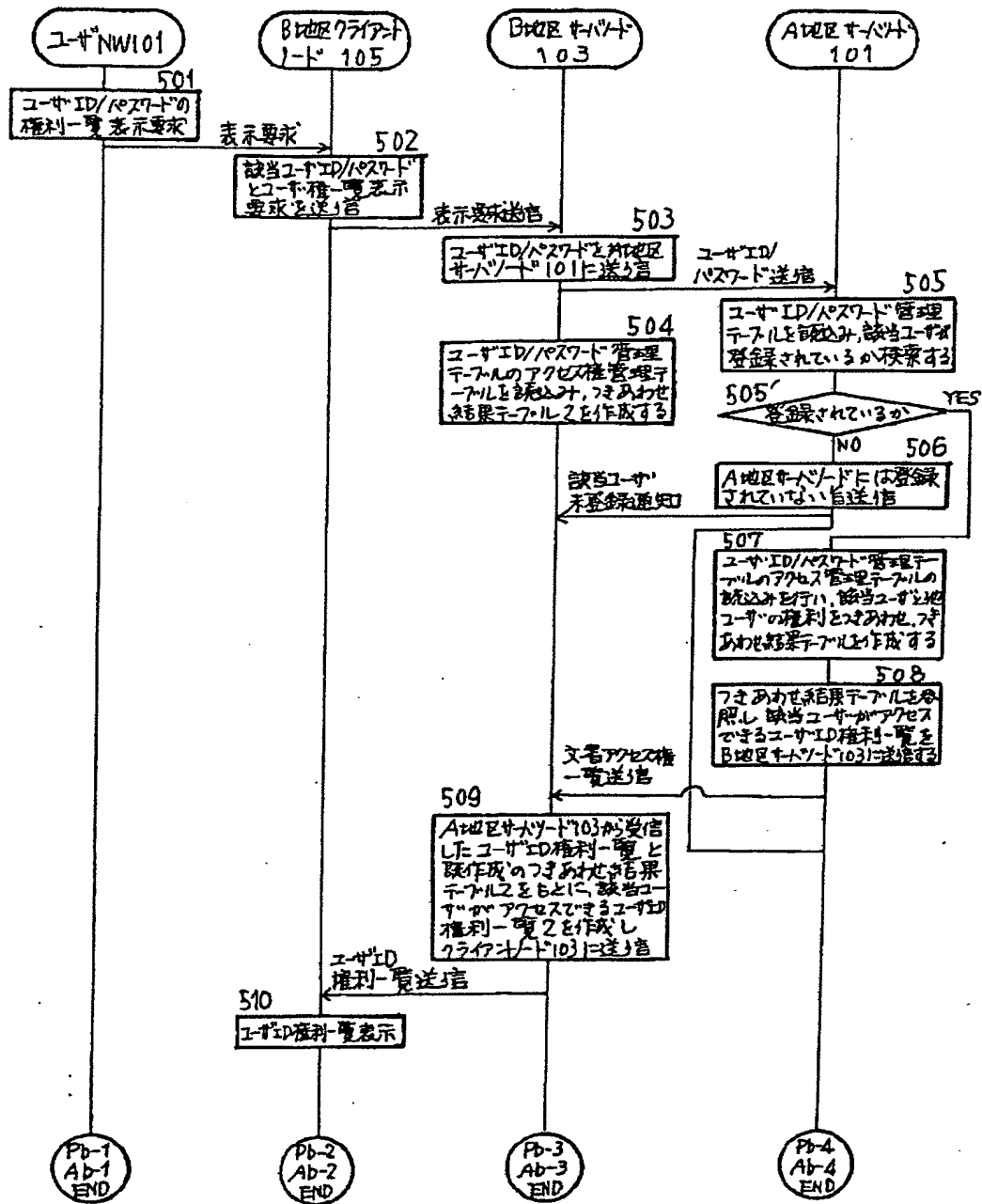
【図5】



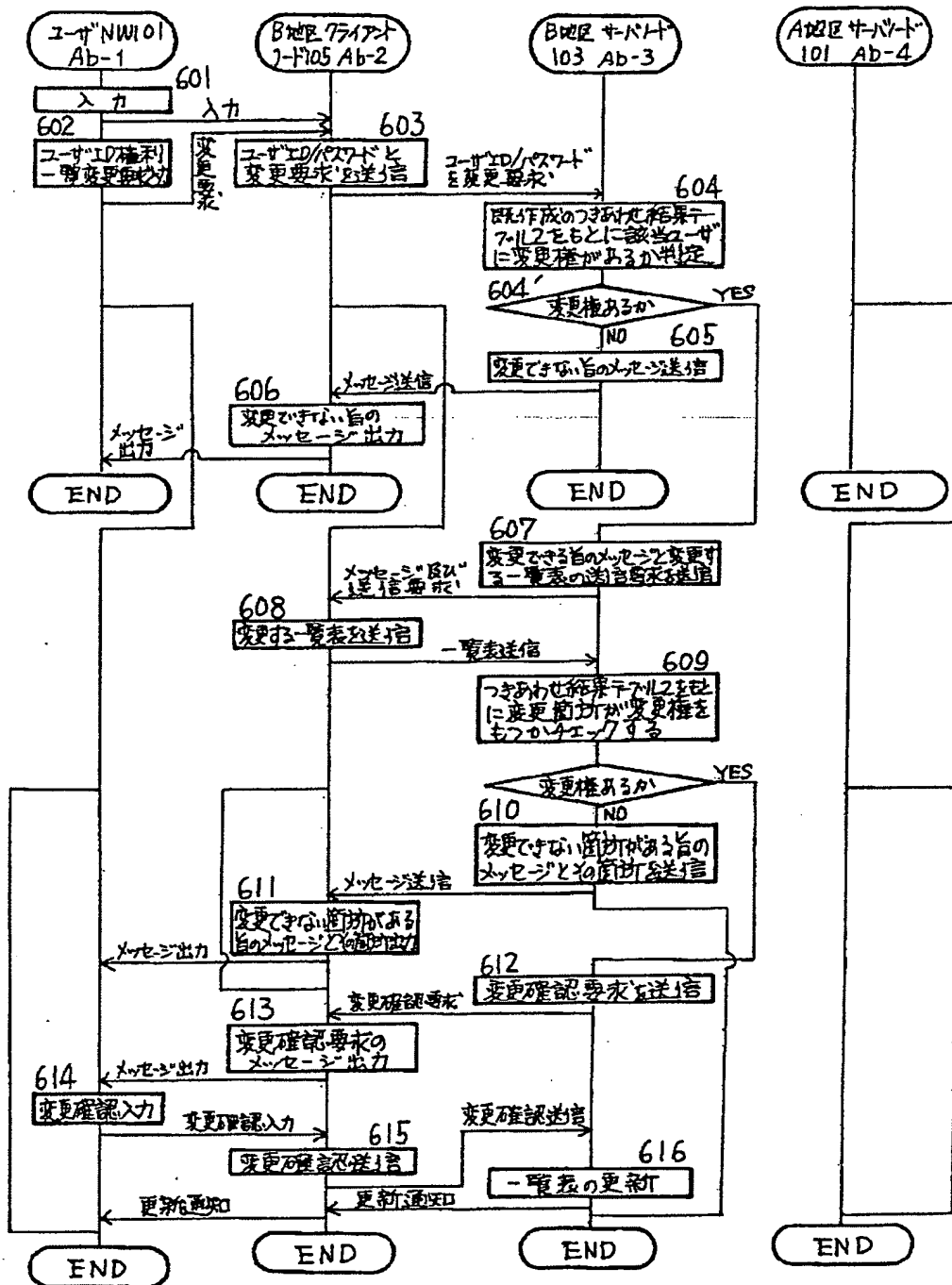
【図6】



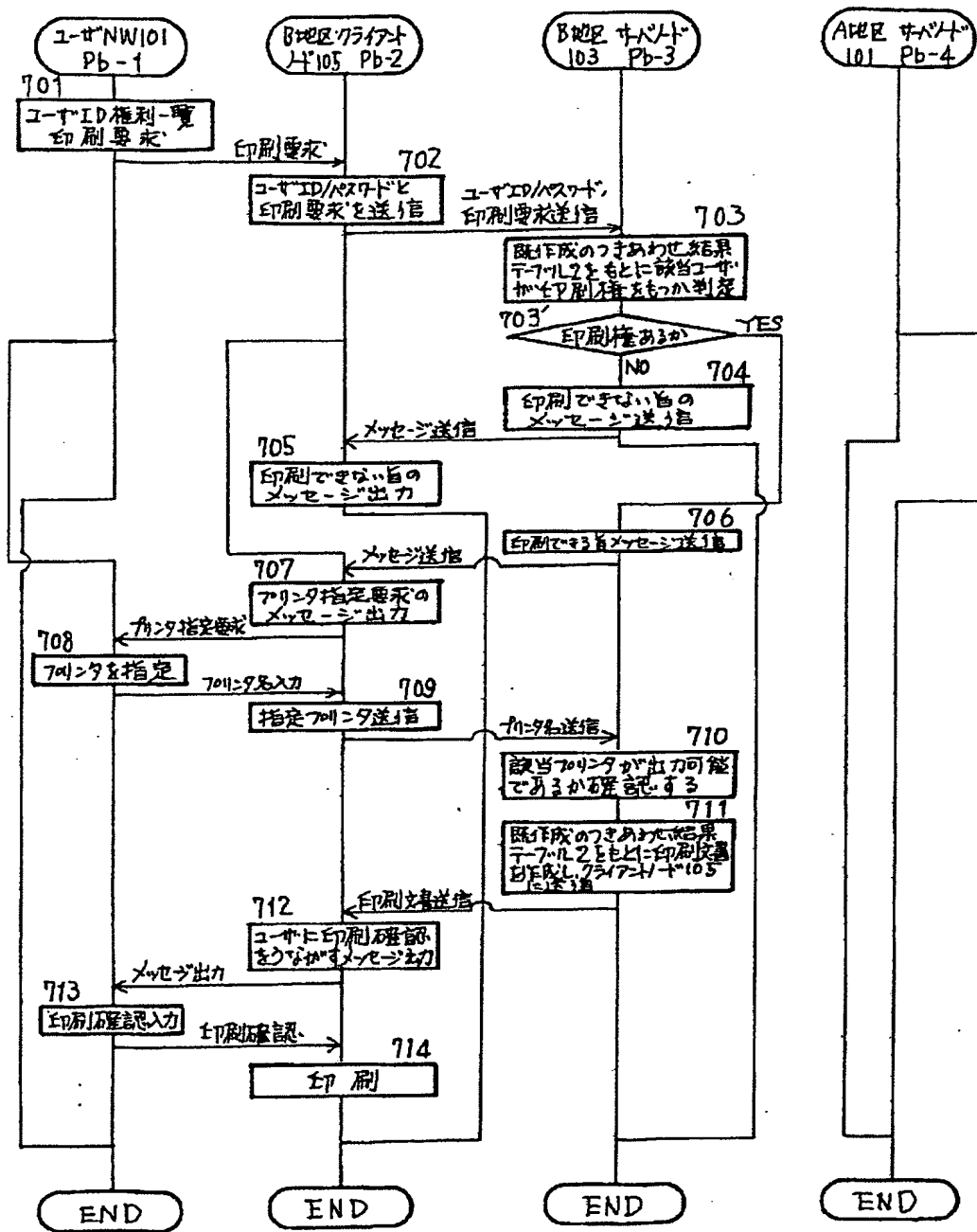
【図7】



【図8】



【図9】



【図10】

ユーザID/パスワード管理テーブル

1000

No.	ユーザID	パスワード	読み込み権	更新権	印刷権	登録日時/登録者	最新変更日時/変更者	備 考
1	NW101	1129	P2	P2	P2	91.11.13/NW101	91.12.1/OP110	
2	OP110	1003	P1	P1	P1	91.11.13/NW101	—	
3	BS320	0325	P3	P3	P3	91.11.13/NW101	—	
4	OA510	1229	P4	P4	P4	91.11.13/NW101	—	
—	—	—	—	—	—	—	—	
—	—	—	—	—	—	—	—	
—	—	—	—	—	—	—	—	
—	—	—	—	—	—	—	—	
—	—	—	—	—	—	—	—	

凡例 P1: 極秘文書もアクセスできる。
P3: 社外秘文書はアクセスできる。

P2: 秘密文書はアクセスできる。
P4: 規制無文書のみアクセスできる。

【図17】

ユーザNW101がアクセスできるユーザID権利一覧2

No.	ユーザID	パスワード	読み込み権	更新権	印刷権
1	NW101	1129	P2	P2	P2
2	BS320	0325	P3	P3	P3
3	OA510	1229	P4	P4	P4

【図11】

文書のアクセス権管理テーブル

1100

No.	ファイル名	文書名	ページ	文書、文 図名	読み権	更新権	印刷権	取扱表示	取扱印字	所有者	備考
1	LAN	CD105	1	図1	P1	P1	P1			サーバ 103	
				表1	P1	P1	P1				
				L5~L7	P1	P1	P1	A2	A2・B2		
				その他	P2	P2	P2				
			2	—	P2	P2	P2	A2・	A2・B2	サーバ103	
2	LAN	TR4	1	—	P4	P4	P4	—	—	サーバ103	
3	HIPANET	PS400	—	—	P3	P3	P3	A3	A3・B2	サーバ101	
—											
—											
—											
—											
—											

凡例 P1: 極秘扱い P2: 秘密扱い P3: 社外秘扱い P4: 規制無扱い

A1: 図 A2: ⑤ A3: 社秘

B1: 関
外係
秘者 B2: 複
写
禁 B3: 取
扱
注
意

【図13】

文書のアクセス権管理テーブルのアクセス権管理テーブル
～1300

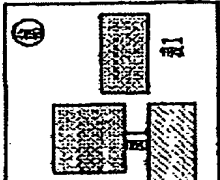
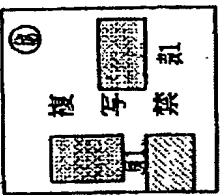
N.º	ファイル名	文書名	ページ	文節、文、図名	表示権	変更権	印刷権
1	LAN	CD105	1	図1	Q1	Q1	Q1
				表1	Q1	Q1	Q1
				L5～L7	Q2	Q2	Q2
				その他	Q3	Q3	Q3
			2	———	Q2	Q2	Q2
2	LAN	TR4	1	———	Q4	Q4	Q4
3	HIPANET	PS400	———	———	Q5	Q5	Q5
—	—	—	—	—	—	—	—
—	—	—	—	—	—	—	—
—	—	—	—	—	—	—	—
—	—	—	—	—	—	—	—
—	—	—	—	—	—	—	—

凡例 Q1 : P1レベルのアクセス権 Q2 : P2レベルのアクセス権
Q3 : P3レベルのアクセス権 Q4 : P4レベルのアクセス権

【図14】

つきあわせ結果テーブル1

凡例 ○:アクセス可 ×:アクセス不可

ユーザ NW101		読み込み権		書き込み権		印刷権		91.11.13/NW101 権限/警告	91.12.1/OS201 権限/警告
文書名		P2	P1	P2	P1	P2	P1	取扱注意	取扱印刷
LAN 105	1ページ 図1	P1	X	P1	X	P1	X		
	1ページ 表1	P1	X	P1	X	P1	X		
	1ページ L5~L7	P1	X	P1	X	P1	X		
	1ページ その他	P2	O	P2	O	P2	O		
	2ページ	P2	O	P2	O	P2	O		
LAN TR4	1ページ	P4	O	P4	O	P4	O		
	1ページ	P3	O	P3	O	P3	O		
NIPA NET		P3		P3	O	P3	O		

【図15】

ユーザNW101がアクセスできる文書一覧1

No	ファイル名	文書名	ページ数	所有者	
1	LAN	CD105	2	サーバ103	
2	LAN	TR4	1	サーバ103	
3	HIPANET	——	1	サーバ101	

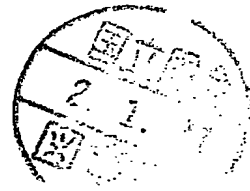
214-1121

C6G-33

学術刊行物
情処研報 Vol. 90, No. 1

情報処理学会研究報告

90 - IS - 27



1990年1月16日

社団法人 情報処理学会

BEST AVAILABLE COPY

超流通のための権利管理機構における会計処理

植木伸一* 大瀧保広** 森亮一***

*筑波大学大学院理工学研究科, 現在ソニー(株)

**筑波大学大学院工学研究科

***筑波大学電子・情報工学系

パーソナルコンピュータに権利管理機構を付加することによって"超流通"を実現しようとする研究が進行中である。権利管理機構は、ソフトウェアの著作権者の利益を守るだけでなく、従来はできなかったサービスをユーザに提供できる可能性を持つ。権利管理には、実用化された先行世代技術が存在しないので、どのようなサービスが可能であるかをなるべく詳細に検討する必要がある。

本論文では、料金体系とそれに対応した権利管理機構の一つの全体的な例における会計処理の主要な点について述べた。ここでとりあげる料金体系は、無料試用、従量料金、買取り、返金を含む。また、特定の計算機や社会の出来事などに応じた設定もできる。

The Accounting Process in Software Usage Monitor for Superdistribution

Shinichi Ueki*, Yasuhiro Ohtaki**, Ryoichi Mori***

*Master's degree program in sciences and engineering,
University of Tsukuba, now SONY Inc.

**Doctoral degree program in engineering, University of Tsukuba

***Institute of Information Science and Electronics, University of Tsukuba

"Superdistribution" is a new concept of software distribution to be realized by adding a SUM (software usage monitor) to a personal computer. The mechanism not only protects of copyrights, but also can provide many new services for users. Investigations of feasible kind of services should be pursued since no practical technique has existed for the software usage monitoring.

An example of a charge system and corresponding accounting processes in the SUM are described. The services include "free trial", "performance charge", "purchase", and "refund". It can also provide special services associated to a particular computer or social events.

1. はじめに

「超流通」とは、有料ソフトウェアの自由な流通であり、次の条件を満たすものをいう。

- 1条：ユーザは、有料ソフトウェアを、任意の経路から、随時無料で入手することができる。入手してあるソフトウェアは、著作権者の指定する条件（次条）のもとで、随時使用できる。入手および使用にあたって面倒な手数はない。
- 2条：著作権者は、そのソフトウェアの使用を許す条件を指定できる。
- 3条：そこで流通されるソフトウェアについて著作権者の指定する範囲外の変更は防止される。

超流通アーキテクチャ(SdA)は、超流通を実現するための物理的手段として森亮一によって提唱されているものである。

超流通の定義2条により、著作権者は料金を指定できる。そして、3条により、その料金は守られる。したがって、著作権者は、いかに料金を支払わずに使用しているユーザを減らすかというようなこと考えなくてよくなる。超流通では、ソフトウェアをたくさん使用してもらうために、ユーザの希望にできるだけ沿うような料金を設定するようになるだろう。

ユーザとしては、ソフトウェアの価格が安いことを望むのはもちろんであるが、それ以外に、料金を支払うだけの価値があるかどうかを知りたい、ソフトウェアを少ししか使用しない場合には使用した分だけの料金を支払いたい、最初からたくさん使用することがわかっている時には買取ってしまいたい、などの希望があるだろう。

また、著作権者としても、売上げが増えるのであれば、「教育機関」などの特定の計算機に応じて、または、「クリスマス前の1週間」のような社会の出来事に応じて、料金を割引く等のサービスをしてよいと考えるであろう。

う。

本論文では、上記の内容を満たすような料金体系をとりあげる。そして、計算機に付加される超流通のための権利管理機構が、料金体系に対応して行なう会計処理について述べる。

2 超流通アーキテクチャ(SdA)

2.1 SdAの基本用語

SdAを理解する上で必要と思われる基本的な用語の説明を行う。

- ・ Sクレジット：ソフトウェアの料金の支払のために権利管理機構内で使用される電子的な貨幣。
- ・ 計量プログラム：著作権者がソフトウェアの実行を許可する条件が記述されているもので、超流通で流通されるソフトウェアに付加されている。通常料金体系などが記述されている。
- ・ 権利管理機構（Sボックス）：計量プログラムの記述に従って、会計処理とソフトウェアの実行制御を行なう機構。計算機に付加されるが、十分な物理的防御の下に置かれている。
- ・ 支払いファイル：Sボックスにおける会計処理において、各ソフトウェアごとに作成される記録。

2.2 SdAの全体像の例

本論文では、支払いファイルを蓄積する機

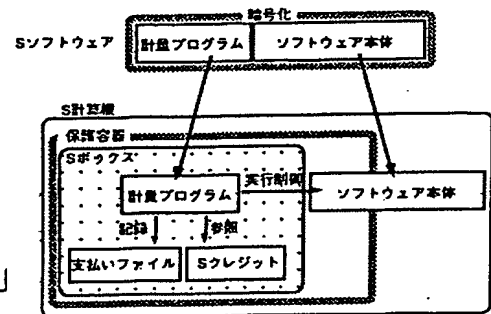


図2.1 SソフトウェアとS計算機

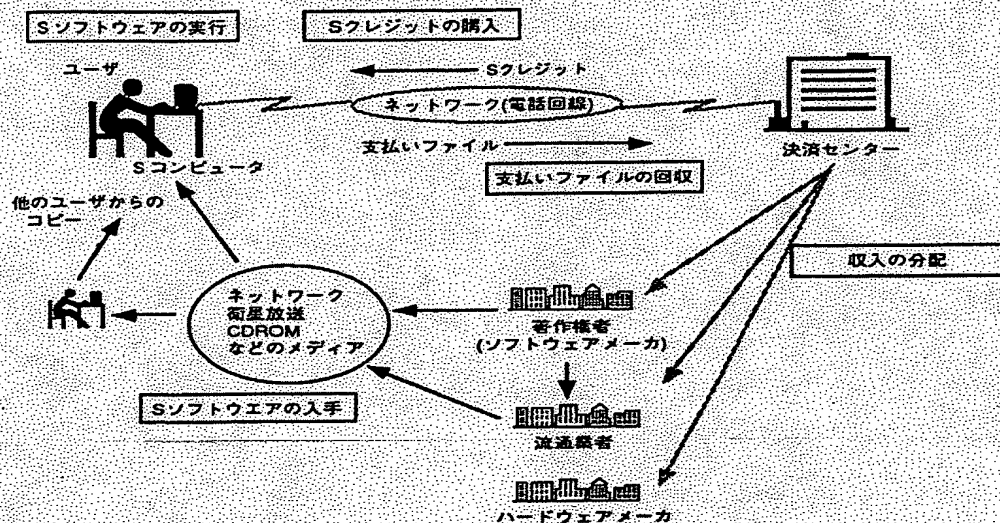


図2.2 超流通の全体像の一例

能を持つSボックス（文献2）を対象とする。この機能は、本論文で述べるような料金体系による高度なサービスをSdAで提供する時に必要となる。超流通の3つの条件を満足させるのみであれば、Sボックスはこの機能を持たなくてもよい。

SdAには、シンプルなサービスしか提供できないが実現が比較的容易であるものから、複雑ではあるが多種多様のサービスを提供できるものまで、様々な実現形態が考えられている。

本論文の対象とするSdAでは、ユーザはまずSクレジットを購入する。ソフトウェアはあらかじめ無料で入手しておくことができる。Sクレジットはどのソフトウェアにも共通に使用できるため、Sクレジットの購入時にどのソフトウェアを使用するかを決めておく必要はない。ソフトウェアを使い始めると、計量プログラムの記述にしたがってSボックスではSクレジットが減算され、支払いファイルが作成される。支払いファイルは定期的に回収され、Sクレジットの売上を著作権者に分配するための情報として使用される。回

収された後も支払いファイルの内の必要な項目はユーザの持つ権利を表すものとして半永久的に権利管理機構内に保管される。

3. 標準の料金体系

3.1 標準の料金体系の概要

本章で述べる標準の料金体系は、現在主流である買取り料金の他に、従量料金、無料試用、買取りに対する返金を含んでいる。また、従量料金を買取価格に繰込むことによって、従量料金で使用した後に買取っても大きく不利になるようなことがないようにしている。（表3.1参照）

買取り料金：Cr10000
従量料金：Cr1/10秒、 一回あたりの上限Cr1000
無料試用限度額：試用量がCr2000相当まで
100%繰込み限度：繰込み金額がCr5000まで 限度後の繰込み率は50%
買取りに対する返金
返金率：50%
返金期限：買取成立から3ヵ月後まで
返金取り消し期限：返金後1週間後まで
(CrはSクレジットの単位)

表3.1 料金の設定例

本論文では、標準の料金体系と4章の特別許諾とを含めたものを料金体系と呼んでいるが、本章では標準の料金体系のことを単に料金体系と呼ぶことにする。

以下の3.2～3.6節では、料金体系を構成する項目について述べる。そして、最後の3.7節で料金体系に対応した会計処理の例を示す。

3.2 無料試用

ソフトウェアを初めて使用する場合には、自動的に試用になり、ユーザが選択する必要はない。初めてであることはSボックスによって自動的に認識される。

著作権者は無料試用限度を定めることができるので、機能を限定しない試用を提供できる。

無料試用限度は、時間または回数により指定できる。また、試用量をSクレジットに換算して、Sクレジットの単位で指定することもできる。試用限度はゼロと指定してもよい。

3.3 従量料金

Sボックスはソフトウェアの使用を計量する。例えば、使用時間や使用回数である。Sボックスの会計処理では、使用量を参照しながら課金を行なえる。

回数と時間の両方の使用量に従う料金が従量料金である。この従量料金は、ソフトウェアの使用時間が少ない間は時間料金を課金し、回数料金に相当する1回当りの上限の料金になると、課金が停止する。この様にすることによって、ソフトウェアの使用前に、時間料金か回数料金を選択するために使用量を推定する必要がなくなる。

また、従量料金は、ソフトウェアの実行が途中で停止しては困るようなソフトウェアに対して有効である。例えば、病院で手術に用いるソフトウェアでは、n回の手術が終了するまでの回数料金を確保してから実行が開始するようにする。そうすれば、n回の実行が終了するまでは、Sクレジットの不足のため

にソフトウェアが停止する心配がなくなる。早く手術が終了した場合は、時間料金分だけ支払えばよい。

3.4 繰込みと買取り

料金体系では、従量料金と買取り料金をそれぞれ別々に考えるのではなく、両方の関係を考慮したものにする。すなわち、従量料金と買取りの関係を、繰込みによって関係付ける。ここで、買取りとは、買取価格相当の金額を支払うことにより、以降は使用時に料金を支払わないでよくなることをいう。

繰込みとは、従量料金の内の何パーセントかを買取り価格の支払いへの充当とみなすものである。繰込み金額の合計が買取り価格に達すると、買取りが自動的に成立する。

3.5 100%繰込みと50%繰込み

従量料金の何パーセントが繰込まれるかによって、買取りと従量の間に差が生じる。100%の繰込み率であれば、従量と買取りに差がないと言える。従量料金の繰込み率が100%の間は、買取りを選択しなくても、買取りが成立するまでに支払う金額が割高にならないからである。従量料金を全く繰込まないと、従量で支払い続けている限り、買取りが成立しない。

料金体系では、無料試用限度を超過すると、従量料金となるようにしている。そして、従量料金の最初の内は繰込み率を100%にする。これは、無料試用を終えても、ユーザがすぐに買取らなくても済むようにするためである。買取るどうかは、繰込み率が100%の従量料金を支払っている間に決めればよい。100%繰込みの間に買取ることを決めた場合は、買取り予約とする。買取り予約は、従量料金と買取りとに差が生じはじめる点で、自動的に買取りが、すなわち残額の一括払いが起こるようにするための

予約である。100%繰込みの間、ユーザは買取り予約とその取消しを任意に選択してよい。

著作権者にとっては、従量よりも買取ってもらった方が、収入を早く得られる。そこで、料金体系では、繰込み額が買取り価格に達する前に、次の様に買取りと従量に差がつくようにする。

繰込み額がある限度を越えたら、繰込み率を100%でなくする。この限度を100%繰込み限度と呼ぶことにする。100%繰込み限度後の繰込み率は、従量料金と買取りとのバランスを考慮すると、50%に設定するのがよいと考えている。これは、100%繰込み限度で買取った時に支払う金額が、100%繰込み限度から自動買取りが成立するまでに従量料金で支払う金額の合計の半分で済むような設定である。

50%繰込みで従量料金を支払っている間は、いつでも買取り申込みをすれば、その時点で買取りが成立する。その時に支払う金額は、買取り価格と、それまでの繰込み金額との差額である。100%繰込み限度の後における買取り申込み時の支払い金額は、50%繰込みで従量料金を支払っていくに従って少なくなっていく。最終的には、自動買取りとなって、ゼロになる。しかし、買取り申込みまでの従量の支払い額の合計と、買取り申込み時に支払った金額とを合せると、申込みが遅くなるほど割高になっていく。

3.6 返金

3.6.1 返金と買取り

返金は、買取りと従量料金に差がある時に買取りに対して適用する。従量料金は返金しない。

返金する額は、買取り時に支払った額と、買取後の使用量を従量料金で換算した額との差額に、返金率を掛けたものとする。また、返金率は100%以下とする。

バグの発見などによって返金をした場合は、返金後は全く使用しないのが普通である。し

たがって、返金後の料金はあまり意味を持たないように見えるかもしれない。しかし、場合によっては、返金後に使用する場合もありうる。そこで、そのような場合には、50%繰込みの時と同じ状態になり、従量料金を支払うか、再び買取れば使用できるようにする。再び買取る時には、買取る以前の従量料金に対する繰込み、および、買取り後に使用した分の繰込み金額を含めるようにする。

返金可能な状態の間は、いつでも返金を選択できるが、誤って返金を選択してしまった場合も考慮している。返金後にソフトウェアを使用していなければ、返金を取り消すことができる。すなわち、返金額を返済して再び買い取ることができる。

3.6.2 返金用クレジットと返金期間

2.2節で述べた、Sクレジットの売上げを著作権者に分配する仕組みにおいて、Sクレジットを返金できるようにする際には、返金するSクレジットと決済機関の収入支出の対応が適切になるように考慮しなければならない。

Sクレジットを返金する場合、その返金のための原資を決済機関が所有している必要がある。何となれば、返金したSクレジットが別のソフトウェアに使用され、その減算額が書込まれた支払いファイルが決済機関に回収された時には、その分のお金を別の著作権者へ分配するからである。

決済機関が返金のための原資を所有しないとすると、既に著作権者へ分配してしまったお金を返してもらわなければならない。しかし、著作権を所有する会社が倒産してしまったような時には、お金を返してもらえないこともあり得る。

以上のようなことを考慮して、Sクレジットの返金をSボックス上で行なうために、返金期間を設け、その間は返金用のSクレジット（返金用クレジットと呼ぶ）をSボ

買取り申込みがあれば、「買取り予約あり」の状態へ遷移する。この時、支払いファイルに買取り予約の記録を書き込む。

買取り予約の取消しがあれば、支払いファイルから買取り予約の記録を消去し、「買取り予約なし」の状態へ遷移する。

「買取り予約なし」の状態で100%繰込み限度を超過したら、50%繰込み従量の状態へ遷移する。100%繰込み限度は、例えば、100%繰込みによる繰込み金額の上限で指定される。

「買取り予約あり」の状態で100%繰込み限度を超過したら、(3)50%繰込み従量の状態の「買取り申込みがあった時の処理」と同じ処理を行ない、返金あり買取りの状態へ遷移する。

ソフトウェアの実行終了時には、繰込み金額とSクレジットの減算額を支払いファイルに書込む。

(3)50%繰込み従量の状態

この状態では、買取り申込みがあるかどうかを識別し、あれば以下の処理を行なって、返金可能買取りの状態へ遷移する。

[買取り申込みがあった時の処理]

買取り申込みがあった時の買取り額は、

$(\text{買取り価格}) - (\text{繰込み額})$

である。この金額分のSクレジットを減算する。

$(\text{買取り時支払い額}) \times (\text{返金率})$ 分の金額は、返金用クレジットとなり、Sボックス内に保持する。ここで、返金率を掛けると端数が出る場合がある。端数を切り捨てると返金の際にユーザが端数分損をすることになる。そこで、ユーザへの好意的処理として端数分は切り上げればよい。

$(\text{買取り時支払い額}) - (\text{返金用クレジット})$ 分の金額は、支払いファイルに書込まれる。

繰込み金額は支払いファイルにある額をそのまま保持しておく。これは、返金があり、その後、50%繰込みの従量の状態に遷移した

時に用いる。

買取り申込みがない時には、以下の処理を行なう。

[買取り申込みがない時の処理]

Sクレジットを従量料金で減算する。同時に、Sクレジットの減算額の50%を繰込み金額に加算する。繰込み金額は小数点以下も含む。買取り申込みがあった時には、ユーザへの好意的処理として、繰込み金額の小数点以下を切上げる。

繰込み金額が買取り価格を越えたら、返金なし買取り状態へ遷移する。

ソフトウェアの実行終了時には、繰込み金額とSクレジットの減算額を支払いファイルに書込む。

(4)返金可能買取りの状態

使用量を従量料金で計量する。

返金期間内で、かつ、返金申込みがあれば、返金額となる、

$(\text{返金用クレジット}) -$

$(\text{使用量}) \times (\text{返金率})$

を計算する。返金額が0より大きければ、その分のSクレジットを返金する。そして、返金取消し可能の状態へ遷移する。

$(\text{使用量}) \times (\text{返金率})$ で端数が出た場合は、ユーザへの好意的処理として端数分は切り捨てる。

ソフトウェアの実行終了時には、 $(\text{使用量}) \times (\text{返金率})$ の金額分が返金用クレジットに残っていれば、その額を支払いファイルに書込む。そして、返金用クレジットからその額を引く。

残っていなければ、返金用クレジットの全額を支払いファイルに書込み、返金用クレジットをにする。

返金用クレジットが0になって買取り後使用量が超過したが、または、返金期間（例えば、3ヵ月）を過ぎた時には、返金なし買取りの状態へ遷移する。

(5) 返金取消し可能の状態

返金取消し期間内（例えば1週間）で、かつ買取り申込みがあれば、返金した分のSクレジットを減算する。減算額分を返金用クレジットとし、返金可能買取りの状態へ遷移する。従量使用を開始するか、または、返金取消し期限を経過したならば、50%繰込み従量の状態へ遷移する。

(6) 返金なし買取り状態

Sクレジットと支払いファイルの更新は行なわない。

4. 特別許諾条件

4.1 特別許諾の定義

社会には、所属・身分、社会的な行事・出来事などによって、料金に変化（通常は割引き）するものがある。SdAでは、このような料金体系も実現できる。

これらの料金体系を一言でいうならば、特定の条件の下で、標準的な体系と異なる許諾を提供している、ということができる。この条件のことを「特別許諾条件」、そして提供される許諾のことを「特別許諾」という。特別許諾条件および特別許諾の内容は、計量プログラムに記述される。

特別許諾条件が計算機外部の情報に依存する場合には、それを何等かの形で計算機にインストールする必要がある。その形を「許諾コード」という。許諾コードを記憶するためのメモリを「許諾コード用メモリ」と呼ぶ。コードは、大きく「ローカルなコード」と「グローバルなコード」に分類できる。「ローカルなコード」は著作権者ごとに異なるコードで、著作者以外の者はこのコードを参照できない。コードの発行者は、通常、著作権者本人である。他の者が参照できないようにする理由は、例えばあるメーカAが社員割引きのコードを設定したときに、メーカBがそのコードを利用して、メーカAの社員に対して割増し料金をとるように料金体系を記述で

きてしまうからである。

「グローバルなコード」は、全ての著作権者が共通に参照できるコードである。グローバルなコードの発行は、全ての著作権者の承認を得た機関が行なう。

4.2 特別許諾の種類

4.2.1 計算機環境に依存する許諾

計算機環境に応じて許諾の内容を変えられるものである。例えば、教育機関で使用されている計算機に「教育機関認定」という許諾コードをインストールする。ソフトウェアの実行時に、この許諾コードを確認することによって、割引きなどのサービスが行なわれるようにする。

また、計算機が特定のユーザに使用されるような環境においても、同様にして、ユーザIDを書込んだ許諾コードを計算機にインストールすればよい。特別許諾を与えたユーザの識別には、個人識別用の情報が書込まれているカードを用いることができる。

4.2.2 イベントに依存する許諾

「イベント」に依存する許諾は、社会的な出来事によって許諾条件が変わるものである。例えば、「プロ野球の球団Aが優勝」というようなイベントがあった場合には期間を定めて料金を割引くサービスを提供し、なかった場合には割引かない、といった許諾ができる。

イベントに依存する許諾を提供する許諾コードをインストールするための信号を、「トリガー信号」と呼ぶ。トリガー信号をできるだけ広範囲のユーザに伝えるためには、信号の送信元とユーザが1対1の通信ではなく、放送が利用されるだろう。トリガー信号を計算機に入力するかどうかはユーザに任される。

このような場合に、一般に通信の同時性は保証されない。そこで、信号が入力された時間によらずに同じ時間にサービスが行

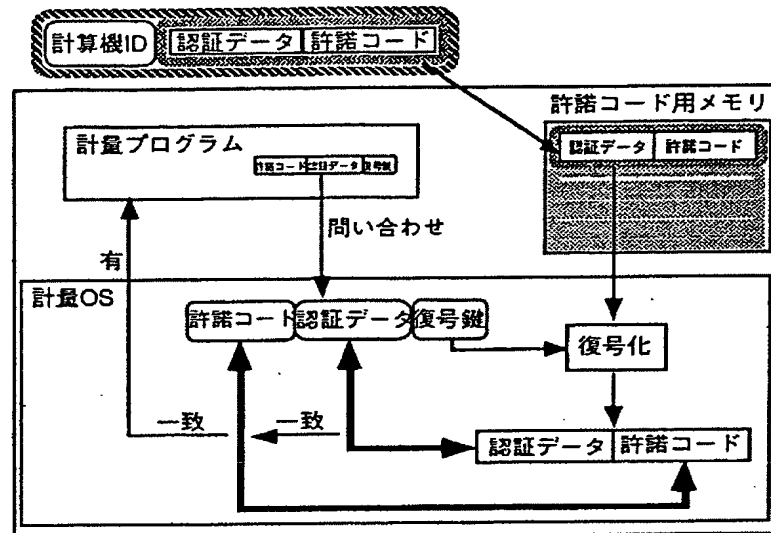


図4.1 許諾コードのインストールと参照

われるように、信号の内部に特別許諾の開始時刻、終了時刻を記述しておくことができる。

「クリスマスの前の1週間だけ割安になる。」、または、「指定した時間にならないと実行できない。(発売時刻指定)」などのような場合は、イベントの内容と時刻があらかじめわかっているの、トリガー信号は不要である。計量プログラムにそのような条件をはじめから記述しておけばよい。

4.3 許諾コードのインストール

許諾コードが不正にインストールされないようにするためには、次の二条件を満足する必要がある。

- (1) 偽造・改変ができないこと。
- (2) 計算機を指定した場合には、他の計算機にインストールできないこと。

前者を満足するには、全体がシステムの共通暗号鍵で暗号化されていけばよい。一般のユーザが、Sソフトウェアを作成できるようなシステム構成である場合には、誰にでもローカルな許諾コードが作成できることになり、共通鍵による暗号化だけでは不十分となる。グローバルなコードの場合には、復号化

鍵をあらかじめSボックス内部に入れておくことにより、専用の暗号鍵による暗号化が利用できる。ローカルなコードの場合には、許諾コードを参照するときに発行者の認証を行なう必要がでてくる。詳しくは4.4節を参照のこと。

後者を満足するには、インストール時に計算機IDを用いて認証を行なえばよい。計算機IDを得る段階で、虚偽の報告がなされないようにする必要があるならば、コードの発行者の承認を得た機関・人物が立合えばよい。

以上の2つの条件を満足するならば、実際のインストールは純電子的に行なうことが可能となる。この時に計算機に入力されるのは、許諾コードをインストールするためのプログラムでもよいし、許諾コードだけでもよい。後者の場合、許諾コードのインストールの機能は、権利管理機構の一部として実現されている必要がある。

4.4 許諾コードを参照する時の会計処理

特別許諾をサポートする計量プログラムに対して、次のような会計処理を行なえば

よい。

(図4.1参照)

- (1)標準の料金体系を使用するように設定する。
- (2)計量プログラムが対応している許諾コードの有無を調べるために、支払いファイルを検索し、存在する場合には対応した料金体系になるように設定する。
- (3)ソフトウェアの実行を開始する。

ユーザがSソフトウェアを開発できるようにシステム構成の場合には、ローカルな許諾コードも作成することができる。許諾コードが偽造されたものでないことを確認するためには、(2)の問い合わせの段階で、発行者と著作権者のみが知る情報を利用した認証手续をとる必要がある。このような場合の、(2)の段階における一つの問い合わせに対するSボックスの動作は次のようになる。

- (1)コード'Cx'はあるか、という問い合わせがくる。
- (2)許諾コード用メモリのなかの、グローバルなコードの領域を検索し、存在すれば'YES'を送信。
- (3)グローバルな領域になかった場合には、計量プログラムに対して、復号鍵と認証用のデータを要求する。許諾コード用メモリのローカルコード領域のコードでソフトウェアIDが同じコードを順に復号鍵で復号し、認証データおよび許諾コードの一致を確認する。確認できたら'YES'を送信。
- (4)それ以外'NO'を送信。

5. 結論

本論文では、超流通におけるサービスの可能性を検討するために料金体系を取上げ、その内容と、権利管理機構のSボックスにおけるそれに対応した会計処理を示した。

本論文で示した料金体系が実現される

と、無料試用、従量料金、買取り、繰込み返金によって、ユーザは快適に使用量に応じて料金を支払えるようになる。また、特別許諾によって、割引きなどサービスも受けられるようになる。そして、著作権者はこのようなサービスを提供することによってソフトウェアの売上げが増えるようになるであろう。

参考文献

- 1)森亮一、田代秀一：ソフトウェア・サービス・システム (SSS) の提案、電子情報通信学会論文誌, VolJ70-D No.1, pp.70-81,1987
- 2)森亮一、田代秀一：ソフトウェア・サービス・システム (SSS) の小規模な試作、同上, VolJ70-D No.2 pp.335-345,1987
- 3)日本電子工業振興協会：マイクロコンピュータに関する調査報告書[II], 第3部 ソフトウェア流通の為の基盤技術, 89-バ-5, pp.107-221, 1989年3月
- 4)超流通アーキテクチャSdA報告書, 筑波大学森亮一研究室, 1988,1989
- 5)森亮一、河原正治、篠崎雅英、大瀧保広：「超流通」研究計画の現在、利用者指向の情報システムシンポジウム論文集, pp.173-180, 1989年12月

情報処理学会研究報告
©情報処理学会 1990

情処研報 Vol. 90, No. 1
1990年1月16日発行

発行所 〒106 東京都港区麻布台2-4-2 保科ビル

社団法人 情報処理学会 TEL 東京(03)505-0505(代表)
郵便振替口座 (東京5-83484)

発行人 社団法人 情報処理学会 櫻間 卓

Technical Report
IPSJ Vol 90 No. 1

Information Processing Society of Japan Technical Report

90-IS-27
16 January 1990

Information Processing Society of Japan

Information systems 27-5
(16 January 1990)

The Accounting Process in Software Usage Monitor for Superdistribution

Shinichi Ueki
Master's degree program in sciences and engineering, University of Tsukuba
now SONY Inc

Yasuhiro Ohtaki
Doctoral degree program in engineering, University of Tsukuba

Ryoichi Mori
Institute of Information Science and Electronics, University of Tsukuba

Research is going forward to realize "superdistribution" by adding a software usage monitor [rights control mechanism] to personal computers. Beyond protecting the rights of software authors, the copyright control mechanism has the potential to provide users services not hitherto available. Because there is no practical next-generation technology for rights control, it is necessary to give detailed study to what kinds of services will be possible.

This paper describes the major points of a charge system and the accounting process that handles the software usage monitor in one general example. The charge system discussed here includes free trials, per-use charges, purchases and refunds. Further, the system can be set up for specified computers or social events.

1. Introduction

The term “superdistribution” refers to free distribution of paid software when the following conditions are met.

Condition 1: The user is able to obtain the software at any time, without charge, from a source of his choice. The software obtained can be used at any time under the terms set by the copyright holder (next condition). There is no difficulty in obtaining or using the software.

Condition 2: The copyright holder can set the terms under which the software is used.

Condition 3: Modification of the software thus distributed, beyond what is specified by the copyright holder, is prevented.

The superdistribution architecture (SdA) is that which has been advocated by Ryoichi Mori as a physical means to realize superdistribution.

Under the second condition of the definition of superdistribution, the copyright holder can set charges. These charges are protected by the third condition. Accordingly, the copyright holder need not be concerned with reducing the number of users who used the software without paying the charge. In superdistribution, the charges will probably be set in accordance with the desires of users, as much as possible, in order to maximize the use of the software.

The user will of course want the price of the software to be inexpensive, but beyond that, he will want to know whether it has enough value to justify paying the charge; if the software will be used infrequently, he will want to pay a charge proportional to his use; and if he knows from the start that he will use it frequently, he may wish to purchase it outright.

The copyright holder, moreover, may think it good to provide services such as discounting the charge to specified computers, perhaps “educational institutions,” or in connection with social events, such as “one week prior to Christmas,” if that would increase sales.

This paper takes up a charge system that would provide the content described above. It also describes an accounting process in which the software usage monitor for superdistribution added to computers handles that charge system

2. Superdistribution Architecture (SdA)

2.1 Basic Vocabulary of SdA

The basic vocabulary thought necessary to understand SdA is explained here.

- **S Credit:** Electric currency that can be used within the software usage monitor to pay software charges.
- **Metering Program:** The conditions for the copyright holder to authorize execution of the software are described, and so they are added to the software that is distributed by superdistribution. The charge system is usually described.

- Software Usage Monitor (S Box): A mechanism that manages the accounting process and execution of the software, as described in the metering program. It is added to the computer, and is placed under adequate physical controls.
- Payment File: A record created in the accounting process in the S box for each piece of software.

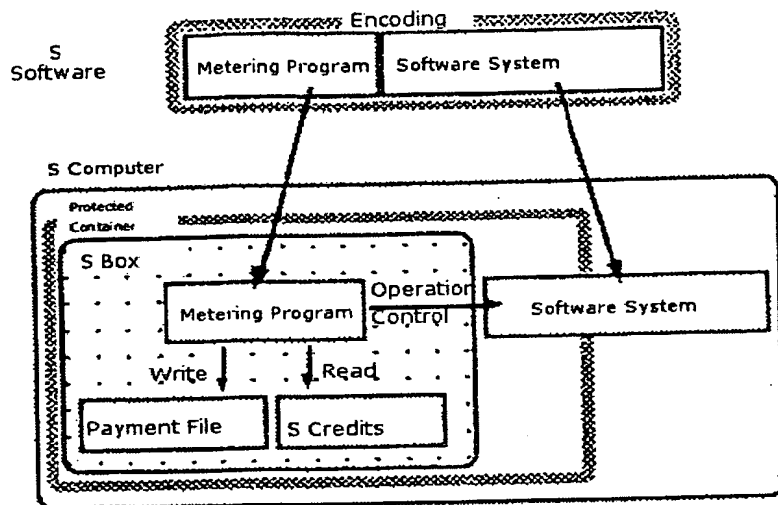


Fig 2.1 S Software and S Computer

2.2 Example of General Image of SdA

This paper deals with an S Box (document 2) that has the function of holding the payment file. This function is necessary for the SdA to provide high-level services by means of the charge system described in this paper. An S Box does not need to have this function, as long as it meets the three conditions of superdistribution.

A variety of modes for realization of SdA can be conceived, from

those that provide only simple services but are relatively easy to realize to those that are complex but offer many diverse services.

In the SdA that is the object of this paper, the user purchases S credits first. The software can be obtained free of charge in advance. Because S credits can be used in common for any software, it is not necessary to decide which software will be used when the S credits are purchased. When use of the software begins, S credits are debited within the S box in accordance with what is recorded by the metering program, and a payment file is created. The payment file is collected periodically, and the information is used to allocate S credit sales to copyright holders. Following collection, necessary data within the payment file is stored semi-permanently in the software usage monitor as an expression of the rights held by the user.

3. Standard Charge System

3.1 Outline of a Standard Charge System

The standard charge system described in this chapter contains, in addition to the purchase charges that are the primary item at present, per-use charges, free trials, and purchase refunds. Further, including per-use charges in the purchase price avoids the great disadvantage to those who purchase the software after using it on the basis of per-use charges. (see Table 3.1)

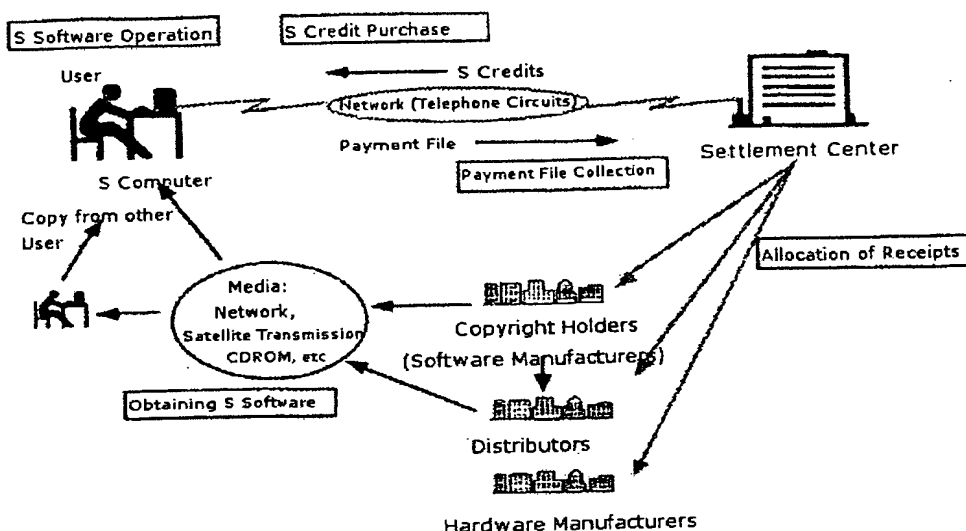


Fig 2.2 Example of General Image of Superdistribution

What is called the charge system in this paper includes both the standard charge system and the special permissions of chapter 4, but in this chapter the standard charge system is just called the charge system.

Paragraphs 3.2 through 3.6 below describe the items

that constitute the charge system. Then, an example of the accounting process that handles the charge system is shown in paragraph 3.7.

Purchase charge	Cr10,000
Per-use charge	Cr1/second (limit of Cr1,000 per time)
Free trial limit	Until trial use is equivalent to Cr2,000
100% inclusion limit	First Cr5,000, after which 50% is included
Refund of purchase charge	
Refund percentage	50%
Refund period	3 months from date of purchase
Refund cancellation period	1 week after refund
Cr is unit of S credits	

Table 3.1 Example of Setting Charges

3.2 Free Trial Use

When the software is first used, it is automatically deemed trial use with no need for a choice by the user. The fact that it is the first use is recognized automatically by the S box.

Because the copyright holder is able to determine the limit of free trial use, he is able to provide trial use with unlimited functionality.

The limit on free trial use can be specified by a period of time or a number of uses. Further, by converting the amount of trial use into S credits, it is possible to specify the limit in S credit units. And the limit on trial use can be specified as zero.

3.3 Per-use charges

The S box meters the use of software. For example, it meters the time used or the number of uses. In the accounting process of the S box, charges are imposed with reference to the amount of usage.

Charges that depend on the amount of usage, either in time used or times used, are per-use charges. The per-use charges are levied as time charges when the software is used for short periods, and stop accruing after reaching an upper limit for one use that corresponds to the charge for times used. Using that method makes it unnecessary for the user to decide, before using the software, to estimate high usage pattern in order to select charges by time or by times used.

Further, per-use charges are effective for software of which early termination of execution would cause problems. For example, in the case of software used in hospitals during surgery, use would begin only after a per use charge had been assured for completion of up to n surgical operations. That way, until the n operations had been completed, there would be no need to terminate use of the software on account of inadequate S credits. In the event the surgery was completed early, it would be possible to pay only for the time used.

3.4 Inclusive Purchase

In the charge system, per-use charges and purchase charges are not considered individually, but rather in terms of the relationship between the two. That is, the per-use charges and the purchase charge are related by inclusion. The meaning of "purchase" here is payment of an amount equivalent to the purchase price so that no further charges need be paid for time used.

The meaning of "inclusion" is application of some percentage of the per-use charges to payment of the purchase price. When the total of the included charges reaches the purchase price, the purchase is accomplished automatically.

3.5 100% Inclusion and 50% Inclusion

The difference between purchase and per-use payment depends on what percentage of the per-use charges is included. If the inclusion rate is 100%, it can be said there is no difference between purchase and per-use payment. That is because as long as the rate of inclusion of per-use charges is 100%, the amount paid until purchase is accomplished is no higher than required for a straight purchase. If the per-use charges are not included at all, purchase will not be accomplished no matter how long per-use payments continue.

In the charge system per-use charges begin once the limit of free trial use is passed. Thus, the initial rate of inclusion of per-use charges is 100%. That is because the user does not need to purchase immediately when the free trial use has ended. He can decide whether to purchase or

not while paying per-use charges with an inclusion rate of 100%. If he decides to purchase within the 100% inclusion period, he will subscribe to a purchase. The purchase subscription will result in an automatic purchase at the point where there begins to be a difference between per-use charges and purchase; in other words, it is a subscription to pay off the remaining amount. During 100% inclusion, the user is free to choose between a purchase subscription and cancellation.

From the perspective of the copyright holder, receipts will be available sooner from purchases than from per-use payments. Therefore, he will create a difference between purchase and per-use payments, in the following way, before the include amount reaches the purchase price.

The inclusion rate ceases to be 100% once the included amount passes a certain limit. This limit is called the limit of 100% inclusion. Considering the balance between per-use charges and purchase, it would be best to set the inclusion rate at 50% after the limit of 100% inclusion is passed. That value is chosen so the amount paid to purchase when the limit of 100% inclusion is passed will be half the total of per-use charges remaining to be paid before automatic purchase is accomplished.

If a purchase request is eventually made during the payment of per-use charges at the 50% inclusion rate, the purchase will be accomplished at that point. The amount to be paid at that time will be the difference between the purchase price and the total of included payments up to that time. The amount paid at the time of a purchase request made after the limit of 100% inclusion will be smaller, in accordance with the payment of per-use charges under the 50% inclusion rate. Ultimately, in the case of automatic purchase, it will be zero. However the total of per-use charges made up to the time of the purchase request will be higher in proportion to the lateness of the request.

3.6 Refunds

3.6.1 Refunds and Purchases Refunds apply to purchases when there is a difference between the purchase and per-use charges. Per-use charges are not refunded.

The amount refunded is found by multiplying the refund rate times the difference between the amount paid at time of purchase and the per-use charges for usage since purchase. The refund rate will not exceed 100%.

If refunds are made for reasons such as discovery of a bug, it is normal not to use the software at all after the refund. Consequently, there may not appear to be much reason for charges after a refund. Depending on the circumstances, however, there is a possibility of use after a refund. In such circumstances, therefore, it will be possible to use the software by paying the per-use charges or by repurchasing, just as during a period of 50% inclusion. In the event of repurchase, per-use charges prior to purchase will be included as well as the inclusion portion of charges for use after purchase.

It is conceivable that a refund will be selected at any time while refunds are possible, and that a refund will be selected mistakenly. It will be possible to cancel the refund if the software has not been used after the refund. That is, it is possible to repurchase the software by returning the refunded amount.

3.6.2 Refund Credits and Refund Period.

In the arrangement for distributing S credit receipts to copyright holders mentioned in paragraph 2.2, it is necessary to consider appropriate handling of refunded S credits and of receipts and disbursements by the settlement institution in the event that S credits are to be refunded.

When S credits are refunded, the settlement institution must possess funds for the refund. That is because the refunded S credits can be used for other software, and when the payment file on which that debit has been written is collected by the settlement institution, that amount of money is distributed to the other copyright holder.

If the settlement institution does not possess the funds for the refund, it becomes necessary to retrieve money that has already been distributed to the copyright holder. When the company that holds the copyright is bankrupt, however, it may not be possible to retrieve the money.

One possible method, in consideration of such a possibility, would be to establish a refund period for making S credit refunds to S boxes and leave the S credits used for refunds (called "S credits for refunds") in the S box for that period. Then the settlement institution would not distribute the refund credit portion of the money to the copyright holder until that period had passed. Under this method, the funds for refunds would be present in the settlement center during the refund period, and there would be no problem even if there was a change of payee for S credit income for refunds.

The credits used for refunds would be debited in accordance with per-use charges when the software was used after purchase, and the amount would be written to the payment file. And the amount of credit used for refunds stored in the S box would be written to the payment file when the refund period had passed. By collection of these payment files, the income would become available to the copyright holder if there had been no refund.

In the event of a refund, the sale corresponding to the refund would be cancelled in the S box. The settlement institution would simply distribute money to the copyright holder on the basis of the payment file collected.

The refund period in the S box would be set for perhaps three months from the date of purchase. This could be set in the SdA in consideration of the cycle for collection of payment files so copyright holders could obtain income, or it could be set in the metering program as desired by the copyright holder. The point at which the refund period begins is the time at which the purchase is accomplished. In the event of a refund during the refund period followed by repurchase prior to further per-use charges, there will be no change to the point at which the

refund period begins. That is, the point at which the refund period begins would remain the time of the original purchase.

3.7 Accounting Process for the Charge System

Figure 3.1 is a flow chart for the charge system. The state in the charge system is called the "charge status."

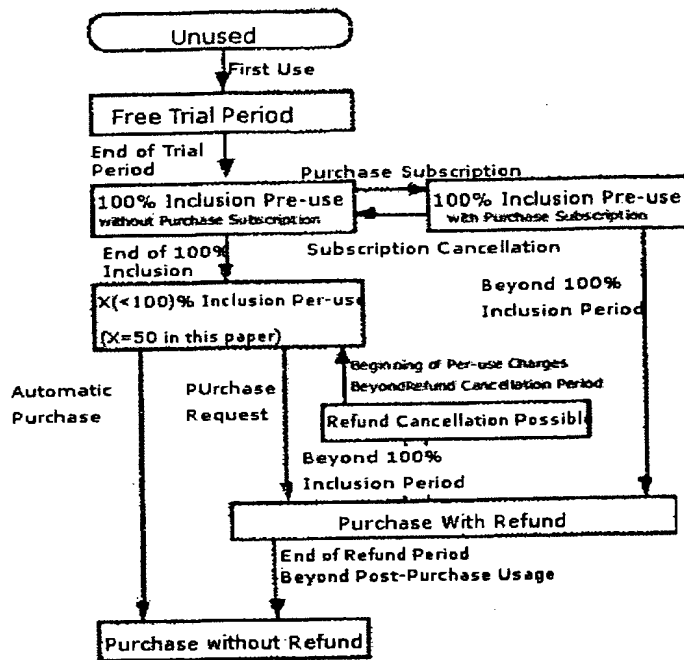


Fig. 3.1 Change of Status in Charge System

When the software is started up, it will be in one charge status or another. This status will change in accordance with usage of the software or the user's choice. Thus, when execution of the software ends, the charge status will be held in the payment file in the S box.

Examples of charge statuses in the S box accounting process are shown below.

(0) Unused Status

When the software is used, a payment file is produced for that software, and the status changes to free trial use.

(1) Free Trial Use Status

Trial use is metered, and when the limit of free trial use is exceeded, the status changes to 100% inclusion per-use.

The trial use will be metered, perhaps on a per-use basis. At this time, the limit of free trial use is in S credit units.

In this status, S credits are not debited.

When execution of the software ends, trial usage is written to the payment file.

(2-1) 100% Inclusion Per-Use Status (without Purchase Subscription)

(2-2) 100% Inclusion Per-Use Status (with Purchase Subscription)

In the 100% inclusion per-use status, S credits are debited as per-use charges whether or not there is a purchase subscription. At the same time, an inclusion amount is added that is equal to the amount of S credits debited.

If there is a purchase request, the status changes to "with purchase subscription." A record of the purchase subscription is written to the payment file at that time.

If the purchase subscription is canceled, the record of the purchase subscription is erased from the payment file and the state changes to "without purchase subscription."

If the 100% inclusion limit is passed while in the "without purchase subscription" status, the status changes to 50% inclusion per-use. The limit for 100% inclusion is defined, for example, by designating an upper limit on the amount that can be included.

If the 100% inclusion limit is passed while in the "with purchase subscription" status, the status changes to purchase with refund by the same process used when a purchase request is made in (3) the 50% inclusion per-use status.

When execution of the software ends, the inclusion amount and the S credit amount to be debited are written to the payment file.

(3) 50% Inclusion Per-Use Status

In this status, a distinction is made between whether a purchase request exists or not; if so, the following process is performed and the status changes to purchase with refund possible.

[Process when a purchase request exists]

The purchase amount when there is a purchase request is (purchase price) – (inclusion amount). S credits are debited in this amount.

An amount equal to (amount paid at purchase) x (refund rate) becomes the credit for refund use, and is held in the S box. Sometimes, multiplying by the refund rate will produce a fraction. If the fraction is rounded down, the user will lose the fractional part in the event of a refund. It is better, therefore, to round up the fraction as a process that favors the user.

An amount equivalent to (amount paid at purchase) – (credits used for refund) is written to the payment file.

The inclusion amount is stored as a certain amount in the payment file. This is used during the change to the 50% inclusion per-use status following any refund.

The following process is used when there is no purchase request.

[Process when no purchase request exists]

S credits are debited as a per-use charge. At the same time, 50% of the value of the debited S credits is added to the inclusion amount. Decimal portions of the inclusion amount are included as well. When there is a purchase request, the decimal portion of the inclusion amount is rounded up as a process that favors the user.

When the inclusion amount exceeds the purchase price, there is a change to the purchase without refund status.

When execution of the software is completed, the inclusion amount and the amount of debited S credits are written to the payment file.

(4) Purchase with Refund Possible Status

Usage is monitored as per-use charges. If a refund is requested during the refund period, the amount is calculated as (credits for refund) – (usage) x (refund rate). If the refund amount is greater than zero, S credits are refunded in that amount, and the status changes to “refund cancellation possible.”

In the event that (usage) x (refund rate) produces a fraction, the fractional portion is dropped as a process favorable to the user.

When execution of the software is completed, the amount of (usage) x (refund rate) is written to the payment file if that much remains in credits for refund use. Then the amount is subtracted from the credits for refund use.

If that amount does not remain, the full amount of credits for refund use is written to the payment file, and it becomes credits for refund use.

When the credits for refund use reach zero and the post-purchase usage is exceeded, or when the refund period (three months, for example) has passed, there is a change of status to “purchase without refund.”

(5) Refund Cancellation Possible Status

If there is a cancellation request within the refund cancellation period (perhaps one week), the refunded S credits will be debited. The debited amount becomes credits for refund use, and the status changes to “purchase with refund possible.” If per-use charges begin or if the refund cancellation period has passed, the status changes to “50% inclusion per use.”

(6) Purchase without Refund Status

S credits and the payment file are not renewed.

4. Conditions for Special Permissions

4.1 Definition of Special Permissions

Charges sometimes change (usually by discounts) because of affiliation, personal status, or social functions or events. Such a charge system can be realized within SdA.

These charge systems are, briefly put, capable of providing permissions under special conditions that differ from those of standard systems. These conditions are called "conditions for special permissions" and the permissions provided are called "special permissions." The content of the conditions for special permissions and of the special permissions is specified in the metering program.

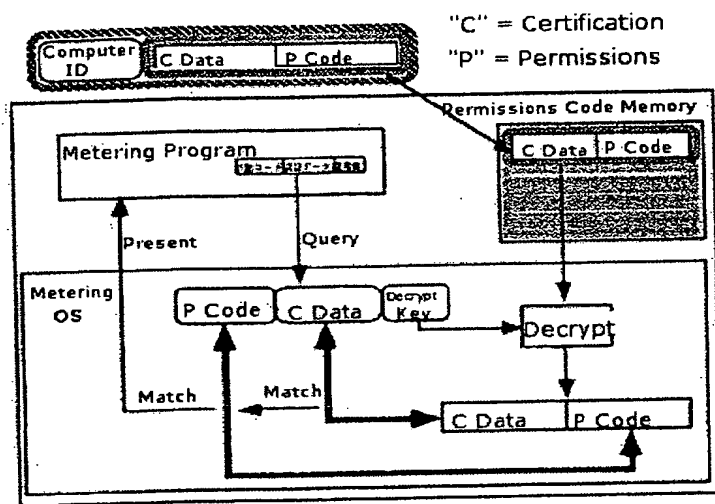


Fig. 4.1 Permissions Code Installation and Reference

In the event that the conditions for special permissions depend on information outside the computer, it is necessary to install them inside the computer in some form. That form is called the "permissions code." The memory in which the permissions code is stored is called the "permissions code memory." The code can be largely classified as "local code" and "global code." Local code is code that differs by the copyright holder; it cannot be seen by persons other than the copyright holder. The party issuing the code is normally the copyright holder

himself. The reason other persons are unable to see the code is that, otherwise, when manufacturer A set a code for company discounts, manufacturer B could use that code to describe a charge system to increase charges to personnel of manufacturer A.

Global code is available to all copyright holders in common. Global code is issued by institutions that have been authorized by all copyright holders.

4.2 Types of Special Permissions

4.2.1 The content of permissions changes in response to the computer's environment. For example, a permissions code called an "educational institution certificate" is installed in computers used by educational institutions. When the software is executed, it can perform discounted services by confirming the presence of that code.

Moreover, in an environment where a computer is used only by specified users, it is possible to install a permissions code in the computer in which the user's ID has been written. Users who have been given special permissions can be distinguished by use of a card on which individual identification information has been recorded.

4.2.2 Event-Dependent Permissions

In the case of permissions that depend on events, the conditions for permissions can be changed in line with social happenings. For example, it is possible to create permissions such that discounted services will be provided for a period in the event of "victory by professional baseball team A," with no discount in the event of a loss. The signal used to install permissions codes that provide event-dependent permissions is known as a "trigger signal." In order to convey the trigger signal to as broad a range of users as possible, a broadcast is used instead of one-on-one communications between the issuer and the user. Whether or not to input the trigger signal into the computer is left up to the user.

In such a case, it is generally not possible to assure simultaneous communications. Therefore, in order that the users can use the service at the same time, no matter when they input the signal, start and stop times for the special permissions can be specified within the content of the signal.

In cases such as "a discount offered during the week before Christmas" or "execution not possible except during the permitted period (start time designation)," the content and time of the event are known in advance, and so no trigger signal is required. It is enough to state such conditions in the metering program from the beginning.

4.3 Installation of Permissions Code

The following two conditions must be met so that the permissions code is not installed improperly.

- (1) Code cannot be counterfeited or modified.
- (2) If the computer is designated, the code cannot be installed in other computers.

If the first is satisfied, encryption of the whole can be done with the system's common encryption key. Any ordinary user is able to create a local permissions code if the system structure is such that S software can be created, and encryption with a common key is inadequate. In the case of global code, the encryption key is placed in the S box in advance, so that encryption can be done with a dedicated encryption key. In the case of local code, it is necessary to certify the issuer's identity for observation of the permissions code. Please see paragraph 4.4 for details.

If the second condition is satisfied, it is sufficient to certify use of the computer ID during installation. If necessary to prevent false reports, it is enough to have an institution or person authorized by the issuer present at the stage where the computer ID is obtained.

As long as the two conditions stated above are satisfied, the actual installation can be done completely electronically. All that is input to the computer at that time is the program to install the permissions code, or just the permissions code. In the latter event, the installation of the permissions code will have to be implemented as part of the software usage monitor.

4.4 Accounting Process During Observation of Permissions Code

The following sort of accounting process should be applied to a metering program that supports special permissions. (see figure 4.1)

(1) The process is set to use the standard charge system.

(2) In order to check the presence of permissions codes applied by the metering program, the process is set to inspect the payment file and to become a charge system applying the codes if they are present.

(3) The process commences execution of the software.

In the event of a system structure such that users are able to develop software, it will also be possible to create local permission codes. In order to confirm that the permissions codes have not been counterfeited, it is necessary to adopt a verification procedure, at the stage of the query in (2) above, that uses information known only to the issuer and the copyright holder. The S box operates as follows in response to a query in stage (2) in such cases.

(1) The query, "Does code 'Cx' exist?" arrives.

(2) The global code domain in the permissions code memory is inspected, and "Yes" is sent back if the code is present.

(3) If it is not present in the global domain, the metering program is asked for the decryption key and confirmation data. Code with the same software ID in the local domain of the permissions code memory is decrypted using the decryption code, and the confirmation data and the permissions code are checked for a match. If they match, "Yes" is sent back .

(4) Otherwise, "No" is sent back.

5. Conclusion

This paper has taken up charge systems for the purpose of studying the possibility of services in superdistribution, and has shown their content and an accounting process to handle that content in the S boxes of software usage monitors.

If the charge system shown in this paper is realized, users will be able to pay appropriate charges for usage in regard to free trials usage, per-use charges, purchases, inclusion, and refunds. Further, it will be possible to receive discounted services by means of special permissions. Accordingly, copyright holders will be able to increase sales of software by providing such services.

Reference Documents

1. R. Mori and S. Tashiro, The concept of Software Service System (SSS), Trans. IEICE, Vol. J70-D, No. 1, pp.70-81, 1987.
2. S.Tashiro and R. Mori, The implementation of a small scale prototype for Software Service System (SSS), Trans. IEICE, Vol. J70-D, No. 2, pp. 335-345, 1987.
3. Japan Electronics Industry Development Association, Research Reports on Microcomputers Vol II part 3, Basic Technology for Software Distribution, pp. 107-221, 1989.
4. Superdistribution Architecture (SdA) Reports, Ryoichi Mori Research Office, Tsukuba University, 1988, 1989.
5. R.Mori, M. Kawahara, M. Shinozaki, and Y. Ohtaki, Today's Research Plan for "Superdistribution," Collected Papers of the Information Systems Symposium for Users, pp. 173-180, 1989.

**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.

超編集・超流通・超管理のアーキテクチャ
シンポジウム 論文集



平成6年2月17日, 18日

 社団法人 情報処理学会

Information Processing Society of Japan

BEST AVAILABLE COPY

歴史的必然としての超流通

森 亮一† 河原正治†
† 筑波技術短期大学

概 要

1983年に超流通が森によって発明されてから今日に至るまでの研究の概要を示す。また、デジタル情報の円滑な流通を実現し、ソフトウェアの量産と部品化・再利用を促進するためには超流通の実現が不可欠であることを解説する。物々交換および貨幣経済への移行が人類にとって必然だったのと同じ意味で、超流通社会が必然であることを示す。

1 超流通

1.1 超流通とは？

超流通とは、デジタル情報の円滑な流通を実現することにより、これまで不可能と思われていた「ソフトウェアの量産」を可能にするための基盤技術である。デジタル情報を「所有する」のではなく、利用するたびに使用記録が管理され、それを回収することによって料金を徴収し収入を再分配するシステムである。

未来指向のデジタル情報の流通・供給システムは、次のような性質を満たすべきである。すなわち、

- 低コストで、大量かつ安全に供給できる。
- 情報提供者の利益は保証される。
- 情報利用者の犠牲を伴わない。

デジタル情報の流通・供給システムとしての「超流通」を以下のように定義する。

超流通とは、次のような性質を満たすものである。

- (1) 情報利用者は、デジタル情報をほとんどまたは全く無料（媒体、手数料程度）で入手することができ、情報提供者（メーカ

または個人）が指定した条件（次項）の下でいつでも利用できる。

- (2) 情報提供者は、その情報の利用を許可する条件（通常は料金の支払い）を指定できる。
- (3) 情報提供者が指定する以外の改変は防止される。
- (4) 以上の各項のために面倒な手数を必要としない。

情報化・サービス化に適した経済学の再構築を目指す試みがいくつかなされているが、情報の価格体系については社会的に明確な回答がなされていないのが実情である。しかし、上記のようなシステムが実現すれば、

- ある情報とそれに競合するような情報との比較、
- ある情報とそれと同等の機能を実行するものとの比較、

が明確な基準によって行なわれることになり、情報の経済的価値を体系づける可能性が生まれる。このことは、情報提供者の権利と、情報利用者の便利さが保証されることによって、健全な「情報市場」が成立することを意味している。

The Inevitable Way Toward Superdistribution

Ryoichi MORI† and Masaji KAWAHARA†

† Tsukuba College of Technology

E-mail: {rmori, masaji}@k.tsukuba-tech.ac.jp

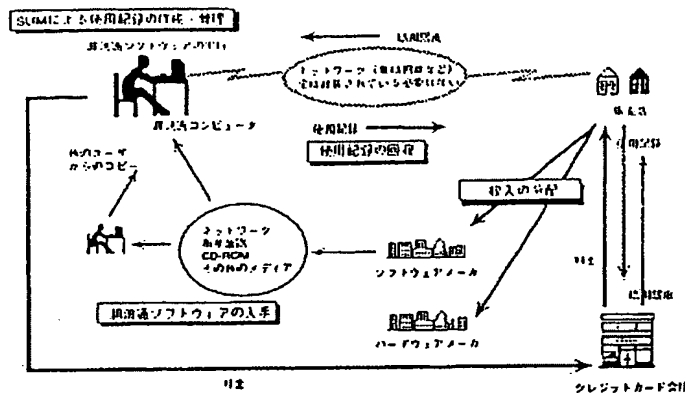


図 1: 超流通システムの一例

コンピュータプログラムの不正コピーやデジタル・オーディオの流通の混乱は、ソフトウェア産業の発展のみでなく、ハードウェア産業の可能な発展をも阻害していることに留意すべきである。これまで、情報提供者の権利の保護と情報利用者の便利さとを両立させるような基本メカニズムが欠如していたので、複製技術の発達のために法の不備が顕在化し、法の精神が有効かつ統一的にあらわれていない状況が長く続いてきた。

超流通システムとデジタル情報時代に対応した法律とを整備することによって、健全な情報化社会が保証される。マルチメディアを含むデジタル情報に対応した法律のあり方については、様々な議論が繰り返されているが、このことについては別の論文で述べる。

超流通システムは次の機能を提供する。

- (1) 情報提供者は、任意の経路（放送、CD-ROM、ネットワークなど）を用いて、大量かつ安全に情報を提供することができる。
- (2) 情報利用者は、任意の経路（放送、CD-ROM、ネットワーク、友人からのコピーなど）を用いて、大量かつ安全に情報を入手し利用することができる。
- (3) 情報提供者の権利と利用者の便利さとは保証される。

このようなシステムでは、価値のある情報を生産・加工したものは誰でも、安価な流通コス

トにより、大量に情報を提供することができる。また、このようにして提供される世界中のあらゆる情報を迅速かつ容易に利用できるようになる。このように、従来の有体物では不可能であった自由な流通形態、すなわち「超供給」と「超需要」とを併せた「超市場」の実現の可能性が生まれる。

超流通システムの一例を図 1 に示す。ユーザは、計算機に SUM (Software Usage Monitor) を取り付け、プログラムが実行されると、SUM はプログラムに電子的に添付された使用条件のもとに使用記録を作成・管理する。ユーザはあらかじめ設定された条件にしたがって使用記録を転送することによって課金される。コンピュータプログラムだけではなく、すべてのデジタル情報について同様のメカニズムで課金することが可能である。

「超流通アーキテクチャ (Superdistribution Architecture)」とは、超流通を実装するための論理的な仕様を規定したものである。したがって、「電子的な使用条件」をどのように実装するか、また、「電子的な使用記録」の回収・処理を行なうためのメカニズムをどうするか、超流通アーキテクチャの主要な部分である。

1.2 超流通と「超流通ラベル」

超流通を機能でなく構成要素の面からみると、次のような定義も可能である。

超流通とは、

- (1) 電子的に添付された使用条件
- (2) 電子的に回収・処理される使用記録

を概念的な構成要素とするものである。

この電子的に添付された使用条件を「超流通ラベル」と呼ぶことにする。超流通ラベルが持つべき性質のいくつかを以下に列挙する。

- ・ 変造、除去、および不当追加などに強いこと。
- ・ 超流通以前のデータ構造、例えば CD-ROM, MD, DCC 等と互換性を持つ超流通ラベルの導入方法があること。
- ・ 超流通ラベルを持つデータブロックが他の超流通ラベルを持つデータブロックを参照する時には、参照された超流通ラベルが自動的にリストされていくこと。
- ・ 部品としてのオブジェクトを組み合わせた時の基本事項として超流通ラベルが考慮されること。

貨幣経済の進歩した社会においては、商品としての有体物は、この超流通ラベルに相当するものを最初から持っていると考えることができる。この性質は、我々が生存するためのもっとも根本的な前提のひとつとなっている。自動車のように見えるものが実は何か別のものであったり、清涼飲料水であるはずのものが実は劇薬であるなどという社会を想像して見れば、超流通ラベルの重要性と必要な諸性質は明らかである。デジタル情報が重要なこれからの社会では、このような超流通ラベルがデジタル情報そのものに添付されていることが不可欠なのである。

ここで、この性質が統計的なものであることを指摘することが公平であろう。例えば、ある種のガスは空気と区別が付きにくい。この場合はガスに匂いをつける。また、偽物の宝石は確かに存在するが、偽って販売することは法律で厳しく罰せられる。「超流通ラベル」の正しさを保証するための様々な制度が我々の社会の基盤なのである。

2 超流通の歴史

2.1 超流通の誕生

1983年に当時筑波大学の森亮一が「超流通」を発明した。当時の名前はソフトウェア・サービス・システム (SSS) であった。これは、Water Service System (水道) にならったものである。その当時は、デジタル情報流通のための電子技術の必要性についての一般の認識は低かったので、革命的な印象を与える「超流通」という名称より穏当であったかも知れない。

ソフトウェア・サービス・システムの満たすべき条件として、以下の9項目があげられた^[A-18]。

- (1) 十分な保護性能を持つ。
- (2) 利用者が、ソフトウェアを複製することは自由であり、それをファイルシステム、ネットワークノードなどの任意の場所に格納し、自由に呼びだして実行できる。
- (3) ハードウェアを付属させる、あるいは媒体に特定のものを使用するといった必要がなく、純粋な情報としてソフトウェアを流通させることができ、電気的伝送手段による低コストの流通が可能である。
- (4) 利用者は、ソフトウェアを無料または極めて低価格で試用することができ、不満足であればそれを返却することができる。
- (5) 利用者を特定せず、不特定の利用者が計算機IDなどを報告する必要がなく匿名で、料金を支払うのみで自由にソフトウェアを利用できる方式から、権利者の指定した利用者だけに限って利用を許す方式まで、幅広い許諾供与方式を提供できる。
- (6) 利用者に対して計算機アーキテクチャを秘密にする必要がない。
- (7) その保護流通システムへの新しい権利者の参入が自由で容易である。
- (8) ソフトウェア部品の市場が成立する。
- (9) システムを社会へ最初に導入する際の抵抗が少ない。

1987年に(社)日本電子工業振興協会の中に「マイクロコンピュータソフトウェア基盤技術専門委員会(通称、SSS委員会、後に超流通委員会)」が設けられた。1988年2月には、この委員

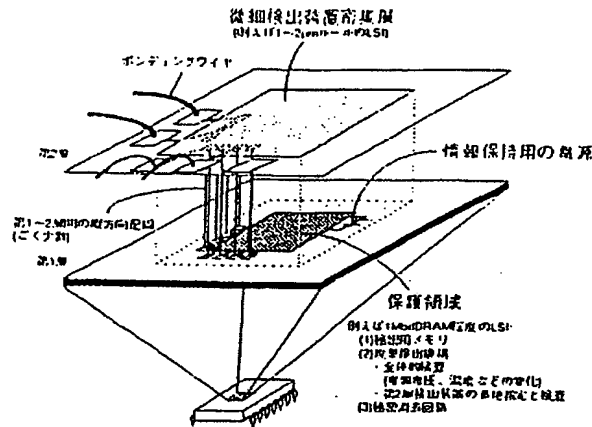


図 2: 3次元 IC による保護容器の一例

会からの報告として「ソフトウェア流通の問題点と対策」が刊行された[A-26]。その序文の中で、「超流通」という呼び名が用いられ、それが「超市場」と「超信頼性」をもたらすことが述べられた。これ以前にも、委員会の議論等では「超流通」が用いられたが、印刷されて残っているものとしては、この報告書が最初と思われる。

2.2 超流通の進化

1986年には、超流通のためのプロトタイプ I が開発された[A-19]。これは、PC-9801のシリアルポートに、SUM(Software Usage Monitor)を接続し使用記録を管理するものである。シリアル接続のためオーバーヘッドは大きい、超流通の実現可能性を明確にした。

1988年には、3次元 IC による保護容器(図 2)が発明された[A-7]。これは、強い物理的防御、すなわち、暗号の鍵を知らない限りその保護容器を設計・製作した人々自身が攻撃しても破れない程度の防御を持つ情報の容器である。保護容器は超流通において重要な役割を果たすが、それ以外の広い分野に応用される可能性を持つ。

1989年には、超流通実用化を意識したプロトタイプ II が開発された[A-40]。これは、SUMを実装するために既存計算機のコプロセッサ・インタフェースを利用したものであり、以下のような特長を持つ。

- オーバヘッドが小さくマルチタスクの計算機にも対応できる
- 1チップ化により低価格化がすすむ

同時に、システム全体の見直しも行なわれ、既存のクレジットカード決済システムを応用することによって、導入時のコストを最小限におさえるシステムが提案された。それまでは、超流通の実現のためには、決済センタと呼ばれる、おおよそ銀行システムと同程度のものが必要であると考えられていたが、この提案によって、様々な規模と種類の超流通システムが大きな初期投資を必要とせず実現される可能性があることが示された。

また、超流通においては、従来の有体物の取引では不可能であったような、様々な課金の形態が可能であり、その可能性が検討された[A-34]。超流通で提供される主な課金システムとしては、試用課金、従量課金、自動買い取り、買い取り後の返金、特別許諾、無料だが使用状況の報告を義務づけるものなどが考えられている。

1990年になって、コンピュータプログラム以外のデジタル情報の取り扱いについても具体的な検討が始まり、デジタル・オーディオのための超流通システムが設計された[A-46]。ここで、デジタル情報を二つの種類に分けて考えるべきであることが指摘されるようになった。すなわち、コンピュータプログラムを主とする実

行型デジタル情報と音楽・映像およびデータベースを主とする参照型のデジタル情報である。この分類を明確にした超流通システムの一例を図3に示す。

3 超流通はなぜこれまで実現されなかったのか

3.1 マイコン革命とソフトウェア

1970年代初めにマイクロプロセッサが生まれ、その性能および売り上げは20年以上にわたって急激に成長し続けており、今後もさらに続くことは疑う余地がない。マイクロプロセッサの出現によって、コンピュータハードウェアの量産が初めて可能になった。マイクロプロセッサは情報一般を扱う工業製品であるが、それ自体は有体物である。このことによって、マイクロプロセッサ産業の劇的な成長には、特別な流通システムと法的基盤を必要としなかったのである。

ところがソフトウェアでは事情が違っていた。オリジナルと同品質のコピーが高速かつ安価に作られるという、本来素晴らしいはずの性質が、混乱を引き起こす結果となった。このために、マイクロプロセッサでは一種類当たり100万個単位の量産はごく当たり前だが、ソフトウェアでは極めて例外的である。また、ハードウェアでは当然かつ不可欠の基盤である部品産業は、ソフトウェアの領域では実用規模にはなっていない。

3.2 ソフトウェア研究技術者が目指すもの

それではなぜソフトウェアの量産は阻害されたのだろうか。この問題は根が深くかつ多様なので、ここでは、以下に述べるいくつかの切り口を提示することによって、今後のソフトウェア産業の健全かつ活発な発展に資したいと思う。

工業製品であるソフトウェアにとって、汎用性の向上と単価の低減が社会への貢献として最も重要であることを、ソフトウェア研究技術者のかなりの人々が見逃しているように見える。たとえば、スプレッドシートプログラムの重要性は、実物がその実力で市場の評価を勝ち取るまで、そしてその後これほど経ってからも、学会等ではあまり研究討論されていない。それよりはるかに難解で価値の少ない論文は山ほどあるように見える。

マイクロプロセッサでは、量産による価格低減が重要な目標であることがよく知られている。しかし、量産の効果がはるかに直接的なソフトウェア（2倍の個数が売れば、単価は半分になる）において、需要個数を大きくするための研究は充分に行なわれたとは言えない。Object Oriented Programmingはまさにこの方向であるという意見がありうる。これについてBrad Coxの見解を紹介しよう。彼は、ACMのOOPSLA'86のプログラム委員でもあり、Object Oriented Programmingの仕掛け人の一人でもあるが、次のように述べている。「Object Oriented Programmingはオブジェクトの再利用を可能にする。ただし料金の精算はできない。超流通はオブジェクト流通の料金精算を可能にする。」[C-12]

一言で言えば、根本的で影響の大きい、汎用性や価格に劇的な変化をもたらす可能性のある

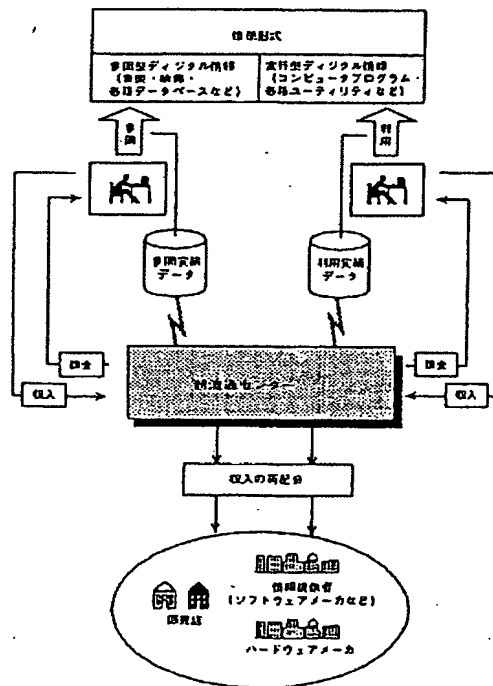


図3: 超流通システムの一例
(吉岡の図[C-17]をもとに書き直したもの)

ような独創的な着想よりも、精緻で難解だがあまり大きな改善をもたらさないような研究の方が高く評価されやすい傾向が見られる。

人々は、ソフトウェアの生産に関して、飛躍的な発展や改善など身近にあるはずがないという精神的バリアを持っているようである。これを突破するために、超流通がなぜ必然であるかを次節で述べよう。

4 歴史的必然としての超流通

4.1 物々交換から貨幣経済へ

有効に働く流通システムは社会にとって不可欠のものであるが、初めから自然に存在しているわけではない。隣人の財産が欲しいときそれを入手する方法は、流通システムがなければ、盗むか殺すかしかない。何万年か前、有体物についてそのような状況が存在した。そして、1930年頃から、デジタル情報の多くのものについてそのような状況が存在している。これを解決するためには超流通が必要である。

「有体物のための超流通」は物々交換経済によって始まった。そして、有体物の場合には、それを抽象化することができないので、物々交換経済は引き続いて貨幣経済へ移行する必要があった。これらが有体物における超流通である。この超流通は社会の不可欠な基盤として普及し、洗練され、極めて多様な、しかし有効かつ必要な複雑性を産み出していった。我々の社会では、このことはあまりに当然なので、ほとんどだれも気がつかないが、社会の多くの部分が、有体物の流通のための特有な構造を持っている。

4.2 産業製品としてのデジタル情報

ここでデジタル情報が重要な産業製品として登場してきた。それは有体物とは大きく異なる性質を持つ。誰でも知っている目立った違いは、例えば

- (1) 有体物 (例えば土地、自動車、農作物、海産物、マイクロプロセッサ) では、それを相手に渡せば自分の持っているものはなくなる。これに対して、デジタル情報ではコピーを相手に渡すための経費はほとんど

ゼロであり、自己の所有する財物はなくなるらない。

- (2) 有体物を地球規模で量産すれば、資源の枯渇および廃棄物処理の問題が生じる。デジタル情報ではこの問題は無視できるほど小さい。問題となる部分は、デジタル情報そのものでなく、媒体である有体物に依存する部分である。

- (3) 有体物の大部分はエネルギー資源を必要とし、例えば自動車の例で見られるように、その効率の改善には原理的で厳しい限界が存在する。これに対してデジタル情報では、ブランクの定数に関わってゆくところまで、原理的な改善の余地が大きく残されている。

などである。

有体物であるための特有の事象は、社会の隅々まで広く我々の生活のあらゆる部分に及んでいる。いくつかの例をあげよう。

- (1) 缶入りのコーヒーと紅茶を買うとき、それぞれが別々の缶に入っているのは当然に見える。デジタル情報においてはそのような必要はなく、ひとつの「缶」に多数のものをを入れるべきである。
- (2) 電気、ガス、水道などは有体物なので、別々な経路によって家庭に届く。デジタル情報ではこのような必要はないにも拘わらず、異なる有料放送が異なる経路で届くことを当然であると感じる人の方が、それが無駄であって我慢できないと感じる人より多い。デジタル情報では、「缶」はひとつにするのが自然である。そしてこの「缶」のどこに何がどのように入っているかの情報、すなわちインデックスが規格化された形で存在することが不可欠である。有体物ではインデックスのない個別の容器が便利であり、デジタル情報ではインデックスのある少数の容器が便利なのである。
- (3) 我々は自分のもつ貨幣を財布によって物理的に防御する。この防御は極く弱い。しかし、我々が銀行の金庫を携帯しないで快適に暮らせる理由は、目には見えないが財布の外側を法律が包んでいるからである。と

はなくな

資源の枯
る。ディ
きるほど
デジタル情
体物に依

を必要と
るように、
い限界が
フル情報で
くところ
く残され

社会の隅々
及んでい

とき、それ
は当然に見
はそのよう
多数のもの

ので、別々
デジタル情
拘わらず、
富くことを
それが無駄
より多い。
とつにする
匠」のどこ
の情報、す
れた形で存
体物ではイ
便利であり、
久のある少

よって物理
弱い。しか
ないで快適
ないが財布
である。と

同時に財布もその中身も有体物であり、盗まれば盗まれたことが目に見えるということが根本にあるからである。デジタル情報には、盗まれたことが目に見えるというこの性質も、そして充分かつ適切という意味では法律による保護もまだない。

- (4) 産業規模での売買では、例えば部品・材料が到着すれば受入れ試験をする。時には改まった試験の必要もないこともある。みかんを購入したときそれが腐でないことは別に測定装置を使わなくともわかるからである。デジタル情報ではこのような都合のよい性質は全くないにも拘わらず、デジタル情報の売買に当たって、適切な受入れ試験が行なわれないことはごく普通に見られる。

このような事象を考えれば、1.2で述べた「超流通ラベル」がどれほど重要な役割を持っているかがわかる。

4.3 ソフトウェア量産革命に向かって

これまで述べたことから見ると、デジタル情報は全く不便で困った性質を多く持っているように見える。ソフトウェア危機が指摘されたりすることの遠因のひとつはここにある。

しかし、そのような考えは最も深い根のところで間違っている。そもそも、我々の社会は有体財物の流通に適するように何千年にも渡って社会システム全体の改善を行ない、それによって社会の機構は精緻で複雑なものとなり、有体物の流通については高い性能を持つ構造へと変わってきたのである。

したがって、それをデジタル情報へそのまま適用すれば無数の不都合が生じるのは当然のことである。にも拘わらず、人々は、既に存在するシステムを流用して何とか苦境から逃れようとする。狸の泥船でも小川なら渡れるかもしれない。しかし、人類の歴史の上で恐らく最初で最後の、有体物からデジタル情報へという最大の変化を泥舟で乗り切ろうとすれば沈没するのは当たり前である。

デジタル情報は、産業製品として、また財物として、有体物の持っていない素晴らしい性

質を限りなく持っている。そして現在の我々の社会はその利点のほとんど全てを無視し捨ててしまっている。その理由は簡単で、これまでの有体物にはそのような利点はなく、人々はそれを楽しんだことがないから、それがどれほど素晴らしいか知らないのである。

デジタル情報の諸性質の中でも飛び離れて重要な性質は、自己の持つ財産を他人に分け与えても財産が減らないという性質である。人類の歴史の上で、ほとんどの流血の原因は財産の取り合いであった。その意味で、デジタル情報が産業製品の主要なものとなることは、人類の社会が長い貧困と流血の時代から離脱するための最も強力な推進力を得ることになる。

しかし同時にデジタル情報に特有の他の性質、すなわち、一度相手に渡した時、特段の措置がなければ、単にそれを渡された人々が、それを創り今後もその品質を維持することに努力する人々と全く同じ立場になってしまうという問題を同時に持っている。この問題を解決することが、これからの社会にとって最も重要な課題なのである。

この課題は、電子技術によって解決可能であり、それによって人類の歴史の上での2回目の超流通、すなわち、「デジタル情報のための超流通」が実現されるのである。

1993年12月31日現在での超流通に関する出版物・特許・関連資料をまとめた。出版物は森のグループによるもので、49件、特許は森が出願した日本特許4件、外国特許3件、関連資料は森のグループ以外の筆者によって超流通に言及されたもの17件である。

出版物一覧

- [A-1] 森亮一: "ソフトウェア・サービスについて", JECC ジャーナル, No.3, pp.16-26 (1983)
[A-2] 森亮一: "ソフトウェア・サービス・システム", 事務と経営, Vol.35, No.440, p1, (1983)
[A-3] 森亮一: "ソフトウェア・サービス・システムの提案", コンピュータワールド, Vol.3,

- No.1, pp.11-15 (1984)
- [A-4] 森亮一：“ソフトウェアサービスシステム”，電子通信学会誌，Vol.67, No.4, pp.431-436 (1984)
 - [A-5] 森亮一：“ソフトウェア・サービス・システムについて”，情報通信学会誌，Vol.2, No.1, pp.49-56 (1984)
 - [A-6] 森亮一：“未知の，次の革命”，日立評論，Vol.66, No.7, 巻頭言 (1984)
 - [A-7] 森亮一，田代秀一，籠田行一：“ソフトウェアサービスシステムにおけるソフトウェアの構造及び流通方式”，情報処理学会第29回全国大会講演論文集，pp.407-408 (1984)
 - [A-8] 森亮一，田代秀一，籠田行一：“ソフトウェアサービスシステムのための計算機付加機構”，情報処理学会第29回全国大会講演論文集，pp.223-224 (1984)
 - [A-9] 森亮一，田代秀一，籠田行一：“ソフトウェアサービスシステムにおけるソフトウェア使用料金の精算方式”，情報処理学会第29回全国大会講演論文集，pp.409-410 (1984)
 - [A-10] 森亮一：“ソフトウェア保護革命”，シャープ技報，31号，pp.5-6 (1985)
 - [A-11] 森亮一：“ソフトウェアのためのモラルと利益”，岩波講座マイクロエレクトロニクス1月報10，pp.1-4 (1985)
 - [A-12] 森亮一：“ソフトウェアの保護と流通 要旨”，通産大臣賞受賞，読売新聞全国版，1985年7月31日付け11面
 - [A-13] 森亮一，田代秀一，籠田行一：“ソフトウェア・サービス・システム (SSS) のための暗号方式”，情報処理学会第31回全国大会講演論文集，pp.183-184 (1985)
 - [A-14] 森亮一，田代秀一：“ソフトウェア・サービス・システム (SSS) のための計算機アーキテクチャ”，情報処理学会第31回全国大会講演論文集，pp.185-186 (1985)
 - [A-15] 田代秀一，小野雅見，森亮一：“データ通信に於ける盗聴および不法侵入防止方式に関する一提案”，情報処理学会第31回全国大会講演論文集，pp.187-188 (1985)
 - [A-16] 森亮一：“トップエンジニア・研究者が予測 15年後のデバイス技術”，電波新聞社 Computer Design 11月号，p64 (1985)
 - [A-17] 小尾信彌，太田次郎，森亮一：“21世紀の科学”，超の世界 No.150，日経サイエンス，Vol.16, No.3, pp.6-7 (1986)
 - [A-18] 森亮一，田代秀一：“ソフトウェア・サービス・システム (SSS) の提案”，電子情報通信学会論文誌，Vol. J70D, No.1, pp.70-81 (1987)
 - [A-19] 田代秀一，森亮一：“ソフトウェア・サービス・システム (SSS) の小規模な試作”，電子情報通信学会論文誌，Vol. J70D, No.2, pp.335-345 (1987)
 - [A-20] 森亮一：“ソフト・サービス・システム”，日経産業新聞，1987年5月4日付け15面
 - [A-21] 森亮一“ソフトウェア・サービス・システム (SSS) の構想”，情報処理振興事業協会技術センター，ワークショップ「パソコンソフト市場と技術の動向」，セキュリティ関連技術の開発に関する報告書 (62技-082)，pp.132-133 (1987)
 - [A-22] 田代秀一，森亮一：“ソフトウェア・サービス・システム (SSS) の技術”，情報処理振興事業協会技術センター，ワークショップ「パソコンソフト市場と技術の動向」，セキュリティ関連技術の開発に関する報告書 (62技-082)，pp.134-138，(1987)
 - [A-23] 森亮一：“技術革新の過去・現在・未来”，情報処理学会，「マイクロコンピュータの現状と将来」シンポジウム，p1 (1987)
 - [A-24] 森亮一：“ソフトウェアサービスシステムの提案”，電子工業月報第29巻第11号，pp.30-33 (1987)
 - [A-25] 森亮一：“ソフトウェア流通を促進する新電子流通システム”，(社)日本パーソナルコンピュータソフトウェア協会主催講演会，pp.1-14 (1987)
 - [A-26] (社)日本電子工業振興協会：“ソフトウェア流通の問題点と対策”，pp.1-70 (1988)
 - [A-27] (社)日本電子工業振興協会：“ソフトウェア流通に関する提案と概要”，マイクロコンピュータに関する調査報告書 (II)，

・研究者が予
 計、電波新聞
 : p64 (1985)
 : "21世紀
]経サイエン
 36)
 エア・サービ
 : 電子情報通
 .1, pp.70-81

 ソウェア・サー
 規模な試作",
 .J70D, No.2,

 ・システム",
 日付け15面
 -ビス・シス
 型振興事業協
 ップ「パソコ
 」, セキュリ
 る報告書 (62

 ウェア・サー
 所", 情報処理
 ワークショッ
 技術の動向",
 発に関する報
 8, (1987)
 現在・未来",
 ンピュータの
 , p1 (1987)
 -ビスシステ
 29 巻第 11 号,

 通を促進する
 日本パーソナ
 ア協会主催講

 : "ソフトウェア
 .1-70 (1988)
 含: "ソフト
 概要", マイク
 査報告書 [II],

- pp.190-218 (1988)
- [A-28] 森亮一: "超流通と保護容器 - 3 次元構造 IC の新しい応用 -", つくば研究支援センター, 最先端技術セミナー「ギガビットを目指す LSI 技術」資料, pp.1-32 (1988)
 - [A-29] 森亮一: "超流通の実現が世界を変える", 日立評論, 巻頭言, 第 70 巻, No12 (1988)
 - [A-30] Ryoichi Mori: "What Lies Ahead", BYTE, Vol.14, No.1, pp.346-348 (1989)
 - [A-31] (社) 日本電子工業振興協会: "ソフトウェア流通のための基盤技術", マイクロコンピュータに関する調査報告書 [II], pp.107-221 (1989)
 - [A-32] 森亮一: "超流通と超市場", 計測と制御, Vol.28, No.5, p68 (1989)
 - [A-33] 森亮一, 河原正治, 篠崎雅英, 大瀧保広: "「超流通」研究計画の現在", 情報処理学会「利用者指向の情報システム」シンポジウム論文集, pp.173-180 (1989)
 - [A-34] 植木伸一, 大瀧保広, 森亮一: "超流通のための権利管理機構における会計処理", 情報処理学会情報システム研究会, 27-5, pp.1-10 (1990)
 - [A-35] 河原正治, 森亮一: "超流通アーキテクチャのためのプロトタイプ II", 情報処理学会情報システム研究会, 27-6, pp.1-9 (1990)
 - [A-36] 森亮一, 河原正治, 篠崎雅英, 秋葉善之, 大瀧保広: "超流通アーキテクチャ", 電子情報通信学会, 情報セキュリティ研究専門委員会, 暗号と情報セキュリティシンポジウム講演論文集, SCIS90-6A, pp. 1-8, (1990)
 - [A-37] 森亮一: "保護容器", 電子情報通信学会, 情報セキュリティ研究専門委員会, 暗号と情報セキュリティシンポジウム講演論文集, SCIS90-6B, pp.1-10, (1990)
 - [A-38] 森亮一, 河原正治: "超流通のためのプロトタイプの実現", 電子情報通信学会情報セキュリティ研究専門委員会, 暗号と情報セキュリティシンポジウム講演論文集, SCIS90-6C, pp.1-9, (1990)
 - [A-39] Ryoichi MORI, Masaji KAWAHARA:

"Superdistribution: An Overview and the Current Status", 電子情報通信学会研究会資料, ISEC89-44, pp.25-35 (1990)

- [A-40] Ryoichi Mori, Masaji Kawahara: "Superdistribution: The Concept and the Architecture", The Trans. of IEICE, Vol E73, No.7, pp. 1133-1146 (1990)
- [A-41] Ryoichi Mori: "On Superdistribution", BYTE, Vol.15, No.9, p344 (1990)
- [A-42] 大瀧保広, 森亮一: "超流通における暗号化の手法について", 電子情報通信学会研究会資料, ISEC90-49, pp.33-42 (1991)
- [A-43] (社) 日本電子工業振興協会: "ソフトウェア超流通技術に関する調査", マイクロコンピュータに関する調査報告書 [III], pp.103-158 (1991)
- [A-44] 森亮一, 河原正治: "科学が変えた 20 年 IV コンピュータ", 別冊日経サイエンス 創刊 20 周年記念号, pp.59-63 (1991)
- [A-45] 森亮一: "マイクロコンピュータの意義", 情報処理学会マイクロコンピュータとワークステーション研究会資料, 71-3-1, pp.1-8 (1992)
- [A-46] (社) 日本電子工業振興協会: "超流通技術報告", マイクロコンピュータに関する調査報告書 [III], pp.71-110 (1992)
- [A-47] (社) 日本電子工業振興協会 ソフトウェア超流通研究会: "超流通技術調査報告", pp.1-27 (1993)
- [A-48] 森亮一: "マイコン革命と超流通", 日刊工業新聞, 1993 年 4 月 19 日, 全国版 19-20 面
- [A-49] 森亮一: "ソフトウェア流通の混乱を打開する超流通, 使用記録の回収で料金徴収", 日経エレクトロニクス, No.582, pp.98-101 (1993)

特許一覧

- [B-1] "ソフトウェア管理方式" (特願昭 58-186100 号, 出願日: 1983 年 10 月 5 日, 出願人: 森亮一)

- [B-2] "ソフトウェア利用管理方式" (特願昭 59 046923 号, 出願日: 1984 年 3 月 12 日, 出願人: 森亮一)
- [B-3] "SYSTEM FOR STORING HISTORY OF USE OF PROGRAMS INCLUDING USER CREDIT DATA AND HAVING ACCESS BY THE PROPRIETOR" (US Patent: 5,103,392, 出願日: 1984 年 10 月 4 日, 出願人: Ryoichi Mori, 交付日: 1992 年 4 月 7 日)
- [B-4] "ソフトウェア権利管理制御方式" (特願 昭 62-227106 号, 出願日: 1987 年 9 月 10 日, 出願人: 森亮一)
- [B-5] "TAMPER RESISTANT MODULE HAVING LOGICAL ELEMENTS ARRANGED IN MULTIPLE LAYERS ON THE OUTER SURFACE OF A SUBSTRATE TO PROTECT STORED INFORMATION" (US Patent: 5,185,717, 出願日: 1989 年 7 月 31 日, 出願人: Ryoichi Mori, 交付日: 1993 年 2 月 9 日)
- [B-6] "ソフトウェア管理方式" (特願平 02-211406 号, 出願日: 1990 年 8 月 9 日, 出願人: 森亮一)
- [B-7] "TAMPER RESISTANT MODULE WITH LOGICAL ELEMENTS ARRANGED ON A SUBSTRATE TO PROTECT INFORMATION STORED IN THE SAME MODULE" (米国特許出願番号: 07/972,747, 出願日: 1992 年 10 月 6 日, 出願人: Ryoichi Mori, 許可通知: 1993 年 10 月 8 日)

関連資料一覧

- [C-1] 坂村健: "Making of TRON", Bit, Vol.18, No.12, p.68 (1986)
- [C-2] 坂村健: "TRON からの発想", pp.197-200, 岩波書店, (1987)
- [C-3] 坂村健: "TRON を創る", pp.134-135, 共立出版 (1987)
- [C-4] YHP Hi-Tech Report 27, 日経サイエンス, Vol.17, No.7, pp.0-2 (1987)
- [C-5] YHP Hi-Tech Report 27, 日経ビジネス, 1987 年 7 月 6 日号, pp.86-87 (1987)
- [C-6] "新しいソフトの販売形態", プロンプト, Vol.3, No.8, p.88 (1987)
- [C-7] 原達三: "次世代パーソナル・コンピューティングの概念-次世代 PC 研究会からの提言-", Computer Today 1987/5 No.19, pp.19-27 (1987)
- [C-8] "IC カードによるソフト流通システム構想-電子協に専門委発足-", 日経パソコン, 1987 年 7 月 20 日号, p.120 (1987)
- [C-9] "Electronic Distribution Method Would Hinder Piracy, Japanese Say", BYTE, Vol.12, No.9, pp.38 (1987)
- [C-10] 中村達郎, 松本勉, 今井秀樹: "IC カードを用いたソフトウェア保護の方式", 電子情報通信学会, 1988 年暗号と情報セキュリティシンポジウム資料 F-1, pp.1-10 (1988)
- [C-11] 那野比古: "ソフト「超流通」で新たな文化基盤の創出へ" 所有する"ではなく"利用する"を徹底", OA 情報 OA ビジネスパソコン別冊, 1991 年春号, pp.18-25, 電波新聞社 (1991)
- [C-12] Brad Cox: "Superdistribution and Electronic Objects", Dr. Dobbs' Journal, No.193, pp.44-48 (1992)
- [C-13] "ソフト流通, 「物」から「使用権」へ暗号化して CD-ROM で配布", 日経エレクトロニクス, 1993.9.13(No.590), p.146 (1993)
- [C-14] "Interview of Ryoichi Mori", Intertek, Vol.3.4, pp.22-23 (1993)
- [C-15] 伊勢呂裕史: "著作権をめぐる諸問題, コピーライト", Vol.33, No.389, p.5, (社) 著作権情報センター (1993)
- [C-16] 飯田恭子: "「超流通」が変えるソフト社会", パソコン倶楽部, No.22, pp.110-112 (1993)
- [C-17] 吉岡誠: "SGML のススメ", pp.154-155, オーム社 (1993)

複写をされる方に

本誌(冊)に掲載された著作物は、政令が指定した図書館で行うコピーサービスや、教育機関で教授者が講義に利用する複写をする場合等、著作権法で認められた例外を除き、著作権者に無断で複写すると違法になります。そこで、本著作物を合法的に複写するには、著作権者から複写に関する権利の委託を受けている次の団体と、複写をする人またはその人が所属する企業・団体等との間で、包括的な許諾契約を結ぶようにしてください。

学協会著作権協議会内日本複写権センター支部
〒107 東京都港区赤坂 9-6-42-704
Phone 03 (3475) 4621-5618
Fax 03 (3403) 1738

Notice about photocopying

In the U.S.A., authorization to photocopy the copyrighted publication or parts thereof for internal or personal use, or the internal or personal use of specific clients, is granted by Information Processing Society of Japan, provided that designated fees are paid directly to Copyright Clearance Center. For those organizations that have been granted a photocopy license by CCC, a separate system of payment has been arranged.

Copyright Clearance Center, Inc.
27 Congress St.
Salem, MA 01970
Phone (508) 744-3350
Fax (508) 741-2318



超編集・超流通・超管理の
アーキテクチャ シンポジウム

©情報処理学会 1994

発行所 〒160 東京都新宿区西新宿1-24-1 エステック情報ビル27F

社団法人 情報処理学会

情処シンポジウム論文集 Vol.94, No.1
1994年2月17日 発行

TEL 東京 (03)5322-3535(代表)
郵便振替口座 (東京 5-83484)

発行人 社団法人 情報処理学会

飯塚 浩司

Information Processing Society of Japan

**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.

Z16-940

信学技報 Vol. 94 No. 240

電子情報通信学会技術研究報告

ISEC 94-13~22

〔情報セキュリティ〕

1994年9月21日



BEST AVAILABLE COPY

EIC 法人 電子情報通信学会

超流通における電子オブジェクト課金方式の検討

河原 正治

筑波技術短期大学 教育方法開発センター

茨城県つくば市春日 4-12

あ ら ま し 本論文では、ソフトウェア部品やマルチメディア素材といった電子オブジェクトを円滑に流通させるための超流通システムについて述べる。アプリケーションソフトウェアのための超流通は、ソフトウェアの潜在的需要を100%に近いレベルまで引き出す技術である。これに対して、電子オブジェクトのための超流通は、ソフトウェア部品化の潜在的可能性を引き出し、ソフトウェアメーカーやユーザが標準的なソフトウェア部品を用いることによってソフトウェア開発費を劇的に引き下げる技術となる。現在、UNIX ワークステーション上でネットワークを利用したプロトタイプを開発中である。

和文キーワード 超流通, ソフトウェア流通, ソフトウェア再利用, 電子オブジェクト, マルチメディア, 著作権

Superdistribution for Electronic Objects

Masaji KAWAHARA

Research Center on Educational Media,
Tsukuba College of Technology

4-12 Kasuga, Tsukuba-shi, Ibaraki 305, Japan
Email : masaji@k.tsukuba-tech.ac.jp

Abstract This paper describes Superdistribution system that can freely distribute payable electronic objects such as "software parts" and multimedia components. Superdistribution for application software could draw out potential use of software. Superdistribution for electronic objects could enable both software vendors and end-users utilize standard software parts, consequently drop significantly in software development cost. We are developing prototypes on UNIX workstations using UNIX interprocess communication facilities.

英文 key words Superdistribution, software distribution, software reusability, electronic object, multimedia, copyright

1. はじめに

マイクロプロセッサは、コンピュータハードウェアの量産を初めて可能にした。その性能および売り上げは20年以上にわたって急激な成長を続けており、マイクロプロセッサを汎用部品として利用することにより、情報機器の急激な価格低下がもたらされた。

このような急激な価格低下はソフトウェア開発には起こっていない。オリジナルと同品質のコピーが高速かつ安価に作られるという、本来ならば優れた性質が、混乱を引き起こす原因となった。このために、マイクロプロセッサでは1種類当たり100万個単位の量産は普通に行われるが、ソフトウェアでは例外的である。また、ハードウェアでは当然かつ不可欠の基盤である部品産業は、ソフトウェアの領域では実用規模にはなっていない。

また、マルチメディア分野においては、上のような問題と同時に、マルチメディアデータ(素材)に関する著作権の問題点として、マルチメディアソフトウェア制作者が、すべての権利者との合意を得ることの困難さが指摘されている。

ソフトウェアの部品産業を確立し、マルチメディアの普及を促進するためには、ソフトウェア部品やマルチメディア素材を自由に供給し、その対価が得られるような技術的・法的基盤が不可欠である。

超流通は、デジタル情報の自由な流通と利用とを実現するための基盤技術である。超流通は、これまで不可能と考えられていた「ソフトウェアの量産」を可能にする。デジタル情報を「所有する」のではなく、利用するたびに使用記録が管理され、それを回収することによって料金を徴収し収入を再分配するシステムである。これまで、パーソナルコンピュータのアプリケーションソフトウェアのための超流通についてプロトタイプの開発などを行ってきた[1, 2, 3]。

本論文では、ソフトウェアの部品化を進め、また、マルチメディアにおける権利処理を円滑に行うことを目的とした、「ソフトウェア部品」の課金方式について述べる。アプリケーションソフトウェアのための超流通は、ソフトウェア量産による収益の増大をもたらす。これに対して、ソフトウェア部品のための超流通は、ソフトウェアの開発費の劇的な低下をもたらすだろう。また、メーカおよび個人が様々な

ソフトウェア部品やアプリケーションソフトウェア製品を自由に提供し、収益を得られる「超流通ソフトウェア部品コミュニティ(図1)」の可能性について論じる。

2. 超流通

2.1 超流通と「超流通ラベル」

超流通を実現するためには、流通させるデジタル情報に対して、使用条件を電子的に添付する必要がある。この使用条件を「超流通ラベル」と呼ぶ。超流通ラベルが持つべき性質として以下のものがある[3]。

- (1) 超流通ラベルを持つデジタル情報が他の超流通ラベルを持つデジタル情報を参照する時には、参照された超流通ラベルが自動的にリストされていくこと。
- (2) 変造、除去、および不当追加などに強いこと。
- (3) 超流通以前のデータ構造、例えば CD-ROM, MD, DCC 等と互換性を持つ超流通ラベルの導入方法があること。

ソフトウェア部品やマルチメディア素材としての電子オブジェクトを組み合わせて流通させ、利用料金を課金するための基本事項として超流通ラベルが考慮されていることを(1)の性質は示している。

流通するデジタル情報を処理する機器には、「超流通ラベルリーダ」が付加される。これは、超流通ラベルの記述にしたがって、デジタル情報の使用記録を管理するものである。

2.2 アプリケーションソフトウェア課金のための超流通

超流通においては、従来の有体物の取引では不可能であったような、様々な課金の形態が可能である。試用課金、従量課金、自動買い取り、買い取り後の返金、特別許諾、無料だが使用状況の報告を義務づけるものなどが検討された。ソフトウェアの特色に応じて、単純な課金から複雑なものまで自由に選択することができる。

また、使用料金の徴収・分配には既存のクレジットカード決済システムを応用することによって、導入時のコストを最小限におさえるシステムも提案されている。

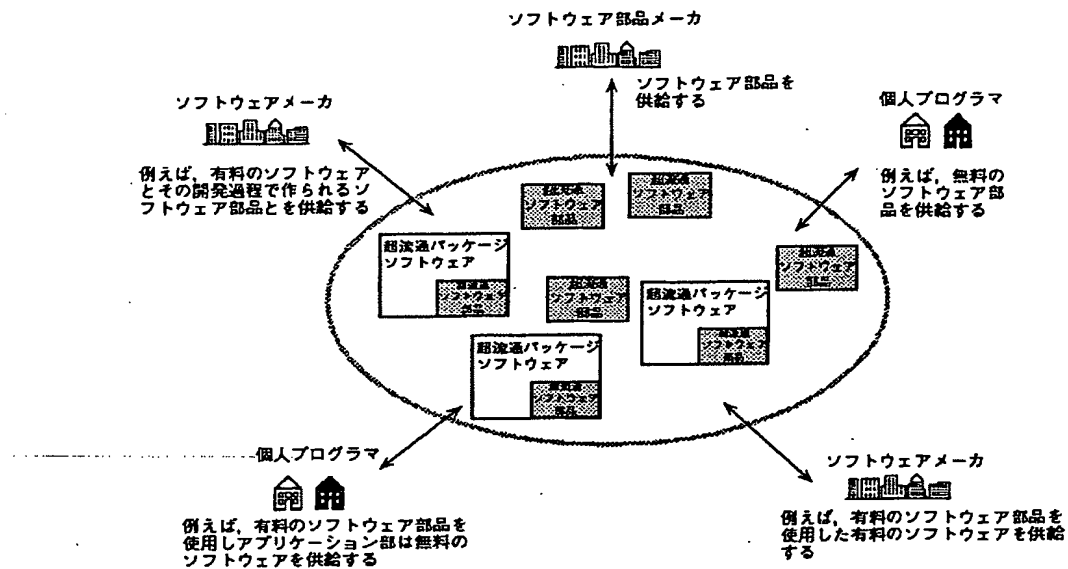


図 1: 超流通ソフトウェア部品コミュニティ

これまでに、パーソナルコンピュータのアプリケーションソフトウェアの使用記録を管理するための二つのプロトタイプが開発されている。超流通ラベルリーダを実装するために既存計算機のシリアルインタフェースを利用するものと、コプロセッサインタフェースを利用するものである。後者は以下のような特長を持つ。

- オーバヘッドが小さくマルチタスクおよびマルチプロセッサの計算機にも対応できる。
- 1チップ化により低価格化がすすむ。

以下では、開発した超流通計算機のプロトタイプ

の動作を例に、アプリケーションソフトウェアの課金がどのように行われるかについて説明する^[1]。

図 2, 3 にプロトタイプの画面例を示す。利用者は、まず、計算機に超流通ラベルリーダを取り付ける。超流通計算機上でプログラムが実行されると、ラベルリーダはプログラムに添付された超流通ラベルをもとに使用記録を作成・管理する。利用者はあらかじめ設定された条件にしたがって使用記録を転送することによって使用料金が徴収される。

図 2 は、プロトタイプ計算機で動作している超流通ソフトウェアの超流通ラベルを表示した例である。

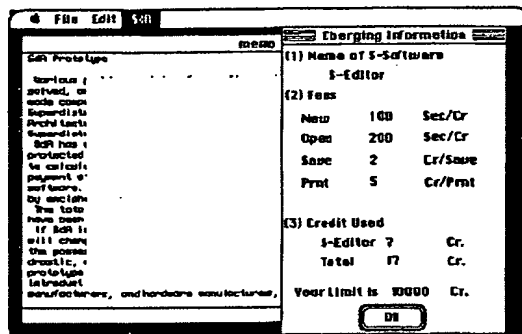


図 2: 超流通ラベルの表示

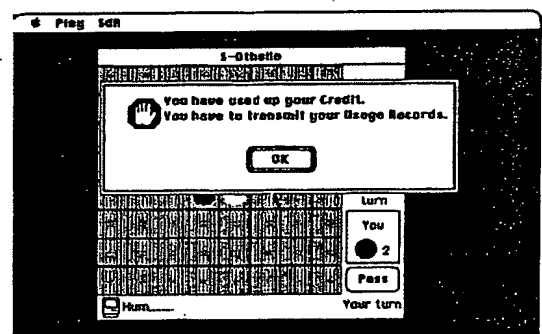


図 3: プログラム停止画面

これは、超流通ラベルリーダとともに提供される超流通ユーザユーティリティによって行われる。エディタの料金設定として、時間制料金と機能別の料金を一つの例として実現している。

使用記録の回収についても様々な方式が実現可能である。現在稼働しているプロトタイプ計算機では、利用者の超流通ラベルリーダにあらかじめ使用額の上限を設定する方法を用いている。使用額の累計がその上限に達すると、超流通ラベルリーダは超流通プログラムの実行を停止し、使用記録の転送を促す。利用者は、使用記録を転送することによってプログラムの使用を再開することができる。図3にプログラムの停止画面の例を示した。

上述の例は、コンピュータプログラムの課金を示しているが、同様のメカニズムを用いることによって、すべてのデジタル情報に対して課金することが可能である。

3. 電子オブジェクトのための超流通の目標

本節では、電子オブジェクトの超流通がもたらす効果について、マルチメディアの権利処理およびソフトウェア開発コストの削減という二つの側面から考察する。

3.1 マルチメディアにおける権利処理

マルチメディアは、文字、音声、画像などをデジタル処理し、利用者との間の双方向性を実現するソフトウェア技術として注目されている。マルチメディアを普及させるために、マルチメディア素材を、利用者が自由に加工・合成し、オリジナリティの高いマルチメディアソフトウェアを提供できるようにすることが望まれている。

したがって、多様で大量のデジタル情報を自由に利用できることが求められる。しかしながら、現状の法的・技術的基盤の下では、コンピュータソフトウェアの違法コピーの問題、デジタルオーディオの混乱などがこの分野においても繰り返されることは間違いない。

また、もう一つの問題点は、マルチメディアソフトウェア制作者は、すべての素材の著作権者との間で個別に合意を得なければならないことである。多数の権利者から許可を得るための時間的・金銭的コストが膨大なものとなる。

以下では、マルチメディア素材やソフトウェア部品に対して、上で述べたような問題を電子的手段を用いて自動的に解決する方法について検討し、超流通がそのための技術的基盤であることを述べる。

3.2 ソフトウェア開発コストの削減

マイクロプロセッサでは、量産による価格低減が重要な目標の一つであることが広く認識されている。しかし、量産の効果がはるかに直接的なソフトウェア(n 倍の個数が売れば、単価は $1/n$ にできる)において、需要個数を大きくするための研究は充分に行なわれていない。2.2で述べた、アプリケーションのための超流通技術は、このソフトウェアの需要を最大限に引き出すことを可能にする。

これまで、高性能なハードウェア資源を投入することを含めて、ソフトウェア開発の効率化には様々な試みがあった。しかし、ソフトウェア開発の過程においてより優れたツールを使ったり、開発プロセスの改善に努めたとしても、コストを劇的に下げることができないことは明らかである。汎用性のある再利用可能な部品の利用を促進することによってしか、ソフトウェア開発のコストを劇的に下げることができない。

ソフトウェアの部品化・再利用技術の研究は、ソフトウェア開発コストの低減を目的として活発である。しかしながら、そのような部品を電子オブジェクトとして自由に市場に供給し、対価を得ることができるような技術的基盤については、これまでのところ全く議論されていない。ソースコードレベルでの部品化・再利用が行われたとしても、それが組織内にとどまっている限り大きな経済効果はなく、下で述べるような再利用に対する障壁を乗り越える力とはならない。ソフトウェア開発過程において生成されるソフトウェア部品を自由に供給しその対価が得られるような基盤が整備されて、始めてソフトウェア部品化・再利用の効果が得られるのである。

ソフトウェア再利用を進めるための大きな障壁として、つぎのことが指摘されている^[4]。

- (1) 再利用可能な部品を構築するためのコストの問題
- (2) 再利用部品の品質の問題

ソフトウェアを再利用可能な部品として構築するためには、当然余分な費用と時間がかかる。費用で

は50%、時間は30%程度余分にかかるという調査もある。つまり、厳しいスケジュールと経費カットの圧力の下で、再利用可能な部品をつくるための時間と費用を捻出することには強い抵抗がある。

これは、現在のところ、ソフトウェア部品に市場価値がないことが最大の原因である。超流通では、ソフトウェア部品を自由に市場に供給し対価が得られるようになるため、ソフトウェア開発過程で生成される再利用可能な部品がそのまま市場価値を持つ。再利用のための余分のコストが収益をあげるための当然の経費として、企業活動に不可欠なものと認められるようになる。

また、再利用を成功させるためには、ソフトウェアの部品の信頼性について、それらを利用した時の障害発生率が0であるか、または、それに近いレベルに到達していることが保証されている必要がある。

マイクロプロセッサの高い信頼性は、著しい量産効果と広い範囲での利用によって達成されている。ソフトウェアの信頼性が低い最大の原因は、マイクロプロセッサほどに量産されず、広い利用がないことである。これに対して、超流通では、ソフトウェア部品が自由に流通され、これまでとは比較にならない程広範囲に利用されるようになり、信頼性の低い部品は市場において自然淘汰されていくことになる。

例えば、ワードプロセッシングソフトウェアを開発・供給しているメーカーがその開発過程において生成されたソフトウェア部品も供給する場合を考えてみる。内部処理は、全く同じモジュールを使い、ユーザインタフェースのみ異なるというアプリケーションソフトウェアが多数でてくるだろう。そのことによって、ユーザインタフェースのデザインに優れたメーカーと部品を供給したメーカーの双方が利益を得ることが可能である。また、そのソフトウェア部品は、広い範囲の利用によって、より信頼性の高い部品へと改善されていくだろう。

このようなソフトウェア部品の市場においても、3.1で述べたような、マルチメディアの直面している著作権処理の問題を解決しなければならないことに留意するべきである。

4. ソフトウェア部品の流通

4.1 ソフトウェア部品価格の検討

アプリケーションソフトウェアの課金については、2.2で述べたような様々な課金方式が検討されてきたが、利用者の使い勝手を考慮すると、単純な価格設定が好まれよう。したがって、アプリケーションソフトウェアの課金は、おそらく1回の利用当たりの料金か時間当たりの料金またはそれらの組み合わせとして提供されるようになる。また、利用額の増加に応じて使用料金が減額されるサービスや割引などは、従来の有体物の取り引きにおいても一般的であり、歓迎されるだろう。

このような前提に基づいて、ソフトウェア部品の課金について検討する。以下では、ソフトウェア部品は、複数のオブジェクトモジュールを含んだライブラリの形式とする。そのようなソフトウェア部品と、それらをリンクしたアプリケーションソフトウェアが流通することを想定している。つぎのような課金方式が考えられる。

(1) 売切り制課金

ソフトウェア部品をリンクしているアプリケーションソフトウェアが、利用者の超流通計算機で初めて実行されたときのみに、ソフトウェア部品の使用を記録する。これはソフトウェア部品の買い取りに対応している。

回数に応じた課金を行うアプリケーションソフトウェアが、このような売切り制のソフトウェア部品をリンクしている場合には、初めてアプリケーションソフトウェアを利用するとき、ソフトウェア部品の価格すなわち課金額がアプリケーションソフトウェアの1回当たりの課金額を上回ることがあり得る。このような場合は、利用者に例えば、「初回のみA社製部品の使用料金 XC_r ¹が必要です。2回目の使用からは、1回当たり YC_r となります。」というようなメッセージを出力し、継続するかどうかを決めることができる。

使用記録を回収し料金を徴収・分配をする機関でアプリケーションソフトウェア使用料金とソフトウェア部品の使用料金とを調整することによって、利用者からは、ソフトウェア部品

¹ C_r は、超流通ソフトウェアの使用料を表す単位

の使用料金を見えなくすることも可能である。この場合、使用記録回収機関の作業が多少複雑になる。

- (2) アプリケーションソフトウェア実行同期課金
アプリケーションソフトウェアが実行されるたびに、リンクされているすべてのソフトウェア部品の使用が記録される。実際にそのソフトウェア部品に含まれるモジュールが呼ばれたかどうか、また、何回呼ばれたかには依存しない。

- (3) アプリケーションソフトウェア実行非同期課金

リンクされたソフトウェア部品が利用されるたびに課金する。このとき、複数のモジュールを含んだソフトウェア部品単位または、モジュール単位で記録を残す場合がある。

一般に、ソフトウェアの価格を決めるための最も重要な要因は開発コストである。例えば、複数のモジュールから構成されるアプリケーションソフトウェアについて考える。使用頻度の低いモジュールの開発コストが低いというわけではなく、各関数の使用頻度と開発コストとの相関関係を一般的に見出すことはできないであろう。したがって、一般的なソフトウェア部品に対して、利用頻度による課金を適用することは合理的ではない。

このような観点から、ソフトウェア部品がリンクされているアプリケーションソフトウェアが起動された時点で、そのソフトウェア部品に含まれるモジュールが呼び出されるかどうかに関係なく課金されるアプリケーションソフトウェア実行同期課金が、提供者にとってわかりやすい。

一般に、アプリケーションソフトウェアの動作中に何回ソフトウェア部品が利用されるかは、実際に実行してみなければわからない。また、内部の処理がどのように実現されているかについて、通常、利用者が意識することはない。(3)の場合は、利用者は実際にアプリケーションソフトウェアを使用した後でなければ利用料金が把握できないことになる。したがって、利用者の立場からも、アプリケーションソフトウェアの使用料金をあらかじめ確認できるアプリケーションソフトウェア実行同期課金が歓迎されよう。

しかし、使用頻度は低いが開発コストが極めて大

きいものがあることも事実である。例えば、特許を取得したアルゴリズムを実装したソフトウェア部品などである。このような場合は、使用頻度による課金を指定し、利用者に加算料金があることを知らせ了解を得た後、アプリケーションソフトウェアの実行を継続するというようなメカニズムを実装することは困難ではない。

4.2 ソフトウェア部品の課金方式の提案

超流通ラベルとそれを解釈するためのメカニズムに関して簡単な例を示す。

現在、つぎのようなプロトタイプを開発中である。SUN ワークステーションのネットワークアプリケーションソフトウェアとして動作し、超流通ラベルリーダは、UNIX 上の一つのプロセスとして実行され、アプリケーションソフトウェアと超流通ラベルリーダ間の通信は UNIX のプロセス間通信を利用して実装される。

ソフトウェア部品の典型として、スタティックリンク用のライブラリとダイナミックリンク用ライブラリの二つを考える。スタティックリンク用のライブラリを用いるアプリケーションソフトウェアは、アプリケーションソフトウェアに添付された超流通ラベルからライブラリの超流通ラベルへ直接リンクを張った状態で提供されるモデルを考える。このことは、アプリケーションソフトウェア提供者が意図しない改変を防止するために役立つ。また、ダイナミックリンクライブラリを用いるアプリケーションソフトウェアの場合には、アプリケーションソフトウェアの超流通ラベルとライブラリのラベルは直接リンクされていないモデルを考える。このことは、利用者がダイナミックリンクライブラリを自由に入れ換えられることを意味している。利用者は、より安価で性能の高いダイナミックリンクライブラリに乗り換えて、アプリケーションソフトウェアを利用することができるようになる。

図 4において、

- アプリケーションソフトウェアの課金は 1 回当たり 10Cr,
- ライブラリ A の料金は 0.5Cr,
- ライブラリ B の料金は 0.2Cr,
- ライブラリ C の料金は 0.3Cr

のアプリケーションソフトウェア実行同期課金の例

を考える。もちろん、ライブラリの中で別のライブラリがコールされる場合もありうる。

アプリケーションソフトウェアが起動されると、つぎのような動作をする^[5]。

- (1) 超流通アプリケーションソフトウェアが、UNIXカーネルによってメモリ上にマップされる。
- (2) 実行時リンカが、UNIXカーネルによってメモリ上にマップされ、カーネルから実行時リンカに制御が渡される。
- (3) 実行時リンカによって、ダイナミックリンクライブラリがマップされ、各モジュールの初期化セクションが呼び出される。この時点で、ダイナミックリンクライブラリの超流通ラベルがラベルリーダプロセスに転送される。
- (4) 超流通アプリケーションソフトウェアに制御がわたる。この時点で、アプリケーションソフトウェアの超流通ラベルがラベルリーダに転送される。
- (5) 超流通アプリケーションソフトウェアおよびダイナミックリンクライブラリは、ラベルリーダプロセスと通信しながら動作する。

超流通ラベルリーダは、例えば、超流通ラベルに含まれるチェックルーチンを用いてアプリケーションソフトウェアと通信することによって、転送されたラベルが正当なものかどうかを確認する。正当なラベルであれば、使用記録に関する処理が始められる。ライブラリのラベルが送られていない状態ではライブラリは動作しない。ライブラリが呼び出された時点でアプリケーションソフトウェアの実行が中断される。

超流通ラベルの表示は、超流通ユーザユーティリティプログラムの機能として提供される。ユーザユーティリティは、まず、アプリケーションソフトウェアの超流通ラベルをラベルリーダに送る。ラベルリーダからの応答にしたがって、ライブラリの超流通ラベルを転送する。ラベルリーダでは、アプリケーションソフトウェアおよびライブラリの料金を計算しユーティリティプログラムに転送する。上記例の場合は、利用料金は1回の使用で11Crと表示される。

5. マルチメディアへの応用の可能性

5.1 MHEG モデル

MHEG(Multimedia and Hypermedia Information Coding Expert Group) は、現在、各社独自のマルチメディアプラットフォームの環境上で、マルチメディアデータを共有するためのデータの交換方式の標準として、ISO/IEC において開発されている^[6,7]。

マルチメディアパッケージとは、マルチメディア素材とそれらをどのようなタイミングで表示・再生するかを記述するシナリオが格納されたものである。MHEG では、オブジェクト指向のアプローチを採用しており、パッケージの構成要素であるオブジェクト群が互いにメッセージをやりとりしながら、シナリオを実行していくというモデルになっている。

MHEG は、あるメーカーのプラットフォームと別のメーカーのプラットフォーム間でデータの交換を可能とするための中間言語の役割を果たすものである。あるプラットフォームで記述されたマルチメディアパッケージを別のプラットフォーム上で実行しようとする時、MHEG 形式を通して、別のプラットフォームに変換される。

5.2 マルチメディアのための超流通

超流通をマルチメディア素材に適用するためには、

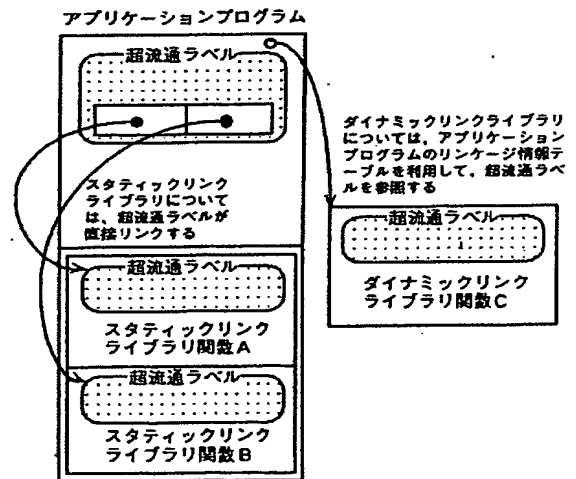


図4: プロトタイプにおけるアプリケーションソフトウェアとライブラリ関数の超流通ラベル

つぎのような点を考慮する必要がある(図5)。

- (1) 各々のマルチメディアオブジェクトに超流通ラベルを添付する。
- (2) リンクオブジェクトと各コンテンツオブジェクトの間に超流通ラベルの転送メッセージを実装する。
- (3) リンクオブジェクトと超流通ラベルリーダとの間のプロトコルを規定する。

また、マルチメディア素材については、「貴重マルチメディア部品」と「汎用マルチメディア部品」に類別する必要がある。貴重マルチメディア部品は、例えば有名写真家の撮った静止画などが考えられよう。それを鑑賞するだけで商品価値があるものである。このようなマルチメディア部品を提供する場合には、利用者が知りたい時には、いつでもその料金を表示するための機能が必要である。

汎用マルチメディア部品は、このような価値はもたないが、マルチメディアパッケージを作成するためには不可欠なデータである。

マルチメディア素材は本来両者の性格を合わせ持っている。動画の背景に流れる音楽などは、汎用マルチメディア部品として取り扱えた方が、パッケージ製作者や利用者にとっては快適である。

6. むすび

本論文では、ソフトウェア部品やマルチメディア素材といった電子オブジェクトを円滑に流通させるための超流通システムについて述べた。本論文で述

べたようなシステムが実現されれば、電子オブジェクトを自由に市場に供給し対価を得られるようになり、マルチメディアにおける権利処理が円滑に行われるようになる。

アプリケーションソフトウェアのための超流通は、潜在的需要を100%近くまで掘り起こす技術であると考えられてきた。これに対して、電子オブジェクトのための超流通は、ソフトウェア部品化の潜在的可能性を引き出し、ソフトウェア開発費を劇的に引き下げる技術となる。

参考文献

- [1] 河原正治, 森亮一: “超流通アーキテクチャのためのプロトタイプ II”, 情報処理学会情報システム研究会, 27-6, pp.1-9 (1990)
- [2] Ryoichi Mori, Masaji Kawahara: “Superdistribution: The Concept and the Architecture”, The Trans. of IEICE, Vol E73, No.7, pp. 1133-1146 (1990)
- [3] 森亮一, 河原正治: “歴史的必然としての超流通”, 情報処理学会 超編集・超流通・超管理のアーキテクチャシンポジウム 講演論文集 pp.67-76 (1994)
- [4] Capers Jones: “Economics of software reuse” IEEE Computer, pp.106-107, July 1994
- [5] Sun Microsystems, Inc.: “SunOS 5.1 Link and Libraries Manual”, pp.19-41 (1992)
- [6] Françoise Colaitis: “Opening Up Multimedia: Object Exchange with MHEG”, IEEE Multimedia, Summer 1994, pp.80-84 (1994)
- [7] 山岸靖明: “マルチメディア/ハイパメディア符号化”, 電子情報通信学会誌, Vol.77, No.8 pp.840-843 (1994)

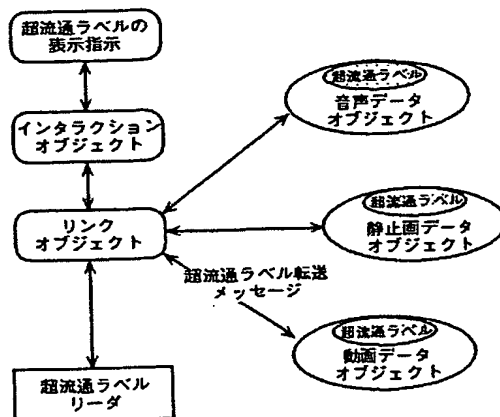


図5: MHEG モデルと超流通

複写をされる方に

本誌に掲載された著作物は、政令が指定した図書館で行うコピーサービスや、教育機関で教授者が講義に利用する複写をする場合等、著作権法で認められた例外を除き、著作権者に無断で複写すると違法になります。そこで、本著作物を合法的に複写するには、著作権者から複写に関する権利の委託を受けている次の団体と、複写をする人またはその人が所属する企業・団体等との間で、包括的な許諾契約を結ぶようにして下さい。

日本複写権センター 〒107 東京都港区北青山3-3-7 第一青山ビル
Phone 03-3401-2382, 03-3475-5618 Fax 03-3401-2386

Notice about photocopying

In order to photocopy any work from this publication legally, you or your organization needs to obtain permission from the following organization that has been delegated for the copyright clearance by the copyright owner of this publication.

[Japan] Japan Reprographic Rights Center
Daiichi Aoyama Bldg. 3F,
3-3-7, Kita-Aoyama, Minato-ku, Tokyo, 107 Japan
Phone +81-3-3401-2382 Telefax +81-3-3401-2386

[U.S.A.] Copyright Clearance Center, Inc.
222 Rosewood Drive, Danvers, MA 01923, USA
Phone (508) 750-8400 Telefax (508) 750-4744

電子情報通信学会技術研究報告 信学技報 Vol. 94 No. 240 IEICE Technical Report 1994年9月21日発行

©電子情報通信学会 1994

Copyright : © 1994 by the Institute of Electronics, Information and Communication Engineers (IEICE)

発行人 東京都港区芝公園3丁目5番8号 機械振興会館内

社団法人 電子情報通信学会 事務局長 田村 恵一

発行所 東京都港区芝公園3丁目5番8号

社団法人 電子情報通信学会 電話 (03) 3433-6691
郵便振替口座 東京2-35300番

The Institute of Electronics, Information and Communication Engineers,
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku,
TOKYO, 105 JAPAN

本技術研究報告に掲載された論文の著作権は(社)電子情報通信学会に帰属します。

Copyright and reproduction permission: All rights are reserved and no part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission in writing from the publisher. Notwithstanding, instructors are permitted to photocopy isolated articles for noncommercial classroom use without fee.

**Technical Report of the Institute of Electronics,
Information and Communication Engineers**

ISEC 94-13~22
(Information Security)
21 September 1994

[National Diet Library datestamp: 7 October 1994]

EIC Institute of Electronics,
Information and Communication Engineers

Technical Report of IEICE
ISEC 94-15 (1994-09)

Consideration of a Method of Charging for Electronic Objects in Superdistribution

Masaji Kawahara
Research Center on Educational Media
Tsukuba College of Technology
4-12 Kasuga, Tsukuba, Ibaraki

Abstract: This paper describes a superdistribution system to freely distribute electronic objects: software parts and data for multimedia use. Superdistribution for application software is a technology that can bring the potential demand for the software to nearly the 100% level. In contrast, superdistribution for electronic objects is a technology that can draw out the latent potential of the use of software parts and bring about a dramatic reduction of software development costs through the use of standard software parts by software manufacturers and users. A prototype that uses a network with UNIX workstations is now under development.

1. Introduction

Microprocessors became possible only with the mass production of computer hardware. Their performance and sales have grown vigorously over the past 20 years, and microprocessors have come to be used as general-purpose parts, which has brought about a sharp drop in the price of information equipment.

This sharp drop in prices has not occurred in software development. An innate superior characteristic, which is that copies can be made quickly and cheaply with the same quality as the original, has become a source of disorder. For that reason, microprocessors are normally mass produced by the millions, but that is exceptional for software. Further, the parts industry, which is a natural and indispensable base for hardware, has not reached a practical scale in the domain of software.

In the multimedia field, moreover, that problem is accompanied by the problem of copyrights for multimedia data (raw material); creators of multimedia software have pointed out the difficulty of obtaining the agreement of all rights holders.

To firmly establish a software parts industry and encourage the spread of multimedia, it is essential to create a technical and legal base that will freely supply software parts and multimedia data and secure compensation for them.

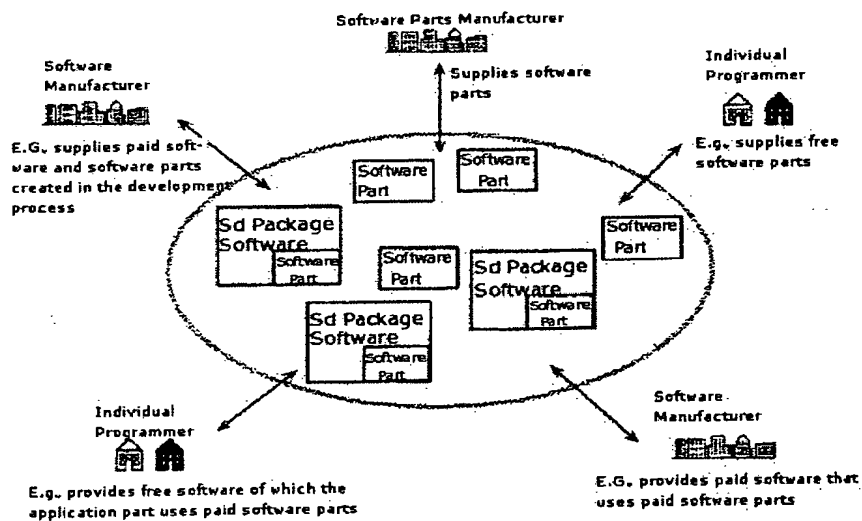


Fig. 1 Superdistribution Software Parts Community

Superdistribution is a basic technology to bring about the free distribution and use of digital information. Superdistribution will enable mass production of software, which until now has been considered impossible. Superdistribution is a system that does not control the possession of software, but rather monitors usage records every time the

software is used, imposes charges based on collection of the usage records, and then redistributes the revenue. To date, prototype development has gone forward regarding superdistribution for application software on personal computers.^[1,2,3]

This paper discusses a method of charging for software parts that has the purpose of encouraging the use of software as parts and easing the treatment of rights within multimedia. Superdistribution of application software will bring about increased earnings from mass production of software. In contrast, superdistribution for software parts will bring about a dramatic reduction of software development costs. The paper also speaks of the possibility of a "superdistribution software parts community" (fig. 1) that will allow manufacturers and individuals to freely provide a variety of software parts and application software parts and to receive earnings for them.

2. Superdistribution and "the Superdistribution Label"

In order to realize superdistribution, it is necessary to electronically add the conditions of use to the digital information that is distributed. These conditions of use are called "the superdistribution label." The following items are of a nature appropriate to a superdistribution label.^[3]

- (1) That when digital information that has a superdistribution label refers to other digital information that has a superdistribution label, the digital information referred to will be listed automatically.
- (2) That items resistant to counterfeiting, removal, and improper additions.
- (3) That there is a way for introduction of superdistribution labels having compatability with data structures that predate superdistribution, such as CD-ROM, MD [minidisc], and DCC [digital compact cassette].

That electronic objects are combined and distributed as software parts or multimedia data and a superdistribution label has been considered as a basic matter to impose usage charges shows the nature in (1) above.

"Superdistribution label readers" can be added to the equipment that processes distributed digital information. These monitor the usage records of digital information in accordance with what is stated on the digital label.

2.2 Superdistribution to Charge for Application Software

Within superdistribution there is the possibility of various forms of applying charges that were not possible in transactions with tangibles in the past. Such things as trial use charges, per-use charges, automatic purchases, post-purchase refunds, special permissions, and free products with usage reports required have been studied. Given the

distinctive features of software, it is possible to choose freely in a range from simple charges to something more complex.

Further, a system has been proposed to minimize costs at introduction by applying the existing credit card settlement system to collection and distribution of usage charges.

To date, two prototypes have been developed to monitor usage records for application software in personal computers. One uses the computer's existing serial interface to mount the superdistribution label reader, and one uses a co-processor interface. The latter has the following advantages.

- There is less overhead, and it can be applied to multi-tasking or multi-processor computers.
- The cost is being reduced by moving to one-chip architecture.

How charging for application software can be done is explained below, using as an example the operation of a superdistribution computer prototype that has been developed.

Figures 2 and 3 show prototype screens. The user first attaches a superdistribution label reader to the computer. When the program is executed in the superdistribution computer, the label reader creates/monitors a usage record based on the superdistribution label attached to the program. The usage charge is collected when the user transmits the usage record in accordance with conditions that have been set in advance.

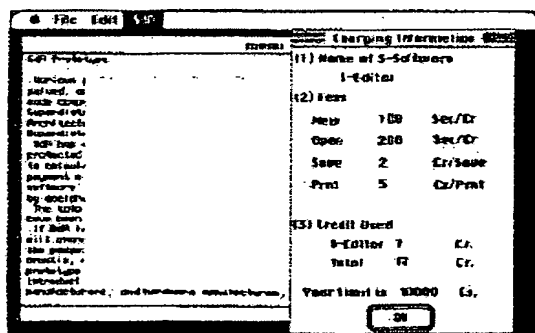


Fig. 2 Display of Superdistribution Label

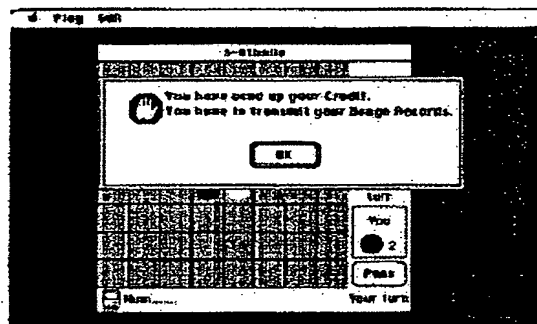


Fig. 3 Program Termination Screen

Figure 2 is an example of the display of the superdistribution label of superdistribution software operating on the prototype computer.

This is done by means of a superdistribution user utility supplied along with the superdistribution label reader. The charges set for this editor software were brought into being as one example of hourly charges and charges by function.

Collection of the usage records can be realized by various methods. The prototype computer that is working now uses the method of setting an upper limit for the usage amount, in advance, in the user's superdistribution label. When the accumulated usage

amount reaches the upper limit, the superdistribution label reader terminates execution of the superdistribution program and urges transmission of the usage record. The user can resume use of the program by transmitting the usage record. Figure 3 shows an example of the program's termination screen.

The example above shows charging for a computer program, but a similar mechanism can be used to charge for digital information.

3. The Objective of Superdistribution for Electronic Objects

This paragraph considers the effects of superdistribution for electronic objects from the two aspects of processing multimedia rights and reducing software development costs.

3.1 Rights Processing in Multimedia

Multimedia is noted as a software technology that does digital processing of texts, sounds, and images and brings about bi-directionality between users. In order to popularize multimedia, it is desirable to provide highly original multimedia software with which users can freely process and synthesize multimedia data.

Accordingly, there is a need to be able to freely use large volumes of diverse digital information. Given the current legal and technical situations, however, there is no question that the problem of illegal copies of computer software and the chaos of digital audio will cause difficulties in this field as well.

One further problem is that creators of multimedia software have to obtain individually the agreement of the copyright holders for all data. Obtaining the permission of numerous rights holders has great costs in both time and money.

A method to automatically resolve these problems for multimedia data and software parts, using electronic means, is discussed below; superdistribution is a technical foundation for this purpose.

3.2 Reduction of software Development Costs

It is widely recognized that reduction of price by means of mass production is an important objective in regard to microprocessors. In the case of software, however, where the effect of mass production is far more direct (an n -fold increase in sales can mean a change of unit price to $1/n$), there has not been adequate research to increase demand numbers. The superdistribution technology for applications described in 2.2 will make it possible to elicit the maximum demand for this software.

There have been numerous attempts so far to make software development more efficient, including the investment of high-performance hardware resources. However, using superior tools in the course of software development and trying to improve the development process have clearly not been able to reduce costs dramatically.

Research on a move to software parts and technology for reuse, with the purpose of reducing software development costs, is very active. However, to date there has been no debate at all regarding a technical foundation that would allow such parts to be freely provided to the market as digital objects and compensation to be received. Even if there is a move to software parts and technology for reuse at the source code level, there will be no great economic impact as long as they are kept within the organization; there will not be sufficient force to overcome the barriers to reuse that are stated below. Only when a foundation is prepared that allows software parts that are produced in the course of software development to be freely supplied and compensated can the benefits of software parts and reuse be obtained.

The following items have been pointed out as major barriers to promotion of software reuse.^[4]

- (1) The problem of the cost of building reusable parts
- (2) The problem of quality of reused parts

Naturally, extra expense and time are required to build software as reusable parts. A survey shows the expense is increased by about 50% and the time by about 30%. In other words, under the pressures of tight schedules and cuts in outlays, there is strong resistance to scraping together the time and money to build reusable parts.

This is the major reason that software parts have no market value at present. Because software parts would be freely supplied to the market and compensated under superdistribution, reusable software parts produced in the course of software development would have market value in that state. The extra cost of reusability would come to be recognized as indispensable to company operations as a natural expense for the purpose of producing revenue.

Moreover, in order to make a success of reuse it will be necessary, in regard to the reliability of software parts, to assure a failure rate of zero when they are used or to reach a level near that.

The high reliability of microprocessors was achieved through the marked effects of mass production and the wide range of uses. The major reason for the low reliability of software is that it is not mass produced like microprocessors and it is not widely usable. Under superdistribution, on the other hand, software parts would be freely distributed and would be used on an unprecedentedly wide scale, and parts with low reliability would of course be weeded out in the marketplace.

Consider, for example, a case in which a manufacturer who develops and supplies word processing software supplying software parts produced in the course of development. There could be a number of software applications that all used the same module for internal processing and differed only in the user interface. That would mean the possibility of profits for both a manufacturer who excels at design of user interfaces and

the maker who supplies the part. Further, because the part would be used more widely, it would improve to become a part with higher reliability.

It should be remembered that, even in this sort of software parts market, it will be necessary to resolve the copyright problems mentioned in 3.1 that confront multimedia.

4. Distribution of Software Parts

4.1 Consideration of software Parts Pricing

Consideration has been given to various methods, as mentioned in 2.2, of charging for application software, but considering the strong self-interest of users, it is preferable to set a simple price. Accordingly, the method of charging for application software will probably be by the number of times used, the length of time used, or a combination of the two. Moreover, such things as a service or discount to reduce the usage charge in response to increased use is common in traditional transactions in tangibles, and would be welcome.

Such a premise is the basis for consideration of charging for software parts. Software parts are looked at below in the form of a library that includes a number of object modules. The distribution of such software parts and application software that links them is assumed. The following charging methods are conceivable.

(1) Charging at Time of Sale

The application software that links the software parts records the use of a software part only when it is first executed in the user's superdistribution computer. This corresponds to purchase of the software part.

In the event that this sort of time-of-sale software part is linked to application software that is charge for each use, the charge per use of the application software will be increased by the price of the software part the first time the application software is used. In that case, a message such as "A one-time charge of X Cr¹ is necessary for a part made by Company A. For the second and subsequent uses, the charge will be Y Cr per use" will be output to the user, who will then be able to decide whether to continue.

The institution that collects usage records and receives and distributes usage charges will adjust the usage charges for the application software and the software part, and so the usage charge for the software part will not have to be seen by the user. In this case, the work of the usage record collection institution is somewhat more complicated.

(2) Charge Synchronized with Execution of Application Software

Every time the application software is executed, the use of all linked software parts is recorded. It does not matter whether the module including the software part is called or how often it is called.

¹ Cr is the unit that represents usage charges for superdistribution software.

(3) Charge Not Synchronized with Execution of Application Software

A charge is made only when the linked software part is used. The record made at that time may be based on a software part that includes several modules, or it may be based on individual modules.

Development costs are generally the most important factor in determining the price of software. Consider the example of application software that comprises several modules. The cost of developing an infrequently used module is not necessarily lower; it is generally impossible to discover a relationship between the development cost and the frequency of use of each function. Consequently, it is generally not reasonable to apply charges for software parts based on their frequency of use.

From that perspective, the charge synchronized with the execution of application software, which is made when the application software to which the software part is linked is started up and which is unrelated to whether the modules included in the software part are called, is most understandable for the suppliers.

Without actually executing application software, it cannot be known how many times a software part will be used during its operation. Moreover, the user normally is not aware of how the internal working of the software comes about. In the case of (3), the user will be unable to grasp what the usage charge will be until he has actually used the software. From the user's standpoint, therefore, the charge not synchronized with the execution of application software should be welcome because the application software usage charge can be known in advance.

However, the fact is that there are things for which the frequency of use is low but development costs are quite high. For example, there are software parts that employ patented algorithms. In such cases, there would be no difficulty in designating charges by frequency of use and implementing a mechanism to inform the user of an additional charge and get his approval before continuing execution of the application software.

4.2 Proposal of Charge Method for Software Parts

Here is a simple example of a superdistribution label and a mechanism to interpret it.

The following prototype is under development now. It runs as network application software on a Sun workstation; the superdistribution label reader is executed as one process under UNIX, and communication between the application software and the superdistribution label reader employs UNIX inter-process communications.

Two models for software parts are static link libraries and dynamic link libraries. A conceivable model for application software using a static link library is provision of a direct link from the superdistribution label applied to the application software to the superdistribution label applied to the library. This is useful for preventing unintentional modifications by the application software provider. Further, in the case of application software using dynamic link libraries, the model does not have a direct link between the

superdistribution label of the application software and the library label. That means that the user can freely replace libraries. The user would be able to use the application software after changing over to a less expensive and higher performance dynamic link library.

Figure 4 is an example of the charge synchronized with execution of the application software, in which

- the charge for the application software is 10 Cr per use,
- the charge for library A is 0.5 Cr,
- the charge for library B is 0.2 Cr, and
- the charge for library C is 0.3 Cr.

There are, of course, times when one library will call another library.

The following actions take place when the application software starts up.^[5]

- 1) The superdistribution application software is mapped into memory by the UNIX kernel.
- 2) The runtime linker is linked into memory by the UNIX kernel, and control is transferred from the kernel to the runtime linker.
- 3) The dynamic link library is mapped by the runtime linker, and the initialization section of each module is called. At this time, the superdistribution label of the dynamic link library is sent to the label reader process.
- 4) Control passes to the superdistribution application software. At this time, the superdistribution label of the application software is sent to the superdistribution label reader.
- 5) The superdistribution application software and the dynamic link library operate while communicating with the label reader process.

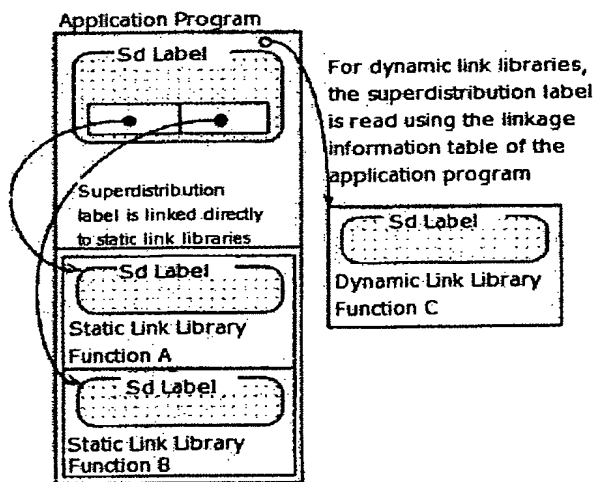


Fig. 4 Superdistribution Labels of Application Software and Library Functions in Prototype

The superdistribution label reader uses, for example, a check routine included in the superdistribution label and communicates with the application software, and so confirms whether the transmitted label is legitimate. If it is a legitimate label, processing of the usage record begins. The library will not operate as long as a library label has not been sent; execution of the application software is suspended at the point when the library is called.

The superdistribution label display is provided to function as a superdistribution user utility program. The user utility first sends the superdistribution label of the application software to the label reader.

The superdistribution labels of the libraries are sent in accordance with the response from

the label reader. The label reader calculates the charges for the application software and the libraries and sends them to the utility program. In the example above, it would display a usage charge of 11 Cr for one use.

5. Potential for Application to Multimedia

5.1 MHEG Model

In ISO/IEC, MHEG (the Multimedia and Hypermedia Information Coding Expert Group) has been developing multimedia data in the environment of the multimedia platforms of individual companies, as a standard exchange method for data to be shared.^[6,7]

A multimedia package is a container for multimedia data and a scenario that describes the timing with which to display or reproduce the data. MHEG has adopted the object-oriented approach; its model is one that executes the scenario as messages are exchanged among the object that are the elements constituting the package.

MHEG performs the role of an intermediate language to enable the exchange of data between one manufacturer's platform and the platforms of other manufacturers. When a multimedia package described on a certain platform is to be executed on a different platform, it is converted for the other platform by way of the MHEG format.

5.2 Superdistribution for Multimedia

The following points must be considered in order to apply superdistribution to multimedia data. (Fig. 5)

- 1) Adding a superdistribution label to each multimedia object
- 2) Transmitting superdistribution label messages between the link object and individual content objects
- 3) Regulating the protocol between the link object and the superdistribution label reader

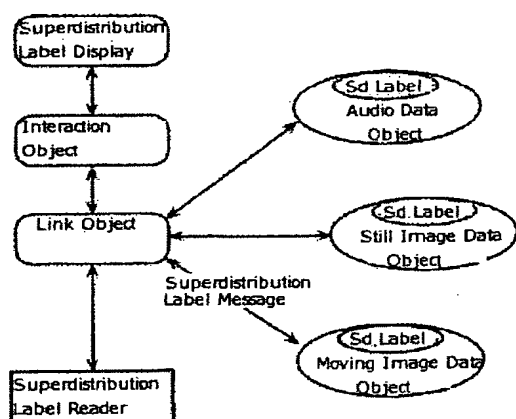


Fig. 5 MHEG Model and Superdistribution

Further, it will be necessary to categorize multimedia data as "valuable multimedia parts" or "general multimedia parts." Such things as still images from famous photographers can be considered valuable multimedia parts. These have commercial value just because they are admired. If multimedia parts of that sort are provided, displaying the charge whenever the user wants to know it will be a necessary function.

General multimedia parts do not have that kind of value, but they are indispensable data for creating multimedia packages.

A dual nature, both valuable and general, is intrinsic to multimedia data. Changing the music that plays in the background of moving images, for example, as a general multimedia part, may be agreeable to both the package creator and the user.

6. Conclusion

This paper has discussed superdistribution for freely distributing the electronic objects, software parts and multimedia data. If the system mentioned in this paper is realized, it will become possible to freely provide electronic objects to the market and receive compensation, and to process multimedia rights harmoniously.

Superdistribution for application software has been thought of as a technology to draw out nearly 100% of potential demand. Superdistribution for electronic objects, on the other hand, is a technology to elicit the potential demand for a move to software parts and to dramatically reduce software development costs.

Reference Documents

1. M. Kawahara, R. Mori: "Prototype II for Superdistribution Architecture," IPSJ Information Systems SIG, 27-6, pp. 1-9, (1990).
2. R. Mori, M. Kawahara: "Superdistribution: The Concept and the Architecture," The Trans. of IEICE, Vol. E73, No. 7, pp. 1133-1146 (1990).
3. R. Mori, M. Kawahara: "Superdistribution as a Historical Necessity," IPSJ Symposium on Supercompilation/Superdistribution/Superadministration Architecture Papers, pp.67-76 (1994)
4. C. Jones: "Economics of Software Reuse," IEEE Computer, pp. 106-107, July 1994.
5. Sun Microsystems, Inc.: "SunOS 5.1 Linker and Libraries Manual," pp. 19-41 (1992).
6. F. Colaitis: "Opening Up Multimedia Object Exchange with MHEG," <http://www.vldb.org/dblp/db/journals/ieeemm/ieeemm1.html> - Colaitis94IEEE Multimedia, Summer 1994, pp. 80-84 (1994).
7. Y. Yamagishi: "Multimedia/Hypermedia Encoding," Trans. of IEICE, Vol. 77, No. 8, pp. 840-843 (1994).

**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.

事務管理

富士通

昭和57年12月3日 第三版印刷物 昭和57年6月10日印刷 平成元年7月1日発行 32頁 317円
発行所 並行販売 並行販売 並行販売 並行販売 並行販売 並行販売 並行販売 並行販売 並行販売 並行販売
並行販売 並行販売 並行販売 並行販売 並行販売 並行販売 並行販売 並行販売 並行販売 並行販売

新・システムの実務誌



【実績No.1のオフィスプロセッサを、さらに強化。】

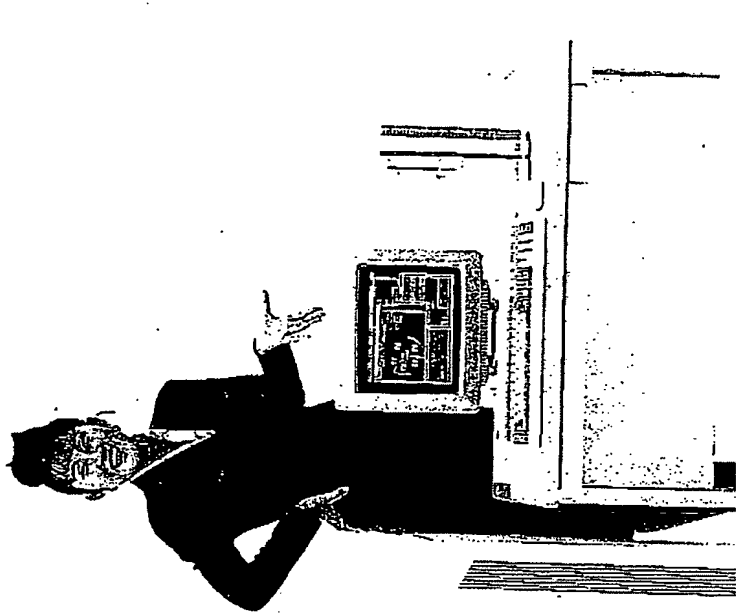
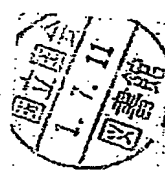
カバーレージをさらに拡大、

ラップトップが新登場。

386種類の豊富なソフト。

新しいニュースです。実績No.1のオフィスプロセッサシリーズに、最上位機種とラップトップが登場。使いやすさも拡張性も、いっぴんと高まりました。32ビットワークステーションによるマルチタスク環境、マルチプロセス方式の採用、高水準のルーティン化、データベースなど、新シリーズは最先端の機能を提供します。いいコンピュータをお選びください。理想的な仕事環境を構築するその時、新シリーズは貴社のパートナーとなります。

新・



事務管理

1989
Vol.28
No.8

増大号
特集
最新ワークステーション活用術
—OWSユーザーのための導入ガイド—

特別定価 1350円

又々-163

BEST AVAILABLE COPY

WSIはこう使おう! — 解説

ワークステーションの機能と活用 —三菱マルチWSを中心に—

武田 浩良*

最近のワークステーションの機能は、非常に広範で高度になってきているが、それらの機能をユーザーが十分に使いこなすにはどうしたらよいか。以下では、三菱電機のワークステーションを例にとりて述べるが、機能的には他社のものも同様の内容をもっていると思われる(編者部)。

三菱マルチワークステーション(MWS)は、オフィスにおける高次元な情報活用と情報共有を効果的に実現する、異なるホストコンピュータを自

在に結合ネットワークで会社/部門/個人のレベルでの情報資源を共有し、だれでも必要とする情報を簡単な手段で、そして最適な形で手にとることができる。

統一された操作、機能の高められたOAソフトウェアなど、「しない仕事を、簡単にすぐできる」使いやすさを提供しており、企業の統合OAシステムの中核として広範囲分野で活用できる。

製品レバートリーは表1のとおり。なお、MWSは拡張日本語コンカレントCP/M-86とマルチモードMS-DOSの2種のOSをシステムの用途によって使い分けている。

MWSは、次のような要件のもとで、そのもつ機能・効果を発揮する。

- ①ホストと密接な連携をもつ端末システム
- ②ネットワーク統合システム
- ③高機能な要求される端末システム
 - ・1120×756ドット高解像度表示
 - ・マルチタスク、マルチジョブ機能で通信、

表1 三菱マルチワークステーションのレバートリー

機 種	モデル	CPU主/メモリー	固定ディスク
	CIV	80386	130~250MB



〒100 東京都千代田区千代田1-10-1 TEL 03(7)45-1948
内線 900
外部 03(7)45-1948

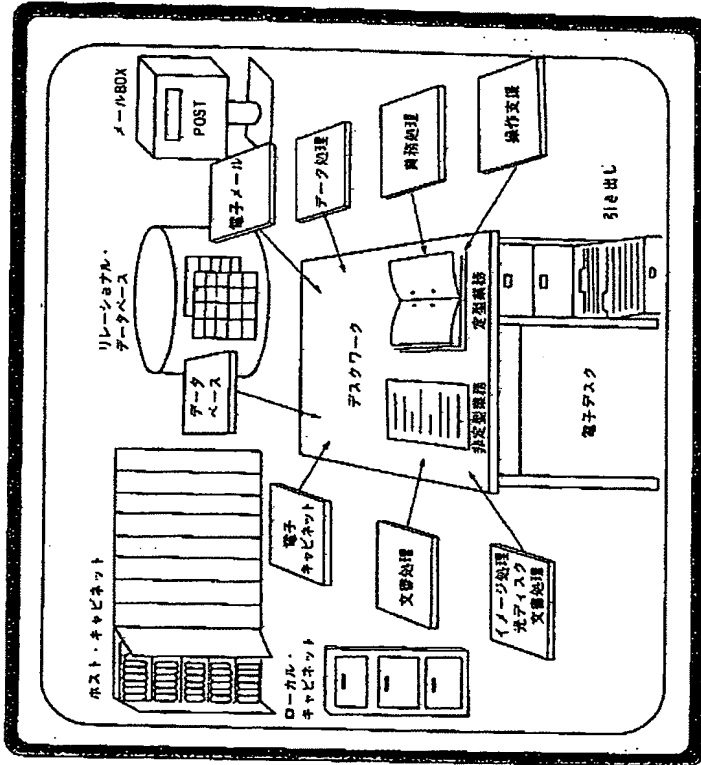


図1 総合OAシステムの概念

OA、業務など4つのジョブを同時並行処理

- ④付加価値機能を利用したシステム
 - ・イメージQCRシステム
 - ・イメージ処理システム
 - ・先ディクシステム
 - ・CADシステム

以下に、MWSの提供する機能と活用例を紹介する。

(1) 電子デスク

以下に述べるデータベースから操作変換までの機能を、ホストを認識することなく端末で統一された操作で動かせる。マルチウィンドウのもとにアイコンをマウスでクリックする簡単な操作で、ホストコンピュータにはほとんど負担をかけることなく各機能が利用できる。

(2) データベース

ホストのリレーショナルデータベースのデータをMWS側に抽出する機能。

(3) 電子キャビネット

ドキュメントをファイリング機能で、ホストとMWS両方のキャビネットが利用できる。

企業内統合OAシステム

ネットワークを通して効率的な運用を目指した統合OAシステムである。ホストとの分散・統合処理機能とともに、文書処理中心の非定形業務、

BEST AVAILABLE COPY

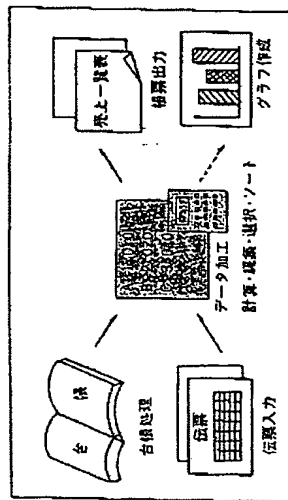


図5 業務処理サービス

ドキュメントにパスワードを付けて保護することができ、

(3) 検索機能
ドキュメントの属性情報(キーワードなど)をキーとして検索することができ、「あいまい参照文字」による、「あいまい検索」も可能。

2. 活用例
ホストキャビネットやローカルキャビネットの操作は基本的に同じである。また、光ディスク文書処理システムとの操作も同じである。アイコンを使用してビジュアルに操作することができ、電子キャビネットの活用事例として、電子機器を製造しているユーザーの設計情報ネットワークシステムについて以下に紹介する。

(1) システムのねらい
従来の手書きやスタンダードアロンのワープロによる設計ドキュメントの作成では、切り貼り作業による不効率や誤謬、連絡ミスによる設計仕様の遅延が多く発生していた。そこで、各種ツールと情報連携を活用した情報ネットワークシステムを構築した。

(2) 効果
次のようなメリットが生まれることを想定している。

- ① ワークステーションでの技術文書の作成効率化と高品質化
- ② ネットワークを利用した設計情報伝達の高

の増強

- ⑤ 設計情報一元化による計画・管理の合理化
これらのメリットは、開発・設計環境の改革にもつながる。

電子メールサービス

電子メールサービスは、ホストコンピュータを経由して文書などを他部門に配送する機能で、いわば、電子の郵便局といったシステムである。

1. 主な機能

(1) ノールの送信機能
ノールの送信とは、ドキュメント(電子キャビネットと同様にMWS側のファイルが対象)に、宛て先や送信属性を付けて送信(POST)に投函する機能である。

個人宛て、または同僚関係(最大32名)が可能な、送信属性には、メールアドレス、優先度(普通/至急)や秘密性(普通/秘密/機密)などの指定が可能である。

ノールの宛て先はユーザー選択機能や宛て先候補を使って簡単に指定できる。また、いつも定型的に送信している宛て先は、毎回指定しないで行える。

(2) ノールの受信機能
ノールの受信とは、受信ボックス(個人ノールボックス)を開いてノールの内容を確認する機能である。受信ボックスは個人単位でホストに設定できる。また、ホストコンピュータで管理しているため、受信状況や確認状況を一目で判断できる。

ノールボックスからノールを取り出せるのは、その持ち主だけで、第三者にのぞかれる心配はない。また、ノールボックスはホストコンピュータで一管理しているためで保護は万全。

(3) ノール検索機能
受信ボックスや表示版の中に保存されているノールを検索する機能で、ノール名や至急、返信要

きるユーザーを指定することができるので、利用目的に応じた格納を作成できる。格納機能によって、不特定多数のユーザーと情報交換ができる。

2. 活用例

電子メールサービスを使用して、オフィスとオフィスを結ぶことができる。このサービス機能の活用例は、次のとおり。

- ① メール機能を使って
報告書の送付、会議開催通知の配布(同報)、議事録の配布(同報)、問合せ(返信要求)、私的なこと、
- ② 掲示板機能を使って
プロジェクト内の情報交換、行事案内、社内ニュース、告示など。

業務処理サービス

画面と対話しながら、業務プログラムを作成するサービス機能で、作成された業務プログラムは、業務メニューアイコンから起動することができる(図5)。

1. 主な機能
業務処理サービスの主な機能は、次のとおりである。

- ① 台帳を作成する。
 - ② 伝票を入力する。
 - ③ データを加工する(計算、編集、選択、結合、並び替え)。
 - ④ 帳票を作成する。
- これらの業務は、以下のツールを利用して、一連の業務システムとして構築することができる。

- ・メニュー画面作成ツール
- ・伝票入力ツール
- ・ファイルメンテナンスツール
- ・ファイル検索ツール
- ・ソートツール
- ・プリント出力ツール

2. 主な機能

- (1) 業務プログラムを広く活用可能

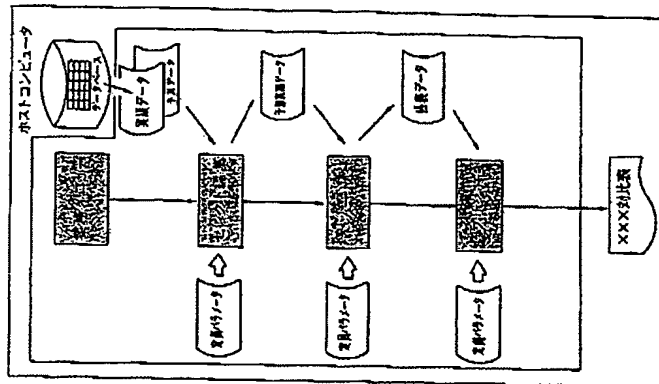


図6 業務処理サービスの利用例

- (3) ユーティリティとして使用可能
テストデータを作成する場合、ファイルメンテナンスツールを利用すると、容易にテストデータを作成できる。
- このように各種ツールは、ユーティリティソフトとして利用できる。
- (4) COBOLとの共有使用が可能
COBOLで作成したプログラムもシステムの一部として連動させることができる。
- (5) データのビジュアル化
Multigraphとの組合せにより、データを直線、表形式またはグラフとして表示できる。

3. 活用例

- ホストコンピュータのデータベースから引き出すデータやデータベースの運用状況を、

BEST AVAILABLE COPY

TOSHIBA

(私たちはV-7000の
テクノロジーを語ります。)

シリーズ—— NETWORK VNETとMMLにより、 高度な情報ネットワーク システムの提供。

経営環境の激しい変化の中を生む現代
企業は、問題解決能力の強化を求められ、
そのための経営を要するコンピュータは、
高度な情報ネットワークによる情報資源の共
有が求められることになる。

システム内の情報資源を 自由に活用できるVNET

この問題について我々は、V-7000シリーズ
同士をオンラインで結ぶVNETと、閉鎖化さ
れたMML(マイクローラインフレームワーク)に
よって克服することにした。

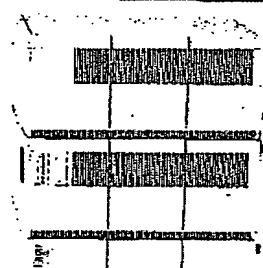
VNET、のもつリアルタイム処理(NAT)、
や「仮想網本機(NSF)」によってユーザは、
目的とするアプリケーションの場所(ノード)に存在
する、自分のワークステーションなどの場所
(ノード)にあるかを認識せずに、システム内の
情報資源を自由に活用することができる。なぜ
このようなことが可能になるか、それは、各社コン
ピュータの遠近地帯が一貫してシングルアー
キテクチャを出現しているからである。

ユーザニーズに柔軟に対応 MMLの階層化

VNETとMMLの階層化についてはMMLの
階層化についても考えている。
MMLの目的は、ホストコンピュータワークス
テーションのソフトウェア環境により、ユーザニ
ーズに柔軟に対応し、情報資源の共有を容
易に実現することである。しかし、MMLはホスト
コンピュータワークステーションと接続した場
合には、ホストコンピュータに多大な負荷がか
かることになる。ゆえに、ホストコンピュータワー

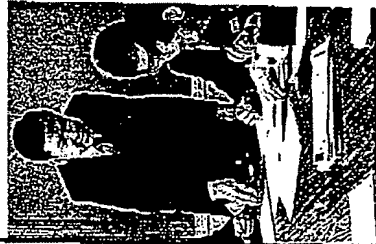
クステーションと結ぶ、中間ノードが必要と
され、MMLが階層化される。
中間ノードとしてV-7000シリーズが存在する
場合、どのようなMMLが構築されることになる
のか。当社のMMLは、複数の他社環境が
存在するコンピュータシステムにのみ対応するこ
と、いわゆるマルチコンピュータ・リンクが明
瞭である。しかし、通信エミュレータの場合と
は異なり、複数の他社MMLをそのまゝマテ
提供するのは、ワークステーションの
上では、当社のO/A機能(MCALC, OACGRAPH
など)が統一ソフトウェア・ベースとして稼動し、
それが他社コンピュータのデータベースにア
クセスして、MMLを実現するものである。ゆえ
に、他社コンピュータで稼動するMML用の
ソフトウェアについても、順次当社が提供して
ゆく予定である。

要するに、シングルアーキテクチャであるV-
7000シリーズの世界は、VNETが自由自在
な情報交換を可能にし、これら他社コンピ
ュータのMMLによって、ユーザ本位の高度な
情報ネットワークシステムを構築しているのである。



図V-7000のシステム構成
主機はV-7000/4000/6000/8000/10000/12000/14000/16000/18000/20000/22000/24000/26000/28000/30000/32000/34000/36000/38000/40000/42000/44000/46000/48000/50000/52000/54000/56000/58000/60000/62000/64000/66000/68000/70000/72000/74000/76000/78000/80000/82000/84000/86000/88000/90000/92000/94000/96000/98000/100000

V-7000 シリーズ



51
4-47
5
4

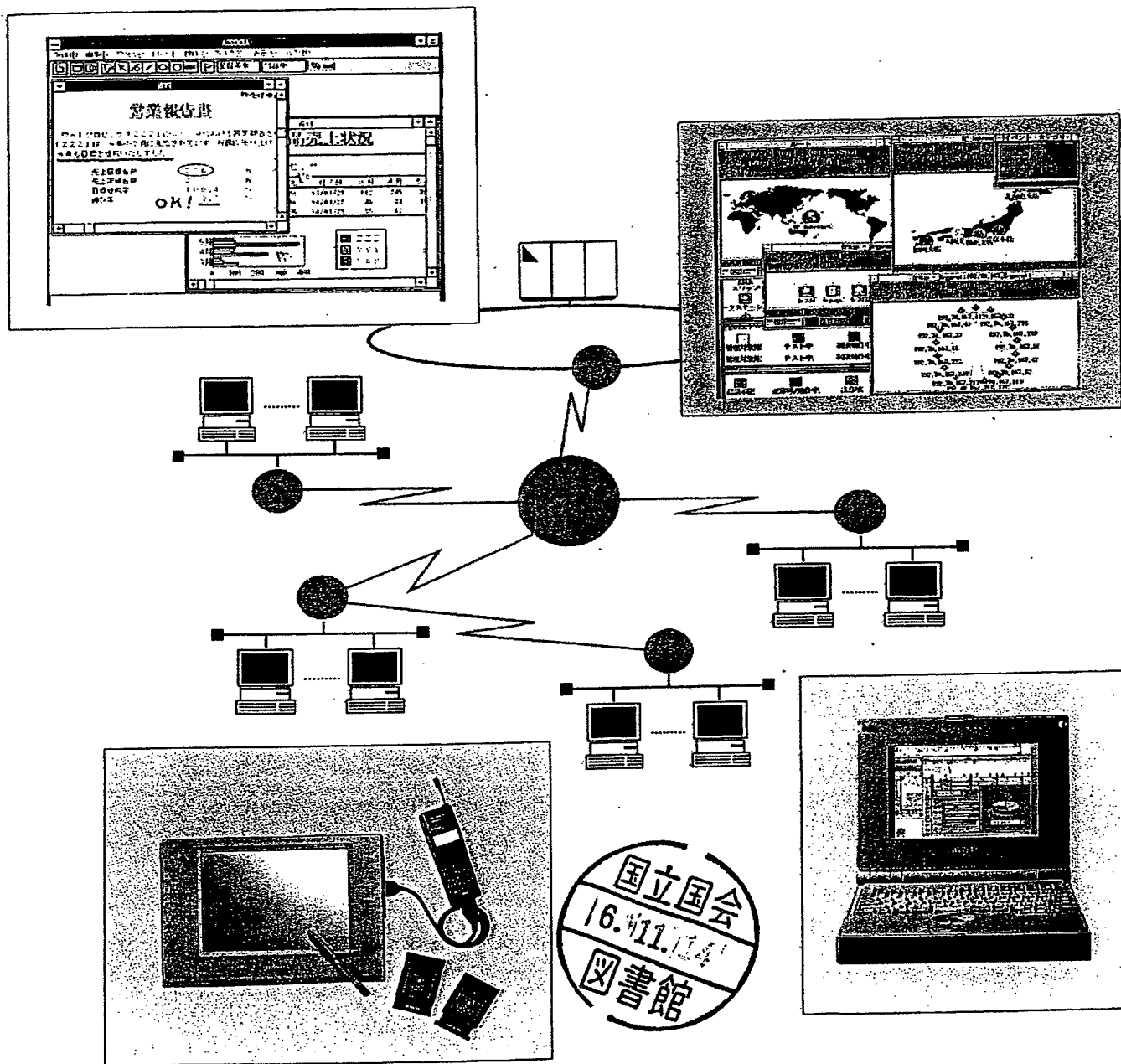
日立評論

THE HITACHI HYORON

1994 11 Vol.76

11

特集 業務革新のための情報通信ネットワークシステム



BEST AVAILABLE COPY

HITACHI

BEST AVAILABLE COPY

Petitioner Apple Inc. - Exhibit 1004, p. 369

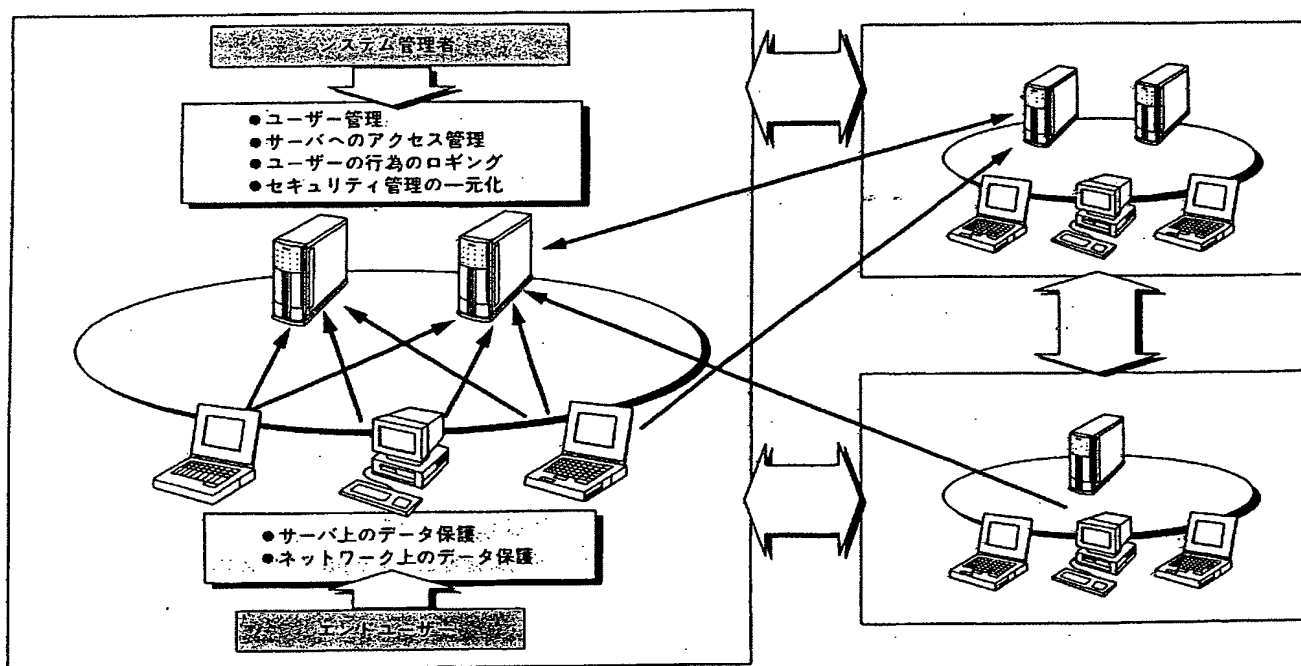
オープン分散環境における 企業情報システムのセキュリティ

Information Technology System Security for Open Distributed Computing Environment

久芳 靖* Yusushi Kuba

斉藤洋子* Yoko Saito

洲崎誠一** Seitchi Susaki



オープン分散環境のセキュリティ管理

オープン分散環境では、ネットワークを意識したエンドユーザーの視点に立ったセキュリティ対策が求められる。そのため、日立製作所はパソコン向けの暗号ライブラリを開発し、情報セキュリティを考慮した分散セキュリティシステムの実現を図っていく。

コンピュータシステムとネットワークのオープン化は、エンドユーザーを主体とした企業情報システム構築を促すとともに、ネットワークを介した利用者相互間での情報共有や伝達の可能性を広げている。

情報は現代の企業活動にとって、人・物・金に続く第四の財産とも言われており、企業情報システムでは必要不可欠なものである。そのため、エンドユーザー主体のシステムでは使いやすさを追求する一方で、そこで取り扱われる情報を保護するということが重要である。

日立製作所は、オープン環境をベースとしたエンドユーザーシステムが発展していくためにセキュリティを一つの重要な基盤であると考え、エンドユーザーの使い勝手を損なわずにセキュリティを確保するための技術開発と製品化を推進している。その第一歩として、共有ファイルやネットワーク上のデータ保護に広く適用可能な、パソコン(パーソナルコンピュータ)向けの暗号ライブラリを他社に先駆けて開発し、さらにシームレスでかつ安全な、オープンシステムの利用環境の実現を図っていく。

* 日立製作所 ソフトウェア開発本部

** 日立製作所 システム開発研究所

1 はじめに

企業の基幹業務を担っているホスト集中型システムは、システムの堅牢(ろう)さと集中化されたシステム運用により、セキュリティが十分に確保されてきた。しかし、オープンシステム技術をベースとし、エンドユーザーが主体となる分散コンピューティング環境では、企業活動に必要なセキュリティの確保という課題を残したまま、そのシステム規模や利用形態が発展してきている。例えばLAN上を伝達されるデータは、そのネットワークプロトコルがオープンであるために、簡単な機器さえあれば傍受することが可能であるし、現状の多くのシステムでは他人のファイルを見ることが、勝手にシステムの稼動を妨害することさえ容易である(図1参照)。

日立製作所は、分散コンピューティング環境が社会の重要な一基盤として健全に発展するためには、情報を必要に応じて適切に保護できなければならないと考えている。このため、分散コンピューティング環境の利点を生かしながら、情報セキュリティをも考慮した分散セキュリティシステムの実現を図っていく。ここでは、分散セキュリティシステム実現の考え方と、その第一歩として提案するパソコン向け暗号ライブラリについて述べる。

2 分散コンピューティング環境でのセキュリティ

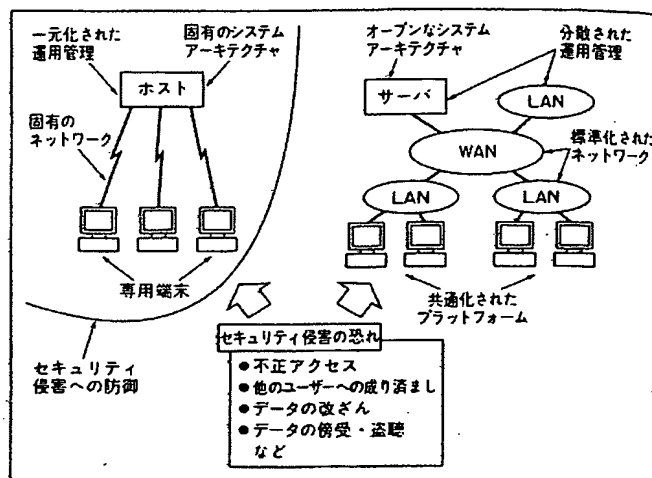
2.1 分散セキュリティの実現

ホスト集中型システムでのセキュリティ管理は、主としてユーザーによるシステムへのアクセスを管理するためのユーザー識別、認証およびユーザーの行為の追跡、確認(監査)が中心であった。

分散コンピューティング環境では、これに加えネットワークを意識したデータの機密保護、および分散環境全体の一元的な管理が必要であり、エンドユーザーを意識した使い勝手も重要である。

このような技術としては、OSF^{※1)}/DCE^{※2)}のセキュリティ機能があり、現在あるセキュリティ管理ソフトウェアとしては最も完成度が高いと言われている。

従来のセキュリティ機能は、システム独自のプロトコルなど固有の方式で実現されていたため、オープンな分散環境に対応することは困難であった。そこでセキュリティ機能をオープンな分散環境で実現させるため、世界



注：略語説明 WAN (Wide Area Network)

図1 オープン環境におけるセキュリティ上の脅威
オープン環境では標準化、運用の分散などにより、セキュリティ上安全とは言えない。

最大のユーザー組織であるX/Open^{※3)}を中心にシステム間で共通に利用できる汎(はん)用API(Application Programming Interface)〔GSS-API(Generic Security Service-API)〕の規定が推進されており、OSF/DCEでも採用される見通しである。

2.2 分散セキュリティシステム実現の考え方

分散コンピューティング環境の情報セキュリティを次のような考え方で実現していく。

- (1) 複数のプラットフォーム相互間で一元的に活用可能なセキュリティ管理の実現
- (2) エンドユーザーデータの保護
- (3) 使い勝手の良さの確保

上記(1)の実現にあたっては、個々のプラットフォームのセキュリティ機能を活用し、その上のアプリケーションにGSS-APIのセキュリティ機能を利用させる。また、異種プラットフォームにまたがる分散環境のセキュリティ管理の一元化を目標として、OSF/DCEを基盤技術として適用する。さらには、情報のより高い機密性を確保するために、日立製作所独自の技術であるMULTI(Multimedia Encryption)暗号アルゴリズムを提供し、各アプリケーションから利用できるようにする。

この方針に基づいて、セキュリティ機能を実現するための機能群をモデル化したものを図2に示す。各シス

※1) OSFは、Open Software Foundationの商標である。

※2) OSF/DCEは、OSFの商品名称である。

※3) X/Openは、X/Open Company Limitedの商標である。

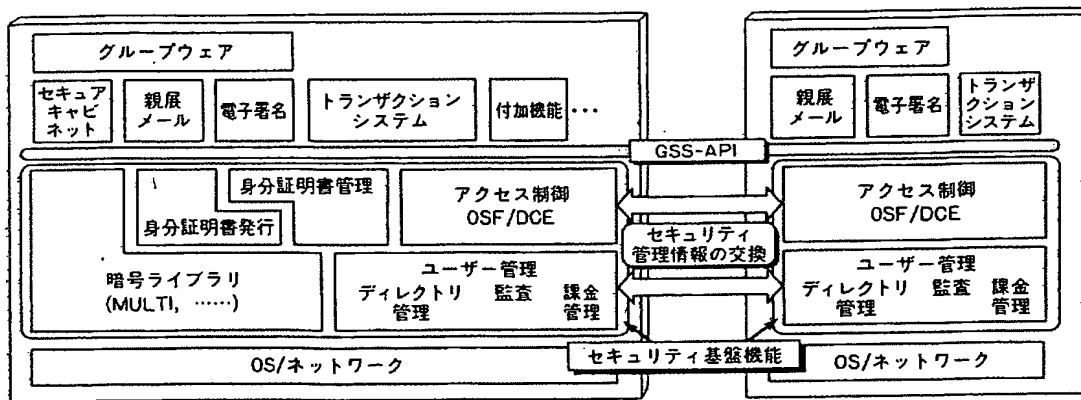


図2 分散環境に対応したセキュリティ製品提供の考え方

システム間相互のセキュリティ管理は、OSF/DCEを利用して一元化する。

ムで共通かつ最低限必要となるセキュリティ基盤機能として、OSが備えているユーザー管理機能、およびユーザーのアクセス管理機能を提供する。さらに、データの機密保護を実現するための基盤として、暗号ライブラリを提供する。セキュリティ基盤機能を利用するアプリケーションには、メール、セキュアキャビネット、電子署名などを提供し、ユーザーニーズに合わせたシステム構築を可能としていく。

3 分散セキュリティ実現のための基盤技術

3.1 GSS-API

GSS-APIの特長は、データ保護などのセキュリティ機能を特定の環境、セキュリティメカニズム、および通信プロトコルに依存せずに、アプリケーションに提供できることにある。

二つの異なるシステム上のアプリケーションやユーザー間でセキュアな通信を行う場合、相手のアプリケーションやユーザーが互いに正しい相手であることを確認し合うこと、送信側が転送したデータを確かに相手に伝えることが必要である。

例えば、アプリケーションがGSS-APIを利用して互いに通信する場合、アプリケーションに代わってセキュリティ基盤機能がアプリケーションの身分を証明する身分証明書をセキュリティサーバに発行してもらい（図3の①参照）、それを互いに交換することで安全な通信路を確立することができる（同図②参照）。さらにデータ転送時、アプリケーションはデータを確実に送受信するために暗号化・復号化および署名の作成・確認という機能を利用できる（同図③参照）。

このようにGSS-APIを採用したシステムでエンドユーザーは、セキュリティのメカニズムを意識せずに、セキュリティ機能を利用することが可能である。

3.2 MULTI暗号

(1) MULTI暗号の概要

今回分散コンピューティング環境に適用するMULTI暗号は、日立製作所が1989年にMULTI2-Nという名称で発表した暗号アルゴリズムである。

MULTI暗号では、基本的なデータ扱(かく)乱方法の一つである換字処理、転置処理を計算機で複雑に繰り返すことにより、暗号強度を任意に高めることが可能である。

データの暗号化、暗号文の復号化時には暗号キーを使用するが、MULTI暗号は暗号化時と復号化時に同一の暗号キーを使用する対称暗号方式である。この方式は異なる暗号キーを使用する非対称暗号方式と比べて処理速度が速いという特長を持つ。また、MULTI暗号は、暗号アルゴリズムを公開した手続き公開型暗号でもある。仕様が公開されているため、オープンな環境で使われる製品に適している。

このような特長を持つ暗号アルゴリズムとしては、他に米国標準のDES(Data Encryption Standard)暗号、日本電信電話株式会社によるFEAL(Fast Data Encipherment Algorithm)暗号がある。

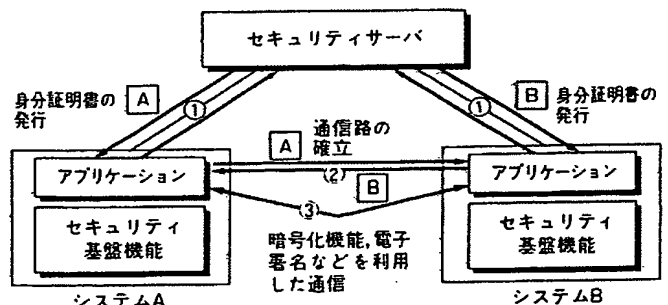


図3 GSS-APIを利用したアプリケーション間通信

セキュリティ基盤機能が提供するGSS-APIを利用することにより、アプリケーションに意識させずに安全な通信路を確保し、データ保護を図れる。

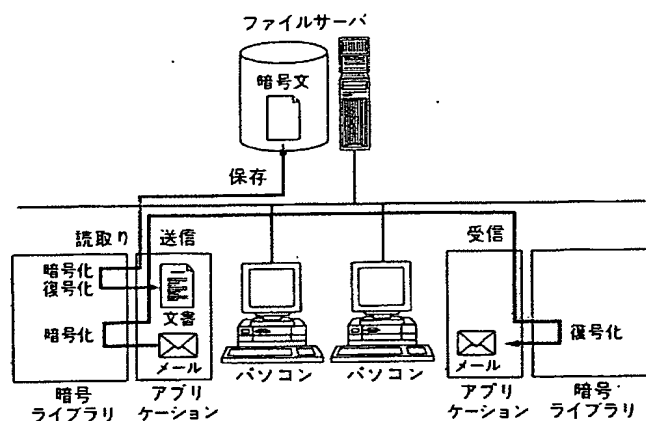


図4 暗号ライブラリを使用したセキュリティシステムの構築例
暗号ライブラリを用いて、ワープロ文書などを暗号化したうえでネットワーク上のファイルサーバに保管し、またメールの内容を暗号化して相手に送ることができる。

(2) 分散コンピューティング環境適用の優位性

MULTI暗号では換字・転置処理を32ビット単位で行っているため、最近主流となった32ビットパソコンでは、1ビット単位で処理しているDES暗号に比べ数十倍の性能が得られる。したがって、トラフィックの高いネットワーク上のデータに対しても暗号化を容易に行うことができる。

また、手続き公開型暗号であり、ソフトウェア処理でも高速であることから、オープン環境の各種のプラットフォーム上で共通に使用可能なソフトウェアとして提供できる。ハードウェアを必要としないことから、低コストでのシステム構築が容易である。

4 暗号ライブラリの開発

4.1 暗号ライブラリの概要

企業情報システムでは、重要情報を含む種々の情報処理をパソコンで行っており、パソコンでの適切な安全確保が緊急の課題となっている。このため、エンドユーザーの情報を保護するための基盤機能として、以下の特長を持つ暗号ライブラリを開発した。

(1) 身分証明書によるユーザー管理

個々のユーザーを識別するために、ユーザー名とパスワードを含んだ「身分証明書」を使用する。暗号ライブラリの使用開始時に、事前に発行した身分証明書を照合することで、不正利用を防止できる。

(2) 暗号による機密保護

ユーザーやアプリケーションが指定するデータを暗号化する。正しい身分証明書を持ったユーザーだけが、暗号文をもとのデータに復号化することができる。データの暗号化・復号化のための暗号技術としては、3章で述べたMULTI暗号を利用している。

(3) 標準APIの採用

アプリケーションから暗号ライブラリを利用するためのインタフェースとして、GSS-APIを使用している。標準化されたインタフェースであるため、アプリケーションの適用範囲が広がることが期待できる。

4.2 オープンシステムへの適用

オープンな環境で、暗号ライブラリを利用したセキュリティシステムの構築例を図4に示す。このような形態で情報の機密保護を実現する利点を次に述べる。

(1) 暗号処理部分をアプリケーションから切り離しているため、複数のアプリケーションから共通に利用でき、あるアプリケーションが生成したファイルを別のアプリケーションでも利用可能である。

(2) 特別なハードウェアを使用しないため、低コストであり、システム構築が容易である。

(3) 個人単位やグループ単位で暗号キーを持つことにより、必要なユーザー単位で情報の共有・専有を使い分けることが可能である。

5 おわりに

ここでは、分散セキュリティシステムの実現の考え方と暗号ライブラリについて述べた。分散コンピューティング環境の発展に必要なセキュリティ基盤の確立と、セキュリティ技術を積極的に活用した新しい企業情報システムへの展開を二つの軸として、今後とも企業を支える情報システムのセキュリティに取り組んでいく。

参考文献

- 1) 宝木、外：マルチメディア向け高速暗号アルゴリズム Hisecurity-Multi 2 の開発と利用方法、1989年情報理論とその応用、暗号と情報セキュリティジョイントワークショップ資料、電子情報通信学会、167～173(平1-8)
- 2) 小山、外：現代暗号理論、電子通信学会発行、41～62

(昭61-9)

- 3) S.Miyaguchi, et al.: Expansion of FEAL Cipher, NTT Review, Vol. 2, No. 6 (1990)
- 4) 柴宮、外：セキュリティ管理の技術、日本科学技術連盟 (1993)

11月号特集取りまとめ 松香茂道

企画委員

委員長	三木義照
委員	大町一彦
〃	菱沼孝
〃	守田康
〃	渡邊健
〃	茂地二
〃	武田道
〃	新井治
〃	久保忠
〃	本川世
〃	花田夫
〃	前明
〃	高橋武
〃	川船成
〃	藤安辰
〃	小林正
幹事	遠藤洋

日立評論 Vol.76 No. 12 次号予告

最近の送変電技術の動向

日立評論 第76巻第11号

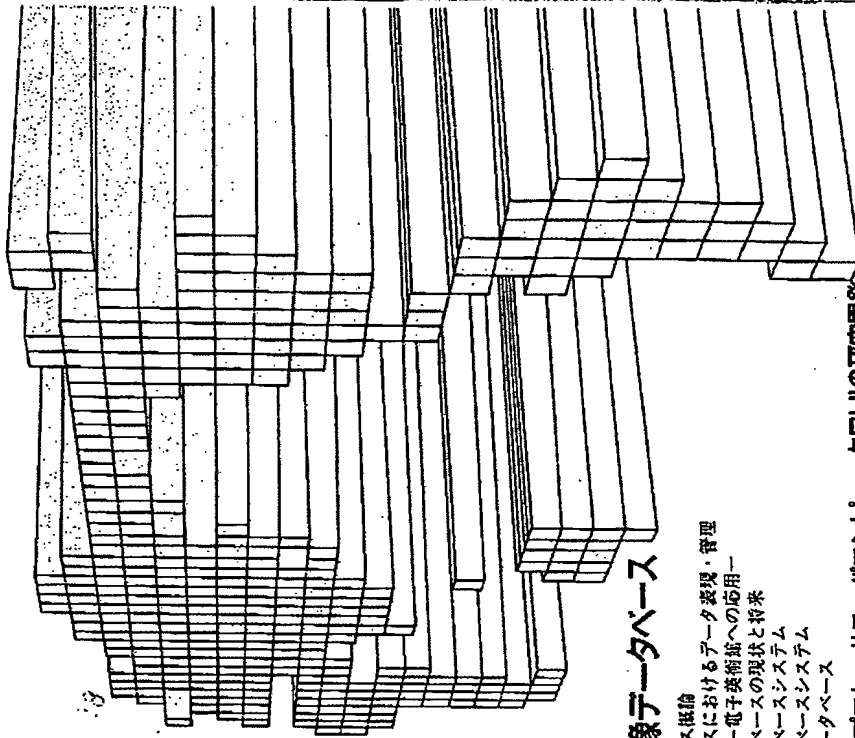
発行日	平成6年11月1日発行
発行所	日立評論社 東京都千代田区神田駿河台四丁目6番地 ㊞101-10 電話(03)3258-1111(大代)
編集兼発行人	小林正三郎
印刷所	日立印刷株式会社
定価	1部730円(本体709円)送料別 年間購読料 9,500円(送料含む)
取次店	株式会社オーム社 東京都千代田区神田錦町三丁目1番 ㊞101 電話(03)3233-0641(代) 振替口座 東京6-20018

214-357

第445号 3月30日 第三任編集長
平成4年5月15日発行 (4月13日印刷)
ISSN 0447-8053

情報処理

C4A-25 5
1992



特集 画像データベース

画像データベース概論
画像データベースにおけるデータ表現・管理
画像の内容検索—電子芸術館への応用—
医用画像データベースの現状と将来
地図情報データベースシステム
文書画像データベースシステム
金属材料画像データベース

特集 マルチプロセッサ—パソコンCPUの研究開発

スーパーコンピュータプロジェクトの成果概要
マルチプロセッサ—パソコンCPUのためのジョセフソン素子技術開発

BEST AVAILABLE COPY

6.5 道路管理システムにおけるシステム構成
道路管理センター各支部においては、スーパーミニコンピュータが設置され、おのおの地区単位に道路・占用物件のデータベースが構築されている。さらにグラフィック端末、デジタイザ、ハードコピー装置、静電プロッタ、プリンタなどの周辺装置より構成されている。

国、市、区の道路管理者においてはグラフィック端末、ハードコピーを設置し、この支部のスーパーミニコンピュータとオンライン接続し、データベースの検索業務を行っている。

各公益事業者は自社のコンピュータと WAN (Wide Area Network) で結合し、検索、設計をオンラインで行っている事業者と MT 交換によりデータを授受し、オフラインで行っている事業者とわかれていく。

図-11 に道路管理システムのシステム構成例を示す。

7. 今後の展望

東京をはじめとして、多くの大都市においては都市機能を維持するためのユーティリティが年々増大し、これらの維持管理が次第に困難になりつつある。地理情報データベースシステムは、これらの入り組んだ都市ユーティリティ情報を分岐・整理し、管理していくためのもので、都市機能を円滑に運営していくためには、必要不可欠のものである。

地図データベースのインフラストラクチャ構築には多額の金と時間が必要であるが、現在これらのデータベース構築がガス、上水、下水の各事業体において実現されつつある。

参 考 文 献

- 1) Shinoaki, S.: Computer Mapping in Tokyo Gas, AM/FM Conference (1982).
- 2) Shinoaki, S.: Total Utility Mapping System, NCGA (1988).
- 3) 日本経済新聞社: コンピュータマッピング入門.
- 4) Tada, Shinoaki, Hoge (他): National Computer Mapping System, IFIP (1991).
- 5) 計測と制図, 9月号 (1991).
- 6) (財)道路管理センター: 道路管理センター情報, 9月号.
- 7) 建設省: 都市情報データベース.
- 8) 財団法人計測技術研究所: 地図情報処理, (平成3年9月24日受付)



東明佐久良 (正会員)
昭和20年生。昭和42年東京大学工学部計測工学科卒業。同年東京ガス(株)に入社。昭和52年よりコンピュータマッピングシステムの開発に従事。昭和68年より69年にかけて、東京ガスにおけるマッピングシステムを完成。昭和61年より図版プロジェクトである、道路管理システムの開発を行っている。現在、マッピングセンター所属、AM/FM インターナショナル日本の幹事役員。

解 説

画像データベース



文書画像データベースシステム†

飯 沢 篤 志†

とワーワードやタイトル、作成者、日付などの付加情報による方法が提供される。しかし、文書を画像データとして扱うだけでなく、文書作成システムとの融合や、WP のファイル管理との運動など、次第に MMDB としての文書 DB が要求されてきている。

最近ローカルエリアネットワーク (LAN) が普及しつつあり、パーソナルコンピュータ (PC) やワークステーション (WS) が LAN に接続され、LAN を通じて他のマシンのファイルをアクセスすることができるようになった。この分散環境では、文書も分散して管理し、効率よく共有化や再活用することが望まれる。

本稿では、上記の背景を踏まえ、オフィスにおける文書の特徴を考慮し、それを管理、管理するデータベースに要求される機能やその実現上必要とされる関連技術について解説する。最後に文書データベースのシステム例を紹介する。

2. オフィス文書の特徴とデータベース機能

オフィス文書の特徴と、紙を媒体とする問題点、DB を利用する長所、データベース管理システム (DBMS) にとつての問題などを説明する。

2.1 マルチメディア

オフィス文書は、テキスト、図形、画像、グラフなどのマルチメディアデータ (MMD) により構成され、また、手書き、WP、机上出版 (DTP)、FAX など多様な手段により生成、獲得される。DB を構築する上では、MMD に対する効率のよ

1. はじめに
オフィスでは、社内外への連絡用文書、報告書、管理用文書、会議議事録、伝票、説明資料、パンフレット・カタログ、手紙など多種多様な文書が活用され、文書が重要な位置を占めている。また、複写機やファクシミリ (FAX)、ワープロ (WP) という OA 機器が普及するにつれてますますオフィスで生成される文書の量が増えている。文書を整理するために、キャビネットなどが利用されてきた。しかし、検索手段が、タイトル、分類、所有者程度のものしか提供されないため、既存の文書を検索するのに手間がかかり、情報の共有化や再利用が難しい。そのため重複文書や不要文書が多くなり、スペースを浪費してしまふ。そこで、文書を電子媒体に置き換え、データベース (DB) で管理し、共有化と省スペース化を図り、オフィスの作業効率を高める動きがある。

文書 DB を構築する上で画像との関係を考える。と次のような二つの立場がある。
・文書を画像として管理する
・文書の構成要素として画像が含まれる
前者の立場には、先ファインディングシステムの例がある。後者は、文書をテキスト、図、写真などの構成要素に分解し管理するマルチメディア DB (MMDB) として管理しようという立場である。先ファインディングでは、文書をすべて画像として入力し、現実のキャビネットに類似した内部構造

オフィス文芸には、カタログなど社外に公開し
て構わないものから、社外秘のもの、特定部署の
人のみアクセス可能なもの、特定ランク以上の人

フィス文書の特長から、扱う画像はモノクロ
データが多いが、今後カラーの入出力機器が普及
格納装置の容量も大きくなるとオフィス文書
カラー画像を利用するようになることは必至
る。

カラー画像を蓄積・伝送するには、圧縮技術が重要な課題である。カラーの自然画像について JPEG (Joint Photographic Expert Group) で国際標準として符号化方式が、早ければ 91 年末に完成すると見られており、圧縮率としては、情報量を 20 分の 1 以下に圧縮できるが、情報量により元のイメージは完全に復元できない。

に入力する必要がある。そのような場合、画像を認識しコードデータに変換することで、入力処理の自動化や省力化、データ容量の減少、検索機能の強化を図ることができる。ここでは、オフィス文書にとって重要な OCR (Optical Character Reader) と書式認識の技術を紹介する。

OCR は、画像データからテキストデータへ変換する技術である¹¹⁾。現在、居字の日本語と英数字が実用化されている。手書きの英数字も実用化段階に達しているが、使用条件が限定されていることが多い。手書き日本語は今後の研究課題である。

大量の文書を画像で入力し、OCR を使って全文 DB を作成する試みもある¹²⁾。その場合問題となるのは、OCR の認識率である。文字の形だけでは判別が難しい場合 (たとえば、「一」や「ー」) もあるので、認識された文字列を自然言語処理を用いて修正する必要がある。また、1 種類のアルゴリズムによる OCR では、認識率に限りがある。そこで、複数の OCR を同時に実行し多数決をとることで、誤りを訂正し認識率を上げるという提案もある¹³⁾。しかし、認識率を 100% にすることは難しいので、実用的には OCR で得られた結果を効率よく修正するエディタを組み合わせた必要がある。

オフィスで使われる文書の中には、伝票や履歴用紙などの定型書式がある。その書式に記入されたデータを DB に自動入力するためには、入力された画像データがどの書式かを認識する必要がある。書式が認識できれば、必要なデータの欄が分り、OCR を使ってその欄のデータを自動的に抽出することができる。書式を認識する方法としては、以下の方法が考えられる。

- (1) 特定の位置の文字列を OCR で認識する
 - (2) 特定の位置のバーコードを認識する
 - (3) 書式全体の特徴により認識する
- 使う書式を新しく作成してよいのなら、(1)、(2) の方式が有力な候補である。既存の書式を認識する場合には、(3) のように書式の特徴を抽出

るなど、文献検索の対象となっていない文献とは異なる性質をもつ。しかし、付加情報を利用した検索を考えたときには、文献検索の分野の成果は大いに参考になる。

文書に対して、キーワードを見つけて作業は、主題の把握、概念の把握、キーワードの付与というステップを踏む¹⁴⁾。文書の本文からキーワードを自動的に付与するという研究も多い¹⁵⁾。しかし、文書中にまったく現れない語をキーワードとすべき場合もあり¹⁶⁾、完全な自動化は今後の課題である。

キーワードを一定の語彙から選択するという方法もある¹⁷⁾。その語彙をソーラスと呼ぶ。ソーラスは、語の同義関係、階層関係、およびその他の関係を示したものであり、検索において標準的な語彙を与える。

ただし、一般オフィスでは、文書のキーワードを付ける際に、ソーラスで定める語を使うのは難しい。文献検索においては、文献の登録は専門家が行うが、オフィス文書を登録するのは専門家でないためである。そこで、オフィス文書 DB を構築するときには、キーワードは自由につけるようにし、それらのキーワードからソーラスを構築していくという運用を考えるべきである。そうすることによりソーラス自体も共通の DB として運用・管理し、日常業務での更新の対象となる。

4. システム例

文書を扱うデータベースのシステム例を紹介する。ここで紹介するシステムは、オフィス文書に限ったものではないが、一般文書データベースの事例を見ることが、今後、オフィス文書にどのような運用できるかを考える基礎にできると考える。

4.1 ホスト WS 連携のシステム

安田らは、ホスト計算機で文書管理をし、それに接続された多数の WS で文書作成、編集を行うというシステムを実現している¹⁸⁾。ホストに

を行うシステムを実現した。

文書の性質と文書管理の形態によって以下のよう三つの運用モデルにパターン化している。

構造化文書モデル：最大文書を複数人で共同作成するモデルである。全体を部分構造に分割して別に管理し、最後にそれらを結合する。

体系的文書モデル：管理者が社内規格にあった文書群の体系を決定し、その文書群を作成していくモデルである。

流動的文書群モデル：多数の文書群が作成、蓄積された後、実行構造的な文書体系を作成していくモデルである。

この運用モデルをもとに長大文書、大量文書の文書管理に必要な共通機能として次のような機能を出している。なお、ここでいう「クラス」とは論理的分類のことである。

- (1) 単一文書の操作
 - 文書の登録、取だし、置換、削除、拙写、移動
- (2) 格納構造の操作
 - 格納構造の作成、変更、削除、拙写、移動
- (3) クラス単位の操作
 - クラス単位の登録、取だし、置換、更新、削除
- (4) 検索
 - クラス単位の拙写、移動、分類、結合
 - クラス単位の文書管理情報の集計
- (5) 版数管理
 - 版数の更新
 - 版数指定の文書取だし、置換
- (6) 文書資源の保全
 - 文書資源の追跡、復元
 - 文書資源の権限保護
 - 文書資源の異常時のリカバリ

文書データベースは、論理構造、物理構造、

である。物理構造は、物理的資源と格納情報に対応付けている。格納構造と物理構造は従来の文書ファイルで提供される機能である。

文書処理を行うアプリケーションは WS 上で稼働し、対象となる文書構成要素をホストで検索し、取り出して処理を行う。文書の修正は WS に取り出して行うので、修正中の文書構成要素は「更新状態」となるように管理される。従来の DBMS が提供するトランザクションだけでは、文書修正のような長時間にわたるトランザクションを実現することはできない。このシステムでは、文書の階層構造のノードの状態を変更することで実現している。

4.2 CAMMS

マニュアル作成・管理においては、改訂作業が多く、新しい版では 70~80% が再利用できるので、部分文書の共有化が重要な課題となる。CAMMS¹⁹⁾ は、マニュアル原稿の構成部品を管理し、ひとつのマニュアルに必要な原稿を組版システムに渡すという機能をもっている。CAMMS への原稿入力には、SGML²⁰⁾ の考え方を取り入れた独自のマークアップ言語を使う。また、DB 内の原稿をマージして新しい原稿を作成する機能もある。

CAMMS で扱う論理構造は、ジョブ、区分、層別、実体という 4 階層の構造をもつ (図-1)。ジョブはひとつのマニュアル全体に対応し、区分の列

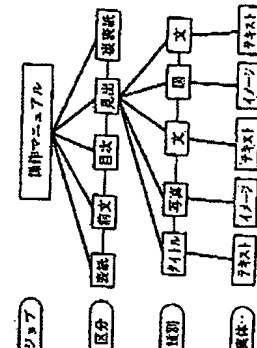


図-1 インスタンスオブジョブ

がマニュアルの論理的構成を表す。実体には、テキスト、イメージ、CAD図面などのデータが対応する。区分と種別内側のオブジェクトには、幾何学順序関係をもたせている。また、区分と種別にたいは、共有ノードを実現するたに、他のノードを間接的に指すよう、原相オブジェクトを導入している。

階層構造などの構造を表現するに
は、それを表現するのに適したDBMS
を利用することが望ましい。CAMMS
は、リンク指向 DBMS G-BASE¹⁰⁾ 上
に実現されている。G-BASE は、リ
レーショナルモデルを拡張したグラフ
データベース¹¹⁾に基づいて開発された

4.3 MASCOT

- Language (SGML) (1989).
- 29) 平岡：リンク指向 DBMS G-BASE におけるリ
ンク機能の拡張とその応用、情報処理学会研究報告
91-DBS-82, 82-5 (1991).
- 24) Kuniti, H. L.: Graph Data Model and Its Data
Language, Springer-Verlag Tokyo (1990).
- 25) Chen, P. P. S.: The Entity-Relationship Model
— toward a unified view of data, ACM Trans.
Database Syst., Vol. 1, No. 1, pp. 9-35 (1976).
- 26) Kuwano, N. et al.: Applications of Object-
Oriented Databases to Publishing Systems, Proc.
of DASFAA, pp. 421-429 (1991).

(平成 3 年 10 月 8 日受付)



飯沢 真志 (正会員)
1957 年生。1980 年東京大学理学
部情報科学科卒業。1982 年同大学院
理学系研究科情報科学専門課程修士
課程修了。現在、(株)リコーソフト
ウェア事業部ソフトウェア研究開発部、データベース管
理システム、イメージデータベース構築ツールの研究・
開発に従事。日本ソフトウェア科学会会員。



解説 画像データベース



金属材料画像データベース†

稲田 清崇†† 小林 克之††

1. 背景

材料の分野では、評価のために計算機利用が進んでいる。すなわち、応力・歪の解析などの力学解析、材料の内部構造の応力・歪・変形解析や接合・接合部解析などの力学特性評価、亀裂の進展・成長評価やシミュレーションなどの強度評価、さらには材料・構造の設計などにおいて、計算機利用の機会が増えているのが現状である¹⁾。

計算機の有効利用のためには、多くの材料の各種のパラメータ (材料特性データ) と材料規格とを正確に使う必要があり、各機関でデータベース化が行われている。材料学会の金属材料強度データベース²⁾、JICST/NRIM 金属材料強度データベース³⁾、動力燃料事業団の SMAT などがあるが、いずれも数値データが主であり、画像データまでは扱っていないのが現状である。

しかし一方、材料評価の方法にも画像処理技術が利用され始めており、従来の人間の視覚による分類・評価を、画像処理により定量化する方向が急速に進んできている。力学的な変形・形状の計測や、欠陥・亀裂などの位置・形状の測定や状態の表示、シールド面やマクロ破面の解析、電子顕微鏡の像や腐食面の解析、材料組織の状態分析・結晶性の解析などに画像処理技術を用いた定量評価が行われはじめた。

また、材料の状態の保存という観点からも、従来の写真での保存から、画像入力装置の高精度化と記憶媒体のコストダウンおよび信頼性の向上から、電子的媒体で保存する機会が増えてきて

こうした背景により、材料の分野においても、数値データと画像データを扱う画像データベースへの要求が強くなってきている。

材料分野における画像データベースは、使用目的により大きく 2 種類に分れる。

一つは、材料設計・評価のための研究開発用データベースであり、もう一つは、品質保証・製造管理のためのデータベースである。

設計評価用のデータベースとしては、プラント材料の設計を目的としたファクト型のデータベース⁴⁾や、高分子材料の設計を目的としたグラフィックデータベース⁵⁾、高分子材料の設計を目的としたグラフィックデータベース⁶⁾が発表されている。高分子画像データベースは、記録媒体として通記型光ディスクを用い、パソコンのリレーショナルデータベースソフトを利用し、788 件のデータが登録されている。また、ガラスの材料設計支援システムの一部として、ガラス化範囲を示す約 300 件の状態図が登録されたものもある⁷⁾。また、金属材料の画像処理評価の支援システムとして、金属材料の定量的評価の支援を行う画像処理手順を画像データから検索するシステムも開発されている⁸⁾。このように、設計評価のための画像データベースにおいても、また数値データ主体から脱して、小規模な試作段階のものが多い。

品質保証・製造管理のための画像データベースは、大量の画像データを製造条件・品質検査結果などに対応付けて長期に保存し、必要時には最早く報告が作成提出できることが必要である。このため、画像の大量ファイルリング、遠隔地での

BEST AVAILABLE COPY

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1482	705/51.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/19 13:54
L2	1835	usage adj right	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/19 13:54
L3	1031356	render or rendering or rendered	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/19 13:54
L4	763367	software	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/19 13:54
L5	1767809	(owner or controller or director or overseer or supervisor)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/19 13:54
L6	12127836	(condition or action or happening or position or status)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/19 13:54
L7	84081	(digital adj rights adj management) or DRM or license or licensing	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/19 13:54
L8	393	L7 and L2 and L3 and L4 and L5 and L6	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/19 13:54

EAST Search History

L9	118	L8 and l1	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/19 13:54
S1	1	11/304794	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/17 17:01
S2	53	"08344760"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/17 17:01
S3	39	"08/344760"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/17 17:26
S5	10267	(digital adj rights adj management) or DRM	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/17 18:27
S6	1834	usage adj right	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/17 17:33
S7	6945	(usage or use or utilize or usability) adj5 right	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/17 17:48
S8	1030727	render or rendering or rendered	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/17 17:51

EAST Search History

S9	762029	software	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/17 17:56
S10	1766377	(owner or controller or director or overseer or supervisor)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/17 18:02
S11	12123580	(condition or action or happening or position or status)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/17 18:25
S12	250	S5 and S6 and S8 and S9 and S10 and S11	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/17 18:31
S14	83972	(digital adj rights adj management) or DRM or license or licensing	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/17 18:27
S15	316	S5 and S7 and S8 and S9 and S10 and S11	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/17 18:31
S16	392	S14 and S6 and S8 and S9 and S10 and S11	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	NEAR	ON	2007/04/19 13:54



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/304,794	12/16/2005	Mark J. Stefik	111325-028700	6096
22204	7590	05/01/2007		
NIXON PEABODY, LLP 401 9TH STREET, NW SUITE 900 WASHINGTON, DC 20004-2128			EXAMINER LUDWIG, PETER L	
			ART UNIT 3621	PAPER NUMBER
			MAIL DATE 05/01/2007	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

11/304,794

Applicant(s)

STEFIK ET AL.

Examiner

Peter L. Ludwig

Art Unit

3621

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 25 August 2006.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-37 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-37 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 13 March 2006 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)	4) ☐ Interview Summary (PTO-413)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date. _____
3) ☒ Information Disclosure Statement(s) (PTO/SB/08)	5) ☐ Notice of Informal Patent Application
Paper No(s)/Mail Date <u>8/25/2006, 12/16/2005</u> .	6) ☐ Other: _____

DETAILED ACTION

Acknowledgements

1. This Office Action has been given Paper No. 20070417 for reference purposes only.
2. This Office Action is in correspondence to the Information Disclosure Statement filed with the USPTO on 08/25/2006.
3. Claims 1-37 are currently pending.

Continuations

4. This application is a continuation application of U.S. application no. 11/198,216 filed on August 8, 2005, now U.S. Application no. 10/176,608 filed on June 24, 2002 now U.S. Patent no. 6,934,693, which is a CON of 09/777,845 filed on February, 7, 2001, which is a DIV of 08/08/967,084 filed on November 10, 1997 now Patent no. 6,236,971 which is a CON of 08/344,760 filed on November 23, 1994 "Parent Application"). See MPEP §201.07. In accordance with MPEP §609.02 A. 2 and MPEP §2001.06(b) (last paragraph), the Examiner has reviewed and considered the prior art cited in the Parent Application. Also in accordance with MPEP §2001.06(b) (last paragraph), all documents cited or considered 'of record' in the Parent Application are now considered cited or 'of record' in this application. Additionally, Applicant(s) are reminded that a listing of the information cited or 'of record' in the Parent Application need not be resubmitted in this application unless Applicants desire the information to be printed on a patent issuing from this application. See MPEP §609.02 A. 2. Finally, Applicants are reminded that the prosecution history of the Parent Application is relevant in this application.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claims 3-6, 9-11, 12-13, 15-18, 21-25, 27-29, and 33-37 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

a. Claims 3-6, 9-11, 15-18, 21-23, 27-29, and 33-35 recites the limitation "the content" or "the digital content". There is insufficient antecedent basis for this limitation in the claim. The Examiner is unsure what content the Applicants are speaking of and the Examiner asks a proper amendment is put into place to correct such rejection.

b. Claims 12-13, 24-25, and 36-37 are rejected under 35 U.S.C. 112, second paragraph due to them being vague and indefinite. The Examiner is unclear which statutory class the Applicants are intending to claim. With the claim directed at a method and then the Applicants are trying to claim structural components, such as, in claim 13 a computer readable medium configured to cause one or more computer processors to perform the steps of the method. The USPTO deems these types of claims "hybrid" claims and deems them non-statutory under 35 U.S.C. 112, second paragraph and 35 U.S.C. 101 if the Applicants are intending to claim structural components within the method claims. The Examiner will be viewing the claims as processes without structural elements, but the Applicants must amend if Applicants are trying to claim structural elements within the method.

Double Patenting

7. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

8. Claims 1-37 are provisionally rejected on the ground of nonstatutory double patenting over claims 15-98 of copending Application No. 11/198,216. This is a provisional double patenting rejection since the conflicting claims have not yet been patented.

The subject matter claimed in the instant application is fully disclosed in the referenced copending application and would be covered by any patent granted on that copending application since the referenced copending application and the instant application are claiming common subject matter, as follows: Both applications teach a method of rendering a document.

Furthermore, there is no apparent reason why applicant would be prevented from presenting claims corresponding to those of the instant application in the other copending application. See *In re Schneller*, 397 F.2d 350, 158 USPQ 210 (CCPA 1968). See also MPEP § 804.

Claim Rejections - 35 USC § 102

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

10. Claims 1-37 are rejected under 35 U.S.C. 102(b) as being anticipated by Perritt JR – Knowbots, Permissions Headers and Contract Law (Reference U on attached PTO – 892 form)[hereinafter Perritt JR].

11. As per claim 1, Perritt JR clearly discloses a method for securely rendering digital documents, comprising:

- **storing a digital document in a document platform** (page 3 – “Basic Concepts”);
- **storing at least one usage right associated with the digital document, the at least one usage right specifying a manner of use indicating the manner in which the digital document can be rendered by the document platform** (page 1 – “Introduction”);
- **determining whether the digital document may be rendered by the document platform based on at least the at least one usage right** (page 1 – “If the request message and the permissions header match, the user would obtain access to the work.”);
- **if the at least one usage right allows the digital document to be rendered on the document platform, rendering the digital document by the document platform** (page 1 – i.e. “If the request message and the permissions header match, the user would obtain access to the work.”).

12. As per claim 2, Perritt JR clearly discloses the method of claim 1 as described above.

Perritt JR further teaches **wherein the manner of use includes the number of times the digital document can be rendered ()**.

13. As per claim 3, Perritt JR clearly discloses the method as recited in claim 1 as described above. Perritt JR further teaches **wherein at least one part of the content and at least one usage right are included in different files** (page 4 – i.e. “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.”).

14. As per claim 4, Perritt JR clearly discloses the method as recited in claim 1 as described above. Perritt JR further discloses **wherein at least one part of the content and at least one usage right are stored in the same file** (page 1 – i.e. “a work would have attached to it a permissions header”).

15. As per claim 5, Perritt JR clearly discloses the method as recited in claim 1 as described above. Perritt JR further discloses **wherein at least a portion of the digital content is a software program** (page 1 – “Introduction”; Examiner is interpreting the content in the digital library as software programs).

16. As per claim 6, Perritt JR clearly discloses the method as recited in claim 1 as described above. Perritt JR further teaches **wherein at least one usage right comprises a revenue**

Art Unit: 3621

identifier for identifying a revenue owner of the digital content (page 4 - i.e. "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." Examiner is interpreting that if the user must pay a certain fee and the fee is included in the permissions header, then the user must know who to pay the money to, which in turn leads to knowing who the "revenue owner" of the content is).

17. As per claim 7, Perritt JR clearly discloses the method as recited in claim 1 as described above. Perritt JR further teaches **wherein at least one usage right also specifies one or more conditions which must be satisfied before the manner of rendering may be exercised** (page 8 "Legal Issues" – "If an outright transfer is intended, then the present copyright statute requires a writing signed by the owner of the rights conveyed.").

18. As per claim 8, Perritt JR clearly discloses the method as recited in claim 7 as described above. Perritt JR further teaches **wherein at least one condition includes determining the presence of a digital ticket** (According to the explanation from Applicant's Specification "In the present invention, a "digital ticket" is used to enable the ticket holder to exercise usage rights specifying the requirement of the digital ticket." The Examiner is interpreting the "permission header" as a digital ticket due to it enabling the ticket holder to exercise usage rights specifying

the requirement of the digital ticket (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 (page 1 – "Introduction"), i.e. 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 (page 16)).

19. As per claim 9, Perritt JR clearly discloses the method as recited in claim 1 as described above. Perritt JR further teaches **wherein at least one of the usage rights or a part of the content are stored on a removable storage device** (page 22, #23-24 and 26).

20. As per claim 10, Perritt JR clearly discloses the method as recited in claim 1 as described above. Perritt JR further teaches **wherein at least one part of the content and at least one**

usage right are stored on the same device (page 1 – i.e. “a work would have attached to it a permissions header”).

21. As per claim 11, Perritt JR clearly discloses the method as recited in claim 1 as described above. Perritt JR further teaches **wherein at least one of the content and at least one usage right are stored on different devices** (page 6 – “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.””).

22. As per claim 12, Perritt JR clearly discloses the method as recited in claim 1 as described above. Perritt JR further teaches **wherein the method is implemented with a computer system including one or more hardware and/or software components** (page 3 – “Basic Concepts”).

23. As per claim 13, Perritt JR clearly discloses the method as recited in claim 1 as described above. Perritt JR further teaches **wherein the method is implemented via one or more computer readable instructions embedded on a computer readable medium and configured to cause one or more computer processors to perform the steps of the method** (page 3 – “Basic Concepts”).

24. As per claim 14, Perritt JR clearly discloses a method for securely rendering digital documents, comprising:

- **storing a digital document in a document repository** (page 3 – “Basic Concepts”);

- **storing at least one usage right associated with the digital document (page 1 – “Introduction”);**
- **requesting access to the digital document from a document platform (page 1 – “Introduction”);**
- **determining whether the request may be granted based on the usage right, the determining step including authenticating the document platform and determining whether the usage right includes a manner of use that allows transfer of the digital document to the document platform (page 1 – “Introduction”);**
- **if the usage right allows the transfer of the digital document to the document platform, transferring at the digital document to the document platform (page 1 – “Introduction”); and**
- **rendering the digital document by the document platform (page 1 – “Introduction”).**

25. As per claim 15, Perritt JR clearly discloses the method as recited in claim 14 as described above. Perritt JR further teaches **wherein at least one part of the content and at least one usage right are included in different files** (page 6, “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.””).

26. As per claim 16, Perritt JR clearly discloses the method as recited in claim 14 as described above. Perritt JR further teaches **wherein at least one part of the content and at least one usage right are stored in the same file** (page 1 – “Introduction”).

27. As per claim 17, Perritt JR clearly discloses the method as recited in claim 14 as described above. Perritt JR further teaches **wherein at least a portion of the digital content is a software program** (page 1 – “Introduction”; Examiner is interpreting the content in the digital library as software programs).

28. As per claim 18, Perritt JR clearly discloses the method as recited in claim 14 as described above. Perritt JR further teaches **wherein at least one usage right comprises a revenue identifier for identifying a revenue owner of the digital content** (page 4 - i.e. “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.” Examiner is interpreting that if the user must pay a certain fee and the fee is included in the permissions header, then the user must know who to pay the money to, which in turn leads to knowing who the “revenue owner” of the content is).

29. As per claim 19, Perritt JR clearly discloses the method as recited in claim 14 as described above. Perritt JR further teaches **wherein at least one usage right also specifies one or more conditions which must be satisfied before the manner of rendering may be exercised** (page 8 “Legal Issues” – “If an outright transfer is intended, then the present copyright statute requires a writing signed by the owner of the rights conveyed.”).

30. As per claim 20, Perritt JR clearly discloses the method as recited in claim 19 as described above. Perritt JR further teaches **wherein at least one condition includes determining the presence of a digital ticket** (According to the explanation from Applicant’s Specification “In the present invention, a “digital ticket” is used to enable the ticket holder to exercise usage rights specifying the requirement of the digital ticket.” The Examiner is interpreting the “permission header” as a digital ticket due to it enabling the ticket holder to exercise usage rights specifying the requirement of the digital ticket (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 (page 1 – “Introduction”), i.e. 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

Art Unit: 3621

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 (page 16)).

31. As per claim 21, Perritt JR clearly discloses the method as recited in claim 14 as described above. Perritt JR further teaches **wherein at least one of the usage rights or a part of the content are stored on a removable storage device** (page 22, #23-24 and 26).

32. As per claim 22, Perritt JR clearly discloses the method as recited in claim 14 as described above. Perritt JR further teaches **wherein at least one part of the content and at least one usage right are stored on the same device** (page 1 – “Introduction”).

33. As per claim 23, Perritt JR clearly discloses the method as recited in claim 14 as described above. Perritt JR further teaches **wherein at least one of the content and at least one usage right are stored on different devices** (page 6, “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.””).

Art Unit: 3621

34. As per claim 24, Perritt JR clearly discloses the method as recited in claim 14 as described above. Perritt JR further teaches **wherein the method is implemented with a computer system including one or more hardware and/or software components** (page 3 – “Basic Concepts”).

35. As per claim 25, Perritt JR clearly discloses the method as recited in claim 14 as described above. Perritt JR further teaches **wherein the method is implemented via one or more computer readable instructions embedded on a computer readable medium and configured to cause one or more computer processors to perform the steps of the method** (page 3 – “Basic Concepts”).

36. As per claim 26, Perritt JR clearly discloses a method for securely rendering digital documents, comprising:

- **receiving a request to transfer a digital document from a document repository to a document platform the digital document having at least one usage right associated therewith** (page 1 – “Introduction”);
- **authenticating the document platform by accessing at least one identifier associated with the document platform or with a user of the document platform and determining whether the identifier is associated with at least one of a document platform and user authorized to use the digital document** (page 7 – “Role of Encryption”);

- **if the authenticating step is successful, determining whether the digital document may be transferred and stored on a document platform based on a manner of use included in the at least one usage right (page 7 – “Role of Encryption”);**
- **if the at least one usage right allows the digital document to be transferred to a document platform, transferring the digital document to the document platform (page 7 – “Role of Encryption”); and**
- **rendering the digital document by the document platform (page 7 – “Role of Encryption”).**

37. As per claim 27, Perritt JR clearly discloses the method as recited in claim 26 as described above. Perritt further teaches **wherein at least one part of the content and at least one usage right are included in different files** (page 6, “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.””).

38. As per claim 28, Perritt JR clearly discloses the method as recited in claim 26 as described above. Perritt JR further teaches **wherein at least one part of the content and at least one usage right are stored in the same file** (page 1 – “Introduction”).

39. As per claim 29, Perritt JR clearly discloses the method as recited in claim 26 as described above. Perritt JR further teaches **wherein at least a portion of the digital content is**

a software program (page 1 – “Introduction”; Examiner is interpreting the content in the digital library as software programs).

40. As per claim 30, Perritt JR clearly discloses the method as recited in claim 26 as described above. Perritt JR further teaches **wherein at least one usage right comprises a revenue identifier for identifying a revenue owner of the digital content** (page 4 - i.e. “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.” Examiner is interpreting that if the user must pay a certain fee and the fee is included in the permissions header, then the user must know who to pay the money to, which in turn leads to knowing who the “revenue owner” of the content is).

41. As per claim 31, Perritt JR clearly discloses the method as recited in claim 26 as described above. Perritt JR further teaches **at least one usage right also specifies one or more conditions which must be satisfied before the manner of rendering may be exercised** (page 8 “Legal Issues” – “If an outright transfer is intended, then the present copyright statute requires a writing signed by the owner of the rights conveyed.”).

Art Unit: 3621

42. As per claim 32, Perritt JR clearly discloses the method as recited in claim 31 as described above. Perritt JR further teaches **wherein at least one condition includes determining the presence of a digital ticket** (According to the explanation from Applicant's Specification "In the present invention, a "digital ticket" is used to enable the ticket holder to exercise usage rights specifying the requirement of the digital ticket." The Examiner is interpreting the "permission header" as a digital ticket due to it enabling the ticket holder to exercise usage rights specifying the requirement of the digital ticket (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 (page 1 – "Introduction"), i.e. 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 (page 16)).

43. As per claim 33, Perritt JR clearly discloses the method as recited in claim 26 as described above. Perritt JR further teaches **wherein at least one of the usage rights or a part of the content are stored on a removable storage device** (page 22, #23-24 and 26).

44. As per claim 34, Perritt JR clearly discloses the method as recited in claim 26 as described above. Perritt JR further teaches **wherein at least one part of the content and at least one usage right are stored on the same device** (page 1 – “Introduction”).

45. As per claim 35, Perritt JR clearly discloses the method as recited in claim 26 as described above. Perritt JR further teaches **wherein at least one of the content and at least one usage right are stored on different devices** (page 6, “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.””).

46. As per claim 36, Perritt JR clearly discloses the method as recited in claim 26 as described above. Perritt JR further teaches **wherein the method is implemented with a computer system including one or more hardware and/or software components** (page 3 – “Basic Concepts”).

Art Unit: 3621

47. As per claim 37, Perritt JR clearly discloses the method as recited in claim 26 as described above. Perritt JR further teaches **wherein the method is implemented via one or more computer readable instructions embedded on a computer readable medium and configured to cause one or more computer processors to perform the steps of the method** (page 3 – “Basic Concepts”).

Claim Interpretations

48. Applicants are reminded that functional recitations using the word “for,” “configured to,” or other functional terms (*e.g. see* claim 6 which recites “for identifying a revenue owner of the digital content”) have been considered but are given little patentable weight¹ because they fail to add any structural limitations and are thereby regarded as intended use language. To be especially clear, all limitations have been considered. However a recitation of the intended use in a product claim must result in a structural difference between the claimed product and the prior art in order to patentably distinguish the claimed product from the prior art. If the prior art structure is capable of performing the intended use, then it reads on the claimed limitation. *In re Casey*, 370 F.2d 576, 152 USPQ 235 (CCPA 1967) (“The manner or method in which such machine is to be utilized is not germane to the issue of patentability of the machine itself.”); *In re Otto*, 136 USPQ 458, 459 (CCPA 1963). See also MPEP §§ 2114 and 2115. Unless expressly noted otherwise by the Examiner, the claim interpretation principles in this paragraph apply to all examined claims currently pending.

49. As per claims 1, 2, 7, 14 and 26, the USPTO interprets claim limitations that contain “*if, may, might, can, when and could*” statement(s), as optional language. As matter of linguistic precision, optional claim elements do not narrow claim limitations, since they can always be omitted (*In re Johnston*, 77 USPQ2d 1788 (CA FC 2006)). Language that suggests or makes optional but does not require steps to be performed or does not limit a claim to a particular structure does not limit the scope of a claim or claim limitation. For the sake of speeding up the

¹ See *e.g. In re Gulack*, 703 F.2d 1381, 217 USPQ 401, 404 (Fed. Cir. 1983)(stating that although all limitations must be considered, not all limitations are entitled to patentable weight.).

Art Unit: 3621

prosecution process, the Examiner has taken these limitations into account, but will not be given patentable weight unless positively recited.

Examiner Note

50. Examiner has cited particular columns and line numbers in the references as applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings in the art and are applied to the specific limitations within the individual claim, other passages and figures may be applied as well. It is respectfully requested from the applicant, in preparing responses, to fully consider the reference in its entirety as potentially teaching all of part of the claimed invention as well as the context of the passage as taught by the prior art or disclosed by the examiner.

Art Unit: 3621

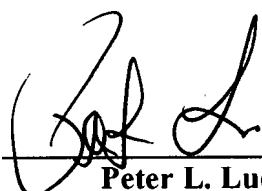
Conclusion

51. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Peter L. Ludwig whose telephone number is 571-270-1365. The examiner can normally be reached on Mon-Fri 7:30-5:00, 1st Fri. Off, 2nd Fri. 7:30-4:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Andrew Fischer can be reached on 571-272-6779. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.


ANDREW J. FISCHER
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 3600

 4/25/2007
Peter L. Ludwig
Patent Examiner
Art Unit 3621

Search Notes 	Application/Control No. 11304794	Applicant(s)/Patent Under Reexamination STEFIK ET AL.
	Examiner Ludwig, Peter L	Art Unit 3621

SEARCHED			
Class	Subclass	Date	Examiner
705	51	04/19/2007	Peter L. Ludwig

SEARCH NOTES		
Search Notes	Date	Examiner
see EAST search history	04/19/2007	Peter L. Ludwig
PALM inventor search	04/19/2007	Peter L. Ludwig

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
 United States Patent and Trademark Office
 Address: COMMISSIONER FOR PATENTS
 P.O. Box 1450
 Alexandria, Virginia 22313-1450
 www.uspto.gov



Bib Data Sheet

CONFIRMATION NO. 6096

SERIAL NUMBER 11/304,794	FILING OR 371(c) DATE 12/16/2005 RULE	CLASS 705	GROUP ART UNIT 3621	ATTORNEY DOCKET NO. 111325-028700

APPLICANTS

Mark J. Stefik, Portola Valley, CA;
 Peter L. T. Pirolli, San Francisco, CA;

** CONTINUING DATA *****

This application is a CON of 11/198,216 08/08/2005 which is a CON of 10/176,608 06/24/2002 PAT 6,934,693
 which is a CON of 09/777,845 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

OK - PL

** FOREIGN APPLICATIONS *****

NONE - PL

IF REQUIRED, FOREIGN FILING LICENSE GRANTED

** 01/25/2006

Foreign Priority claimed ☐ yes ☒ no	STATE OR COUNTRY CA	SHEETS DRAWING 13	TOTAL CLAIMS 37	INDEPENDENT CLAIMS 3	
35 USC 119 (a-d) conditions ☐ yes ☒ no ☐ Met after met Allowance					
Verified and Acknowledged	Examiner's Signature <i>[Signature]</i>	Initials <i>[Initials]</i>			

ADDRESS

22204

TITLE

System for controlling the distribution and use of digital works

FILING FEE RECEIVED 1850	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

Knowbots, Permissions Headers & Contract Law - Perritt
page 1

Knowbots, Permissions Headers and Contract Law

paper for the conference on

Technological Strategies for Protecting Intellectual
Property in the Networked Multimedia Environment

April 2-3, 1993 with revisions of 4/30/93

Copyright 1993

Henry H. Perritt, Jr.

Professor of Law

Villanova Law School

Villanova, PA 19085

(215) 645-7078

FAX (215) 645-7033, (215) 896-1723

Internet: perritt@ucis.vill.edu

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 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

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 gaining 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 was 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 of remote retrieval of complete information objects using electronic technologies is in wide use through the WESTLAW, Dialog, LEXIS, NEXIS, and National Library of Medicine databases. These remotely accessible databases, however, unlike the digital library involved 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.39.6. 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 permit automatic refinement of searches according to statistical matching techniques.

The Corporation for National Research Initiatives 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 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 RRS¹¹ - 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 on only 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 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. Mr. Linn's "envelope" 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.

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 signature. 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 CDROMs 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.

While encrypted content affords greater security to the owner of copyrighted material. Someone who has not paid the price to the copyright owner must incur 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 market place militate against embracing encryption for content. Encryption interferes with realization of electronic markets, because producer and consumer must have the same encryption and description protocols. Encryption burdens processing of electronic information objects because it adds another layer. Some specific implementations have 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 and also, by impairing open architectures may perpetuate some of papers' advantages with respect with 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 trade secret. Thus, there may be serious questions about the legal efficacy of use restrictions suggested in ____, 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 depend, not 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 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 more 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 and "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 affect of their conduct because the writing in the signature involve 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 and 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 now²⁵
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 channel²⁶
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 to 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 that all participants are bound by. 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 transferor and 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 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 examine. 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 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 used⁴¹

The key in adapting the business records exception to electronic permissions in a digital library system are points 3 and 4. Establishing these propositions and the propositions set forth in section ___ of 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 parties archiving system is properly designed, it can provide evidence sufficient to establish all of the propositions identified in ___.

This third party intermediary concept is somewhat different from the concept for 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 of 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 "Standard" 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 Pool 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 as Mr. Linn suggests.

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 user⁴⁵ 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 concept, 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 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, 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 CDROM.

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, down

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.

Finally, is the purchase of a relatively small consumer item, say a box of diskettes. Neither the law or 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 [make sure these function and the citation appears earlier]. The electronic publishing analogy might be use permission for part of a work.

Cost effectiveness = risk-proportional security

traditional transaction	institutions	electronic equivalent
real estate closing	registrar of deeds	entire library - software contents
will/power of attorney	witnesses, notary public	contents of entire CDROM
auto purchase	UCC financing statement	complete work - transfer copyright
television set purchase	written sale agreement	complete work - use perm
box of diskettes	-	part of a work - use permission

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 notary 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 102(e) and (f) in the High Performance Computing Act looks very much like King James I's licensing of printing presses. It also looks 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.

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 at 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.

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 an 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. (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).

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 require 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, as ____ of this paper notes, 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 concepts 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 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. 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, Computer Data and Reliability: A Call for Authentication of Business Records Under the Federal Rules of Evidence, 80 Nw.U.L.Rev. 956, 979 (1986) (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 alternations 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 ____ (1981).

32 Cite for when extrinsic evidence is admissible.

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

justiciable 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, Computer Data and Reliability: A Call for Authentication of Business Records Under the Federal Rules of Evidence, 80 Nw.U.L.Rev. 956, 978-80, 984-85 (1986) (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, Computer Data and Reliability: A Call for Authentication of Business Records Under the Federal Rules of Evidence, 80 Nw.U.L.Rev. 956, 963-64 (1986) (identifying steps and trend resulting in F.R.E.).

40 Peritz, Computer Data and Reliability: A Call for Authentication of Business Records Under the Federal Rules of Evidence, 80 Nw.U.L.Rev. 956, 963 (1986).

41 Peritz, Computer Data and Reliability: A Call for Authentication of Business Records Under the Federal Rules of Evidence, 80 Nw.U.L.Rev. 956, 974 (1986) (reporting four requirements of Manual, and endorsing their use generally).

42 See Henry H. Perritt, Jr., ___, ___ Jurimetrics ___ (1993) (distinguishing between format and content standardization).

43 See Marc Lauritsen, ___ (explaining relationship between substantive legal systems and the field of artificial intelligence).

44 See Thorne, McCarty; Kevin Ashley; and Gardner.

45 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.

Index of Claims 	Application/Control No. 11304794	Applicant(s)/Patent Under Reexamination STEFIK ET AL.
	Examiner Ludwig, Peter L	Art Unit 3621

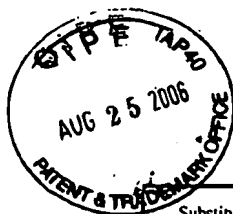
✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant			☐ CPA			☐ T.D.			☐ R.1.47		
CLAIM		DATE									
Final	Original	04/19/2007									
	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	✓									

<i>Index of Claims</i> 	Application/Control No. 11304794	Applicant(s)/Patent Under Reexamination STEFIK ET AL.
	Examiner Ludwig, Peter L	Art Unit 3621

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant		☐ CPA		☐ T.D.		☐ R.1.47	
CLAIM		DATE					
Final	Original	04/19/2007					
	37	✓					



Substitute for form 1449A/PTO				Complete if Known			
INFORMATION DISCLOSURE STATEMENT BY APPLICANT (use as many sheets as necessary)				Application Number	11/304,794		
				Filing Date	December 16, 2005		
				First Named Inventor	Mark J. STEFIK, et al.		
				Art Unit	3621		
				Examiner Name	~~Not Yet Assigned~~ Peter Ludwig/		
Sheet		of		Attorney Docket Number	111325-028700		
U.S. PATENT DOCUMENTS							
Examiner Initials*	Cite No. ¹	U.S. Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	
		Number - Kind Code ² (if known)					
		US-					
		US-					
FOREIGN PATENT DOCUMENTS							
Examiner Initials*	Cite No. ¹	Foreign Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T ³
		Country Code ²	Number ⁴ Kind Code ² (if known)				
/P.L./ ↓		JP	06-214862	08/05/1994	SATOMI et al.	English Abstract	
		JP	05-298174	11/12/1993	NUKUI	English Abstract	
		JP	06-230847	08/19/1994	MASUI	English Abstract	
		JP	03-014109	01/22/1991	NARITA	English Abstract	
		JP	06-318167	11/15/1994	TAKAHASI	English Abstract	
		JP	06-103286	04/15/1994	HARTRICK et al.	English Abstract	
OTHER PRIOR ART - NON PATENT LITERATURE DOCUMENTS							
Examiner Initials*	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 ³
/P.L./		Henry M. Levy, "Capability-Based Computer Systems", Digital Press, 1984, P1-18 USA					
↓		Kawahara Masaharu, "Consideration of a Method of Charging for Electronic Objects in Superdistribution", Technical Report of the Institute of Electronics, Information and Communication Engineers, EIC Institute of Electronics, Information and Communication Engineers, September 21, 1994, Shingaku Giho Vol. 94 No. 240, P17-24 Japan - English Translation					
		Ryoichi Mori et al., "The Inevitable Way Toward Superdistribution", Information Symposium Papers, Information Processing Society of Japan, February 17, 1994, Vol. 94, No. 1, P67-76 Japan					
		Shinichi Ueki et al., "The Accounting Process in Software Usage Monitor for Superdistribution", Information Processing Society of Japan Technical Report, 90-Is-27, January 16, 1990, Vol. 90, No. 1, P1-10 Japan English Translation					
		Naoya Torii et al., "System Architecture of Superdistribution", Technical Report of the Institute of Electronics, Information and Communication Engineers, EIC Institute of Electronics, Information and Communication Engineers, September 21, 1994, Shingaku Giho Vol. 94 No. 240, P59-66 Japan					
		Yasushi Kuho et al., "Information Technology System Security for Open Distributed Computing Environment", The Hitachi Hyoron, November 1994, Vol. 76, P49-52 Japan					
		Hiroyoshi Takeda, "Function and Use of Workstation" The Office Management, Nikkan Kogyo Shinbun, August 1, 1989, Vol. 28, No. 8, P8-17 Japan					
↓		Atsushi Iizawa, "Document Image Database System", Information Processing Society, May 15, 1992, Vol. 33, No. 5, P497-504 Japan					
Examiner Signature	/Peter Ludwig/			Date Considered	04/19/2007		

*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 Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. 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 Applicant's unique citation designation number (optional). ² Applicant is to place a check mark here if English language Translation is attached.

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.

Substitute for form 1449A/PTO				Complete if Known			
INFORMATION DISCLOSURE STATEMENT BY APPLICANT (use as many sheets as necessary)				Application Number	Not Yet Assigned 11/304,794		
				Filing Date	December 16, 2005		
				First Named Inventor	Mark J. STEFIK, et al.		
				Art Unit	3621		
				Examiner Name	James A. Reagan/Peter Ludwig/		
Sheet	1	of	10	Attorney Docket Number	111325-028700		
U.S. PATENT DOCUMENTS							
Examiner Initials ¹	Cite No. ¹	U.S. Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	
		Number - Kind Code ² (if known)					
/P.L./		US-3,263,158		07/26/1966	D.W. Barga et al.		
		US-3,609,697		09/28/1971	Blevins et al.		
		US-3,790,700		02/05/1974	Callais et al.		
		US-3,798,605		03/19/1974	Feistel		
		US-4,159,468		06/26/1979	Barnes et al.		
		US-4,220,991		09/02/1980	Hamano et al.		
		US-4,278,837		07/14/1981	Best		
		US-4,323,921		04/06/1982	Guillou		
		US-4,442,486		04/10/1984	Mayer		
		US-4,529,870		07/16/1985	Chaum		
		US-4,558,176		12/10/1985	Arnold et al.		
		US-4,593,376		06/03/1986	Volk		
		US-4,614,861		09/30/1986	Pavlov et al.		
		US-4,644,493		02/17/1987	Chandra et al.		
		US-4,658,093		04/14/1987	Hellman		
		US-4,713,753		12/15/1987	Beobert et al.		
		US-4,796,220		01/03/1989	Wolfe		
		US-4,817,140		03/28/1989	Chandra et al.		
		US-4,827,508		05/02/1989	Shear		
		US-4,868,376		09/19/1989	Lessin et al.		
FOREIGN PATENT DOCUMENTS							
Examiner Initials ¹	Cite No. ¹	Foreign Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T ⁶
		Country Code ³ Number ⁴	Kind Code ⁵ (if known)				
/P.L./		0 084 441	EP	07/27/1983	TABS LIMITED		
		0 180 460	EP	05/07/1986	SONY CORPORATION		
		0 332 707	EP	09/20/1989	HONDA GIKEN KOGYO KABUSHIKI KAISHA		
		0 651 554	EP	05/03/1995	EASTMAN KODAK CO.		
		0 668 695	EP	08/23/1995	VICTOR COMPANY OF JAPAN LIMITED		
		0 725 376	EP	08/07/1996	SONY CORP.		
		2 136 175	GB	09/12/1984	ATALLA CORP.		
		2 236 604	GB	04/10/1991	SUN MICROSYSTEMS INC		
		0 715 241	JP	06/05/1996	MITSUBISHI CORP.		
		04-369068	JP	12/21/1992	CHIYUUBU NIHON DENKI SOFUTOEUA KK		Abst
Examiner Signature	/Peter Ludwig/			Date Considered	04/19/2007		

*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 Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at 222.uspto.gov or MPEP 901.04. 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.

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.

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.

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Complete if Known	
				Application Number	Not Yet Assigned 11/304,794
				Filing Date	December 16, 2005
				First Named Inventor	Mark J. STEFIK, et al.
				Art Unit	3621
				Examiner Name	James A. Reagan /Peter Ludwig/
Sheet	2	of	10	Attorney Docket Number	111325-028700

U.S. PATENT DOCUMENTS					
Examiner Initials*	Cite No. ¹	U.S. Patent Document	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Number - Kind Code ² (if known)			
		US-4,891,838	01/02/1990	Faber	
		US-4,924,378	05/08/1990	Hershey et al.	
		US-4,932,054	06/05/1990	Chou et al.	
		US-4,937,863	06/26/1990	Robert et al.	
		US-4,949,187	08/14/1990	Cohen	
		US-4,953,209	08/28/1990	Ryder, Sr. et al.	
		US-4,961,142	10/02/1990	Elliott et al.	
		US-4,975,647	12/04/1990	Downer et al.	
		US-4,977,594	12/11/1990	Shear	
		US-4,999,806	03/12/1991	Chernow et al.	
		US-5,010,571	04/23/1991	Katznelson	
		US-5,014,234	05/07/1991	Edwards, Jr.	
		US-5,023,907	06/11/1991	Johnson et al.	
		US-5,047,928	09/10/1991	Wiedemer	
		US-5,050,213	09/17/1991	Shear	
		US-5,052,040	09/24/1991	Preston et al.	
		US-5,058,164	10/15/1991	Elmer et al.	
		US-5,103,476	04/07/1992	Waite et al.	
		US-5,113,519	05/12/1992	Johnson et al.	
		US-5,136,643	08/04/1992	Fischer	

FOREIGN PATENT DOCUMENTS							
Examiner Initials ¹	Cite No. ¹	Foreign Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T ⁴
		Country Code ³	Number ³ Kind Code ² (if known)				
		05-268415	JP	10/15/1993	RICOH CO LTD		Abst
		06-175794	JP	06/24/1994	FUJI XEROX CO LTD		Abst
		06-215010	JP	08/05/1994	SONY CORP.		Abst
		07-084852	JP	03/31/1995	HITACHI LTD.		Abst
		07-200317	JP	08/04/1995	TOSHIBA CORP.		Abst
		07-244639	JP	09/19/1995	FUJITSU LTD		Abst
		62-241061	JP	10/21/1987	NEC CORP.		Abst
		64-068835	JP	03/14/1989	RYOICHI MORI		Abst
		WO 01/63528	PCT	08/30/2001	IPDN COPR.		
		WO 92/20022	PCT	11/12/1992	DIGITAL EQUIPMENT CORP.		

Examiner Signature	/Peter Ludwig/	Date Considered	04/19/2007

*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 Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at 222.uspto.gov or MPEP 901.04. 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.

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.

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.

Substitute for form 1449A/PTO		Complete if Known	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT (use as many sheets as necessary)		Application Number	Not Yet Assigned 11/304,794
		Filing Date	December 16, 2005
		First Named Inventor	Mark J. STEFIK, et al.
		Art Unit	3621
		Examiner Name	James A. Reagan /Peter Ludwig/
Sheet 3 of 10	Attorney Docket Number	111325-028700	

U.S. PATENT DOCUMENTS						
Examiner Initials ¹	Cite No. ²	U.S. Patent Document ³	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	
		Number - Kind Code ⁴ (if known)				
		US-5,138,712	08/11/1992	Corbin		
		US-5,146,499	09/08/1992	Geffrotin		
		US-5,148,481	09/15/1992	Abraham et al.		
		US-5,159,182	10/27/1992	Eisele		
		US-5,183,404	02/02/1993	Aldous et al.		
		US-5,191,193	03/02/1993	Le Roux		
		US-5,204,897	04/20/1993	Wyman		
		US-5,222,134	06/22/1993	Waite et al.		
		US-5,235,642	08/10/1993	Wobber et al.		
		US-5,247,575	09/21/1993	Sprague et al.		
		US-5,255,106	10/19/1993	Castro		
		US-5,260,999	11/09/1993	Wyman		
		US-5,263,157	11/16/1993	Janis		
		US-5,263,158	11/16/1993	Janis		
		US-5,276,444	01/04/1994	McNair		
		US-5,276,735	01/04/1994	Boebert et al.		
		US-5,291,596	03/01/1994	Mita		
		US-5,301,231	04/05/1994	Abraham et al.		
		US-5,311,591	05/10/1994	Fischer		
		US-5,319,705	06/07/1994	Halter et al.		
FOREIGN PATENT DOCUMENTS						
Examiner Initials ¹	Cite No. ²	Foreign Patent Document ³	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	7 ⁴
		Country Code ⁵ Number ⁶ Kind Code ⁷ (if known)				
		WO 93/01550 PCT	01/21/1993	INFOLOGIC SOFTWARE, INC		
		WO 94/01821 PCT	01/20/1994	SECURE COMPUTING CORP.		
		WO 96/24092 PCT	08/08/1996	BENSON, Greg		
		WO 97/48203 PCT	12/18/1997	INTEL CORP.		
		WO 98/11690 PCT	03/19/1998	GLOVER, John J.		
		WO 98/42098 PCT	09/24/1998	CRYPTOWORKS, INC.		
		WO 99/49615 PCT	09/30/1999	MICROTOME		
		EP 0 538 216 A1	04/21/1993	TELEVERKET		
Examiner Signature	/Peter Ludwig/			Date Considered	04/19/2007	

*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 Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at 222.uspto.gov or MPEP 901.04. 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.

Burdock 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.

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.

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT (use as many sheets as necessary)				Complete if Known	
				Application Number	Not Yet Assigned 11/304,794
				Filing Date	December 16, 2005
				First Named Inventor	Mark J. STEFIK, et al.
				Art Unit	3621
				Examiner Name	James A. Reagan /Peter Ludwig/
Sheet	4	of	10	Attorney Docket Number	111325-028700

U.S. PATENT DOCUMENTS					
Examiner Initials [*]	Cite No. ¹	U.S. Patent Document	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Number - Kind Code ² (if known)			
		US-5,337,357	08/09/1994	Chou et al.	
		US-5,339,091	08/16/1994	Yamazaki et al.	
		US-5,341,429	08/23/1994	Stringer et al.	
		US-5,347,579	09/13/1994	Blandford	
		US-5,381,526	01/10/1995	Ellson	
		US-5,394,469	02/28/1995	Nagel et al.	
		US-5,410,598	04/25/1995	Shear	
		US-5,412,717	05/02/1995	Fischer	
		US-5,428,606	06/27/1995	Moskowitz	
		US-5,432,849	07/11/1995	Johnson et al.	
		US-5,438,508	08/01/1995	Wyman	
		US-5,444,779	08/22/1995	Daniele	
		US-5,453,601	09/26/1995	Rosen	
		US-5,455,953	10/03/1995	Russell	
		US-5,457,746	10/10/1995	Dolphin	
		US-5,473,687	12/05/1995	Lipscomb et al.	
		US-5,473,692	12/05/1995	Davis	
		US-5,499,298	03/12/1996	Narasimhalu et al.	
		US-5,502,766	03/26/1996	Boebert et al.	
		US-5,504,814	04/02/1996	Miyahara	

FOREIGN PATENT DOCUMENTS						
Examiner Initials [*]	Cite No. ¹	Foreign Patent Document	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T ³
		Country Code ⁴ Number ⁵ Kind Code ² (if known)				

Examiner Signature	/Peter Ludwig/	Date Considered	04/19/2007

^{*}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 Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at 222.uspto.gov or MPEP 901.04. 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.

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.

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.

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT (use as many sheets as necessary)				Complete if Known	
				Application Number	Not Yet Assigned 11/304,794
				Filing Date	December 16, 2005
				First Named Inventor	Mark J. STEFIK, et al.
				Art Unit	3621
				Examiner Name	James A. Reagan /Peter Ludwig/
				Attorney Docket Number	111325-028700
Sheet	5	of	10		

U.S. PATENT DOCUMENTS						
Examiner Initials*	Cite No. ¹	U.S. Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Number - Kind Code ² (if known)				
		US-5,504,818		04/02/1996	Okano	
		US-5,504,837		04/02/1996	Griffeth et al.	
		US-5,509,070		04/16/1996	Schull	
		US-5,530,235		06/25/1996	Stefik et al.	
		US-5,532,920		07/02/1996	Hartrick et al.	
		US-5,534,975		07/09/1996	Stefik et al.	
		US-5,539,735		07/23/1996	Moskowitz	
		US-5,563,946		10/08/1996	Cooper et al.	
		US-5,568,552		10/22/1996	Davis	
		US-5,621,797		04/15/1997	Rosen	
		US-5,629,980		05/13/1997	Stefik et al.	
		US-5,633,932		05/27/1997	Davis et al.	
		US-5,634,012		05/27/1997	Stefik et al.	
		US-5,638,443		06/10/1997	Stefik et al.	
		US-5,649,013		07/15/1997	Stuckey et al.	
		US-5,655,077		08/05/1997	Jones et al.	
		US-5,708,717		01/13/1998	Alasia	
		US-5,734,823		03/31/1998	Saigh et al.	
		US-5,734,891		03/31/1998	Saigh	
		US-5,737,413		04/07/1998	Akiyama et al.	

FOREIGN PATENT DOCUMENTS						
Examiner Initials ¹	Cite No. ¹	Foreign Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Country Code ³	Number ⁴ Kind Code ⁵ (if known)			

Examiner Signature	/Peter Ludwig/	Date Considered	04/19/2007

*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 Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at 222.uspto.gov or MPEP 901.04. 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.

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.

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.

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Complete if Known	
				Application Number	Not Yet Assigned 11/304,794
				Filing Date	December 16, 2005
				First Named Inventor	Mark J. STEFIK, et al.
				Art Unit	3621
				Examiner Name	James A. Reagan /Peter Ludwig/
				Attorney Docket Number	111325-028700
Sheet	6	of	10		

U.S. PATENT DOCUMENTS					
Examiner Initials	Cite No.	U.S. Patent Document	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Number - Kind Code ¹ (if known)			
		US-5,737,416	04/07/1998	Cooper et al.	
		US-5,745,569	04/28/1998	Moskowitz et al.	
		US-5,748,783	05/05/1998	Rhoads	
		US-5,757,907	05/26/1998	Cooper et al.	
		US-5,761,686	06/02/1998	Bloomberg	
		US-5,765,152	06/09/1998	Erickson	
		US-5,768,426	06/16/1998	Rhoads	
		US-5,825,892	10/20/1998	Braudaway et al.	
		US-5,892,900	04/06/1999	Ginter et al.	
		US-5,910,987	06/08/1999	Ginter et al.	
		US-5,915,019	06/22/1999	Ginter et al.	
		US-5,917,912	06/29/1999	Ginter et al.	
		US-5,920,861	07/06/1999	Hall et al.	
		US-5,940,504	08/17/1999	Griswold	
		US-5,943,422	08/24/1999	Van Wie et al.	
		US-5,949,876	09/07/1999	Ginter et al.	
		US-5,982,891	11/09/1999	Ginter et al.	
		US-5,999,949	12/07/1999	Crandall	
		US-6,047,067	04/04/2000	Rosen	
		US-6,112,181	08/29/2000	Shear et al.	

[illegible]

Examiner Signature	/Peter Ludwig/	Date Considered	04/19/2007

*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 Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at 222.uspto.gov or MPEP 901.04. 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.

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.**

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.

Substitute for form 1449A/PTO				Complete if Known <table border="1"> <tr> <td>Application Number</td> <td>Not Yet Assigned 11/304,794</td> </tr> <tr> <td>Filing Date</td> <td>December 16, 2005</td> </tr> <tr> <td>First Named Inventor</td> <td>Mark J. STEFIK, et al.</td> </tr> <tr> <td>Group Art Unit</td> <td>3621</td> </tr> <tr> <td>Examiner Name</td> <td>James A. Reagan /Peter Ludwig/</td> </tr> <tr> <td>Attorney Docket Number</td> <td>111325-028700</td> </tr> </table>		Application Number	Not Yet Assigned 11/304,794	Filing Date	December 16, 2005	First Named Inventor	Mark J. STEFIK, et al.	Group Art Unit	3621	Examiner Name	James A. Reagan /Peter Ludwig/	Attorney Docket Number	111325-028700
Application Number	Not Yet Assigned 11/304,794																
Filing Date	December 16, 2005																
First Named Inventor	Mark J. STEFIK, et al.																
Group Art Unit	3621																
Examiner Name	James A. Reagan /Peter Ludwig/																
Attorney Docket Number	111325-028700																
INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>																	
Sheet	7	of	10														

U.S. PATENT DOCUMENTS

Examiner Initials*	Case No.†	U.S. Patent Document	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Number - Kind Code* (if known)			
		US-6,115,471	09/05/2000	Oki et al.	
		US-6,138,119	10/24/2000	Hall et al.	
		US-6,157,721	12/05/2000	Shear et al.	
		US-6,185,683	02/06/2001	Ginter et al.	
		US-6,226,618	05/01/2001	Downs et al.	
		US-6,233,684	05/15/2001	Stefik et al.	
		US-6,237,786	05/29/2001	Ginter et al.	
		US-6,240,185	05/29/2001	Van Wie et al.	
		US-6,253,193	06/26/2001	Ginter et al.	
		US-6,266,618	07/24//2001	Ye et al.	
		US-6,292,569	09/18/2001	Shear et al.	
		US-6,301,660	10/09/2001	Benson	
		US-6,327,652	12/04/2001	England et al.	
		US-6,330,670	12/11/2001	England et al.	
		US-6,345,256	02/05/2002	Milsted et al.	
		US-6,363,488	03/26/2002	Ginter et al.	
		US-6,389,402	05/14/2002	Ginter et al.	
		US-4,882,752	11/21/1989	Lindman et al.	
		US-5,224,163	06/29/1993	Gasser et al.	
		US-5,299,263	03/29/1994	Beller et al.	

FOREIGN PATENT DOCUMENTS

[illegible]

Examiner Signature	/Peter Ludwig/	Date Considered	04/19/2007

***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 Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at 222.uspto.gov or MPEP 901.04. 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.

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.**

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.

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Complete if Known	
				Application Number	Not Yet Assigned 11/304,794
				Filing Date	December 16, 2005
				First Named Inventor	Mark J. STEFIK, et al.
				Art Unit	3621
				Examiner Name	James A. Reagan /Peter Ludwig/
Sheet	8	of	10	Attorney Docket Number	111325-028700

OTHER PRIOR ART - NON PATENT LITERATURE DOCUMENTS				
Examiner Initials ²	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 ³	
		"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 the IEICE, Vo. E 73, No. 7, Tokyo, JP		
		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. 1-3, April 1994, Computer Shopper, Vol. 14, No. 4, ISSN 0886-0556		

Examiner Signature	/Peter Ludwig/	Date Considered	04/19/2007

* 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 Applicant's unique citation designation number (optional). 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.

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.

Substitute for form 1449A/PTO				Complete if Known	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT (use as many sheets as necessary)				Application Number	Not Yet Assigned 11/304,794
				Filing Date	December 16, 2005
				First Named Inventor	Mark J. STEFIK, et al.
				Art Unit	3621
				Examiner Name	James A. Reagan /Peter Ludwig/
				Attorney Docket Number	111325-028700
Sheet	9	of	10		

OTHER PRIOR ART - NON PATENT LITERATURE DOCUMENTS			
Examiner Initials*	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
		O'Conner, M., "New Distribution Option for Electronic Publishers; iOpener Data Encryption and Metering System for CD-ROM use; Column", pp. 1-6, March 1994, CD-ROM Professional, Vol. 7, No. 2, ISSN: 1409-0833	
		Willet, 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	
		Simmell, 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	

Examiner Signature	/Peter Ludwig/	Date Considered	04/19/2007

* 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 Applicant's unique citation designation number (optional). 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.

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.

Substitute for form 1449A/PTO				Complete if Known	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT (use as many sheets as necessary)				Application Number	Not Yet Assigned 11/304,794
				Filing Date	December 16, 2005
				First Named Inventor	Mark J. STEFIK, et al.
				Art Unit	3621
				Examiner Name	James A. Reagan / Peter Ludwig
				Attorney Docket Number	111325-028700
Sheet	10	of	10		

OTHER PRIOR ART - NON PATENT LITERATURE DOCUMENTS			
Examiner Initials ²	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 ³
		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", PP. 1-24, 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	
		Henry H. Perritt, Jr., "Technological Strategies for Protecting Intellectual Property in the Networked Multimedia Environment", April 2-3, 1993, Knowbots, Permissions Headers & Contract Law	
		Perritt, Henry H.; Knowbots, Permissions Headers and Contract Law paper for conference on <i>Technological Strategies for Protecting Intellectual Property in the Networked Multimedia Environment</i> , April 2-3, 1993, pages 1-22	
		European Search Report dated September 8, 2004 in corresponding European Application No. EP 03 015 128.6	
		European Search Report dated September 13, 2004 in corresponding European Application No. EP 03 015 127.8	
		European Search Report for Corresponding European Application 95308422.5	

Examiner Signature	/Peter Ludwig/	Date Considered	04/19/2007

² 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 Applicant's unique citation designation number (optional). ³ 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.

Notice of References Cited	Application/Control No. 11/304,794	Applicant(s)/Patent Under Reexamination STEFIK ET AL.	
	Examiner Peter L. Ludwig	Art Unit 3621	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-			
	B	US-			
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	Henry H. Perritt, JR, Knowbots, Permissions Headers and Contract Law, 30 April 1993, Villanova Law School, pages 1-24, http://www.ifla.org/documents/infopol/copyright/perh2.txt .
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Electronic Patent Application Fee Transmittal

Application Number:	11304794			
Filing Date:	16-Dec-2005			
Title of Invention:	System for controlling the distribution and use of digital works			
First Named Inventor/Applicant Name:	Mark J. Stefik			
Filer:	Stephen M. Hertzler			
Attorney Docket Number:	111325-028700			
Filed as Large Entity				
Utility Filing Fees				
Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Extension - 3 months with \$0 paid	1253	1	1050	1050

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Miscellaneous:				
Total in USD (\$)				1050

REMARKS

Claims 1-37 were pending in this application. By this amendment, claims 1, 3-6, 9-18, 21-29, and 33-37 are amended. Thus, claims 1-37 remain pending in this application. In view of the above amendments and the following remarks, Applicants respectfully request reconsideration and allowance of the above-identified application in view of the above amendments and the following remarks.

Claims 3-6, 9-11, 12-13, 15-18, 21-25, 27-29, and 33-37 stand rejected under 35 U.S.C. § 112, second paragraph, as being indefinite.

In particular, with respect to claims 3-6, 9-11, 15-18, 21-25, 27-29, and 33-35, the Examiner asserts that the limitation “the content” or “digital content” has insufficient antecedent basis. However, claims 3-6, 9-11, 15-18, 21-25, 27-29, and 33-35 have been amended herein to recite “digital document” instead of “content” or “digital content,” which clearly has antecedent basis.

With respect to claims 12-13, 24-25, and 36-37, the Examiner asserts that these claims are vague and indefinite, particularly regarding which statutory class of invention is being claimed. However, claims 12, 24, and 36, as amended herein, are directed to computer systems, and claims 13, 25, and 37, as amended herein, are directed to computer readable media having one or more computer readable instructions embedded thereon, both of which are believed to be proper statutory classes.

Thus, Applicants respectfully request that the rejection of claims 3-6, 9-11, 12-13, 15-18, 21-25, 27-29, and 33-37 under 35 U.S.C. § 112 be reconsidered and withdrawn.

Claims 1-37 stand provisionally rejected on the ground of nonstatutory double patenting over claims 15-98 of copending application no. 11/198,216. While Applicants believe claims 1-37 are patentably distinct from claims 15-98 of copending application no. 11/198,216, Applicants respectfully request that this rejection be held in abeyance until the claims are otherwise in condition for allowance.

Claims 1-37 stand rejected under 35 U.S.C. § 102(b) in view of Perritt (Perritt JR – Knowbots, Permissions Headers and Contract Law). However, Applicants respectfully submit that Perritt fails to disclose or suggest each and every feature recited in claims 1-37, as presented herein.

For example, independent claim 1, as presented herein, recites a method for securely rendering digital documents, comprising *retrieving a digital document and at least one usage right associated with the digital document from a document repository*, the at least one usage right specifying a manner of use indicating the manner in which the digital document can be rendered, *storing the digital document and the at least one usage right separately in a document platform*, determining whether the digital document may be rendered by the document platform based on at least the at least one usage right, and if the at least one usage right allows the digital document to be rendered on the document platform, rendering the digital document by the document platform.

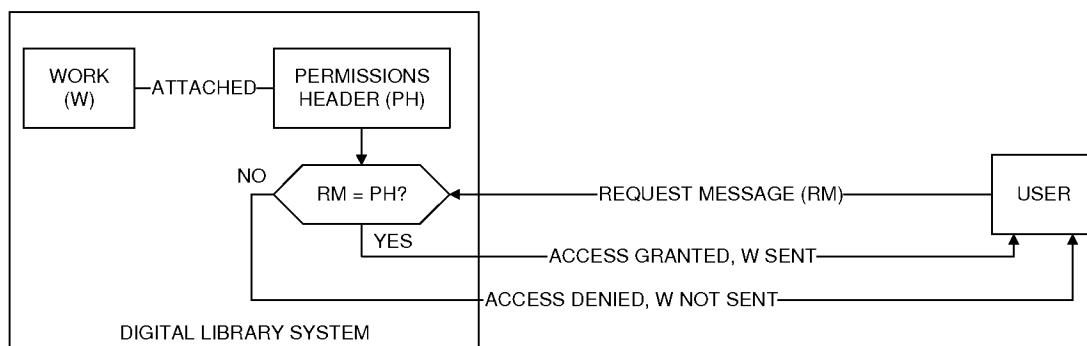
In addition, independent claim 14, as presented herein, recites a method for securely rendering digital documents, comprising *storing a digital document and at least one usage right separately in a document repository*, wherein the at least one usage right is associated with the digital document, receiving a request from a document platform for access to the digital document, determining whether the request may be granted based on the at least one usage right, the determining step including authenticating the document platform and determining whether the at least one usage right includes a manner of use that allows transfer of the digital document to the document platform, if the usage right allows the transfer of the digital document to the document platform, *transferring the digital document and at least one of the at least one usage rights associated with the digital document to the document platform*, *storing the digital document and the at least one of the at least one usage rights in the document platform*, wherein the at least one usage right is stored separately from the digital document, and rendering the digital document by the document platform.

Furthermore, independent claim 26, as presented herein, recites a method for securely rendering digital documents, comprising receiving a request to transfer a digital document from a document repository to a document platform, the digital document having at least one

usage right associated therewith, authenticating the document platform by accessing at least one identifier associated with the document platform or with a user of the document platform and determining whether the identifier is associated with at least one of a document platform and user authorized to use the digital document, if the authenticating step is successful, determining whether the digital document may be transferred and stored on a document platform based on a manner of use included in the at least one usage right, if the at least one usage right allows the digital document to be transferred to a document platform, *transferring the digital document and at least one of the at least one usage rights associated with the digital document to the document platform, storing the digital document and the at least one of the at least one usage rights in the document platform, wherein the at least one of the at least one usage rights is stored separately from the digital document*, and rendering the digital document by the document platform.

In contrast, as is shown in the diagram below, Perritt is directed to a digital library system, wherein permissions header (PH) is attached to a work (W) and a request message (RM) from a user is matched against the permissions header. If there is a match between the request message and the permissions header, access to the work is granted and the work is sent to the user. If, however, there is no match between the request message and the permissions header, access to the work is not granted and the work is not sent to the user.

Perritt



In addition, Perritt fails to disclose or suggest any sort of client side engagement for controlling use of a document, including, for example, a license is issued to a web service

client. Instead, Perritt merely discloses that control is purely managed on the server side. According to Perritt, the use of the document is no longer controlled at the client side when the document leaves the server domain.

Thus, Perritt fails to disclose or suggest at least “retrieving a digital document and at least one usage right associated with the digital document from a document repository,” “storing the digital document and the at least one usage right separately in a document platform,” and “if the at least one usage right allows the digital document to be rendered on the document platform, rendering the digital document by the document platform” as is recited in independent claim 1, “transferring the digital document and at least one of the at least one usage rights associated with the digital document to the document platform,” “storing the digital document and the at least one of the at least one usage rights in the document platform, wherein the at least one usage right is stored separately from the digital document,” and “rendering the digital document by the document platform” as is recited in independent claim 14, and “transferring the digital document and at least one of the at least one usage rights associated with the digital document to the document platform,” “storing the digital document and the at least one of the at least one usage rights in the document platform, wherein the at least one of the at least one usage rights is stored separately from the digital document,” and “rendering the digital document by the document platform” as is recited in independent claim 26. These novel features provide a degree of control on the client side not disclosed, suggested, or rendered obvious by Perritt.

Accordingly, for at least the above reasons, Perritt fails to disclose or suggest each and every feature recited in independent claims 1, 14, and 26 as amended herein. Dependent claims 2-13, 15-25, and 27-37 are also allowable for at least the above reasons, and on their own merits. Therefore, Applicants respectfully submit that the rejection of claims 1-37 under 35 U.S.C. § 102(b) in view of Perritt should be reconsidered and withdrawn.

In view of the foregoing, it is submitted that the present application is in condition for allowance and a notice to that effect is respectfully requested. If, however, the Examiner deems that any issue remains after considering this response, the Examiner is invited to contact the undersigned attorney to expedite the prosecution and engage in a joint effort to

work out a mutually satisfactory solution.

Except for issue fees payable under 37 C.F.R. § 1.18, the Commissioner is hereby authorized by this paper to charge any additional fees during the entire pendency of this application including fees due under 37 C.F.R. §§ 1.16 and 1.17 which may be required, including any required extension of time fees, or credit any overpayment to Deposit Account No. 19-2380. This paragraph is intended to be a **CONSTRUCTIVE PETITION FOR EXTENSION OF TIME** in accordance with 37 C.F.R. § 1.136(a)(3).

Respectfully submitted,
NIXON PEABODY, LLP

Date: November 1, 2007

/Stephen M. Hertzler, Reg. No. 58,247/
Stephen M. Hertzler
Registration No. 58,247

Customer No.: 22204
NIXON PEABODY LLP
401 9th Street, N.W., Suite 900
Washington, D.C. 20004-2128
(202) 585-8000

Amendments to the Claims

1. (Currently Amended) A method for securely rendering digital documents, comprising:

retrieving a digital document and at least one usage right associated with the digital document from a document repository, the at least one usage right specifying a manner of use indicating the manner in which the digital document can be rendered;

storing ~~[[a]]~~ the digital document and the at least one usage right separately in a document platform;

~~storing at least one usage right associated with the digital document, the at least one usage right specifying a manner of use indicating the manner in which the digital document can be rendered by the document platform;~~

determining whether the digital document may be rendered by the document platform based on at least the at least one usage right; and

if the at least one usage right allows the digital document to be rendered on the document platform, rendering the digital document by the document platform.

2. (Original) The method of claim 1, wherein the manner of use includes the number of times the digital document can be rendered.

3. (Currently Amended) The method as recited in claim 1, wherein at least one part of the ~~content~~ digital document and at least one usage right are included in different files.

4. (Currently Amended) The method as recited in claim 1, wherein at least one part of the ~~content~~ digital document and at least one usage right are stored in the same file.

5. (Currently Amended) The method as recited in claim 1, wherein at least a portion of the ~~digital content~~ digital document is a software program.

6. (Currently Amended) The method as recited in claim 1, wherein at least one usage right comprises a revenue identifier for identifying a revenue owner of the ~~digital content~~

digital document.

7. (Original) The method as recited in claim 1, wherein at least one usage right also specifies one or more conditions which must be satisfied before the manner of rendering may be exercised.

8. (Original) The method as recited in claim 7, wherein at least one condition includes determining the presence of a digital ticket.

9. (Currently Amended) The method as recited in claim 1, wherein at least one of the usage rights or a part of the ~~content~~-digital document are stored on a removable storage device.

10. (Currently Amended) The method as recited in claim 1, wherein at least one part of the ~~content~~-digital document and at least one usage right are stored on the same device.

11. (Currently Amended) The method as recited in claim 1, wherein at least one of the ~~content~~-digital document and at least one usage right are stored on different devices.

12. (Currently Amended) ~~A The method as recited in claim 1, wherein the method is implemented with a computer system including one or more hardware and/or software components, the computer system being operable to implement the method of claim 1.~~

13. (Currently Amended) ~~A computer readable medium having The method as recited in claim 1, wherein the method is implemented via one or more computer readable instructions embedded [[on a]] thereon, the computer readable medium and instructions being configured to cause one or more computer processors to perform the steps of the method of claim 1.~~

14. (Currently Amended) A method for securely rendering digital documents, comprising:

storing a digital document and at least one usage right separately in a document repository, wherein the at least one usage right is associated with the digital document;

~~storing at least one usage right associated with the digital document;~~

receiving a request from a document platform for requesting access to the digital document from a document platform;

determining whether the request may be granted based on the at least one usage right, the determining step including authenticating the document platform and determining whether the at least one usage right includes a manner of use that allows transfer of the digital document to the document platform;

if the usage right allows the transfer of the digital document to the document platform, transferring ~~[[at]]~~ the digital document and at least one of the at least one usage rights associated with the digital document to the document platform;

storing the digital document and the at least one of the at least one usage rights in the document platform, wherein the at least one usage right is stored separately from the digital document; and

rendering the digital document by the document platform.

15. (Currently Amended) The method as recited in claim 14, wherein at least one part of the ~~content~~ digital document and at least one usage right are included in different files.

16. (Currently Amended) The method as recited in claim 14, wherein at least one part of the ~~content~~ digital document and at least one usage right are stored in the same file.

17. (Currently Amended) The method as recited in claim 14, wherein at least a portion of the ~~digital content~~ digital document is a software program.

18. (Currently Amended) The method as recited in claim 14, wherein at least one usage right comprises a revenue identifier for identifying a revenue owner of the ~~digital content~~ digital document.

19. (Original) The method as recited in claim 14, wherein at least one usage right also specifies one or more conditions which must be satisfied before the manner of rendering may be exercised.

20. (Original) The method as recited in claim 19, wherein at least one condition includes determining the presence of a digital ticket.

21. (Currently Amended) The method as recited in claim 14, wherein at least one of the usage rights or a part of the ~~content~~ digital document are stored on a removable storage device.

22. (Currently Amended) The method as recited in claim 14, wherein at least one part of the ~~content~~ digital document and at least one usage right are stored on the same device.

23. (Currently Amended) The method as recited in claim 14, wherein at least one of the ~~content~~ digital document and at least one usage right are stored on different devices.

24. (Currently Amended) A ~~The method as recited in claim 14, wherein the method is implemented with a computer system including one or more hardware and/or software components, the computer system being operable to implement the method of claim 14.~~

25. (Currently Amended) A computer readable medium having ~~The method as recited in claim 14, wherein the method is implemented via one or more computer readable instructions embedded [[on a]] thereon, the computer readable medium and instructions being configured to cause one or more computer processors to perform the steps of the method of~~ claim 14.

26. (Currently Amended) A method for securely rendering digital documents, comprising:

receiving a request to transfer a digital document from a document repository to a document platform, the digital document having at least one usage right associated therewith;

authenticating the document platform by accessing at least one identifier associated with the document platform or with a user of the document platform and determining whether the identifier is associated with at least one of a document platform and user authorized to use the digital document;

if the authenticating step is successful, determining whether the digital document may be transferred and stored on a document platform based on a manner of use included in the at least one usage right;

if the at least one usage right allows the digital document to be transferred to a document platform, transferring the digital document and at least one of the at least one usage rights associated with the digital document to the document platform;

storing the digital document and the at least one of the at least one usage rights in the document platform, wherein the at least one of the at least one usage rights is stored separately from the digital document; and

rendering the digital document by the document platform.

27. (Currently Amended) The method as recited in claim 26, wherein at least one part of the ~~content~~ digital document and at least one usage right are included in different files.

28. (Currently Amended) The method as recited in claim 26, wherein at least one part of the ~~content~~ digital document and at least one usage right are stored in the same file.

29. (Currently Amended) The method as recited in claim 26, wherein at least a portion of the ~~digital content~~ digital document is a software program.

30. (Original) The method as recited in claim 26, wherein at least one usage right comprises a revenue identifier for identifying a revenue owner of the digital content.

31. (Original) The method as recited in claim 26, wherein at least one usage right also specifies one or more conditions which must be satisfied before the manner of rendering may be exercised.

32. (Original) The method as recited in claim 31, wherein at least one condition includes determining the presence of a digital ticket.

33. (Currently Amended) The method as recited in claim 26, wherein at least one of the usage rights or a part of the ~~content~~ digital document are stored on a removable storage device.

34. (Currently Amended) The method as recited in claim 26, wherein at least one part of the ~~content~~ digital document and at least one usage right are stored on the same device.

35. (Currently Amended) The method as recited in claim 26, wherein at least one of the ~~content~~ digital document and at least one usage right are stored on different devices.

36. (Currently Amended) A ~~The method as recited in claim 26, wherein the method is implemented with a computer system including one or more hardware and/or software components, the computer system being operable to implement the method of claim 26.~~

37. (Currently Amended) A computer readable medium having ~~The method as recited in claim 26, wherein the method is implemented via one or more computer readable instructions embedded [[on a]] thereon, the computer readable medium and instructions being configured to cause one or more computer processors to perform the steps of the method of~~ claim 26.

Electronic Acknowledgement Receipt

EFS ID:	2416243
Application Number:	11304794
International Application Number:	
Confirmation Number:	6096
Title of Invention:	System for controlling the distribution and use of digital works
First Named Inventor/Applicant Name:	Mark J. Stefik
Customer Number:	22204
Filer:	Stephen M. Hertzler
Filer Authorized By:	
Attorney Docket Number:	111325-028700
Receipt Date:	01-NOV-2007
Filing Date:	16-DEC-2005
Time Stamp:	22:30:59
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment was successfully received in RAM	\$ 1050
RAM confirmation Number	3832
Deposit Account	192380
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: Charge any Additional Fees required under 37 C.F.R. Section 1.16 and 1.17	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes) /Message Digest	Multi Part /.zip	Pages (if appl.)
1		028700_2007-11-01_Response_to_Office_Action.pdf	69908 e4bfff75d5541a5cf9fdd6ace685759d22911e4ee	yes	12
	Multipart Description/PDF files in .zip description				
	Document Description		Start	End	
	Amendment - After Non-Final Rejection		1	1	
	Claims		2	7	
	Applicant Arguments/Remarks Made in an Amendment		8	12	
Warnings:					
Information:					
2	Fee Worksheet (PTO-06)	fee-info.pdf	8147 d4bc73f58093d872f1e01a36b1e8acfbff0223da	no	2
Warnings:					
Information:					
Total Files Size (in bytes):			78055		
This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503. <u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application. <u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course. <u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.					

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:	)	Confirmation No.: 6096
Mark J. Stefik, et al.	)	Group Art Unit: 3621
Serial No. 11/304,794	)	Examiner: Ludwig, Peter L.
Filed: December 16, 2005	)	
For: SYSTEM FOR CONTROLLING	)	Date: November 1, 2007
THE DISTRIBUTION AND	)	
USE OF DIGITAL WORKS	)	

AMENDMENT

MAIL STOP AMENDMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In response to the office action mailed May 1, 2007, Applicants respectfully request reconsideration and allowance of the above-identified application based on the following amendments and remarks.

Amendments to the Claims begin on page 2 of this paper.

Remarks begin on page 8 of this paper.

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.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875					Application or Docket Number 11/304,794		Filing Date 12/16/2005		☐ To be Mailed	
APPLICATION AS FILED – PART I										
(Column 1)			(Column 2)		SMALL ENTITY ☐ OR		OTHER THAN SMALL ENTITY			
FOR	NUMBER FILED	NUMBER EXTRA	RATE (\$)	FEE (\$)	OR	RATE (\$)	FEE (\$)			
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A				
☐ SEARCH FEE (37 CFR 1.16(k), (l), or (m))	N/A	N/A	N/A			N/A				
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A				
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X \$	=		X \$	=			
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X \$	=		X \$	=			
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$250 (\$125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).									
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))										
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL				
APPLICATION AS AMENDED – PART II										
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY OR		OTHER THAN SMALL ENTITY		
AMENDMENT	11/01/2007	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$)	ADDITIONAL FEE (\$)	OR	RATE (\$)	ADDITIONAL FEE (\$)	
	Total (37 CFR 1.16(i))	* 37	Minus	** 37	=	0		X \$50=	0	
	Independent (37 CFR 1.16(h))	* 3	Minus	***3	=	0		X \$210=	0	
☐ Application Size Fee (37 CFR 1.16(s))										
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))										
					TOTAL ADD'L FEE			TOTAL ADD'L FEE	0	
(Column 1)		(Column 2)		(Column 3)		SMALL ENTITY OR		OTHER THAN SMALL ENTITY		
AMENDMENT		CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$)	ADDITIONAL FEE (\$)	OR	RATE (\$)	ADDITIONAL FEE (\$)	
	Total (37 CFR 1.16(i))	*	Minus	**	=			X \$	=	
	Independent (37 CFR 1.16(h))	*	Minus	***	=			X \$	=	
☐ Application Size Fee (37 CFR 1.16(s))										
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))										
					TOTAL ADD'L FEE			TOTAL ADD'L 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.										

Legal Instrument Examiner:
ANTHONY WILLIAMS

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

<i>Index of Claims</i> 	Application/Control No. 11304794	Applicant(s)/Patent Under Reexamination STEFIK ET AL.
	Examiner Ludwig, Peter L	Art Unit 3621

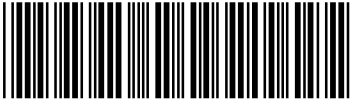
✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant				☐ CPA		☐ T.D.		☐ R.1.47	
CLAIM		DATE							
Final	Original	04/19/2007	01/14/2008						
	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	✓	✓						

<i>Index of Claims</i> 	Application/Control No. 11304794	Applicant(s)/Patent Under Reexamination STEFIK ET AL.
	Examiner Ludwig, Peter L	Art Unit 3621

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant		☐ CPA		☐ T.D.		☐ R.1.47			
CLAIM		DATE							
Final	Original	04/19/2007	01/14/2008						
	37	✓	✓						

Search Notes 	Application/Control No. 11304794	Applicant(s)/Patent Under Reexamination STEFIK ET AL.
	Examiner Ludwig, Peter L	Art Unit 3621

SEARCHED			
Class	Subclass	Date	Examiner

SEARCH NOTES		
Search Notes	Date	Examiner
East Search, see attached search history	1/14/2008	JM
Search Help from SPE Kambiz Abdi (3692)	1/11/2008	JM

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	6136	rights with ((digital near3 document) or (content))	US-PGPUB; USPAT; FPRS; EPO; JPO	OR	OFF	2008/01/14 07:53
L2	124	1 and (@ad<="19941123" OR @pd<="19941123" OR @prad<="19941123" OR @rlad<="19941123")	US-PGPUB; USPAT; FPRS; EPO; JPO	OR	OFF	2008/01/14 07:55
L3	46	2 NOT Stefik.in.	US-PGPUB; USPAT; FPRS; EPO; JPO	OR	OFF	2008/01/14 07:56

1/14/2008 8:27:48 AM

C:\Documents and Settings\jmurrough\My Documents\EAST\Workspaces\Default.wsp

Notice of References Cited	Application/Control No. 11/304,794		Applicant(s)/Patent Under Reexamination STEFIK ET AL.	
	Examiner Joshua Murdough		Art Unit 3621	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,135,646 A	10-2000	Kahn et al.	709/217
*	B	US-7,130,087 B2	10-2006	Rhoads, Geoffrey B.	358/3.28
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.

11/304,794

12/16/2005

Mark J. Stefik

111325-028700

6096

22204 7590 01/23/2008

NIXON PEABODY, LLP

401 9TH STREET, NW

SUITE 900

WASHINGTON, DC 20004-2128

EXAMINER

MURDOUGH, JOSHUA A

ART UNIT

PAPER NUMBER

3621

MAIL DATE

DELIVERY MODE

01/23/2008

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 11/304,794	Applicant(s) STEFIK ET AL.	
	Examiner Joshua Murdough	Art Unit 3621	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 01 November 2007.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-37 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-37 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 13 March 2006 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)	4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)	5) ☐ Notice of Informal Patent Application
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____	6) ☐ Other: _____

DETAILED ACTION

Acknowledgements

1. This Office Action has been given Paper No. 20071231 for reference purposes only.
2. The Examiner for this application has changed. Please note that the Examiner of record is now Joshua A. Murdough for any subsequent correspondence.
3. Claims 1-37 are currently pending.

Response to Amendment

4. The amendment to the claims received 1 November 2007 has been entered.

Claim Rejections - 35 USC § 112 2nd Paragraph

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claims 12-13, 24-25, and 36-37 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

- a. Claims 12-13, 24-25, and 36-37 are hybrid claims. See MPEP 2173.05(p) II. The Examiner is unclear which statutory class the Applicants are intending to claim. With the claim directed at a method and then the Applicants are trying to claim structural components, such as, in claim 13 a computer readable medium configured to cause one or more computer processors to perform the steps of the method. The USPTO deems these types of claims “hybrid” claims and deems them non-statutory under 35 U.S.C. 112,

second paragraph and 35 U.S.C. 101. The Examiner will be viewing the claims as processes without structural elements, but the Applicants must amend if Applicants are trying to claim structural elements within the method.

Claim Rejections - 35 USC § 101

7. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

8. Claims 12, 13, 24, 25, 36, 37 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

a. Claims 12, 24, and 36 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. As stated in the respective claims, the “computer system” could be comprised of “one or more software components”.

Computer software is, per se, non-statutory because it lacks tangibility.

b. Regarding claims 12, 13, 24, 25, 36, and 37, these claims are hybrid claims. See the discussion above on hybrid claims.

Double Patenting

9. It is noted that Applicant requested that the double patenting rejections be held in abeyance until patentable subject matter is found in the instant application.

The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or

improper timewise extension of the “right to exclude” granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

Claims 1-37 are provisionally rejected on the ground of nonstatutory double patenting over claims 15-61 of allowed but not yet issued Application No. 11/198,216. This is a provisional double patenting rejection since the conflicting claims have not yet been patented.

The subject matter claimed in the instant application is fully disclosed in the referenced copending application and would be covered by any patent granted on that copending application since the referenced copending application and the instant application are claiming common subject matter, as follows: Both applications teach a method of rendering a document.

Furthermore, there is no apparent reason why applicant would be prevented from presenting claims corresponding to those of the instant application in the other copending application. See *In re Schneller*, 397 F.2d 350, 158 USPQ 210 (CCPA 1968). See also MPEP § 804.

Claim Rejections - 35 USC § 102

10. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

11. Claims 1-37 are rejected under 35 U.S.C. 102(b) as being anticipated by Perritt JR – Knowbots, Permissions Headers and Contract Law (Reference U on attached PTO – 892 form)[hereinafter Perritt JR].

12. As per claim 1, Perritt JR clearly discloses a method for securely rendering digital documents, comprising:

- **Retrieving a digital document** (information object, page 3) **and at least one usage right associated with the digital document** (in permissions header, page 1) **from a document repository** (library, page 3), **the at least one usage right specifying a**

manner of use indicating the manner in which the digital document can be rendered

(page 4 – “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.' The EBR, thus, is apparently equivalent to the permissions header discussed in this paper.”

- **storing a digital document and at least one usage right separately in a document platform** (page 3 – “Basic Concepts”);
- **determining whether the digital document may be rendered by the document platform based on at least the at least one usage right** (page 1 – “If the request message and the permissions header match, the user would obtain access to the work.”);
- **if the at least one usage right allows the digital document to be rendered on the document platform, rendering the digital document by the document platform** (page 1 – i.e. “If the request message and the permissions header match, the user would obtain access to the work.”).

13. As per claim 2, Perritt JR clearly discloses the method of claim 1 as described above.

Perritt JR further teaches **wherein the manner of use includes the number of times the digital document can be rendered** (page 5 – “Elements of Data Structure”, restricted usage and quantitative limits).

As per claim 3, Perritt JR clearly discloses the method as recited in claim 1 as described above. Perritt JR further teaches **wherein at least one part of the digital document and at least one usage right are included in different files** (page 4 – i.e. “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.”).

As per claim 4, Perritt JR clearly discloses the method as recited in claim 1 as described above. Perritt JR further discloses **wherein at least one part of the digital document and at least one usage right are stored in the same file** (page 1 – i.e. “a work would have attached to it a permissions header”).

As per claim 5, Perritt JR clearly discloses the method as recited in claim 1 as described above. Perritt JR further discloses **wherein at least a portion of the digital document is a software program** (page 1 – “Introduction”; Examiner is interpreting the content in the digital library as software programs).

As per claim 6, Perritt JR clearly discloses the method as recited in claim 1 as described above. Perritt JR further teaches **wherein at least one usage right comprises a revenue identifier for identifying a revenue owner of the digital document** (page 4 - i.e. “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.” Examiner is interpreting that if the user must pay a certain fee and the fee is included in the permissions header, then the user must know who to pay the money to, which in turn leads to knowing who the “revenue owner” of the content is).

As per claim 7, Perritt JR clearly discloses the method as recited in claim 1 as described above. Perritt JR further teaches **wherein at least one usage right also specifies one or more conditions which must be satisfied before the manner of rendering may be exercised** (page 8 “Legal Issues” – “If an outright transfer is intended, then the present copyright statute requires a writing signed by the owner of the rights conveyed.”).

As per claim 8, Perritt JR clearly discloses the method as recited in claim 7 as described above. Perritt JR further teaches **wherein at least one condition includes determining the presence of a digital ticket** (According to the explanation from Applicant’s Specification “In the present invention, a "digital ticket" is used to enable the ticket holder to exercise usage rights specifying the requirement of the digital ticket.” The Examiner is interpreting the “permission header” as a digital ticket due to it enabling the ticket holder to exercise usage rights specifying the requirement of the digital ticket (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 (page 1 – “Introduction”), i.e. 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 (page 16)).

As per claim 9, Perritt JR clearly discloses the method as recited in claim 1 as described above. Perritt JR further teaches **wherein at least one of the usage rights or a part of the digital document are stored on a removable storage device** (page 22, #23-24 and 26).

As per claim 10, Perritt JR clearly discloses the method as recited in claim 1 as described above. Perritt JR further teaches **wherein at least one part of the digital document and at least one usage right are stored on the same device** (page 1 – i.e. “a work would have attached to it a permissions header”).

As per claim 11, Perritt JR clearly discloses the method as recited in claim 1 as described above. Perritt JR further teaches **wherein at least one of the digital document and at least one usage right are stored on different devices** (page 6 – “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."").

As per claim 12, Perritt JR clearly discloses the method as recited in claim 1 as described above. Perritt JR further teaches **wherein the method is implemented with a computer system including one or more hardware and/or software components** (page 3 – “Basic Concepts”).

As per claim 13, Perritt JR clearly discloses the method as recited in claim 1 as described above. Perritt JR further teaches **wherein the method is implemented via one or more computer readable instructions embedded thereon, the computer readable instructions being configured to cause one or more computer processors to perform the steps of the method** (page 3 – “Basic Concepts”).

As per claim 14, Perritt JR clearly discloses a method for securely rendering digital documents, comprising:

- **storing a digital document** (information object, page 3) **and at least one usage right separately** (in permissions header, page 1) **in a document repository** (library, page 3), **wherein the at least one usage right is associated with the digital document** (page 3 – “Basic Concepts”);
- **receiving a request from a document platform** (operated by a user) **for access to the digital document** (page 1 – “Introduction”);

- **determining whether the request may be granted based on the at least one usage right, the determining step including authenticating the document platform and determining whether the at least one usage right includes a manner of use that allows transfer of the digital document to the document platform** (page 1 – “Introduction”);
- **if the usage right allows the transfer of the digital document to the document platform, transferring the digital document and at least one of the at least one usage rights associated with the digital document to the document platform** (page 1 – “Introduction”);
- **storing the digital document and the at least one of the at least one usage rights in the document platform** (page 4, paragraph 4 – the Examiner understands copy, as used herein, to mean saving a duplicate version to the user’s computer), **wherein the at least one usage right is stored separately from the digital document** (a header is a separate section of the file containing the information object and therefore is stored separately);
and
- **rendering the digital document by the document platform** (page 1 – “Introduction”).

As per claim 15, Perritt JR clearly discloses the method as recited in claim 14 as described above. Perritt JR further teaches **wherein at least one part of the digital document and at least one usage right are included in different files** (page 6, “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."").

As per claim 16, Perritt JR clearly discloses the method as recited in claim 14 as described above. Perritt JR further teaches **wherein at least one part of the digital document and at least one usage right are stored in the same file** (page 1 – “Introduction”).

As per claim 17, Perritt JR clearly discloses the method as recited in claim 14 as described above. Perritt JR further teaches **wherein at least a portion of the digital document is a software program** (page 1 – “Introduction”; Examiner is interpreting the content in the digital library as software programs).

As per claim 18, Perritt JR clearly discloses the method as recited in claim 14 as described above. Perritt JR further teaches **wherein at least one usage right comprises a revenue identifier for identifying a revenue owner of the digital document** (page 4 - i.e. “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.” Examiner is interpreting that if the

user must pay a certain fee and the fee is included in the permissions header, then the user must know who to pay the money to, which in turn leads to knowing who the “revenue owner” of the content is).

As per claim 19, Perritt JR clearly discloses the method as recited in claim 14 as described above. Perritt JR further teaches **wherein at least one usage right also specifies one or more conditions which must be satisfied before the manner of rendering may be exercised** (page 8 “Legal Issues” – “If an outright transfer is intended, then the present copyright statute requires a writing signed by the owner of the rights conveyed.”).

As per claim 20, Perritt JR clearly discloses the method as recited in claim 19 as described above. Perritt JR further teaches **wherein at least one condition includes determining the presence of a digital ticket** (According to the explanation from Applicant’s Specification “In the present invention, a "digital ticket" is used to enable the ticket holder to exercise usage rights specifying the requirement of the digital ticket.” The Examiner is interpreting the “permission header” as a digital ticket due to it enabling the ticket holder to exercise usage rights specifying the requirement of the digital ticket (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 (page 1 – “Introduction”), i.e. 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 (page 16)).

As per claim 21, Perritt JR clearly discloses the method as recited in claim 14 as described above. Perritt JR further teaches **wherein at least one of the usage rights or a part of the digital document are stored on a removable storage device** (page 22, #23-24 and 26).

As per claim 22, Perritt JR clearly discloses the method as recited in claim 14 as described above. Perritt JR further teaches **wherein at least one part of the digital document and at least one usage right are stored on the same device** (page 1 – “Introduction”).

As per claim 23, Perritt JR clearly discloses the method as recited in claim 14 as described above. Perritt JR further teaches **wherein at least one of the digital document and at least one usage right are stored on different devices** (page 6, “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."").

As per claim 24, Perritt JR clearly discloses the method as recited in claim 14 as described above. Perritt JR further teaches **A computer system including one or more hardware and/or software components, the computer system being operable to implement the method of claim 14** (page 3 – “Basic Concepts”).

As per claim 25, Perritt JR clearly discloses the method as recited in claim 14 as described above. Perritt JR further teaches **a computer readable medium having one or more computer readable instructions embedded thereon, the computer readable instructions being configured to cause one or more computer processors to perform the steps of the method of claim 14** (page 3 – “Basic Concepts”).

As per claim 26, Perritt JR clearly discloses a method for securely rendering digital documents, comprising:

- **receiving a request** (page 1 – “Introduction”) **to transfer a digital document from a document repository to a document platform** (operated by a user) **the digital document having at least one usage right associated therewith** (page 3 – “Basic Concepts”);
- **authenticating the document platform by accessing at least one identifier associated with the document platform or with a user of the document platform and determining whether the identifier is associated with at least one of a document**

platform and user authorized to use the digital document (page 7 – “Role of Encryption”);

- **if the authenticating step is successful, determining whether the digital document may be transferred and stored on a document platform based on a manner of use included in the at least one usage right** (page 7 – “Role of Encryption”);
- **if the at least one usage right allows the digital document to be transferred to a document platform, transferring the digital document and at least one of the at least one usage rights associated with the digital document to the document platform** (page 7 – “Role of Encryption”);
- **storing the digital document and the at least one of the at least one usage rights in the document platform** (page 4, paragraph 4 – the Examiner understands copy, as used herein, to mean saving a duplicate version to the user’s computer), **wherein the at least one usage right is stored separately from the digital document** (a header is a separate section of the file containing the information object and therefore is stored separately);
and
- **rendering the digital document by the document platform** (page 7 – “Role of Encryption”).

As per claim 27, Perritt JR clearly discloses the method as recited in claim 26 as described above. Perritt further teaches **wherein at least one part of the digital document and at least one usage right are included in different files** (page 6, “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."").

As per claim 28, Perritt JR clearly discloses the method as recited in claim 26 as described above. Perritt JR further teaches **wherein at least one part of the digital document and at least one usage right are stored in the same file** (page 1 – “Introduction”).

As per claim 29, Perritt JR clearly discloses the method as recited in claim 26 as described above. Perritt JR further teaches **wherein at least a portion of the digital document is a software program** (page 1 – “Introduction”; Examiner is interpreting the content in the digital library as software programs).

As per claim 30, Perritt JR clearly discloses the method as recited in claim 26 as described above. Perritt JR further teaches **wherein at least one usage right comprises a revenue identifier for identifying a revenue owner of the digital content** (page 4 - i.e. “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.” Examiner is interpreting that if the user must pay a certain fee and the fee is included in the permissions header, then the user must know who to pay the money to, which in turn leads to knowing who the “revenue owner” of the content is).

As per claim 31, Perritt JR clearly discloses the method as recited in claim 26 as described above. Perritt JR further teaches **at least one usage right also specifies one or more conditions which must be satisfied before the manner of rendering may be exercised** (page 8 “Legal Issues” – “If an outright transfer is intended, then the present copyright statute requires a writing signed by the owner of the rights conveyed.”).

As per claim 32, Perritt JR clearly discloses the method as recited in claim 31 as described above. Perritt JR further teaches **wherein at least one condition includes determining the presence of a digital ticket** (According to the explanation from Applicant’s Specification “In the present invention, a "digital ticket" is used to enable the ticket holder to exercise usage rights specifying the requirement of the digital ticket.” The Examiner is interpreting the “permission header” as a digital ticket due to it enabling the ticket holder to exercise usage rights specifying the requirement of the digital ticket (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 (page 1 – “Introduction”), i.e. 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 (page 16)).

As per claim 33, Perritt JR clearly discloses the method as recited in claim 26 as described above. Perritt JR further teaches **wherein at least one of the usage rights or a part of the digital document are stored on a removable storage device** (page 22, #23-24 and 26).

As per claim 34, Perritt JR clearly discloses the method as recited in claim 26 as described above. Perritt JR further teaches **wherein at least one part of the digital document and at least one usage right are stored on the same device** (page 1 – “Introduction”).

As per claim 35, Perritt JR clearly discloses the method as recited in claim 26 as described above. Perritt JR further teaches **wherein at least one of the digital document and at least one usage right are stored on different devices** (page 6, “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."").

As per claim 36, Perritt JR clearly discloses the method as recited in claim 26 as described above. Perritt JR further teaches **a computer system including one or more hardware and/or software components, the system being operable to implement the method of claim 26** (page 3 – “Basic Concepts”).

As per claim 37, Perritt JR clearly discloses the method as recited in claim 26 as described above. Perritt JR further teaches **wherein the method is implemented via one or more computer readable instructions embedded on a computer readable medium and configured to cause one or more computer processors to perform the steps of the method** (page 3 – “Basic Concepts”).

Claim Interpretations

Applicants are reminded that functional recitations using the word “for,” “configured to,” or other functional terms (*e.g. see* claim 6 which recites “for identifying a revenue owner of the digital content”) have been considered but are given little patentable weight¹ because they fail to add any structural limitations and are thereby regarded as intended use language. To be especially clear, all limitations have been considered. However a recitation of the intended use in a product claim must result in a structural difference between the claimed product and the prior art in order to patentably distinguish the claimed product from the prior art. If the prior art structure is capable of performing the intended use, then it reads on the claimed limitation. *In re Casey*, 370 F.2d 576, 152 USPQ 235 (CCPA 1967) (“The manner or method in which such machine is to be utilized is not germane to the issue of patentability of the machine itself.”); *In re Otto*, 136 USPQ 458, 459 (CCPA 1963). See also MPEP §§ 2114 and 2115. Unless expressly noted otherwise by the Examiner, the claim interpretation principles in this paragraph apply to all examined claims currently pending.

As per claims 1, 2, 7, 14 and 26, the Examiner interprets claim limitations that contain “*if, may, might, can, when and could*” statement(s), as optional language. As matter of linguistic precision, optional claim elements do not narrow claim limitations, since they can always be omitted (*In re Johnston*, 77 USPQ2d 1788 (CA FC 2006)). Language that suggests or makes optional but does not require steps to be performed or does not limit a claim to a particular structure does not limit the scope of a claim or claim limitation. For the sake of speeding up the

¹ See *e.g. In re Gulack*, 703 F.2d 1381, 217 USPQ 401, 404 (Fed. Cir. 1983)(stating that although all limitations must be considered, not all limitations are entitled to patentable weight.).

prosecution process, the Examiner has taken these limitations into account, but will not be given patentable weight unless positively recited.

Response to Arguments

Applicant's arguments filed 1 November 2007 have been fully considered but they are not persuasive.

Applicants Argue:

Perritt does not show “client side engagement for controlling use of a document.” (Page 10, bottom of page)

Examiner's Response:

Perritt discloses the use of a “Rights Management System (RMS)” (page 3, paragraph 6), which “would be an interactive distributed system capable of granting rights on line and permitting the use of copyrighted material in the Digital Library System.” (Id.) The Examiner interprets the system to include client and server computers.

Applicants Argue:

The limitations of the independent claims as amended are not shown. The remaining arguments to the merits of the claims are directed to the amendment, in part or in whole. (Pages 9-11)

Examiner's Response:

The Examiner has addressed the new limitations above in the rejections on the merits. Furthermore, the Examiner has added additional citations to allow Applicants to better understand the Examiner's grounds.

Examiner Note

Examiner has cited particular columns and line numbers in the references as applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings in the art and are applied to the specific limitations within the individual claim, other passages and figures may be applied as well. It is respectfully requested from the applicant, in preparing responses, to fully consider the reference in its entirety as potentially teaching all of part of the claimed invention as well as the context of the passage as taught by the prior art or disclosed by the examiner.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Kahn (6,135,646) and Rhoads (7,130,087) both show different means for establishing client-side usage controls. Rhoads makes use of containers and encryption to control distribution by clients, while Kahn uses tracking information that enables the distributor to know where the copy originated.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Joshua Murdough whose telephone number is (571) 270-3270. The examiner can normally be reached on Monday - Thursday, 7:00 a.m. - 5:00 p.m.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Andrew Fischer can be reached on (571) 272-6779. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/J. M./
Examiner, Art Unit 3621

/Andrew J Fischer/
Supervisory Patent Examiner, Art Unit 3621



SPRO
/S

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:	) Confirmation No.: 6096
Mark STEFIK et al.	) Group Art Unit: 3621
Application No.: 11/304,794	) Examiner: Murdough, Joshua A.
Filed: December 16, 2005	)
For: SYSTEM FOR CONTROLLING THE	) Date: March 6, 2008
DISTRIBUTION AND USE OF	)
DIGITAL WORKS	)

INFORMATION DISCLOSURE STATEMENT

United States Patent and Trademark Office
Customer Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

Dear 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)(2)(ii), copies of the cited U.S. patents (*i.e.*, Reference Cite Nos. 1–101) are not enclosed. Copies of the cited Foreign patents (*i.e.*, Reference Cite Nos. 102–173) are enclosed. Copies of the cited non-patent references (*i.e.*, Reference Cite Nos. 174–213) are enclosed. The references have been cited in recent oppositions in the European Patent Office relating to cases owned by assignee.

The Commissioner is hereby authorized to charge the **Deposit Account No. 19-2380** in the amount of **\$180.00** representing filing fees.

It is requested that the accompanying PTO/SB/08A be considered and made of record in the above-identified application. To assist the Examiner, the documents are listed on the attached form PTO/SB/08A. 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, or credit any overpayment to Deposit Account No. 19-2380.

03/07/2008 MAHMED1 00000023 192380 11304794
01 FC:1806 180.00 DA

Respectfully submitted,
NIXON PEABODY LLP

Date: March 6, 2008

By: /Marc S. Kaufman, Reg. # 35,212/
Marc S. Kaufman
Registration No. 35,212

NIXON PEABODY LLP
CUSTOMER NO.: 22204
401 9th Street, N.W., Suite 900
Washington, DC 20004
Tel: 202-585-8000
Fax: 202-585-8080



TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	11/304,794		
	Filing Date	December 16, 2005	
	First Named Inventor	Mark STEFIK et al.	
	Group Art Unit	3621	
	Examiner Name	Murdough, Joshua A.	
Total Number of Pages in This Submission		Attorney Docket Number	111325-028700

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	☐ Drawing(s) ☐ 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 ☐ Appeal Communication to Board of Appeals and Interferences ☐ Appeal Communication to Group (Appeal Notice, Brief, Reply Brief) ☐ Proprietary Information ☐ Status Letter ☒ Other Enclosure(s) (please identify below): 1. PTO Form 1449 2. One Box including 112 cited references
Remarks		☒ The Director 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	Marc S. Kaufman Registration No. 35,212 Nixon Peabody LLP 401 9th Street, N.W., Suite 900 Washington, D.C. 20004-2128
Signature	/Marc S. Kaufman, Reg. # 35,212/
Date	March 6, 2008

CERTIFICATE OF MAILING OR TRANSMISSION

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, P.O. Box 1450, Alexandria, Virginia 22313-1450, or facsimile transmitted to the U.S. Patent and Trademark Office (Fax No. (571) 273-8300) on the date shown below.			
Name (Print/Type)			
Signature		Date	

Effective on 12/08/2004.
Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

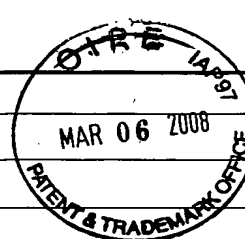
FEE TRANSMITTAL FOR FY 2008

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT \$180.00

Complete if Known

Application Number 11/304,794
Filing Date December 16, 2005
First Named Inventor Mark STEFIK et al.
Examiner Name Murdough, Joshua A.
Art Unit 3621
Attorney Docket No. 111325-028700



METHOD OF PAYMENT (check all that apply)

☐ Check ☐ Credit Card ☐ Money Order ☐ None ☐ Other (please identify): _____
☒ Deposit Account Deposit Account Number: 19-2380 Deposit Account Name: Nixon Peabody LLP

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

☒ Charge fee(s) indicated below ☐ Charge fee(s) indicated below, except for the filing fee
☒ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 ☒ Credit any overpayments

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-20238.

FEE CALCULATION

1. BASIC FILING, SEARCH AND EXAMINATION FEES

Application Type	FILING FEES		SEARCH FEES		EXAMINATION FEES		Fees Paid (\$)
	Fee (\$)	Small Entity Fee (\$)	Fee (\$)	Small Entity Fee (\$)	Fee (\$)	Small Entity Fee (\$)	
Utility	310	155	510	255	210	105	
Design	210	105	100	50	130	65	
Plant	210	105	310	155	160	80	
Reissue	310	155	510	255	620	310	
Provisional	210	105	N/A	N/A	N/A	N/A	

2. EXCESS CLAIM FEES

Fee Description	Fee (\$)	Small Entity Fee (\$)
Each claim over 20 (including Reissues)	50	25
Each independent claim over 3 (including Reissues)	210	105
Multiple dependent claims	370	185

Total Claims **Extra Claims** **Fee (\$)** **Fee Paid (\$)** **Multiple Dependent Claims**
_____ - 20 or HP = _____ x _____ = _____
Fee (\$): _____ Fee Paid (\$): _____
HP = highest number of total claims paid for, if greater than 20

Indep. Claims **Extra Claims** **Fee (\$)** **Fee Paid (\$)**
_____ - 3 or HP = _____ x _____ = _____
HP = highest number of independent claims paid for, if greater than 3

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$260 (\$130 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets **Extra Sheets** **Number of each additional 50 or fraction thereof** **Fee (\$)** **Fee Paid (\$)**
_____ - 100 = _____ / 50 = _____ (round up to a whole number) x _____ = _____

4. OTHER FEE(S)

Non-English Specification, \$130 fee (no small entity discount)

Other: Information Disclosure Filing Fee

Fees Paid (\$)

\$180.00

SUBMITTED BY

Signature	/Marc S. Kaufman, Reg. # 35,212/	Registration No. 35,212 (Attorney/Agent)	Telephone (202) 585-8000
Name (Print/Type)	Marc S. Kaufman	Date	March 6, 2008

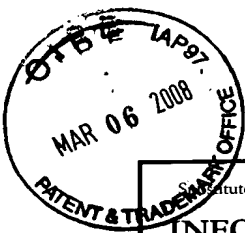
CERTIFICATE OF MAILING OR TRANSMISSION [35 CFR 1.8(a)]

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on _____.

Signature: _____

Name: _____

SEND TO: Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450



Substitute for form 1449A/PTO

**INFORMATION DISCLOSURE
STATEMENT BY APPLICANT**

(use as many sheets as necessary)

Complete if Known

Sheet	1	of	9	Application Number	11/304,794
				Filing Date	December 16, 2005
				First Named Inventor	Mark Stefik et al.
				Art Unit	3621
				Examiner Name	Murdough, Joshua A.
				Attorney Docket Number	111325/028700

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No. ¹	U.S. Patent Document Number – Kind Code ² (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	1	US 20010009026 A1	07-19-2001	Terao et al.	
	2	US 20010011276 A1	08-02-2001	Durst Jr. et al.	
	3	US 20010014206 A1	08-16-2001	Artigalas et al.	
	4	US 20010037467 A1	11-01-2001	O'Toole Jr. et al.	
	5	US 20010039659 A1	11-08-2001	Simmons et al.	
	6	US 20020001387 A1	01-03-2002	Dillon	
	7	US 20020035618 A1	03-21-2002	Mendez et al.	
	8	US 20020044658 A1	04-18-2002	Wasilewski et al.	
	9	US 20020056118 A1	05-09-2002	Hunter et al.	
	10	US 20020069282 A1	06-06-2002	Reisman	
	11	US 20020099948 A1	07-25-2002	Kocher et al.	
	12	US 20020127423 A1	09-12-2002	Kayanakis	
	13	US 20030097567 A1	05-22-2003	Terao et al.	
	14	US 20040052370 A1	03-18-2004	Katznelson	
	15	US 20040172552 A1	09-02-2004	Boyles et al.	
	16	US 4,159,468	06-26-1979	Barnes et al.	
	17	US 4,200,700	04-29-1980	Mäder	
	18	US 4,361,851	11-30-1982	Asip et al.	
	19	US 4,423,287	12-27-1983	Zeidler	
	20	US 4,429,385	01-31-1984	Cichelli et al.	
	21	US 4,621,321	11-04-1986	Boebert et al.	
	22	US 4,736,422	04-05-1988	Mason	
	23	US 4,740,890	04-26-1988	William	
	24	US 4,796,220	01-03-1989	Wolfe	
	25	US 4,816,655	03-28-1989	Musyck et al.	
	26	US 4,888,638	12-19-1989	Bohn	
	27	US 4,937,863	06-26-1990	Robert et al.	
	28	US 4,953,209	08-28-1990	Ryder et al.	
	29	US 4,977,594	12-11-1990	Shear	
	30	US 5,014,234	05-07-1991	Edwards	
	31	US 5,129,083	07-07-1992	Cutler et al.	
	32	US 5,138,712	08-11-1992	Corbin	
	33	US 5,174,641	12-29-1992	Lim	
	34	US 5,204,897	04-20-1993	Wyman	
	35	US 5,247,575	09-21-1993	Sprague et al.	
	36	US 5,260,999	11-09-1993	Wyman	
	37	US 5,276,444	01-04-1994	McNair	
	38	US 5,291,596	03-01-1994	Mita	
	39	US 5,293,422	03-08-1994	Loiacono	

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 Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at 222.uspto.gov or MPEP 901.04. 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.

10886767.1

Substitute for form 1449A/PTO				Complete if Known	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Application Number	11/304,794
				Filing Date	December 16, 2005
				First Named Inventor	Mark Stefik et al.
				Art Unit	3621
				Examiner Name	Murdough, Joshua A.
Sheet	2	of	9	Attorney Docket Number	111325/028700

U.S. PATENT DOCUMENTS					
Examiner Initials*	Cite No. ¹	U.S. Patent Document Number – Kind Code ² (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	40	US 5,335,275	08-02-1994	Millar et al.	
	41	US 5,337,357	08-09-1994	Chou et al.	
	42	US 5,386,369	01-31-1995	Christiano	
	43	US 5,453,601	09-26-1995	Rosen	
	44	US 5,485,577	01-16-1996	Eyer et al.	
	45	US 5,504,816	04-02-1996	Hamilton et al.	
	46	US 5,530,235	06-25-1996	Stefik et al.	
	47	US 5,535,276	07-09-1996	Ganesan	
	48	US 5,557,678	09-17-1996	Ganesan	
	49	US 5,629,980	05-13-1997	Stefik et al.	
	50	US 5,636,346	06-03-1997	Saxe	
	51	US 5,638,443	06-10-1997	Stefik et al.	
	52	US 5,708,709	01-13-1998	Rose	
	53	US 5,715,403	02-03-1998	Stefik	
	54	US 5,745,879	04-28-1998	Wyman	
	55	US 5,764,807	06-09-1998	Pearlman et al.	
	56	US 5,765,152	06-09-1998	Erickson	
	57	US 5,787,172	07-28-1998	Arnold	
	58	US 5,790,677	08-04-1998	Fox et al.	
	59	US 5,812,664	09-22-1998	Bernobich et al.	
	60	US 5,825,876	10-20-1998	Peterson	
	61	US 5,825,879	10-20-1998	Davis	
	62	US 5,838,792	11-17-1998	Ganesan	
	63	US 5,848,154	12-08-1998	Nishio et al.	
	64	US 5,848,378	12-08-1998	Shelton et al.	
	65	US 5,850,433	12-15-1998	Van Oorschot et al.	
	66	US 5,915,019	06-22-1999	Ginter et al.	
	67	US 5,917,912	06-29-1999	Ginter et al.	
	68	US 5,933,498	08-03-1999	Schneck et al.	
	69	US 5,940,504	08-17-1999	Griswold	
	70	US 5,982,891	11-09-1999	Ginter et al.	
	71	US 5,987,134	11-16-1999	Shin et al.	
	72	US 5,999,624	12-07-1999	Hopkins	
	73	US 6,006,332	12-21-1999	Rabne et al.	
	74	US 6,020,882	02-01-2000	Kinghorn et al.	
	75	US 6,047,067	04-04-2000	Rosen	
	76	US 6,073,234	06-06-2000	Kigo et al.	
	77	US 6,091,777	07-18-2000	Guetz et al.	
	78	US 6,112,239	08-29-2000	Kenner et al.	

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 Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at 222.uspto.gov or MPEP 901.04. 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.

10886767.1

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Complete if Known	
				Application Number	11/304,794
				Filing Date	December 16, 2005
				First Named Inventor	Mark Stefik et al.
				Art Unit	3621
				Examiner Name	Murdough, Joshua A.
Sheet	3	of	9	Attorney Docket Number	111325/028700

U.S. PATENT DOCUMENTS					
Examiner Initials [*]	Cite No. ¹	U.S. Patent Document Number – Kind Code ² (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	79	US 6,135,646	10-24-2000	Kahn et al.	
	80	US 6,141,754	10-31-2000	Choy	
	81	US 6,157,719	12-05-2000	Wasilewski et al.	
	82	US 6,169,976 B1	01-02-2001	Colosso	
	83	US 6,185,683 B1	02-06-2001	Ginter et al.	
	84	US 6,189,037 B1	02-13-2001	Adams et al.	
	85	US 6,189,146 B1	02-13-2001	Misra et al.	
	86	US 6,209,092 B1	03-27-2001	Linnartz	
	87	US 6,216,112 B1	04-10-2001	Fuller et al.	
	88	US 6,219,652 B1	04-17-2001	Carter et al.	
	89	US 6,236,971 B1	05-22-2001	Stefik et al.	
	90	US 6,307,939 B1	10-23-2001	Vigarie	
	91	US 6,353,888 B1	03-05-2002	Kakehi et al.	
	92	US 6,397,333 B1	05-28-2002	Söhne et al.	
	93	US 6,401,211 B1	06-04-2002	Brezak Jr. et al.	
	94	US 6,405,369 B1	06-11-2002	Tsuria	
	95	US 6,424,717 B1	07-23-2002	Pinder et al.	
	96	US 6,424,947 B1	07-23-2002	Tsuria et al.	
	97	US 6,487,659 B1	11-26-2002	Kigo et al.	
	98	US 6,516,052 B2	02-04-2003	Voudouris	
	99	US 6,516,413 B1	02-04-2003	Aratani et al.	
	100	US 6,523,745 B1	02-25-2003	Tamori	
	101	US 6,796,555 B1	09-28-2004	Blahut	

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 Applicant's unique citation designation number (optional). ² See Kinds Codes of USPTO Patent Documents at 222.uspto.gov or MPEP 901.04. 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.

10886767.1

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Complete if Known	
				Application Number	11/304,794
				Filing Date	December 16, 2005
				First Named Inventor	Mark Stefik et al.
				Art Unit	3621
				Examiner Name	Murdough, Joshua A.
Sheet	4	of	9	Attorney Docket Number	111325/028700

FOREIGN PATENT DOCUMENTS						
Examiner Initials ¹	Cite No. ¹	Foreign Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Country Code ²	Number ⁴ Kind Code ³ (if known)			
	102	WO	83/04461 A1	12-22-1983	Western Electric Company, Inc.	
	103	WO	92/20022 A1	11-12-1992	Digital Equipment Corporation	
	104	WO	93/01550 A1	01-21-1993	Infologic Software, Inc.	
	105	WO	93/11480 A1	06-10-1993	Intergraph Corporation	
	106	WO	94/03003 A1	02-03-1994	Crest Industries, Inc.	
	107	WO	96/24092 A2	08-08-1996	Benson	
	108	WO	96/27155 A2	09-06-1996	Electronic Publishing Resources, Inc.	
	109	WO	97/25800 A1	07-17-1997	Mytec Technologies Inc.	
	110	WO	97/37492 A1	10-09-1997	Macrovision Corporation	
	111	WO	97/41661 A2	11-06-1997	Motorola Inc.	
	112	WO	97/43761 A2	11-20-1997	Intertrust Technologies Corp.	
	113	WO	98/09209 A1	03-05-1998	Intertrust Technologies Corp.	
	114	WO	98/10561 A1	03-12-1998	Telefonaktiebolaget LM Ericsson	
	115	WO	98/11690 A1	03-19-1998	Glover	
	116	WO	98/19431 A1	05-07-1998	Qualcomm Incorporated	
	117	WO	98/43426 A1	10-01-1998	Canal+Societe Anonyme	
	118	WO	98/45768 A1	10-15-1998	Northern Telecom Limited	
	119	WO	99/24928 A2	05-20-1999	Intertrust Technologies Corp.	
	120	WO	99/34553 A1	07-08-1999	V-One Corporation	
	121	WO	99/35782 A1	07-15-1999	Cryptography Research, Inc.	
	122	WO	99/48296 A1	09-23-1999	Intertrust Technologies Corporation	
	123	WO	99/60461 A1	11-25-1999	International Business Machines Corporation	
	124	WO	99/60750 A2	11-25-1999	Nokia Networks Oy	
	125	WO	00/04727 A2	01-27-2000	Koninklijke Philips Electronics N.V.	
	126	WO	00/05898 A2	02-03-2000	Optivision, Inc.	
	127	WO	00/59152 A2	10-05-2000	Microsoft Corporation	
	128	WO	00/72118 A1	11-30-2000	Compaq Computers Inc.	
	129	WO	00/73922 A2	12-07-2000	Entera, Inc.	
	130	WO	01/37209 A1	05-25-2001	Teralogic, Inc.	
	131	EP	0 067 556 B1	12-22-1982	Data General Corporation	
	132	EP	0 257 585 A2	03-02-1988	NEC Corporation	

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 Applicant's unique citation designation number (optional). ² Applicant is to place a check mark here if English language Translation is attached.

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Complete if Known			
				Application Number		11/304,794	
				Filing Date		December 16, 2005	
				First Named Inventor		Mark Stefik et al.	
				Art Unit		3621	
				Examiner Name		Murdough, Joshua A.	
Sheet	5	of	9	Attorney Docket Number		111325/028700	

FOREIGN PATENT DOCUMENTS						
Examiner Initials ¹	Cite No. ¹	Foreign Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Country Code ²	Number ²	Kind Code ² (if known)		
	133	EP 0 332 304	A2	09-13-1989	Digital Equipment Corporation	
	134	EP 0 393 806	A2	10-24-1990	TRW Inc.	
	135	EP 0 450 841	A2	10-09-1991	GTE Laboratories Incorporated	
	136	EP 0 529 261	A2	03-03-1993	International Business Machines Corporation	
	137	EP 0 613 073	A1	08-31-1994	International Computers Limited	
	138	EP 0 678 836	A1	10-25-1995	Tandem Computers Incorporated	
	139	EP 0 679 977	A1	11-02-1995	International Business Machines Incorporated	
	140	EP 0 715 243	A1	06-05-1996	Xerox Corporation	
	141	EP 0 715 244	A1	06-05-1996	Xerox Corporation	
	142	EP 0 715 245	A1	06-05-1996	Xerox Corporation	
	143	EP 0 731 404	A1	09-11-1996	International Business Machines Corporation	
	144	EP 0 763 936	A2	03-19-1997	LG Electronics Inc.	
	145	EP 0 818 748	A2	01-14-1998	Murakoshi, Hiromasa	
	146	EP 0 840 194	A2	05-06-1998	Matsushita Electric Industrial Co., Ltd.	
	147	EP 0 892 521	A2	01-20-1999	Hewlett-Packard Company	
	148	GB 1483282		08-17-1977	Compagnie Internationale Pour L'Informatique C11-Honeywell-Bull	
	149	GB 2236604	A	04-10-1991	Sun Microsystems Inc.	
	150	GB 2309364	A	07-23-1997	Northern Telecom Limited	
	151	GB 2316503	A	02-25-1998	ICL Personal Systems Oy	
	152	BR 9810967	A (Abstract only)	10-30-2001	Scientific Atlanta Inc.	
	153	EP 0 934 765	A1	08-11-1999	Canal+Societe Anonyme	
	154	EP 0 946 022	A2	09-29-1999	Nippon Telegraph and Telephone Corporation	
	155	EP 0 964 572	A1	12-15-1999	Canal+Societe Anonyme	
	156	EP 1 103 922	A2 (Abstract only)	05-30-2001	CIT Alcatel	
	157	GB 2022969	A	12-19-1979	Data Recall Limited	
	158	GB 2354102	A	03-14-2001	Barron McCann Limited	
	159	JP 11031130	A2 (Abstract only)	02-02-1999	Fuji Xerox Co. Ltd.	

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 Applicant's unique citation designation number (optional). ² Applicant is to place a check mark here if English language Translation is attached.

10886767.1

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Complete if Known	
				Application Number	11/304,794
				Filing Date	December 16, 2005
				First Named Inventor	Mark Stefik et al.
				Art Unit	3621
				Examiner Name	Murdough, Joshua A.
Sheet	6	of	9	Attorney Docket Number	111325/028700

FOREIGN PATENT DOCUMENTS						
Examiner Initials*	Cite No. ¹	Foreign Patent Document		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		Country Code ²	Number ⁴ Kind Code ³ <i>(if known)</i>			
	160	JP	11032037 A2 (Abstract only)	02-02-1999	Fuji Xerox Co. Ltd.	
	161	JP	11205306 A2 (Abstract only)	07-30-1999	Fuji Xerox Co. Ltd.	
	162	JP	11215121 A2 (Abstract only)	08-06-1999	Fuji Xerox Co. Ltd.	
	163	JP	2000215165 A2 (Abstract only)	08-04-2000	Nippon Telegraph and Telephone	
	164	JP	2005218143 A2 (Abstract only)	08-11-2005	Scientific Atlanta Inc.	
	165	JP	2005253109 A2 (Abstract only)	09-15-2005	Scientific Atlanta Inc.	
	166	JP	2006180562 A2 (Abstract only)	07-06-2006	Intarsia Software LLC; Mitsubishi Corp.	
	167	JP	5168039 A2 (Abstract only)	07-02-1993	Sony Corp.	
	168	WO	96/13814 A1	05-09-1996	Vazvan	
	169	WO	00/46994 A1	08-10-2000	Canal+Societe Anonyme	
	170	WO	00/62260 A1 (Abstract only)	10-19-2000	Swisscom Mobile AG	
	171	WO	01/03044 A1	01-11-2001	Transcast International, Inc.	
	172	WO	04/103843 (Abstract only)	12/02/2004	S2F Flexico	
	173	WO	04/34223 A2	04-22-2004	Legal IGaming, Inc.	

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 Applicant's unique citation designation number (optional). ² Applicant is to place a check mark here if English language Translation is attached.

Substitute for form 1449A/PTO				Complete if Known	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Application Number	11/304,794
				Filing Date	December 16, 2005
				First Named Inventor	Mark Stefik et al.
				Art Unit	3621
				Examiner Name	Murdough, Joshua A.
Sheet	7	of	9	Attorney Docket Number	111325/028700

OTHER PRIOR ART – NON PATENT LITERATURE DOCUMENTS				
Examiner Initials*	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	
	174	BLAZE et al, "Divertible Protocols and Atomic Proxy Cryptography" 1998 Advances in Cryptography – Euro Crypt International Conference on the Theory and Application of Crypto Techniques, Springer Verlag, DE		
	175	BLAZE et al, "Atomic Proxy Cryptography" DRAFT (Online) (November 2, 1997) XP002239619 Retrieved from the Internet		
	176	NO AUTHOR, "Capability- and Object-Based Systems Concepts," Capability-Based Computer Systems, pp. 1-19 (no date)		
	177	COX, "Superdistribution" Wired Magazine (September 1994) XP002233405 URL:http://www.wired.com/wired/archive/2.09/superdis_pr.html&gt		
	178	DUNLOP et al, Telecommunications Engineering, pp. 346-352 (1984)		
	179	ELGAMAL, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," IEEE Transactions on Information Theory IT-31(4):469-472 (July 1985)		
	180	GHEORGHU et al, "Authorization for Metacomputing Applications" (no date)		
	181	IANNELLA, ed., Open Digital Rights Language (ODRL), pp. 1-31 (November 21, 2000)		
	182	KAHLE, wais.concepts.txt, Wide Area Information Server Concepts, Thinking Machines Version 4, Draft, pp. 1-18 (November 3, 1989)		
	183	KAHN, "Deposit, Registration and Recordation in an Electronic Copyright Management System," Technical Report, Corporation for National Research Initiatives, Reston, Virginia (August 1992) URL:http://www.cni.org/docs/ima.ip-workshop/kahn.html		
	184	KAHN et al, "The Digital Library Project, Volume 1: The World of Knowbots (DRAFT), An Open Architecture for a Digital Library System and a Plan for its Development," Corporation for National Research Initiatives, pp. 1-48 (March 1988)		
	185	KOHL et al, Network Working Group Request for Comments: 1510, pp. 1-112 (September 1993)		
	186	LEE et al, CDMA Systems Engineering Handbook (1998) [excerpts but not all pages numbered]		
	187	MAMBO et al, "Protection of Data and Delegated Keys in Digital Distribution," Information Security and Privacy. Second Australian Conference, ACISP '97 Proceedings, pp. 271-282 (Sydney, NSW, Australia, 7-9 July 1997, 1997 Berlin, Germany, Springer-Verlag, Germany), XP008016393 ISBN: 3-540-63232-8		
	188	MAMBO et al, "Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertexts," IEICE Trans. Fundamentals VOL. E80-A, NO. 1:54-63 (January 1997) XP00742245 ISSN: 0916-8508		
	189	Microsoft Word, Users Guide, Version 6.0, pp. 487-89, 549-55, 560-64, 572-75, 599-613, 616-31 (1993)		
	190	OJANPERA and PRASAD, eds., Wideband CDMA for Third Generation Mobile Communications (1998) [excerpts but not all pages numbered]		
	191	PERRITT, "Knowbots, Permissions Headers and Contract Law," Paper for the Conference on Technological Strategies for Protecting Intellectual Property in the Networked Multimedia Environment, pp. 1-22 (April 2-3, 1993 with revisions of April 30, 1993)		

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 Applicant's unique citation designation number (optional). ² Applicant is to place a check mark here if English language Translation is attached.

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>		Complete if Known			
		Application Number	11/304,794		
		Filing Date	December 16, 2005		
		First Named Inventor	Mark Stefik et al.		
		Art Unit	3621		
		Examiner Name	Murdough, Joshua A.		
Sheet	8	of	9	Attorney Docket Number	111325/028700

OTHER PRIOR ART – NON PATENT LITERATURE DOCUMENTS			
Examiner Initials ¹	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
	192	RAGGETT, (Hewlett Packard), "HTML+(Hypertext markup language)," pp. 1-31 (12 July 1993) URL:http://citeseer.ist.psu.edu/correct/340709	
	193	SAMUELSON et al, "Intellectual Property Rights for Digital Library and Hypertext Publishing Systems: An Analysis of Xanadu," Hypertext '91 Proceedings, pp. 39-50 (December 1991)	
	194	NO AUTHOR, "Softlock Services Introduces... Softlock Services" Press Release (January 28, 1994)	
	195	NO AUTHOR, "Appendix III - Compatibility with HTML," NO TITLE, pp. 30-31 (no date)	
	196	NO EDITOR, NO TITLE, Dictionary pages, pp. 469-72, 593-94 (no date)	
	197	BENOIT, Digital Television MPEG-1, MPEG-2 and Principles of the DVB System, pp. 75-80, 116-121 (no date)	
	198	BENOIT, Digital Television MPEG-1, MPEG-2 and Principles of the DVB System, 2 nd edition, pp. 74-80 (no date)	
	199	AH Digital Audio and Video Series, "DTV Receivers and Measurements," Understanding Digital Terrestrial Broadcasting, pp. 159-64 (no date)	
	200	O'DRISCOLL, The Essential Guide to Digital Set-Top Boxes and Interactive TV, pp. 6-24 (no date)	
	201	IUS MENTIS, "The ElGamal Public Key System," pp. 1-2 (October 1, 2005) online at http://www.iusmentis.com/technology/encryption/elgamal/	
	202	SCHNEIER, "Crypto Bibliography," Index of Crypto Papers Available Online, pp. 1-2 (online) (no date)	
	203	NO AUTHOR, NO TITLE, pp. 344-55 (no date)	
	204	NO AUTHOR, "Part Four Networks," NO TITLE, pp. 639-714 (no date)	
	205	Microsoft Word User's Guide, pp. 773-74, 315-16, 487-89, 561-64, 744, 624-33 (1993)	
	206	NO AUTHOR, "What is the ElGamal Cryptosystem," p. 1 (November 27, 2006) online at http://www.x5.net/faqs/crypto/q29.html	
	207	JOHNSON et al., "A Secure Distributed Capability Based System," ACM, pp. 392-402 (1985)	
	208	Wikipedia, "El Gamal Encryption," pp.1-3 (last modified November 2, 2006) online at http://en.wikipedia.org/wiki/ElGamal_encryption	
	209	BLAZE, "Atomic Proxy Cryptography," p. 1 Abstract (October 20, 1998)	
	210	BLAZE, "Matt Blaze's Technical Papers," pp. 1-6 (last updated August 6, 2006)]	
	211	Online Search Results for "inverted file", "inverted index" from www.techweb.com , www.cryer.co.uk , computing-dictionary.thefreedictionary.com , www.nist.gov , en.wikipedia.org , www.cni.org , www.tiscali.co.uk (July 15-16, 2006)	
	212	Corporation for National Research Initiatives, "Digital Object Architecture Project", http://www.nnri.reston.va.us/doa.html (updated 28 Nov 2006)	

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 Applicant's unique citation designation number (optional). ² Applicant is to place a check mark here if English language Translation is attached.

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.

Substitute for form 1449A/PTO INFORMATION DISCLOSURE STATEMENT BY APPLICANT <i>(use as many sheets as necessary)</i>				Complete if Known	
				Application Number	11/304,794
				Filing Date	December 16, 2005
				First Named Inventor	Mark Stefik et al.
				Art Unit	3621
				Examiner Name	Murdough, Joshua A.
Sheet	9	of	9	Attorney Docket Number	111325/028700

OTHER PRIOR ART – NON PATENT LITERATURE DOCUMENTS			
Examiner Initials*	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
	213	STEFIK, Summary and Analysis of A13 (Kahn, Robert E and Vinton G Cerf, "The Digital Library Project, Volume 1: The World of Knowbots (DRAFT), An Open Architecture for a Digital Library System and a Plan for its Development," Corporation for National Research Initiatives (March 1988)), pp. 1-25 (May 30, 2007)	

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 Applicant's unique citation designation number (optional). ² Applicant is to place a check mark here if English language Translation is attached.

10886767.1



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

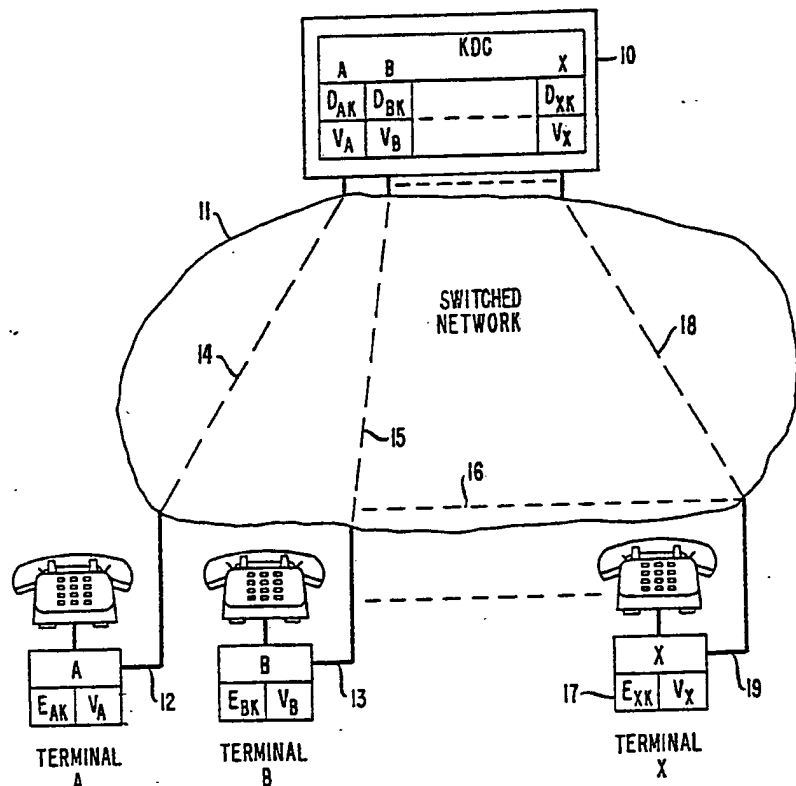
(51) International Patent Classification ³ : H04L 9/00		A1	(11) International Publication Number: WO 83/ 04461
			(43) International Publication Date: 22 December 1983 (22.12.83)
(21) International Application Number: PCT/US83/00030 (22) International Filing Date: 11 January 1983 (11.01.83) (31) Priority Application Number: 386,805 (32) Priority Date: 9 June 1982 (09.06.82) (33) Priority Country: US (71) Applicant: WESTERN ELECTRIC COMPANY, INC. [US/US]; 222 Broadway, New York, NY 10038 (US). (72) Inventors: EVERHART, Joseph, Robert ; P.O. Box 228, 3 Old Mill Road, Holmdel, NJ 07733 (US). OSBORN, Jeffrey, George ; 242 Madison Gardens, Old Bridge, NJ 08857 (US). (74) Agents: HIRSCH, A., E., Jr. et al.; Post Office Box 901, Princeton, NJ 08540 (US).		(81) Designated States: AT (European patent), AU, BE (European patent), CH (European patent), DE (European patent), FR (European patent), GB (European patent), JP, LU (European patent), NL (European patent), SE (European patent). Published With international search report.	

(54) Title: ENCRYPTION SYSTEM KEY DISTRIBUTION METHOD AND APPARATUS

KDC CONFIGURATION

(57) Abstract

Encryption systems typically rely on the distribution of cipher keys between terminals for scrambling and unscrambling transmitted messages. Elaborate security precautions are necessary to protect the cipher keys since a compromise of the key could result in a compromise of the transmission. There is disclosed a key distribution method and apparatus which uses a channel (14, 15, 18) from identified terminals (A, B, X) to a central key distribution center (KDC) for the establishment, on a one-session basis, of the key which is to be used for the next session between those terminals. The key establishing link (16) is itself encoded using a cipher key which changes after each usage. Provision is made to verify, for each new connection, that a compromise has not priorly occurred.



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	LI	Liechtenstein
AU	Australia	LK	Sri Lanka
BE	Belgium	LU	Luxembourg
BR	Brazil	MC	Monaco
CF	Central African Republic	MG	Madagascar
CG	Congo	MR	Mauritania
CH	Switzerland	MW	Malawi
CM	Cameroon	NL	Netherlands
DE	Germany, Federal Republic of	NO	Norway
DK	Denmark	RO	Romania
FI	Finland	SE	Sweden
FR	France	SN	Senegal
GA	Gabon	SU	Soviet Union
GB	United Kingdom	TD	Chad
HU	Hungary	TG	Togo
JP	Japan	US	United States of America
KP	Democratic People's Republic of Korea		

- 1 -

ENCIPHERMENT SYSTEM KEY DISTRIBUTION
METHOD AND APPARATUS

Background of the Invention

5 This invention relates to the establishment and distribution of cipher keys in a cryptographic system.

 Cryptographic systems are now gaining favor, both for voice as well as data transmission. In such systems it is typically necessary that the parties to a particular transmission each have cryptographic keys to encrypt and
10 decrypt the cipher transmissions. It follows that a compromise to a cryptographic key will in turn reduce the security of subsequent transmissions involving that key. Thus, great precautions must be taken to distribute the
15 cryptographic keys among the system users. Such distribution, for example, using secure couriers to manually update the keys may be possible when the community of users is priorly known but becomes increasingly more difficult when either the number of parties is large or
20 parties who seldom communicate with each other wish to do so. The responsibility for keeping the cryptographic key secure after distribution rests with each user and the longer the key remains effective the greater the risk of it becoming compromised.

25 Thus, from a practical point of view it is desirable to have the cryptographic key effective for a single session, requiring a new key for each new session. When couriers are used, however, this becomes costly and time consuming, especially when a party wishes to place
30 many secure calls or have many secure sessions.

 Attempts have been made to electronically distribute cryptographic keys between users from a key distribution center. One such example is shown in Rosenblum Patent No. 4,182,933, issued January 8, 1980.
35 While such attempts have found some degree of success they all suffer from the problem that they are subject to

- 2 -

compromise because they usually rely on the security of the transmission media between the key distribution center and the terminal for the distribution of session key information. Thus, an intruder need only compromise the key distribution channel to obtain subsequent session keys. Elaborate systems have sometimes been established to detect such a compromise, all of which are either costly or minimally effective.

Another problem with key distribution centers is that the center can derive the information used to decrypt the secure data exchange between users and thus could theoretically monitor the secure session transmission.

Summary of the Invention

We have solved the above-identified problems by arranging a key distribution center (KDC) which communicates over a channel with the individual terminals. The channel, or data link, can be a dial-up telephone line, a packet-switched data network, dedicated lines, or other communications channel types, over which secure communication is possible. The terminals operate in conjunction with the KDC to establish a session key for secure transmission between two or more terminals. The session key at a terminal is constructed from information generated at that terminal in conjunction with information communicated from the KDC and is known fully only to the terminals involved in the session and not to the KDC. Thus, when two terminals have established a session key, they may securely communicate with each other for the duration of that session.

At the conclusions of the secure data exchange, the session keys should be destroyed, and when either station wishes to establish additional secure communication either between themselves or to other stations, a new session key will be established in cooperation with the KDC.

Both the terminal-KDC channel and the KDC-terminal channel, as mentioned above, are secure links in

SUBSTITUTE SHEET

- 3 -

that they are protected by cryptographic key information which is unique to each terminal and to the KDC on a one-call-only basis. Accordingly, whenever a connection is established between a terminal and the KDC, each has

5 information previously stored, referred to as terminal-unique key information, and this priorly stored information is used to establish both new KDC-terminal link keys, referred to as call-setup key information, and new session

10 keys, the terminal and the KDC each modify their respective terminal-unique key information so that on a next call between the KDC and the same terminal, this new key information must be used in order to establish a secure communication path. The precise manner in which this

15 happens will be discussed hereinafter. In this manner, an intruder on the key distribution between a terminal and the KDC must be adding and substituting information on the channel from the beginning and must stay on the channel throughout several calls, since once the intruder leaves it

20 is possible to detect, at least by hindsight, that a compromise has occurred. This is a result of the fact that the intruder is substituting random information that may be monitored.

One aspect of our system is that an intruder, in

25 order to obtain useful information exchanged between two valid users of the system, must gain the terminal-unique information that is stored at the terminal, and he must also gain the terminal-unique information that is stored in the key distribution center for that specific terminal.

30 The intruder then, on the very next key exchange involving that terminal and the key distributing center, must actively participate, i.e., substitute his own generated key information on that channel. Then the intruder must also substitute information on the channel between the two

35 communicating terminals, and also must continue the above substitutions on the channels for an indefinite period of time or risk detection.

SUBSTITUTE SHEET

Petitioner Apple Inc. - Exhibit 1004, p. 513



- 4 -

Brief Description of the Drawing

These attributes of our invention, together with the operation and utilization of the invention in a specific embodiment, will be more fully apparent from the illustrative embodiment shown in conjunction with the drawing which:

FIG. 1 shows an overall system using a KDC and several terminals;

FIG. 2 shows an implementation of the initial establishment of information in both the KDC and the terminal within a secure area;

FIGS. 3 and 4 show a flow chart detailing what occurs within each terminal;

FIG. 5 shows a flow chart detailing what occurs within the KDC;

FIGS. 6-19 show, in sequence, an implementation of the establishment of key information and control data within each terminal; and

FIGS. 21-28 show, in sequence, an implementation of the establishment of key information and control data within the KDC. In this system we have a variety of terminals.

General Description

FIG. 1 shows a number of terminals, A, B and X, connectable to each other and to KDC 10 via some transport network (e.g., public switched network). These terminals should be able to set up a secure channel between themselves in order to exchange secure information. In this process they must both communicate with the KDC. The transmission line 12 from terminal A is connected through link 16 to transmission line 13 to initiate a secure call to terminal B. Once the users decide to initiate a secure data exchange, each terminal sets up a transmission line, such as link 14 for terminal A, to the KDC.

An exchange of information will then occur from terminal A to the KDC and from terminal B to the KDC. Once the KDC has received both of these messages, it will

SUBSTITUTE SHEET

Petitioner Apple Inc. - Exhibit 1004, p. 516

BUREAU
OMPI
WIPO

- 5 -

formulate two distinct messages that will be sent respectively to terminal A via link 14 and to terminal B via link 15. These individual messages will contain session key information, as well as other pertinent information described below. This session key information has originated at terminal A and at terminal B and is exchanged through the KDC. Once the exchange has taken place between the two terminals and the KDC, link 14, which is the key distribution link between terminal A and the KDC, is then taken down, and key distribution link 15 between the KDC and terminal B is taken down. Link 16, which is the session link between terminals A and B, is re-established. Further key information is exchanged based on the prior partial exchanges so as to derive independently at both terminals the session key, and finally using that session key information, data (i.e., digital data or digital voice) can be transmitted in secure fashion on data link 16.

Since further session information was derived between terminals A and B independent of the KDC, a malicious operator of the KDC cannot derive the key information need to decrypt the secure messages sent between terminals A and B without actively substituting information on the session channel.

Also, at this point, as will be seen, contained within the messages that were sent between the KDC and the terminals was new terminal-unique key information to secure the next key distribution between the terminals and the KDC. This new information is independent of the previous information and therefore is unique to it.

Detailed Description

Turning now to FIG. 2 the initial setup between the terminal and the KDC must be made in an authentic manner such that the information transported to the terminals from the KDC is not modified. One implementation is where the transport is made within a secured area, such as secured area 23. Since subsequent communications

SUBSTITUTE SHEET



- 6 -

between the KDC and each terminal depend upon the prior communication, it is important that at some period in time they both contain the proper information for start-up, and ideally this is done in the secured area so that there can
5 be no breach of security.

On the initial system setup (based on the secured area implementation shown in FIG. 2) the terminals are brought within the secured area 23, and the KDC can generate terminal-unique key pairs for each terminal. The
10 exact function of these key pairs will be described later. The KDC will generate a terminal-unique decryption key for each terminal and the corresponding encryption key. This encryption key must be placed in the terminal-unique key storage for each terminal with the corresponding decryption
15 key stored in the terminal-unique key storage at the KDC under the address of that terminal. In addition, a random number, U_a for terminal A, unique to each terminal is stored in the verification information storage at the KDC also at the address of this terminal. This same random
20 number must be loaded and stored in the verification information storage in the terminals and will be used for a verification check on the first call setup to the KDC.

FIGS. 3 and 4 are flow charts representing the action that occurs within a terminal, for example,
25 terminal A.

FIG. 5 is a flow chart representing what actions occur within the key distribution center.

The discussion which will follow is a discussion with respect to a time sequence between the terminal and
30 the KDC to illustrate both how terminal-unique keys are updated, and how call-setup and session keys are distributed. This discussion will occur with respect to FIGS. 6 through 28. FIGS. 6 through 19 show the apparatus within the terminal and show on a step-by-step basis how
35 the call-setup keys and the session keys are established. FIGS. 20 through 28 show the apparatus within the KDC, each figure showing a specific operational aspect of the

SUBSTITUTE SHEET

- 7 -

establishment of the keys.

Turning now to FIG. 6, we will discuss the specific apparatus used in the terminals. The actual generation of the numbers will be discussed hereinafter.

- 5 Apparatus 72 is a random number generator which is a device or algorithm that produces bits (zeros and ones) that are equally likely to occur. This generation may be based upon a noisy diode and any number of algorithms can be used to attain statistically independent output of 0's and 1's.
- 10 The more equally likely these random number generators are, i.e., the more random this function is, the higher the security level will be. The output of the random number generator is a serial stream of zeroes and ones where the correlation between one or a group of bits is zero. The
- 15 bidirectional asymmetric key generator, apparatus 73, takes as input a random number from random number generator 72 and will compute an encryption key and the matching decryption key such that the encryption key cannot be derived from the decryption key and vice versa. The
- 20 generation of these keys as an example could be done in accordance with the RSA algorithm, as described by Rivest, Shamir, and Adleman in a paper entitled, "A Method for Obtaining Digital Signatures and Public Key Crypto Systems," which publication is hereby incorporated by
- 25 reference, which appeared in CACM, Vol. 21, No. 2, February, 1978, on pages 120-126.

Apparatus 74 implements a bidirectional asymmetric cryptographic algorithm (e.g., the RSA algorithm) that is, a cryptographic algorithm based on two

30 distinct keys where the encryption key cannot be derived from the decryption key and vice versa. Apparatus 74 has two inputs (I and K) and one output (O). The input I is the bits to be encrypted or decrypted. The input K is the key, either encryption or decryption (the RSA algorithm

35 performs the same function regardless of encryption or decryption). The output will be the inputted bits encrypted or decrypted with the supplied key. This

SUBSTITUTE SHEET



- 8 -

algorithm is also described in the aforementioned paper. Functionally, apparatus 75 is the embodiment of two functions f and g such that: given $f(R, P)$ and P , one cannot determine R ; $g(R1, f(R2, P), P) = g(R2, f(R1, P), P)$; and given $f(R1, P)$, $f(R2, P)$, and P one cannot determine $R1$, $R2$, or $g(R1, f(R2, P), P)$.

Apparatus 75 performs the above functions via, for example, the Diffie-Hellman algorithm, which is described in a paper by Diffie and Hellman entitled "New
10 Directions in Cryptography," published by the IEEE Transactions on Information Theory, Vol. IP-22, November, 1976, on pages 644-655, which is hereby incorporated by reference. The input to this algorithm is a base Y , a modulus Q and an exponent EXP . The output is Y raised to
15 the EXP power modulus the Q . The functions f and g are the same as discussed above in this example.

The storage requirements are depicted by registers 71, 70 and 76. These are the semi-permanent register 71 which contains both the verification
20 information Va and the terminal-unique key information Eak used to encrypt messages to the KDC. Temporary register 70 can be in any state initially and is used during the interaction with the KDC on a secure call setup. The address register permanently contains the address (i.e., a
25 public piece of information that uniquely identifies A to the KDC) of the terminal (terminal A in this case) where it is located. During a secure session (or call) setup, the address register will also contain the address of the terminal which is being called. The registers containing
30 verification information and encryption and decryption information may vary in size depending upon the specific algorithm used but in this example should be on the order of 1,000 bits each. Information pertaining to the symmetric session key and the random number should be on
35 the order of 100 bits, and the address information will be dependent upon a terminal numbering plan both unique and known to the KDC. For example, it could be the telephone

SUBSTITUTE SHEET

BUREAU
OMPI
WIPO

- 9 -

number of the specific terminal or it could be the serial number of the terminal.

Turning to FIG. 20, we will now discuss the working of the modules within the key distribution unit.

5 The address register at the KDC, register 200, performs the same function as the address register at the terminal. The RSA function at the KDC, apparatus 210, performs the same function as the RSA function at the terminal, as previously described. The random number generator, apparatus 211,
10 performs the same function as the random number generator at the terminal previously mentioned. The generator of the encryption and decryption keys apparatus 212 has the same function as described previously in the terminal. Apparatus 213 is a generator of the parameters used as
15 inputs to the apparatus 75 described previously. For this particular example these parameters are the base and modulus for the Diffie-Hellman algorithm. It requires as input the output of the random number generator, apparatus 211. The method of generation is described in
20 the aforementioned paper by Diffie.

There is a semi-permanent storage at the KDC, registers 214 and 216, which stores verification information Va and terminal-unique decryption key information Dak between calls. Semi-permanent
25 registers 215 and 217 are used to store information during the call setup progress. These registers have the same functions as described previously for the terminal.

System Operation

The operation of the system will now be explained
30 beginning with FIG. 3. Initially the key management equipment in the terminal will be in the wait state until a request is received from the terminal controller processor to initiate a secure call. At this point, as discussed, there is stored in the terminal the terminal-unique
35 encryption key that will be used to encrypt information that is sent to the KDC. Also stored is the verification information. These two pieces of information were stored

SUBSTITUTE SHEET



- 10 -

from the last call (or from the initial setup) that was made by this terminal. This is shown in FIG. 6 as Va and Eak.

Once a request is received to initiate a secure call, the address of the called party must be given to the key management equipment via the controller processor. This is seen in FIG. 3, box 31. At this point, there are generated new call-setup keys. This is shown in box 32 and in FIG. 7 as Eka and Dka. In box 33 there is shown the generation of partial session keys that will be used to encrypt data on the link from terminal B to terminal A. This is shown in FIG. 8 as Eba and DbA.

At this point, the verification information is updated using the keys that were just generated. The update function is specified as follows:

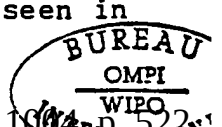
$$Val' = f(Val, E1) \text{ and } Va2' = f(Va2, E2)$$

where ' denotes updated and $ValVa2 = Va$. Va is the stored verification information and the E's are the just-generated encryption keys. The properties of f are as follows:

- (1) for every V, E1, E2: $f(V, E1) \neq f(V, E2)$ where $E1 \neq E2$;
- (2) for every V1, V2, E: $f(V1, E) \neq f(V2, E)$ where $V1 \neq V2$;
- (3) given V and $V' \neq f(V, E)$ it is difficult to determine E; and
- (4) in the case where E is an asymmetric encryption key, D cannot be determined from E.

For this example, $Va' = Val'|Va2'$ where $Va = Val|Va2$, Val' is equal to Val encrypted with Eka, and Va2' is equal to Va2 encrypted with Eba. This update process is depicted in FIG. 9. The first half of the verification information Val is read from storage and provided as an input to the RSA algorithm. The key that is used to encrypt this information is the call-setup key, Eka, that was just generated. This becomes Val' and overwrites Val as seen in

SUBSTITUTE SHEET



- 11 -

FIG. 10. Next, the second half of the verification information Va2 is encrypted using Eba just generated. The result Va2' overwrites Va2 in the storage register. This is shown in FIG. 3, box 34, and in summary, the updated verification information Va" is the verification information stored from the previous call, or given to the terminal on the initial setup from the KDC, where half is encrypted using the encryption part of the partial session key generated on this call and the other half is encrypted using the call-setup key for that call.

At this point, as shown in box 36, FIG. 3, and in FIG. 11, the message can be formatted to the KDC. The contents of this message are the encryption parts of the two keys that were just generated. Both the partial session key to be established between terminal A and B, Eba, and the new call-setup key Eka are encrypted using the terminal-unique encryption key Eak stored from the previous call from the KDC to the terminal or given to the terminal on the initial setup. At this point, the information that can be destroyed from the terminal is the terminal-unique encryption key, Eak, stored at the terminal from the previous call, and both the call-setup encryption key, Eka, and the partial session encryption key, Eba, that were generated by the terminal. The encrypted message is then appended to the address, A, of the originating terminal followed by the address, B, of the called terminal. This message is now sent to the KDC.

The terminal now will enter a wait state waiting for the information to be received from the KDC. This is depicted in box 37 of FIG. 3.

As shown in FIG. 5, the KDC will be in a wait state until a message is received from terminal A. This is shown in FIG. 5, box 50. Once the message is received, the KDC reads the address information within the message into the address register which gives it the index of the decryption key that must be used to decrypt the message. The KDC has in its storage from the previous call the

SUBSTITUTE SHEET

- 12 -

matching verification information for each terminal and the terminal-unique decryption key for each terminal. This is depicted in FIG. 20, boxes 214 and 216.

5 The message from terminal A is decrypted using the terminal-unique decryption key corresponding to that terminal, Dak. The keys, both the new call setup key Eka and the partial session key Eba (to be distributed to terminal B) is temporarily stored in the KDC memory as depicted in FIG. 21.

10 At this point, as shown in FIG. 22, the KDC can update its verification information in the exact same manner as the terminal. This is done by encrypting each half of the stored verification information Va with the received session key information Eba and the received
15 call-setup key information Eka, shown in FIG. 23. This produces the update verification information Va*.

The key distribution center, as shown in FIG. 24, will now generate a bidirectional asymmetric encryption/decryption key pair, Eak', Dak'. The primes
20 denote updated information. Eak' will be distributed to terminal A to be used on the next call setup to the key distribution center. The decryption key Dak' overwrites the decryption key Dak that was stored from the previous call.

25 Two other pieces of information are also generated at this time. These are the parameters that will be used by the terminals to create symmetric session keys; in this case they are the parameters of the Diffie-Hellman algorithm. One is the base Y and the other is the
30 modulus Q as previously described. Functionally, the amount of information that is generated at the KDC and sent to each terminal may vary depending upon the precise algorithm. This information is stored in temporary storage and will be used as part of the message sent back to both
35 terminal A and terminal B. This generation process is depicted in FIG. 25 and refers to the flow chart box 55, FIG. 5. By this point, as shown in FIG. 26, the KDC must

SUBSTITUTE SHEET

Petitioner Apple Inc. - Exhibit 1004, p. 524

BUREAU
OMPI

- 13 -

have received a message from terminal B in order to complete the call to terminal A. If not, the KDC process for terminal A must wait until the process for terminal B has reached this point. This is so it can give terminal A
5 the partial session key information Eab generated at terminal B and also to be able to give terminal B the partial session key Eba generated at terminal A. Coordination between the processes must take place so that the same parameters generated by one process overwrites the
10 parameters generated by the other process. This insures that the parameters sent to the terminals for the purpose of generating symmetric session keys are the same.

Once the internal exchange is made between the A registers and the B registers to coordinate the information
15 inside the key distribution center, the messages can now be formatted for the terminals. This is shown in FIG. 27. The message to terminal A will consist of the new terminal-unique key information Eak' that will be used on a subsequent call to the KDC. It will also consist of the
20 partial session key information Eab which it received from terminal B. It will also consist of the verification information Va" or a known reduction of Va" in terms of the number of bits. It will also consist of the base Y and the modulus Q of the Diffie-Hellman algorithm. These five
25 pieces of information will be encrypted using the call-setup key Eka received in the message from terminal A. The KDC destroys Eka, Eba, Eak', Y, and Q corresponding to terminal A and destroys Ekb, Eab, Ebk', Y, and Q corresponding to terminal B. The KDC will then send this
30 output message back to terminal A. An analogous encrypted message is sent from the KDC to terminal B. At this point the KDC is finished with its processing.

FIG. 28 shows the configuration of the KDC after the call to terminal A has been dropped. The KDC has
35 updated verification information Va" and updated terminal-unique decrypt key information Dak' which will be used on a subsequent call between terminal A and the KDC.

SUBSTITUTE SHEET

- 14 -

Referring back to the flow chart, FIG. 3, for terminal A, the key management equipment at the terminal has been in a wait state while the KDC has been functioning. FIG. 12 shows the key information stored at the terminal during this wait state. It is the updated verification Va" information and both decrypt keys Dka and Dbk corresponding to the previously generated encryption keys.

FIG. 13 shows how the information received from the KDC is used in accordance with the box 38, FIG. 3. The call-setup decryption key Dka is used to decrypt the message received from the KDC. The five values (previously discussed) sent from the KDC are now used in the following way. The first piece of information is the new distribution key Eak' that is stored in the semi-permanent register 71 and will be used on a following call made from this terminal to the KDC. It is the updated terminal-unique encryption key. The second piece of information is the partial session key Eab which was generated at B and sent through the KDC to terminal A. The third piece of information is the updated verification information Va", which can now be compared with the verification information stored at terminal A. The fourth and fifth pieces of information are the parameters to the Diffie-Hellman algorithm, the base Y and the modulus Q, which terminal A stores in temporary storage.

Referring to FIG. 4, box 40, at this point the terminal will compare the verification information it received from the KDC and either the verification information which is presently stored or some known reduction of that verification information - FIG. 14. If this matches, then the process will continue as normal. If this does not match, an alarm could be given to the terminal controller processor of a potential intruder threat on a previous call.

Assuming a success of the compared verification, the terminal can now take down the channel to the KDC and

SUBSTITUTE SHEET

Petitioner Apple Inc. - Exhibit 1004, p. 526

BUREAU
OMPI

- 15 -

establish a channel to terminal B, if not already established. At this point, terminal A and terminal B can communicate data securely using the asymmetric session keys Eab and Eba. If a symmetric session key is needed, the following steps can be taken. The calculation of the message to be sent to terminal B is shown in FIG. 15. First, the base Y and modulus Q of the Diffie-Hellman algorithm are used along with a random number Ra generated by the random number generator 72. These inputs are given to the Diffie-Hellman algorithm 75 and the output is then an input to the RSA function 73. The random number Ra is also stored in temporary storage. Eab is used as the key to the RSA function 73. At this point the session key information Eab received from terminal B and the base number Y may be destroyed. The output of the RSA algorithm is sent to terminal B.

Terminal A' key management equipment will now enter a wait state shown in FIG. 4, box 44, waiting for a message to be returned from terminal B. The idle state is depicted in FIG. 16 and in storage is the decrypt session key Dab which terminal A generated, the modulus Q of the Diffie-Hellman algorithm generated by the KDC and the random Ra number that was generated by terminal A.

As shown in FIG. 17, upon receipt of the message from terminal B, terminal A will decrypt the message using its decryption key Dba stored from the initial generation of the partial session key. Dba can now be destroyed. The output of this will be fed into the Diffie-Hellman algorithm as the base. The exponent will be the random number Ra which was priorly generated and the modulus Q is also input into the algorithm. The output of the Diffie-Hellman algorithm will be symmetric session key information which will equal the session key information that terminal B has calculated. Q and Ra can now be destroyed.

At this point, terminals A and B have established symmetric session key information between themselves that is not derivable by the KDC. This key information may be

SUBSTITUTE SHEET

- 16 -

used in a symmetric key algorithm like the Data Encryption Standard (DES) to encrypt data. What is stored now in the terminal until the next request for a secure session (or call), as shown in FIG. 18, is the updated verification
5 information V_a'' and the terminal-unique key E_{ak}' which it received from the KDC to be used to encrypt the next message to the KDC.

It should be noted that the actual generation of the desired data at the terminal and at the KDC is
10 operative under control of a computer processor and is programmed in accordance with the flow charts shown in FIGS. 3-5 to perform the sequence of data transfers detailed herein. Such a processor, while not shown, can be any one of several well-known microprocessors, such as for
15 example, the Intel 8086 microprocessor, working in conjunction with the terminal and KDC apparatus shown and detailed herein above.

It should also be noted that one skilled in the art could use different encryption algorithms and different
20 equipments to achieve the same results disclosed herein without departing from the spirit and scope of our invention.

SUBSTITUTE SHEET



- 17 -

Claims

1. A key distribution method for communicating cipher keys between two terminals via a key distribution center, KDC, said method comprising
- 5 establishing between any one terminal and said key distribution center a terminal-unique cipher key, cooperating between said KDC and said one terminal on a subsequent connection between said KDC and said one terminal to establish a session key for use by said one
- 10 terminal in a subsequent secure transmission between said one terminal and a second terminal, and changing in response to said subsequent connection between said one terminal and said KDC said priorly established terminal-unique cipher key.
- 15 2. The invention set forth in claim 1 wherein said session key is generated from the asymmetric exchange of information between said one terminal and said KDC plus the subsequent exchange of information between said first and second terminals.
- 20 3. The invention set forth in claim 2 wherein said session key at said one terminal is random with respect to information at said KDC.
4. The invention set forth in claim 2 wherein said session key at said one terminal is underivable with
- 25 respect to any information at said KDC.
5. A key distribution center for controlling the dissemination of session cipher keys between remotely located terminals, said center arranged for switched access to a plurality of said terminals, said center comprising
- 30 means for establishing communication cipher keys between said center and each said terminal having access thereto, each cipher key unique to each said terminal, means operative when one of said terminals accesses said center for bidirectional asymmetrically
- 35 exchanging information with said accessed terminal using, as a foundation for said exchange, said priorly established communication cipher keys, and

SUBSTITUTE SHEET



- 18 -

means responsive to said exchanged information for communicating to said terminal information allowing said terminal to establish a session cipher key for use with an identified other terminal also having access to said center.

6. The invention set forth in claim 5 wherein said key distribution center further comprising means for changing said established communication cipher keys as a result of said exchanged information.

7. The invention set forth in claim 5 wherein said cipher key establishing means uses information from a prior transmission from a particular terminal for establishing said cipher keys to said particular terminal.

8. The invention set forth in claim 5 wherein said exchanged information includes information generated in part at said center for the random generation of said session key allowing said session key to be underivable with respect to any information at said center.

9. A key distribution center for controlling the distribution of cipher control information among a number of terminals, said center comprising

means for individually exchanging encoded information between any of said terminals, said exchange for any particular terminal based partially upon a last information exchange between said particular terminal and said center,

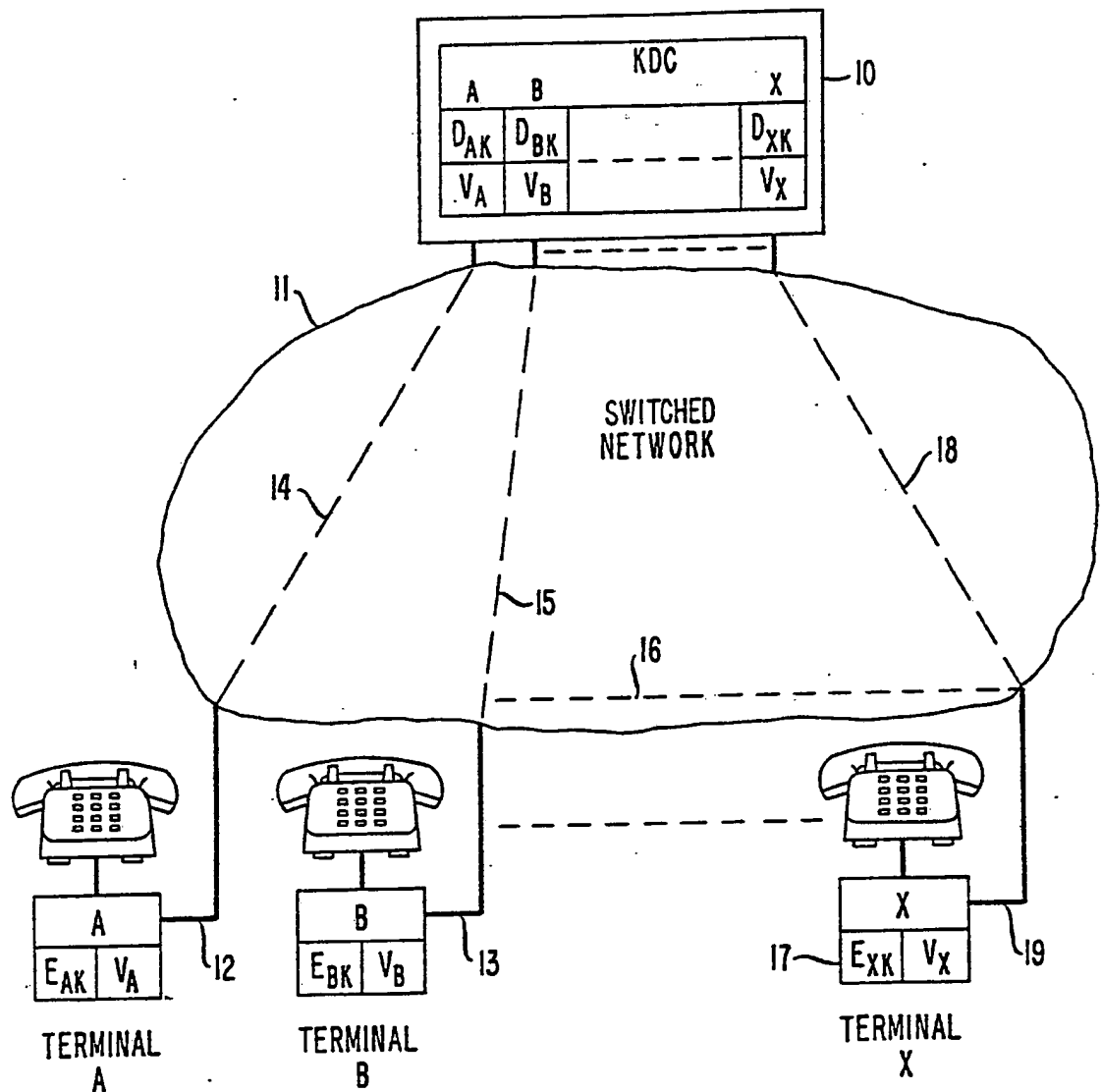
means for identifying at least two terminals where encrypted session information is to be exchanged and for accepting from said identified terminals certain encryption control information, and

means for modifying, according to a pre-established pattern, accepted information from said identified terminals and for communicating said modified information to the other of said terminals so as to allow each of said terminals to thereafter establish, independent of any information available at said center, a cipher key allowing said session information to be encrypted.

SUBSTITUTE SHEET

1/17

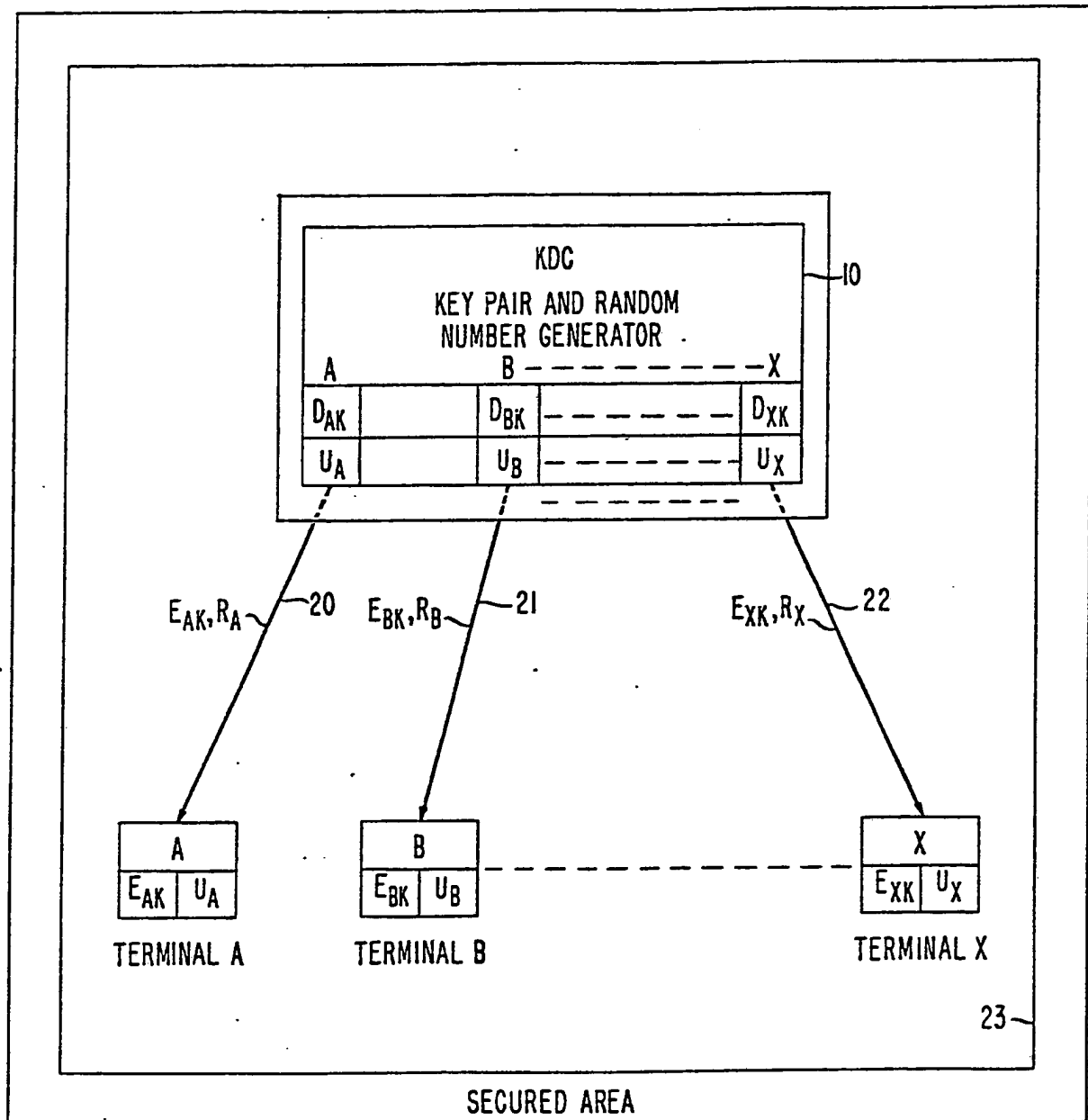
FIG. 1
KDC CONFIGURATION



2/17

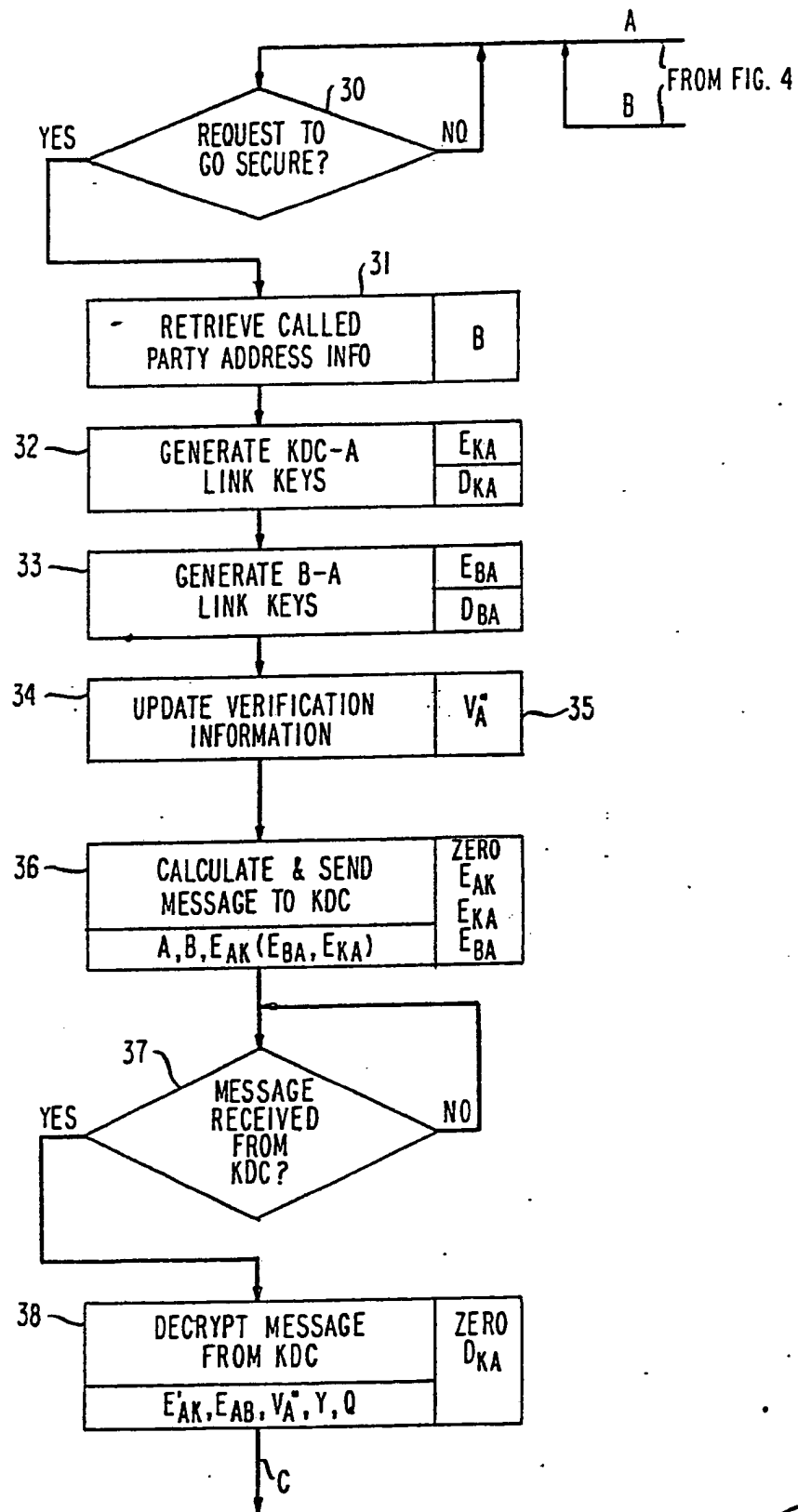
FIG. 2

INITIAL SYSTEM SETUP



3/17

FIG. 3
TERMINAL A



4/17

FIG. 4
TERMINAL A

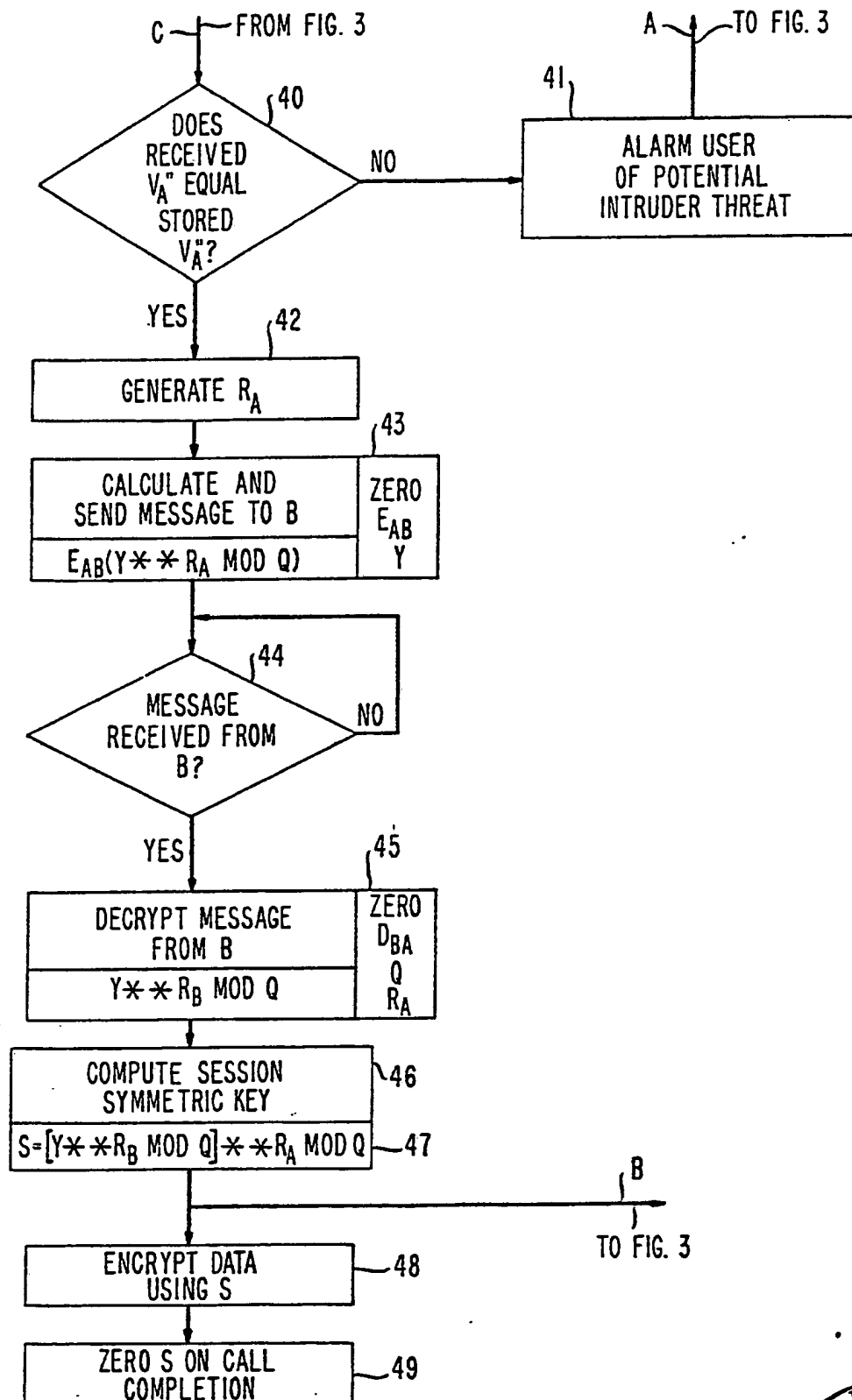
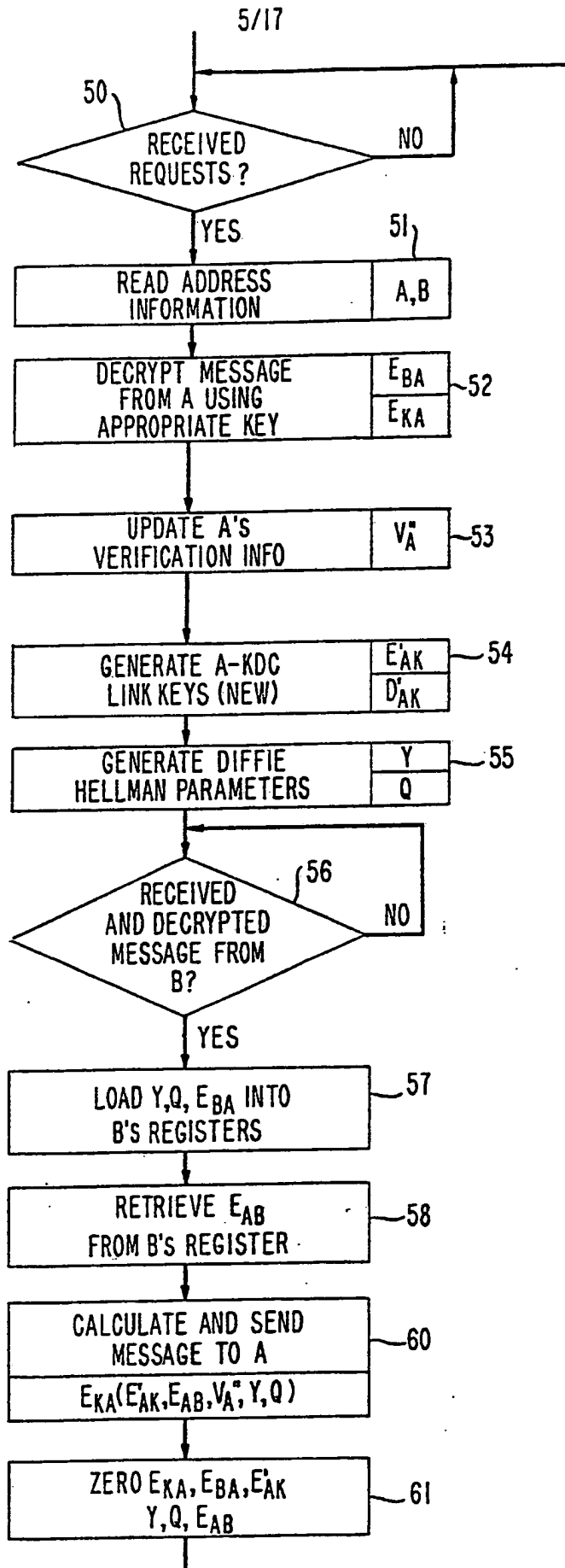
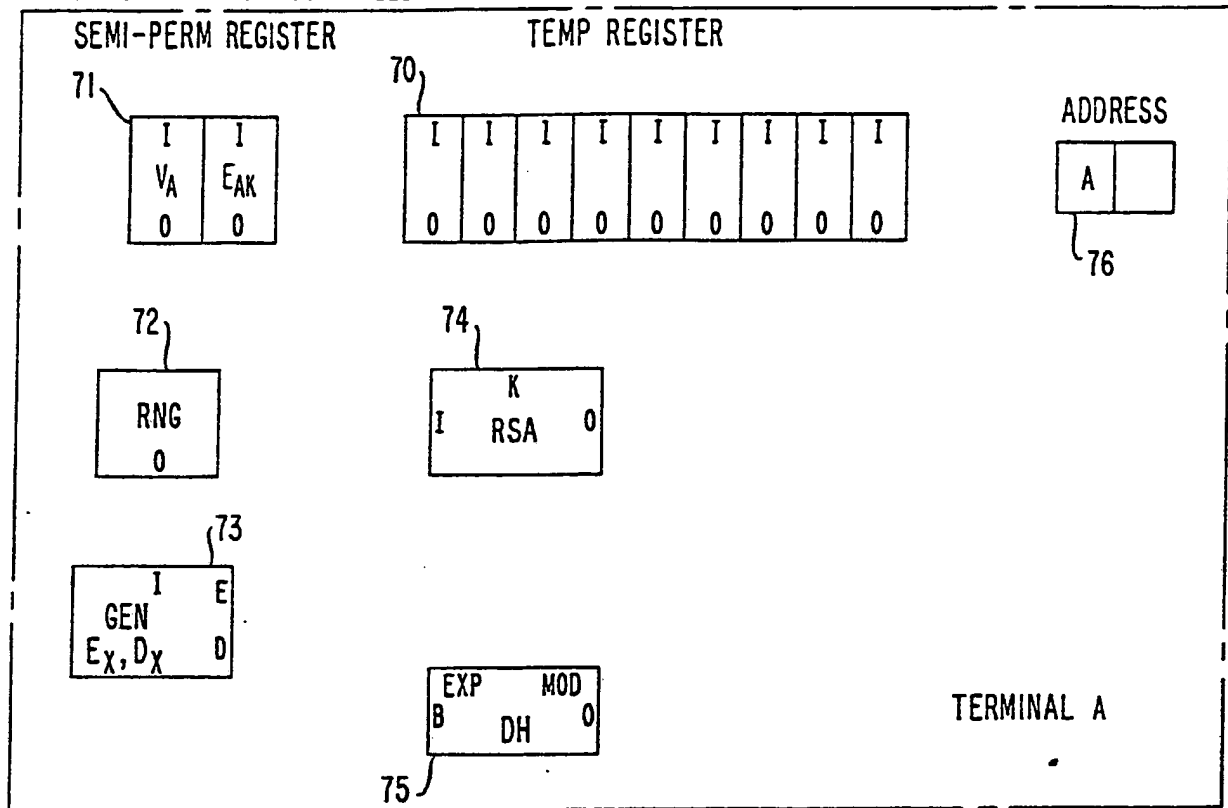
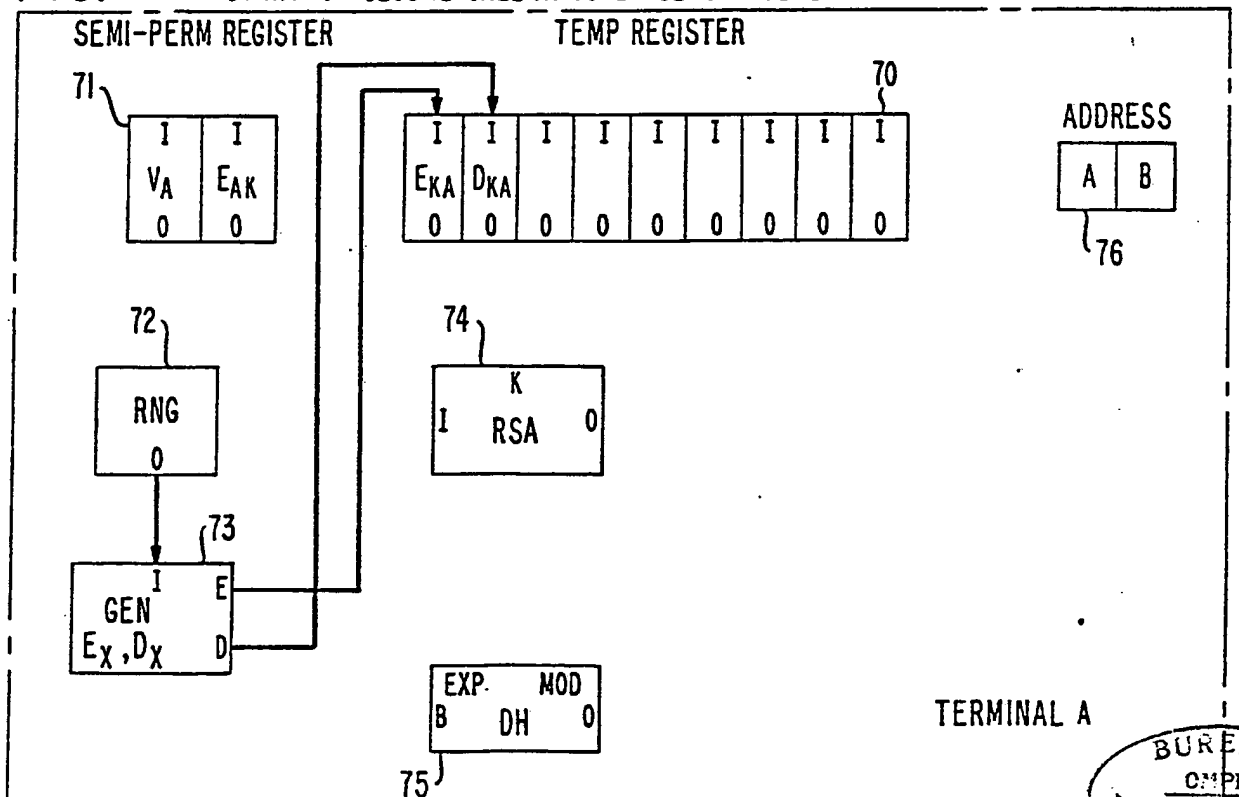


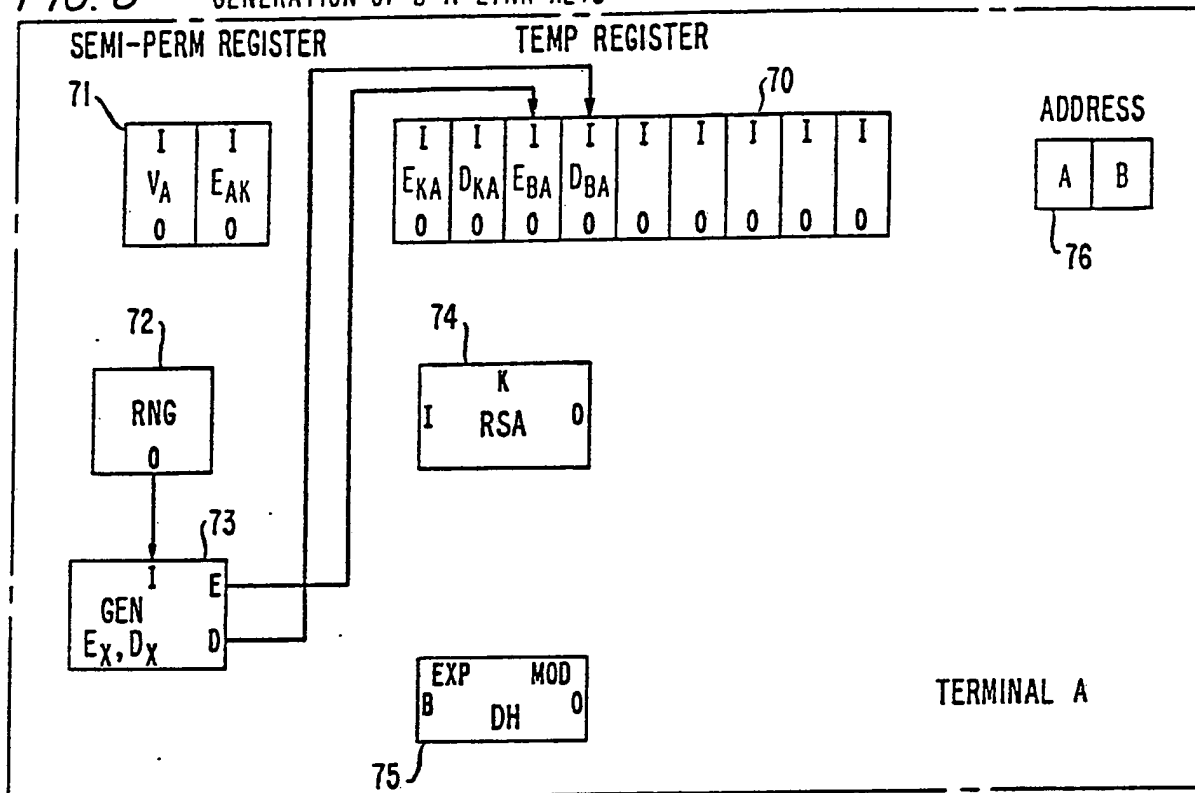
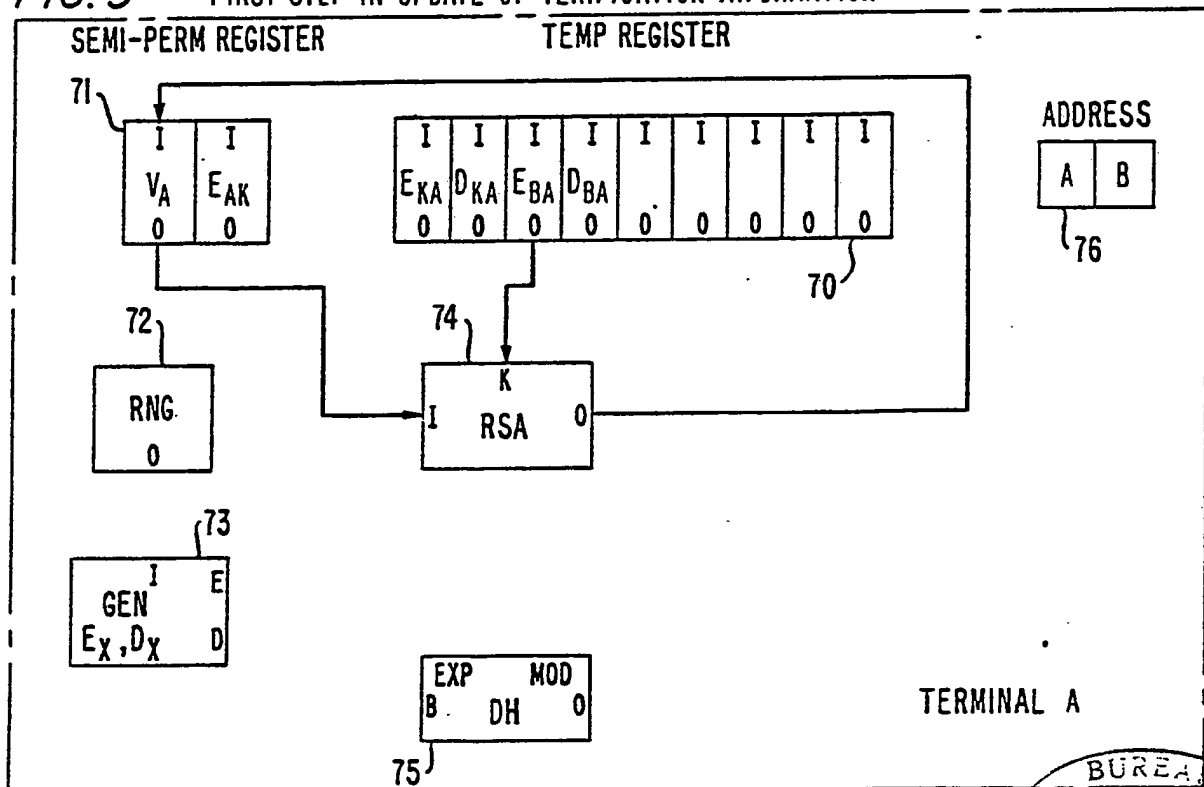
FIG. 5
KDC



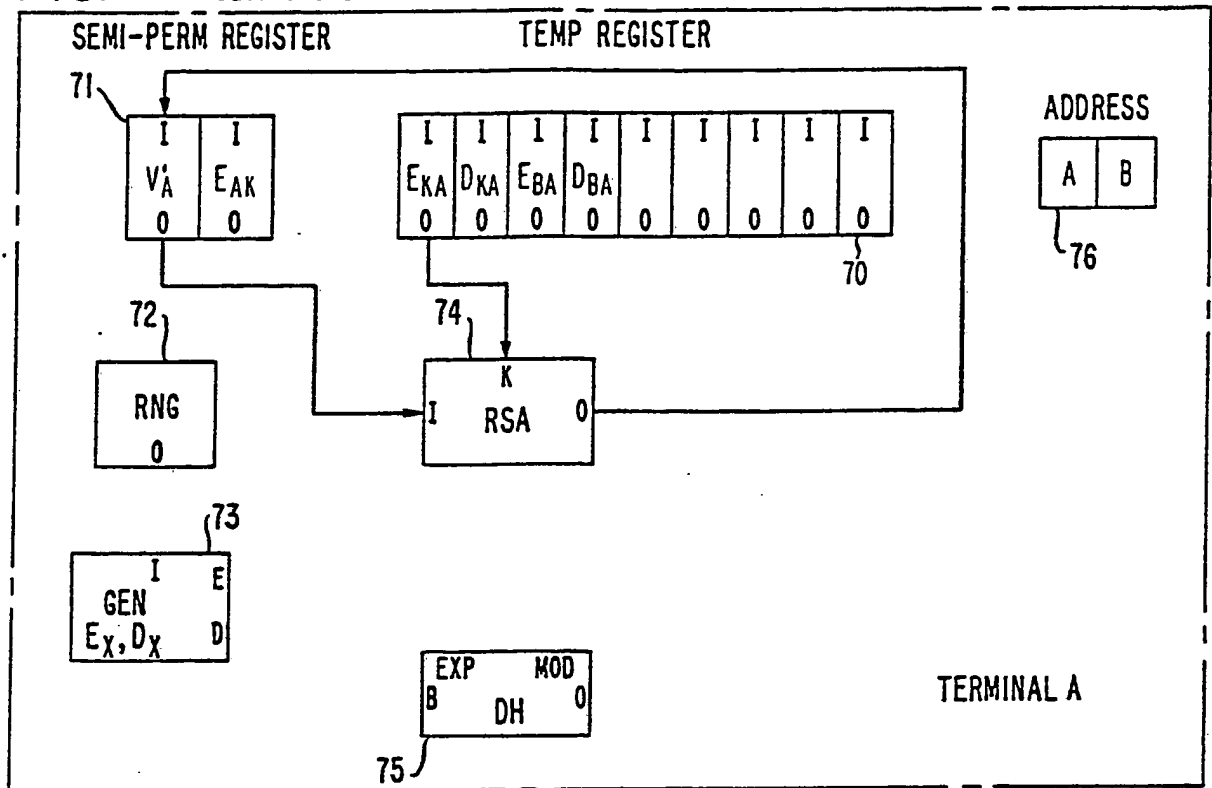
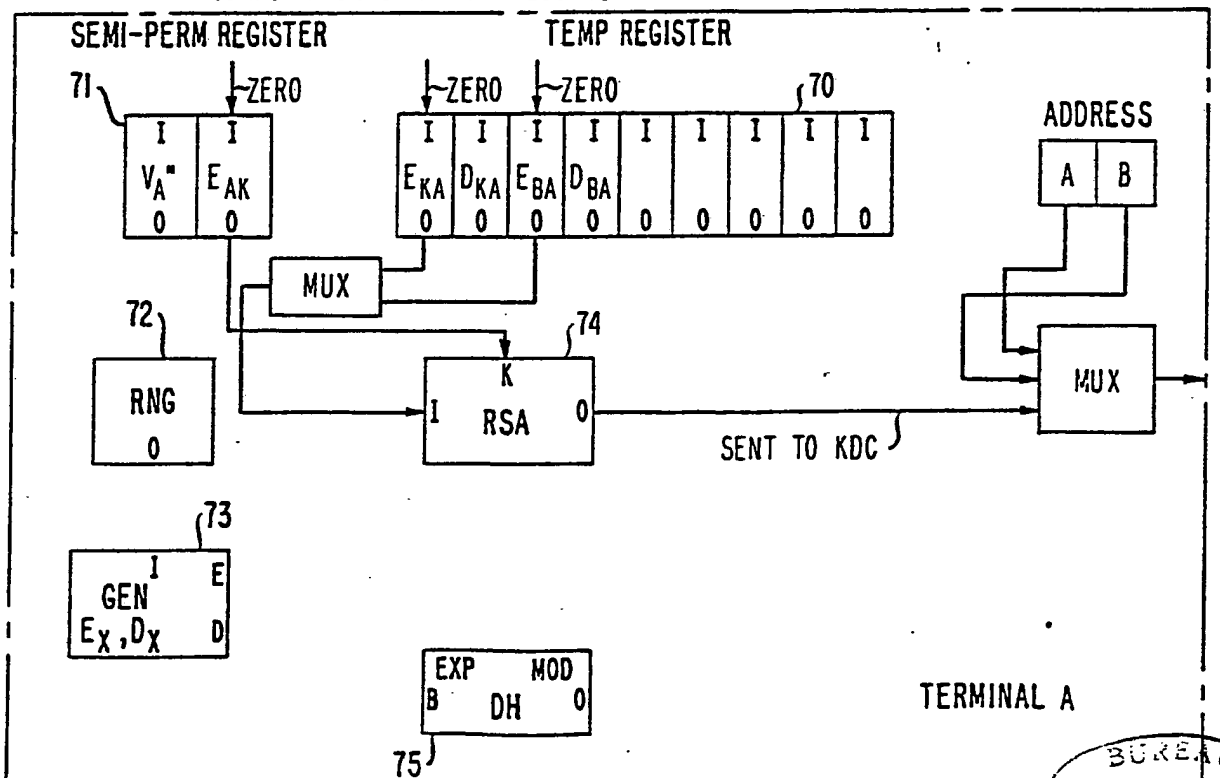
6/17

FIG. 6 BETWEEN CALLS IDLE STATE (FOR TERMINAL A)**FIG. 7** START OF SECURE CALL (A TO B) SETUP - GENERATION OF KDC-A LINK KEYSBUREAU
OMPI

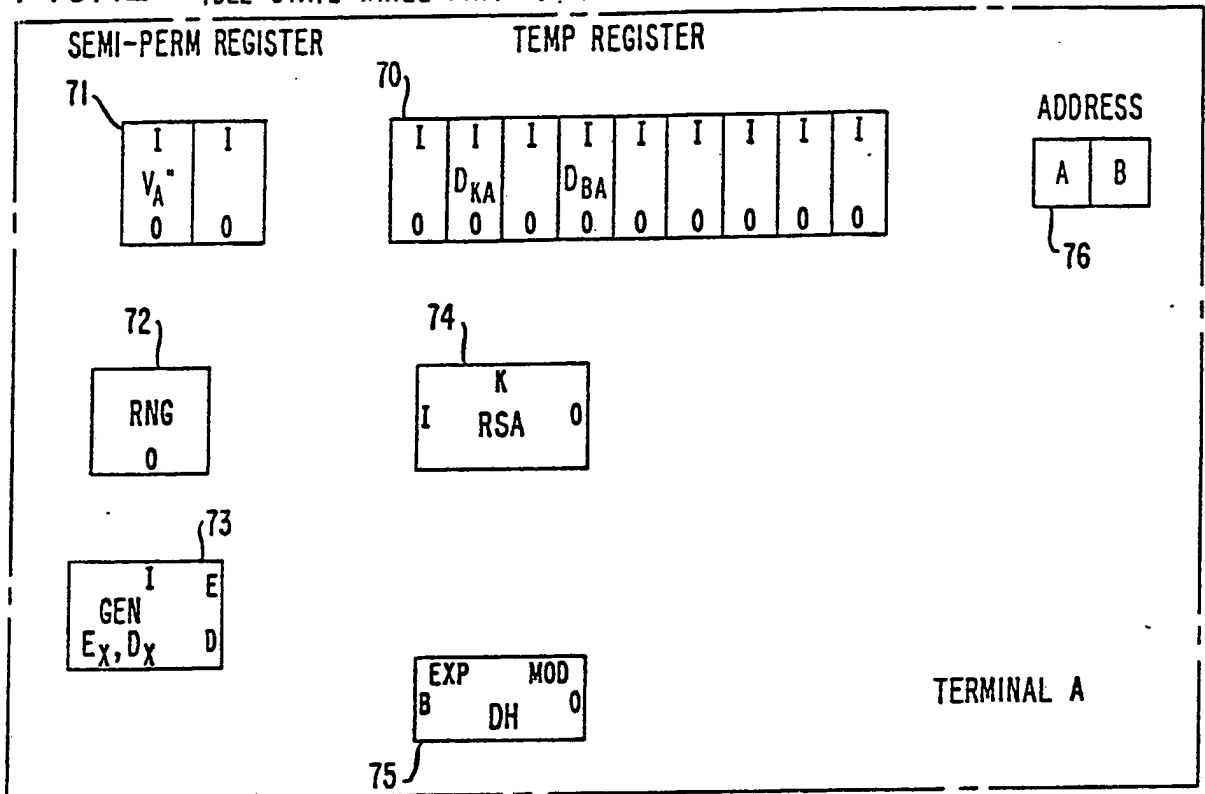
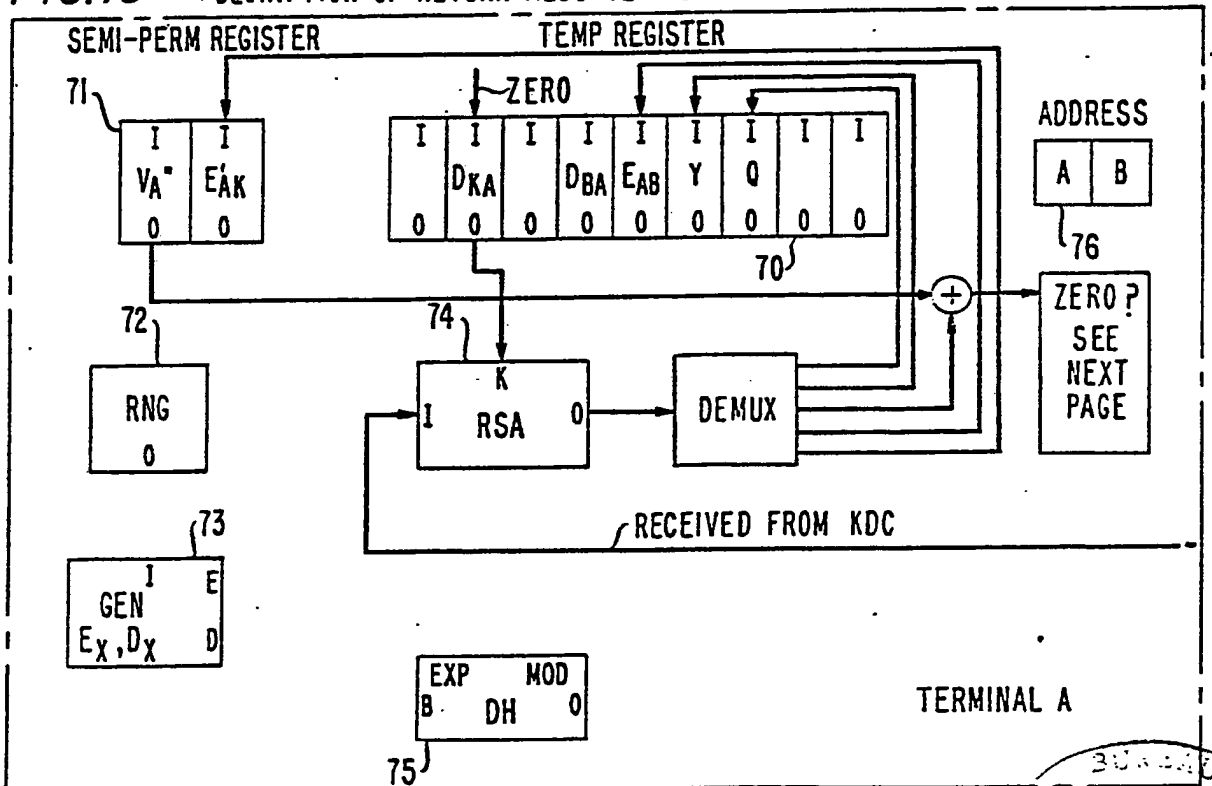
7/17

FIG. 8 GENERATION OF B-A LINK KEYS**FIG. 9** FIRST STEP IN UPDATE OF VERIFICATION INFORMATION

8/17

FIG. 10 SECOND STEP IN UPDATE OF VERIFICATION INFORMATION**FIG. 11** COMPUTATION OF A-KDC MESSAGE

9/17

FIG. 12 IDLE STATE WHILE WAITING FOR RETURN MESSAGE FROM KDC**FIG. 13** DECRYPTION OF RETURN MESSAGE FROM KDC

10/17

FIG. 14 VERIFICATION CHECK

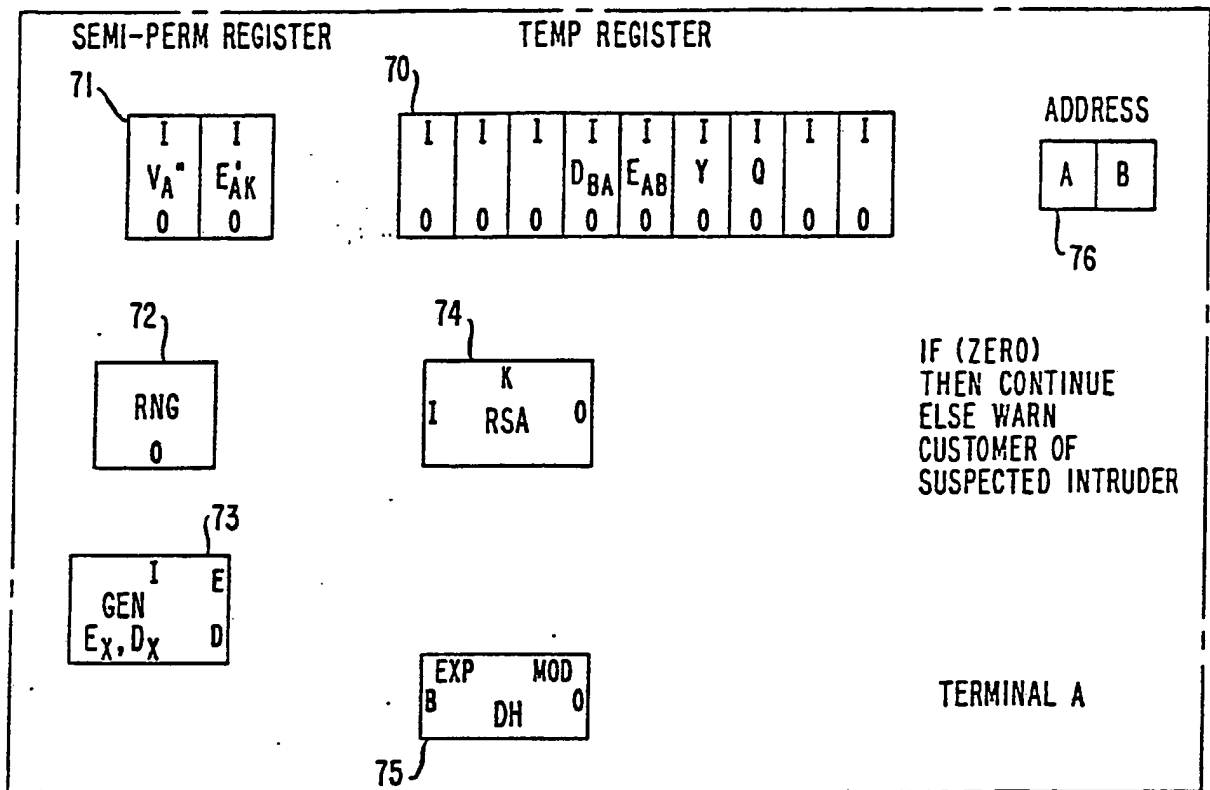
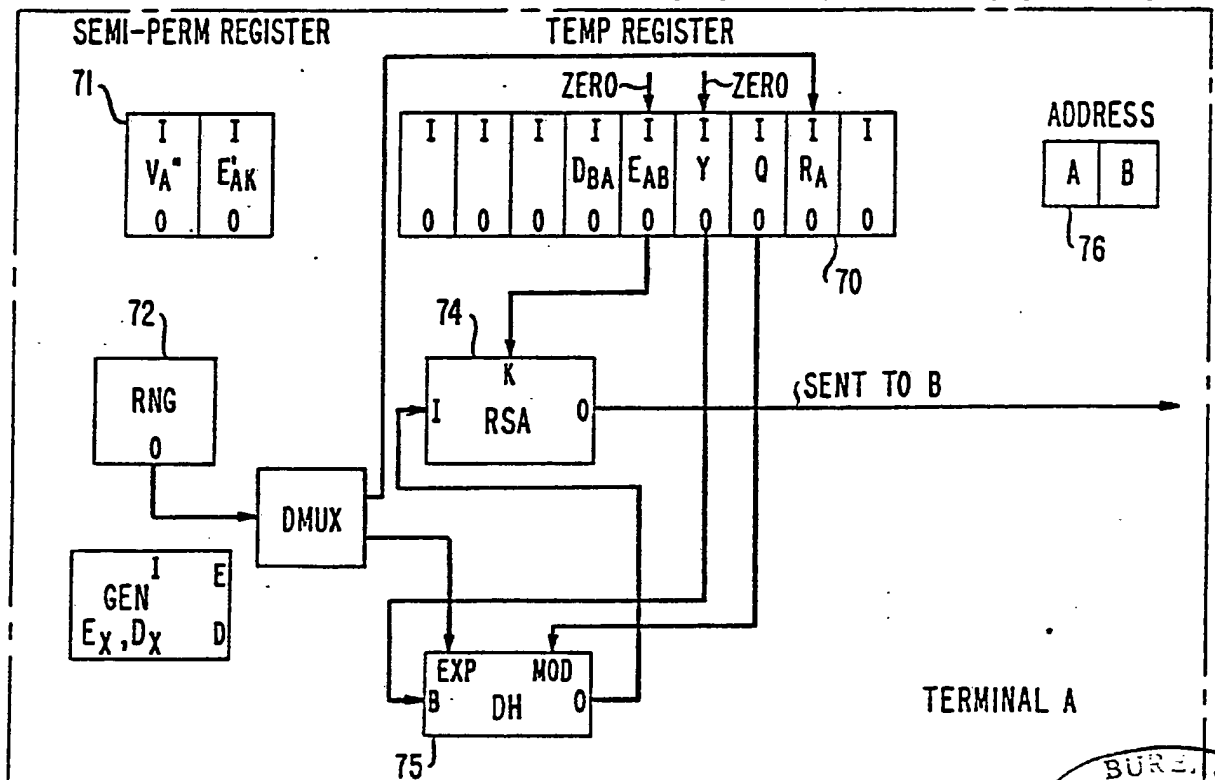


FIG. 15 START OF KEY EXCHANGE WITH B CALCULATION OF DIFFIE-HELLMAN KEYS



11/17
FIG. 16 IDLE WAIT STATE FOR RETURN MESSAGE FROM B

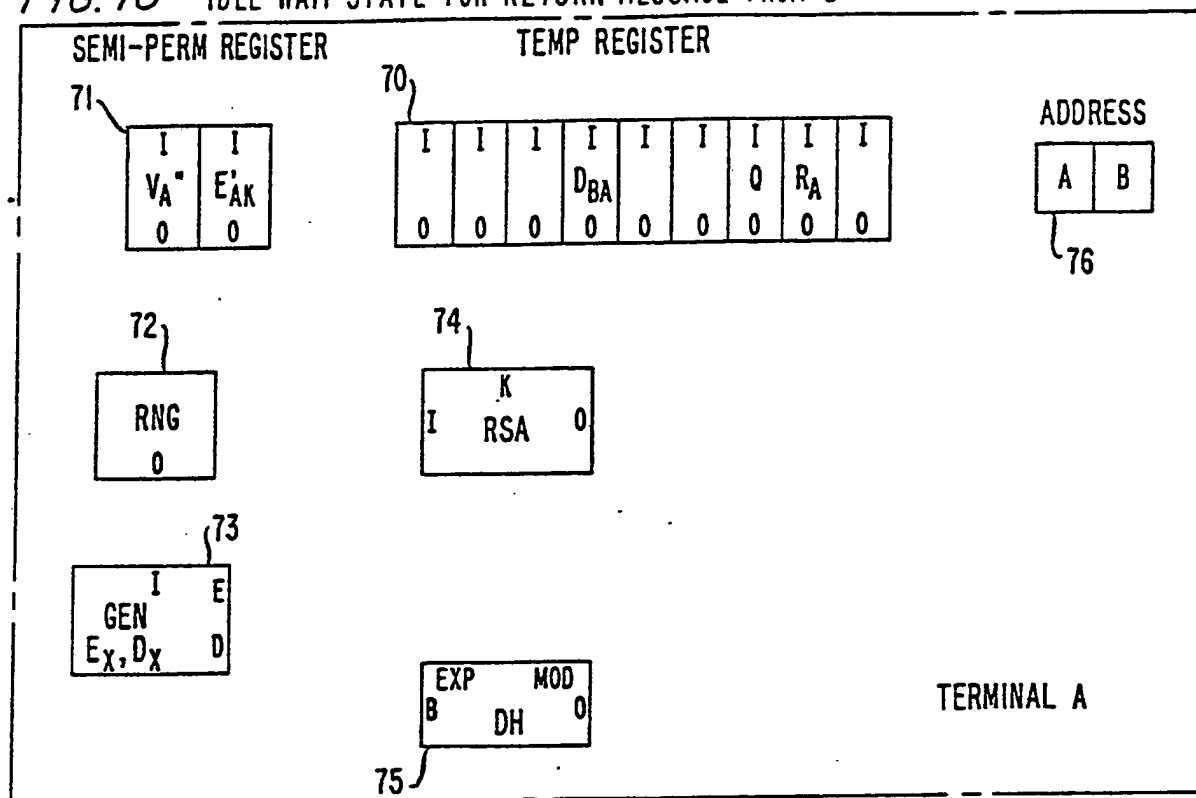
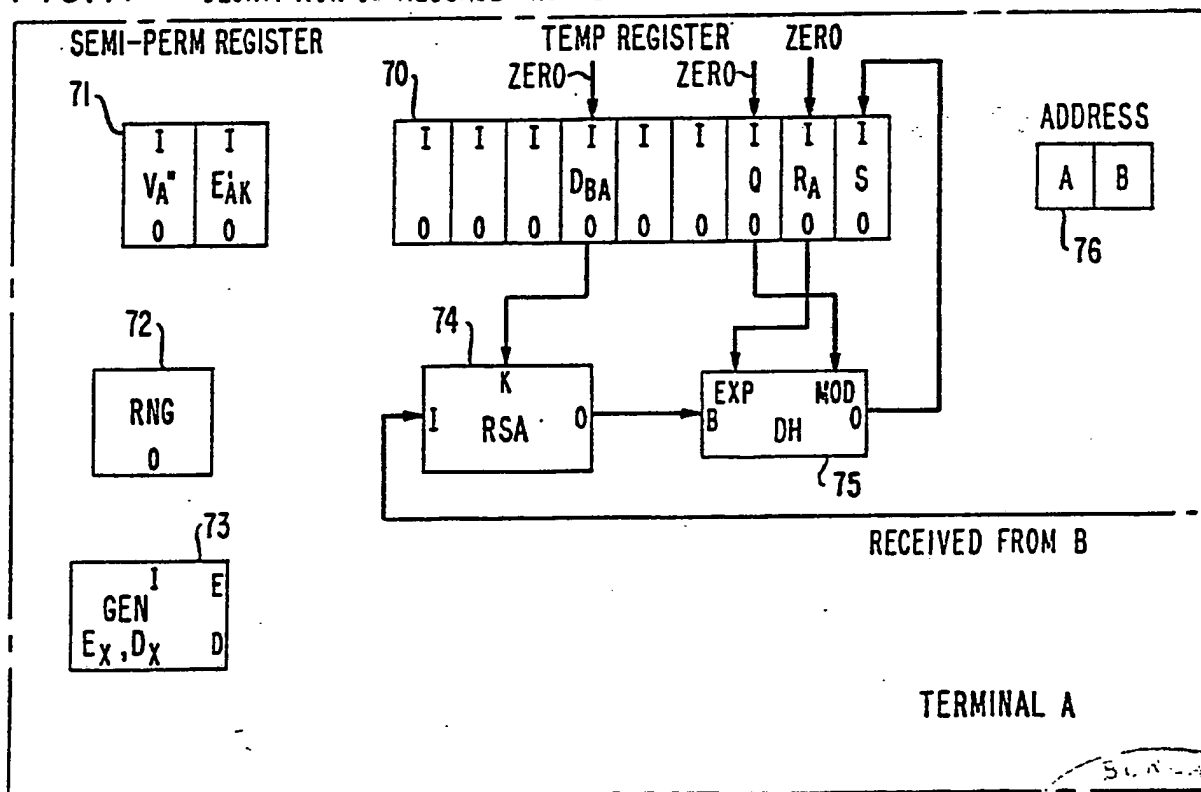


FIG. 17 DECRYPTION OF MESSAGE FROM B AND CALCULATION OF SESSION KEY-S



12/17

FIG. 18 KEY STORAGE DURING CALL

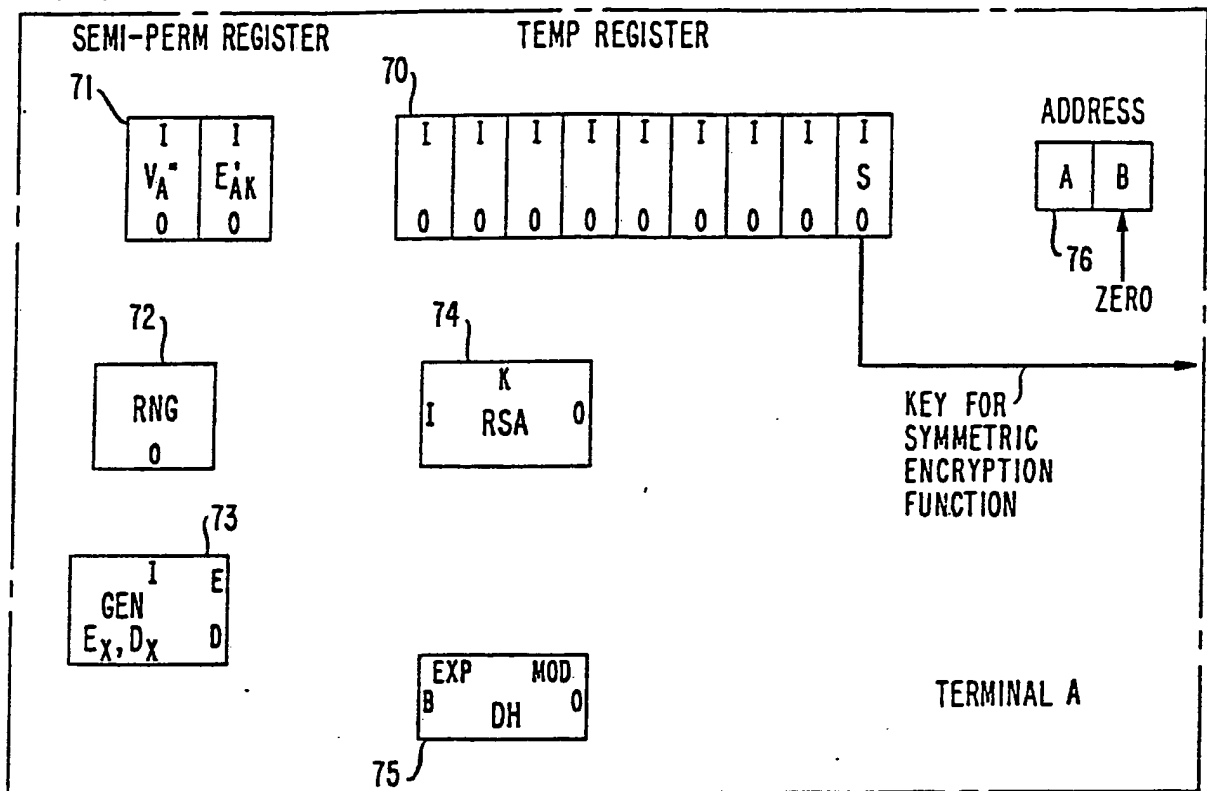
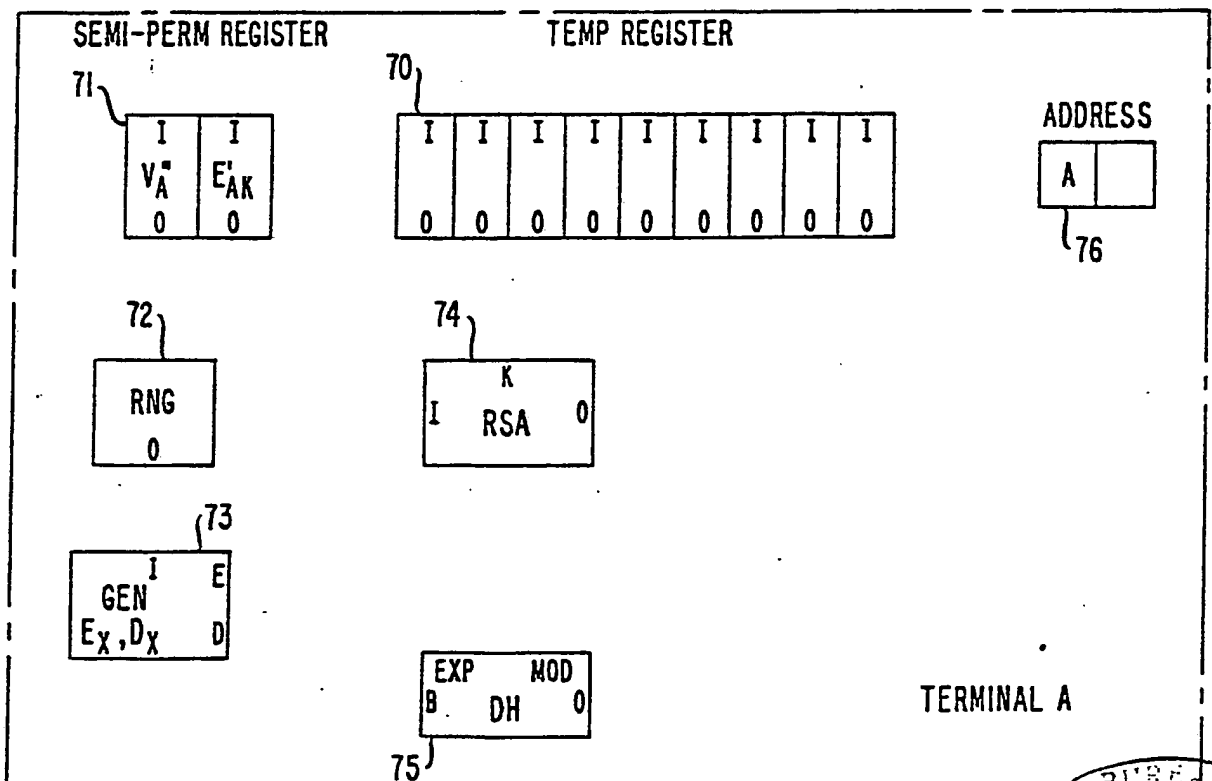


FIG. 19 IDLE STATE FOLLOWING CALL COMPLETION



13/17

FIG. 20 IDLE STATE BETWEEN CALLS

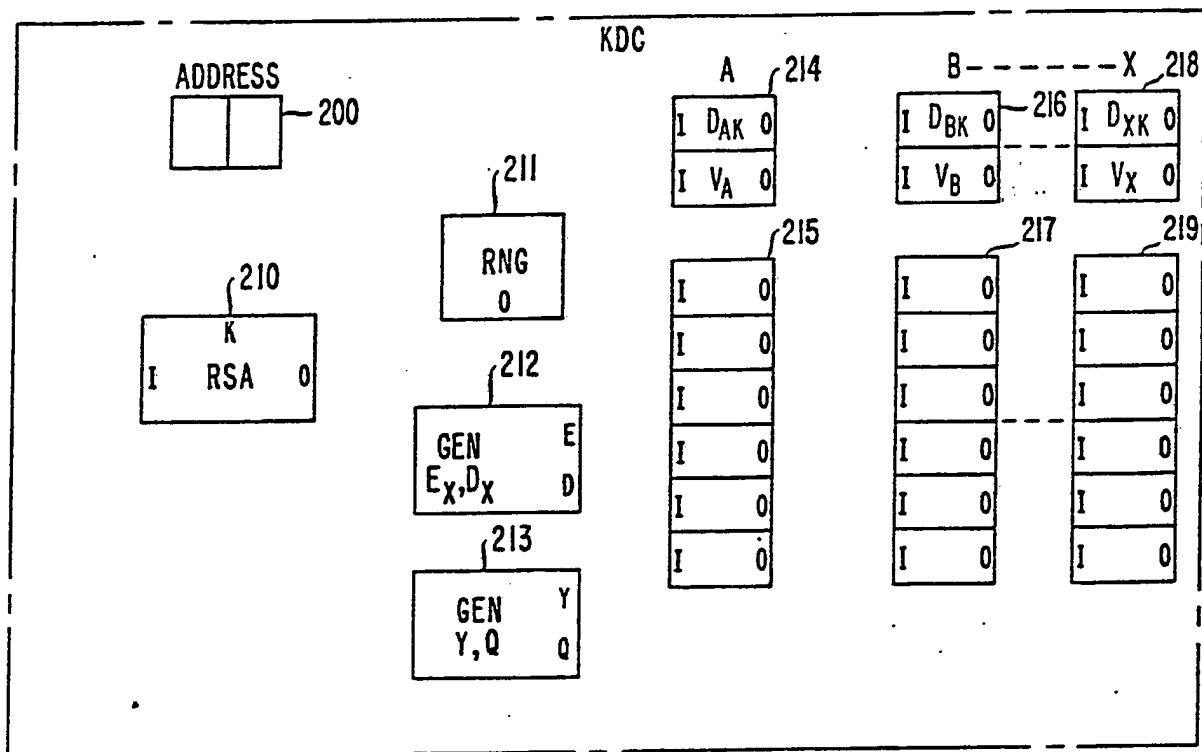
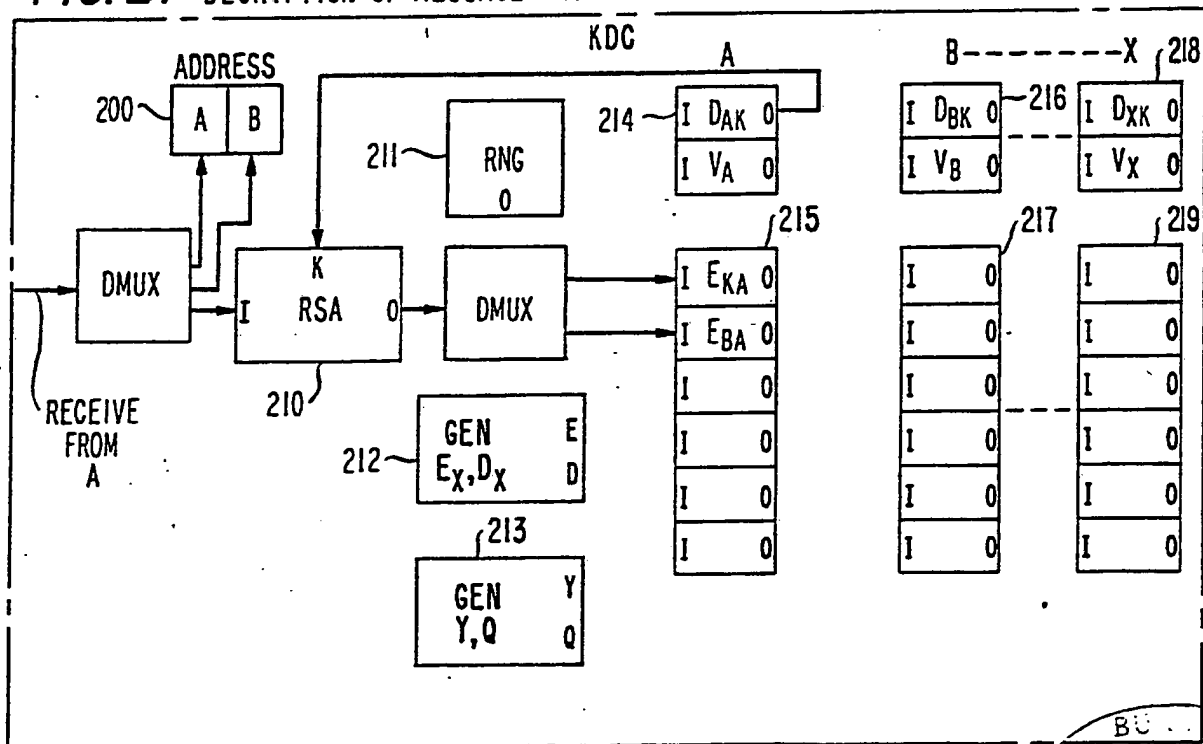
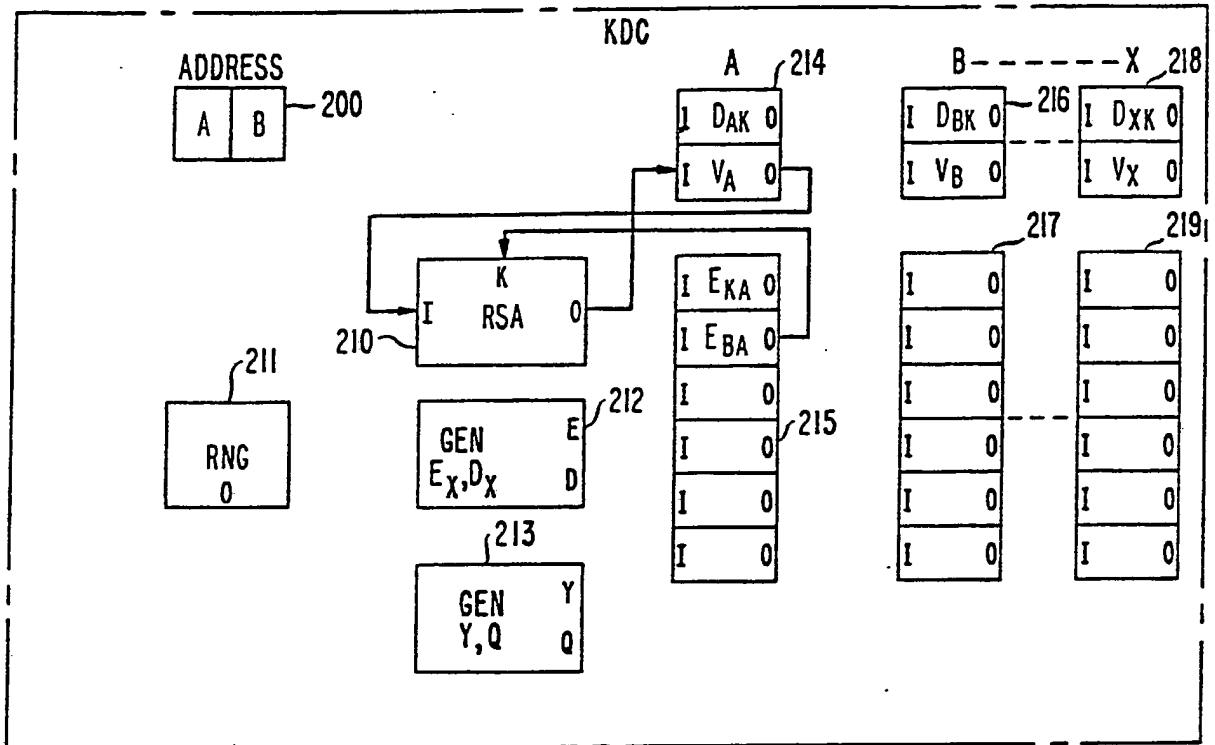
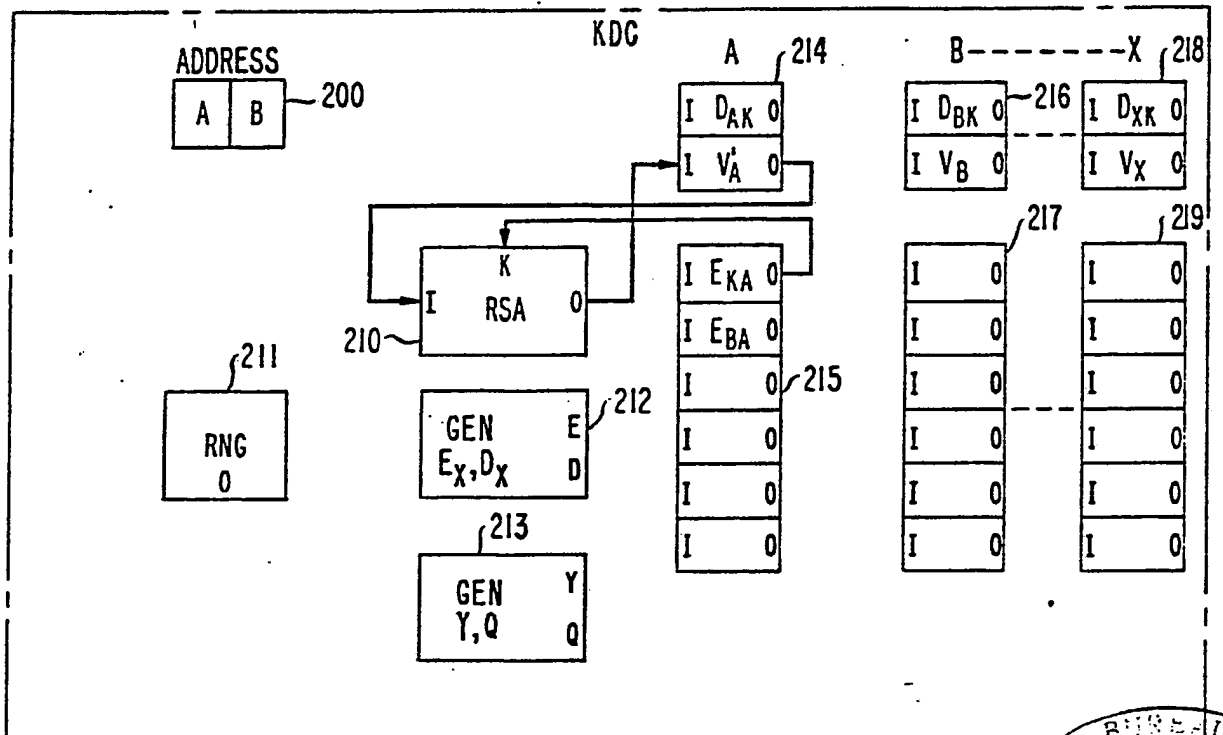


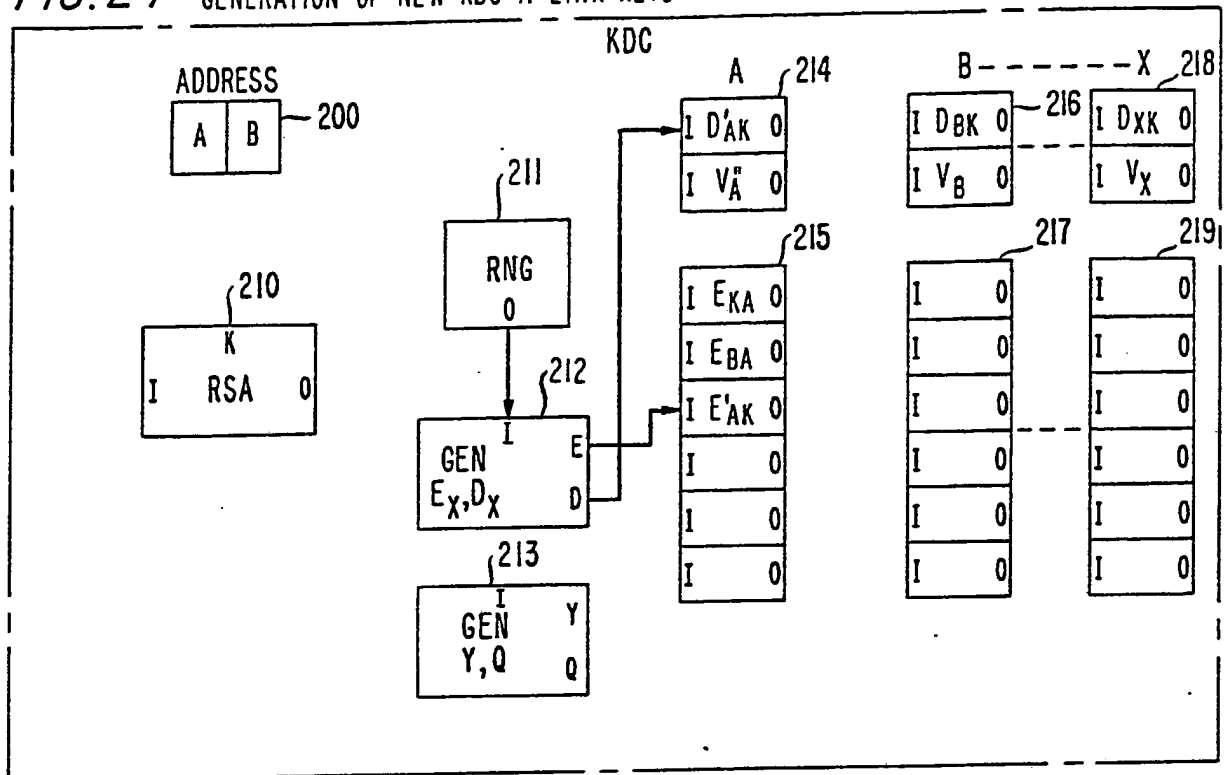
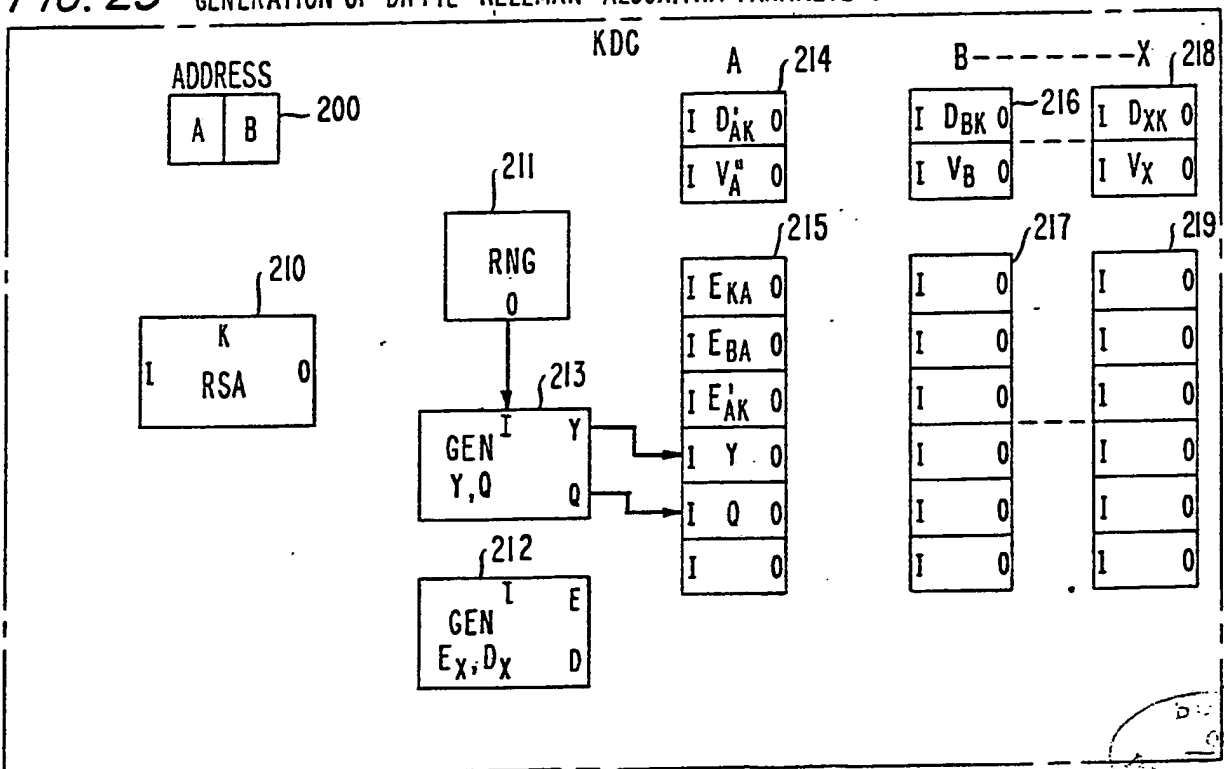
FIG. 21 DECRYPTION OF MESSAGE FROM A



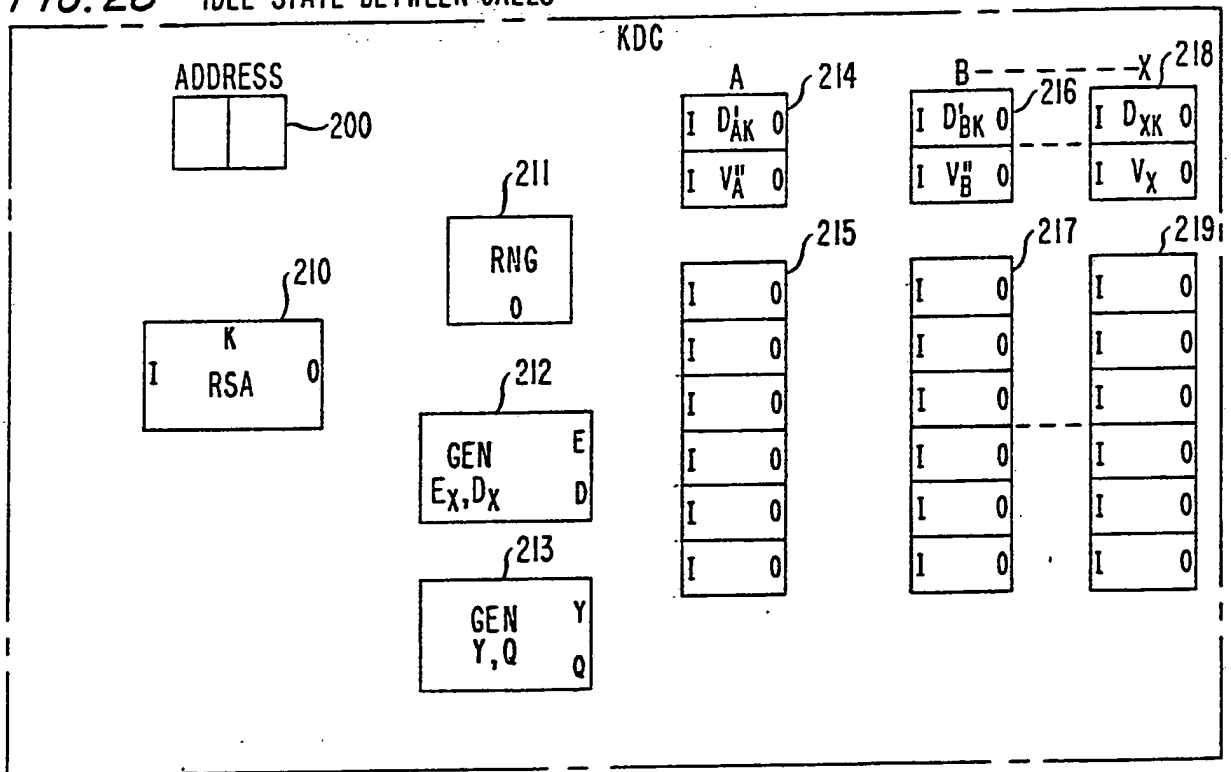
14/17

FIG. 22 FIRST STEP IN THE UPDATE OF VERIFICATION INFORMATION**FIG. 23** SECOND STEP IN THE UPDATE OF VERIFICATION INFORMATION

15/17

FIG. 24 GENERATION OF NEW KDC-A LINK KEYS**FIG. 25** GENERATION OF DIFFIE-HELLMAN ALGORITHM PARAMETERS

17/17

FIG. 28 IDLE STATE BETWEEN CALLS

INTERNATIONAL SEARCH REPORT

International Application No. PCT/US 83/00030

I. CLASSIFICATION OF SUBJECT MATTER (If several classification symbols apply, indicate all) ³ According to International Patent Classification (IPC) or to both National Classification and IPC IPC³: H 04 L 9/00																				
II. FIELDS SEARCHED Minimum Documentation Searched ⁴ <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <tr> <th style="width: 30%; padding: 5px;">Classification System</th> <th style="padding: 5px;">Classification Symbols</th> </tr> <tr> <td style="padding: 10px; vertical-align: top;">IPC³</td> <td style="padding: 10px; vertical-align: top;">H 04 L 9/00; H 04 L 9/02; H 04 K 1/00; H 04 L 9/04</td> </tr> </table> Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched ⁵			Classification System	Classification Symbols	IPC ³	H 04 L 9/00; H 04 L 9/02; H 04 K 1/00; H 04 L 9/04														
Classification System	Classification Symbols																			
IPC ³	H 04 L 9/00; H 04 L 9/02; H 04 K 1/00; H 04 L 9/04																			
III. DOCUMENTS CONSIDERED TO BE RELEVANT ¹⁴ <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <tr> <th style="width: 10%; padding: 5px;">Category ⁶</th> <th style="width: 70%; padding: 5px;">Citation of Document, ¹⁵ with Indication, where appropriate, of the relevant passages ¹⁷</th> <th style="width: 20%; padding: 5px;">Relevant to Claim No. ¹⁸</th> </tr> <tr> <td style="text-align: center; vertical-align: top; padding: 10px;">A</td> <td style="padding: 10px;">EP, A1, 0048903 (LICENTIA) 7 April 1982 see page 2, line 23 - page 4, line 22 --</td> <td style="text-align: center; vertical-align: top; padding: 10px;">1,2</td> </tr> <tr> <td style="text-align: center; vertical-align: top; padding: 10px;">A</td> <td style="padding: 10px;">US, A, 4182933 (ROSENBLUM) 8 January 1980 see column 7, lines 1-21; column 8, lines 37-50; column 11, lines 17-41 cited in the application --</td> <td style="text-align: center; vertical-align: top; padding: 10px;">1</td> </tr> <tr> <td style="text-align: center; vertical-align: top; padding: 10px;">A</td> <td style="padding: 10px;">DATAMATION, vol. 22, no. 8, August 1976 (Barrington, US) Sykes: "Protecting data by encryption", pages 81-85, see page 84, left-hand column, lines 23-42 --</td> <td style="text-align: center; vertical-align: top; padding: 10px;">1</td> </tr> <tr> <td style="text-align: center; vertical-align: top; padding: 10px;">A</td> <td style="padding: 10px;">IBM-Technical Disclosure Bulletin, vol. 22, no. 2, July 1979 (New York, US) Lennon et al.: "Composite cryptographic session keys for enhanced communication security", pages 643-646, see page 643, first line to page 644, line 8 --</td> <td style="text-align: center; vertical-align: top; padding: 10px;">1,2</td> </tr> <tr> <td style="text-align: center; vertical-align: top; padding: 10px;">A</td> <td style="padding: 10px;">Fifth International Conference on Digital Satellite Communications, 23-26 March 1981 (New York, US) Bic et al.: --</td> <td style="text-align: center; vertical-align: top; padding: 10px;">./.</td> </tr> </table>			Category ⁶	Citation of Document, ¹⁵ with Indication, where appropriate, of the relevant passages ¹⁷	Relevant to Claim No. ¹⁸	A	EP, A1, 0048903 (LICENTIA) 7 April 1982 see page 2, line 23 - page 4, line 22 --	1,2	A	US, A, 4182933 (ROSENBLUM) 8 January 1980 see column 7, lines 1-21; column 8, lines 37-50; column 11, lines 17-41 cited in the application --	1	A	DATAMATION, vol. 22, no. 8, August 1976 (Barrington, US) Sykes: "Protecting data by encryption", pages 81-85, see page 84, left-hand column, lines 23-42 --	1	A	IBM-Technical Disclosure Bulletin, vol. 22, no. 2, July 1979 (New York, US) Lennon et al.: "Composite cryptographic session keys for enhanced communication security", pages 643-646, see page 643, first line to page 644, line 8 --	1,2	A	Fifth International Conference on Digital Satellite Communications, 23-26 March 1981 (New York, US) Bic et al.: --	./.
Category ⁶	Citation of Document, ¹⁵ with Indication, where appropriate, of the relevant passages ¹⁷	Relevant to Claim No. ¹⁸																		
A	EP, A1, 0048903 (LICENTIA) 7 April 1982 see page 2, line 23 - page 4, line 22 --	1,2																		
A	US, A, 4182933 (ROSENBLUM) 8 January 1980 see column 7, lines 1-21; column 8, lines 37-50; column 11, lines 17-41 cited in the application --	1																		
A	DATAMATION, vol. 22, no. 8, August 1976 (Barrington, US) Sykes: "Protecting data by encryption", pages 81-85, see page 84, left-hand column, lines 23-42 --	1																		
A	IBM-Technical Disclosure Bulletin, vol. 22, no. 2, July 1979 (New York, US) Lennon et al.: "Composite cryptographic session keys for enhanced communication security", pages 643-646, see page 643, first line to page 644, line 8 --	1,2																		
A	Fifth International Conference on Digital Satellite Communications, 23-26 March 1981 (New York, US) Bic et al.: --	./.																		
[*] Special categories of cited documents: ¹⁶ "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 <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <tr> <td style="width: 50%; padding: 5px;"> Date of the Actual Completion of the International Search ² 20th April 1983 </td> <td style="width: 50%; padding: 5px;"> Date of Mailing of this International Search Report ³ 03 MAI 1983 </td> </tr> <tr> <td style="padding: 5px;"> International Searching Authority ¹ EUROPEAN PATENT OFFICE </td> <td style="padding: 5px;"> Signature of Authorized Officer ¹⁹ G.L.M. Huydenberg </td> </tr> </table>			Date of the Actual Completion of the International Search ² 20th April 1983	Date of Mailing of this International Search Report ³ 03 MAI 1983	International Searching Authority ¹ EUROPEAN PATENT OFFICE	Signature of Authorized Officer ¹⁹ G.L.M. Huydenberg														
Date of the Actual Completion of the International Search ² 20th April 1983	Date of Mailing of this International Search Report ³ 03 MAI 1983																			
International Searching Authority ¹ EUROPEAN PATENT OFFICE	Signature of Authorized Officer ¹⁹ G.L.M. Huydenberg																			

FURTHER INFORMATION CONTINUED FROM THE SECOND SHEET

"Privacy over digital satellite links", pages 243-249, see page 246, right-hand column, lines 32-43; page 247, left-hand column, lines 6-11; figure 3

1

V. ☐ OBSERVATIONS WHERE CERTAIN CLAIMS WERE FOUND UNSEARCHABLE ¹⁰

This international search report has not been established in respect of certain claims under Article 17(2) (a) for the following reasons:

1. ☐ Claim numbers _____, because they relate to subject matter ¹² not required to be searched by this Authority, namely:

2. ☐ Claim numbers _____, because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out ¹³, specifically:

VI. ☐ OBSERVATIONS WHERE UNITY OF INVENTION IS LACKING ¹¹

This International Searching Authority found multiple inventions in this international application as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims of the international application.

2. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims of the international application for which fees were paid, specifically claims:

3. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claim numbers:

4. ☐ As all searchable claims could be searched without effort justifying an additional fee, the International Searching Authority did not invite payment of any additional fee.

Remark on Protest

☐ The additional search fees were accompanied by applicant's protest.

☐ No protest accompanied the payment of additional search fees.



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : G06F 1/00, 12/14		A2	(11) International Publication Number: WO 96/24092
			(43) International Publication Date: 8 August 1996 (08.08.96)
(21) International Application Number: PCT/SE96/00115		(81) Designated States: AL, AM, AT, AT (Utility model), AU, AZ, BB, BG, BR, BY, CA, CH, CN, CZ, CZ (Utility model), DE, DE (Utility model), DK, DK (Utility model), EE, EE (Utility model), ES, FI, FI (Utility model), GB, GE, HU, IS, JP, KE, KG, KP, KR, KZ, 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), TJ, TM, TR, TT, UA, UG, US, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), Eurasian patent (AZ, BY, KG, KZ, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).	
(22) International Filing Date: 1 February 1996 (01.02.96)			
(30) Priority Data: 9500355-4 1 February 1995 (01.02.95) SE			
(71)(72) Applicant and Inventor: BENSON, Greg [US/SE]; Dalbackavägen 3, S-240 10 Dalby (SE).			
(72) Inventor; and			
(75) Inventor/Applicant (for US only): URICH, Gregory, H. [US/SE]; Warholmsvägen 8 B, S-224 65 Lund (SE).			
(74) Agent: AWAPATENT AB; P.O. Box 5117, S-200 71 Malmö (SE).			

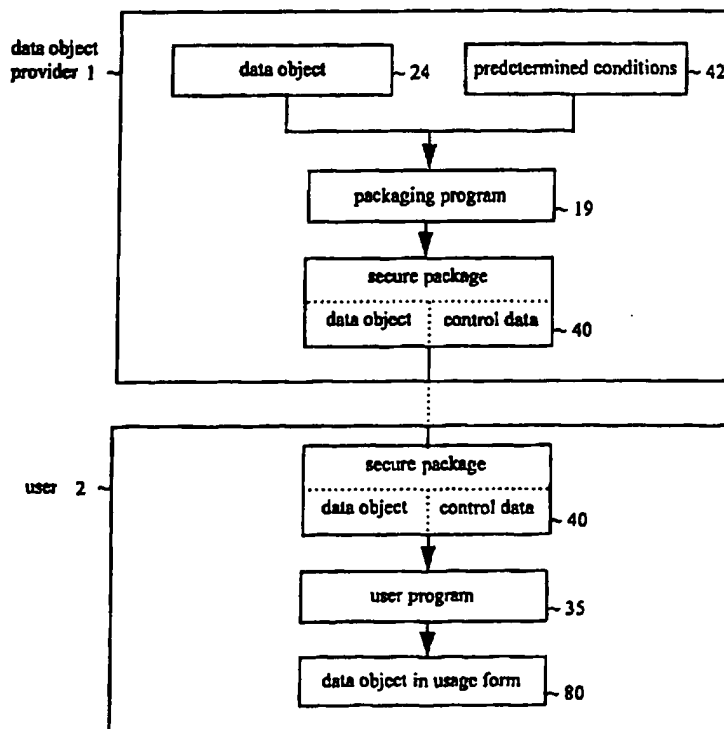
Published

Without international search report and to be republished upon receipt of that report.

(54) Title: **A METHOD AND SYSTEM FOR MANAGING A DATA OBJECT SO AS TO COMPLY WITH PREDETERMINED CONDITIONS FOR USAGE**

(57) Abstract

The present invention relates to a method and a system for managing a data object so as to comply with predetermined conditions for usage of the data object. To control the usage of the data object, a set of control data, defining usages of the data object which comply with the predetermined conditions, is created for the data object. The data object is concatenated with the user set of control data, encrypted and transferred to the user. When the user wants to use the data object, a special user program checks whether the usage complies with the control data. If so, the usage is enabled. Otherwise it is disabled.



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.

AM	Armenia	GB	United Kingdom	MW	Malawi
AT	Austria	GE	Georgia	MX	Mexico
AU	Australia	GN	Guinea	NE	Niger
BB	Barbados	GR	Greece	NL	Netherlands
BE	Belgium	HU	Hungary	NO	Norway
BF	Burkina Faso	IE	Ireland	NZ	New Zealand
BG	Bulgaria	IT	Italy	PL	Poland
BJ	Benin	JP	Japan	PT	Portugal
BR	Brazil	KE	Kenya	RO	Romania
BY	Belarus	KG	Kyrgyzstan	RU	Russian Federation
CA	Canada	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic	KR	Republic of Korea	SE	Sweden
CG	Congo	KZ	Kazakhstan	SG	Singapore
CH	Switzerland	LJ	Liechtenstein	SI	Slovenia
CI	Côte d'Ivoire	LK	Sri Lanka	SK	Slovakia
CM	Cameroon	LR	Liberia	SN	Senegal
CN	China	LT	Lithuania	SZ	Swaziland
CS	Czechoslovakia	LU	Luxembourg	TD	Chad
CZ	Czech Republic	LV	Latvia	TG	Togo
DE	Germany	MC	Monaco	TJ	Tajikistan
DK	Denmark	MD	Republic of Moldova	TT	Trinidad and Tobago
EE	Estonia	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	UG	Uganda
FI	Finland	MN	Mongolia	US	United States of America
FR	France	MR	Mauritania	UZ	Uzbekistan
GA	Gabon			VN	Viet Nam

A METHOD AND SYSTEM FOR MANAGING A DATA OBJECT SO AS TO
COMPLY WITH PREDETERMINED CONDITIONS FOR USAGE

Technical Field

The present invention relates to data processing and more particularly to a method and a system for managing data objects so as to comply with predetermined conditions for usage.

Background

Much has been written recently regarding the puzzle of universal connectivity. A typical vision of the data highway has long distance high speed data carriers interconnecting regional networks which provide telecommunications services and a wide range of interactive on-line services to consumers. Many of the pieces are already in place, others are in development or testing. In fact, even though the data highway is under construction it is currently open to limited traffic. On-line services are springing up daily and video on demand services are currently being tested.

The potential to benefit society is immense. The scope of information available to consumers will become truly global as the traditional barriers to entry for distribution of, and access to, information are lowered dramatically. This means that more diverse and specialized information will be made available just as conveniently as generic sources from major vendors used to be. The end result is that organizations and individuals will be empowered in ways heretofore only imagined.

However, a fully functioning data highway will only be as valuable as the actual services which it provides. Services envisioned for the data highway that involve the delivery of data objects (e.g. books, films, video, news, music, software, games, etc.) will be and are currently limited by the availability of such objects. Library and educational services are similarly affected. Before owners will allow their data objects to be offered they

must be assured of royalty payments and protection from piracy.

Encryption is a key component of any solution to provide copy protection. But encryption alone is not
5 enough. During transmission and storage the data objects will be protected by encryption, but as soon as anyone is given the key to decipher the content he will have unlimited control over it. Since the digital domain permits data objects to be reproduced in unlimited quantities
10 with no loss of quality, each object will need to be protected from unlimited use and unauthorized reproduction and resale.

The protection problem must not be solved by a separate solution for each particular data format, because
15 then the progress will indeed be slow. It is important to consider the effect of standardization on an industry. Consider how the VHS, the CD and the DAT formats, and the IBM PC compatibility standards have encouraged growth in their respective industries. However, if there is to be
20 any type of standardization, the standard must provide universal adaptability to the needs of both data providers and data users.

The data object owner may want to have permanent secure control over how, when, where, and by whom his
25 property is used. Furthermore, he may want to define different rules of engagement for different types of users and different types of security depending on the value of particular objects. The rules defined by him shall govern the automated operations enabled by data
30 services and networking. The owner may also want to sell composite objects with different rules governing each constituent object. Thus, it is necessary to be able to implement variable and extensible control.

The user on his part wants to be able to search for
35 and purchase data objects in a convenient manner. If desired, the user should be able to combine or edit purchased objects (i.e. for creating a presentation).

Furthermore, the user may want to protect his children from inappropriate material. A complete solution must enable these needs as well.

What is needed is a universally adaptable system and
5 method for managing the exchange and usage of data objects while protecting the interests of data object owners and users.

Prior Art

A method for enforcing payment of royalties when
10 copying softcopy books is described in the European patent application EP 0 567 800. This method protects a formatted text stream of a structured document which includes a royalty payment element having a special tag. When the formatted text stream is inputted in the user's
15 data processor, the text stream is searched to identify the royalty payment element and a flag is stored in the memory of the data processor. When the user for instance requests to print the document, the data processor requests authorization for this operation from a second
20 data processor. The second data processor charges the user the amount indicated in the royalty payment element and then transmits the authorization to the first data processor.

One serious limitation of this method is that it can
25 only be applied to structured documents. The description of the above-mentioned European patent application defines a structured document as: a document prepared in accordance with an SGML-compliant type definition. In other words it can not be applied to documents which are
30 not SGML compliant and it cannot be applied to any other types of data objects.

Furthermore, this method does not provide for variable and extensible control. Anyone can purchase a soft-copy book on a CD, a floppy disc or the like, and the
35 same royalty amount is indicated in the royalty payment element of all softcopy books of the same title.

Thus, the method described in EP 0 567 800 does not satisfy the above-mentioned requirements for universally adaptable protection of data objects.

Summary of the Invention

5 Accordingly, it is a first object of the invention to provide a method and a data processing system for managing a data object in a manner that is independent of the format and the structure thereof, so as to comply with predetermined conditions for usage control and
10 royalty payment.

 It is a further object of the invention to provide such a method and system which is universally adaptable to the needs of both the owner and the user of the data object.

15 A further object of the invention is to provide such a method and system which enables a data object provider to distribute his data object while maintaining control of the usage thereof.

 Yet another object of the invention is to provide a
20 method and system which allows a data object provider to select the level of security for his data object in a flexible way.

 Yet another object of the invention is to provide such a method and system which makes it possible to
25 establish an audit trail for the data object.

 Yet another object is to provide such a method and system which makes it possible to sell and buy data objects in a secure way.

 The above-mentioned objects are achieved by a method
30 and a system having the features of claims 1, 16, 21, 24 and 27.

 Particular embodiments of the inventions are recited in the subclaims.

 More particularly, a data object provider, e.g. the
35 owner of a data object or his agent (broker), stores the data object in a memory device, e.g. a bulk storage device, where it is accessible by means of the data

provider's data processor. The data object can consist of digital data, analog data or a combination or hybrid of analog and digital data.

5 A general set of control data, which is based on the predetermined conditions for usage of the data object, is created and stored in the same memory device as the data object or another memory device where it is accessible by the data provider's data processor. The predetermined conditions for usage may be defined by the data object
10 owner, by the broker or by anyone else. They may differ between different data objects.

The general set of control data comprises at least one or more usage control elements, which define usages of the data object which comply with the predetermined
15 conditions. These usages may encompass for instance the kind of user, a time limit for usage, a geographical area for usage, allowed operations, such as making a hard copy of the data object or viewing it, and/or claim to royalty payment. The general set of control data may comprise
20 other kinds of control elements besides the usage control element. In a preferred embodiment, the general set of control data comprises a security control element which defines a security procedure which has to be carried out before usage of the data object. It also comprises an
25 identifier, which uniquely identifies the general set of control data.

The general set of control data is concatenated with a copy of the data object. Thus, the control data does not reside in the data object, but outside it, which
30 makes the control data independent of the format of and the kind of data object and which allows for usage control independently of the data object format.

At least the usage control element(s) and the data object are encrypted, so that the user is unable to use
35 the data object without a user program which performs the usage control and which decrypts the data object. Alter-

natively, the whole set of control data and the copy of the data object may be encrypted.

5 A user may request authorization for usage of a data object residing at a data provider's processor via a data network or in any other appropriate way. The authorization may or may not require payment. When a request for authorization for usage is received, a user set of control data is created by the data provider's processor. The user set of control data comprises the general set of control data or a subset thereof including at least one of said usage control elements which is relevant for the actual user. It typically also includes a new identifier which uniquely identifies this set of control data. If relevant, the user set of control data also comprises an indication of the number of usages authorized. If more than one kind of usage is authorized, the number of each kind of usage may be specified. Finally, the user set of control data is concatenated with a copy of the data object, and at least the usage control elements and the copy of the data object are encrypted to create a secure data package ready for transfer to the user.

Before the data package is transferred to the user, it should be confirmed that the request for authorization for usage has been granted. The check is preferably carried out before the user set of control data is created. However, it can also be carried out in parallel with or after the creation of the user control data. In the latter case, the number of usages requested by the user is tentatively authorized and included in the user set, but if the request is refused the user set is cancelled or changed.

The data package may be transferred to the user by electronic means or stored on bulk storage media and transferred to the user by mail or by any suitable transportation means.

Once the data object has been packaged in the above-described manner, it can only be accessed by a user

program which has built-in usage control and means for
decrypting the data package. The user program will only
permit usages defined as acceptable in the control data.
Moreover, if the control data comprises a security con-
5 trol element, the security procedure prescribed therein
has to be complied with. In one embodiment, the usage
control may be performed as follows. If the user decides
to use a data object, the user program checks the control
data to see if this action is authorized. More particu-
10 larly, it checks that the number of authorized usages of
this kind is one or more. If so, the action is enabled
and the number of authorized usages decremented by one.
Otherwise, the action is interrupted by the user program
and the user may or may not be given the opportunity to
15 purchase the right to complete the action.

After the usage, the user program repackages the
data object in the same manner as it was packaged before.

When a data object is redistributed by a user or a
broker, new control elements are added in the control
20 data to reflect the relation between the old user/broker
and the new user/broker. In this way, an audit trail for
the data object may be created.

According to another aspect of the invention at
least two data packages are stored on a user's data
25 processor, which examines the usage control elements of
the data packages in order to find a match. If a match is
found, the user's data processor carries out an action
which is specified in the user set of control data. This
method can be used for selling and buying data objects.

30 Brief Description of Drawings

Fig. 1 is a flow diagram showing the general data
flow according to the invention.

Fig. 2 is a system block diagram of a data object
provider's data processor.

35 Fig. 3 is a block diagram showing the different
modules of a data packaging program according to the
invention.

Fig. 4 is a data flow diagram of a data packaging process.

Fig. 5 is an example of a header file.

Fig. 6 is an example of a usage data file.

5 Fig. 7 is a data flow diagram of loading an object to the data object provider's data processor.

Figs 8a and 8b are examples of control data for a data object on the data object provider's data processor and for an object ready to be transferred to a user,
10 respectively.

Fig. 9 is a data flow diagram of data packaging on the data object provider's data processor.

Fig. 10 is a flow diagram of a data packaging procedure.

15 Fig. 11 is a memory image of a data object and its control data.

Fig. 12a is a memory image of the concatenated control data and data object.

Fig. 12b is a memory image of the concatenated and
20 encrypted control data and data object.

Fig. 13 is a system block diagram of a user's data processor.

Fig. 14 is a block diagram showing the different modules of a user program according to the invention.

25 Fig. 15 is a flow diagram of using a data object on the user's data processor.

Fig. 16 is a flow diagram of how the user program operates in a specific application example.

Fig. 17 is an example of various data package structures for composite objects.
30

Description of the Best Mode for Carrying Out the Invention

General Overview

Fig. 1 is a flow diagram showing the general data
35 flow according to the invention. The flow diagram is divided into a data object provider part 1 and a user part 2.

In the data object provider part 1, a data object 24 is created by an author. The data object can consist of digital data, analog data or a combination or hybrid of analog and digital data. The primary difference between
5 analog data objects and digital data objects is the means for storage, transfer and usage.

The author also determines the conditions 42 for the usage of the data object 24 by a user. The data object 24 and the usage conditions 42 are input to a data packaging
10 program 19, which creates a secure data package 40 of the data object and of control data which are based on the input usage conditions 42. Once packaged in this way, the data object can only be accessed by a user program 35.

The data object may be packaged together with a
15 general set of control data, which is the same for all users of the data object. This may be the case when the data object is sent to a retailer or a bulletin board, wherefrom a user may obtain it. The data object may also be packaged as a consequence of a request from a user for
20 usage of the data object. In that case, the package may include control data which is specifically adapted to that user. This control data is called a user set of control data. It may for example comprise the number of usages purchased by the user. Typically, the user set of
25 control data will be created on the basis of the general set of control data and include at least a subset thereof. A user set of control data need not always be adapted for a specific user. All sets of control data which are created on the basis of a general set of control data
30 will be called a user set of control data. Thus, a set of control data can be a general set in one phase and a user set in another phase.

The above-mentioned data packaging can be carried out by the author himself by means of the data packaging
35 program 19. As an alternative, the author may send his data object to a broker, who inputs the data object and the usage conditions determined by the author to the data

packaging program 19 in order to create a secure package 3. The author may also sell his data object to the broker. In that case, the broker probably wants to apply his own usage conditions to the data packaging program.

- 5 The author may also provide the data object in a secure package to the broker, who repackages the data object and adds further control data which is relevant to his business activities. Various combinations of the above alternatives are also conceivable.

- 10 In the user part 2 of the flow diagram, the secure package 40 is received by a user, who must use the user program 35 in order to unpackage the secure package 40 and obtain the data object in a final form 80 for usage. After usage, the data object is repackaged into the
15 secure package 40.

The different parts of the system and the different steps of the method according to the invention will now be described in more detail.

The data provider's data processor:

- 20 Fig. 2 is a system block diagram of a data object provider's data processor. As mentioned above, the data object provider may be an author of a data object, an owner of a data object, a broker of a data object or anyone else who wants to distribute a data object, while
25 retaining the control of its usage. The data processor is a general or special purpose processor, preferably with network capabilities. It comprises a CPU 10, a memory 11 and a network adapter 12, which are interconnected by a bus 13. As shown in Fig. 2, other conventional means,
30 such as a display 14, a keyboard 15, a printer 16, a bulk storage device 17, and a ROM 18, may also be connected to the bus 13. The memory 11 stores network and telecommunications programs 21 and an operating system (OS) 23. All the above-mentioned elements are well-known to the
35 skilled person and commercially available. For the purpose of the present invention, the memory 11 also stores a data packaging program 19 and, preferably, a database

20 intended for control data. Depending upon the current operation, one or more data objects 24 can be stored in the memory 11 as shown or in the bulk storage 17. The data provider's data processor is considered secure.

5 The Data Packaging Program:

The data packaging program 19 is used for creating control data for controlling the usage of a data object and for packaging the data object and the control data into a secure package.

10 As shown in Fig. 3, it comprises a program control module 301, a user interface module 302, a packaging module 303, a control data creation module 304, an encryption module 305, one or more format modules 306, and one or more security modules 307.

15 The control module 301 controls the execution of the other modules. The user interface module 302 handles interaction with the data object provider. The packaging module 303 packages the control data and the data object. It uses the control data creation module 304, the format
20 modules 306, the security modules 307 and the encryption module 305 as will be described more in detail below.

The format modules 306 comprise program code, which is required to handle the data objects in their native format. They can fulfill functions such as data compression and data conversion. They can be implemented by any
25 appropriate, commercially available program, such as by means of a routine from the PKWARE Inc. Data Compression Library for Windows and the Image Alchemy package from Handmade Software Incorporated, respectively. They can
30 also be implemented by custom designed programs.

The security modules 307 comprise program code required to implement security, such as more sophisticated encryption than what is provided by the encryption module 305, authorization algorithms, access control and usage
35 control, above and beyond the basic security inherent in the data package.

The data packaging program 19 can contain many different types of both format and security modules. The program control module 301 applies the format and security modules which are requested by the data provider.

5 The encryption module 305 may be any appropriate, commercially available module, such as "FileCrypt" Visual Basic subprogram found in Crescent Software's QuickPak Professional for Windows - FILECRPT.BAS, or a custom designed encryption program.

10 The control data creation module 304 creates the control data for controlling the usage of the data object. An example of a control data structure will be described more in detail below.

The Control Data:

15 The control data can be stored in a header file and a usage data file. In a preferred embodiment, the header file comprises fields to store an object identifier, which uniquely identifies the control data and/or its associated data object, a title, a format code, and a
20 security code. The format code may represent the format or position of fields in the usage data file. Alternatively, the format code may designate one or more format modules to be used by the data packaging program or the user program. The security code may represent the en-
25 ryption method used by the encryption module 305 or any security module to be used by the data packaging program and the user program. The header file fields will be referred to as header elements.

30 The usage data file comprises at least one field for storing data which controls usage of the data object. One or more usage data fields which represent one condition for the usage of the data object will be referred to as a usage element. In a preferred embodiment, each usage element is defined by an identifier field, e.g. a serial
35 number, a size field, which specifies the size of the usage element in bytes or in any other appropriate way, and a data field.

The header elements and the usage elements are control elements which control all operations relating to the usage of the object. The number of control elements is unlimited. The data provider may define any number of control elements to represent his predetermined conditions of usage of the data object. The only restriction is that the data packaging program 19 and the user program 35 must have compatible program code to handle all the control elements. This program code resides in the packaging module and the usage manager module, to be described below.

Control elements can contain data, script or program code which is executed by the user program 35 to control usage of the related data object. Script and program code can contain conditional statements and the like which are processed with the relevant object and system parameters on the user's data processor. It would also be possible to use a control element to specify a specific proprietary user program which can only be obtained from a particular broker.

It is evident that the control data structure described above is but one example. The control data structure may be defined in many different ways with different control elements. For example, the partitioning of the control data in header data and usage data is not mandatory. Furthermore, the control elements mentioned above are but examples. The control data format may be unique, e.g. different for different data providers, or defined according to a standard.

30 The operation of the data packaging program

The operation of a first embodiment of the data packaging program will now be described with reference to the block diagram of Fig. 3 and the flow diagram of Fig. 4.

35 First a data provider creates a data object and saves it to a file, step 401. When the data packaging program is started, step 402, the user interface module

302 prompts the data object provider to input, step 403, the header information consisting of e.g. an object identifier, a title of the data object, a format code specifying any format module to be used for converting the
5 format of the data object, and a security code specifying any security module to be used for adding further security to the data object. Furthermore, the user interface module 302 prompts the data object provider to input usage information, e.g. his conditions for the usage of
10 the data object. The usage information may comprise the kind of user who is authorized to use the data object, the price for different usages of the object etc. The header information and the usage information, which may be entered in the form of predetermined codes, is then
15 passed to the control module 301, which calls the packaging module 303 and passes the information to it.

The packaging module 303 calls the control data creation module 304, which first creates a header file, then creates header data on the basis of the header
20 information entered by the data object provider and finally stores the header data, step 404-405. Then a usage data file is created, usage data created on the basis of the usage information entered by the data provider, and finally the usage data is stored in the usage
25 data file, step 406-407.

The packaging module 303 then applies any format and security modules 306, 307 specified in the header file, steps 408-413, to the data object.

Next, the packaging module 303 concatenates the
30 usage data file and the data object and stores the result as a temporary file, step 414. The packaging module 303 calls the encryption module 305, which encrypts the temporary file, step 415. The level of security will depend somewhat on the quality of the encryption and key methods
35 used.

Finally, the packaging module 303 concatenates the header file and the encrypted temporary file and saves

the result as a single file, step 416. This final file is the data package which may now be distributed by file transfer over a network, or on storage media such as CD-ROM or diskette, or by some other means.

5 Example 1

 An example of how the data packaging program 19 can be used will now be described with reference to Figs 5 and 6. In this example the data object provider is a computer graphics artist, who wants to distribute an image
10 that can be used as clip art, but only in a document or file which is packaged according to the method of the invention and which has usage conditions which do not permit further cutting or pasting. The artist wants to provide a free preview of the image, but also wants to be
15 paid on a per use basis unless the user is willing to pay a rather substantial fee for unlimited use. The artist will handle payment and usage authorization on a dial-up line to his data processor.

 The artist uses some image creation application,
20 such as Adobe's Photoshop to create his image. The artist then saves the image to file in an appropriate format for distribution, such as the Graphical Interchange Format (GIF). The artist then starts his data packaging program and enters an object identifier, a title, a format code
25 and a security code, which in this example are "123456789", "image", "a", and "b", respectively. In this example, the format code "a" indicates that no format code need be applied, and this code is selected since the GIF format is appropriate and already compressed.
30 Furthermore, the security code "b" indicates that no security module need be applied and this code is selected since the security achieved by the encryption performed by means of the encryption module 305 is considered appropriate by the artist.

35 Then the artist enters his dial-up phone number, his price for a single use of the image and for unlimited use of the data object, a code for usage types approved, and

for number of usages approved. For this purpose, the user interface module 302 may display a data entry form.

The data packaging program 19 creates control data on the basis of the information entered by the artist and stores the data in the header file and in the usage data file as shown in Figs 5 and 6, respectively. This data constitutes a general set of control data which is not specifically adapted to a single user, but which indicates the conditions of usage determined by the artist for all future users.

Then the package program 19 concatenates the data object and the control data in accordance with steps 414-416 of Fig. 4 to achieve the secure package. No format module or security module is applied to the data object, since they are not needed according to the data in the header file.

When the secure package has been obtained, the artist sends it to a bulletin board, from where it can be retrieved by a user.

20 Example 2

Below, another embodiment of the data packaging program 19 will be described with reference to Figs 7-12b. In this example, the data object consists of a video film, which is created by a film company and sent to a broker together with the predetermined conditions 42 for usage of the video. The broker loads the video 24 to the bulk storage 17 of his data processor. Then, he uses his data packaging program 19 to create a general set of control data 50 based on the predetermined conditions 42 for usage indicated by the film company. Furthermore, the address to the video in the bulk storage 17 is stored in an address table in the control database 20 or somewhere else in the memory 11. It could also be stored in the general set of control data 50. Finally, the general set of control data 50 is stored in the control database 20. It could also be stored somewhere else in the memory 11.

After these operations, which correspond to steps 401-407 of Fig. 4, the data packaging program is exited.

Fig. 8a shows the general set of control data for the video according to this example. Here the control data includes an identifier, a format code, a security code, the number of usage elements, the size of the data object, the size of the usage elements and two usage elements, each comprising an identifier field, a size field and a data field. The identifier may be a unique number in a series registered for the particular broker. In this example, the identifier is "123456789", the format code "0010", which, in this example, indicates the format of a AVI video and the security code is "0010". Furthermore, the first usage element defines the acceptable users for the video and the second usage element data defines the number of viewings of the video purchased by a user. The first usage element data is 1 which, for the purposes of this example will signify that only education oriented users are acceptable to the film company. The data field of the second usage element data is empty, since at this stage no viewings of the video has been purchased.

Managing Object Transfer:

The broker wants to transfer data objects to users and enable controlled usage in return for payment of usage fees or royalties. Managing the broker-user business relationship and negotiating the transaction between the broker and the user can both be automated, and the control data structure can provide unlimited support to these operations. The payment can be handled by transmitting credit card information, or the user can have a debit or credit account with the broker which is password activated. Preferably, payment should be confirmed before the data object is transferred to the user.

Data packaging:

When a user wants to use a data object, he contacts the broker and requests authorization for usage of the data object. When the request for authorization is received

ved in the broker's data processor, a data program compares the usage for which authorization is requested with the usage control elements of the control data of the data object to see if it complies with the predetermined conditions for usage indicated therein. The comparison may include comparing the user type, the usage type, the number of usages, the price etc. If the requested usage complies with the predetermined conditions the authorization is granted, otherwise it is rejected.

Fig. 9 is a data flow diagram of the data packaging on the broker's data processor, which occurs in response to a granted request from a user for authorization for usage of the video, e.g. a granted request for the purchase of two viewings.

In response to a granted request, the broker again applies the data packaging program 19. The general set of control data 50 and the data object 24 are input to the program from the control database 20 and the bulk storage 17, respectively. The program creates a user set of control data 60 on the basis of the general set of control data 50 and concatenates the user set 60 and the data object 24 to create a secure data package 40, which may then be transferred to the user by any suitable means. A copy of the user set of control data is preferably stored in the broker's control database. This gives the broker a record with which to compare subsequent use, e.g. when a dial-up is required for usage.

Fig. 10 is a flow diagram of an exemplary procedure used for creating a user set of control data and for packaging the user set of control data and the video into a secure package. Here, the procedure will be described with reference to the general set of control data shown in Fig. 8a.

The user set of control data 60, i.e. a set of control data which is adapted to the specific user of this example, is created in steps 1001-1003 of Fig. 11. First, the general set of control data 50 stored in the control

database is copied to create new control data, step 1001. Second, a new identifier, here "123456790", which uniquely identifies the user set of control data, is stored in the identifier field of the new control data 60, step 5 1002. Third, the data field of the second usage element is updated with the usage purchased, i.e. in this example with two, since two viewings of the video were purchased, step 1003.

The thus-created user set of control data, which 10 corresponds to the general set of control data of Fig. 8a is shown in Fig. 8b.

The user set of control data is stored in the control database 20, step 1004. Then, the video, which is stored in the bulk storage 17, is copied, step 1005. The 15 copy of the video is concatenated with the user set of control data, step 1006. The security code 0010 specifies that the entire data package 40 is to be encrypted and that the user program 35 must contain a key which can be applied. Accordingly, the whole data package is encrypted, 20 step 1007. Finally, the encrypted data package is stored on a storage media or passed to a network program, step 1008, for further transfer to the user.

Fig. 11 is a memory image of the video 24 and the user control data 60. The user control data and a copy of 25 the video 24 are concatenated as shown in Fig. 12a. The encrypted data package 40 is shown in Fig. 12b.

The procedure of Fig. 10 can be implemented by the data packaging program of Fig. 3. As an alternative to the procedure of Fig. 10, the user set of control data 30 can be created as in steps 1001-1003 and saved in a header file and in a usage data file, whereafter steps 408-416 of the data packaging program of Fig. 4 can be performed to create the secure package.

The above-described process for creating a user- 35 adapted set of control data may also be used by a user who wants to redistribute a data object or by a broker who wants to distribute the data object to other brokers.

Obviously, redistribution of the data object requires that redistribution is a usage approved of in the control data of the data object. If so, the user or the broker creates a user set of control data by adding new control elements and possibly changing the data fields of old control element to reflect the relation between the author and the current user/broker and between the current user/broker and the future user/broker. In this way, an audit trail is created.

10 The user's data processor:

The user's data processor, which is shown in Fig. 13, is a general or special purpose processor, preferably with network capabilities. It comprises a CPU 25, a memory 26, and a network adapter 27, which are interconnected by a bus 28. As shown in Fig. 13, other conventional means, such as a display 29, a keyboard 30, a printer 31, a sound system 32, a ROM 33, and a bulk storage device 34, may also be connected to the bus 28. The memory 26 stores network and telecommunications programs 37 and an operating system (OS) 39. All the above-mentioned elements are well-known to the skilled person and commercially available. For the purpose of the present invention, the memory 26 also stores a user program 35 and, preferably, a database 36 intended for the control data. Depending upon the current operation, a data package 40 can be stored in the memory 26, as shown, or in the bulk storage 34.

25 The user program:

The user program 35 controls the usage of a data object in accordance with the control data, which is included in the data package together with the data object.

As shown in Fig. 14, the user program 35 comprises a program control module 1401 a user interface module 1402, a usage manager module 1403, a control data parser module 1404, a decryption module 1405, one or more format modules 1406, one or more security modules 1407, and a file transfer program 1409.

The control module 1401 controls the execution of the other modules. The user interface module 1402 handles interactions with the user. The usage manager module 1403 unpackages the secure package 40. It uses the control
5 data parser module 1404, the decryption module 1405, the format modules 1406, and the security modules 1407.

The format modules 1406 comprise program code, which is necessary to handle the data objects in their native format, such as decompression and data format procedures.
10 The security modules 1407 comprises program code required to implement security above the lowest level, such as access control, usage control and more sophisticated decryption than what is provided by the basic decryption module 1405.

15 The user program 35 can contain many different types of both format and security modules. However, they should be complementary with the format and security modules used in the corresponding data packaging program. The usage manager module 1401 applies the format and security
20 modules which are necessary to use a data object and which are specified in its control data. If the proper format and security modules are not available for a particular data object, the usage manager module 1401 will not permit any usage.

25 The decryption module 1405 can be the above-mentioned FileCrypt Visual Basic subprogram or some other commercially available decryption program. It can also be a custom designed decryption module. The only restriction is that the decryption module used in the user program is
30 complementary with the encryption module of the data packaging program.

The control data parser module 1403 performs the reverse process of the control data creation module 304 in Fig. 3.

35 The user program 35 can have code which controls use of the program by password or by any other suitable method. A password may be added in a password control

element during packaging of the data object. The password is transferred to the user by registered mail or in any other appropriate way. In response to the presence of the password control element in the control data structure, the user program prompts the user to input the password. The input password is compared with the password in the control data, and if they match, the user program continues, otherwise it is disabled.

The user program 35 can also have procedures which alter the behavior of the program (e.g. provide filters for children) according to the control data of the user object 41. It is important to mention that the user program 35 never stores the object in native format in user accessible storage and that during display of the data object the print screen key is trapped.

The file transfer program 1409 can transfer and receive files via network to and from other data processor.

Since the data object is repackaged into the secure package after the usage, the user program should also include program code for repackaging the data object. The program code could be the same as that used in the corresponding data packaging program 19. It could also be a separate program which is called from the user program.

Operation of the user program:

The operation of an embodiment of the user program 35 will now be described with reference to the block diagram of Fig. 14 and the flow diagram of Fig. 15.

First the user receives a data package 40 via file transfer over a network, or on a storage media such as CD-ROM or diskette, or by any other appropriate means, step 1501. He then stores the data package as a file on his data processor, step 1502.

When the user wants to use the data object, he starts the user program 35, step 1503. Then he requests usage of the data object, step 1504. The request is received by the user interface module 1402, which noti-

fies the control module 1401 of the usage request. The control module 1401 calls the usage manager module 1403 and passes the usage request.

The usage manager module 1403 reads the format code
5 from the data package to determine the control data format. Then it calls the decryption module 1405 to decrypt and extract the control data from the data package. The usage manager module 1403 applies the decryption module 1405 incrementally to decrypt only the control data.

10 Finally, it stores the control data in memory, step 1505.

The usage manager module 1403 then calls the control data parser module 1404 to extract the data fields from the usage elements.

The usage manager module 1403 then compares the user
15 request for usage with the corresponding control data, steps 1506-1507. If the requested usage is not permitted in the control data, the requested usage is disabled, step 1508. However, if the requested usage is approved of in the control data, the usage manager module 1403 applies any format and security modules 1406, 1407 specified
20 in the header data or usage data, steps 1509-1514, to the data package.

Then the usage manager module 1403 calls the decryption module 1405, which decrypts the object data, step
25 1515, whereafter the requested usage is enabled, step 1516. In connection with the enabling of the usage, the control data may need to be updated, step 1517. The control data may for instance comprise a data field indicating a limited number of usages. If so, this data field
30 is decremented by one in response to the enabling of the usage. When the user has finished usage of the data object, the user program 35 restores the data package in the secure form by repackaging it, step 1518. More particularly, the data object and the usage elements are
35 reconcatenated and reencrypted. Then the header elements are added and the thus-created package is stored in the user's data processor.

Example 1 contd.

A specific example of how the user program operates will now be described with reference to Figs 6 and 15. The example is a continuation of Example 1 above, where
5 an artist created an image and sent it to a bulletin board.

Assume that a user has found the image at an electronic bulletin board (BBS) and is interested in using it. He then loads the data package 40 containing the image to
10 his data processor and stores it as a file in the bulk storage. The user then executes the user program 35 and requests to preview the image. The user program then performs steps 1505-1507 of the flow diagram in Fig. 15. The request for a preview of the image is compared with the
15 data field of the usage element "code for usage type approved". In this example, the code "9" designates that previews are permitted. Thus, the requested preview is OK. Then, the user program 35 performs step 1509-1515 of Fig. 15. Since the format code "a" and the security code
20 "b" of the header data indicate that neither conversion, nor decompression, nor security treatment is required, the user program only decrypts the object data. The usage manager module 1403 then displays the preview on the user's data processor and passes control back to the user
25 interface 1402.

When the user is finished previewing the image, the user interface module 1402 displays the costs for usage of the image in accordance with the price usage data of the control data ("price for single use" and "price for
30 unlimited use" in Fig. 6) and prompts the user to enter a purchase request. The user decides to buy unlimited use of the image, and the user interface module 1402 inputs purchase information, such as an identification, billing, and address for that request and passes the request to
35 the control module 1401. The control module calls the file transfer program 1409, which dials the artist's dial-up number as indicated in the usage data ("control

element for artist's phone number" in Fig. 6) and transfers the request and purchase information to a broker program on the artist's data processor. Upon approval of the purchase, the broker program returns a file containing an update for "usage type approved" control elements. The update is "10" for the usage type approved, which in this example indicates that unlimited use by that user is permitted. The file transfer program 1409 passes this update to the usage manager module 1403 which updates the control data with the "usage type approved" code. The user interface module 1402 then displays a confirmation message to the user.

Subsequently, the user interface module inputs a request to copy the image to a file packaged according to this invention, on the user's machine. The usage manager module then compares the user request control data. The usage manager module examines the data filed for "usage type approved", which now is "10". The usage manager module copies the image to the file.

When the user is finished with the image, the usage manager module 1403 repackages the image as before except with updated control data. This repackaging process is exactly like that shown in Fig. 4, except that the header and usage data already exist, so the process starts after step 406 where control data is created.

Improved security

If the data object provider wants to improve the security of a data package containing a data object, a security module 307 containing a sophisticated encryption algorithm, such as RSA, could be used. In that case the packaging module 303 calls the security module 307 in step 412 of the flow diagram of Fig. 4. The security module encrypts the image and passes a security algorithm code to the control data creation module 302, which adds a control element for the security module code, which will be detected by the user program 35. Then the data packaging continues with step 414. When the data package

is sent to the user, the public key is mailed to the user by registered mail. When the user program is executed in response to a request for usage of this data object, the usage manager module will detect the security module code
5 in the control data and call the security module. This module passes control to the user interface module 1402, which requests the user to input the public key. If the key is correct, the user security module applies complementary decryption using that key and passes a usage
10 approved message to the usage manager module, which enables the usage.

As another example of improved security, a security module may implement an authorization process, according to which each usage of the data object requires a dial-up
15 to the data processor of the data object provider. When the corresponding security module code is detected by the user program 35, the relevant security module is called. This module passes a request for authorization to the control module 1401, which calls the file transfer program 1409, which dial the data object provider's dial-up
20 number, which is indicated in a usage element and transfers the request for authorization of usage. Upon a granted authorization, the data provider's data processor returns a usage approved message to the user security
25 module, which forwards the approval to the usage control module, which enables one usage. If the user requests further usages of the data object, the authorization process is repeated. This procedure results in a permanent data object security.

30 Example 2 contd.

A further specific example of how the user program 35 operates will now be described with reference to Fig. 16. The example is a continuation of Example 2 above, where a user purchased two viewings of a video film from
35 a broker.

The user wants to play the video which was purchased and transferred from the broker. The user applies the

user program 35, step 1601, and requests to play the video, step 1602. The user program 35 first examines the user set of control data 60, step 1603. In this example, the user program 35 contains only those format and security modules for objects with format code of 0010 and
5 with a security code of 0010. Consequently, only those types of data objects may be used. If the program encounters other codes it will not enable the usage action, step 1604-1605.

10 Next, the user program 35 compares the first control element data which is 1, for educational users only, to user information entered by the user on request of the user program. Since the user type entered by the user is the same as that indicated in the first usage element the
15 process continues, steps 1606-1607. Then the user program checks the second control element data which specifies that the number of plays purchased is 2. Consequently, the usage is enabled, step 1609. The user program applies the decryption module with the universal key and the AVI
20 format video is displayed on the display unit 29. Then, the second control element data is decremented by one, step 1610. Finally, the video is repackaged, step 1611
Implementation of Variable and Extensible Object Control:

Object control is achieved through the interaction
25 of the data packaging program 19 and the usage program 35 with the control data. Variation of object control can be applied to a particular object by creating a control data format with control elements defining the control variation and the circumstances in which the variation is applied.
30 Program procedures should then be added to program modules to process the control elements. For example, suppose a broker wants to allow students to print a particular article for free but require business users to pay for it. He defines control elements to represent the
35 user types student and business and the associated costs for each. He then adds program logic to examine the user type and calculate costs accordingly. Object control is

extensible in the sense that the control data format can have as many elements as there are parameters defining the rules for object control.

Implementation of Variable and Extensible Object

5 Security:

Object security is also achieved through the interaction of the data packaging program 19 and the user program 35 with the control data. Security process and encryption/decryption algorithms can be added as program modules. Variation of object security can be applied to a particular object by creating a control data format with control elements defining the security variation and the circumstances in which the variation is applied. Program procedures should be added to program modules to process the control elements. For example, suppose a broker wants to apply minimal security to his collection of current news articles but to apply tight security to his encyclopedia and text books. He defines a control element for security type. He then adds program logic to apply the security algorithms accordingly. Object security is extensible in the sense that multiple levels of security can be applied. The level of security will of course be dependent on the encryption/key method which is implemented in the security modules. One level of security may be to require online confirmation when loading a data object to the user's data processor. This can be implemented in program code in a security module. This permits the broker to check that the object has not already been loaded as well as double check all other parameters.

30 It is also important to have version control with time stamping between the usage program and the user's control database. Otherwise the database can be duplicated and reapplied to the user program. The user program can place a time stamp in the control database and in a hidden system file each time the control database is accessed. If the time stamps are not identical, the control database has been tampered with and all usage is

disabled. Program code for handling time stamps can reside in a security module.

Handling Composite Objects:

A composite object can be handled by defining a control data format with control elements defining relationships between constituent objects and by defining a parent/child element and a related object id element. For example, suppose a broker wants to include a video and a text book in an educational package. He creates a parent object with control elements referring to the video and textbook objects. He also includes control elements in the control data for the video object and the textbook object referring to the parent object. Finally, he adds program procedures to program modules to process the control elements.

In other words, when the data object is a composite data object including at least two constituent data objects, a respective general set of control data is created for each of the constituent data object and the composite data object. In response to a request from a user, a respective user set of control data is created for each of the constituent data objects as well as for the composite data object.

Examples of various data package structures for composite objects are given in Fig. 17.

Another side of composite objects is when the user wants to combine data objects for some particular use. Combination is a usage action that must be permitted in each constituent data object. A new data object is created with control data linking the constituent data objects. Each constituent data object retains its original control data which continues to control its subsequent usage.

When a user requests authorization for usage of one constituent data object in a composite data object, a user set of control data is created only for that consti-

tuent data object and concatenated only with a copy of that constituent data object.

Scaleable Implementation:

5 The flexible control data structure and modular program structure permit almost boundless extensibility with regard to implementation of the owner's requirements for usage control and royalty payment. The control data structure can include control elements for complex user types, usage types, multiple billing schemes, artistic or
10 ownership credit requirements and others. Security modules can be included which interact with any variation of the control data structure and the control data. Security modules could require a dial up to the brokers data processor to approve loading or usage actions and to implement approval authentication mechanisms.
15

User acting as a broker:

A limited or full implementation of the broker's data packaging program can be implemented on the user's machine to permit further distribution or reselling. However, only those data objects with control data permitting further distribution or reselling are enabled in that way.
20

Rebrokering

25 An author of a data object may want to allow his original broker to distribute his data object to other brokers whom will also distribute his image. He then includes a control element which enables rebrokering in the control data before distributing the data object with its associated control data to the original broker. Upon
30 request for rebrokering, the original broker copies the general set of control data and updates the copy to create a user set of control data which will function as the general set of control data on the subsequent brokers data processor. The original broker packages the data
35 object with the user set of control data and transfers the package to the subsequent broker. The subsequent broker then proceeds as if he were an original broker.

Automated transaction negotiation

This is an example of how the predetermined conditions for usage included in the control data can be used for achieving automated transaction negotiation.

5 Suppose some company wants to provide a computer automated stock trading. Buy and sell orders could be implemented in the form of data packages and a user program could process the data packages and execute transactions. Data packages could carry digital cash and
10 manage payment based on conditions defined in the control data.

 In this example, the buy order is created using a data packaging program according to the invention on the buyer's data processor. The sell order is created using
15 the data packaging program on the seller's data processor. Both orders are used by the the user program on the stock trader's data processor. The usages would take the form of using a sell order data package to sell stock and a buy order data package to buy stock. The rules or conditions for buying and selling stocks could be indicated
20 in the control data of the packages. The data object consists of digital money. In this context it is important to remember that digital money is merely data which references real money or virtual money that is issued and
25 maintained for the purpose of digital transactions.

 In this example the buyer starts with a digital money data file. He uses the data packaging program to create control data, e.g. kind of stock, price, quantity, for the purchase, and he then packages the digital money
30 data file and the control data into a secure package as described above.

 The seller starts with an empty data file. This empty file is analogous to the digital money data file except it is empty. The seller creates control data, e.g.
35 kind of stock, price, quantity, and packages the empty file and the control data into a secure package.

Both the sell order package and the buy order package are transferred to the data processor of the stock trading company, where they are received and stored in the memory. The user program of the stock trading company
5 examines the control data of the buy and sell order packages in the same way as has been described above and looks for a match. Upon identifying matched buy and sell orders the user program executes a transaction, whereby the digital money is extracted from the buy order data
10 package and transferred to the sell order package. Then the control data of the data packages is updated to provide an audit trail. Both packages are repackaged in the same manner as they were previously packaged and then transferred back to their authors.

15 The above described technique could be used for selling and buying any object as well as for automated negotiations. Payment may be carried out in other ways than by digital money.

In the general case, the data processor of the user
20 decrypts the usage control elements of the user sets of control data and examines the usage control elements to find a match. In response to the finding of a match, the user's data processor carries out an action which is specified in the user set of control data.

25

CLAIMS

1. A method for managing a data object so as to comply with predetermined conditions for usage of the data object, comprising the steps of:
- storing the data object in a memory device, where it is accessible by means of a data object provider's data processor;
 - creating, by said data processor, a general set of control data for the data object based on said predetermined conditions for usage, said general set of control data comprising at least one or more usage control elements defining usages of the data object which comply with said predetermined conditions;
 - storing said general set of control data in a memory device, where it is accessible by said data processor;
 - concatenating the general set of control data with a copy of the data object; and
 - encrypting at least the copy of the data object and said one or more usage control elements to create a secure data package which is ready for transfer to a user.
2. A method as set forth in claim 1, wherein the step of encrypting comprises encrypting the data object and the general set of control data.
3. A method as set forth in claims 1 or 2, wherein the step of creating control data comprises creating an identifier which uniquely identifies the general set of control data.
4. A method as set forth in claims 1, 2 or 3, wherein the step of creating a general set of control data comprises creating a security control element which identifies a security process to be applied before usage of the data object is allowed.
5. A method as set forth in any of the preceding claims, wherein the step of creating a general set of

control data comprises creating a format control element which identifies the format of the control data.

6. A method as set forth in any of the preceding claims, comprising the further steps of:

- 5 - creating, in response to a request for authorization for usage of the data object by a user, a user set of control data, which comprises at least a subset of the general set of control data, including at least one of said usage control elements;
- 10 - using the user set of control data instead of the general set of control data in said concatenating step;
- using the at least one usage control element of the user set of control data instead of the one or more usage control elements of the general set of control data
- 15 in the encrypting step;
- checking, before allowing transfer of the data package to the user, that said request for authorization for usage of the data object has been granted.

7. A method as set forth in any of the preceding
20 claims, further comprising the steps of receiving in said data processor the request for authorization for usage by a user; comparing the usage for which authorization is requested with said one or more usage control elements of the general set of control data and granting the authori-
25 zation if the usage for which authorization is requested complies with the usages defined by said one or more usage control elements.

8. A method as set forth in claim 7, further comprising the step of securing payment for the requested
30 authorization for usage before granting the authorization.

9. A method as set forth in any one of claims 6-8, wherein the data object is composed of at least two constituent data objects and wherein the user set of control
35 data, in response to a request for authorization for usage of one of said constituent data objects by a user, is created only for that constituent data object and

concatenated only with a copy of that constituent data object.

10. A method as set forth in any one of claims 6-9, wherein the data provider's data processor is connected to a data network and the request for authorization is received from a data processor of the user, which is also connected to the data network, further comprising the step of transferring the data package through the data network to the user's data processor.

11. A method as set forth in any one of claims 6-8 or 10, wherein the data object is a composite data object including at least two constituent data objects and wherein the step of creating a general set of control data comprises the step of creating a respective general set of control data for each of the constituent data objects and the composite data object and wherein the step of creating a user set of control data comprises the step of creating a respective user set of control data for each of the constituent data objects and the composite data object.

12. A method as set forth in any one of claims 6-11, comprising the further step of storing a copy of the user set of control data in the data object provider's processor.

13. A method as set forth in any of the preceding claims, comprising the further steps of:

- receiving the data package in a user's data processor;
- storing the data package in a memory device where it is accessible by means of the user's data processor;
- decrypting said one or more usage control elements;
- checking, in response to a request by the user for usage of the data object, whether the requested usage complies with the usage defined by the at least one usage control element of the general set of control data;

- decrypting, in response to the requested usage complying with the usage defined by the at least one usage control element of the general set of control data, the data object and enabling the requested usage, otherwise disabling it.

14. A method as set forth in any one of claims 6-12, comprising the further steps of:

- receiving the data package in a user's data processor;

10 - storing the data package in a memory device where it is accessible by means of the user's data processor;

- decrypting the at least one usage control element of the user set of control data;

15 - checking, in response to a request by the user for usage of the data object, whether the requested usage complies with the usage defined by the at least one usage control element of the user set of control data;

20 - decrypting, in response to the requested usage complying with the usage defined by the at least one usage control element of the user set of control data, the data object and enabling the requested usage, otherwise disabling it.

15. A method as set forth in claims 13 or 14, comprising the further steps of reconcatenating, after the usage of the data object, the data object and the one or more usage control elements, reencrypting at least the data object and the one or more usage control elements, and storing the thus-repackaged data package in the memory of the user's data processor.

30 16. A method for controlling the usage by a user of a data object so as to comply with predetermined conditions for usage of the data object, comprising the steps of:

35 - storing a data package in a memory device, where it is accessible by means of a data processor of the user, said data package comprising the data object and control data, which comprises at least one usage control

element defining a usage of the data object which complies with the predetermined conditions, the data object and said at least one usage control element being encrypted;

- 5 - receiving a request by the user for usage of the data object;
- decrypting the control data;
- checking, in response to the request by the user for usage of the data object, whether the requested usage
- 10 complies with the usage defined by the at least one usage control element of the control data;
- decrypting, in response to the requested usage complying with the usage defined by the at least one usage control element of the control data, the data
- 15 object and enabling the requested usage, otherwise disabling it.

17. A method as set forth in claim 16, wherein the usage control element is updated after the usage of the data object.

- 20 18. A method as set forth in claims 16 or 17, wherein said control data comprises an indication of the number of times the user is authorized to use the data object in accordance with said at least one user control element; wherein the requested usage of the data object
- 25 is only enabled when said number of times is one or more; and wherein said number of times is decremented by one when the requested usage is enabled.

19. A method as set forth in any one of claims 16-18, wherein the control data comprise a security control element, and further comprising the step of carrying out, before each usage of the data object, a security procedure defined in the security control element.
- 30

20. A method as set forth in any one of claims 16-19, wherein the step of checking whether the requested
- 35 usage complies with the usage defined by the at least one usage control element comprises the step of checking that the user's data processor is capable of carrying out the

security procedure specified in the security control element of the user set of control data, and if not, disabling the usage.

21. A method as set forth in any one of claims
5 16-20, comprising the further steps of reconcatenating, after the usage of the data object, the data object and the one or more usage control elements, reencrypting at least the data object and the one or more usage control elements, and storing the thus-repackaged data package in
10 the memory of the user's data processor.

22. A system for managing a data object so as to comply with predetermined conditions for usage of the data object, comprising

- first means in the data object provider's data
15 processor for creating a general set of control data for the data object based on the predetermined conditions for usage, said general set of control data comprising at least one or more usage control elements defining usages of the data object which comply with the predetermined
20 conditions;

- storing means, which are accessible by means of said data processor, for storing the data object and the general set of control data;

- concatenating means for concatenating the general
25 set of control data with a copy of the data object; and

- encrypting means for encrypting the copy of the data object and at least said one or more usage control elements to create a secure data package, which is ready for transfer to a user.

23. A system as set forth in claim 22, further comprising

- second means in said data processor for creating, in response to a request for authorization for usage of the data object by a user, a user set of control data, which comprises at least a subset of the general set of
35 control data, which subset comprises at least one of said usage control elements; and

- checking means in said data processor for checking that said request for authorization for usage of the data object has been granted before allowing transfer of the data package to the user.

5 24. A system as set forth in claims 22 or 23, wherein the general set of control data comprises a control data element which defines the right to further distribution of the data object by the user.

10 25. A system for controlling the usage by a user of a data object so as to comply with predetermined conditions for usage of the data object, comprising

15 - storing means for storing a data package which comprises a data object and a control data comprising at least one usage control element defining a usage of the data object which complies with the predetermined conditions;

 - means for decrypting the at least one usage control element and the data object;

20 - checking means for checking whether a usage requested by the user complies with the usage defined by said at least one usage control element;

 - enabling means for enabling the usage requested by the user when the usage complies with the usage defined by said at least one usage control element; and

25 - disabling means for disabling the usage requested by the user when the usage does not comply with the usage defined by said at least one usage control element.

30 26. A system as set forth in claim 25, further comprising means for repackaging the data object after usage thereof.

 27. A method for controlling the usage by a user of data objects so as to comply with predetermined conditions for usage of the data objects, comprising the steps of:

35 - storing at least two data packages in a memory device, where they are accessible by a data processor of the user, each said data package comprising a data object

and a user set of control data, which comprises at least one usage control element defining a usage of the data object which complies with the predetermined conditions, the data object and said at least one usage control
5 elements being encrypted;

- decrypting the usage control elements of the user sets of control data;

- examining the usage control elements of said at least two data packages to find a match;

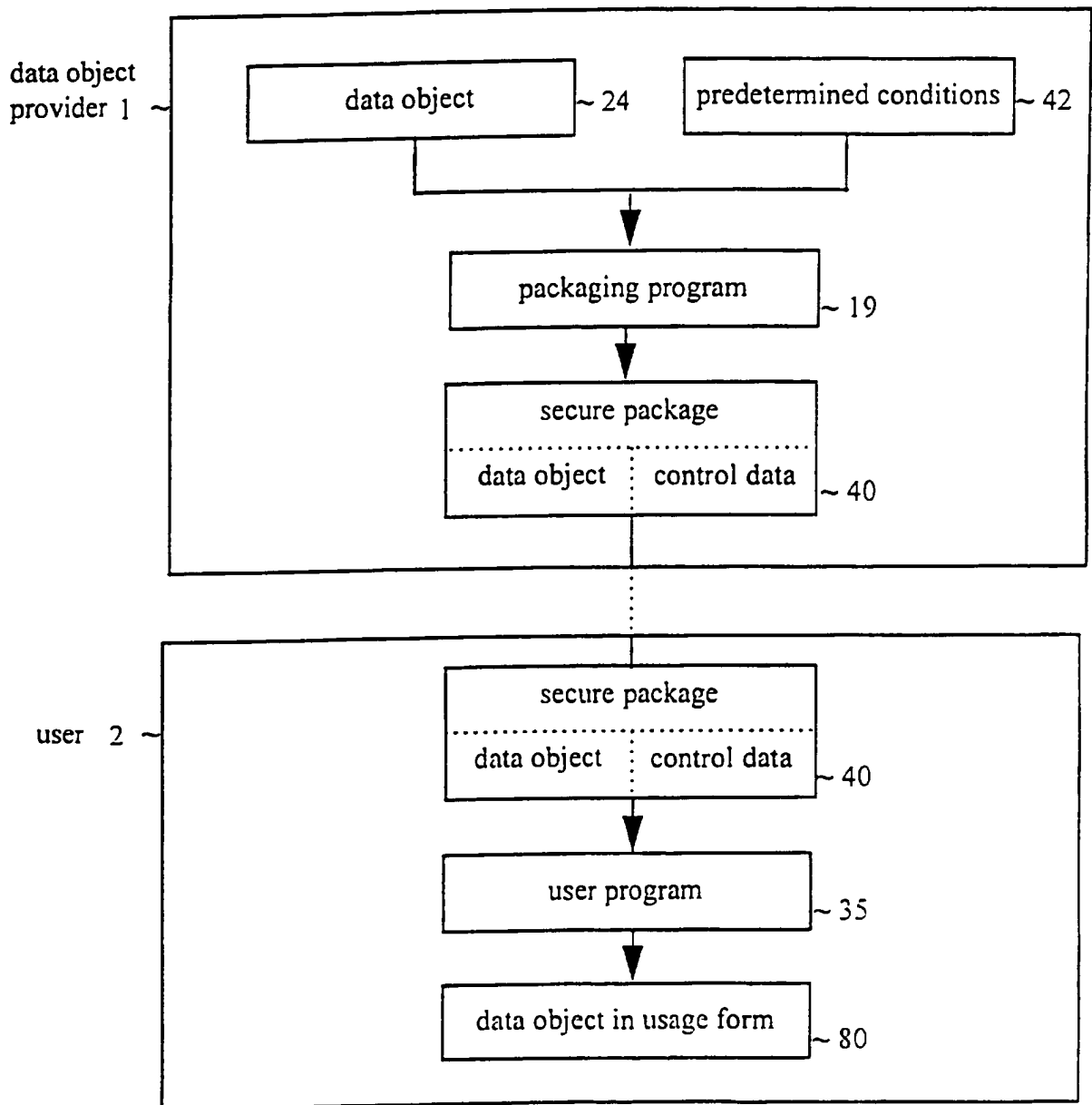
10 - using, in response to the finding of a match, the data processor to carry out an action, which is specified in the user sets of control data.

28. A method as set forth in claim 27, comprising the further steps of updating the usage control element
15 of each data package, reconcatenating after the usage of the data objects, each of the data object and its usage control element, reencrypting each of the concatenated data objects and its usage control element and transferring the repackaged data objects to their creators.

20

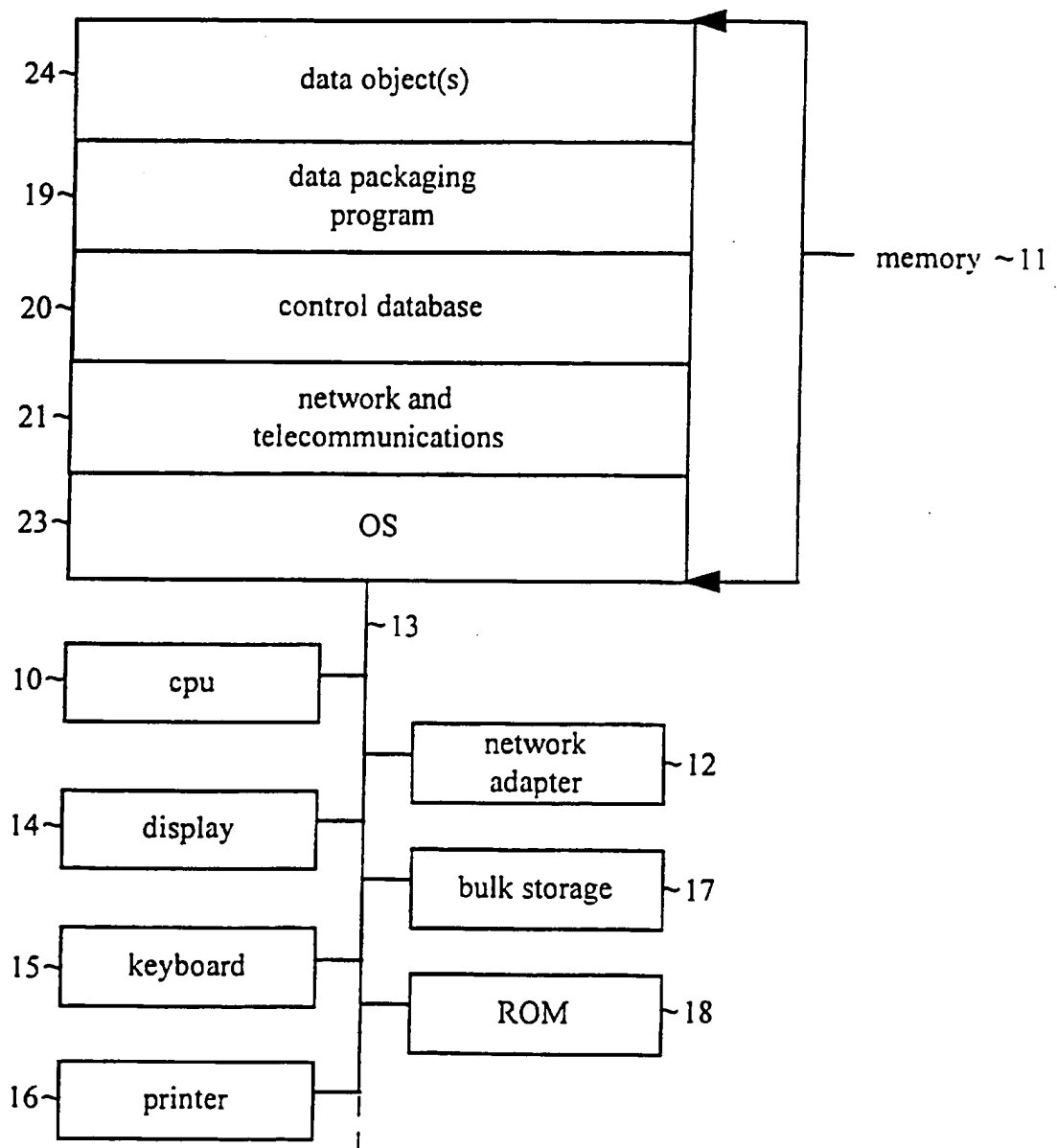
1/15

Fig 1

**SUBSTITUTE SHEET**

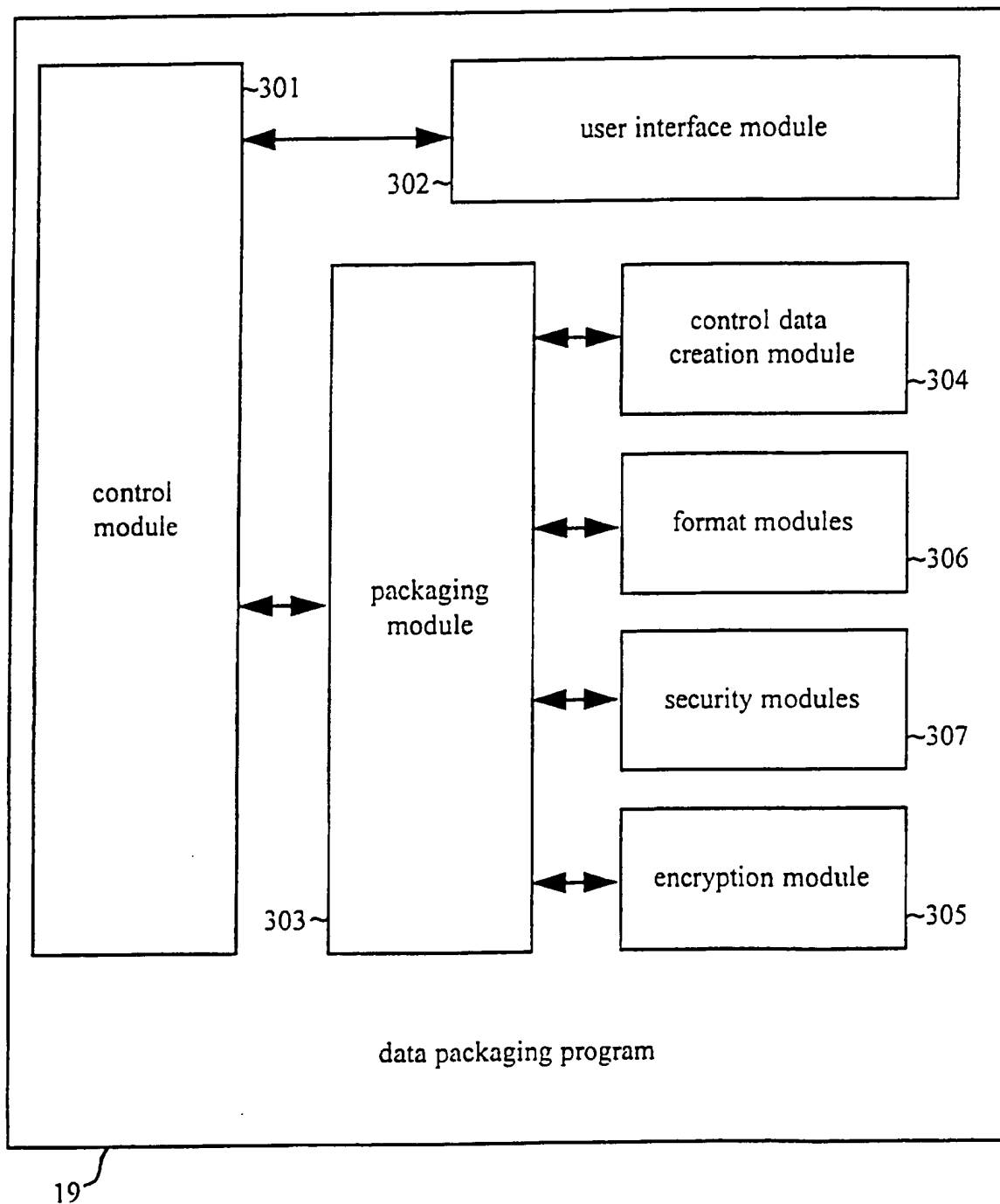
2/15

Fig 2

**SUBSTITUTE SHEET**

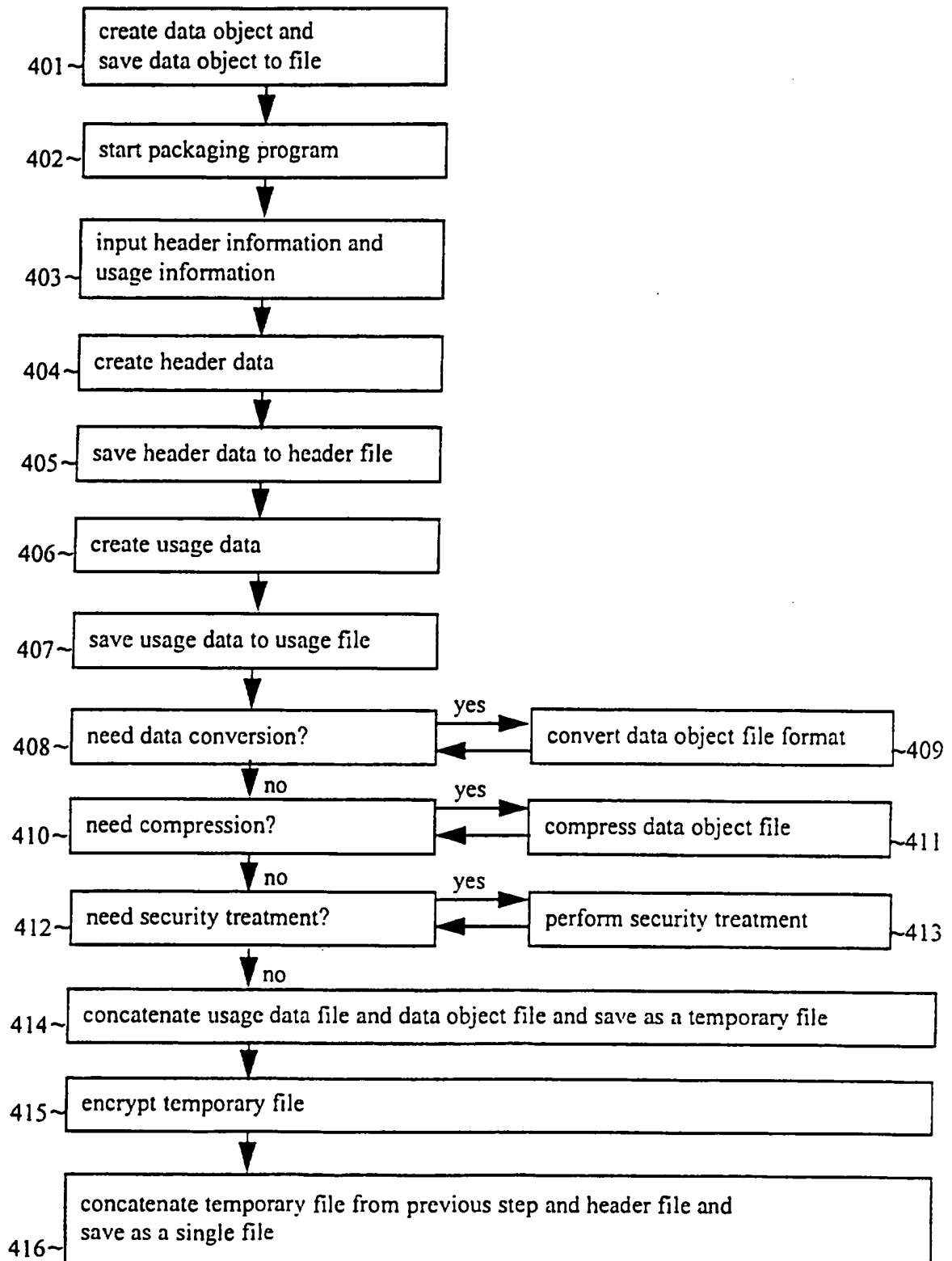
3/15

Fig 3

**SUBSTITUTE SHEET**

4/15

Fig 4



5/15

Fig 5

file identifier	123456789
title	image
format code	a
security code	b

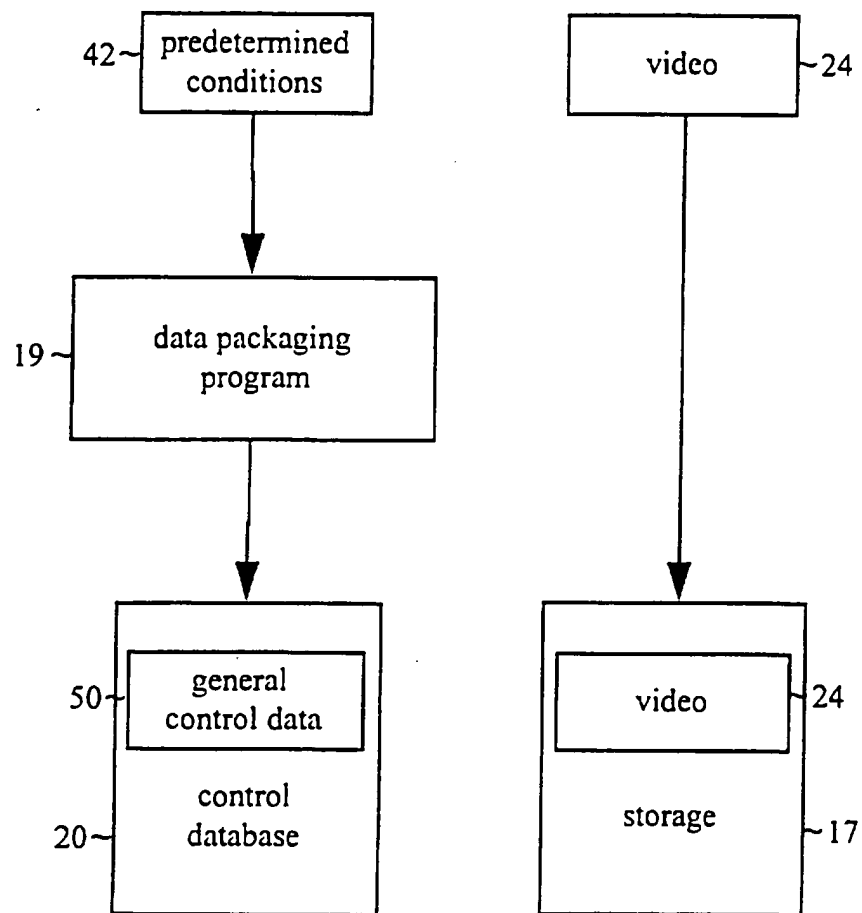
Fig 6

usage element for author's phone number	<table> <tbody> <tr> <td>identifer</td> <td>1</td> </tr> <tr> <td>size</td> <td>13</td> </tr> <tr> <td>data</td> <td>716 381 5356</td> </tr> </tbody> </table>	identifer	1	size	13	data	716 381 5356
identifer	1						
size	13						
data	716 381 5356						
...price for single use	<table> <tbody> <tr> <td>identifer</td> <td>2</td> </tr> <tr> <td>size</td> <td>4</td> </tr> <tr> <td>data</td> <td>.50</td> </tr> </tbody> </table>	identifer	2	size	4	data	.50
identifer	2						
size	4						
data	.50						
...price for unlimited use	<table> <tbody> <tr> <td>identifer</td> <td>3</td> </tr> <tr> <td>size</td> <td>4</td> </tr> <tr> <td>data</td> <td>50.00</td> </tr> </tbody> </table>	identifer	3	size	4	data	50.00
identifer	3						
size	4						
data	50.00						
...code for usage type approved	<table> <tbody> <tr> <td>identifer</td> <td>4</td> </tr> <tr> <td>size</td> <td>2</td> </tr> <tr> <td>data</td> <td>9</td> </tr> </tbody> </table>	identifer	4	size	2	data	9
identifer	4						
size	2						
data	9						
...code for number of usages approved	<table> <tbody> <tr> <td>identifer</td> <td>5</td> </tr> <tr> <td>size</td> <td>2</td> </tr> <tr> <td>data</td> <td>1</td> </tr> </tbody> </table>	identifer	5	size	2	data	1
identifer	5						
size	2						
data	1						

SUBSTITUTE SHEET

6/15

Fig 7

**SUBSTITUTE SHEET**

7/15
Fig 8a

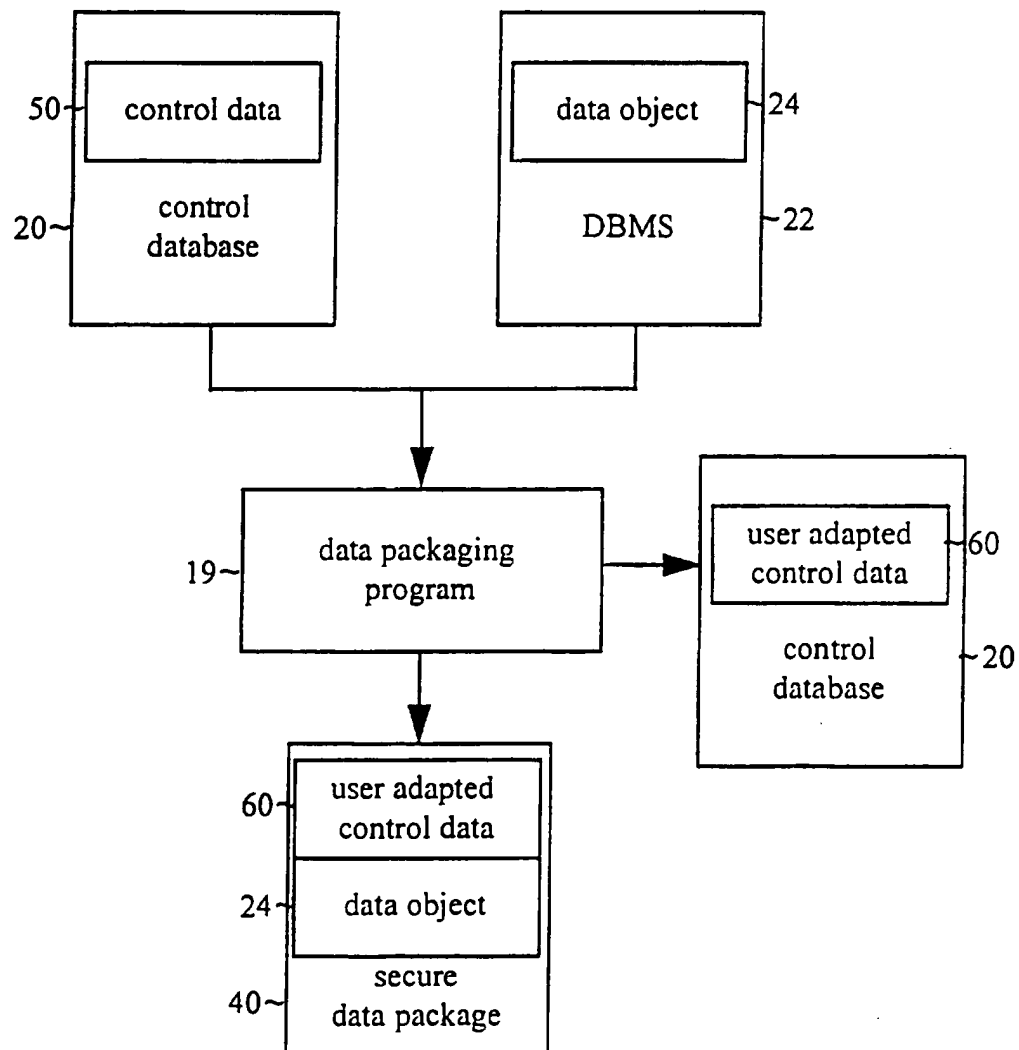
header	object identifier	123456789
	format code	0010
	security code	0010
	number of usage elements	2
	size of usage data	17
	size of data object	273
	1st usage element id	001
	1st usage element size	6
	1st usage element data	1
	2nd usage element id	002
	2nd usage element size	3
	2nd usage element data	

Fig 8b

header	object identifier	123456790
	format code	0010
	security code	0010
	number of usage elements	2
	size of usage data	17
	size of data object	273
	1st usage element id	001
	1st usage element size	6
	1st usage element data	1
	2nd usage element id	002
	2nd usage element size	3
	2nd usage element data	2

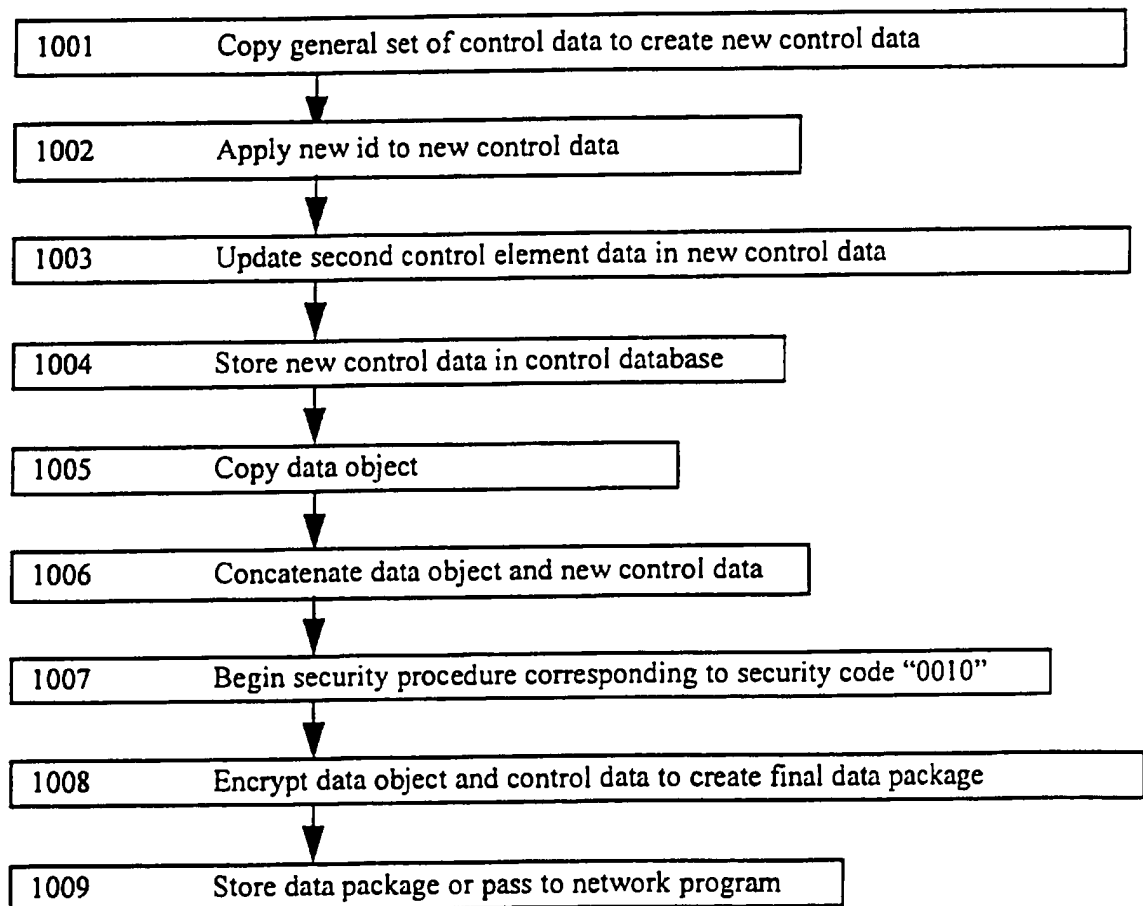
8/15

Fig 9

**SUBSTITUTE SHEET**

9/15

Fig 10

**SUBSTITUTE SHEET**

10/15

Fig 11

control data in memory

123456789001000102172730016100232

AVI file in memory

RIFF0,00AVLIST0000hdrlavih8000j000\$W00CVA
D0000i000000000000P00@000\$0000000A
q00w00000V6LIST0000strlstrh8000vidscvid000æ
\$000({{hhhhhh{On—α\$—α\$hm{hhhh000""{{|sdg™\$
Ç!°,||"8+=000000000000000{On0"0'!\$"—0{Onqvd
000½½½————hhh{On"½!||{On—α\$½||"qvd{On,,%øwm[,
%øw|,"—α\$"½\$Ç!°,||"{{{{{""\$DD\$aaaaaaUUU½

Fig 12a

concatenated control data
and AVI file in memory

123456789001000102172730016100232RIFF0,
 00AVILIST0000hdrlavih8000j000ŠW000CVAD00
 00i000000000000P000@000000000000A700
 w000000V6LISTi0000strlstrh8000vidscvid000αŠ00
 0{{{hhhhhh{On—αŠ—αŠhm{hhhh000""{{{sdc™ŠÇi;,"
 8=+0000000000000000{On0"0'š"-0{Onqvd000
 ½½½————hhh{On'š;,"{On—αŠ;,"qvd{On,,%owm[,,%owj;
 "—αŠ'šÇi;,"{""DDDDaaaaaaUUU½

Fig 12b

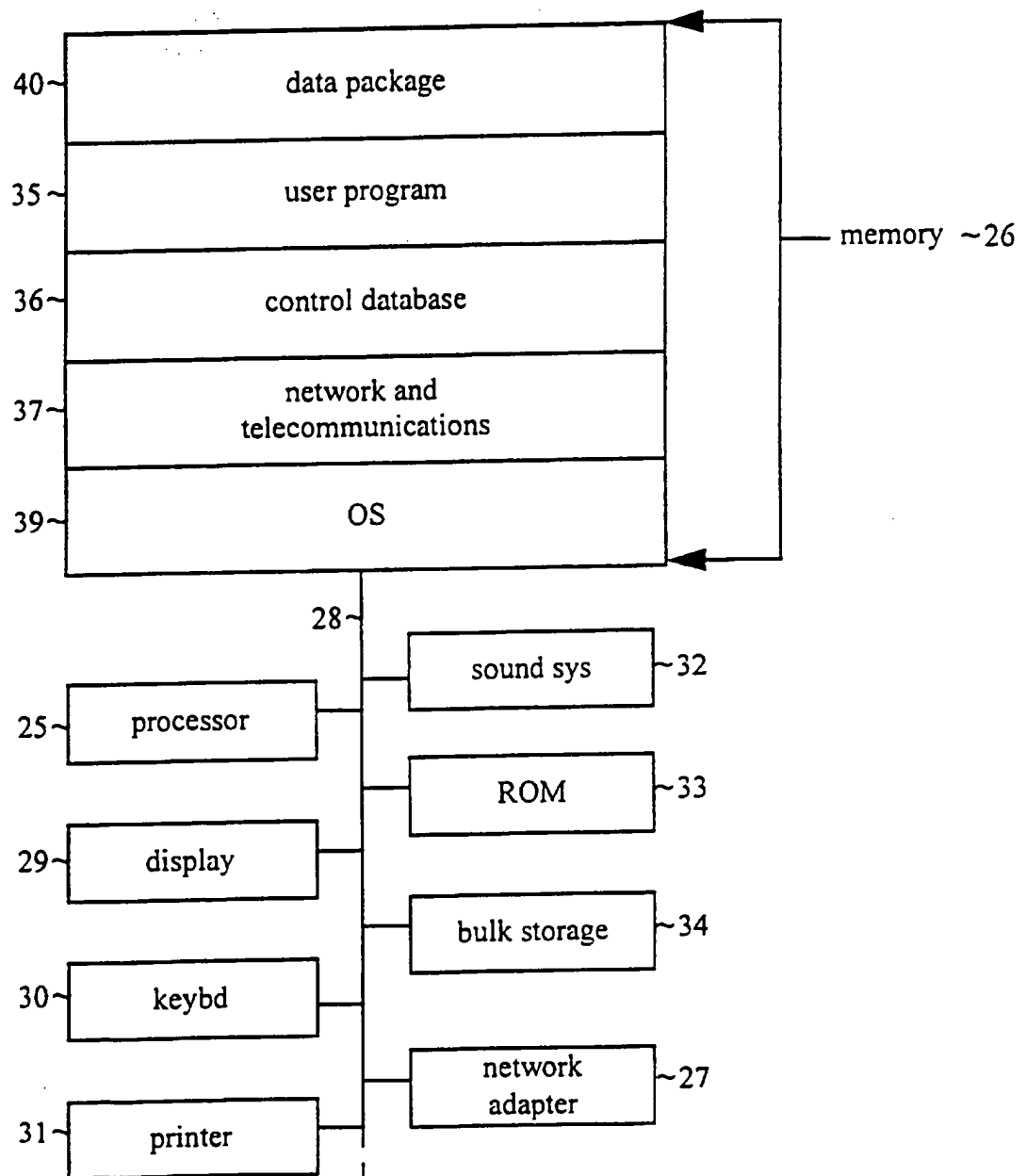
concatenated and encrypted control data and AVI file in memory

[illegible]

SUBSTITUTE SHEET

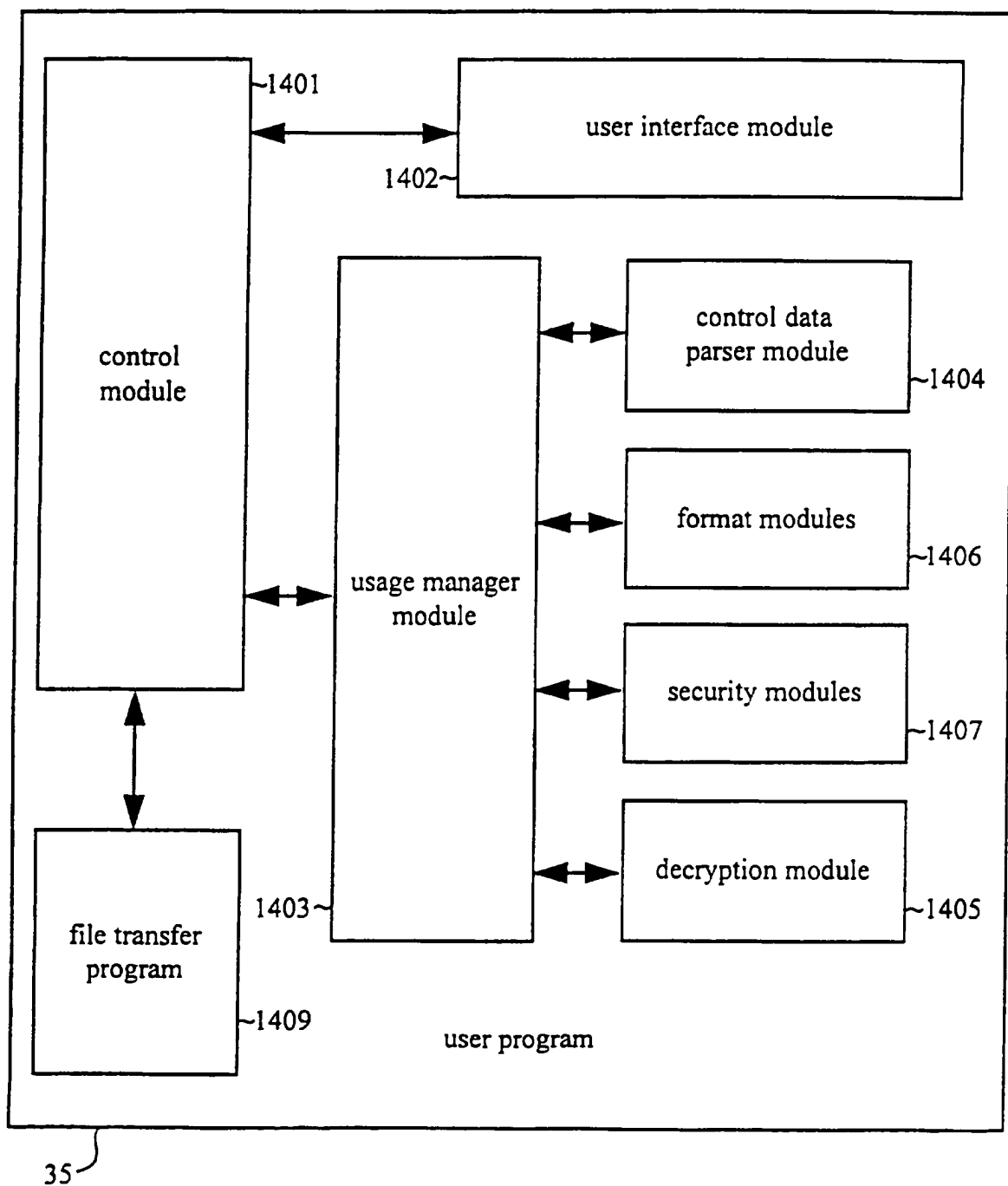
11/15

Fig 13

**SUBSTITUTE SHEET**

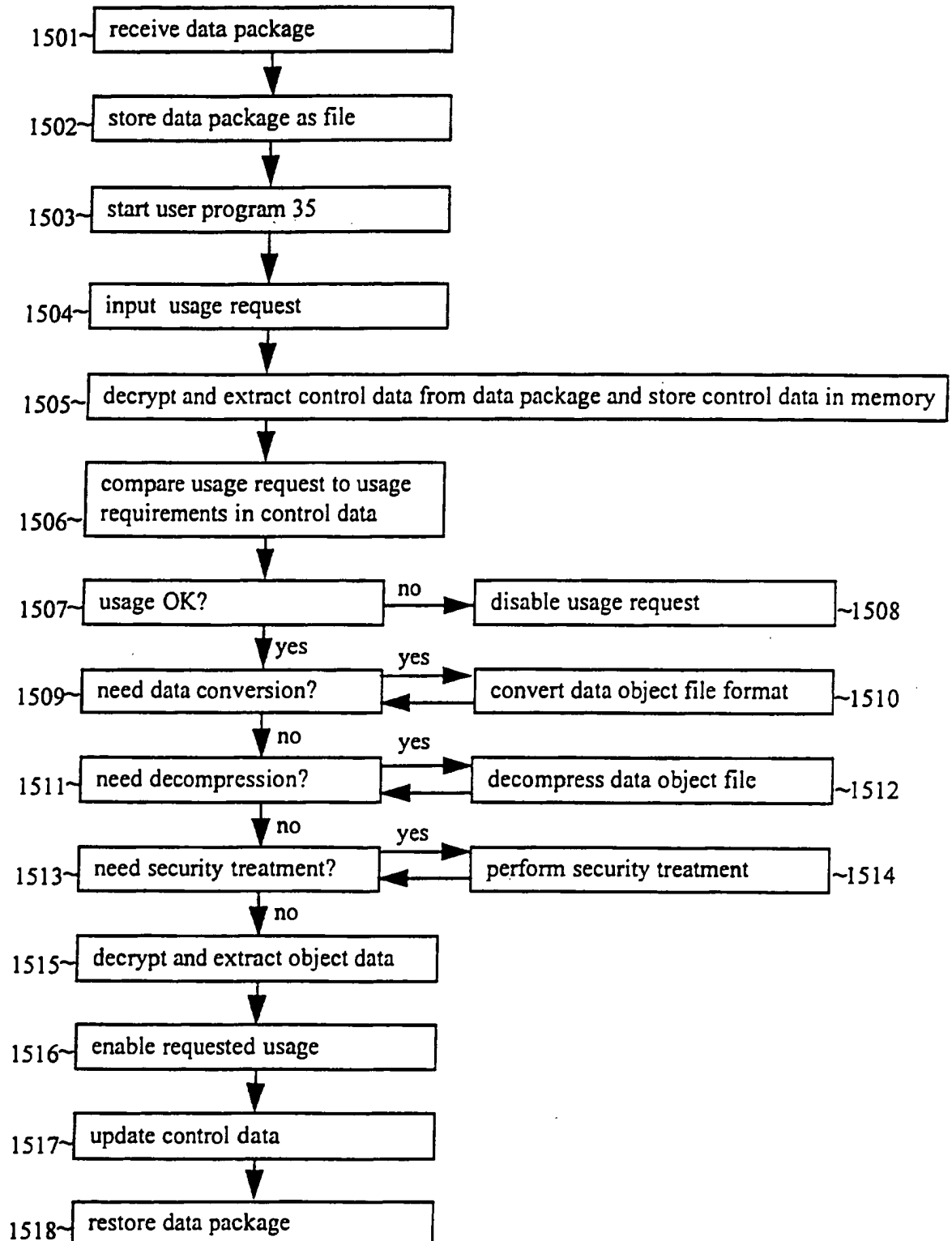
12/15

Fig 14

**SUBSTITUTE SHEET**

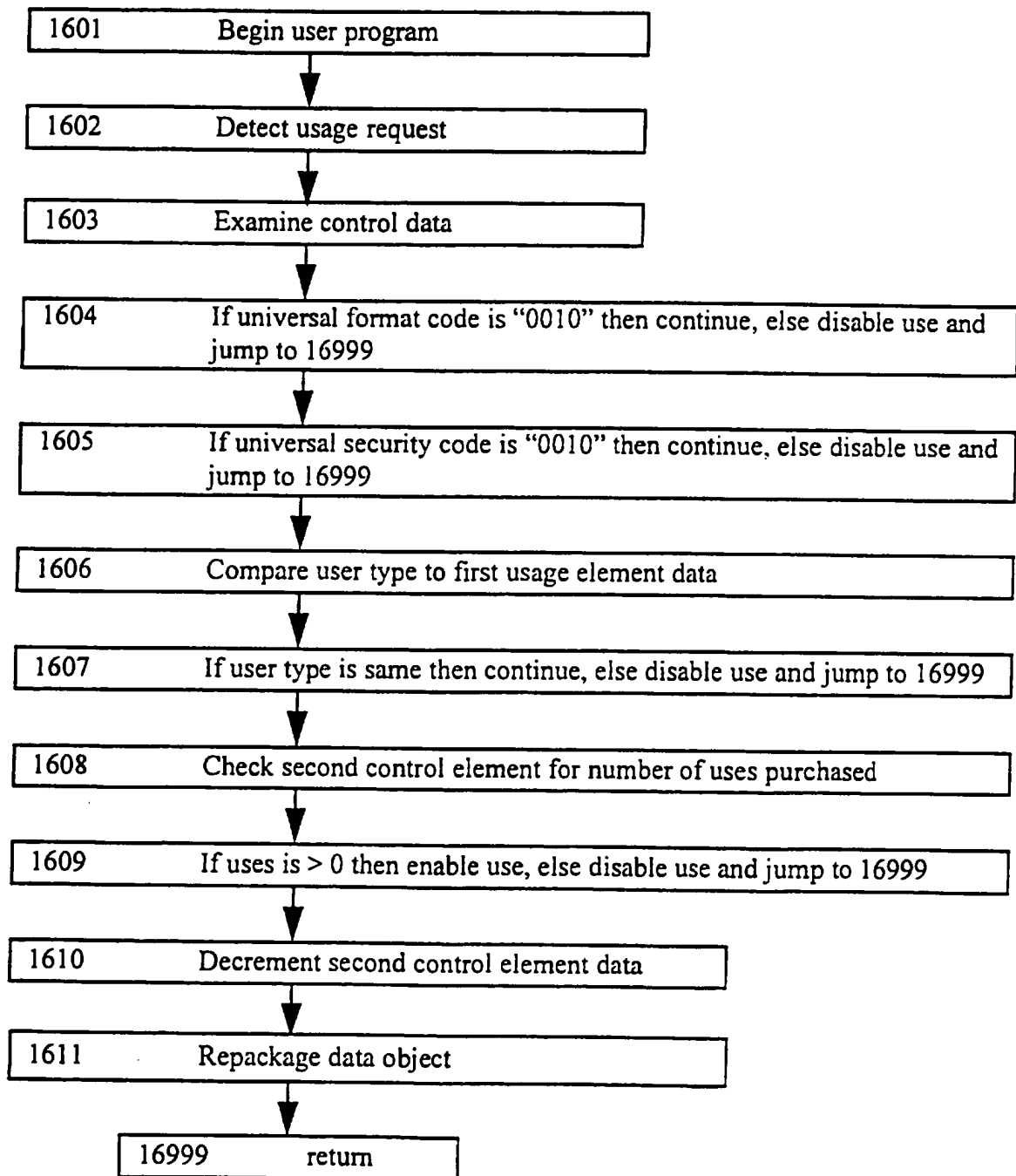
13/15

Fig 15

**SUBSTITUTE SHEET**

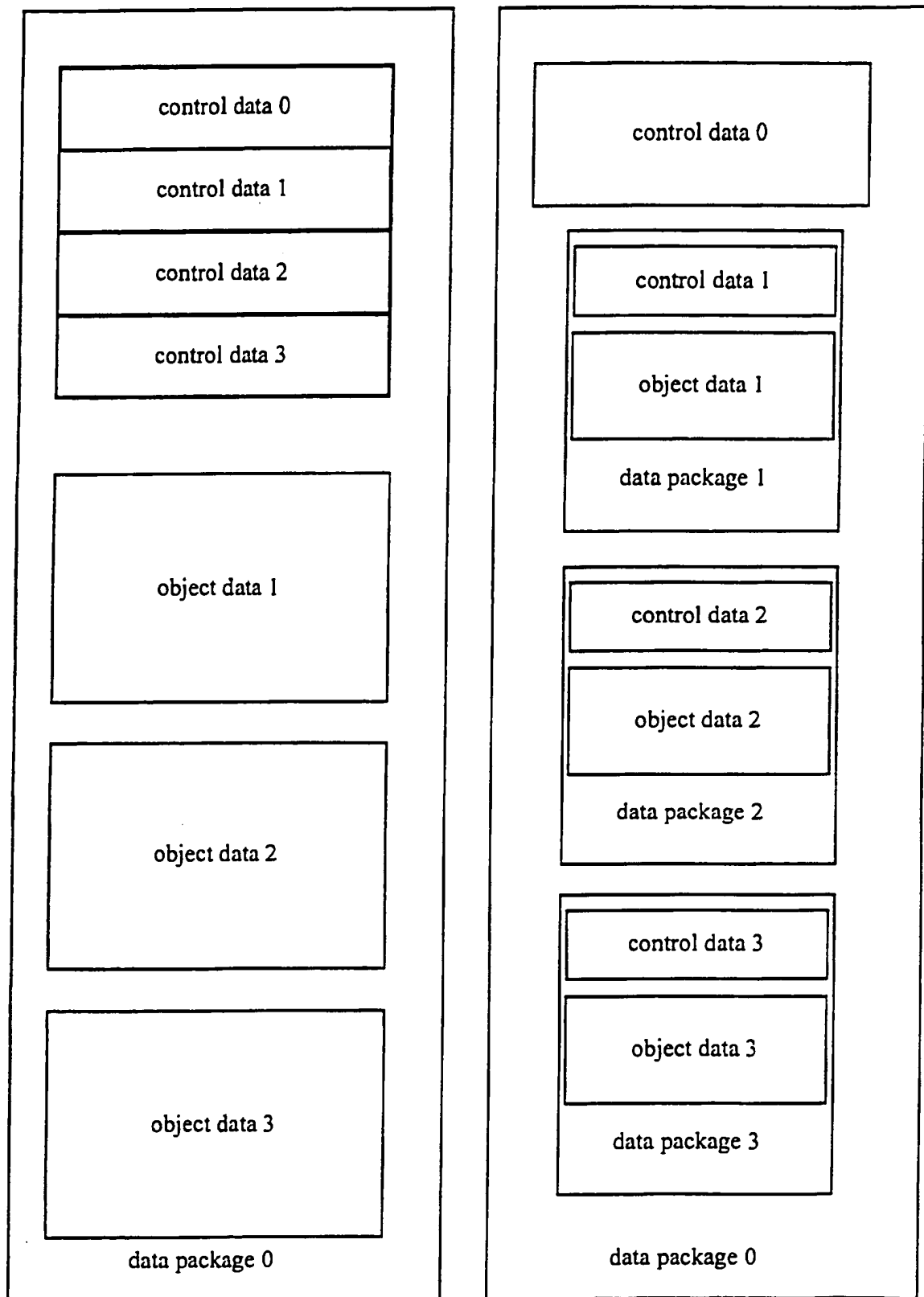
14/15

Fig 16

**SUBSTITUTE SHEET**

15/15

Fig 17

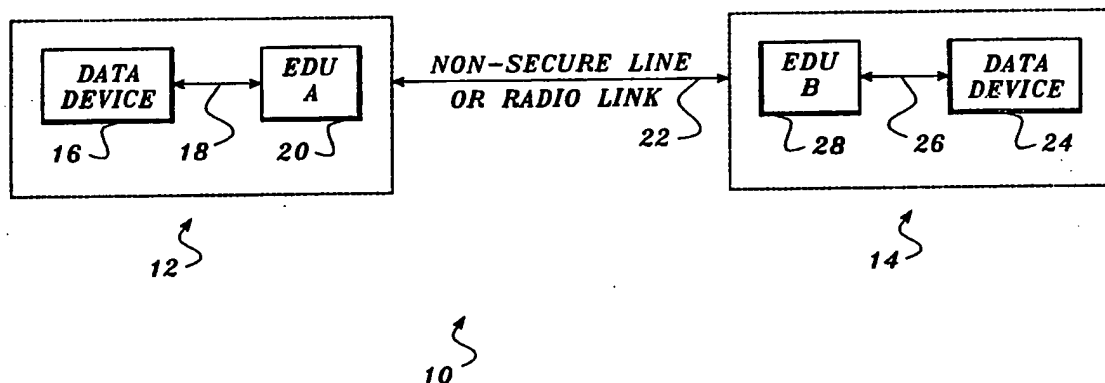




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁵ : H04L 9/00		A1	(11) International Publication Number: WO 94/03003
			(43) International Publication Date: 3 February 1994 (03.02.94)
(21) International Application Number: PCT/US93/04340 (22) International Filing Date: 4 May 1993 (04.05.93) (30) Priority data: 07/917,598 23 July 1992 (23.07.92) US (71) Applicant: CREST INDUSTRIES, INC. [US/US]; 201 Frontage Road North, Suite B, Pacific, WA 98047 (US). (72) Inventors: RASMUSSEN, Harry, Ronald ; 6105-4th Street Court Northeast, Tacoma, WA 98422 (US). LaBOUNTY, Jack, Daley ; 16931 Southeast 32nd Place, Bellevue, WA 98008 (US). ROSENOW, Michael, James ; 1420 Northwest Gillman, Suite 2305, Issaquah, WA 98027 (US).			(74) Agent: ANDERSON, Ronald, M.; Christensen, O'Connor, Johnson & Kindness, 2800 Pacific First Centre, 1420 Fifth Avenue, Seattle, WA 98101-2347 (US). (81) Designated States: AT, AU, BB, BG, BR, CA, CH, CZ, DE, DK, ES, FI, GB, HU, JP, KP, KR, KZ, LK, LU, MG, MN, MW, NL, NO, NZ, PL, PT, RO, RU, SD, SE, SK, UA, VN, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG). Published With international search report.

(54) Title: ENCRYPTION/DECRYPTION APPARATUS WITH NON-ACCESSIBLE TABLE OF KEYS



(57) Abstract

An encryption/decryption unit (EDU) that handles management of encryption keys used in the secure exchange of data over non-secure communication links. Each EDU includes a central processing unit (CPU) that controls its operation, random access memory (RAM) in which tables of key exchange keys (KEKs) are stored, and a data encryption standard (DES) coprocessor that implements a data encryption algorithm developed by the U.S. National Bureau of Standards - all comprising a module that is embedded in a potting material. Attempts to remove the potting material either by mechanical or solvent means are likely to result in loss of the data and program code stored in the module. The CPU includes special circuitry enabling it to operate in an encrypted mode so that it can not be interrogated to discover the program or data stored therein. This program enables the EDU (20) to establish secure communications with another similar EDU (28) over a non-secure link. Each EDU establishing a secure communications session randomly generates a portion of a session data encryption key (DEK) that is encoded by using a KEK from either a public or private table of keys stored in the embedded RAM. The two EDUs exchange the encrypted portions of the DEK, decrypt the portions, and then logically combine them to determine the current session DEK. Use of a stored EDU ID in each EDU comprising the link prevents a third EDU from bridging the link to tap into the communications between two stations.

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	FR	France	MR	Mauritania
AU	Australia	GA	Gabon	MW	Malawi
BB	Barbados	GB	United Kingdom	NE	Niger
BE	Belgium	GN	Guinea	NL	Netherlands
BF	Burkina Faso	GR	Greece	NO	Norway
BG	Bulgaria	HU	Hungary	NZ	New Zealand
BJ	Benin	IE	Ireland	PL	Poland
BR	Brazil	IT	Italy	PT	Portugal
BY	Belarus	JP	Japan	RO	Romania
CA	Canada	KP	Democratic People's Republic of Korea	RU	Russian Federation
CF	Central African Republic	KR	Republic of Korea	SD	Sudan
CG	Congo	KZ	Kazakhstan	SE	Sweden
CH	Switzerland	LI	Liechtenstein	SI	Slovenia
CI	Côte d'Ivoire	LK	Sri Lanka	SK	Slovak Republic
CM	Cameroon	LU	Luxembourg	SN	Senegal
CN	China	LV	Latvia	TD	Chad
CS	Czechoslovakia	MC	Monaco	TC	Togo
CZ	Czech Republic	MG	Madagascar	UA	Ukraine
DE	Germany	ML	Mali	US	United States of America
DK	Denmark	MN	Mongolia	UZ	Uzbekistan
ES	Spain			VN	Viet Nam
FI	Finland				

ENCRYPTION/DECRYPTION APPARATUS WITH NON-ACCESSIBLE TABLE OF KEYS

Field of the Invention

The present invention generally pertains to apparatus for encrypting and
5 decrypting data, and more specifically, to apparatus for implementing the encryption
and decryption process with secret encryption keys.

Background of the Invention

Procedures for encrypting and decrypting data for transmission over non-
secure radio or telephone links have been highly refined to meet the needs of the
10 military and industry. An encryption algorithm that is virtually unbreakable in any
reasonable time frame, by even the most powerful of high-speed computers, has been
developed and published by U.S. National Bureau of Standards and sanctioned for use
by industry in this country as an acceptable method for protecting computerized data
conveyed over non-secure channels. In fact, integrated circuits designed specifically
15 for encryption and decryption of data in accordance with this Data Encryption
Algorithm (DEA) are readily available from several vendors, such as Western
Digital™. The algorithm, like most encryption schemes, uses an encryption key to
encrypt data. Successful use of the DEA, and almost any other encryption/decryption
algorithm commonly employed, requires that the station receiving the encrypted
20 transmission have the same key used to encrypt the data in order to decrypt it.
Accordingly, no unauthorized party should know or have access to the encryption key
that is being used.

Unfortunately, for any prior art encryption/decryption system using the DEA
or similar algorithms, extensive security measures are required for managing and

periodically changing the encryption keys that are used. Any third party that gains access to the encryption key being used to encrypt data can tap into a non-secure line over which encrypted messages are transmitted and then use the key to decrypt messages that are intercepted. Even if knowledge of the encryption key used is limited to those operating the encryption/decryption equipment, there can be no assurance that others outside an organization will not breach security and learn the encryption key due to failure of someone in the organization to follow security procedures. As the size of a network over which secure communications must be maintained expands, the difficulty in managing the encryption keys used on the network grows exponentially.

Since any person with access to the encryption keys can breach the security of encrypted communications between members of the network, encryption keys must be changed on a regular basis. Frequent changes in the encryption keys in use minimizes the risk of disclosure by individuals that previously had access to the keys. However, any such change requires that the new encryption keys be distributed to all stations in the network. Typically, the new encryption keys are hand carried to each station site by bonded couriers; nevertheless, it is possible that a courier may compromise security. Even if a security breach does not occur, the cost of regularly distributing encryption keys to each station of a large network in this manner may be prohibitive.

For these reasons, it is preferable to use encryption keys at each station in a network that are not known to anyone, even those operating the encryption/decryption apparatus. Various techniques have been developed to access encryption keys stored in an electronic memory for this purpose. For example, a new encryption key can be selected for subsequent encryption of communications between stations based on the last encryption key that was used, by applying a secret formula to generate the new key. However, if the formula is discovered or otherwise becomes known by someone who is outside the organizational network, security of the encryption system is breached, since that person can generate the encryption keys that will subsequently be used, simply by applying the formula to any previously discovered key.

Clearly, it would be preferable to randomly generate the encryption key that is used to encrypt data transmitted to another station each time that communications are initiated. Yet, random generation of an encryption key at one station inherently renders the receiving station unable to decrypt the message, because it does not have the encryption key used. What is therefore required are means for transmitting the encryption key from one station to another in an encrypted form, with some provision

that enables the receiving station to decrypt the encryption key. Prior art encryption/decryption apparatus do not provide means to accomplish this task in an efficient manner that is not easily circumvented. Any key exchange key (KEK) that is used in the process of transferring an encryption key for encrypting and decrypting the message to the other station must be available to both stations, but can not be available to anyone outside the secure network of stations. Even if the encryption apparatus is available to someone outside the organization, it should be virtually impossible to discover the KEKs used by stations comprising the network, if secure communications are to be maintained.

The foregoing aspects and many of the attendant advantages of this invention over the prior art will become more readily appreciated as the same becomes better understood by reference to the following detailed description, when taken in conjunction with the accompanying drawings.

Summary of the Invention

In accordance with the present invention, encryption/decryption apparatus for ensuring secure communications between two stations include encryption processor means for encrypting and decrypting data using a session data encryption key (DEK) that is input thereto. Control means coupled to the encryption processor means are provided for controlling the operation of the encryption processor means. The control means supply the encryption processor means with the data for encryption and decryption and with an encryption key for use in encrypting and decrypting the data to produce an output signal in response to programmed instructions. These programmed instructions cause the control means to automatically randomly select a part of a session DEK and to combine it with another part of the session DEK received from the other station to determine the session DEK that will be used by the encryption processor means to encrypt data. Non-volatile memory means that are coupled to the control means store a plurality of key encryption keys that are used by the encryption processor means in encrypting a part of the session DEK for transmission to the other station. The control means select the key encryption key from the plurality of key encryption keys as a function of a check value determined with the part of the session key.

Within the non-volatile memory means is disposed an internal power source that provides electrical power to maintain storage of the plurality of key encryption keys. Potting means encapsulate the encryption processor means, the control means, and the non-volatile memory means in a radio and light wave opaque material that is sufficiently hard and resistant to dissolution by solvents to prevent its removal without

damage to interconnections coupling the non-volatile memory means to the control means and damage to interconnections supplying electrical power to the non-volatile memory means from the internal power source. Such damage causes erasure of the plurality of key encryption keys stored in the non-volatile memory means. In addition, the control means respond to any attempt to externally interrogate the non-volatile memory means by causing erasure of the key encryption keys stored therein.

Multiplexer means are coupled to the control means to receive a data signal and a select signal therefrom, and are also coupled to the encryption processor means, an output port, and the memory means; the multiplexer means selectively convey the data signal to one of the encryption processor means, the output port, and the memory means, in response to the select signal. The control means include a non-volatile memory for retaining program steps and a unique identification code that identifies a specific encryption/decryption apparatus. In addition, the control means include means for locking the control means and its non-volatile memory to prevent data and program steps from being read externally after storage of the program steps in the non-volatile memory is complete. The means for locking include means for encrypting data and memory addresses defining memory storage locations within the non-volatile memory of the control means and within the non-volatile memory means.

Brief Description of the Drawings

FIGURE 1 is a block diagram of a communications network comprising two stations, each provided with an encryption/decryption unit (EDU) in accordance with the present invention, thereby enabling the stations to establish secure communications over a non-secure line or radio link;

FIGURE 2 is a schematic block diagram of one of the EDUs shown in FIGURE 1;

FIGURE 3 is a flow chart illustrating the logical steps implemented at one station by the EDU in selecting and encrypting a first portion of a session encryption key for transmittal to another station;

FIGURE 4 is a flow chart illustrating the logical steps implemented by the EDU at the other station in decrypting the first portion of the session encryption key, and in selecting and encrypting a second portion of the session encryption key for transmittal to the one station; and

FIGURE 5 is a flow chart illustrating the logical steps implemented by the EDU at the one station to decrypt the second portion of the session encryption key.

Detailed Description of the Preferred Embodiment

As noted above, one of the more difficult problems in establishing and maintaining an encrypted communication network is distributing secure DEKs to each station in the network on a regular basis. In FIGURE 1, a simple network for carrying out encrypted communications is shown generally at reference numeral 10. Network 10 is shown simply as two stations, including a station 12 and a station 14, but it will be appreciated that the network can comprise many other such stations.

Both stations 12 and 14 use similar components for encrypting and decrypting communications. For example, station 12 includes a data device 16, which may, for example, comprise a facsimile machine or personal computer (neither shown separately). Data device 16 is connected through lines 18 to an EDU A 20. Station 12 uses EDU A 20 to establish secure communications over a non-secure line (or radio link) 22 with station 14, which includes an EDU B 28. EDU B 28 is connected to a data device 24 over lines 26. Data device 24 is the same type of device as data device 16. Thus, if data devices 16 and 24 are facsimile machines, communications network 10 permits secure communication of facsimile information in an encrypted form between stations 12 and 14 over non-secure line 22.

Because of the manner in which secure communications are established between EDU A 20 and EDU B 28, tapping into non-secure line 22 using a similar EDU (not shown) would NOT enable a third party to breach secure communications between stations 12 and 14. In the preferred form of the present invention, communications between EDU A 20 and EDU B 28 are carried out using a session encryption key that is changed with each session and comprises two parts, one part randomly selected by EDU A 20, and the other part randomly selected by EDU B 28. Thus, the present invention comprises the EDU at each of the communicating stations 12 and 14. In establishing secure communications between two stations 12 and 14, the EDU at each station randomly select its respective portion of the session encryption key, encrypts that portion of the session encryption key, and transmits the encrypted respective portion of the session encryption key to the other station. Once both station 12 and station 14 have decrypted the portion of the session encryption key developed by the other station, the two portions are logically combined at each station to produce the complete or final session encryption key used for encrypting data transmitted between stations 12 and 14 during the current session. In addition, the EDUs are preprogrammed to ensure that the intended station in a two-way communication link is actually receiving or transmitting the encrypted data, to guard against a third party tapping into non-secure line 22 with another EDU. The EDUs

also ensure that the two portions of the session encryption key that are exchanged between stations 12 and 14 are correctly received and decrypted, thereby protecting against data errors that might have arisen in the transmission of the encrypted portions of the session encryption key between the two stations or in their decryption.

5 A block diagram of EDU 20 is shown in FIGURE 2; EDU 28 is exactly the same, except for having a different EDU identification number stored within it. EDU 20 includes a potted module 30 and an external input/output (I/O) bus 32 for providing interconnections between the EDU and the data device (or to other
10 apparatus) that will provide the data to be encrypted or will receive the data that is decrypted by the EDU. Module 30, which comprises virtually the entire EDU, is encapsulated within a radio opaque and light opaque potting compound 34 to prevent discovery of the internal circuitry and to prevent forced electromagnetic or visual
15 tapping, monitoring, or other forms of penetration that might be attempted to uncover encryption keys and other information included therein. The potting compound is sufficiently hard and resistant to abrasion to prevent its removal without damaging the components comprising the EDU or at least causing loss of important data stored therein. Of greatest sensitivity to maintaining the security of communications between
20 EDUs comprising a network is the need to protect against discovery of KEKs that are encrypted using a key that is unique to each EDU and is assigned to it when it is initialized. The encrypted KEKs are stored as tables within each EDU and are utilized for encrypting portions of the session encryption key that are exchanged between two stations and subsequently logically combined at each EDU to produce a session DEK that is used for encryption of data exchanged over non-secure line 22. To avoid
25 breaching the security of communications on network 10, it is absolutely imperative that these KEKs not become publicly known.

 In the preferred form of module 30, two sets or tables of KEKs are stored in encrypted form in a random access memory (RAM) 42. One set is called a "public" set, since each EDU that will be sold will include this set. The other set is a "private"
30 set of KEKs, which optionally may be randomly generated by a user for distribution to and storage in those EDUs comprising a private network of stations. The significance of the KEKs will be apparent from the description that follows. Any attempt to expose the internal circuitry of module 30 by use of a chemical, solvent, or mechanical means in order to access RAM 42 electronically or physically so as to access these
35 data will cause loss of the KEKs that are stored therein. RAM 42 preferably comprises a Dallas Semiconductor™ type DS 1213 smart socket in which is installed

a memory integrated circuit (not separately shown) comprising 128K x 8 bits of storage, i.e., yielding 1,048,576 bits of non-volatile static RAM organized as 131,072 words by 8 bits. This memory integrated circuit is a dual in-line package (DIP) package configuration of generally conventional design, but the smart socket contains
5 an internal battery supply (not separately shown) sufficient to maintain data integrity in the absence of externally applied power for a period in excess of 10 years. Dallas Semiconductor also supplies an integrated circuit non-volatile memory device that includes an integral internal battery supply, and this type of device can be used in place of the smart socket and more conventional memory device combinations. In the
10 event a chemical solution is used to dissolve potting compound 34 in an attempt to discover the KEKs stored in RAM 142, the material comprising RAM 142 (smart socket or memory device that includes the integral internal battery supply) will also be dissolved, thereby disconnecting the internal battery supply and erasing the KEKs stored therein.

15 Operation of module 30 to establish and conduct secure communications is controlled by a CPU 36, which includes 32K of embedded RAM (not separately shown). In the preferred embodiment, a Dallas Semiconductor™ type DS 5000 microchip integrated circuit is used for CPU 36. The DS 5000 integrated circuit includes non-volatile embedded RAM (not separately shown) and all information and
20 programming stored therein are preserved in the absence of an externally applied voltage for up to 10 years. In addition, the internal data registers and key configuration registers of the DS 5000 integrated circuit are non-volatile. Data stored within the embedded RAM that comprise program steps carried out by CPU 36 in establishing secure communications can be modified after encapsulation of module 30
25 has been accomplished with potting material 34; however, initial loading of the embedded RAM within the DS 5000 microchip comprising CPU 36 is accomplished with a conventional universal asynchronous receiver/transmitter (UART) interface (not shown) that is connected through external I/O bus 32 by lines 76. In addition, control lines 50 connect CPU 36 to external I/O bus 32 and convey write, read,
30 interrupt, and signals for ports 0-3 (P1.0-P1.3) of the CPU.

Data lines (D0-D7) 54 interconnect CPU 36 with RAM 42 and with a buffer 46. Buffer 46 comprises an SN 74HCT245 octal bus transceiver with a three-state output that is used to block external access to internal data transfers occurring within module 30, thereby preventing an external device from accessing KEKs stored
35 in RAM 42 and other data transferred between components of the module. Buffer 46 is enabled via control signals supplied over a line 74 by CPU 36 when it is appropriate

to allow bi-directional data transfer to and from external I/O bus 32 through lines 52, and therefore to and from an external device.

To provide additional security, CPU 36 operates in an encrypted mode. The encrypted mode is deactivated prior to the initial loading of program steps and data into the embedded RAM of CPU 36. Before the initial loading of program code and data begins during manufacture of the EDU, a 40-bit encryption mode key is selected for use by CPU 36 in the encrypted mode. A data encryptor circuit and an address encryptor circuit (neither separately shown) within CPU 36 respectively control the form in which the program code is stored in the embedded RAM of the CPU and the addresses at which it is stored. As the initial loading of program steps is performed, the data encryptor circuit uses the 40-bit encryption mode key to transform or encrypt opcodes, operands, and data bytes at each memory location defined by the software. Similarly, the address encryptor circuit uses the encryption mode key in a different encryption algorithm to translate or encrypt a logical address of each data byte location into an encrypted address at which the data are actually stored. The contents of the embedded RAM are then verified, and the encrypted mode is enabled by setting a security lock bit. After the security lock bit is set to enable operation in the encrypted mode, the contents of the CPU's embedded RAM is unintelligible to an observer that might attempt to tap into its circuitry to discover the program code and other data stored therein. The address and data encryptor circuitry provides real time translation or decryption of program code and address locations to CPU 36 during subsequent operation of the EDU. Only program code and data stored in the CPU's embedded RAM that does NOT affect secure operation of the EDU can be changed after the security lock bit is set. Any attempt to externally interrogate the CPU to discover the 40-bit encryption key causes its erasure, rendering the contents of the embedded RAM useless. Even if the encrypted program code and data are thereafter read back from the embedded RAM in CPU 36, they can not be decrypted without the 40-bit encryption mode key, which is lost.

CPU 36 selects a specific storage location for a KEK within RAM 42 by setting 16 address bits. Lines 58 connect CPU 36 to a latch 44, and lines 60 connect latch 44 to RAM 42. To minimize the total number of pins required on CPU 36, the first eight address bits (A0-A7) and eight bits of data (D0-D7) use the same pins on CPU 36. These address bits and data are alternatively passed between CPU 36, latch 44, and RAM 42 over lines 58 and 60, respectively. The eight most significant bits of the address are conveyed on lines 56b directly from CPU 36 to RAM 42 and to external I/O bus 32. The least significant eight address bits (A0-A7) are carried on

lines 56a. In the preferred embodiment, the 16 address bits are available on lines 56 at external I/O bus 32 to address the embedded RAM in CPU 36 when it is initially loaded or subsequently modified.

Although CPU 36 controls the operation of module 30, the actual encryption
5 and decryption of data is implemented by a data encryption standard (DES) coprocessor 38. DES coprocessor 38 is designed to encrypt and decrypt 64-bit blocks of data using the algorithm specified in the Federal Information Processing Data Encryption Standard (No. 46). A transfer rate of 807 kilobytes per second is implemented by DES coprocessor 38 under the control of a 10 megahertz clock
10 circuit 48, to which the DES coprocessor is connected through lines 70. Data are transferred between CPU 36 and DES coprocessor 38 over lines 72. In the preferred embodiment, a Western Digital™ type DES WD20C03A integrated circuit is used for DES coprocessor 38, although other such devices are available from other suppliers. A decoder/multiplexer (MUX) 40 is connected through lines 68 to DES
15 coprocessor 38 and through lines 66 to CPU 36. Decoder/MUX 40 is a three-line to eight-line circuit that decodes one of eight lines, dependent upon three binary select inputs and three enable inputs. Lines 66 carry the three binary select signals and the output signal from decoder/MUX 40 and line 68 carries selectable input 7. In addition, lines 62 carry selectable inputs 5 and 6 from RAM 42, while lines 64, which
20 extend between decoder/MUX 40 and external I/O bus 32 convey selectable inputs 0-4.

The embedded non-volatile RAM in CPU 36 is loaded with the appropriate program steps for controlling the operation of EDU 20 at the time module 30 is manufactured. In addition, RAM 42 is loaded with a set of 65,535 public KEKs that
25 are randomly generated from over 72 quadrillion possibilities. Each EDU that is thus produced stores the same table of 65,535 randomly generated public encryption keys. Any EDU can establish secure encrypted communications with any other EDU using the public KEKs. Also stored in RAM 42 is a user-generated table of over 65,535 randomly generated private encryption keys. These private KEKs are used for
30 initiating secure communications with another EDU in the private network that has the same table of private KEKs stored within its RAM 42.

The steps involved in establishing secure communications between two EDUs are shown in FIGURES 3, 4, and 5. Not shown are any handshaking steps necessary to connect two EDUs in communication with each other so that data for a specific
35 device can be transmitted between them. Preferably, such handshaking steps are

implemented by transmitting predefined data blocks between the two devices, but do not necessarily require action by either EDU.

In FIGURE 3, a flow chart 100 identifies the steps taken by EDU B 28, acting as the intended recipient, to establish secure communications. It will be apparent the steps in flow chart 100 could also be carried out by EDU A 20; however, the choice was made in the preferred embodiment to have the receiving station start the process of determining a session data encryption key, thereby avoiding the possibility that a third party posing as another station might tap into the unsecured line with an EDU to initiate the secure communications link. The method begins with a start block 102. In a block 104, EDU B 28 generates a 64-bit random data encryption key 1 (DEK1), which is one of over 72 quadrillion possible data encryption keys (i.e., all possible combinations of 56 bits).

The DEK1 is the first portion of a session data encryption key that will be subsequently used for transmitting encrypted data between the two EDUs. In a block 106, EDU B 28 then uses the DEK1 as the encryption key in implementing the DEA to encrypt one block of data. The use of the DEA to encrypt a single block of data is referred to as an electronic code book (ECB) method and is carried out by DES coprocessor 38 under the control of CPU 36. The ECB method employs the key (DEK1) to encrypt a 64-bit zero function, i.e., a function comprising 64 logical zeros, the result being used to determine a check value.

In a block 108, a KEK table entry value KEK1 comprising the 16 least significant bits (LSBs) of the 64-bit check value from block 106 is determined. The EDU uses the public or private table for KEKs, as specified by EDU A 20 during the handshaking that preceded establishing the secure communications link. The public table and private table of KEKs each represent a linear array of data, that can be taken in groups of four 16-bit words or 64-bits at a time, to define a KEK. The 16 LSBs of the check value determine the starting point or table entry value in the selected table to determine the 64 bits used as a KEK, as indicated in a block 110. Using the 64-bit KEK selected from the table as the encryption key, the EDU encrypts the value DEK1 using the ECB method in a block 112. A cyclic redundancy check (CRC) value for the KEK table that was selected is then determined in the conventional manner.

In a block 114, the EDU encrypts the KEK table CRC, its own EDU ID number (which is stored in within module 30 and is not user modifiable), and the KEK1 entry value using a predefined header encryption key and the ECB method to produce an encrypted key header. The header encryption key is stored in the embedded RAM within CPU 36 at the time that its programming is initially loaded

and is the same for each EDU. In a block 116, the EDU transmits the encrypted key header and encrypted DEK1 to EDU A 20, which initiated the communication. Although both parts of this transmission are encrypted, they are encrypted at different levels of security, since the encrypted key header is always sent encrypted with the same predefined (although inaccessible) key and the encrypted DEK1 uses a different key with virtually every communication session between two EDUs. The method for establishing secure communications continues with the other EDU, at a block B1 118.

In FIGURE 4, a flow chart 120 shows the steps carried out by EDU A 20 (the EDU that initiated the communication). Flow chart 120 begins at block B1 118 and proceeds to a block 122 wherein the encrypted key header and encrypted DEK1 received from EDU B 28 are parsed. In a block 124, the encrypted key header is decrypted using the predefined header encryption key with the ECB method, enabling the EDU to determine the KEK table CRC, the encoded EDU ID number of the EDU that transmitted the encrypted header, and KEK1.

A decision block 126 causes the CPU to determine if the KEK table CRC is correct, thereby ensuring that the KEK table used to encrypted the header is the same as the KEK table that will be used by EDU A 20. This step prevents two EDUs from attempting to communicate if they are using different private KEK tables or if the public table in used by one has become corrupted or is different than the normal public table of KEKs for some other reason. If the CRC value does not match the expected value, a block 128 stops communication between the EDUs. Under most circumstances, however, the KEK table CRC is correct and the logic proceeds to a block 130.

In block 130, EDU A 20 determines the 64-bit KEK that was previously selected from the public or private table by EDU B 28, using the KEK1 value that it just received as an offset to enter the table. The 64-bit KEK is then used with the ECB method to decrypt the value DEK1, as shown in a block 132.

In a block 134, a validity check is made to ensure that the decryption process was carried out correctly and that the encrypted data were not affected by noise or other problems during transmission. The validity check is carried out by using the decrypted DEK1 value and the ECB method to encrypt the 64-bit zero function. The result provides a check value, the 16 LSBs of which are a value KEK1'. The accuracy of the encryption/decryption process and transmission is confirmed in a decision block 136 if the EDU determines that KEK 1 equals KEK 1'. If not, a block 138 provides for indicating that an error has occurred in establishing secure communications, which leads to a stop block 140.

On the other hand, assuming that KEK 1 equals KEK 1', a block 142 directs the EDU to generate a 64-bit random value, DEK2, which is the second portion of the session data encryption key that will be used to encrypt data transmissions between the two EDUs. In a block 144, EDU A 20 performs a logical XOR to combine the first portion of the session key, DEK1, and the second portion, DEK2, to determine the final session data encryption key DEK.

In a block 146, DEK2 is used with the ECB method to encrypt the 64-bit zero function in order to determine a second check value. Using the 16 LSBs of the check value in a block 148, the EDU determines a table entry value KEK2. By entering the specified public or private table at the address offset determined by KEK2, four consecutive 16-bit words comprising a 64-bit KEK are determined in a block 150. The EDU uses the value of KEK from the table and the ECB method to encrypt DEK2 in a block 152.

With the predefined header encryption key, the EDU A 20 encrypts the KEK table CRC, its own EDU ID, and the table entry value KEK2, producing an encrypted key header in a block 154. The encrypted key header just produced and the encrypted DEK2 will be transmitted to EDU B 28 only if the next test is passed in a decision block 155.

Decision block 155 now determines whether the EDU ID that was decrypted from the header received from EDU B 28 in block 124 matches that of the EDU that was initially called, i.e., confirms that the intended recipient has responded. Since the encryption of the EDU ID is carried out automatically by EDU B 28, and can not be modified or affected by external signals, it is virtually impossible for a third party to use another EDU to break into a communications link and take part in establishing secure communications, since the encrypted EDU ID that is returned to the station that initiated the communication would then not match the expected EDU ID. A negative response to decision block 155 causes the process for establishing secure communications to be halted at a stop block 157. Otherwise, the process for establishing a secure communications link proceeds to a block 156. Block 156 provides for transmitting the encrypted key header and encrypted DEK2 to the other EDU, i.e., to EDU B 28, which is the intended recipient for subsequent encrypted communications. Thereafter, the logic proceeds to a block A2 158 in FIGURE 5.

FIGURE 5 illustrates a flow chart 160 defining the steps next implemented by EDU B 28. Following block 158, a block 162 provides for parsing the encrypted key header and encrypted DEK2. The encrypted key header is then decrypted in a block 164 using the ECB method in connection with the predefined header encryption

key, enabling EDU B 28 to determine the KEK table CRC, the EDU ID of the transmitting station, and the KEK2 table entry value. In a decision block 166, EDU B 28 determines if the KEK table CRC value is correct, i.e., confirms that the public or private table of KEKs used by EDU A 20 is the same as that being used by
5 EDU B 28. If not, the communication process is halted at a block 168. Otherwise, the process continues with a block 170.

Block 170 provides for selecting a 64-bit KEK from the designated table of KEKs using the entry value KEK2 as an offset. In a block 172, the EDU uses the selected KEK value in connection with the ECB method to decrypt the encrypted
10 DEK2. It then performs a validity check in a block 174, by using the DEK2 value in connection with the ECB method to encrypt the 64-bit zero function, thereby determining a check value and a table entry value KEK 2' that is based upon the 16 LSBs of the check value. A decision block 176 causes CPU 36 to determine if the decrypted KEK2 equals KEK2' that was just determined in block 174. If not, a
15 block 178 provides for indicating that an error has occurred, leading to a stop block 180.

However, assuming that the validity check has a positive response, in a block 182, the EDU logically XORs DEK1 and DEK2 to determine the value of DEK for this session. At this point, both the receiving and transmitting station EDUs have
20 established the current session data encryption key DEK. Before the communication session can proceed, one final check is made in a decision block 183.

Decision block 183 determines if the EDU ID sent by EDU A 20 in the key header that was decrypted in block 164 by EDU B 28 matches an expected EDU ID. If not, block 180 stops the process of establishing secure communications between the
25 two EDUs. Decision block 183 thus determines if a third EDU has been used to intercept communications between EDU A 20 and EDU B 28; if not, the communication of encrypted data proceeds at a block 184.

The session DEK is used in a block 184 by EDU A 20 to encrypt data (such as facsimile or computer data) for transmission to EDU B 28, which then decrypts it
30 using the same DEK. When EDU B 28 determines that the last of the data to be transmitted has been received and decrypted, a block 186 provides for resetting both EDUs to await the next communication. Thereafter, a stop block 188 terminates further communication between the two stations.

During the process of establishing secure communications, neither of the
35 EDUs linking together transmits DEK1 or DEK2 in the clear. Either the public or private table of KEKs is used for encrypting the first and second portions of the

current session DEK. Consequently, only another EDU provided with the same control program and the same table of KEKs (producing the same CRC value) would be able to decrypt either the encrypted first or second portions of the DEK. Since the software program controlling the operation of the EDUs requires that the EDU ID number of the stations be encrypted as part of the key header information that is exchanged, a third EDU cannot be used to surreptitiously substitute for the intended receiving station or transmitting station during the establishment of the secure communication link. Consequently, only the two EDUs at the receiving and transmitting stations comprising a link are able to communicate to establish a session DEK and thereafter carry on secure communications.

Only an EDU having the same session DEK used to encrypt data can decrypt the data. Furthermore, although any EDU can establish secure communications with any other EDU using the public table of KEKs, only EDUs having the same private table of KEKs (determined from the KEK table CRC value) can establish a session DEK to communicate with each other. As a result, a corporation that generates its own table of private KEKs can ensure that secure communications are initiated only with other stations comprising its private network that include the same table of private KEKs.

While the DES algorithm is used in the preferred form of the present invention, it will be appreciated that other encryption algorithms that use an encryption key can also be employed. Further, when determining a check value, a predefined function other than the zero function can be used. It should also be apparent that the encrypted key header need not include the EDU ID, if a lower level of security is acceptable, for example, in a local network of EDUs exclusively using private KEKs. These and other modifications to the present invention will be apparent to those of ordinary skill in the art. Accordingly, it is not intended that the invention be in any way limited by the description of the preferred embodiment and modifications thereto, but instead that the scope of the invention be determined entirely by reference to the claims that follow.

The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

1. Encryption/decryption apparatus for ensuring secure communications between two stations, said encryption/decryption apparatus disposed at each station comprising:

(a) encryption processor means for encrypting and decrypting data using an encryption key that is input thereto;

(b) control means, coupled to the encryption processor means, for controlling the operation of the encryption processor means, said control means supplying the encryption processor means with data for encryption and decryption and with the encryption key for use in encrypting and decrypting the data to produce an output signal in response to programmed instructions that cause it to automatically randomly select a part of a session data encryption key for use by the encryption processor means to encrypt data when combined with another part of the session data encryption key received from the other station; and

(c) non-volatile memory means, coupled to the control means, for storing a plurality of key encryption keys used by the encryption processor means in encrypting the part of the session data encryption key for transmission to the other station, said control means selecting the key encryption key from said plurality of key encryption keys as a function of a check value determined by the control means with the part of the session key.

2. The encryption/decryption apparatus of Claim 1, wherein said non-volatile memory means include an internal power source that supplies electrical power to maintain storage of the plurality of key encryption keys.

3. The encryption/decryption apparatus of Claim 1, further comprising potting means for encapsulating the encryption processor means, the control means, and the non-volatile memory means in a radio and light wave opaque material, said potting means being sufficiently hard and resistant to dissolution by solvents to prevent its removal without causing damage to interconnections coupling the non-volatile memory means to the control means and damage to interconnections supplying electrical power to the non-volatile memory means from the internal power source, such damage causing erasure of the plurality of key encryption keys stored in

the non-volatile memory means, said control means also responding to any attempt to externally interrogate the non-volatile memory means by causing erasure of the key encryption keys stored therein.

4. The encryption/decryption apparatus of Claim 1, further comprising multiplexer means, coupled to the control means to receive a select signal therefrom, and coupled to the encryption processor means, an output port, and the memory means, for selectively conveying a data signal thereto in response to the select signal.

5. The encryption/decryption apparatus of Claim 1, wherein the control means include a non-volatile memory for storing the programmed instructions and for storing a unique identification code that identifies a specific encryption/decryption apparatus.

6. The encryption/decryption apparatus of Claim 5, wherein the control means include means for locking the control means and its non-volatile memory to prevent data and program steps from being read externally or changed after storage of the programmed instructions in said non-volatile memory is complete.

7. The encryption/decryption apparatus of Claim 6, wherein the means for locking include means for encrypting data and memory addresses defining memory storage locations within the non-volatile memory of the control means and within the non-volatile memory means.

8. Encryption/decryption apparatus for ensuring secure communications, comprising:

(a) processor means for randomly selecting a partial session data encryption key;

(b) encryption means for encrypting the partial session data encryption key, producing an encrypted part key and decrypting another partial session data encryption key selected at another location; and

(c) means for conveying the encrypted part key to an output port so that it can be transmitted to the other location and for conveying an encrypted signal from an input port, said encryption means decrypting the other partial session data encryption key received from the other location as the encrypted signal, said

processor means combining the partial session data encryption key to determine the current session data encryption key that is subsequently used by it to encrypt data transmitted to the other location and to decrypt encrypted signals received from the other location.

9. The encryption/decryption apparatus of Claim 8, further comprising memory means for storing a plurality of key encryption keys, wherein the encryption means select a specific key encryption key from the plurality of key encryption keys as a function of a check value, said encryption means encrypting a predefined set of characters with said part of the encryption key to determine the check value.

10. The encryption/decryption apparatus of Claim 9, wherein the means for transmitting also transmit the check value determined by the encryption means.

11. The encryption/decryption apparatus of Claim 10, wherein the decryption means use a check value received from said other location to determine a specific key encryption key that was used to encrypt the other partial session data encryption key.

12. The encryption/decryption apparatus of Claim 11, wherein:

(a) the encryption means use the other partial session data encryption key decrypted by the decryption means to encrypt the predefined set of characters, producing a test value;

(b) said processor means compare the test value with a check value received from the other location and detect an error if the test value differs from said check value received from the other location; and

(c) if an error is detected in (b), said processor means halt communications with said other location.

13. The encryption/decryption apparatus of Claim 9, wherein said memory means store a unique identification code for that apparatus.

14. The encryption/decryption apparatus of Claim 9, wherein the memory means comprise a non-volatile memory circuit including an internal power source, said internal power source supplying electrical current to the non-volatile memory circuit to retain data stored therein, said memory means being encapsulated in a material that precludes physical inspection of the memory circuit, preventing discovery of the data stored therein, further comprising means for interrupting electrical current supplied from the internal power source to the memory circuit so that the data stored therein are erased if the material encapsulating the memory means is removed therefrom.

15. The encryption/decryption apparatus of Claim 8, wherein the processor means comprise a central processing unit that is programmed to control the encryption means and the decryption means according to a predefined set of instructions.

16. The encryption/decryption apparatus of Claim 8, wherein the encryption means comprise an integrated circuit that implements encryption and decryption of data from a plurality of sources in response to signals from the processor means, using the current session data encryption key, in accordance with a predefined encryption algorithm and a corresponding predefined decryption algorithm.

17. The encryption/decryption apparatus of Claim 9, wherein the memory means store a plurality of sets of key exchange keys, further comprising means for selecting one of the sets of key exchange keys from which the specific key exchange key is determined.

18. Encryption/decryption apparatus for ensuring secure communications, comprising:

(a) a sealed circuit encapsulated in a material opaque to radio and light waves, said sealed circuit comprising:

(i) a central processing unit that receives and transmits data in both an encrypted and decrypted form;

(ii) a memory circuit coupled to the central processing unit, at least one predefined set of key exchange keys being stored in the memory circuit,

said key exchange keys stored in the memory circuit being externally inaccessible, both physically by inspection and by downloading through the central processing unit;

(iii) an encryption/decryption coprocessor coupled to the central processing unit to receive data therefrom, said encryption/decryption coprocessor encrypting and decrypting the data under control of the central processing unit based upon a specified encryption key, the encryption/decryption coprocessor selectively generating a second set of key exchange keys that are also stored in the memory circuit;

(b) connector means for interconnecting the sealed circuit with external data input and output lines, the encryption/decryption coprocessor selectively encrypting the second set of key exchange keys and the connector means conveying the second set of key exchange keys in an encrypted form to an external device for distribution to other encryption/decryption apparatus comprising a limited network, whereby only encryption/decryption apparatus comprising the limited network can securely communicate with each other using the second set of key exchange keys, but can securely communicate with other like encryption/decryption apparatus that do not comprise the limited network using the predefined set of key exchange keys.

19. The encryption/decryption apparatus of Claim 18, further comprising memory means coupled to the central processing unit, for storing program steps controlling automatic determination of a session data encryption key for use in encrypting and decrypting data, said session data encryption key being determined in part by the central processing unit logically combining a first randomly selected portion of the session data encryption key that is received in an encrypted form from another location with a second randomly selected portion of the session data encryption key that the central processing unit transmits to the other location in an encrypted form.

20. The encryption/decryption apparatus of Claim 19, wherein one of the predefined set and the second set of key exchange keys is selectively used for encrypting said other portion of the session data encryption key.

21. The encryption/decryption apparatus of Claim 18, wherein the memory circuit stores a unique identification code for the sealed circuit that can not be changed, said central processing unit halting operation of the sealed circuit if data are received from the other location that specify a different identification code, thereby

preventing secure communications with an unintended encryption/decryption apparatus.

22. Encryption/decryption apparatus for ensuring secure communications between two stations, comprising:

(a) first processor means at one of the stations for randomly selecting a first part encryption key and second processor means at the other of the two stations for randomly selecting a second part encryption key;

(b) encryption means at said one station for encrypting the first part encryption key, producing an encrypted first part key;

(c) means for transmitting the encrypted first part key to said other station;

(d) decryption means at said other station for decrypting the encrypted first part key to determine the first part encryption key;

(e) encryption means at said other station for encrypting the second part encryption key, producing an encrypted second part key;

(f) means for transmitting the encrypted second part key to said one station; and

(g) decryption means at said one station for decrypting the encrypted second part key to determine the second part encryption key, said first processor means at said one station and said second processor means at said other station then combining the first part encryption key and the second part encryption key to determine an encryption key that is used to encrypt and decrypt subsequent communications between the two stations.

23. The encryption/decryption apparatus of Claim 21, further comprising memory means for storing a plurality of key encryption keys at each of the two stations, wherein the encryption means at each station select a specific key encryption key from the plurality of key encryption keys as a function of a first check value and a second check value, respectively, said encryption means at said one station encrypting a predefined set of characters with said first part encryption key to determine the first check value, and said encryption means at said other station encrypting the predefined set of characters with said second part encryption key to determine said second check value.

24. The encryption/decryption apparatus of Claim 22, wherein the means for transmitting from each station also transmit the respective first or second check value determined by the encryption means at each station.

25. The encryption/decryption apparatus of Claim 23, wherein the decryption means at said other station use the first check value received from said one station to determine the specific key encryption key that was used by the encryption means at said one station to encrypt the first part encryption key, and wherein the decryption means at said one station use the second check value received from said other station to determine the specific key encryption key that was used by the encryption means at said other station to encrypt the second part encryption key.

26. The encryption/decryption apparatus of Claim 24, wherein:

(a) the encryption means at said other station uses the first encryption key decrypted by the decryption means to encrypt the predefined set of characters, producing a test check value;

(b) said second processor means compares the test check value with the first check value and detects an error if the test check value differs from the first check value;

(c) the encryption means at said one station uses the second encryption key decrypted by the decryption means to encrypt the predefined set of characters, producing a test check value;

(d) said first processor means at said one station compares the test check value with the second check value and detects an error if the test check value differs from the second check value; and

(e) if an error is detected in (b), said second processor means halt communications with said one station, and if an error is detected in (d), said first processor means halt communications with said other station.

27. The encryption/decryption apparatus of Claim 22, wherein said memory means at each station store a unique identification code for that station.

28. The encryption/decryption apparatus of Claim 26, wherein the encryption means at said one station encrypt the unique identification code of said other station, the means for transmitting then transmitting an encrypted identification code to said other station, said decryption means at said other station decrypting the unique identification code.

29. The encryption/decryption apparatus of Claim 27, wherein said second processor means compare the decrypted unique identification code with the unique identification code stored in the memory means and if not identical, halt communications with said one station.

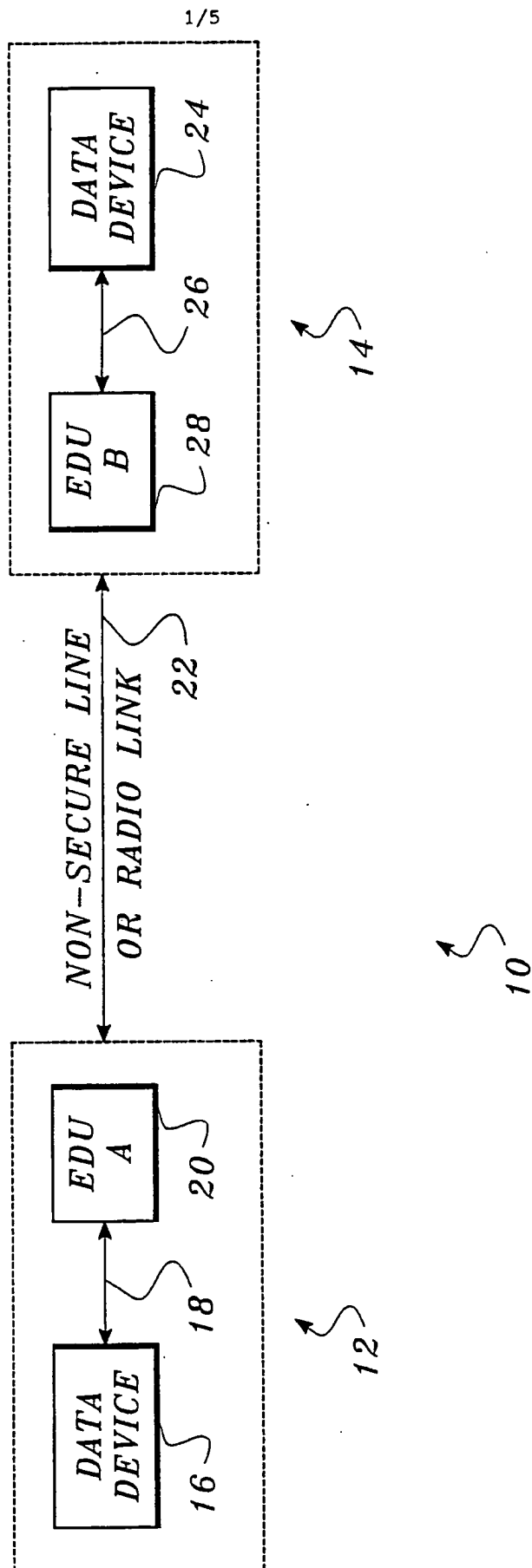


FIG. 1.

2/5

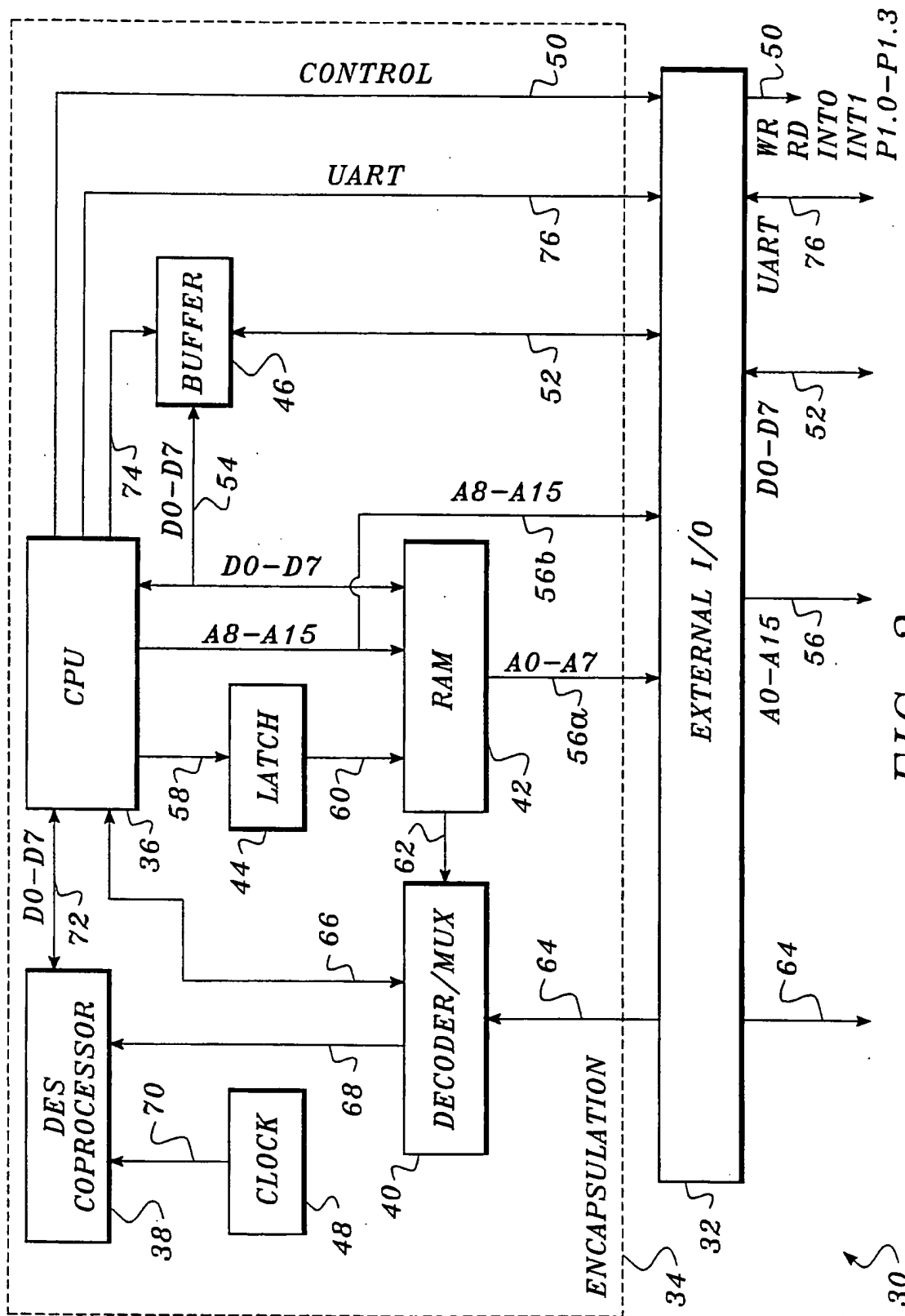
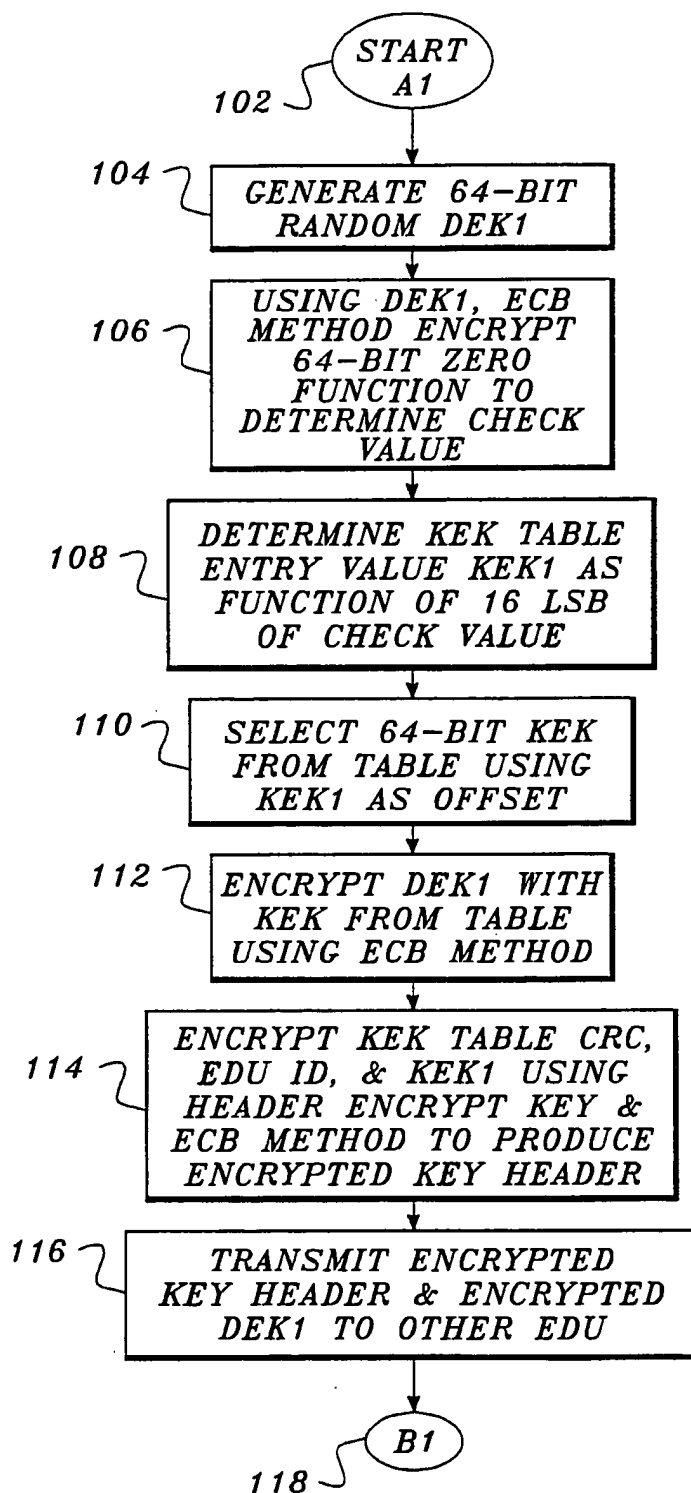


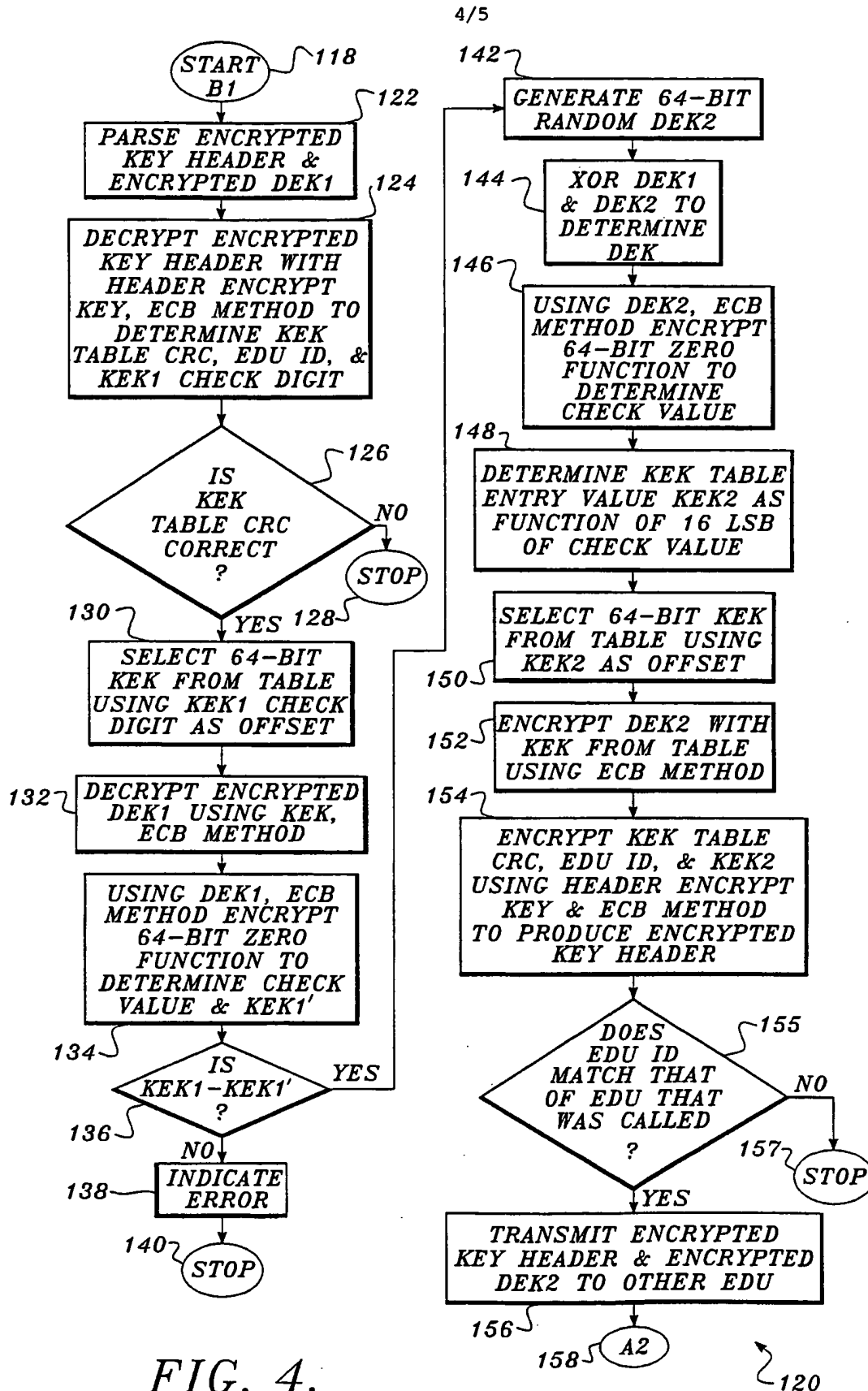
FIG. 2.

3/5



100

FIG. 3.



5/5

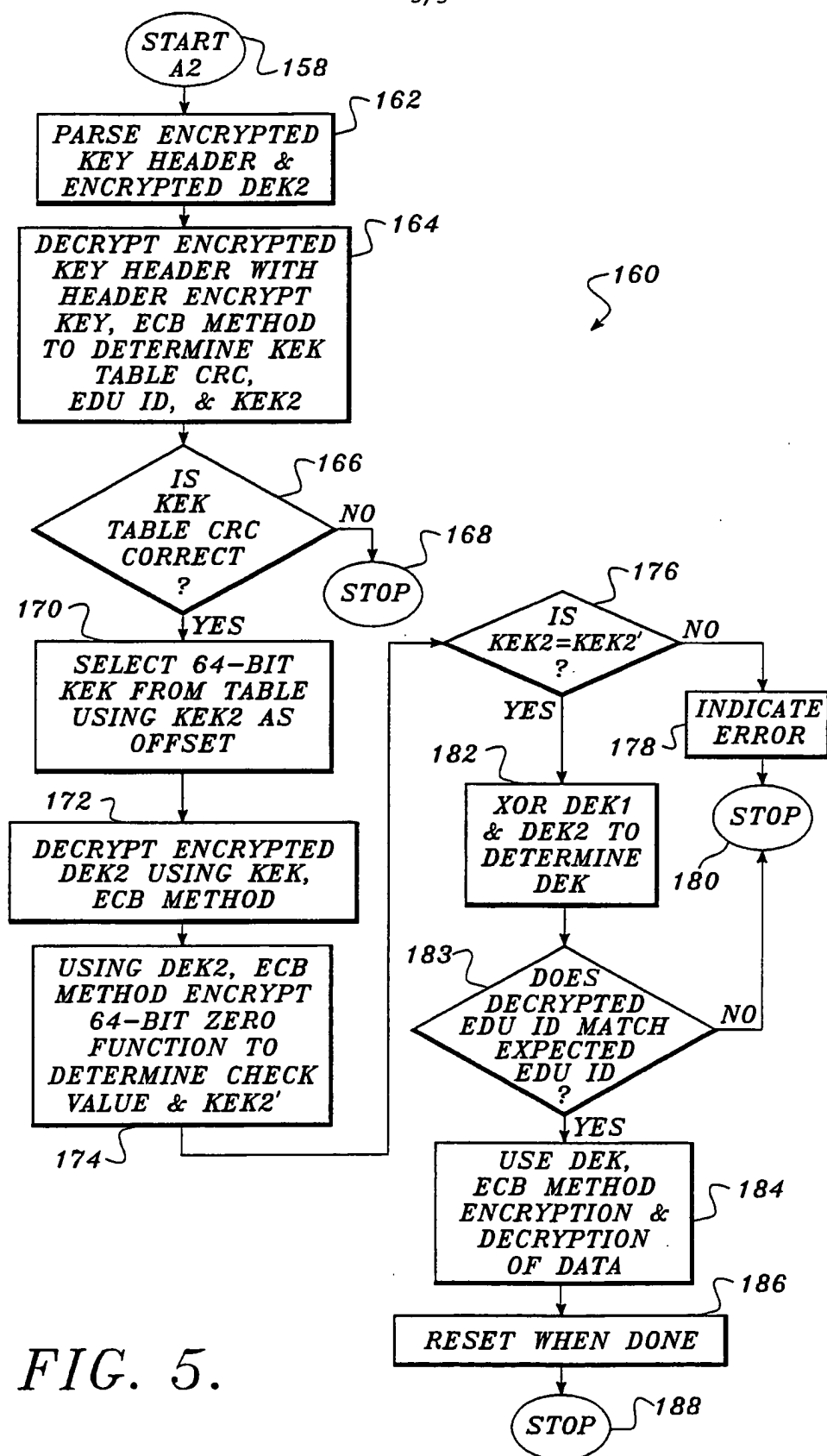


FIG. 5.

INTERNATIONAL SEARCH REPORT

 International application No.
 PCT/US93/04340
A. CLASSIFICATION OF SUBJECT MATTER
 IPC(5) :HO4L 9/00
 US CL :380/21, 49, 28

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. : 380/21, 49, 28

 Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
 380/52, 46

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, A 5,029,208 (Tanaka) 02 July 1991	1-29
A	US, A 5,124,117 (Tatebayashi, et al.) 23 June 1992	1-29
A, P	US, A 5,144,665 (Takaragi, et al.) 01 September 1992	1-29
A	US, A 4,607,137 (Jansen, et al.) 19 August 1986	1-29
A	US, A RE33189 (Lee, et al.) 27 March 1990	1-29
A	US, A 4,578,531 (Everhart, et al.) 25 March 1986	1-29
A	US, A 4,876,716 (Okamoto) 24 October 1989	1-29

☐ 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
 10 August 1993

 Date of mailing of the international search report
 26 AUG 1993

 Name and mailing address of the ISA/US
 Commissioner of Patents and Trademarks
 Box PCT
 Washington, D.C. 20231

Authorized officer

David Cain

Facsimile No. NOT APPLICABLE

Telephone No. 308-0463

Form PCT/ISA/210 (second sheet)(July 1992)*

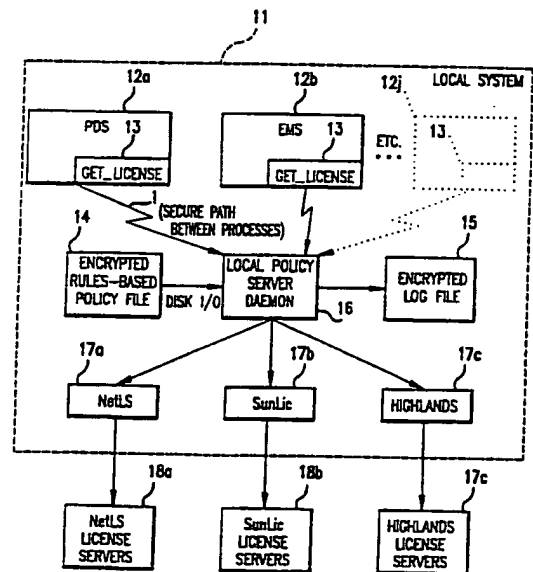


INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : G06F 1/00, 11/34	A1	(11) International Publication Number: WO 93/11480 (43) International Publication Date: 10 June 1993 (10.06.93)
(21) International Application Number: PCT/US92/10215 (22) International Filing Date: 24 November 1992 (24.11.92) (30) Priority data: 07/798,934 27 November 1991 (27.11.91) US (71) Applicant: INTERGRAPH CORPORATION [US/US]; One Madison Industrial Park, Huntsville, AL 35894 (US). (72) Inventors: BAINS, Jeffrey, E. ; 134 Michli Road, Madison, AL 35758 (US). CASE, Willard, W. ; 104 Arden Avenue, Madison, AL 35758 (US). (74) Agents: SUNSTEIN, Bruce, D. et al.; Bromberg & Sunstein, 10 West Street, Boston, MA 02111 (US).		(81) Designated States: CA, JP, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>With international search report.</i>

(54) Title: SYSTEM AND METHOD FOR NETWORK LICENSE ADMINISTRATION**(57) Abstract**

Disclosed is a system for administration, on a computer network, of license terms (a so-called license server) for a software product (12a, 12b... 12j) provided to said network. Said license server (17c, 18a, 18b) being realized by one of the network computers and which tasks comprise e.g. tracking of a software product (12a, 12b... 12j) usage in the system, issuing usage permits (licenses) to the different network users in accordance with predefined conditions, monitoring expirations and violations (e.g. the maximum number of users simultaneously using a software product) of issued licenses and when necessary, withdrawing issued software product licenses. In one embodiment, the system includes a policy server database (14) maintained on each node (11) of the system, where said predefined conditions are specified under which usage of a software product (12a, 12b... 12j) is permitted at the respective system nodes (11). Each node also has a policy server "daemon" (16) in association with said policy server database (14) for interaction with the license server (17c, 18a, 18b) in order to enforce license terms for a software product.



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	FR	France	MR	Mauritania
AU	Australia	GA	Gabon	MW	Malawi
BB	Barbados	GB	United Kingdom	NL	Netherlands
BE	Belgium	GN	Guinea	NO	Norway
BF	Burkina Faso	GR	Greece	NZ	New Zealand
BG	Bulgaria	HU	Hungary	PL	Poland
BJ	Benin	IE	Ireland	PT	Portugal
BR	Brazil	IT	Italy	RO	Romania
CA	Canada	JP	Japan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo	KR	Republic of Korea	SE	Sweden
CH	Switzerland	KZ	Kazakhstan	SK	Slovak Republic
CI	Côte d'Ivoire	LJ	Liechtenstein	SN	Senegal
CM	Cameroon	LK	Sri Lanka	SU	Soviet Union
CS	Czechoslovakia	LU	Luxembourg	TD	Chad
CZ	Czech Republic	MC	Monaco	TG	Togo
DE	Germany	MG	Madagascar	UA	Ukraine
DK	Denmark	ML	Mali	US	United States of America
ES	Spain	MN	Mongolia	VN	Viet Nam
FI	Finland				

-4-

SYSTEM AND METHOD FOR NETWORK LICENSE ADMINISTRATION

5 **Technical Field**

The present invention relates to systems and methods for controlling and monitoring the use, on computers and computer networks, of computer programs.

Background Art

10 As computers have proliferated in availability, the investment in computer software has also grown, and there have been developed various methods for charging the computer user for use of computer software products. Typically computer software products are licensed, rather
15 than sold, to the computer user under various arrangements. The simplest common license arrangement gives the user the right to use a software product on a single computer, i.e., to employ only one central processing unit (CPU) in connection with operation of the software product. Although
20 many such licenses are for indefinite periods of time, a license may also be for a limited duration and extendable, so that the entity marketing the product can charge a periodic fee (for example, annually) for use of the software product. Or use may be absolutely time-limited (for
25 example, one-day), so that the user may evaluate the software product for possible purchase of a regular license.

Since software can be copied and moved easily from one like machine to another, companies have invented methods to prevent unauthorized use of their software products. Some
30 licensors require passwords to activate software on a particular machine. The password may be keyed to the hardware's identification number as a condition for operation of the software. Such systems can effectively lock software to a particular machine, but do not address

35

- 2 -

software that is licensed for concurrent or simultaneous use. Some licensors use hardware locks that attach to a parallel printer port or a serial port on a machine; each time the software is activated, it looks for a specified code, in the hardware lock, as a condition for operation of the software. Using hardware locks resolves the problem of unauthorized moving of software among machines; however, hardware locks do not handle multiple software products on a single machine, and they require time and expense to deliver to the end user.

When computer software products are used in a network environment (which may include computers running in various roles as workstations and servers of various types linked together over a data path), additional licensing challenges are present. For example, a network may permit a user at one node (which may be a terminal or workstation, for instance) to utilize a software product running at another node (which may be the network server or even another workstation). Consequently, the terms of the single-computer type of software license might not cover the usage of the software product on the network, or worse still (from the point of view of the licensor) might actually permit such a usage without additional compensation to the licensor. One approach to network licensing is to grant permission to use the program based on all of the nodes on the network, and to require a license for each node. Then typically the license fee may be increased as the number of nodes on the network increases. Another approach bases the license fee for a software product running on a network on the total number of individual users who might actually run the software, regardless of the number of nodes either on the network or running the software product at a given time. These approaches, however, have usually required the cooperation of the licensee, because additional nodes may be added to the network, or additional users may utilize the software, without the knowledge of the licensor, who is typically not present on the premises of the licensee. The licensor may

- 3 -

reserve the right to audit the licensee's site, but such an audit is intrusive, expensive, and may alienate potential or actual customers for licenses. Although other approaches exist under which one might charge a single fee per server
5 or per site or per entity, often on an individually negotiated basis, these approaches are often impractical or inflexible, in that they also typically do not take into account the possible wide variation over time in the number of nodes or users and also require reliance on licensee
10 cooperation.

The same circumstances that make license enforcement difficult for the licensors of software products for a network environment also make license compliance difficult for the conscientious administrator, for example, of a
15 Management Information System (MIS) or Computer Aided Design (CAD) department of a company using software products. The administrator may be called upon to ensure that the number of workstations using a variety of software products in a network environment complies with the terms of a variety of
20 license agreements. Such an administrator may have to develop and promulgate a series of directives about the terms of permitted workstation usage and must depend primarily upon the goodwill and voluntary compliance of unit personnel with such directives.

25 Recently it has become practical in some network environments to determine and limit the number of nodes that may access a software product at a given time, and to charge a license fee based on the maximum number of nodes that are permitted to use the software product concurrently. This is
30 called "concurrent licensing". In these environments, a computer program, acting as "librarian" and running on a computer node designated as a license server, is typically used to distribute license keys (sometimes called "tokens") over the network to nodes requesting access to run a
35 software product; the number of keys is tracked by the librarian; and if at a given time, the permitted maximum number of keys would be exceeded by usage of the software

- 4 -

product on a requesting node, the node can be denied, at such time, access to invoke the software product.

Examples of software-based concurrent licensing arrangements may be found in Unix applications running in connection with software products sold under the trademarks NetLS (available from Gradient Technologies, Inc., 577 Main Street, Suite 4, Hudson, Massachusetts 01749), and SunLic (available from Sun Microsystems, Inc., Mountain View, California), and Flexible License Manager (available from Highland Software, Inc., 1001 Elwell Court, Palo Alto, California 94303). However these arrangements suffer from a number of disadvantages. NetLS, for example, includes mechanisms for tracking which nodes have been given keys to run a given software product and the number of keys available for running such software product. However, it is up to the designers of each software product to program such product to implement the terms of any license agreement, and, in particular, to program into the product calls to the NetLS software to provide information to the computer running the software product and to write code in the applicable product to prevent use of the product when the license terms have not been obeyed. Thus a computer system utilizing ten different software products that rely on NetLS for license enforcement will generally have ten different substantial software portions (one in each computer product) to achieve license enforcement. In addition to this complexity, if the license server running NetLS fails, or if the network itself fails, then a workstation loaded with the software product cannot run the software product, since the product requires NetLS interaction to be activated.

The foregoing difficulties are applicable generally not just to NetLS but to "metering software" generally. The Microcomputer Managers Association has issued a White Paper (October 2, 1991), reprinted in Infoworld, pages 46-42 (October 14, 1991) on the problems of network licensing, Commenting on the problem that each software product requires its own interface to the metering software (as well

- 5 -

as possible input of administrative information), the White Paper suggests that "[i]t makes much more sense to have a single package provide the metering for all application software on the network." Infoworld (October 14, 1991),
5 supra, at page 51, column 4. Such an approach has its own difficulty, however. Each application would still have to interface with the single metering package, and the interface to such a package must somehow deal with the varying licensing terms of each software product. Moreover,
10 with the metering package running on the license server, a failure of the server or the network would prevent all software applications from running anywhere on the network.

Summary of the Invention

In a preferred embodiment, the present invention
15 provides an improved system for administration, on a computer network, of license terms for a software product on the network. The improved system is of the type having an arrangement, such as NetLS, for tracking software product usage, associated with one of the computers acting as a
20 license server. This arrangement permits the license server (i) to identify the current set of nodes that are using the software product at a given time, (ii) to handle license data concerning conditions under which usage of the software product is permitted at any given node, and (iii) to
25 determine whether at any given time the conditions would still be satisfied if a given node is then added to this set of nodes. The software product may thus include instructions to interface with the license server to cause enforcement of the license terms. The improvement, in one
30 embodiment, to the system includes a policy server database maintained on each node, containing data specifying conditions under which usage of the software product is permitted on such node. Each node also has a policy server "daemon" (which may be implemented in software) in
35 association with the corresponding policy server database, for (i) communicating with the license server, (ii) interfacing with both the software product and the

- 6 -

corresponding policy server database, (iii) making a permission-to-run availability determination, with respect to local usage of the software product, on the basis of applicable data from the license server and the

5 corresponding policy server database, so that enforcement of license terms applicable to the software product at a given local node is achieved on the basis of both license policy maintained at such local node as well as applicable data from the license server.

10 In a further embodiment, each policy server database contains data specifying conditions under which usage of each of a plurality of software products is permitted on the node on which the database is maintained. Additionally, each policy server daemon interfaces with each software
15 product. In this manner, enforcement of license terms applicable to each software product at a given node is achieved on the basis of both locally maintained license policy and applicable data from the license server.

In a further embodiment, each node has a log file
20 maintained, in association with each policy server daemon, to record recent software product usage on that node. The policy server daemon is accordingly configured to handle instances when data from the license server is unavailable -- for example, when the computer acting as the
25 license server is non-operational or when the network is non-operational. In particular, the policy server daemon may permit a node to run a software product, in the absence of license server data, if the node's log file indicates a sufficient level of recent usage of the software product on
30 the node. The circumstances under which such a permission-to-run availability determination is favorable may be established by the node's policy server database.

In yet further embodiments, the policy server database and the log file may be encrypted. Furthermore the
35 interface between the policy server daemon and each software product may be made secure. When one or more of the software products are subject to concurrent licensing

- 7 -

restrictions specified in the policy server databases, the policy server daemon may be permitted to reserve a predetermined time interval over which the applicable node has a guaranteed opportunity to utilize a given software product. The reservation is accomplished by having the node's policy server daemon communicate to the license server over the predetermined time interval that the node is using the given software product, regardless whether the software product is actually being used.

It can be seen that the present invention permits a single database at each node to specify all of the conditions under which the node may access any of the software products on the network. Furthermore, as described in further detail below, in order to invoke the licensing administration function carried out in accordance with the present invention, each software product need contain only a simple and short segment including the instruction:

ILic-get_license

followed by parameters identifying license details for the particular software product. A branching routine (which may be made available to all the software products, and called by the particular software product after this instruction) then specifies program flow depending on whether a license is available (the remainder of the program can be run) or not (the program operation is terminated and a message is displayed to the user).

Brief Description of the Drawings

The foregoing features of the invention will be more readily understood by reference to the following description taken with the accompanying drawings in which:

Fig. 1 is a block diagram showing operation of a preferred embodiment of the invention in a network;

Fig. 2 is a block diagram illustrating the interrelation of important modules of the embodiment;

Fig. 3 is a block diagram of the main logical flow of

- 8 -

the computer program used in the embodiment;

Fig. 4 is a block diagram of the main processing of a validated "get license" message;

Fig. 5 is a block diagram of the manner in which the policy server database is structured;

Fig. 6 is a block diagram providing more detail than Fig. 4 of the logical flow of the processing of a "get license" message;

Fig. 7 is a block diagram of license acquisition processing referred to as item 623 in Fig. 6;

Fig. 8 is a block diagram of clock message processing for licenses on the main list of licenses that have been established; and

Fig. 9 is a block diagram of clock message processing for licenses moved to the recovery list by item 88 of Fig. 8.

Detailed Description of Specific Embodiments

The invention is applicable to computer networks of the type having an arrangement, such as NetLS, for tracking software product usage, associated with one of the computers on the network acting as a license server. The present embodiment is described with respect to a Unix network; however, the software used by the license server in implementing such an arrangement, and the particular network type, are a matter of design choice.

Fig. 1 shows the manner in which a preferred embodiment of the invention may be implemented on a Unix network. Each computer node 11 of the network may be running a variety of software products, such as PDS (item 12a), EMS (item 12b), and so forth (shown through item 12j). Each of these products includes a call "get_license" to the local policy server daemon 16 for a determination whether a license is available to run the product in question. As used in this detailed description, the term "license" refers not to a written document between the licensor and the licensee, but rather to the availability of permission to run the software product. The local policy server daemon 16 operates at the

- 9 -

computer node 11 and makes the permission-to-run availability determination by reference to its associated policy server database 14, also located at the node, to identify the rules specifying the circumstances under which a license would be granted. The daemon 16 also communicates over the network with the applicable license server. In this figure, three separate license servers are shown: one (item 18a) running NetLS; another (item 18b), SunLic; and another (item 18c), Highlands. The license server communicates with the daemon 16 using the applicable license software NetLS (item 17a), SunLic (item 17b), or Highlands (item 17c), and informs the daemon whether usage of the software product on the network is such that a license may be granted in accordance with the policy established by the database 14. If so, the daemon 16 reports the license to the applicable software product 12, and to the applicable license server 18. If there is no successful communication with the applicable license server 18, if the database 14 so permits, the daemon 16 will consult a log file 15 recording instances of recent software product usage, and if there has been a sufficient level of recent software product usage that has been licensed, the daemon will grant a temporary user license (TUL) to run the software product.

The communication between the applicable software product 12 and the local policy server daemon 16 is handled as an interprocess communication in Unix. Here the Unix "message" is used as the means of communication, but this is a matter of choice, and other means of communication, such as pipes or shared memories, may be used. In order to reduce to risk of tampering by the licensee with the license availability determination made by the policy server daemon 16, the rules database 14 and the log file 15 may be encrypted using techniques known in the art. Similarly, the message communication from the application to the policy server daemon 16 can be subject to validation using techniques known in the art to assure that the message is indeed from the pertinent software product.

- 10 -

The embodiment described herein has been implemented for use with a variety of types of licenses. (Numerous license types may be created and enforced by the invention, but the following types are illustrative.) One type is the concurrent use license. A concurrent use license is issued, from the server running the Licensing System (sometimes called the "license server" in this description and the claims following), with respect to a software product being used on a node in a network. The license server controls the levels of concurrent usage of the software product. Concurrent use licenses are returned to the license server when they are no longer needed. For example, if the Licensing System on the server permits five concurrent licenses for a given software product, then five users on the network can run the software product concurrently.

The concurrent use license is actually implemented as part of a two-tier structure. The first tier is a "base license," and the second tier is a "version-specific" license. The base license controls the number of simultaneous users of a software product. The version license controls the version of the software product that may be utilized by the user. The base license typically expires at the end of each year, and may be renewed. The version license typically never expires. The version license provides a mechanism for controlling how many base licenses are for a software product that is under a maintenance agreement. As an example, a user may have purchased a license to five copies of version A of a software product, but kept maintenance on only three copies. In such a case the user would receive five base licenses (which expire each year and were replenished unless the applicable computers were sold), plus five version A licenses that never expire. This user would subsequently receive only three version B licenses for the three copies under maintenance. Under such an arrangement, the user could still run five copies of version A of the software product, or a mix of version A and version B software as

- 11 -

long as the mix does not exceed five copies in total and does not exceed three copies of version B.

The policy server database file 14 of a computer node stores the license requirements for each software product to be run at that node. For each software product, the database may identify the number of base license "tokens" and version license "tokens", obtained from the server running the Licensing System, that are necessary for operation of that software product on the particular computer constituting the node. (The particular computer, for example, may be particularly fast in processing, and therefore a higher license fee may be required for running the software product on such computer, resulting here in a larger number of tokens required for the base and version licenses.)

Another type of license is a node-locked license, which is tied to a particular computer node and cannot be used by other nodes. The node-locked license token is designated for a particular node when created. In a further variation of the node-locked license, a "reserved" license may be established, that is, the policy server daemon may be permitted to reserve a predetermined time interval over which the applicable node has a guaranteed opportunity to utilize a given software product. (The reservation is accomplished by having the node's policy server daemon communicate to the license server over the predetermined time interval that the node is using the given software product, regardless whether the software product is actually being used.)

A single use license can be used only for one invocation of the software product. Single use licenses are useful for emergency situations, peaks in usage, or demonstrations. A day use license is similar to a single use license, except that a day use license remains available on the computer node that acquired it for 24 hours after the time of acquisition.

A temporary user license (TUL), described above, is

- 12 -

issued on a temporary basis when the server running the Licensing System becomes unavailable. A TUL is designed for emergency situations and is granted on a per user, per node, per software product, per usage history basis.

5 Fig. 2 illustrates the structure of a program implementing the embodiment of Fig. 1 for a single license server running one or more Licensing Systems, such as NetLS. The program is written in standard C. A communication module 21 handles communication with the various software
10 products, one of which is here shown as item 26. If the software product includes the "get_license" instruction, the communication module 21 refers to the licensing dispatch module 22. The licensing dispatch module 22, by reference to the policy server database 14 and the applicable
15 Licensing System, makes the license availability determination. The Licensing System shown here is No. 1, and client portion 25 is accessed by licensing dispatch 22, which may access other Licensing Systems depending on the software product 26 and information in the policy server
20 database file 14. The client portion of the Licensing System 25 communicates over the network with the server portion 251. In the event that there is no successful communication with the server portion 26, the communication module may trigger the temporary user license (TUL) module
25 27 to consult with the history log file 15 to determine if there is a sufficient level of recent licensed usage of the software product at this node to permit the grant of a temporary user license (TUL). In any event, the communication module 21 reports the license availability
30 determination by directing a message to the software product's process. The communication module is also responsible for sending a periodic signal (a "ping") to the license server to indicate continued use of a license. Another module 28 causes recordation of license usage in
35 license usage file 281 for reporting purposes. A file 252 of node-locked licenses is maintained locally. The communication module 21 is controlled by timer interval

- 13 -

handler 29, which in turn receives periodic signals from PS driver 291 that has been incorporated into the operating system.

Fig. 3 illustrates the main logical flow of the program carried out by the communication module 21. After initialization 31, the program gets the next message from any processes, and, in particular, from any software products that may be invoked from the node on which the program is running. Next the message is validated (item 33), and then the message is processed (item 34). After processing of the current message, the program loops to seek the next message again.

The most important message is "get_license", and this message is processed as shown in Fig. 4. The first step 41 is to determine the availability of a license. The license availability determination is made in the licensing dispatch module 22.

After the license availability determination is made as shown in step 41 of Fig. 4, if a license is granted, that fact is reported to the software product in step 43. If the license is not granted because of a lost connection to the server running the Licensing System (determination in step 44), there is a check to see if usage of the software product is possible "under grace", that is, whether there has been sufficient recent licensed usage of the software product at the node to permit granting of a TUL. If so, a TUL is granted (step 47). If not, or if the license was denied for reasons other than a lost connection, the program communicates (step 46) the fact of no license availability to the software product.

Additionally, the communication module of the policy server daemon may reserve a predetermined time interval over which the applicable node has a guaranteed opportunity to utilize a given software product. The reservation is accomplished by having the module communicate to the license server over the predetermined time interval that the node is using the given software product, regardless whether the

- 14 -

software product is actually being used.

The construction of the policy server database is shown in Fig. 5. License prices are initially established by management decision in price book 51, which forms the basis for assigning token values (step 52) required for license grant. The license cost to use a software product can also vary as a function of the hardware platform (i.e., the model of the computer) on which the product is running. Accordingly, the platform indicator data 54 and the rules defining the different types of licenses 53 all form a part of the structure of the policy server database 55. In order to assure integrity of the database, it is encrypted.

Fig. 6 is a block diagram providing more detail than Fig. 4 of the logical flow of the processing of a "get license" message. Initially (step 61), memory is allocated for the structure of the applicable license to be added to the list of license structures in memory. Unless the structure shows a reserved license (tested in block 62), the policy server database file 14 of Figs. 1 and 2 is accessed (step 621) to determine the applicable license terms. If access is successful (tested in block 622), then license acquisition processing (described in connection with Fig. 7) follows (step 623).

If, as a result of license acquisition processing, a license is granted (tested in block 625), the history log file 15 of Figs. 1 and 2 is then updated (step 631) to reflect this event. Thereafter, the policy server driver 291 of Fig. 2 is informed (step 63), the license usage file 281 of Fig. 2 is updated for use in generating later reports (step 64), the return status the operation is checked (step 641), and a status message is built and sent (step 65) to the software product that had included the "get license" call. If the return status is a failure (tested in step 641), the license structure is removed from memory (step 642) before sending the the status message to the software product.

If, as a result of the license acquisition processing

- 15 -

of step 623, a license has not been granted, the error messages produced by the Licensing System are analyzed to a single reason (step 626), and the return status for the software product is determined. If the embodiment described
5 herein is not in the enforcement mode (determined in step 627), then the return status is simply a warning (generated in step 643). If the embodiment is in the enforcement mode, there is a check (step 647) to determine if the connection with the license server is lost. If there is no lost
10 connection, the policy server database file 14 is checked (step 646) for the appropriate license failure conditions, and then the return status is determined (step 644). If there is a lost connection, processing follows (step 645), to determine on the basis of the history log file 15 and
15 data in the policy server database file 14 whether a TUL is available. If a TUL is available, the return status is a warning (step 643), as in the case when the system is not in the enforcement mode. Once the return status has been determined, processing is the same as if a license has been
20 granted; that is, the driver is informed, the license usage file is updated, the return status is checked and if necessary the license structure is removed from memory, and the appropriate status message is built and sent to the software product (steps 63, 64, 641, 642, and 65).

25 If after the determination (step 647) that there is a lost connection, and a TUL is not available (step 645), processing loops back to license acquisition processing (step 623) to attempt again to get a license from the license server. If the policy server database file 14 cannot be
30 successfully accessed in step 621 to determine the relevant license rules (a matter checked in step 622), the processing goes to determine (in step 627) whether the system is in the enforcement mode and to generate an appropriate return status. If in step 62, the license structure shows a
35 reserved license, access to the policy server database file 14 is skipped altogether, and the driver is informed (step 63) directly.

- 16 -

Fig. 7 is a block diagram of the license acquisition processing referred to as item 623 in Fig. 6. In accordance with this processing, there is first sought a "base" license token and then a "version" license token, where "base" and "version" have the meanings described above following the description of Fig. 1. Initially, the policy server database file 14 is cycled through to determine the enabled base token type (step 71)--for example node-locked, or concurrent access, or use once. The Licensing System on the server is then called (step 711) to seek the designated enabled token. If the base token is granted (checked at step 712), the policy server database file 14 is then cycled through to determine the enabled version token type (step 72). If the version token is granted (checked at step 722), the return is "license granted" (step 73). In each case if processing through the policy server database is not complete (checked for, in the case of the base token at step 713 and in the case of the version token at step 723), the database is cycled through again, the Licensing System is called to seek an enabled token, and there is a test to see if the token is granted. If the end of the list has been reached (tested at step 713 for the base token and 723 for the version token) and the applicable token has not been obtained, a failure is returned (step 725). If the base token has been granted, but the version token denied, then the base token is first freed (step 724) before the failure is returned in step 725.

Fig. 8 is a block diagram of clock message processing for licenses on the main list of licenses that have been established. First, a license is picked as part of a cycle through the main list of licenses in memory (step 81). Next there is a check whether a process exists for this license (step 82). If there is no process, the license is returned to the Licensing System, and associated housekeeping is done (step 821), and the program then picks the next license (step 81) to begin processing again. If it is determined that there is a process, then it is determined whether the

- 17 -

license needs to be "pinged" to satisfy requirements of the Licensing System to keep the license (step 83). The implementation here generates a ping every 10 minutes. If no ping is currently necessary, the program again picks the next license (step 81) to begin processing again. If a ping is necessary, it is sent (step 84), and if successful (i.e., the Licensing System reports that the license is still valid (tested in step 85), the program again picks the next license (step 81) to begin processing again.

10 If the ping is unsuccessful, a failure counter is incremented (step 86), and there is a test (step 87) to determine if the failure counter is above an established threshold. If it is, then the failure counter is cleared (step 88) and the license in question is moved to the recovery list (step 89). If it is not, then the program again picks the next license (step 81) to begin processing again.

Fig. 9 is a block diagram of clock message processing for licenses moved to the recovery list in step 89 of Fig. 8. First, a license is picked as part of a cycle through the recovery list of licenses in memory (step 91). Next there is a check whether a process exists for this license (step 911). If there is no process, any remaining part of the license is returned to the Licensing System, and associated housekeeping is done (step 94), and the program then picks the next license (step 91) to begin processing again. A check (in step 912) is made to determine whether the exit flag had been set in step 935, and if so, the process of the software product (application) is signalled to exit (step 913), the exit counter is decremented (step 914), and a test (step 915) is made to determine if the exit counter has reached zero. If so, the application process is killed (step 916). In either event, the next license is picked from the recovery list (91), and processing for the next license resumes as before.

If the exit flag had not been set, then a replacement license is sought (step 921), and a test (922) is made to

- 18 -

determine whether a license has been granted. If a replacement license has been granted, then

If a replacement license has not been granted, the replacement failure counter is incremented (step 93) and
5 then tested (step 931) to determine if it is above a threshold (here typically 3). If it is not above the threshold, then the next license is picked from the recovery list (91), and processing for the next license resumes as before. If it is above the threshold, the
10 polciy server database file 14 is consulted (step 932) to determine whether running of the software product is permitted (step 933). If not, then the exit counter and exit flag are set up; if running is permitted, the replacement failure counter is decremented (step 934). In
15 either case, the next license is picked from the recovery list (91), and processing for the next license resumes as before.

Many other implementations of the invention described herein are possible. For example, the particular types of
20 licenses described here are merely examples. The use of base and version licenses are thus a matter of design choice. The manner in which the failure to obtain a license is handled can also be tailored to suit the policies of the licensor of the software products in question.

- 19 -

What is claimed is:

1. An improved system for administration, on a computer network, of license terms for use of a software product on the network, the system being of the type wherein the
5 network has a plurality of digital computers, each computer at a node, in communication with each other over a data path, and the system has usage tracking means, associated with one of the computers acting as a license server, for (i) causing the storage of the number of licenses available
10 for running the software product on nodes of the network, (ii) identifying the current set of nodes with respect to which a license has been granted to run the software product at a given time, and (iii) determining whether at any given time any licenses remain to be granted for permitting an
15 additional node to run the software product, so that the software product may include instructions to cause enforcement of the license terms;

wherein the improvement comprises:

(a) a policy server database containing data
20 specifying conditions under which usage of the software product is permitted on any given node; and
(b) policy server means, maintained and operating locally as an independent process, on each computer, with respect to which the license terms are to be enforced, in
25 association with the policy server database, for (i) communicating with the license server, (ii) interfacing with both the software product and the policy server database, and (iii) making a permission-to-run availability determination, with respect to local usage of the software
30 product, on the basis of applicable data from the license server and the policy server database, so that enforcement of license terms applicable to the software product at a given local node is achieved on the basis of both license policy maintained in the policy server database as well as
35 applicable data from the license server.

2. A system according to claim 1, wherein each computer at a node with respect to which license terms are to be

- 20 -

enforced includes means for maintaining locally a policy server database, containing data specifying conditions under which usage of the software product is permitted on such node.

5 3. A system according to claim 1, further comprising:

(c) log means for recording and maintaining a log file of recent software product usage on each computer at a node with respect to which license terms are to be enforced, such log file being accessible to such policy server means, and
10 wherein such policy server means includes means for making a permission-to-run availability determination in the absence of data from the license server on the basis of data from the policy server database and the log file, so that a favorable determination is possible if the log file
15 indicates a sufficient level of recent usage of the pertinent software product on the computer on which such policy server means is operating.

4. A system according to claim 2, wherein

(i) each policy server database contains data
20 specifying conditions under which usage of each of plurality of software products is permitted on the computer on which the database is maintained, and

(ii) each policy server means includes means for interfacing with each of the software products,
25 so that enforcement of license terms applicable to each software product at a given local node may be achieved on the basis of both license policy maintained at such local node as well as applicable data from the license server.

5. A system according to claim 4, further comprising:

30 (c) log means, maintained locally in association with each policy server means, for recording and maintaining a log file of recent software product usage on the computer on which such log means is maintained, such log file being accessible to such policy server means, and wherein such
35 policy server means includes means for making a permission-to-run availability determination in the absence of data from the license server on the basis of data from the policy

- 21 -

server database and the log file, so that a favorable determination is possible if the log file indicates a sufficient level of recent usage of the pertinent software product on the computer on which such policy server means is
5 operating.

6. A system according to claim 1, wherein the policy server database is encrypted.

7. A system according to claim 5, wherein the policy server database and the log file are encrypted.

10 8. A system according to claim 4, wherein the policy server means include means for maintaining a secure interface with each of the software products.

9. A system according to claim 7, wherein the policy server means includes means for maintaining a secure
15 interface with each of the software products.

10. A system according to claim 4, wherein one of the policy server databases includes a limit on the number of nodes that may simultaneously use a given software product and wherein the corresponding policy server means associated
20 with such policy server database includes reservation means for informing the license server, over a predetermined time interval, that the node associated with such policy server means is using the given software product, regardless whether such software product is actually being used, so
25 that such node will always be available to use such software product, despite attempts to use the software product at other nodes which if successful would otherwise foreclose use at such node of such software product, with the effect that the reservation means reserves use of such software
30 product at such node over the predetermined time interval.

11. A system according to claim 5, wherein one of the policy server databases includes a limit on the number of nodes that may simultaneously use a given software product and wherein the corresponding policy server means associated
35 with such policy server database includes reservation means for informing the license server, over a predetermined time interval, that the node associated with such policy server

- 22 -

means is using the given software product regardless of whether such software product is actually being used, so that such node will always be available to use such software product despite attempts to use the software product at other nodes which if successful would otherwise foreclose use at such node of such software product, with the effect that the reservation means reserve use of such software product at such node over the predetermined time interval.

12. A system according to claim 9, wherein one of the policy server databases includes a limit on the number of nodes that may simultaneously use a given software product and wherein the corresponding policy server means associated with such policy server database includes reservation means for informing the license server, over a predetermined time interval, that the node associated with such policy server means is using the given software product, regardless whether such software product is actually being used, so that such node will always be available to use such software product, despite attempts to use the software product at other nodes which if successful would otherwise foreclose use at such node of such software product, with the effect that the reservation means reserves use of such software product at such node over the predetermined time interval.

13. A computer network comprising:

(a) a plurality of digital computers, each computer at a node, in communication with each other over a data path;

(b) usage tracking means, associated with one of the computers acting as a license server, for (i) causing the storage of the number of licenses available for running the software product on nodes of the data path, (ii) identifying the current set of nodes with respect to which a license has been granted to run the software product at a given time, and (iii) determining whether at any given time any licenses remain to be granted for permitting an additional node to run the software product;

(c) a policy server database, maintained locally on such computer with respect to which it is desired to enforce

- 23 -

license terms applicable to usage of the software products, containing data specifying conditions under which usage of any given one of the software products is permitted on the computer on which the database is maintained; and

5 (d) policy server means, maintained and operating locally, on each computer with respect to which it is desired to enforce license terms applicable to usage of the software products, and in association with the corresponding policy server database, for (i) communicating with the
10 license server, (ii) interfacing with both (aa) each of the software products and (bb) the corresponding policy server database, and (iii) making a permission-to-run availability determination, with respect to local usage of any given software product, on the basis of applicable data from the
15 license server and the corresponding policy server database, so that enforcement of license terms applicable to the given software product at a given node is achieved on the basis of the license policy maintained at such local node as well as applicable data from the license server.

20 14. A computer network according to claim 13, further comprising:

(e) log means, maintained locally in association with each policy server means, for recording and maintaining a log file of recent software product usage on the computer on
25 which such log means is maintained, such log file being accessible to such policy server means, and wherein such policy server includes means for making a permission-to-run availability determination in the absence of data from the license server on the basis of data from the policy server
30 database and the log file, so that a favorable determination is possible if the log file indicates a sufficient level of recent usage of the pertinent software product on the computer on which such policy server is operating.

15. A computer network according to claim 14, wherein the
35 policy server database and the log file are encrypted.

16. A computer network according to claim 12, wherein the policy server means includes means for maintaining a secure

interface with each of the software products.

17. A computer network according to claim 15, wherein the policy server means includes means for maintaining a secure interface with each of the software products.

5 18. A digital storage medium encoded with instructions for a given computer in a computer network of the type having:

(i) a plurality of digital computers, each computer at a node, in communication with each other over a data path;

(ii) usage tracking means, associated with one of the
10 computers acting as a license server, for (i) causing the storage of the number of licenses available for running the software product on nodes of the network, (ii) identifying the current set of nodes with respect to which a license has been granted to run the software product at a given time,
15 and (iii) determining whether at any given time any licenses remain to be granted for permitting an additional node to run the software product,

the instructions when loaded into the given computer establishing:

20 (a) data structure for a policy server database, maintained locally on the given computer, containing data specifying conditions under which usage of any given one of the software products is permitted on the given computer; and

25 (b) policy server means, maintained and operating locally, on the given computer, and in association with the policy server database, for (i) communicating with the license server, (ii) interfacing with both (aa) each of the software products and (bb) the policy server database, and
30 (iii) making a permission-to-run availability determination, with respect to local usage of any given software product, on the basis of applicable data from the license server and the policy server database, so that enforcement of license terms applicable to the given software product at the given
35 computer is achieved on the basis of the license policy maintained at the given computer as well as applicable data from the license server.

- 25 -

19. A system, for administration of license terms for use of a software product on a computer, comprising:

(a) a policy server database containing data specifying conditions under which usage of the software product is

5 permitted on the computer;

(b) policy server means, operating on the computer, in association with the policy server database, for (i) interfacing with the software product and the policy server database and (ii) making a permission-to-run availability

10 determination, with respect to usage of the software product, on the basis of data from the policy server database.

15

1/9

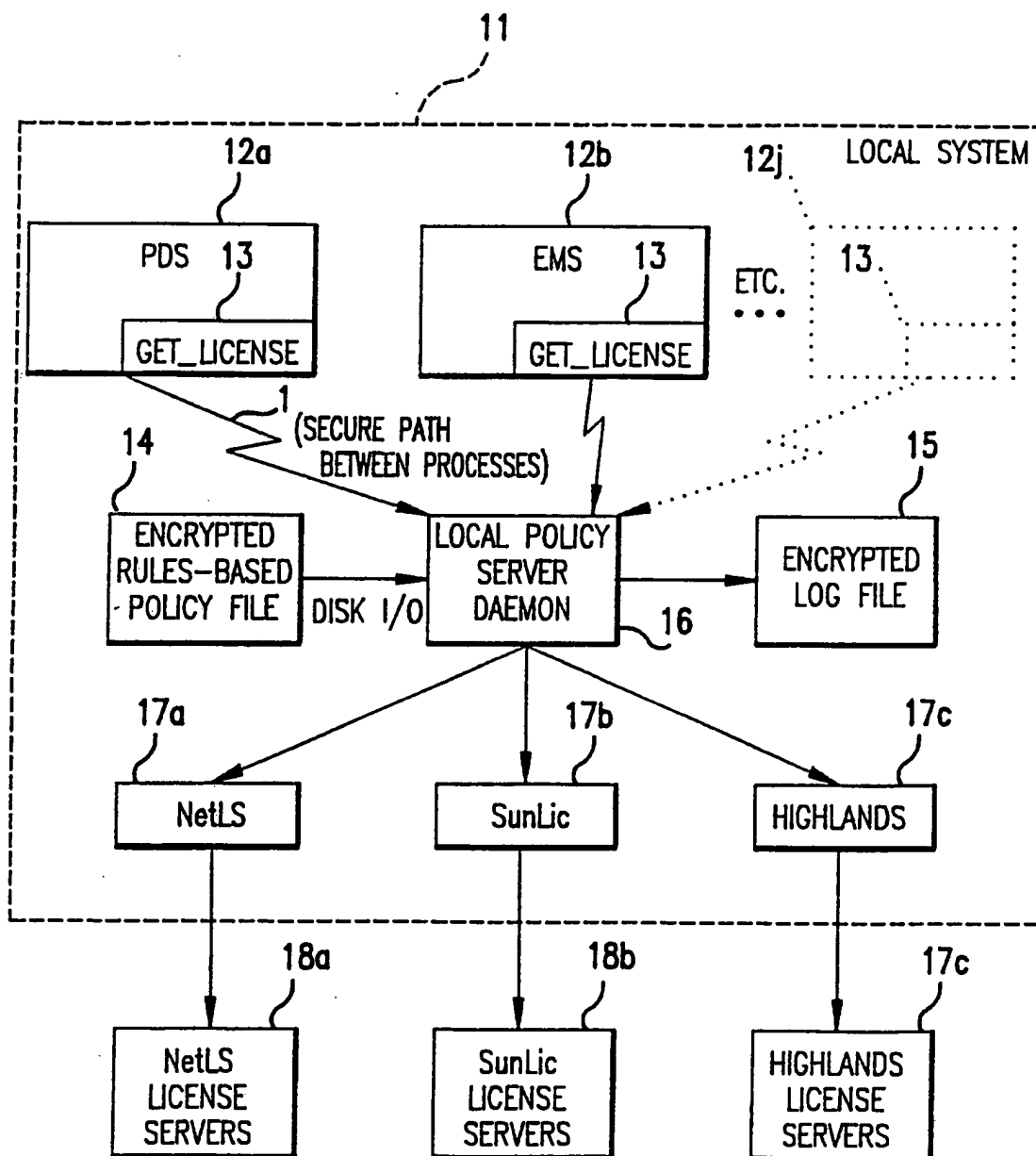
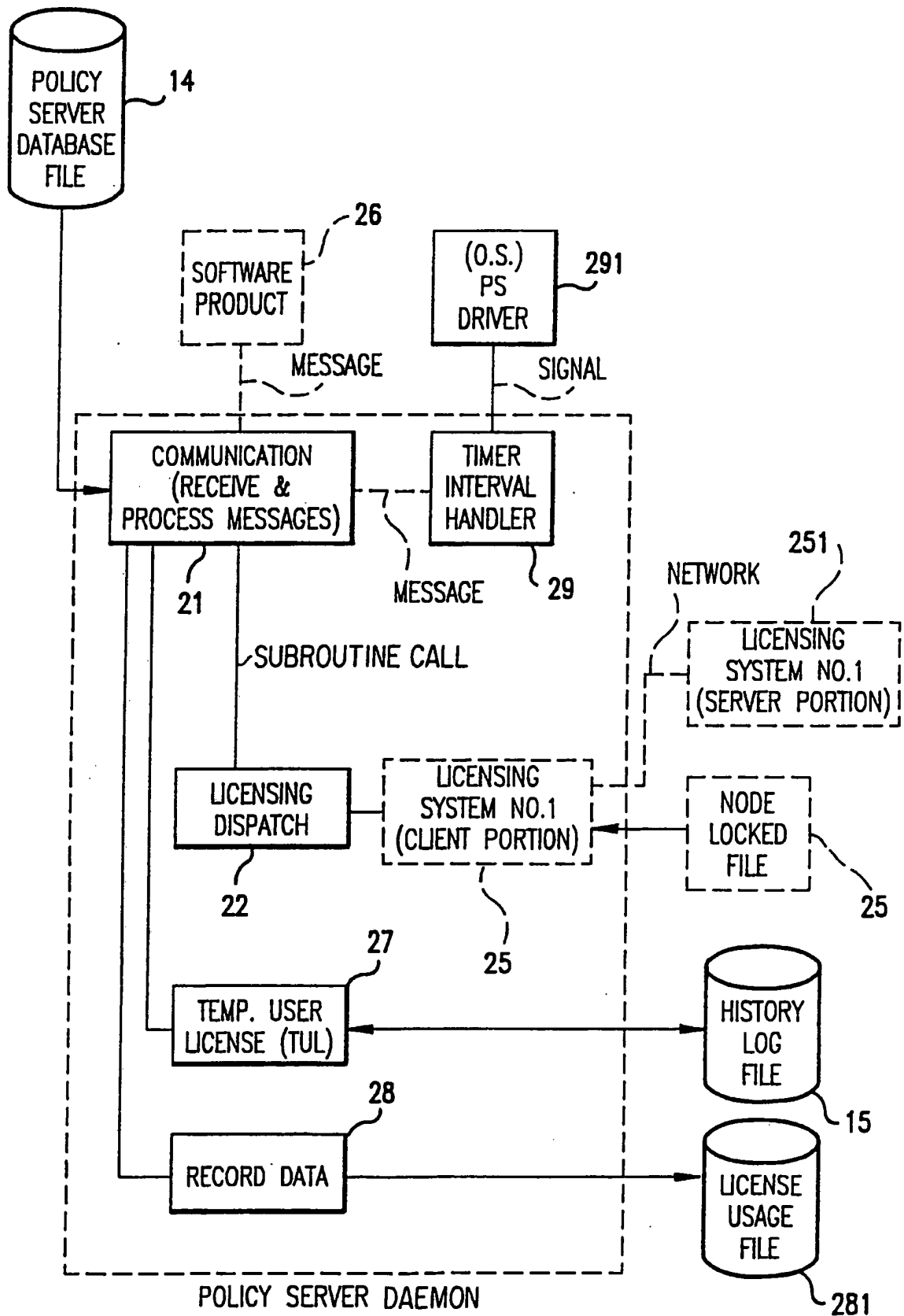


FIG.1

SUBSTITUTE SHEET

2/9



SUBSTITUTE SHEET

FIG.2

3/9

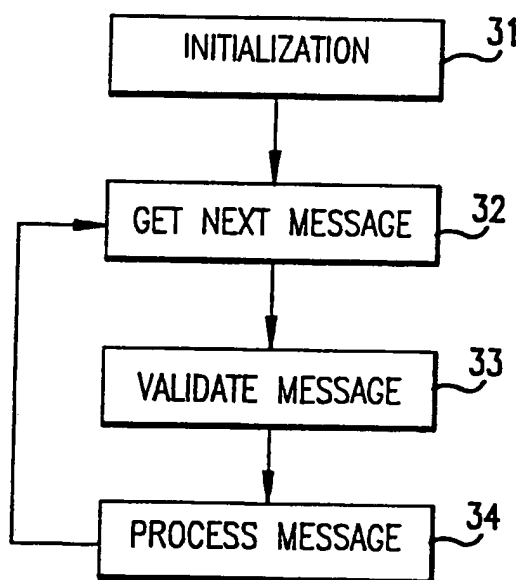


FIG.3

4/9

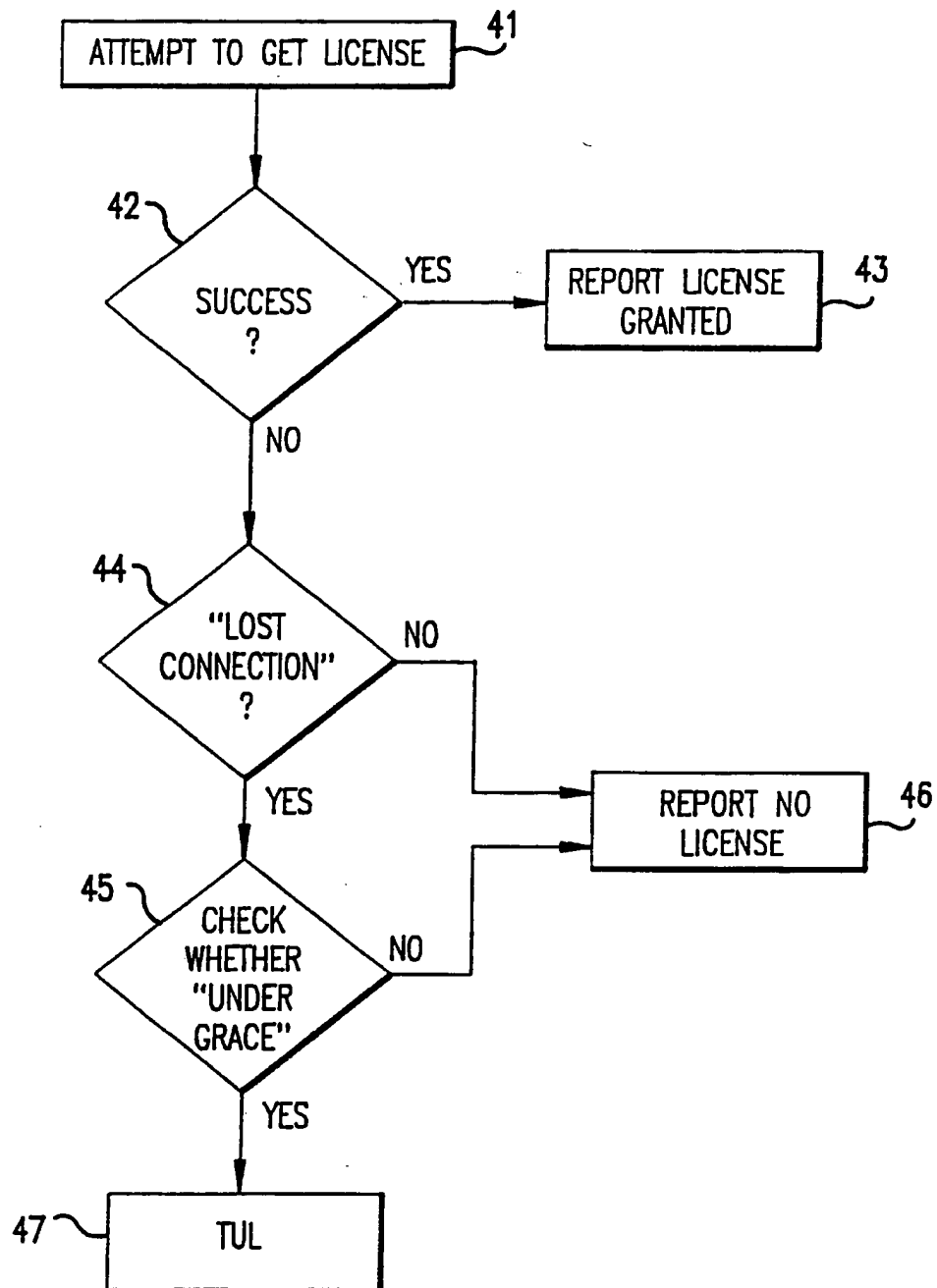


FIG. 4

SUBSTITUTE SHEET

5/9

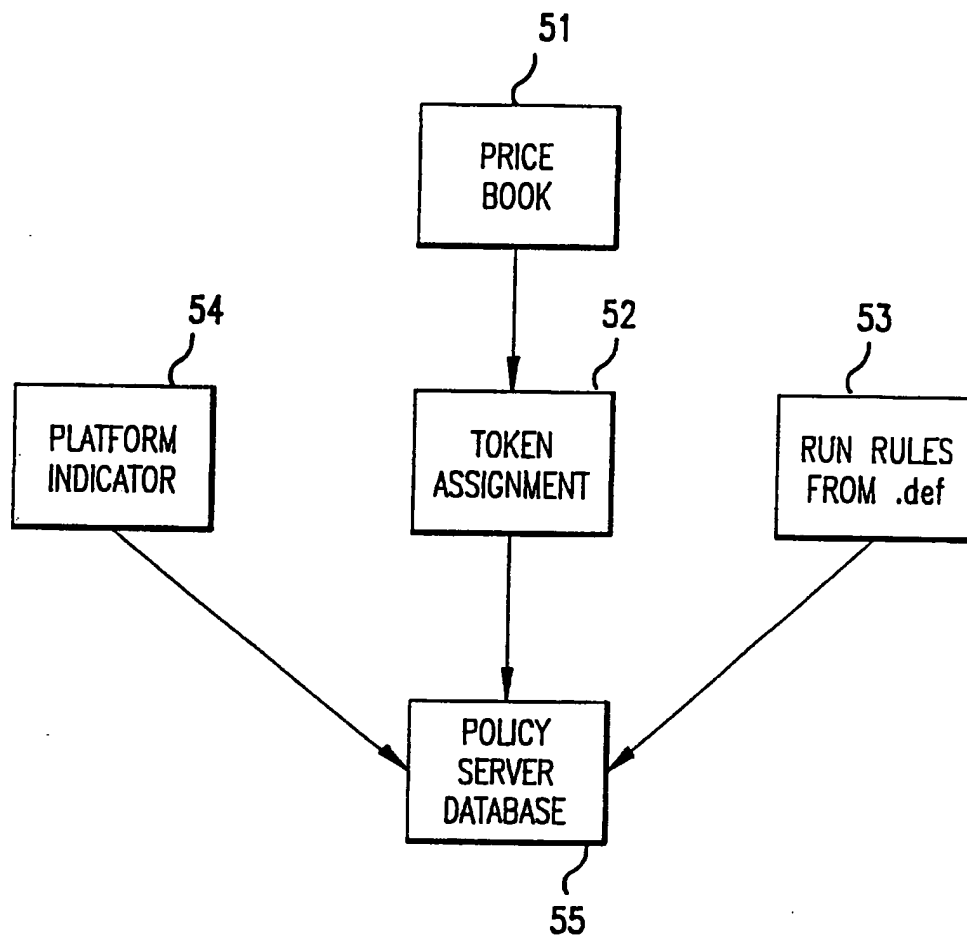


FIG.5

6/9

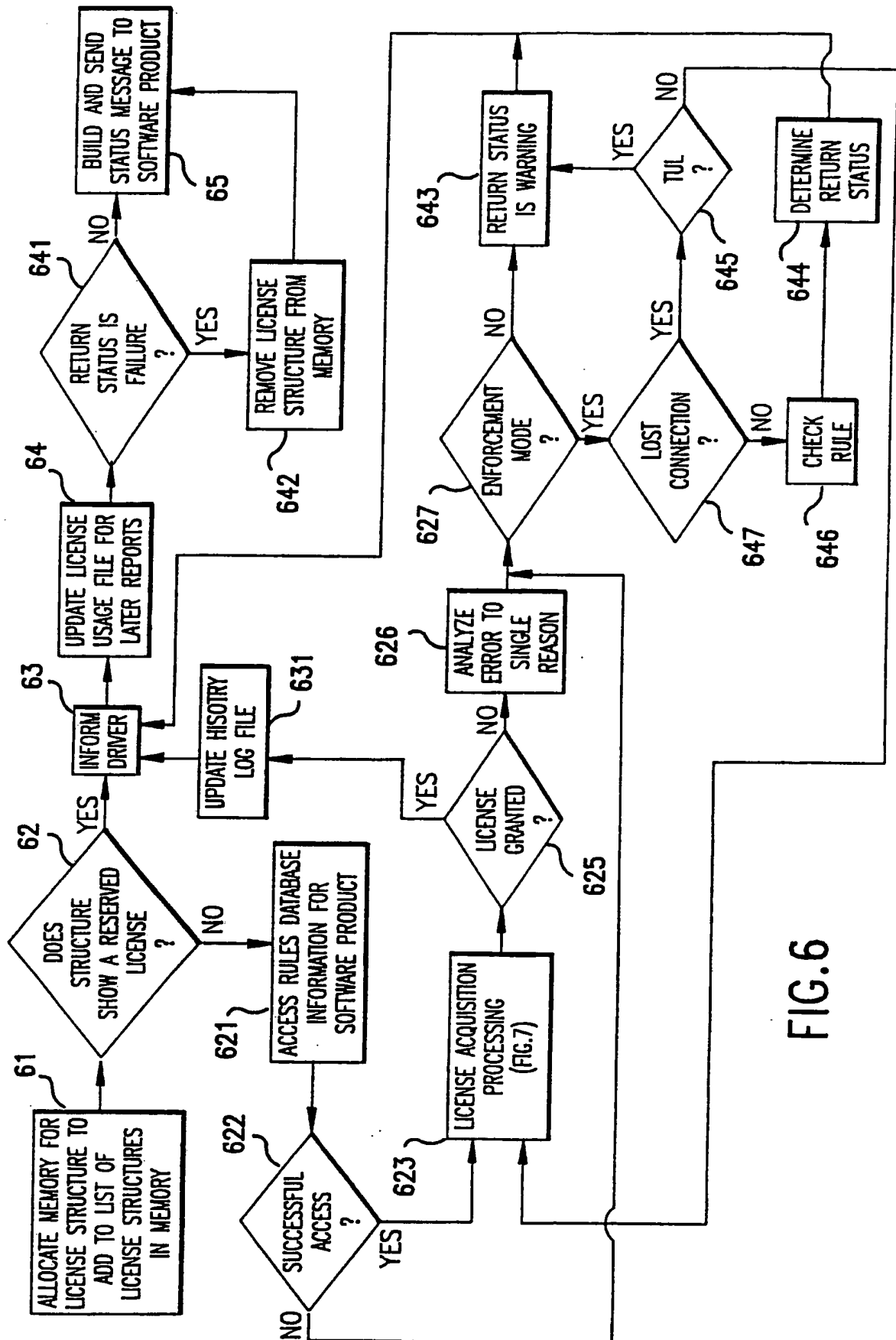
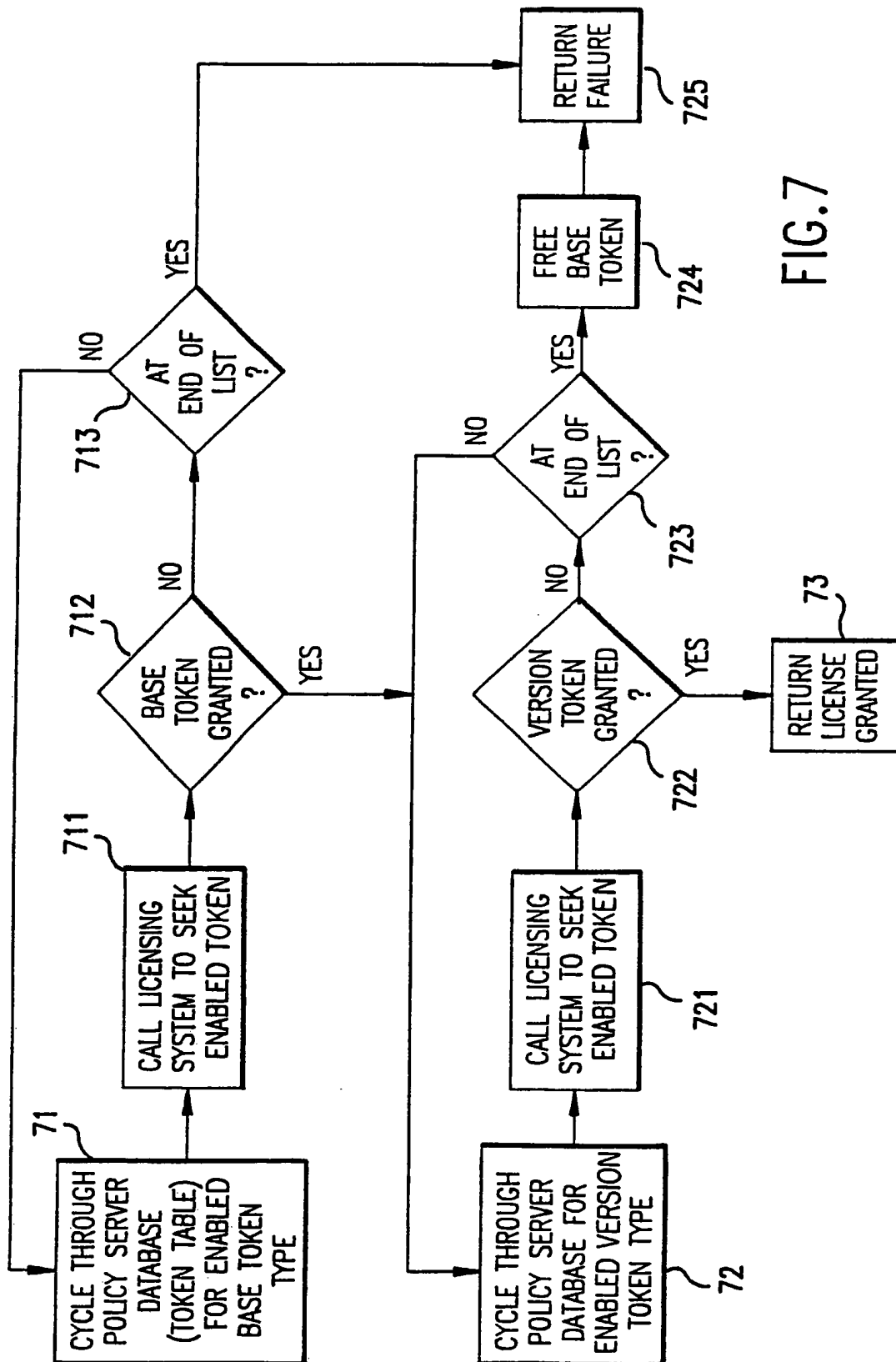
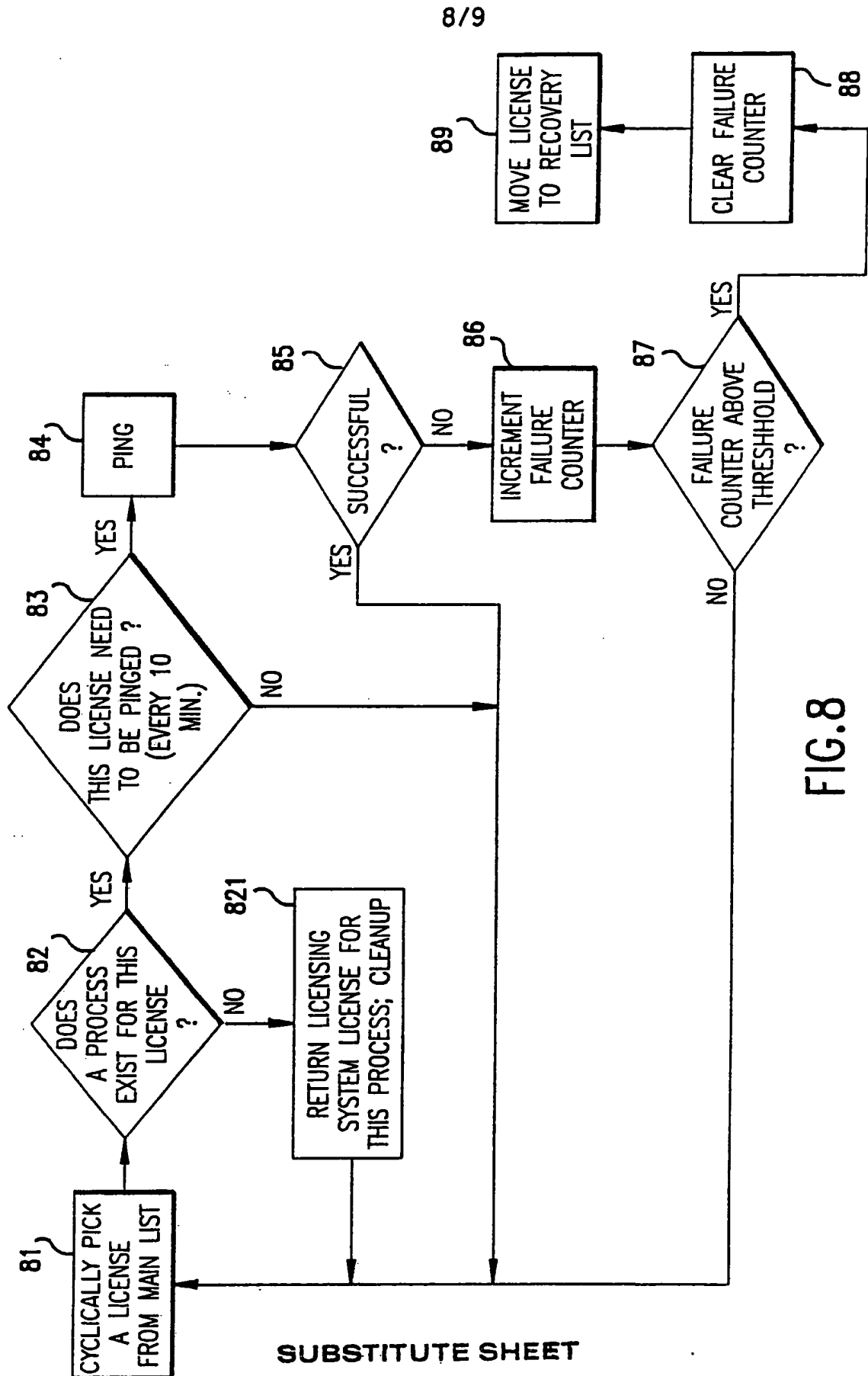


FIG. 6

7/9



SUBSTITUTE SHEET



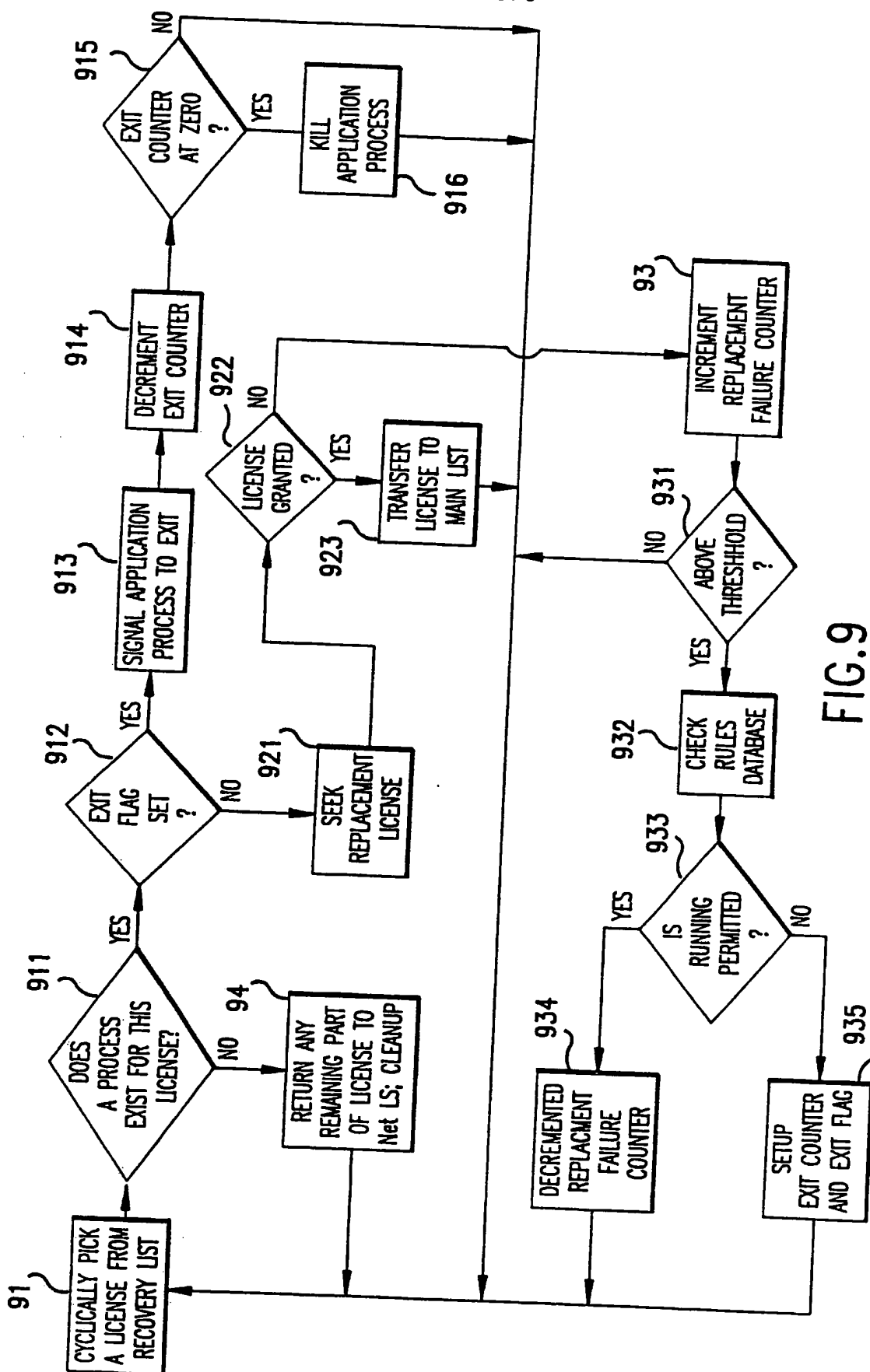


FIG. 9

INTERNATIONAL SEARCH REPORT

PCT/US 92/10215

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; G06F11/34		
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
A	GB,A,2 236 604 (SUN MICROSYSTEMS, INC.) 10 April 1991 see page 9, line 11 - page 10, line 28 see page 12, line 16 - page 13, line 13 see page 14, line 20 - page 16, line 27 see figures 1-3 ---	1-7, 13-15, 18-19
A	US,A,5 023 907 (APOLLO COMPUTER) 11 June 1991 see column 2, line 49 - column 5, line 42 see figures 1,2 --- -/-	1-5, 13-14, 18-19
¹⁰ 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	
26 FEBRUARY 1993	23.03.93	
International Searching Authority	Signature of Authorized Officer	
EUROPEAN PATENT OFFICE	JOHANSSON U.C.	

Form PCT/ISA/210 (second sheet) (January 1983)

III. DOCUMENTS CONSIDERED TO BE RELEVANT (CONTINUED FROM THE SECOND SHEET)		
Category °	Citation of Document, with indication, where appropriate, of the relevant passages	Relevant to Claim No.
A	EP,A,0 332 304 (DIGITAL EQUIPMENT CORP.) 13 September 1989 see column 3, line 31 - column 6, line 8 see column 6, line 42 - column 7, line 23 see column 8, line 33 - column 9, line 44 see figure 1 -----	1,2, 10-13,18

**ANNEX TO THE INTERNATIONAL SEARCH REPORT
ON INTERNATIONAL PATENT APPLICATION NO.**

US 9210215
SA 67461

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. 26/02/93

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
GB-A-2236604	10-04-91	US-A- 5138712	11-08-92
		CA-A- 2025434	03-04-91
		JP-A- 4100148	02-04-92

US-A-5023907	11-06-91	None	

EP-A-0332304	13-09-89	US-A- 4937863	26-06-90
		JP-A- 2014321	18-01-90

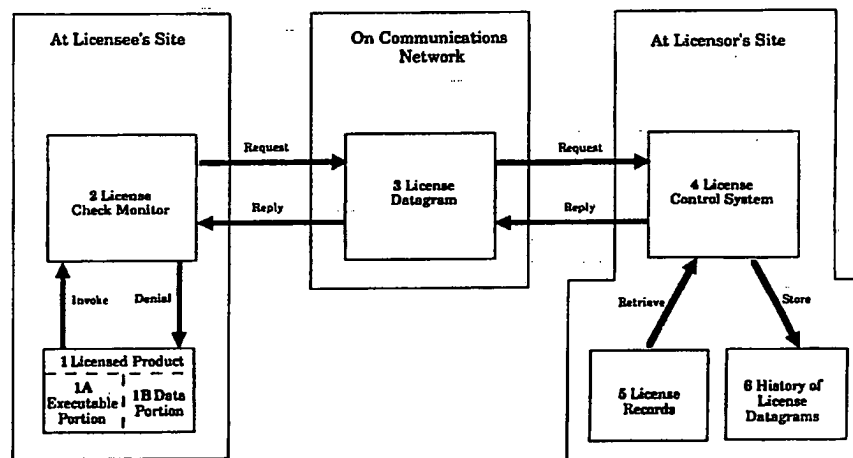
EPO FORM P0479

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 predetermined 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	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	LJ	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 -

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.

10 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."

5 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.

10 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).

25 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.

30 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.

5 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.

15 In the present invention, a licensed product generates request "datagrams," messages transmitted over a communications network. The request datagrams 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

- 16 -

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

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

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

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.

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 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 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 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

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 (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.
15

4
20 Your current use of this licensed product exceeds limits for the number of uses your organization has licensed. Please try again later.

5
25 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
5 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
10 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
15 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
20 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
25 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
30 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.

- 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

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

10 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.

15 28. A system as in claim 23, wherein said monitoring means transmits request datagrams at predetermined time intervals.

 29. A system as in claim 23, wherein:
 said monitoring means incorporates a unique identification code in said request datagram;

20 said controlling means incorporates the same request datagram identification code in said reply authorizing datagram; and

25 said monitoring means rejects any reply authorizing datagram which does not include the same identification code as included in said request 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

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

- 36 -

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

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
5 address of said licensee from said facility without 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

15 receiving a reply authorizing datagram at said licensee's site and denying the use of said product when no reply authorizing datagram is received.

52. A method as in claim 51 wherein:

20 said generating step includes the step of incorporating a model number of said product in said 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:

- 43 -

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
5 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.

71. A system as in claim 61, wherein:
10 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.

73. A system as in claim 71, wherein said data
20 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.

74. A system as in claim 61, wherein said
25 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.

79. A method as in claim 78 further wherein:

said datagrams include a model number of each product; and

5 said counting step includes the step of 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:

10 receiving a request datagram at a licensor's 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

- 48 -

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

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.

15 107. 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
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

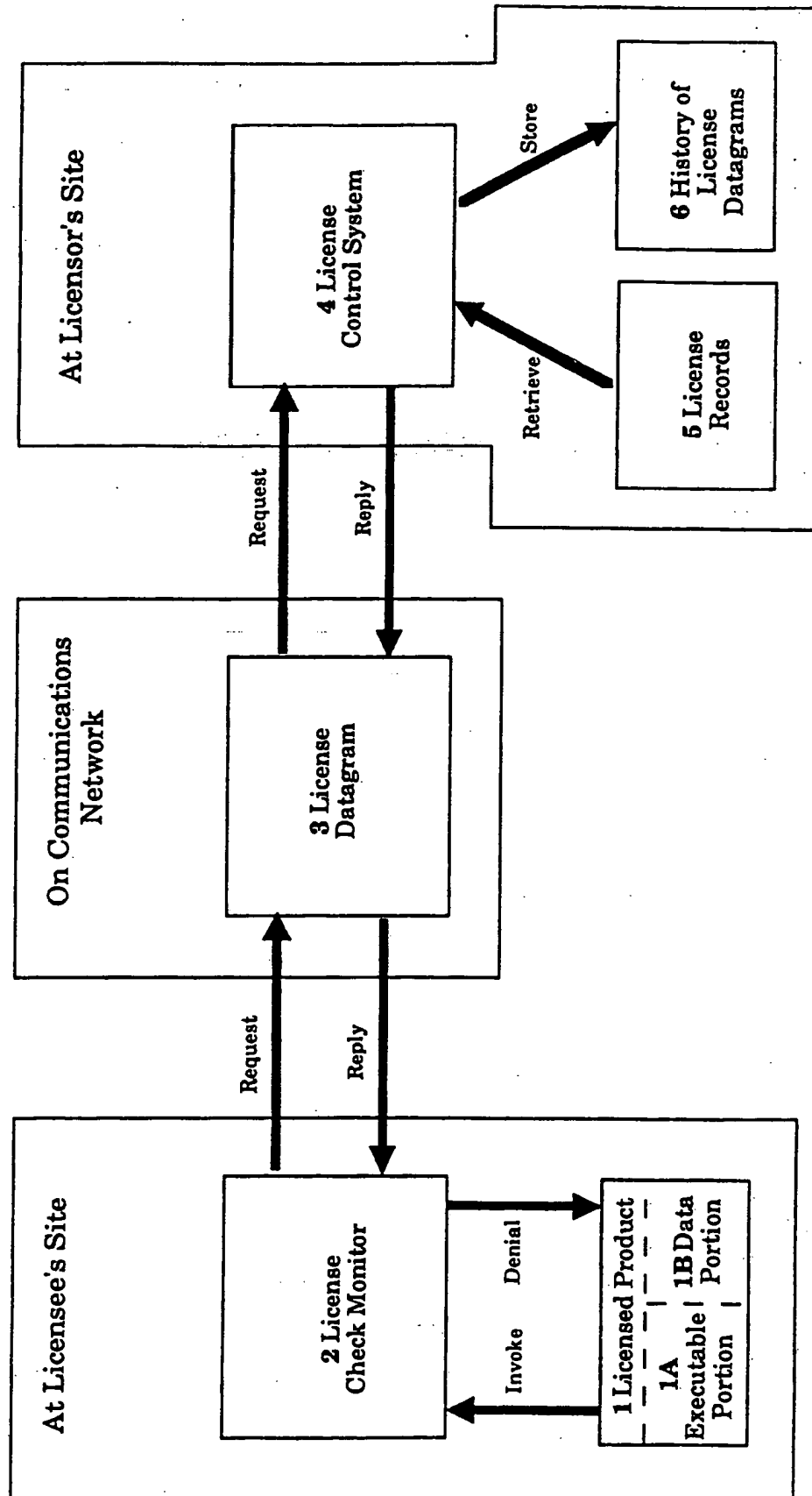
- 52 -

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
10 model number; and
said controlling means separately counts request datagrams for each product model number for each licensee.

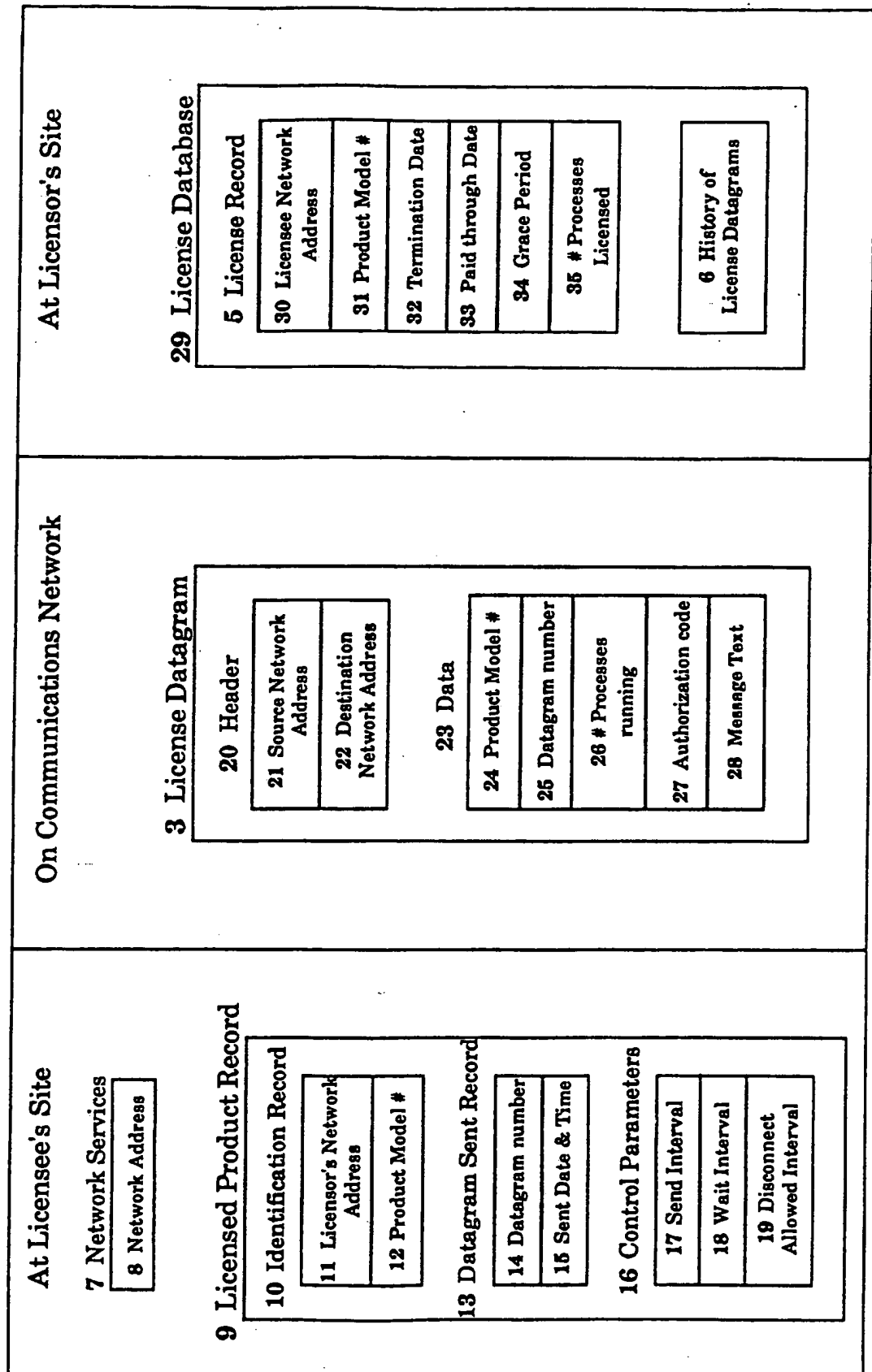
1 / 8

FIG. 1



2 / 8

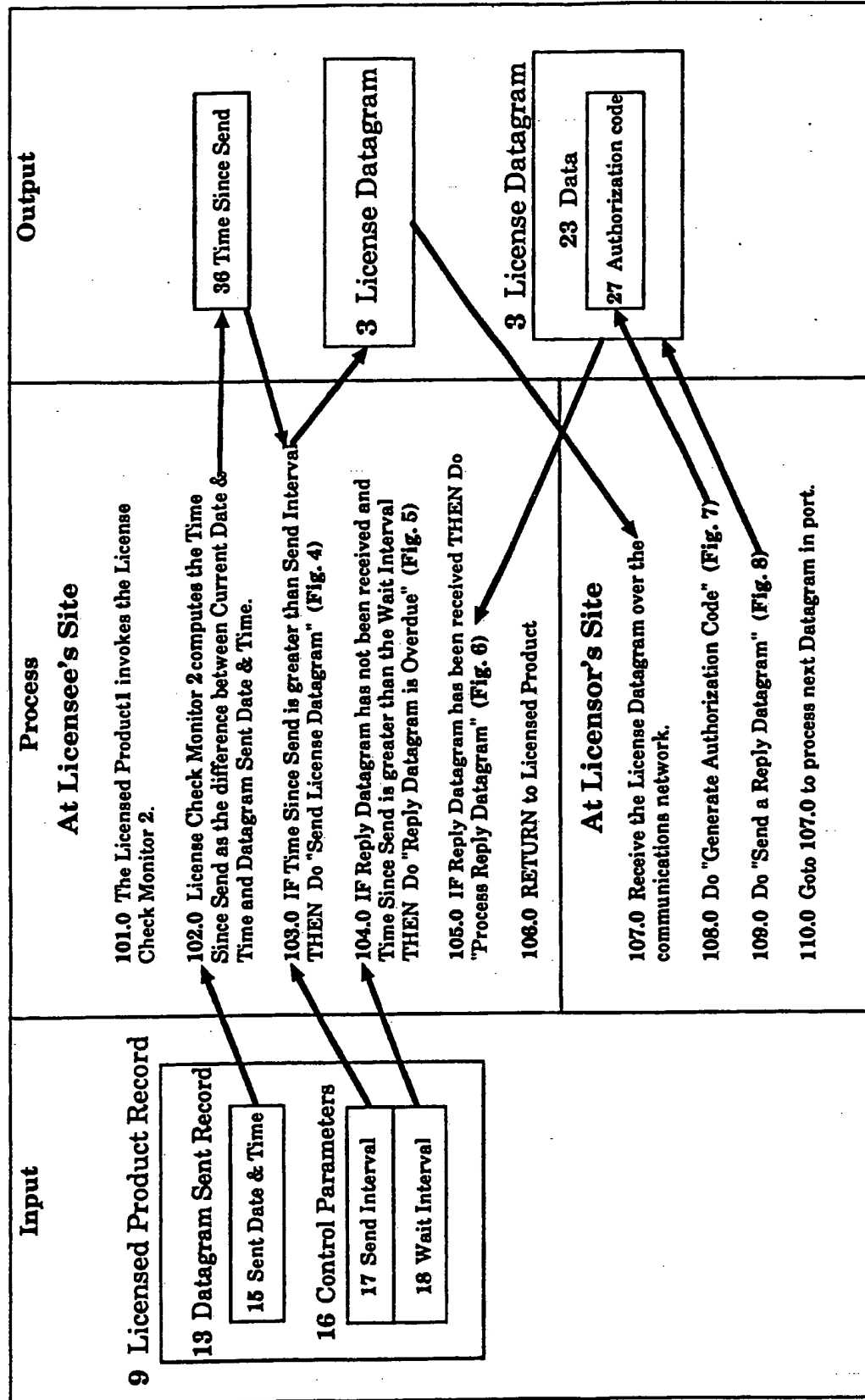
FIG. 2



SUBSTITUTE SHEET

33 / 8

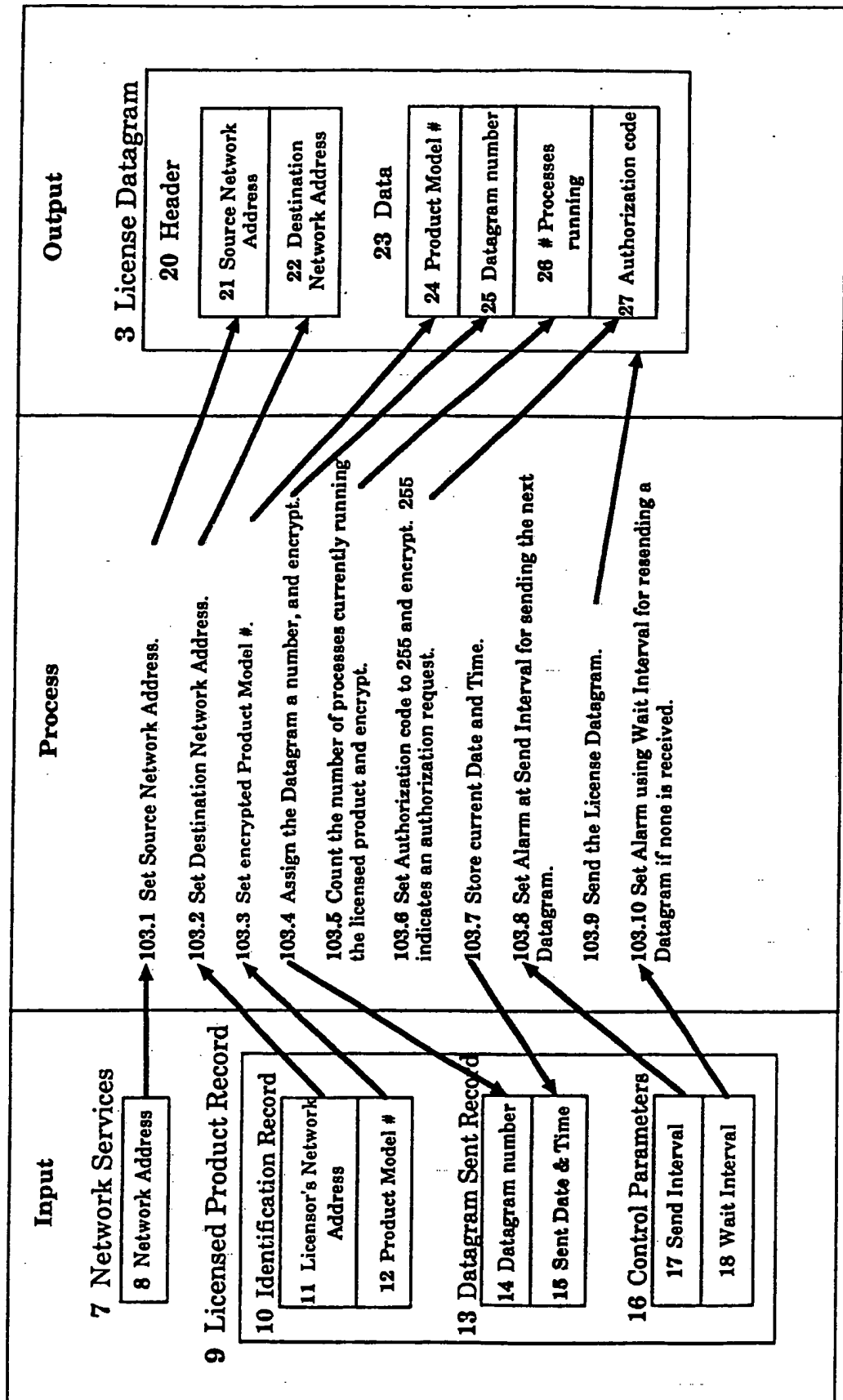
FIG. 3



SUBSTITUTE SHEET

4 / 8

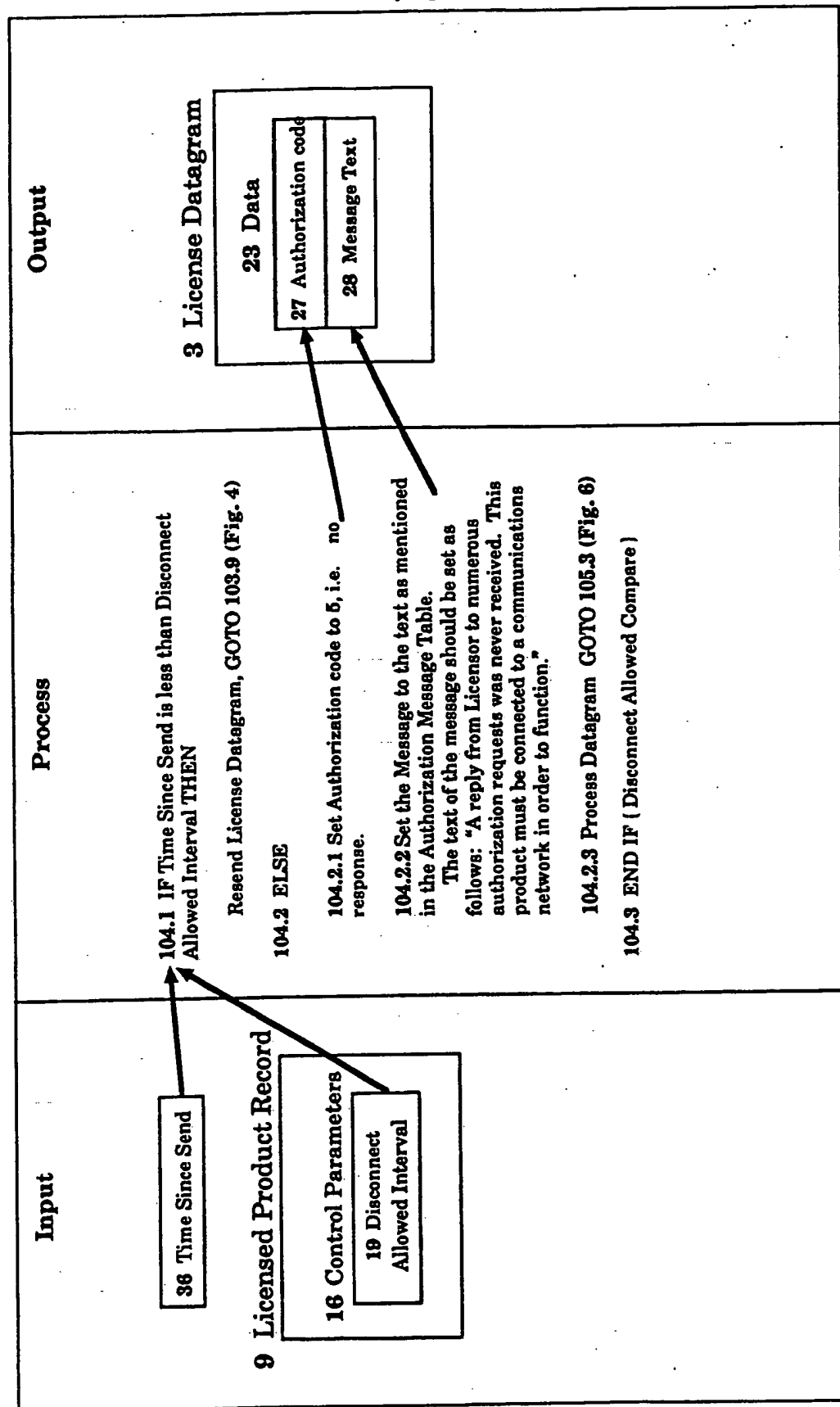
FIG. 4



SUBSTITUTE SHEET

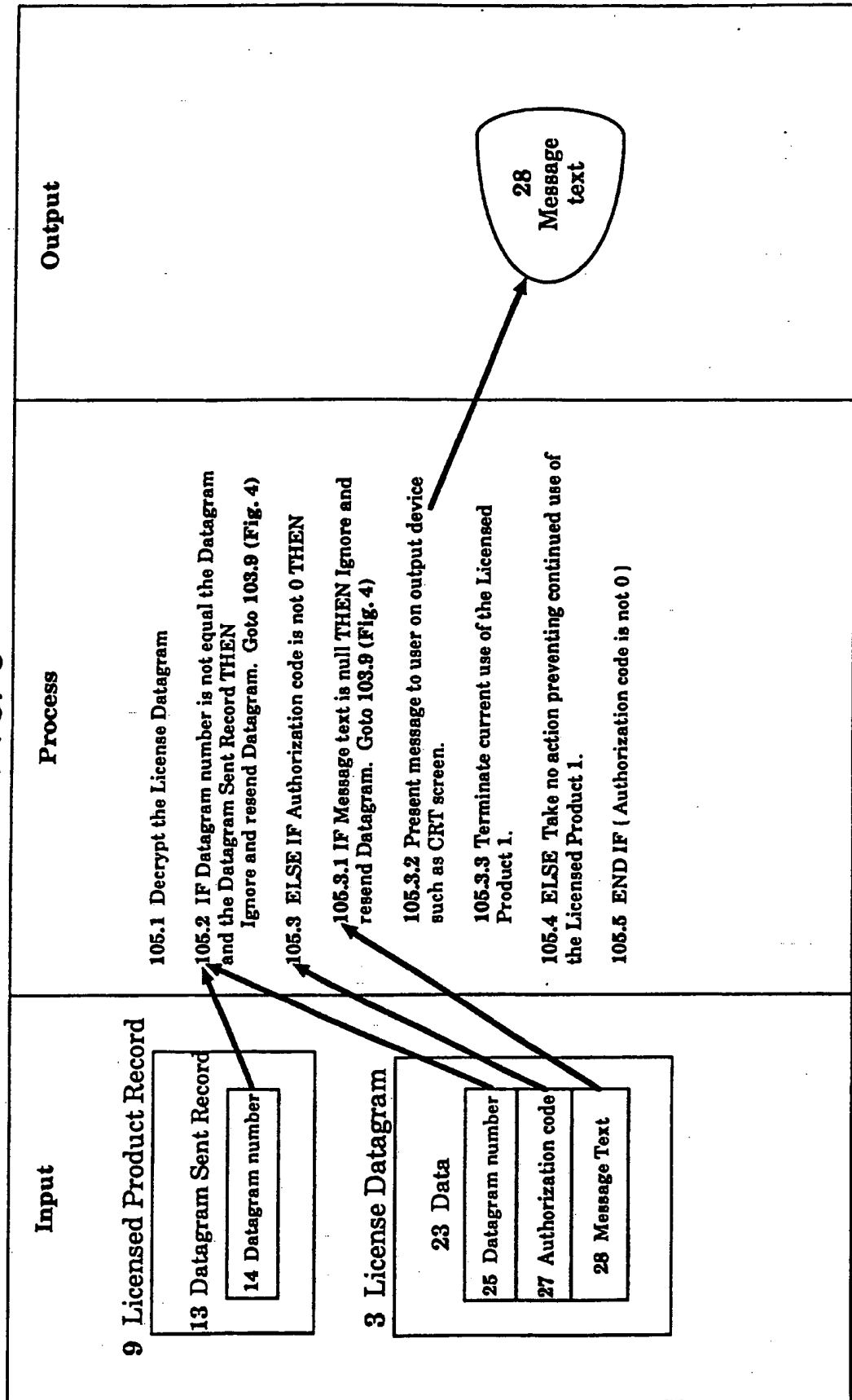
5 / 8

FIG. 5



6 / 8

FIG. 6



SUBSTITUTE SHEET

7 / 8

FIG. 7

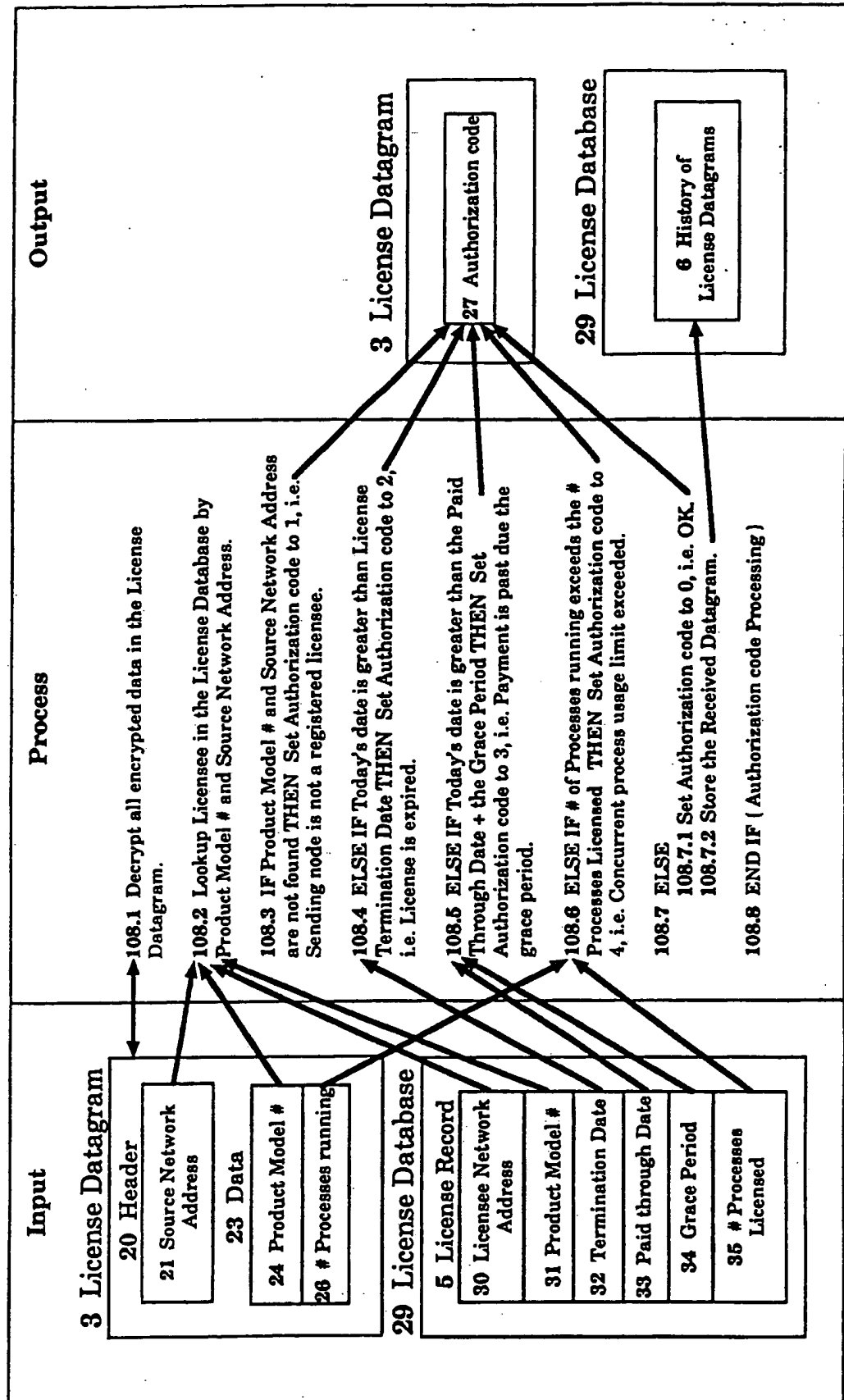
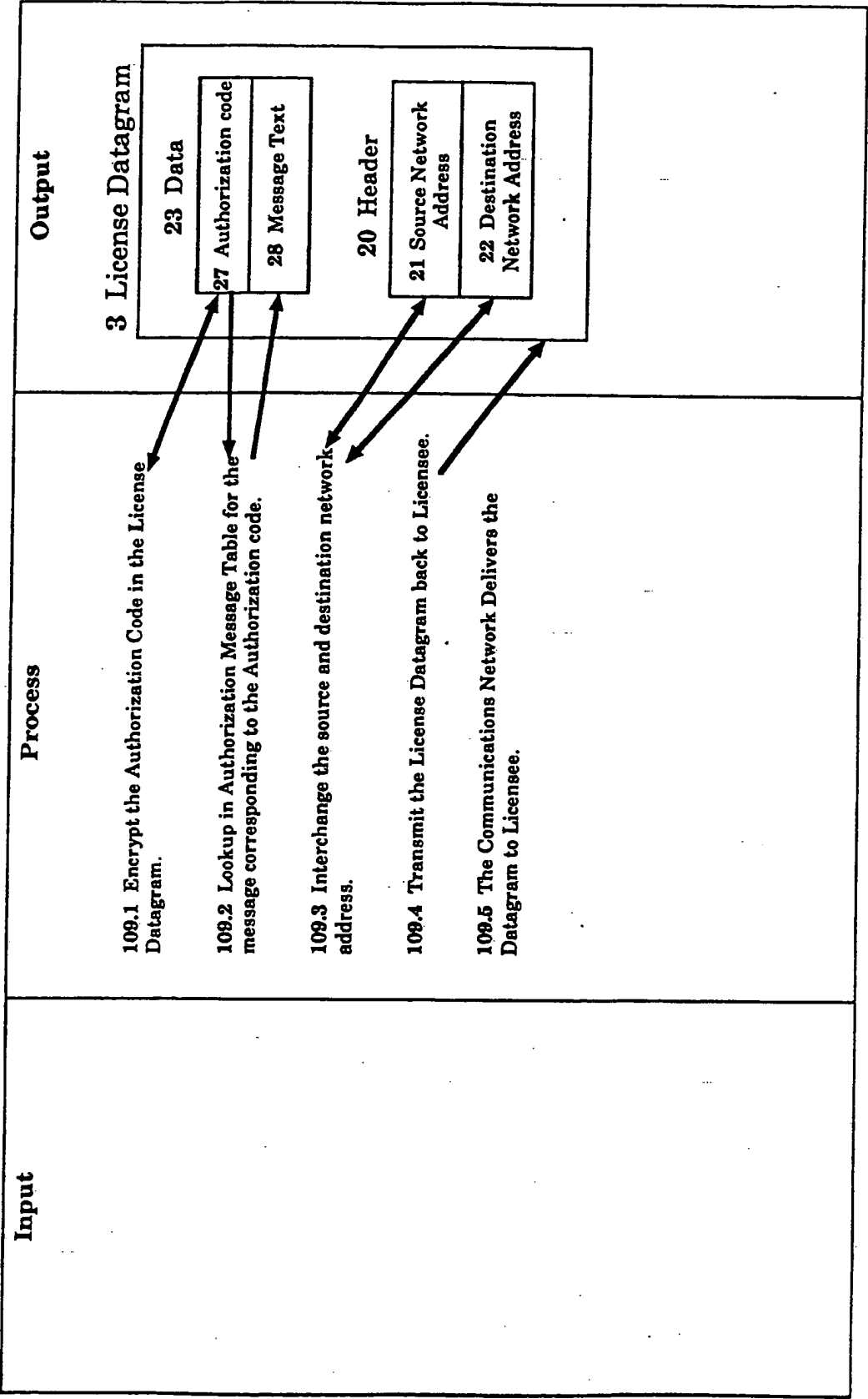


FIG. 8



INTERNATIONAL SEARCH REPORT

International application No.
PCT/US92/05387**A. CLASSIFICATION OF SUBJECT MATTER**

IPC(S) :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

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

(51) International Patent Classification ⁵ :	G06F 1/00	A1	(11) International Publication Number:	WO 92/20022
			(43) International Publication Date:	12 November 1992 (12.11.92)
(21) International Application Number:	PCT/US92/03812		(74) Agents:	NATH, Ram, B. et al.; c/o Joyce D. Lange, Digital Equipment Corporation, 111 Powdermill Road, Maynard, MA 01754 (US).
(22) International Filing Date:	6 May 1992 (06.05.92)		(81) Designated States:	AT, AT (European patent), AU, BB, BE (European patent), BF (OAPI patent), BG, BJ (OAPI patent), BR, CA, CF (OAPI patent), CG (OAPI patent), CH, CH (European patent), CI (OAPI patent), CM (OAPI patent), CS, DE, DE (European patent), DK, DK (European patent), ES, ES (European patent), FI, FR (European patent), GA (OAPI patent), GB, GB (European patent), GN (OAPI patent), GR (European patent), HU, IT (European patent), JP, KP, KR, LK, LU, LU (European patent), MC (European patent), MG, ML (OAPI patent), MR (OAPI patent), MW, NL, NL (European patent), NO, PL, RO, RU, SD, SE, SE (European patent), SN (OAPI patent), TD (OAPI patent), TG (OAPI patent).
(30) Priority data:	697,652	8 May 1991 (08.05.91)	US	
	723,456	28 June 1991 (28.06.91)	US	
	722,840	28 June 1991 (28.06.91)	US	
	723,457	28 June 1991 (28.06.91)	US	
(71) Applicant:	DIGITAL EQUIPMENT CORPORATION [US/US]; 146 Main Street, Maynard, MA 01754 (US).			
(72) Inventor:	WYMAN, Robert, Mark; 410 Second Avenue, South No. 108, Kirkland, WA 98033 (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 the receipt of amendments.	

FIG. 1 is a block diagram illustrating a license management system architecture. The system is divided into a central management layer and multiple client layers.

Central Management Layer:

- PRODUCER (26):** Connected to the LICENSE DOCUMENT GENERATOR (26) via a connection labeled "RIGHT TO ISSUE".
- LICENSE DOCUMENT GENERATOR (26):** Outputs to the LICENSE ISSUER (27) and the OS (27).
- LICENSE MANAGEMENT PROGRAM (10):** Contains sub-components: INTERSERVER IF (34), MANAGE. IF (33), DATABASE IF (32), and CLIENT IF (31).
- SERVER (12):** Connected to the LICENSE MANAGEMENT PROGRAM (10) and the LICENSE DATABASE (23).
- LICENSE DATABASE (23):** Connected to the SERVER (12) and the LOG (24).
- LOG (24):** Receives data from the LICENSE DATABASE (23).

Client Layer (Repeating Units):

- LICENSE MANAGE. (13):** Connected to the DELEGATEE SERVER (14) and the OS (15).
- DELEGATEE SERVER (14):** Connected to the LICENSE MANAGE. (13) and the OS (15).
- OS (15):** Connected to the LICENSE MANAGE. (13) and the DELEGATEE SERVER (14).
- USER CPU (16):** Connected to the OS (15).
- LICENSED PROGRAM (17):** Connected to the OS (15).

Interconnections:

- The central management layer is connected to the client layer via a bus structure (21).
- Each client unit (13, 14, 15, 16, 17) is connected to the central management layer via a connection labeled "22".
- Each client unit (13, 14, 15, 16, 17) is connected to the central management layer via a connection labeled "23".
- Each client unit (13, 14, 15, 16, 17) is connected to the central management layer via a connection labeled "24".
- Each client unit (13, 14, 15, 16, 17) is connected to the central management layer via a connection labeled "25".
- Each client unit (13, 14, 15, 16, 17) is connected to the central management layer via a connection labeled "26".
- Each client unit (13, 14, 15, 16, 17) is connected to the central management layer via a connection labeled "27".
- Each client unit (13, 14, 15, 16, 17) is connected to the central management layer via a connection labeled "28".
- Each client unit (13, 14, 15, 16, 17) is connected to the central management layer via a connection labeled "29".
- Each client unit (13, 14, 15, 16, 17) is connected to the central management layer via a connection labeled "30".
- Each client unit (13, 14, 15, 16, 17) is connected to the central management layer via a connection labeled "31".
- Each client unit (13, 14, 15, 16, 17) is connected to the central management layer via a connection labeled "32".
- Each client unit (13, 14, 15, 16, 17) is connected to the central management layer via a connection labeled "33".
- Each client unit (13, 14, 15, 16, 17) is connected to the central management layer via a connection labeled "34".

A distributed computer system employs a license management system to account for software product usage. A management policy having a variety of alternative styles and contexts is provided. Each licensed product upon start-up makes a call to a license server to check on whether usage is permitted, and the license server checks a database of the licenses, called product use authorizations, that it administers. If the particular use requested is permitted, a grant is returned to the requesting user node. The product use authorization is structured to define a license management policy allowing a variety of license alternatives by values called "style", "context", "duration" and "usage requirements determination method". The license administration may be delegated by the license server to a subsection of the organization, by creating another license management facility duplicating the main facility. The license server must receive a license document (a product use authorization) from an issuer of licenses, where a license document generator is provided. A mechanism is provided for one user node to make a call to use a software product located on another user node; this is referred to as a "calling card", by which a user node obtains permission to make a procedure call to use a program on another node. A management interface allows a license manager at a server to modify the license documents in the database maintained by the server, within the restraints imposed by the license, to make delegations, assignments, etc. The license documents are maintained in a standard format referred to as a license document interchange format so the management system is portable and can be used by all adhering software vendors. A feature of the database management is the use of a filter function.

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	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
10 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
15 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
20 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
25 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 20 49a. The product 17a then makes a call to product 17b in the usual manner of 25

- 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
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
10 field 49 enabled, so calling cards can be issued. This feature is typically separately
priced.

15 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
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.

20 The overdraft field 51 determines whether or not a requested allocation
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
copies of the license. Thus, the product use authorization sent by the issuer 25 to
25

- 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 -

5 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.

15 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.

20

25

- 13 -

5 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.

10 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).

25 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 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.

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.

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. 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 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 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

- 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,

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

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 RdbTM) 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. 15 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 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

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.

25 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 -

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.

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., 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.

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 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;

- 44 -

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;

5 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

10 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
15 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
20 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 or 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 -

5 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).

10 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
15 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
20 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.

- 59 -

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 filed 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 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 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 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 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 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 lur-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 -

5 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.

10 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).

20 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

25

- 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 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.

5 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.

10 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
15	greater or equal	select
	less or equal	request candidates
	greater	simulate request

20 (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
 25 present, this is the substring to compare against the initial portion of the value of

- 93 -

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;

88

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:

93

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;

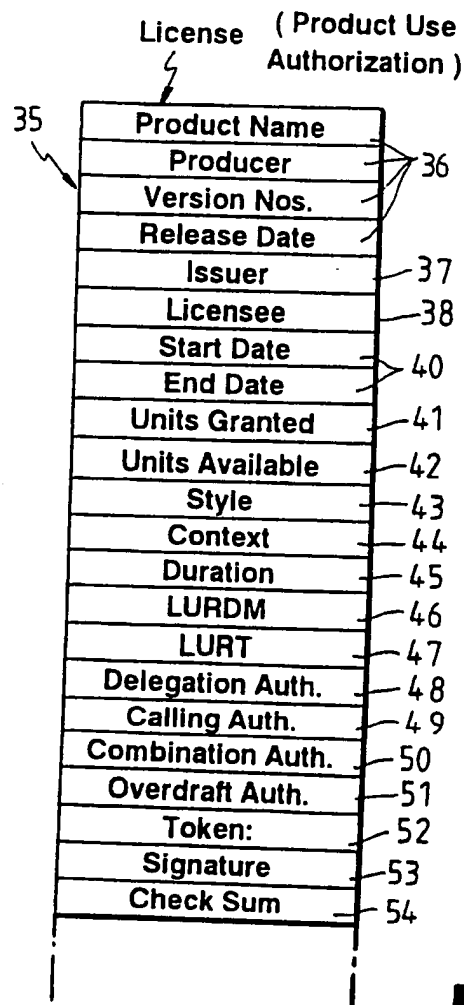
14 means for accessing said store to obtain information from said
15 license authorization for said software item, in response to said
16 request, and comparing said identification of said user and said
17 software item with said information, to produce a grant or refusal
18 of said request; and

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.

2/32



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

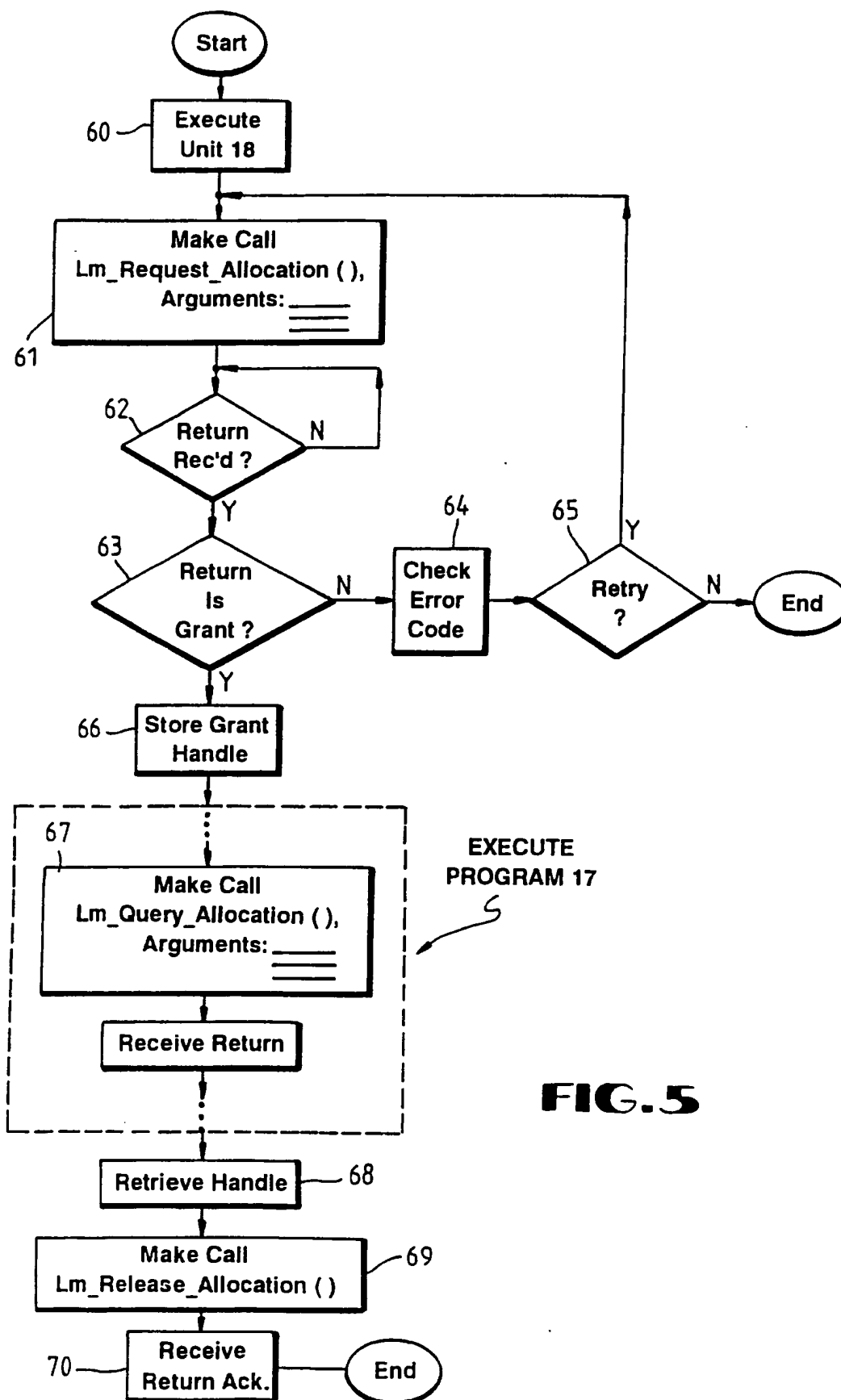
FIG.2

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

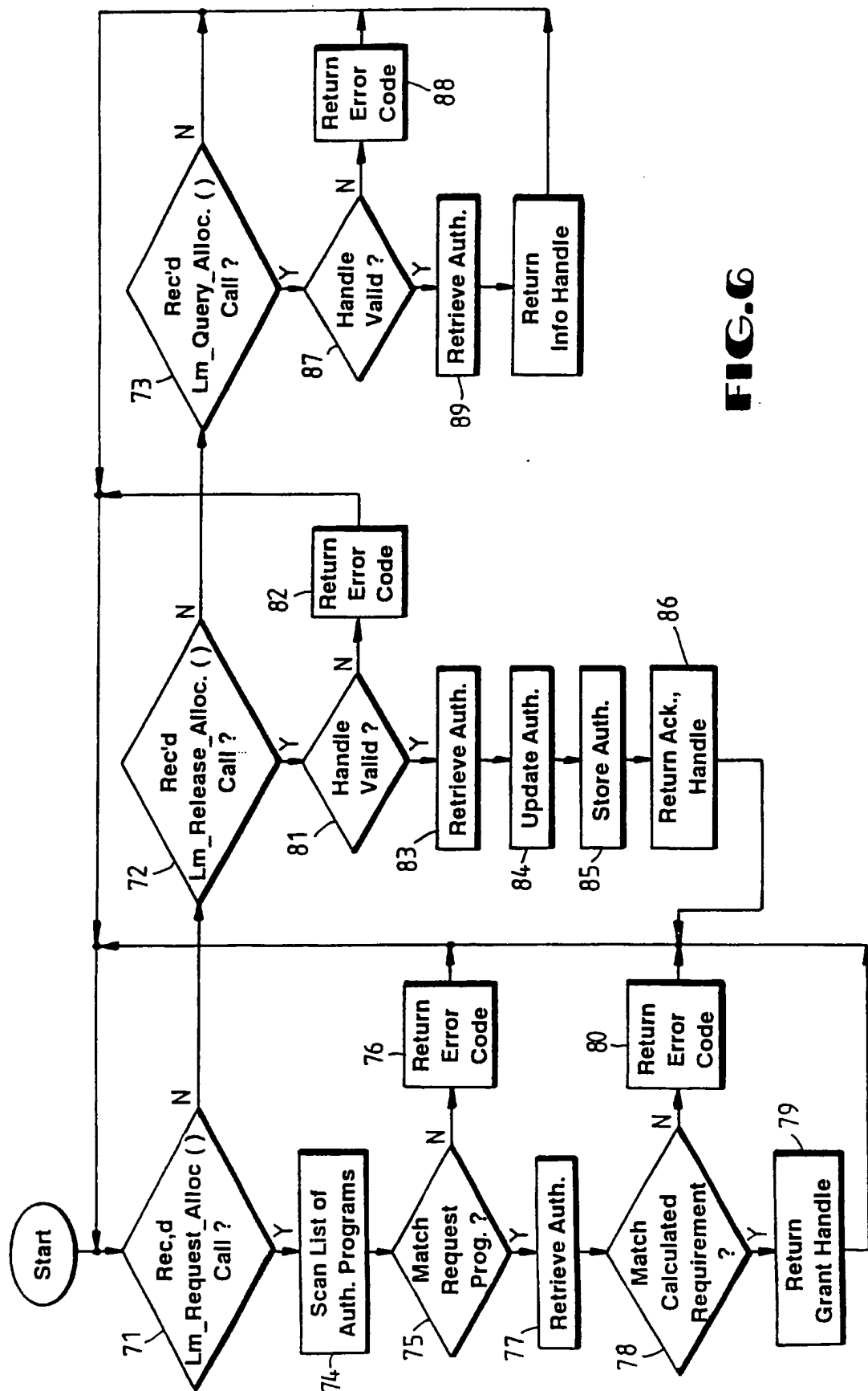
SUBSTITUTE SHEET

3/32

**FIG. 5**

SUBSTITUTE SHEET

4/32

**FIG. 6**

SUBSTITUTE SHEET

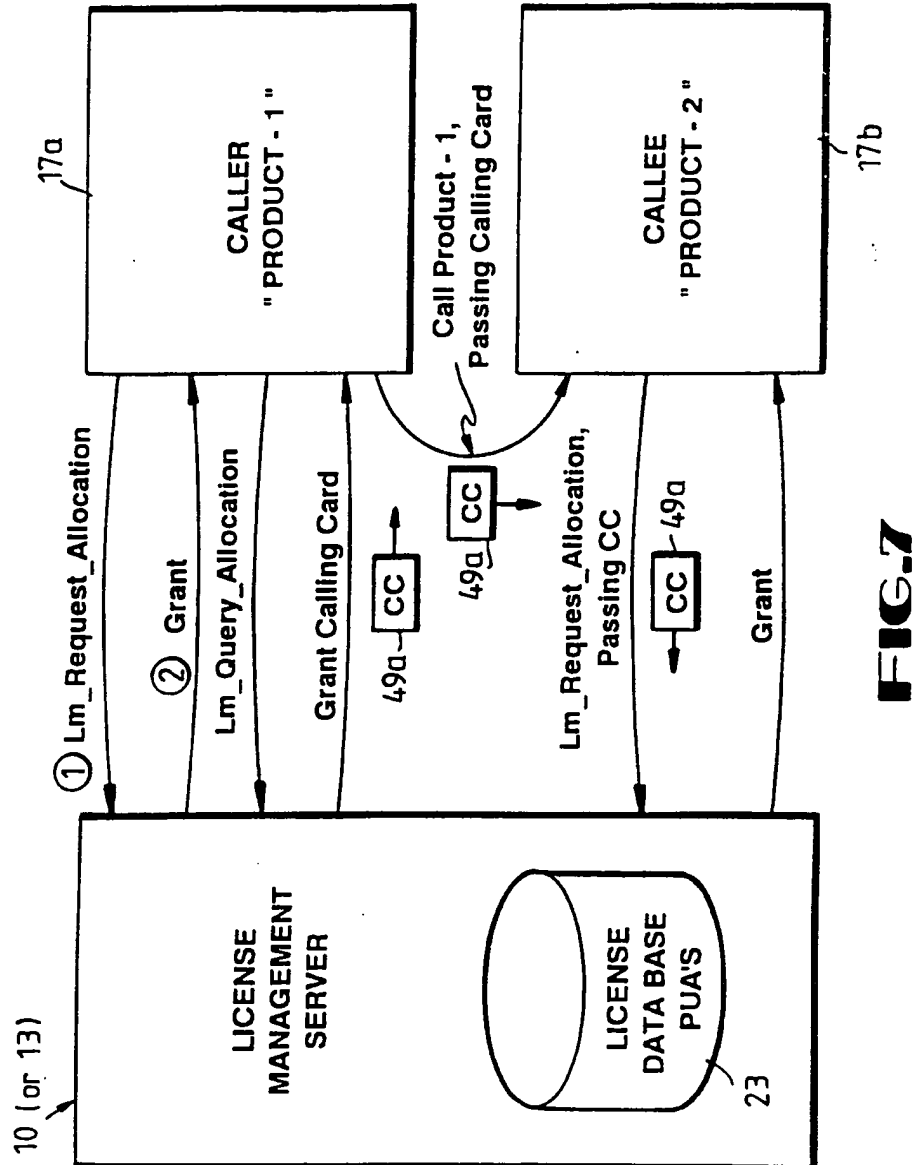


FIG.7

SUBSTITUTE SHEET

```

Object Identifier Value ::= {
    iso(1)
    identified-organization(3)
    icd-ecma(12)
    member-company(2)
    dec(1011)
    data-syntaxes(1)
    cda(3)
    ldif(17)
}

Object Identifier Encoding ::= {
    0x6, 0x8, 0x2B, 0xC, 0x2,
    0x87, 0x73, 0x1, 0x3, 0x11
}

```

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

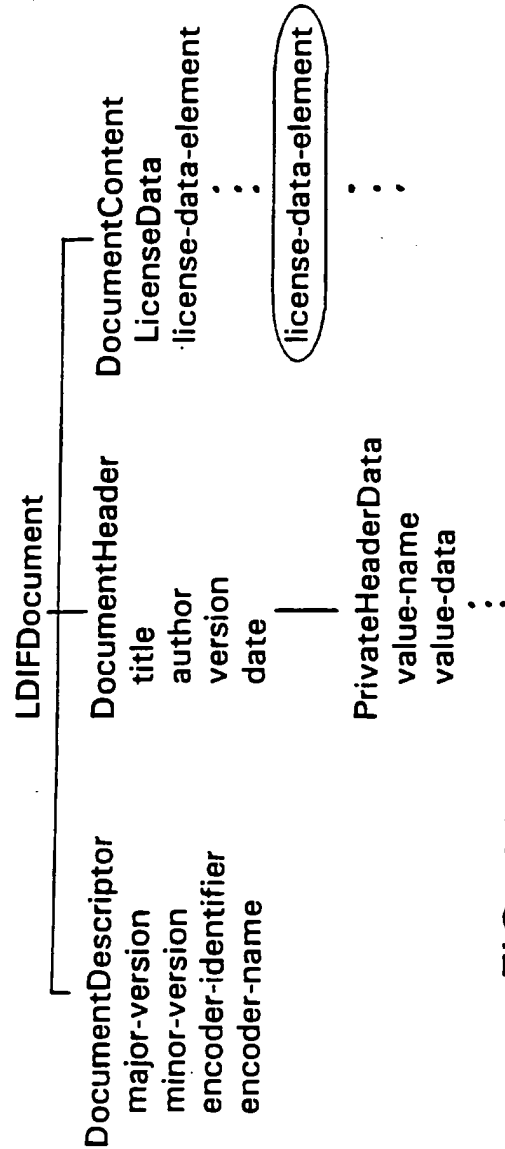


FIG. 10 LDIF Document Structure

```

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

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

11/32

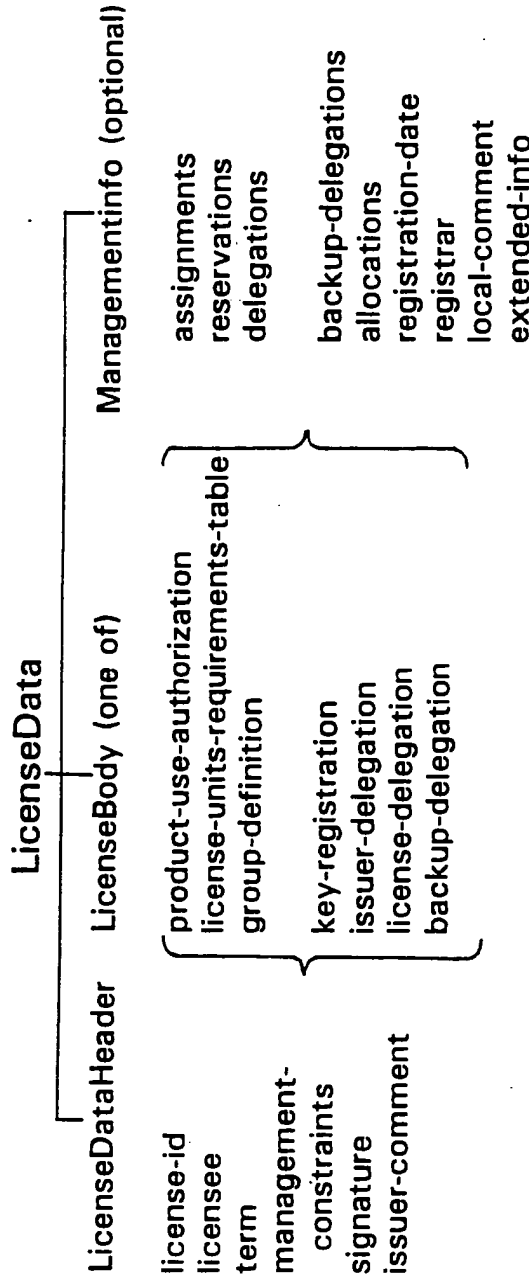


FIG. 16 License Data Structure

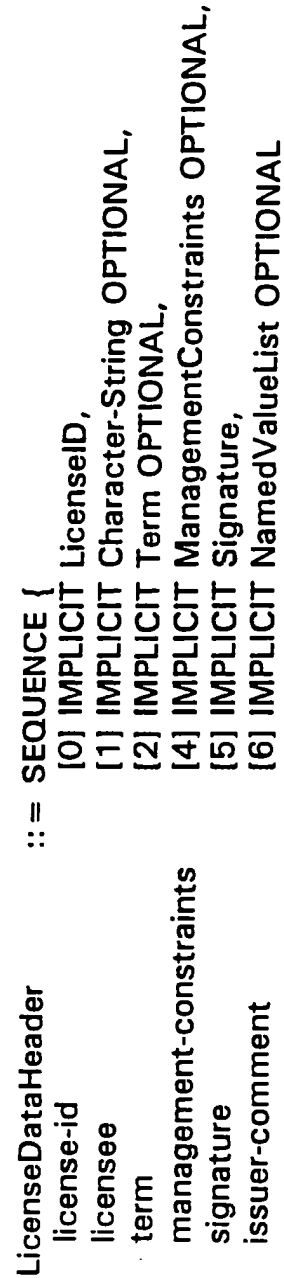


FIG. 17 License Data Header Syntax Diagram


```

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

```

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

14/32

```

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

SUBSTITUTE SHEET

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

18/32

```

ContextList ::= SEQUENCE OF Context
Context ::= SEQUENCE OF Subcontext
SubContext ::= SEQUENCE {
    sub-context-type [0] SubContextType,
    subcontext-value [1] ValueData
}
SubContextType ::= CHOICE {
    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.

20/32

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.

SUBSTITUTE SHEET

21/32

```

DistributionControl ::= SEQUENCE {
    distribution-method
        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

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

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

```

LURDM
    ::= SEQUENCE {
        combination-permitted [0] IMPLICIT BOOLEAN DEFAULT TRUE,
        overdraft-limit [1] IMPLICIT INTEGER DEFAULT 0,
        overdraft-logging-required [2] IMPLICIT BOOLEAN DEFAULT FALSE,
        allocation-size [3] IMPLICIT INTEGER OPTIONAL,
        lurdm-kind [4] IMPLICIT INTEGER {
            lurdt(1),
            constant(2),
            private-lurdm(3)
        },
        named-lurt-id [5] IMPLICIT Character-String OPTIONAL,
        lurdm-value [6] IMPLICIT INTEGER OPTIONAL,
        default-unit-requirement [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

26/32

```

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 ::= SEQUENCE {
 member-product [0] IMPLICIT ProductID,
 member-signature [1] IMPLICIT Signature,
 member-token [2] IMPLICIT NamedValueList OPTIONAL
}

FIG. 37 Member Syntax Diagram

NamedValue ::= SEQUENCE {
 value-name Character-String,
 value-data ValueData
}

ValueData ::= CHOICE {
 value-boolean [0] IMPLICIT BOOLEAN,
 value-integer [1] IMPLICIT INTEGER,
 value-text [2] IMPLICIT SEQUENCE OF Character-String
 value-general [3] IMPLICIT OCTET STRING,
 value-list [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	::= SEQUENCE {
producer	[0] IMPLICIT Character-String,
product-name	[1] IMPLICIT Character-String,
first-version	[2] IMPLICIT Version OPTIONAL,
last-version	[3] IMPLICIT Version OPTIONAL,
first-release-date	[4] IMPLICIT UTCTime OPTIONAL,
last-release-date	[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

```

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

INTERNATIONAL SEARCH REPORT

PCT/IS 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 --- -/-	
¹⁰ 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)		
Category °	Citation of Document, with indication, where appropriate, of the relevant passages	Relevant to Claim No.
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 21
Y A	see the whole document ---	

US 9203812
SA 60557

EPO FORM 10479

Petitioner Apple Inc. - Exhibit 1004, p. 876

**CORRECTED
VERSION***

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : G06F	A2	(11) International Publication Number: WO 96/27155 (43) International Publication Date: 6 September 1996 (06.09.96)
(21) International Application Number: PCT/US96/02303 (22) International Filing Date: 13 February 1996 (13.02.96) (30) Priority Data: 08/388,107 13 February 1995 (13.02.95) US (71) Applicant: ELECTRONIC PUBLISHING RESOURCES, INC. [US/US]; 5203 Battery Lane, Bethesda, MD 20814 (US). (72) Inventors: GINTER, Karl, L.; 10404 43rd Avenue, Beltsville, MD 20705 (US). SHEAR, Victor, H.; 5203 Battery Lane, Bethesda, MD 20814 (US). SPAHN, Francis, J.; 2410 Edwards Avenue, El Cerrito, CA 94530 (US). VAN WIE, David, M.; 1250 Lakeside Drive, Sunnyvale, CA 94086 (US). (74) Agent: FARIS, Robert, W.; Nixon & Vanderhye P.C., 1100 North Glebe Road, Arlington, VA 22201-4714 (US).		(81) Designated States: AL, AM, AT, AU, AZ, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), Eurasian patent (AZ, BY, KG, KZ, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG). Published <i>Without international search report and to be republished upon receipt of that report.</i>
(54) Title: SYSTEMS AND METHODS FOR SECURE TRANSACTION MANAGEMENT AND ELECTRONIC RIGHTS PROTECTION (57) Abstract The present invention provides systems and methods for electronic commerce including secure transaction management and electronic rights protection. Electronic appliances such as computers employed in accordance with the present invention help to ensure that information is accessed and used only in authorized ways, and maintain the integrity, availability, and/or confidentiality of the information. Secure subsystems used with such electronic appliances provide a distributed virtual distribution environment (VDE) that may enforce a secure chain of handling and control, for example, to control and/or meter or otherwise monitor use of electronically stored or disseminated information. Such a virtual distribution environment may be used to protect rights of various participants in electronic commerce and other electronic or electronic-facilitated transactions. Secure distributed and other operating system environments and architectures, employing, for example, secure semiconductor processing arrangements that may establish secure, protected environments at each node. These techniques may be used to support an end-to-end electronic information distribution capability that may be used, for example, utilizing the "electronic highway".		

* (Referred to in PCT Gazette No. 52/1996, Section II)

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.

AM	Armenia	GB	United Kingdom	MW	Malawi
AT	Austria	GE	Georgia	MX	Mexico
AU	Australia	GN	Guinea	NE	Niger
BB	Barbados	GR	Greece	NL	Netherlands
BE	Belgium	HU	Hungary	NO	Norway
BF	Burkina Faso	IE	Ireland	NZ	New Zealand
BG	Bulgaria	IT	Italy	PL	Poland
BJ	Benin	JP	Japan	PT	Portugal
BR	Brazil	KE	Kenya	RO	Romania
BY	Belarus	KG	Kyrgyzstan	RU	Russian Federation
CA	Canada	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic	KR	Republic of Korea	SE	Sweden
CG	Congo	KZ	Kazakhstan	SG	Singapore
CH	Switzerland	LI	Liechtenstein	SI	Slovenia
CI	Côte d'Ivoire	LK	Sri Lanka	SK	Slovakia
CM	Cameroon	LR	Liberia	SN	Senegal
CN	China	LT	Lithuania	SZ	Swaziland
CS	Czechoslovakia	LU	Luxembourg	TD	Chad
CZ	Czech Republic	LV	Latvia	TG	Togo
DE	Germany	MC	Monaco	TJ	Tajikistan
DK	Denmark	MD	Republic of Moldova	TT	Trinidad and Tobago
EE	Estonia	MG	Madagascar	UA	Ukraine
ES	Spain	ML	Mali	UG	Uganda
FI	Finland	MN	Mongolia	US	United States of America
FR	France	MR	Mauritania	UZ	Uzbekistan
GA	Gabon			VN	Viet Nam

**SYSTEMS AND METHODS FOR SECURE TRANSACTION
MANAGEMENT AND ELECTRONIC RIGHTS PROTECTION**

Field(s) of the Invention(s)

This invention generally relates to computer and/or
electronic security.

5

More particularly, this invention relates to systems and
techniques for secure transaction management. This invention
also relates to computer-based and other electronic appliance-
based technologies that help to ensure that information is
10 accessed and/or otherwise used only in authorized ways, and
maintains the integrity, availability, and/or confidentiality of
such information and processes related to such use.

The invention also relates to systems and methods for
15 protecting rights of various participants in electronic commerce
and other electronic or electronically-facilitated transactions.

The invention also relates to secure chains of handling and
control for both information content and information employed to
20 regulate the use of such content and consequences of such use. It
also relates to systems and techniques that manage, including
meter and/or limit and/or otherwise monitor use of electronically
stored and/or disseminated information. The invention

particularly relates to transactions, conduct and arrangements that make use of, including consequences of use of, such systems and/or techniques.

5 The invention also relates to distributed and other operating systems, environments and architectures. It also generally relates to secure architectures, including, for example, tamper-resistant hardware-based processors, that can be used to establish security at each node of a distributed system.

10

Background and Summary of the Invention(s)

Telecommunications, financial transactions, government processes, business operations, entertainment, and personal business productivity all now depend on electronic appliances. Millions of these electronic appliances have been electronically connected together. These interconnected electronic appliances comprise what is increasingly called the "information highway." Many businesses, academicians, and government leaders are concerned about how to protect the rights of citizens and organizations who use this information (also "electronic" or "digital") highway.

20

Electronic Content

Today, virtually anything that can be represented by words, numbers, graphics, or system of commands and

25

instructions can be formatted into electronic digital information.

Television, cable, satellite transmissions, and on-line services transmitted over telephone lines, compete to distribute digital information and entertainment to homes and businesses. The owners and marketers of this content include software developers, motion picture and recording companies, publishers of books, magazines, and newspapers, and information database providers. The popularization of on-line services has also enabled the individual personal computer user to participate as a content provider. It is estimated that the worldwide market for electronic information in 1992 was approximately \$40 billion and is expected to grow to \$200 billion by 1997, according to Microsoft Corporation. The present invention can materially enhance the revenue of content providers, lower the distribution costs and the costs for content, better support advertising and usage information gathering, and better satisfy the needs of electronic information users. These improvements can lead to a significant increase in the amount and variety of electronic information and the methods by which such information is distributed.

The inability of conventional products to be shaped to the needs of electronic information providers and users is sharply in contrast to the present invention. Despite the attention devoted by a cross-section of America's largest telecommunications, computer, entertainment and information provider companies to

some of the problems addressed by the present invention, only the present invention provides commercially secure, effective solutions for configurable, general purpose electronic commerce transaction/distribution control systems.

5

Controlling Electronic Content

The present invention provides a new kind of "virtual distribution environment" (called "VDE" in this document) that secures, administers, and audits electronic information use. VDE
10 also features fundamentally important capabilities for managing content that travels "across" the "information highway." These capabilities comprise a rights protection solution that serves all electronic community members. These members include content creators and distributors, financial service providers, end-users,
15 and others. VDE is the first general purpose, configurable, transaction control/rights protection solution for users of computers, other electronic appliances, networks, and the information highway.

20

A fundamental problem for electronic content providers is extending their ability to control the use of proprietary information. Content providers often need to limit use to authorized activities and amounts. Participants in a business
model involving, for example, provision of movies and advertising
25 on optical discs may include actors, directors, script and other

writers, musicians, studios, publishers, distributors, retailers, advertisers, credit card services, and content end-users. These participants need the ability to embody their range of agreements and requirements, including use limitations, into an "extended" agreement comprising an overall electronic business model. This extended agreement is represented by electronic content control information that can automatically enforce agreed upon rights and obligations. Under VDE, such an extended agreement may comprise an electronic contract involving all business model participants. Such an agreement may alternatively, or in addition, be made up of electronic agreements between subsets of the business model participants. Through the use of VDE, electronic commerce can function in the same way as traditional commerce—that is commercial relationships regarding products and services can be shaped through the negotiation of one or more agreements between a variety of parties.

Commercial content providers are concerned with ensuring proper compensation for the use of their electronic information. Electronic digital information, for example a CD recording, can today be copied relatively easily and inexpensively. Similarly, unauthorized copying and use of software programs deprives rightful owners of billions of dollars in annual revenue according to the International Intellectual Property Alliance. Content providers and distributors have devised a number of limited

function rights protection mechanisms to protect their rights.
Authorization passwords and protocols, license servers,
“lock/unlock” distribution methods, and non-electronic
contractual limitations imposed on users of shrink-wrapped
5 software are a few of the more prevalent content protection
schemes. In a commercial context, these efforts are inefficient
and limited solutions.

Providers of “electronic currency” have also created
10 protections for their type of content. These systems are not
sufficiently adaptable, efficient, nor flexible enough to support
the generalized use of electronic currency. Furthermore, they do
not provide sophisticated auditing and control configuration
capabilities. This means that current electronic currency tools
15 lack the sophistication needed for many real-world financial
business models. VDE provides means for anonymous currency
and for “conditionally” anonymous currency, wherein currency
related activities remain anonymous except under special
circumstances.

20

VDE Control Capabilities

VDE allows the owners and distributors of electronic
digital information to reliably bill for, and securely control, audit,
25 and budget the use of, electronic information. It can reliably

detect and monitor the use of commercial information products.

VDE uses a wide variety of different electronic information delivery means: including, for example, digital networks, digital broadcast, and physical storage media such as optical and magnetic disks. VDE can be used by major network providers, hardware manufacturers, owners of electronic information, providers of such information, and clearinghouses that gather usage information regarding, and bill for the use of, electronic information.

VDE provides comprehensive and configurable transaction management, metering and monitoring technology. It can change how electronic information products are protected, marketed, packaged, and distributed. When used, VDE should result in higher revenues for information providers and greater user satisfaction and value. Use of VDE will normally result in lower usage costs, decreased transaction costs, more efficient access to electronic information, re-usability of rights protection and other transaction management implementations, greatly improved flexibility in the use of secured information, and greater standardization of tools and processes for electronic transaction management. VDE can be used to create an adaptable environment that fulfills the needs of electronic information owners, distributors, and users; financial clearinghouses; and usage information analyzers and resellers.

Rights and Control Information

In general, the present invention can be used to protect the rights of parties who have:

- 5 (a) proprietary or confidentiality interests in electronic information. It can, for example, help ensure that information is used only in authorized ways;
- 10 (b) financial interests resulting from the use of electronically distributed information. It can help ensure that content providers will be paid for use of distributed information; and
- 15 (c) interests in electronic credit and electronic currency storage, communication, and/or use including electronic cash, banking, and purchasing.

Protecting the rights of electronic community members involves a broad range of technologies. VDE combines these
20 technologies in a way that creates a "distributed" electronic rights protection "environment." This environment secures and protects transactions and other processes important for rights protection. VDE, for example, provides the ability to prevent, or impede, interference with and/or observation of, important rights
25 related transactions and processes. VDE, in its preferred

embodiment, uses special purpose tamper resistant Secure Processing Units (SPUs) to help provide a high level of security for VDE processes and information storage and communication.

5 The rights protection problems solved by the present invention are electronic versions of basic societal issues. These issues include protecting property rights, protecting privacy rights, properly compensating people and organizations for their work and risk, protecting money and credit, and generally
10 protecting the security of information. VDE employs a system that uses a common set of processes to manage rights issues in an efficient, trusted, and cost-effective way.

 VDE can be used to protect the rights of parties who create
15 electronic content such as, for example: records, games, movies, newspapers, electronic books and reference materials, personal electronic mail, and confidential records and communications. The invention can also be used to protect the rights of parties who provide electronic products, such as publishers and
20 distributors; the rights of parties who provide electronic credit and currency to pay for use of products, for example, credit clearinghouses and banks; the rights to privacy of parties who use electronic content (such as consumers, business people, governments); and the privacy rights of parties described by
25 electronic information, such as privacy rights related to

information contained in a medical record, tax record, or personnel record.

In general, the present invention can protect the rights of parties who have:

(a) commercial interests in electronically distributed information -- the present invention can help ensure, for example, that parties, will be paid for use of distributed information in a manner consistent with their agreement;

(b) proprietary and/or confidentiality interests in electronic information -- the present invention can, for example, help ensure that data is used only in authorized ways;

(c) interests in electronic credit and electronic currency storage, communication, and/or use -- this can include electronic cash, banking, and purchasing; and

(d) interests in electronic information derived, at least in part, from use of other electronic information.

VDE Functional Properties

VDE is a cost-effective and efficient rights protection solution that provides a unified, consistent system for securing and managing transaction processing. VDE can:

- 5 (a) audit and analyze the use of content,
- (b) ensure that content is used only in authorized ways,
 and
- 10 (c) allow information regarding content usage to be used
 only in ways approved by content users.

In addition, VDE:

- 15 (a) is very configurable, modifiable, and re-usable;
- (b) supports a wide range of useful capabilities that may
 be combined in different ways to accommodate most
 potential applications;
- 20 (c) operates on a wide variety of electronic appliances
 ranging from hand-held inexpensive devices to large
 mainframe computers;

- (d) is able to ensure the various rights of a number of different parties, and a number of different rights protection schemes, simultaneously;
- 5 (e) is able to preserve the rights of parties through a series of transactions that may occur at different times and different locations;
- 10 (f) is able to flexibly accommodate different ways of securely delivering information and reporting usage; and
- 15 (g) provides for electronic analogues to "real" money and credit, including anonymous electronic cash, to pay for products and services and to support personal (including home) banking and other financial activities.

20 VDE economically and efficiently fulfills the rights protection needs of electronic community members. Users of VDE will not require additional rights protection systems for different information highway products and rights problems—nor will they be required to install and learn a new system for each new information highway application.

25

VDE provides a unified solution that allows all content creators, providers, and users to employ the same electronic rights protection solution. Under authorized circumstances, the participants can freely exchange content and associated content control sets. This means that a user of VDE may, if allowed, use the same electronic system to work with different kinds of content having different sets of content control information. The content and control information supplied by one group can be used by people who normally use content and control information supplied by a different group. VDE can allow content to be exchanged "universally" and users of an implementation of the present invention can interact electronically without fear of incompatibilities in content control, violation of rights, or the need to get, install, or learn a new content control system.

15

The VDE securely administers transactions that specify protection of rights. It can protect electronic rights including, for example:

- 20 (a) the property rights of authors of electronic content,
- (b) the commercial rights of distributors of content,
- (c) the rights of any parties who facilitated the
- 25 distribution of content,

- (d) the privacy rights of users of content,
- (e) the privacy rights of parties portrayed by stored
and/or distributed content, and
- (f) any other rights regarding enforcement of electronic
agreements.

5

VDE can enable a very broad variety of electronically enforced
commercial and societal agreements. These agreements can
include electronically implemented contracts, licenses, laws,
regulations, and tax collection.

10

Contrast With Traditional Solutions

Traditional content control mechanisms often require users
to purchase more electronic information than the user needs or
desires. For example, infrequent users of shrink-wrapped
software are required to purchase a program at the same price as
frequent users, even though they may receive much less value
from their less frequent use. Traditional systems do not scale
cost according to the extent or character of usage and traditional
systems can not attract potential customers who find that a fixed
price is too high. Systems using traditional mechanisms are also
not normally particularly secure. For example, shrink-wrapping

20

does not prevent the constant illegal pirating of software once removed from either its physical or electronic package.

Traditional electronic information rights protection
5 systems are often inflexible and inefficient and may cause a content provider to choose costly distribution channels that increase a product's price. In general these mechanisms restrict product pricing, configuration, and marketing flexibility. These compromises are the result of techniques for controlling
10 information which cannot accommodate both different content models and content models which reflect the many, varied requirements, such as content delivery strategies, of the model participants. This can limit a provider's ability to deliver sufficient overall value to justify a given product's cost in the eyes
15 of many potential users. VDE allows content providers and distributors to create applications and distribution networks that reflect content providers' and users' preferred business models. It offers users a uniquely cost effective and feature rich system that supports the ways providers want to distribute information
20 and the ways users want to use such information. VDE supports content control models that ensure rights and allow content delivery strategies to be shaped for maximum commercial results.

Chain of Handling and Control

VDE can protect a collection of rights belonging to various parties having in rights in, or to, electronic information. This information may be at one location or dispersed across (and/or moving between) multiple locations. The information may pass through a "chain" of distributors and a "chain" of users. Usage information may also be reported through one or more "chains" of parties. In general, VDE enables parties that (a) have rights in electronic information, and/or (b) act as direct or indirect agents for parties who have rights in electronic information, to ensure that the moving, accessing, modifying, or otherwise using of information can be securely controlled by rules regarding how, when, where, and by whom such activities can be performed.

VDE Applications and Software

VDE is a secure system for regulating electronic conduct and commerce. Regulation is ensured by control information put in place by one or more parties. These parties may include content providers, electronic hardware manufacturers, financial service providers, or electronic "infrastructure" companies such as cable or telecommunications companies. The control information implements "Rights Applications." Rights applications "run on" the "base software" of the preferred embodiment. This base software serves as a secure, flexible, general purpose foundation that can accommodate many

different rights applications, that is, many different business models and their respective participant requirements.

5 A rights application under VDE is made up of special purpose pieces, each of which can correspond to one or more basic electronic processes needed for a rights protection environment. These processes can be combined together like building blocks to create electronic agreements that can protect the rights, and may enforce fulfillment of the obligations, of electronic information
10 users and providers. One or more providers of electronic information can easily combine selected building blocks to create a rights application that is unique to a specific content distribution model. A group of these pieces can represent the capabilities needed to fulfill the agreement(s) between users and
15 providers. These pieces accommodate many requirements of electronic commerce including:

! the distribution of permissions to use electronic information;

20

! the persistence of the control information and sets of control information managing these permissions;

! configurable control set information that can be selected by users for use with such information;

25

! data security and usage auditing of electronic
information; and

! a secure system for currency, compensation and
5 debit management.

For electronic commerce, a rights application, under the
preferred embodiment of the present invention, can provide
electronic enforcement of the business agreements between all
10 participants. Since different groups of components can be put
together for different applications, the present invention can
provide electronic control information for a wide variety of
different products and markets. This means the present
invention can provide a "unified," efficient, secure, and
15 cost-effective system for electronic commerce and data security.
This allows VDE to serve as a single standard for electronic
rights protection, data security, and electronic currency and
banking.

20 In a VDE, the separation between a rights application and
its foundation permits the efficient selection of sets of control
information that are appropriate for each of many different types
of applications and uses. These control sets can reflect both
rights of electronic community members, as well as obligations
25 (such as providing a history of one's use of a product or paying

taxes on one's electronic purchases). VDE flexibility allows its users to electronically implement and enforce common social and commercial ethics and practices. By providing a unified control system, the present invention supports a vast range of possible transaction related interests and concerns of individuals, communities, businesses, and governments. Due to its open design, VDE allows (normally under securely controlled circumstances) applications using technology independently created by users to be "added" to the system and used in conjunction with the foundation of the invention. In sum, VDE provides a system that can fairly reflect and enforce agreements among parties. It is a broad ranging and systematic solution that answers the pressing need for a secure, cost-effective, and fair electronic environment.

15

VDE Implementation

The preferred embodiment of the present invention includes various tools that enable system designers to directly insert VDE capabilities into their products. These tools include an Application Programmer's Interface ("API") and a Rights Permissioning and Management Language ("RPML"). The RPML provides comprehensive and detailed control over the use of the invention's features. VDE also includes certain user interface subsystems for satisfying the needs of content providers, distributors, and users.

25

Information distributed using VDE may take many forms.

It may, for example, be "distributed" for use on an individual's own computer, that is the present invention can be used to provide security for locally stored data. Alternatively, VDE may
5 be used with information that is dispersed by authors and/or publishers to one or more recipients. This information may take many forms including: movies, audio recordings, games, electronic catalog shopping, multimedia, training materials, E-mail and personal documents, object oriented libraries,
10 software programming resources, and reference/record keeping information resources (such as business, medical, legal, scientific, governmental, and consumer databases).

Electronic rights protection provided by the present
15 invention will also provide an important foundation for trusted and efficient home and commercial banking, electronic credit processes, electronic purchasing, true or conditionally anonymous electronic cash, and EDI (Electronic Data Interchange). VDE provides important enhancements for improving data security in
20 organizations by providing "smart" transaction management features that can be far more effective than key and password based "go/no go" technology.

VDE normally employs an integration of cryptographic and
25 other security technologies (e.g. encryption, digital signatures,

etc.), with other technologies including: component, distributed, and event driven operating system technology, and related communications, object container, database, smart agent, smart card, and semiconductor design technologies.

5

I. Overview

A. VDE Solves Important Problems and Fills Critical Needs

The world is moving towards an integration of electronic information appliances. This interconnection of appliances provides a foundation for much greater electronic interaction and the evolution of electronic commerce. A variety of capabilities are required to implement an electronic commerce environment. VDE is the first system that provides many of these capabilities and therefore solves fundamental problems related to electronic dissemination of information.

15

Electronic Content

VDE allows electronic arrangements to be created involving two or more parties. These agreements can themselves comprise a collection of agreements between participants in a commercial value chain and/or a data security chain model for handling, auditing, reporting, and payment. It can provide efficient, reusable, modifiable, and consistent means for secure electronic content: distribution, usage control, usage payment,

20

25

usage auditing, and usage reporting. Content may, for example, include:

5 ! financial information such as electronic currency and credit;

 ! commercially distributed electronic information such as reference databases, movies, games, and advertising; and

10 ! electronic properties produced by persons and organizations, such as documents, e-mail, and proprietary database information.

15 VDE enables an electronic commerce marketplace that supports differing, competitive business partnerships, agreements, and evolving overall business models.

 The features of VDE allow it to function as the first trusted

20 electronic information control environment that can conform to, and support, the bulk of conventional electronic commerce and data security requirements. In particular, VDE enables the participants in a business value chain model to create an electronic version of traditional business agreement terms and

25 conditions and further enables these participants to shape and

evolve their electronic commerce models as they believe appropriate to their business requirements.

5 VDE offers an architecture that avoids reflecting specific distribution biases, administrative and control perspectives, and content types. Instead, VDE provides a broad-spectrum, fundamentally configurable and portable, electronic transaction control, distributing, usage, auditing, reporting, and payment operating environment. VDE is not limited to being an application or application specific toolset that covers only a limited subset of electronic interaction activities and participants. Rather, VDE supports systems by which such applications can be created, modified, and/or reused. As a result, the present invention answers pressing, unsolved needs by offering a system that supports a standardized control environment which facilitates interoperability of electronic appliances, interoperability of content containers, and efficient creation of electronic commerce applications and models through the use of a programmable, secure electronic transactions management foundation and reusable and extensible executable components. VDE can support a single electronic "world" within which most forms of electronic transaction activities can be managed.

10
15
20

To answer the developing needs of rights owners and content providers and to provide a system that can accommodate

25

the requirements and agreements of all parties that may be involved in electronic business models (creators, distributors, administrators, users, credit providers, etc.), VDE supplies an efficient, largely transparent, low cost and sufficiently secure
5 system (supporting both hardware/ software and software only models). VDE provides the widely varying secure control and administration capabilities required for:

1. Different types of electronic content,
10
2. Differing electronic content delivery schemes,
3. Differing electronic content usage schemes,
- 15 4. Different content usage platforms, and
5. Differing content marketing and model strategies.

VDE may be combined with, or integrated into, many
20 separate computers and/or other electronic appliances. These appliances typically include a secure subsystem that can enable control of content use such as displaying, encrypting, decrypting, printing, copying, saving, extracting, embedding, distributing, auditing usage, etc. The secure subsystem in the preferred
25 embodiment comprises one or more "protected processing

environments", one or more secure databases, and secure
"component assemblies" and other items and processes that need
to be kept secured. VDE can, for example, securely control
electronic currency, payments, and/or credit management
5 (including electronic credit and/or currency receipt,
disbursement, encumbering, and/or allocation) using such a
"secure subsystem."

VDE provides a secure, distributed electronic transaction
10 management system for controlling the distribution and/or other
usage of electronically provided and/or stored information. VDE
controls auditing and reporting of electronic content and/or
appliance usage. Users of VDE may include content creators who
apply content usage, usage reporting, and/or usage payment
15 related control information to electronic content and/or
appliances for users such as end-user organizations, individuals,
and content and/or appliance distributors. VDE also securely
supports the payment of money owed (including money owed for
content and/or appliance usage) by one or more parties to one or
20 more other parties, in the form of electronic credit and/or
currency.

Electronic appliances under control of VDE represent VDE
'nodes' that securely process and control; distributed electronic
25 information and/or appliance usage, control information

formulation, and related transactions. VDE can securely manage the integration of control information provided by two or more parties. As a result, VDE can construct an electronic agreement between VDE participants that represent a "negotiation" between, the control requirements of, two or more parties and enacts terms and conditions of a resulting agreement. VDE ensures the rights of each party to an electronic agreement regarding a wide range of electronic activities related to electronic information and/or appliance usage.

Through use of VDE's control system, traditional content providers and users can create electronic relationships that reflect traditional, non-electronic relationships. They can shape and modify commercial relationships to accommodate the evolving needs of, and agreements among, themselves. VDE does not require electronic content providers and users to modify their business practices and personal preferences to conform to a metering and control application program that supports limited, largely fixed functionality. Furthermore, VDE permits participants to develop business models not feasible with non-electronic commerce, for example, involving detailed reporting of content usage information, large numbers of distinct transactions at hitherto infeasibly low price points, "pass-along" control information that is enforced without involvement or advance knowledge of the participants, etc.

The present invention allows content providers and users
to formulate their transaction environment to accommodate:

- 5 (1) desired content models, content control models, and
content usage information pathways,
- (2) a complete range of electronic media and distribution
means,
- 10 (3) a broad range of pricing, payment, and auditing
strategies,
- (4) very flexible privacy and/or reporting models,
- 15 (5) practical and effective security architectures, and
- (6) other administrative procedures that together with
steps (1) through (5) can enable most "real world"
electronic commerce and data security models,
20 including models unique to the electronic world.

VDE's transaction management capabilities can enforce:

- (1) privacy rights of users related to information regarding their usage of electronic information and/or appliances,
- 5 (2) societal policy such as laws that protect rights of content users or require the collection of taxes derived from electronic transaction revenue, and
- 10 (3) the proprietary and/or other rights of parties related to ownership of, distribution of, and/or other commercial rights related to, electronic information.

VDE can support "real" commerce in an electronic form, that is the progressive creation of commercial relationships that form, over time, a network of interrelated agreements representing a value chain business model. This is achieved in part by enabling content control information to develop through the interaction of (negotiation between) securely created and independently submitted sets of content and/or appliance control information. Different sets of content and/or appliance control information can be submitted by different parties in an electronic business value chain enabled by the present invention. These parties create control information sets through the use of their respective VDE installations. Independently, securely deliverable, component based control information allows efficient

15

20

25

interaction among control information sets supplied by different parties.

5 VDE permits multiple, separate electronic arrangements to be formed between subsets of parties in a VDE supported electronic value chain model. These multiple agreements together comprise a VDE value chain "extended" agreement. VDE allows such constituent electronic agreements, and therefore overall VDE extended agreements, to evolve and
10 reshape over time as additional VDE participants become involved in VDE content and/or appliance control information handling. VDE electronic agreements may also be extended as new control information is submitted by existing participants. With VDE, electronic commerce participants are free to structure
15 and restructure their electronic commerce business activities and relationships. As a result, the present invention allows a competitive electronic commerce marketplace to develop since the use of VDE enables different, widely varying business models using the same or shared content.

20

A significant facet of the present invention's ability to broadly support electronic commerce is its ability to securely manage independently delivered VDE component objects containing control information (normally in the form of VDE
25 objects containing one or more methods, data, or load module

VDE components). This independently delivered control information can be integrated with senior and other pre-existing content control information to securely form derived control information using the negotiation mechanisms of the present invention. All requirements specified by this derived control information must be satisfied before VDE controlled content can be accessed or otherwise used. This means that, for example, all load modules and any mediating data which are listed by the derived control information as required must be available and securely perform their required function. In combination with other aspects of the present invention, securely, independently delivered control components allow electronic commerce participants to freely stipulate their business requirements and trade offs. As a result, much as with traditional, non-electronic commerce, the present invention allows electronic commerce (through a progressive stipulation of various control requirements by VDE participants) to evolve into forms of business that are the most efficient, competitive and useful.

VDE provides capabilities that rationalize the support of electronic commerce and electronic transaction management. This rationalization stems from the reusability of control structures and user interfaces for a wide variety of transaction management related activities. As a result, content usage control, data security, information auditing, and electronic

financial activities, can be supported with tools that are reusable, convenient, consistent, and familiar. In addition, a rational approach—a transaction/distribution control standard—allows all participants in VDE the same foundation set of hardware control and security, authoring, administration, and management tools to support widely varying types of information, business market model, and/or personal objectives.

Employing VDE as a general purpose electronic transaction/distribution control system allows users to maintain a single transaction management control arrangement on each of their computers, networks, communication nodes, and/or other electronic appliances. Such a general purpose system can serve the needs of many electronic transaction management applications without requiring distinct, different installations for different purposes. As a result, users of VDE can avoid the confusion and expense and other inefficiencies of different, limited purpose transaction control applications for each different content and/or business model. For example, VDE allows content creators to use the same VDE foundation control arrangement for both content authoring and for licensing content from other content creators for inclusion into their products or for other use. Clearinghouses, distributors, content creators, and other VDE users can all interact, both with the applications running on their VDE installations, and with each other, in an entirely consistent

manner, using and reusing (largely transparently) the same distributed tools, mechanisms, and consistent user interfaces, regardless of the type of VDE activity.

5 VDE prevents many forms of unauthorized use of electronic information, by controlling and auditing (and other administration of use) electronically stored and/or disseminated information. This includes, for example, commercially distributed content, electronic currency, electronic credit,
10 business transactions (such as EDI), confidential communications, and the like. VDE can further be used to enable commercially provided electronic content to be made available to users in user defined portions, rather than constraining the user to use portions of content that were "predetermined" by a content
15 creator and/or other provider for billing purposes.

VDE, for example, can employ:

- 20 (1) Secure metering means for budgeting and/or auditing electronic content and/or appliance usage;
- (2) Secure flexible means for enabling compensation and/or billing rates for content and/or appliance usage, including electronic credit and/or currency
25 mechanisms for payment means;

- 5
- (3) Secure distributed database means for storing control and usage related information (and employing validated compartmentalization and tagging schemes);
- 10
- (4) Secure electronic appliance control means;
- 15
- (5) A distributed, secure, "virtual black box" comprised of nodes located at every user (including VDE content container creators, other content providers, client users, and recipients of secure VDE content usage information) site. The nodes of said virtual black box normally include a secure subsystem having at least one secure hardware element (a semiconductor element or other hardware module for securely executing VDE control processes), said secure subsystems being distributed at nodes along a pathway of information storage, distribution, payment, usage, and/or auditing. In some
- 20
- embodiments, the functions of said hardware element, for certain or all nodes, may be performed by software, for example, in host processing environments of electronic appliances;
- 25
- (6) Encryption and decryption means;

- 5 (7) Secure communications means employing authentication, digital signaturing, and encrypted transmissions. The secure subsystems at said user nodes utilize a protocol that establishes and authenticates each node's and/or participant's identity, and establishes one or more secure host-to-host encryption keys for communications between the secure subsystems; and
- 10 (8) Secure control means that can allow each VDE installation to perform VDE content authoring (placing content into VDE containers with associated control information), content distribution, and content usage; as well as clearinghouse and other
- 15 administrative and analysis activities employing content usage information.

VDE may be used to migrate most non-electronic, traditional information delivery models (including entertainment, reference materials, catalog shopping, etc.) into an adequately

20 secure digital distribution and usage management and payment context. The distribution and financial pathways managed by a VDE arrangement may include:

25 ! content creator(s),

- ! distributor(s),
- ! redistributor(s),
- ! client administrator(s),
- ! client user(s),
- 5 ! financial and/or other clearinghouse(s),
- ! and/or government agencies.

These distribution and financial pathways may also include:

- 10 ! advertisers,
- ! market survey organizations, and/or
- ! other parties interested in the user usage of
information securely delivered and/or stored using
VDE.

15

Normally, participants in a VDE arrangement will employ the same secure VDE foundation. Alternate embodiments support VDE arrangements employing differing VDE foundations. Such alternate embodiments may employ procedures to ensure certain
20 interoperability requirements are met.

Secure VDE hardware (also known as SPUs for Secure Processing Units), or VDE installations that use software to substitute for, or complement, said hardware (provided by Host
25 Processing Environments (HPEs)), operate in conjunction with

secure communications, systems integration software, and distributed software control information and support structures, to achieve the electronic contract/rights protection environment of the present invention. Together, these VDE components
5 comprise a secure, virtual, distributed content and/or appliance control, auditing (and other administration), reporting, and payment environment. In some embodiments and where commercially acceptable, certain VDE participants, such as clearinghouses that normally maintain sufficiently physically
10 secure non-VDE processing environments, may be allowed to employ HPEs rather VDE hardware elements and interoperate, for example, with VDE end-users and content providers. VDE components together comprise a configurable, consistent, secure and "trusted" architecture for distributed, asynchronous control
15 of electronic content and/or appliance usage. VDE supports a "universe wide" environment for electronic content delivery, broad dissemination, usage reporting, and usage related payment activities.

20 VDE provides generalized configurability. This results, in part, from decomposition of generalized requirements for supporting electronic commerce and data security into a broad range of constituent "atomic" and higher level components (such as load modules, data elements, and methods) that may be
25 variously aggregated together to form control methods for

electronic commerce applications, commercial electronic agreements, and data security arrangements. VDE provides a secure operating environment employing VDE foundation elements along with secure independently deliverable VDE components that enable electronic commerce models and relationships to develop. VDE specifically supports the unfolding of distribution models in which content providers, over time, can expressly agree to, or allow, subsequent content providers and/or users to participate in shaping the control information for, and consequences of, use of electronic content and/or appliances. A very broad range of the functional attributes important for supporting simple to very complex electronic commerce and data security activities are supported by capabilities of the present invention. As a result, VDE supports most types of electronic information and/or appliance: usage control (including distribution), security, usage auditing, reporting, other administration, and payment arrangements.

VDE, in its preferred embodiment, employs object software technology and uses object technology to form "containers" for delivery of information that is (at least in part) encrypted or otherwise secured. These containers may contain electronic content products or other electronic information and some or all of their associated permissions (control) information. These container objects may be distributed along pathways involving

content providers and/or content users. They may be securely moved among nodes of a Virtual Distribution Environment (VDE) arrangement, which nodes operate VDE foundation software and execute control methods to enact electronic information usage control and/or administration models. The containers delivered through use of the preferred embodiment of the present invention may be employed both for distributing VDE control instructions (information) and/or to encapsulate and electronically distribute content that has been at least partially secured.

Content providers who employ the present invention may include, for example, software application and game publishers, database publishers, cable, television, and radio broadcasters, electronic shopping vendors, and distributors of information in electronic document, book, periodical, e-mail and/or other forms. Corporations, government agencies, and/or individual "end-users" who act as storers of, and/or distributors of, electronic information, may also be VDE content providers (in a restricted model, a user provides content only to himself and employs VDE to secure his own confidential information against unauthorized use by other parties). Electronic information may include proprietary and/or confidential information for personal or internal organization use, as well as information, such as software applications, documents, entertainment materials,

and/or reference information, which may be provided to other parties. Distribution may be by, for example, physical media delivery, broadcast and/or telecommunication means, and in the form of "static" files and/or streams of data. VDE may also be
5 used, for example, for multi-site "real-time" interaction such as teleconferencing, interactive games, or on-line bulletin boards, where restrictions on, and/or auditing of, the use of all or portions of communicated information is enforced.

10 VDE provides important mechanisms for both enforcing commercial agreements and enabling the protection of privacy rights. VDE can securely deliver information from one party to another concerning the use of commercially distributed electronic content. Even if parties are separated by several "steps" in a
15 chain (pathway) of handling for such content usage information, such information is protected by VDE through encryption and/or other secure processing. Because of that protection, the accuracy of such information is guaranteed by VDE, and the information can be trusted by all parties to whom it is delivered.

20 Furthermore, VDE guarantees that all parties can trust that such information cannot be received by anyone other than the intended, authorized, party(ies) because it is encrypted such that only an authorized party, or her agents, can decrypt it. Such
information may also be derived through a secure VDE process at
25 a previous pathway-of-handling location to produce secure VDE

reporting information that is then communicated securely to its intended recipient's VDE secure subsystem. Because VDE can deliver such information securely, parties to an electronic agreement need not trust the accuracy of commercial usage and/or other information delivered through means other than those under control of VDE.

VDE participants in a commercial value chain can be "commercially" confident (that is, sufficiently confident for commercial purposes) that the direct (constituent) and/or "extended" electronic agreements they entered into through the use of VDE can be enforced reliably. These agreements may have both "dynamic" transaction management related aspects, such as content usage control information enforced through budgeting, metering, and/or reporting of electronic information and/or appliance use, and/or they may include "static" electronic assertions, such as an end-user using the system to assert his or her agreement to pay for services, not to pass to unauthorized parties electronic information derived from usage of content or systems, and/or agreeing to observe copyright laws. Not only can electronically reported transaction related information be trusted under the present invention, but payment may be automated by the passing of payment tokens through a pathway of payment (which may or may not be the same as a pathway for reporting). Such payment can be contained within a VDE container created

automatically by a VDE installation in response to control information (located, in the preferred embodiment, in one or more permissions records) stipulating the "withdrawal" of credit or electronic currency (such as tokens) from an electronic account
5 (for example, an account securely maintained by a user's VDE installation secure subsystem) based upon usage of VDE controlled electronic content and/or appliances (such as governments, financial credit providers, and users).

10 VDE allows the needs of electronic commerce participants to be served and it can bind such participants together in a universe wide, trusted commercial network that can be secure enough to support very large amounts of commerce. VDE's security and metering secure subsystem core will be present at
15 all physical locations where VDE related content is (a) assigned usage related control information (rules and mediating data), and/or (b) used. This core can perform security and auditing functions (including metering) that operate within a "virtual black box," a collection of distributed, very secure VDE related
20 hardware instances that are interconnected by secured information exchange (for example, telecommunication) processes and distributed database means. VDE further includes highly configurable transaction operating system technology, one or more associated libraries of load modules along with affiliated
25 data, VDE related administration, data preparation, and analysis

applications, as well as system software designed to enable VDE integration into host environments and applications. VDE's usage control information, for example, provide for property content and/or appliance related: usage authorization, usage auditing (which may include audit reduction), usage billing, usage payment, privacy filtering, reporting, and security related communication and encryption techniques.

VDE extensively employs methods in the form of software objects to augment configurability, portability, and security of the VDE environment. It also employs a software object architecture for VDE content containers that carries protected content and may also carry both freely available information (e.g, summary, table of contents) and secured content control information which ensures the performance of control information. Content control information governs content usage according to criteria set by holders of rights to an object's contents and/or according to parties who otherwise have rights associated with distributing such content (such as governments, financial credit providers, and users).

In part, security is enhanced by object methods employed by the present invention because the encryption schemes used to protect an object can efficiently be further used to protect the associated content control information (software control

information and relevant data) from modification. Said object techniques also enhance portability between various computer and/or other appliance environments because electronic information in the form of content can be inserted along with (for
5 example, in the same object container as) content control information (for said content) to produce a "published" object. As a result, various portions of said control information may be specifically adapted for different environments, such as for diverse computer platforms and operating systems, and said
10 various portions may all be carried by a VDE container.

An objective of VDE is supporting a transaction/distribution control standard. Development of such a standard has many obstacles, given the security requirements
15 and related hardware and communications issues, widely differing environments, information types, types of information usage, business and/or data security goals, varieties of participants, and properties of delivered information. A significant feature of VDE accommodates the many, varying
20 distribution and other transaction variables by, in part, decomposing electronic commerce and data security functions into generalized capability modules executable within a secure hardware SPU and/or corresponding software subsystem and further allowing extensive flexibility in assembling, modifying,
25 and/or replacing, such modules (e.g. load modules and/or

methods) in applications run on a VDE installation foundation. This configurability and reconfigurability allows electronic commerce and data security participants to reflect their priorities and requirements through a process of iteratively shaping an evolving extended electronic agreement (electronic control model). This shaping can occur as content control information passes from one VDE participant to another and to the extent allowed by "in place" content control information. This process allows users of VDE to recast existing control information and/or add new control information as necessary (including the elimination of no longer required elements).

VDE supports trusted (sufficiently secure) electronic information distribution and usage control models for both commercial electronic content distribution and data security applications. It can be configured to meet the diverse requirements of a network of interrelated participants that may include content creators, content distributors, client administrators, end users, and/or clearinghouses and/or other content usage information users. These parties may constitute a network of participants involved in simple to complex electronic content dissemination, usage control, usage reporting, and/or usage payment. Disseminated content may include both originally provided and VDE generated information (such as content usage information) and content control information may

persist through both chains (one or more pathways) of content and content control information handling, as well as the direct usage of content. The configurability provided by the present invention is particularly critical for supporting electronic commerce, that is enabling businesses to create relationships and evolve strategies that offer competitive value. Electronic commerce tools that are not inherently configurable and interoperable will ultimately fail to produce products (and services) that meet both basic requirements and evolving needs of most commerce applications.

VDE's fundamental configurability will allow a broad range of competitive electronic commerce business models to flourish. It allows business models to be shaped to maximize revenues sources, end-user product value, and operating efficiencies. VDE can be employed to support multiple, differing models, take advantage of new revenue opportunities, and deliver product configurations most desired by users. Electronic commerce technologies that do not, as the present invention does:

- ! support a broad range of possible, complementary revenue activities,
- ! offer a flexible array of content usage features most desired by customers, and
- ! exploit opportunities for operating efficiencies,

will result in products that are often intrinsically more costly and less appealing and therefore less competitive in the marketplace.

5 Some of the key factors contributing to the configurability intrinsic to the present invention include:

- 10 (a) integration into the fundamental control environment of a broad range of electronic appliances through portable API and programming language tools that efficiently support merging of control and auditing capabilities in nearly any electronic appliance environment while maintaining overall system security;
- 15 (b) modular data structures;
- (c) generic content model;
- 20 (d) general modularity and independence of foundation architectural components;
- (e) modular security structures;
- 25 (f) variable length and multiple branching chains of control; and

(g) independent, modular control structures in the form of executable load modules that can be maintained in one or more libraries, and assembled into control methods and models, and where such model control schemes can "evolve" as control information passes through the VDE installations of participants of a pathway of VDE content control information handling.

10 Because of the breadth of issues resolved by the present invention, it can provide the emerging "electronic highway" with a single transaction/distribution control system that can, for a very broad range of commercial and data security models, ensure against unauthorized use of confidential and/or proprietary
15 information and commercial electronic transactions. VDE's electronic transaction management mechanisms can enforce the electronic rights and agreements of all parties participating in widely varying business and data security models, and this can be efficiently achieved through a single VDE implementation
20 within each VDE participant's electronic appliance. VDE supports widely varying business and/or data security models that can involve a broad range of participants at various "levels" of VDE content and/or content control information pathways of handling. Different content control and/or auditing models and
25 agreements may be available on the same VDE installation.

These models and agreements may control content in relationship to, for example, VDE installations and/or users in general; certain specific users, installations, classes and/or other groupings of installations and/or users; as well as to electronic content generally on a given installation, to specific properties, property portions, classes and/or other groupings of content.

Distribution using VDE may package both the electronic content and control information into the same VDE container, and/or may involve the delivery to an end-user site of different pieces of the same VDE managed property from plural separate remote locations and/or in plural separate VDE content containers and/or employing plural different delivery means. Content control information may be partially or fully delivered separately from its associated content to a user VDE installation in one or more VDE administrative objects. Portions of said control information may be delivered from one or more sources. Control information may also be available for use by access from a user's VDE installation secure sub-system to one or more remote VDE secure sub-systems and/or VDE compatible, certified secure remote locations. VDE control processes such as metering, budgeting, decrypting and/or fingerprinting, may as relates to a certain user content usage activity, be performed in a user's local VDE installation secure subsystem, or said processes may be divided amongst plural secure subsystems which may be

located in the same user VDE installations and/or in a network server and in the user installation. For example, a local VDE installation may perform decryption and save any, or all of, usage metering information related to content and/or electronic
5 appliance usage at such user installation could be performed at the server employing secure (e.g., encrypted) communications between said secure subsystems. Said server location may also be used for near real time, frequent, or more periodic secure receipt of content usage information from said user installation,
10 with, for example, metered information being maintained only temporarily at a local user installation.

Delivery means for VDE managed content may include electronic data storage means such as optical disks for delivering
15 one portion of said information and broadcasting and/or telecommunicating means for other portions of said information. Electronic data storage means may include magnetic media, optical media, combined magneto-optical systems, flash RAM memory, bubble memory, and/or other memory storage means
20 such as huge capacity optical storage systems employing holographic, frequency, and/or polarity data storage techniques. Data storage means may also employ layered disc techniques, such as the use of generally transparent and/or translucent materials that pass light through layers of data carrying discs
25 which themselves are physically packaged together as one

thicker disc. Data carrying locations on such discs may be, at least in part, opaque.

5 VDE supports a general purpose foundation for secure transaction management, including usage control, auditing, reporting, and/or payment. This general purpose foundation is called "VDE Functions" ("VDEFs"). VDE also supports a collection of "atomic" application elements (e.g., load modules) that can be selectively aggregated together to form various VDEF capabilities called control methods and which serve as VDEF applications and operating system functions. When a host
10 operating environment of an electronic appliance includes VDEF capabilities, it is called a "Rights Operating System" (ROS). VDEF load modules, associated data, and methods form a body of information that for the purposes of the present invention are
15 called "control information." VDEF control information may be specifically associated with one or more pieces of electronic content and/or it may be employed as a general component of the operating system capabilities of a VDE installation.

20

VDEF transaction control elements reflect and enact content specific and/or more generalized administrative (for example, general operating system) control information. VDEF capabilities which can generally take the form of applications
25 (application models) that have more or less configurability which

can be shaped by VDE participants, through the use, for example, of VDE templates, to employ specific capabilities, along, for example, with capability parameter data to reflect the elements of one or more express electronic agreements between VDE participants in regards to the use of electronic content such as commercially distributed products. These control capabilities manage the use of, and/or auditing of use of, electronic content, as well as reporting information based upon content use, and any payment for said use. VDEF capabilities may "evolve" to reflect the requirements of one or more successive parties who receive or otherwise contribute to a given set of control information. Frequently, for a VDE application for a given content model (such as distribution of entertainment on CD-ROM, content delivery from an Internet repository, or electronic catalog shopping and advertising, or some combination of the above) participants would be able to securely select from amongst available, alternative control methods and apply related parameter data, wherein such selection of control method and/or submission of data would constitute their "contribution" of control information. Alternatively, or in addition, certain control methods that have been expressly certified as securely interoperable and compatible with said application may be independently submitted by a participant as part of such a contribution. In the most general example, a generally certified load module (certified for a given VDE arrangement and/or content class) may be used with many

or any VDE application that operates in nodes of said arrangement. These parties, to the extent they are allowed, can independently and securely add, delete, and/or otherwise modify the specification of load modules and methods, as well as add,
5 delete or otherwise modify related information.

Normally the party who creates a VDE content container defines the general nature of the VDEF capabilities that will and/or may apply to certain electronic information. A VDE
10 content container is an object that contains both content (for example, commercially distributed electronic information products such as computer software programs, movies, electronic publications or reference materials, etc.) and certain control information related to the use of the object's content. A creating
15 party may make a VDE container available to other parties. Control information delivered by, and/or otherwise available for use with, VDE content containers comprise (for commercial content distribution purposes) VDEF control capabilities (and any associated parameter data) for electronic content. These
20 capabilities may constitute one or more "proposed" electronic agreements (and/or agreement functions available for selection and/or use with parameter data) that manage the use and/or the consequences of use of such content and which can enact the terms and conditions of agreements involving multiple parties
25 and their various rights and obligations.

A VDE electronic agreement may be explicit, through a user interface acceptance by one or more parties, for example by a "junior" party who has received control information from a "senior" party, or it may be a process amongst equal parties who individually assert their agreement. Agreement may also result from an automated electronic process during which terms and conditions are "evaluated" by certain VDE participant control information that assesses whether certain other electronic terms and conditions attached to content and/or submitted by another party are acceptable (do not violate acceptable control information criteria). Such an evaluation process may be quite simple, for example a comparison to ensure compatibility between a portion of, or all senior, control terms and conditions in a table of terms and conditions and the submitted control information of a subsequent participant in a pathway of content control information handling, or it may be a more elaborate process that evaluates the potential outcome of, and/or implements a negotiation process between, two or more sets of control information submitted by two or more parties. VDE also accommodates a semi-automated process during which one or more VDE participants directly, through user interface means, resolve "disagreements" between control information sets by accepting and/or proposing certain control information that may be acceptable to control information representing one or more other parties interests and/or responds to certain user interface

queries for selection of certain alternative choices and/or for certain parameter information, the responses being adopted if acceptable to applicable senior control information.

5 When another party (other than the first applier of rules), perhaps through a negotiation process, accepts, and/or adds to and/or otherwise modifies, "in place" content control information, a VDE agreement between two or more parties related to the use of such electronic content may be created (so long as any
10 modifications are consistent with senior control information). Acceptance of terms and conditions related to certain electronic content may be direct and express, or it may be implicit as a result of use of content (depending, for example, on legal requirements, previous exposure to such terms and conditions,
15 and requirements of in place control information).

 VDEF capabilities may be employed, and a VDE agreement may be entered into, by a plurality of parties without the VDEF capabilities being directly associated with the
20 controlling of certain, specific electronic information. For example, certain one or more VDEF capabilities may be present at a VDE installation, and certain VDE agreements may have been entered into during the registration process for a content distribution application, to be used by such installation for
25 securely controlling VDE content usage, auditing, reporting

and/or payment. Similarly, a specific VDE participant may enter into a VDE user agreement with a VDE content or electronic appliance provider when the user and/or her appliance register with such provider as a VDE installation and/or user. In such events, VDEF in place control information available to the user VDE installation may require that certain VDEF methods are employed, for example in a certain sequence, in order to be able to use all and/or certain classes, of electronic content and/or VDE applications.

10

VDE ensures that certain prerequisites necessary for a given transaction to occur are met. This includes the secure execution of any required load modules and the availability of any required, associated data. For example, required load modules and data (e.g. in the form of a method) might specify that sufficient credit from an authorized source must be confirmed as available. It might further require certain one or more load modules execute as processes at an appropriate time to ensure that such credit will be used in order to pay for user use of the content. A certain content provider might, for example, require metering the number of copies made for distribution to employees of a given software program (a portion of the program might be maintained in encrypted form and require the presence of a VDE installation to run). This would require the execution of a metering method for copying of the property each time a copy

25

was made for another employee. This same provider might also charge fees based on the total number of different properties licensed from them by the user and a metering history of their licensing of properties might be required to maintain this
5 information.

VDE provides organization, community, and/or universe wide secure environments whose integrity is assured by processes securely controlled in VDE participant user
10 installations (nodes). VDE installations, in the preferred embodiment, may include both software and tamper resistant hardware semiconductor elements. Such a semiconductor arrangement comprises, at least in part, special purpose circuitry that has been designed to protect against tampering with, or
15 unauthorized observation of, the information and functions used in performing the VDE's control functions. The special purpose secure circuitry provided by the present invention includes at least one of: a dedicated semiconductor arrangement known as a Secure Processing Unit (SPU) and/or a standard microprocessor,
20 microcontroller, and/or other processing logic that accommodates the requirements of the present invention and functions as an SPU. VDE's secure hardware may be found incorporated into, for example, a fax/modem chip or chip pack, I/O controller, video display controller, and/or other available digital processing
25 arrangements. It is anticipated that portions of the present

invention's VDE secure hardware capabilities may ultimately be standard design elements of central processing units (CPUs) for computers and various other electronic devices.

5 Designing VDE capabilities into one or more standard microprocessor, microcontroller and/or other digital processing components may materially reduce VDE related hardware costs by employing the same hardware resources for both the transaction management uses contemplated by the present
10 invention and for other, host electronic appliance functions. This means that a VDE SPU can employ (share) circuitry elements of a "standard" CPU. For example, if a "standard" processor can operate in protected mode and can execute VDE related instructions as a protected activity, then such an embodiment
15 may provide sufficient hardware security for a variety of applications and the expense of a special purpose processor might be avoided. Under one preferred embodiment of the present invention, certain memory (e.g., RAM, ROM, NVRAM) is maintained during VDE related instruction processing in a
20 protected mode (for example, as supported by protected mode microprocessors). This memory is located in the same package as the processing logic (e.g. processor). Desirably, the packaging and memory of such a processor would be designed using security techniques that enhance its resistance to tampering.

25

The degree of overall security of the VDE system is primarily dependent on the degree of tamper resistance and concealment of VDE control process execution and related data storage activities. Employing special purpose semiconductor packaging techniques can significantly contribute to the degree of security. Concealment and tamper-resistance in semiconductor memory (e.g., RAM, ROM, NVRAM) can be achieved, in part, by employing such memory within an SPU package, by encrypting data before it is sent to external memory (such as an external RAM package) and decrypting encrypted data within the CPU/RAM package before it is executed. This process is used for important VDE related data when such data is stored on unprotected media, for example, standard host storage, such as random access memory, mass storage, etc. In that event, a VDE SPU would encrypt data that results from a secure VDE execution before such data was stored in external memory.

Summary of Some Important Features Provided by VDE in Accordance With the Present Invention

VDE employs a variety of capabilities that serve as a foundation for a general purpose, sufficiently secure distributed electronic commerce solution. VDE enables an electronic commerce marketplace that supports divergent, competitive business partnerships, agreements, and evolving overall business models. For example, VDE includes features that:

“sufficiently” impede unauthorized and/or uncompensated use of electronic information and/or appliances through the use of secure communication, storage, and transaction management technologies.

5 VDE supports a model wide, distributed security implementation which creates a single secure “virtual” transaction processing and information storage environment. VDE enables distributed VDE installations to securely store and communicate

10 information and remotely control the execution processes and the character of use of electronic information at other VDE installations and in a wide variety of ways;

15 ! support low-cost, efficient, and effective security architectures for transaction control, auditing, reporting, and related communications and information storage. VDE may employ tagging related security techniques, the time-ageing of

20 encryption keys, the compartmentalization of both stored control information (including differentially tagging such stored information to ensure against substitution and tampering) and distributed content (to, for many content applications, employ one or

25 more content encryption keys that are unique to the

specific VDE installation and/or user), private key techniques such as triple DES to encrypt content, public key techniques such as RSA to protect communications and to provide the benefits of digital signature and authentication to securely bind together the nodes of a VDE arrangement, secure processing of important transaction management executable code, and a combining of a small amount of highly secure, hardware protected storage space with a much larger "exposed" mass media storage space storing secured (normally encrypted and tagged) control and audit information. VDE employs special purpose hardware distributed throughout some or all locations of a VDE implementation: a) said hardware controlling important elements of: content preparation (such as causing such content to be placed in a VDE content container and associating content control information with said content), content and/or electronic appliance usage auditing, content usage analysis, as well as content usage control; and b) said hardware having been designed to securely handle processing load module control activities, wherein said control processing activities may involve a sequence of required control factors;

! support dynamic user selection of information
subsets of a VDE electronic information product
(VDE controlled content). This contrasts with the
constraints of having to use a few high level
5 individual, pre-defined content provider information
increments such as being required to select a whole
information product or product section in order to
acquire or otherwise use a portion of such product or
section. VDE supports metering and usage control
10 over a variety of increments (including "atomic"
increments, and combinations of different increment
types) that are selected ad hoc by a user and
represent a collection of pre-identified one or more
increments (such as one or more blocks of a
15 preidentified nature, e.g., bytes, images, logically
related blocks) that form a generally arbitrary, but
logical to a user, content "deliverable." VDE control
information (including budgeting, pricing and
metering) can be configured so that it can specifically
20 apply, as appropriate, to ad hoc selection of different,
unanticipated variable user selected aggregations of
information increments and pricing levels can be, at
least in part, based on quantities and/or nature of
mixed increment selections (for example, a certain
25 quantity of certain text could mean associated

images might be discounted by 15%; a greater quantity of text in the "mixed" increment selection might mean the images are discounted 20%). Such user selected aggregated information increments can reflect the actual requirements of a user for information and is more flexible than being limited to a single, or a few, high level, (e.g. product, document, database record) predetermined increments. Such high level increments may include quantities of information not desired by the user and as a result be more costly than the subset of information needed by the user if such a subset was available. In sum, the present invention allows information contained in electronic information products to be supplied according to user specification. Tailoring to user specification allows the present invention to provide the greatest value to users, which in turn will generate the greatest amount of electronic commerce activity. The user, for example, would be able to define an aggregation of content derived from various portions of an available content product, but which, as a deliverable for use by the user, is an entirely unique aggregated increment. The user may, for example, select certain numbers of bytes of information from

various portions of an information product, such as a reference work, and copy them to disc in unencrypted form and be billed based on total number of bytes plus a surcharge on the number of "articles" that provided the bytes. A content provider might reasonably charge less for such a user defined information increment since the user does not require all of the content from all of the articles that contained desired information. This process of defining a user desired information increment may involve artificial intelligence database search tools that contribute to the location of the most relevant portions of information from an information product and cause the automatic display to the user of information describing search criteria hits for user selection or the automatic extraction and delivery of such portions to the user. VDE further supports a wide variety of predefined increment types including:

- ! bytes,
- ! images,
- ! content over time for audio or video, or any other increment that can be identified by content provider data mapping efforts, such as:
- ! sentences,

! paragraphs,
! articles,
! database records, and
! byte offsets representing increments of
5 logically related information.

VDE supports as many simultaneous predefined increment types
as may be practical for a given type of content and business
model.

10 ! securely store at a user's site potentially highly
detailed information reflective of a user's usage of a
variety of different content segment types and
employing both inexpensive "exposed" host mass
storage for maintaining detailed information in the
15 form of encrypted data and maintaining summary
information for security testing in highly secure
special purpose VDE installation nonvolatile
memory (if available).

20 ! support trusted chain of handling capabilities for
pathways of distributed electronic information
and/or for content usage related information. Such
chains may extend, for example, from a content
creator, to a distributor, a redistributor, a client
25 user, and then may provide a pathway for securely

reporting the same and/or differing usage
information to one or more auditors, such as to one
or more independent clearinghouses and then back
to the content providers, including content creators.
5 The same and/or different pathways employed for
certain content handling, and related content control
information and reporting information handling,
may also be employed as one or more pathways for
electronic payment handling (payment is
10 characterized in the present invention as
administrative content) for electronic content and/or
appliance usage. These pathways are used for
conveyance of all or portions of content, and/or
content related control information. Content
15 creators and other providers can specify the
pathways that, partially or fully, must be used to
disseminate commercially distributed property
content, content control information, payment
administrative content, and/or associated usage
20 reporting information. Control information specified
by content providers may also specify which specific
parties must or may (including, for example, a group
of eligible parties from which a selection may be
made) handle conveyed information. It may also
25 specify what transmission means (for example

telecommunication carriers or media types) and
transmission hubs must or may be used.

5 support flexible auditing mechanisms, such as
employing "bitmap meters," that achieve a high
degree of efficiency of operation and throughput and
allow, in a practical manner, the retention and ready
recall of information related to previous usage
activities and related patterns. This flexibility is
10 adaptable to a wide variety of billing and security
control strategies such as:

- P upgrade pricing (e.g. suite purchases),
- P pricing discounts (including quantity
discounts),
- 15 P billing related time duration variables such as
discounting new purchases based on the
timing of past purchases, and
- P security budgets based on quantity of
different, logically related units of electronic
20 information used over an interval of time.

Use of bitmap meters (including "regular" and "wide"
bitmap meters) to record usage and/or purchase of
information, in conjunction with other elements of
25 the preferred embodiment of the present invention,

uniquely supports efficient maintenance of usage history for: (a) rental, (b) flat fee licensing or purchase, (c) licensing or purchase discounts based upon historical usage variables, and (d) reporting to users in a manner enabling users to determine whether a certain item was acquired, or acquired within a certain time period (without requiring the use of conventional database mechanisms, which are highly inefficient for these applications). Bitmap meter methods record activities associated with electronic appliances, properties, objects, or portions thereof, and/or administrative activities that are independent of specific properties, objects, etc., performed by a user and/or electronic appliance such that a content and/or appliance provider and/or controller of an administrative activity can determine whether a certain activity has occurred at some point, or during a certain period, in the past (for example, certain use of a commercial electronic content product and/or appliance). Such determinations can then be used as part of pricing and/or control strategies of a content and/or appliance provider, and/or controller of an administrative activity. For example, the content provider may choose to charge only once for access to

a portion of a property, regardless of the number of times that portion of the property is accessed by a user.

5 ! support "launchable" content, that is content that
can be provided by a content provider to an end-user,
who can then copy or pass along the content to other
end-user parties without requiring the direct
participation of a content provider to register and/or
10 otherwise initialize the content for use. This content
goes "out of (the traditional distribution) channel" in
the form of a "traveling object." Traveling objects are
containers that securely carry at least some
permissions information and/or methods that are
15 required for their use (such methods need not be
carried by traveling objects if the required methods
will be available at, or directly available to, a
destination VDE installation). Certain travelling
objects may be used at some or all VDE installations
20 of a given VDE arrangement since they can make
available the content control information necessary
for content use without requiring the involvement of
a commercial VDE value chain participant or data
security administrator (e.g. a control officer or
25 network administrator). As long as traveling object

control information requirements are available at the user VDE installation secure subsystem (such as the presence of a sufficient quantity of financial credit from an authorized credit provider), at least
5 some travelling object content may be used by a receiving party without the need to establish a connection with a remote VDE authority (until, for example, budgets are exhausted or a time content usage reporting interval has occurred). Traveling
10 objects can travel "out-of-channel," allowing, for example, a user to give a copy of a traveling object whose content is a software program, a movie or a game, to a neighbor, the neighbor being able to use the traveling object if appropriate credit (e.g. an
15 electronic clearinghouse account from a clearinghouse such as VISA or AT&T) is available. Similarly, electronic information that is generally available on an Internet, or a similar network, repository might be provided in the form of a
20 traveling object that can be downloaded and subsequently copied by the initial downloader and then passed along to other parties who may pass the object on to additional parties.

- 5 ! provide very flexible and extensible user
 identification according to individuals, installations,
 by groups such as classes, and by function and
 hierarchical identification employing a hierarchy of
 levels of client identification (for example, client
 organization ID, client department ID, client
 network ID, client project ID, and client employee
 ID, or any appropriate subset of the above).
- 10 ! provide a general purpose, secure, component based
 content control and distribution system that
 functions as a foundation transaction operating
 system environment that employs executable code
 pieces crafted for transaction control and auditing.
15 These code pieces can be reused to optimize
 efficiency in creation and operation of trusted,
 distributed transaction management arrangements.
 VDE supports providing such executable code in the
 form of "atomic" load modules and associated data.
20 Many such load modules are inherently configurable,
 aggregatable, portable, and extensible and
 singularly, or in combination (along with associated
 data), run as control methods under the VDE
 transaction operating environment. VDE can satisfy
25 the requirements of widely differing electronic

commerce and data security applications by, in part,
employing this general purpose transaction
management foundation to securely process VDE
transaction related control methods. Control
5 methods are created primarily through the use of
one or more of said executable, reusable load module
code pieces (normally in the form of executable object
components) and associated data. The component
nature of control methods allows the present
10 invention to efficiently operate as a highly
configurable content control system. Under the
present invention, content control models can be
iteratively and asynchronously shaped, and
otherwise updated to accommodate the needs of VDE
15 participants to the extent that such shaping and
otherwise updating conforms to constraints applied
by a VDE application, if any (e.g., whether new
component assemblies are accepted and, if so, what
certification requirements exist for such component
20 assemblies or whether any or certain participants
may shape any or certain control information by
selection amongst optional control information
(permissions record) control methods. This iterative
(or concurrent) multiple participant process occurs
25 as a result of the submission and use of secure,

control information components (executable code
such as load modules and/or methods, and/or
associated data). These components may be
contributed independently by secure communication
5 between each control information influencing VDE
participant's VDE installation and may require
certification for use with a given application, where
such certification was provided by a certification
service manager for the VDE arrangement who
10 ensures secure interoperability and/or reliability
(e.g., bug control resulting from interaction) between
appliances and submitted control methods. The
transaction management control functions of a VDE
electronic appliance transaction operating
15 environment interact with non-secure transaction
management operating system functions to properly
direct transaction processes and data related to
electronic information security, usage control,
auditing, and usage reporting. VDE provides the
20 capability to manages resources related to secure
VDE content and/or appliance control information
execution and data storage.

! facilitate creation of application and/or system
25 functionality under VDE and to facilitate integration

into electronic appliance environments of load modules and methods created under the present invention. To achieve this, VDE employs an Application Programmer's Interface (API) and/or a transaction operating system (such as a ROS) programming language with incorporated functions, both of which support the use of capabilities and can be used to efficiently and tightly integrate VDE functionality into commercial and user applications.

support user interaction through: (a) "Pop-Up" applications which, for example, provide messages to users and enable users to take specific actions such as approving a transaction, (b) stand-alone VDE applications that provide administrative environments for user activities such as: end-user preference specifications for limiting the price per transaction, unit of time, and/or session, for accessing history information concerning previous transactions, for reviewing financial information such as budgets, expenditures (e.g. detailed and/or summary) and usage analysis information, and (c) VDE aware applications which, as a result of the use of a VDE API and/or a transaction management (for

example, ROS based) programming language
embeds VDE "awareness" into commercial or
internal software (application programs, games, etc.)
so that VDE user control information and services
are seamlessly integrated into such software and can
be directly accessed by a user since the underlying
functionality has been integrated into the
commercial software's native design. For example,
in a VDE aware word processor application, a user
may be able to "print" a document into a VDE
content container object, applying specific control
information by selecting from amongst a series of
different menu templates for different purposes (for
example, a confidential memo template for internal
organization purposes may restrict the ability to
"keep," that is to make an electronic copy of the
memo).

! employ "templates" to ease the process of configuring
capabilities of the present invention as they relate to
specific industries or businesses. Templates are
applications or application add-ons under the
present invention. Templates support the efficient
specification and/or manipulation of criteria related
to specific content types, distribution approaches,

pricing mechanisms, user interactions with content
and/or administrative activities, and/or the like.
Given the very large range of capabilities and
configurations supported by the present invention,
5 reducing the range of configuration opportunities to
a manageable subset particularly appropriate for a
given business model allows the full configurable
power of the present invention to be easily employed
by "typical" users who would be otherwise burdened
10 with complex programming and/or configuration
design responsibilities template applications can also
help ensure that VDE related processes are secure
and optimally bug free by reducing the risks
associated with the contribution of independently
15 developed load modules, including unpredictable
aspects of code interaction between independent
modules and applications, as well as security risks
associated with possible presence of viruses in such
modules. VDE, through the use of templates,
20 reduces typical user configuration responsibilities to
an appropriately focused set of activities including
selection of method types (e.g. functionality) through
menu choices such as multiple choice, icon selection,
and/or prompting for method parameter data (such
25 as identification information, prices, budget limits,

5 dates, periods of time, access rights to specific
 content, etc.) that supply appropriate and/or
 necessary data for control information purposes. By
 limiting the typical (non-programming) user to a
10 limited subset of configuration activities whose
 general configuration environment (template) has
 been preset to reflect general requirements
 corresponding to that user, or a content or other
 business model can very substantially limit
15 difficulties associated with content containerization
 (including placing initial control information on
 content), distribution, client administration,
 electronic agreement implementation, end-user
 interaction, and clearinghouse activities, including
20 associated interoperability problems (such as
 conflicts resulting from security, operating system,
 and/or certification incompatibilities). Use of
 appropriate VDE templates can assure users that
 their activities related to content VDE
25 containerization, contribution of other control
 information, communications, encryption techniques
 and/or keys, etc. will be in compliance with
 specifications for their distributed VDE
 arrangement. VDE templates constitute preset
 configurations that can normally be reconfigurable

to allow for new and/or modified templates that reflect adaptation into new industries as they evolve or to reflect the evolution or other change of an existing industry. For example, the template concept may be used to provide individual, overall frameworks for organizations and individuals that create, modify, market, distribute, consume, and/or otherwise use movies, audio recordings and live performances, magazines, telephony based retail sales, catalogs, computer software, information data bases, multimedia, commercial communications, advertisements, market surveys, infomercials, games, CAD/CAM services for numerically controlled machines, and the like. As the context surrounding these templates changes or evolves, template applications provided under the present invention may be modified to meet these changes for broad use, or for more focused activities. A given VDE participant may have a plurality of templates available for different tasks. A party that places content in its initial VDE container may have a variety of different, configurable templates depending on the type of content and/or business model related to the content. An end-user may have different configurable templates that can be applied

to different document types (e-mail, secure internal documents, database records, etc.) and/or subsets of users (applying differing general sets of control information to different bodies of users, for example, selecting a list of users who may, under certain preset criteria, use a certain document). Of course, templates may, under certain circumstances have fixed control information and not provide for user selections or parameter data entry.

support plural, different control models regulating the use and/or auditing of either the same specific copy of electronic information content and/or differently regulating different copies (occurrences) of the same electronic information content. Differing models for billing, auditing, and security can be applied to the same piece of electronic information content and such differing sets of control information may employ, for control purposes, the same, or differing, granularities of electronic information control increments. This includes supporting variable control information for budgeting and auditing usage as applied to a variety of predefined increments of electronic information, including employing a variety of different budgets and/or

metering increments for a given electronic
information deliverable for: billing units of measure,
credit limit, security budget limit and security
content metering increments, and/or market
5 surveying and customer profiling content metering
increments. For example, a CD-ROM disk with a
database of scientific articles might be in part billed
according to a formula based on the number of bytes
decrypted, number of articles containing said bytes
10 decrypted, while a security budget might limit the
use of said database to no more than 5% of the
database per month for users on the wide area
network it is installed on.

15 ! provide mechanisms to persistently maintain trusted
content usage and reporting control information
through both a sufficiently secure chain of handling
of content and content control information and
through various forms of usage of such content
20 wherein said persistence of control may survive such
use. Persistence of control includes the ability to
extract information from a VDE container object by
creating a new container whose contents are at least
in part secured and that contains both the extracted
25 content and at least a portion of the control

information which control information of the original
container and/or are at least in part produced by
control information of the original container for this
purpose and/or VDE installation control information
5 stipulates should persist and/or control usage of
content in the newly formed container. Such control
information can continue to manage usage of
container content if the container is "embedded" into
another VDE managed object, such as an object
10 which contains plural embedded VDE containers,
each of which contains content derived (extracted)
from a different source.

! enables users, other value chain participants (such
15 as clearinghouses and government agencies), and/or
user organizations, to specify preferences or
requirements related to their use of electronic
content and/or appliances. Content users, such as
end-user customers using commercially distributed
20 content (games, information resources, software
programs, etc.), can define, if allowed by senior
control information, budgets, and/or other control
information, to manage their own internal use of
content. Uses include, for example, a user setting a
25 limit on the price for electronic documents that the

user is willing to pay without prior express user authorization, and the user establishing the character of metering information he or she is willing to allow to be collected (privacy protection).
5 This includes providing the means for content users to protect the privacy of information derived from their use of a VDE installation and content and/or appliance usage auditing. In particular, VDE can prevent information related to a participant's usage
10 of electronic content from being provided to other parties without the participant's tacit or explicit agreement.

! provide mechanisms that allow control information
15 to "evolve" and be modified according, at least in part, to independently, securely delivered further control information. Said control information may include executable code (e.g., load modules) that has been certified as acceptable (e.g., reliable and
20 trusted) for use with a specific VDE application, class of applications, and/or a VDE distributed arrangement. This modification (evolution) of control information can occur upon content control information (load modules and any associated data)
25 circulating to one or more VDE participants in a

pathway of handling of control information, or it may occur upon control information being received from a VDE participant. Handlers in a pathway of handling of content control information, to the extent each is authorized, can establish, modify, and/or contribute to, permission, auditing, payment, and reporting control information related to controlling, analyzing, paying for, and/or reporting usage of, electronic content and/or appliances (for example, as related to usage of VDE controlled property content). Independently delivered (from an independent source which is independent except in regards to certification), at least in part secure, control information can be employed to securely modify content control information when content control information has flowed from one party to another party in a sequence of VDE content control information handling. This modification employs, for example, one or more VDE component assemblies being securely processed in a VDE secure subsystem. In an alternate embodiment, control information may be modified by a senior party through use of their VDE installation secure sub-system after receiving submitted, at least in part secured, control information from a "junior" party, normally in the

form of a VDE administrative object. Control information passing along VDE pathways can represent a mixed control set, in that it may include: control information that persisted through a sequence of control information handlers, other control information that was allowed to be modified, and further control information representing new control information and/or mediating data. Such a control set represents an evolution of control information for disseminated content. In this example the overall content control set for a VDE content container is "evolving" as it securely (e.g. communicated in encrypted form and using authentication and digital signaturing techniques) passes, at least in part, to a new participant's VDE installation where the proposed control information is securely received and handled. The received control information may be integrated (through use of the receiving parties' VDE installation secure sub-system) with in-place control information through a negotiation process involving both control information sets. For example, the modification, within the secure sub-system of a content provider's VDE installation, of content control information for a certain VDE content container may have occurred as

a result of the incorporation of required control information provided by a financial credit provider. Said credit provider may have employed their VDE installation to prepare and securely communicate (directly or indirectly) said required control information to said content provider. Incorporating said required control information enables a content provider to allow the credit provider's credit to be employed by a content end-user to compensate for the end-user's use of VDE controlled content and/or appliances, so long as said end-user has a credit account with said financial credit provider and said credit account has sufficient credit available. Similarly, control information requiring the payment of taxes and/or the provision of revenue information resulting from electronic commerce activities may be securely received by a content provider. This control information may be received, for example, from a government agency. Content providers might be required by law to incorporate such control information into the control information for commercially distributed content and/or services related to appliance usage. Proposed control information is used to an extent allowed by senior control information and as determined by any

5

negotiation trade-offs that satisfy priorities stipulated by each set (the received set and the proposed set). VDE also accommodates different control schemes specifically applying to different participants (e.g., individual participants and/or participant classes (types)) in a network of VDE content handling participants.

10

! support multiple simultaneous control models for the same content property and/or property portion.

15

This allows, for example, for concurrent business activities which are dependent on electronic commercial product content distribution, such as acquiring detailed market survey information and/or supporting advertising, both of which can increase revenue and result in lower content costs to users and greater value to content providers. Such control information and/or overall control models may be applied, as determined or allowed by control

20

information, in differing manners to different participants in a pathway of content, reporting, payment, and/or related control information handling. VDE supports applying different content control information to the same and/or different content and/or appliance usage related activities,

25

and/or to different parties in a content and/or
appliance usage model, such that different parties
(or classes of VDE users, for example) are subject to
differing control information managing their use of
5 electronic information content. For example,
differing control models based on the category of a
user as a distributor of a VDE controlled content
object or an end-user of such content may result in
different budgets being applied. Alternatively, for
10 example, a one distributor may have the right to
distribute a different array of properties than
another distributor (from a common content
collection provided, for example, on optical disc). An
individual, and/or a class or other grouping of
15 end-users, may have different costs (for example, a
student, senior citizen, and/or poor citizen user of
content who may be provided with the same or
differing discounts) than a "typical" content user.

20 ! support provider revenue information resulting from
customer use of content and/or appliances, and/or
provider and/or end-user payment of taxes, through
the transfer of credit and/or electronic currency from
said end-user and/or provider to a government
25 agency, might occur "automatically" as a result of

such received control information causing the generation of a VDE content container whose content includes customer content usage information reflecting secure, trusted revenue summary information and/or detailed user transaction listings (level of detail might depend, for example on type or size of transaction—information regarding a bank interest payment to a customer or a transfer of a large (e.g. over \$10,000) might be, by law, automatically reported to the government). Such summary and/or detailed information related to taxable events and/or currency, and/or creditor currency transfer, may be passed along a pathway of reporting and/or payment to the government in a VDE container. Such a container may also be used for other VDE related content usage reporting information.

! support the flowing of content control information through different “branches” of content control information handling so as to accommodate, under the present invention’s preferred embodiment, diverse controlled distributions of VDE controlled content. This allows different parties to employ the same initial electronic content with differing

(perhaps competitive) control strategies. In this instance, a party who first placed control information on content can make certain control assumptions and these assumptions would evolve into more specific and/or extensive control assumptions. These control assumptions can evolve during the branching sequence upon content model participants submitting control information changes, for example, for use in "negotiating" with "in place" content control information. This can result in new or modified content control information and/or it might involve the selection of certain one or more already "in-place" content usage control methods over in-place alternative methods, as well as the submission of relevant control information parameter data. This form of evolution of different control information sets applied to different copies of the same electronic property content and/or appliance results from VDE control information flowing "down" through different branches in an overall pathway of handling and control and being modified differently as it diverges down these different pathway branches. This ability of the present invention to support multiple pathway branches for the flow of both VDE content control

information and VDE managed content enables an electronic commerce marketplace which supports diverging, competitive business partnerships, agreements, and evolving overall business models which can employ the same content properties combined, for example, in differing collections of content representing differing at least in part competitive products.

10 ! enable a user to securely extract, through the use of the secure subsystem at the user's VDE installation, at least a portion of the content included within a VDE content container to produce a new, secure object (content container), such that the extracted information is maintained in a continually secure manner through the extraction process. Formation of the new VDE container containing such extracted content shall result in control information consistent with, or specified by, the source VDE content container, and/or local VDE installation secure subsystem as appropriate, content control information. Relevant control information, such as security and administrative information, derived, at least in part, from the parent (source) object's control information, will normally be automatically inserted

into a new VDE content container object containing
extracted VDE content. This process typically occurs
under the control framework of a parent object
and/or VDE installation control information
executing at the user's VDE installation secure
subsystem (with, for example, at least a portion of
this inserted control information being stored
securely in encrypted form in one or more
permissions records). In an alternative embodiment,
the derived content control information applied to
extracted content may be in part or whole derived
from, or employ, content control information stored
remotely from the VDE installation that performed
the secure extraction such as at a remote server
location. As with the content control information for
most VDE managed content, features of the present
invention allows the content's control information to:

(a) "evolve," for example, the extractor of content
may add new control methods and/or modify
control parameter data, such as VDE
application compliant methods, to the extent
allowed by the content's in-place control
information. Such new control information
might specify, for example, who may use at

5

least a portion of the new object, and/or how said at least a portion of said extracted content may be used (e.g. when at least a portion may be used, or what portion or quantity of portions may be used);

10

- (b) allow a user to combine additional content with at least a portion of said extracted content, such as material authored by the extractor and/or content (for example, images, video, audio, and/or text) extracted from one or more other VDE container objects for placement directly into the new container;

15

- (c) allow a user to securely edit at least a portion of said content while maintaining said content in a secure form within said VDE content container;

20

- (d) append extracted content to a pre-existing VDE content container object and attach associated control information -- in these cases, user added information may be secured, e.g., encrypted, in part or as a whole, and may

25

be subject to usage and/or auditing control

information that differs from the those applied
to previously in place object content;

- 5 (e) preserve VDE control over one or more
portions of extracted content after various
forms of usage of said portions, for example,
maintain content in securely stored form while
allowing "temporary" on screen display of
content or allowing a software program to be
10 maintained in secure form but transiently
decrypt any encrypted executing portion of
said program (all, or only a portion, of said
program may be encrypted to secure the
program).

15
Generally, the extraction features of the present
invention allow users to aggregate and/or
disseminate and/or otherwise use protected
electronic content information extracted from
20 content container sources while maintaining secure
VDE capabilities thus preserving the rights of
providers in said content information after various
content usage processes.

! support the aggregation of portions of VDE
controlled content, such portions being subject to
differing VDE content container control information,
wherein various of said portions may have been
5 provided by independent, different content providers
from one or more different locations remote to the
user performing the aggregation. Such aggregation,
in the preferred embodiment of the present
invention, may involve preserving at least a portion
10 of the control information (e.g., executable code such
as load modules) for each of various of said portions
by, for example, embedding some or all of such
portions individually as VDE content container
objects within an overall VDE content container
15 and/or embedding some or all of such portions
directly into a VDE content container. In the latter
case, content control information of said content
container may apply differing control information
sets to various of such portions based upon said
20 portions original control information requirements
before aggregation. Each of such embedded VDE
content containers may have its own control
information in the form of one or more permissions
records. Alternatively, a negotiation between control
25 information associated with various aggregated

portions of electronic content, may produce a control information set that would govern some or all of the aggregated content portions. The VDE content control information produced by the negotiation may

5 be uniform (such as having the same load modules and/or component assemblies, and/or it may apply differing such content control information to two or more portions that constitute an aggregation of VDE controlled content such as differing metering,

10 budgeting, billing and/or payment models. For example, content usage payment may be automatically made, either through a clearinghouse, or directly, to different content providers for different portions.

15

! enable flexible metering of, or other collection of information related to, use of electronic content and/or electronic appliances. A feature of the present invention enables such flexibility of metering

20 control mechanisms to accommodate a simultaneous, broad array of: (a) different parameters related to electronic information content use; (b) different increment units (bytes, documents, properties, paragraphs, images, etc.) and/or other organizations

25 of such electronic content; and/or (c) different

categories of user and/or VDE installation types, such as client organizations, departments, projects, networks, and/or individual users, etc. This feature of the present invention can be employed for content security, usage analysis (for example, market surveying), and/or compensation based upon the use and/or exposure to VDE managed content. Such metering is a flexible basis for ensuring payment for content royalties, licensing, purchasing, and/or advertising. A feature of the present invention provides for payment means supporting flexible electronic currency and credit mechanisms, including the ability to securely maintain audit trails reflecting information related to use of such currency or credit. VDE supports multiple differing hierarchies of client organization control information wherein an organization client administrator distributes control information specifying the usage rights of departments, users, and/or projects. Likewise, a department (division) network manager can function as a distributor (budgets, access rights, etc.) for department networks, projects, and/or users, etc.

! provide scalable, integratable, standardized control means for use on electronic appliances ranging from inexpensive consumer (for example, television set-top appliances) and professional devices (and
5 hand-held PDAs) to servers, mainframes, communication switches, etc. The scalable transaction management/auditing technology of the present invention will result in more efficient and reliable interoperability amongst devices functioning
10 in electronic commerce and/or data security environments. As standardized physical containers have become essential to the shipping of physical goods around the world, allowing these physical containers to universally "fit" unloading equipment,
15 efficiently use truck and train space, and accommodate known arrays of objects (for example, boxes) in an efficient manner, so VDE electronic content containers may, as provided by the present invention, be able to efficiently move electronic
20 information content (such as commercially published properties, electronic currency and credit, and content audit information), and associated content control information, around the world. Interoperability is fundamental to efficient electronic
25 commerce. The design of the VDE foundation, VDE

load modules, and VDE containers, are important features that enable the VDE node operating environment to be compatible with a very broad range of electronic appliances. The ability, for example, for control methods based on load modules to execute in very "small" and inexpensive secure sub-system environments, such as environments with very little read/write memory, while also being able to execute in large memory sub-systems that may be used in more expensive electronic appliances, supports consistency across many machines. This consistent VDE operating environment, including its control structures and container architecture, enables the use of standardized VDE content containers across a broad range of device types and host operating environments. Since VDE capabilities can be seamlessly integrated as extensions, additions, and/or modifications to fundamental capabilities of electronic appliances and host operating systems, VDE containers, content control information, and the VDE foundation will be able to work with many device types and these device types will be able to consistently and efficiently interpret and enforce VDE control information. Through this integration users can also

benefit from a transparent interaction with many of the capabilities of VDE. VDE integration with software operating on a host electronic appliance supports a variety of capabilities that would be unavailable or less secure without such integration. Through integration with one or more device applications and/or device operating environments, many capabilities of the present invention can be presented as inherent capabilities of a given electronic appliance, operating system, or appliance application. For example, features of the present invention include: (a) VDE system software to in part extend and/or modify host operating systems such that they possesses VDE capabilities, such as enabling secure transaction processing and electronic information storage; (b) one or more application programs that in part represent tools associated with VDE operation; and/or (c) code to be integrated into application programs, wherein such code incorporates references into VDE system software to integrate VDE capabilities and makes such applications VDE aware (for example, word processors, database retrieval applications, spreadsheets, multimedia presentation authoring tools, film editing software, music editing software

such as MIDI applications and the like, robotics control systems such as those associated with CAD/CAM environments and NCM software and the like, electronic mail systems, teleconferencing software, and other data authoring, creating, handling, and/or usage applications including combinations of the above). These one or more features (which may also be implemented in firmware or hardware) may be employed in conjunction with a VDE node secure hardware processing capability, such as a microcontroller(s), microprocessor(s), other CPU(s) or other digital processing logic.

! employ audit reconciliation and usage pattern evaluation processes that assess, through certain, normally network based, transaction processing reconciliation and threshold checking activities, whether certain violations of security of a VDE arrangement have occurred. These processes are performed remote to VDE controlled content end-user VDE locations by assessing, for example, purchases, and/or requests, for electronic properties by a given VDE installation. Applications for such reconciliation activities include assessing whether

the quantity of remotely delivered VDE controlled content corresponds to the amount of financial credit and/or electronic currency employed for the use of such content. A trusted organization can acquire information from content providers concerning the cost for content provided to a given VDE installation and/or user and compare this cost for content with the credit and/or electronic currency disbursements for that installation and/or user. Inconsistencies in the amount of content delivered versus the amount of disbursement can prove, and/or indicate, depending on the circumstances, whether the local VDE installation has been, at least to some degree, compromised (for example, certain important system security functions, such as breaking encryption for at least some portion of the secure subsystem and/or VDE controlled content by uncovering one or more keys). Determining whether irregular patterns (e.g. unusually high demand) of content usage, or requests for delivery of certain kinds of VDE controlled information during a certain time period by one or more VDE installations and/or users (including, for example, groups of related users whose aggregate pattern of usage is suspicious) may also be useful in determining whether security at

such one or more installations, and/or by such one or more users, has been compromised, particularly when used in combination with an assessment of electronic credit and/or currency provided to one or more VDE users and/or installations, by some or all of their credit and/or currency suppliers, compared with the disbursements made by such users and/or installations.

10 ! support security techniques that materially increase the time required to "break" a system's integrity. This includes using a collection of techniques that minimizes the damage resulting from comprising some aspect of the security features of the present inventions.

15 ! provide a family of authoring, administrative, reporting, payment, and billing tool user applications that comprise components of the present invention's trusted/secure, universe wide, distributed transaction control and administration system. These components support VDE related: object creation (including placing control information on content), secure object distribution and management (including distribution control information, financial

related, and other usage analysis), client internal VDE activities administration and control, security management, user interfaces, payment disbursement, and clearinghouse related functions.

5 These components are designed to support highly secure, uniform, consistent, and standardized: electronic commerce and/or data security pathway(s) of handling, reporting, and/or payment; content control and administration; and human factors (e.g.

10 user interfaces).

! support the operation of a plurality of clearinghouses, including, for example, both financial and user clearinghouse activities, such as

15 those performed by a client administrator in a large organization to assist in the organization's use of a VDE arrangement, including usage information analysis, and control of VDE activities by individuals and groups of employees such as specifying budgets

20 and the character of usage rights available under VDE for certain groups of and/or individual, client personnel, subject to control information series to control information submitted by the client administrator. At a clearinghouse, one or more VDE

25 installations may operate together with a trusted

distributed database environment (which may include concurrent database processing means). A financial clearinghouse normally receives at its location securely delivered content usage information, and user requests (such as requests for further credit, electronic currency, and/or higher credit limit). Reporting of usage information and user requests can be used for supporting electronic currency, billing, payment and credit related activities, and/or for user profile analysis and/or broader market survey analysis and marketing (consolidated) list generation or other information derived, at least in part, from said usage information. this information can be provided to content providers or other parties, through secure, authenticated encrypted communication to the VDE installation secure subsystems. Clearinghouse processing means would normally be connected to specialized I/O means, which may include high speed telecommunication switching means that may be used for secure communications between a clearinghouse and other VDE pathway participants.

securely support electronic currency and credit usage control, storage, and communication at, and

between, VDE installations. VDE further supports automated passing of electronic currency and/or credit information, including payment tokens (such as in the form of electronic currency or credit) or other payment information, through a pathway of payment, which said pathway may or may not be the same as a pathway for content usage information reporting. Such payment may be placed into a VDE container created automatically by a VDE installation in response to control information stipulating the "withdrawal" of credit or electronic currency from an electronic credit or currency account based upon an amount owed resulting from usage of VDE controlled electronic content and/or appliances. Payment credit or currency may then be automatically communicated in protected (at least in part encrypted) form through telecommunication of a VDE container to an appropriate party such as a clearinghouse, provider of original property content or appliance, or an agent for such provider (other than a clearinghouse). Payment information may be packaged in said VDE content container with, or without, related content usage information, such as metering information. An aspect of the present invention further enables certain information

regarding currency use to be specified as unavailable
to certain, some, or all VDE parties ("conditionally"
to fully anonymous currency) and/or further can
regulate certain content information, such as
5 currency and/or credit use related information
(and/or other electronic information usage data) to be
available only under certain strict circumstances,
such as a court order (which may itself require
authorization through the use of a court controlled
10 VDE installation that may be required to securely
access "conditionally" anonymous information).
Currency and credit information, under the
preferred embodiment of the present invention, is
treated as administrative content;

15

! support fingerprinting (also known as
watermarking) for embedding in content such that
when content protected under the present invention
is released in clear form from a VDE object
20 (displayed, printed, communicated, extracted, and/or
saved), information representing the identification of
the user and/or VDE installation responsible for
transforming the content into clear form is
embedded into the released content. Fingerprinting
25 is useful in providing an ability to identify who

5 extracted information in clear form a VDE container,
or who made a copy of a VDE object or a portion of
its contents. Since the identity of the user and/or
other identifying information may be embedded in
an obscure or generally concealed manner, in VDE
container content and/or control information,
potential copyright violators may be deterred from
unauthorized extraction or copying. Fingerprinting
normally is embedded into unencrypted electronic
content or control information, though it can be
embedded into encrypted content and later placed in
unencrypted content in a secure VDE installation
sub-system as the encrypted content carrying the
fingerprinting information is decrypted. Electronic
information, such as the content of a VDE container,
may be fingerprinted as it leaves a network (such as
Internet) location bound for a receiving party. Such
repository information may be maintained in
unencrypted form prior to communication and be
encrypted as it leaves the repository. Fingerprinting
would preferably take place as the content leaves the
repository, but before the encryption step.
Encrypted repository content can be decrypted, for
example in a secure VDE sub-system, fingerprint
information can be inserted, and then the content

can be re-encrypted for transmission. Embedding identification information of the intended recipient user and/or VDE installation into content as it leaves, for example, an Internet repository, would provide important information that would identify or assist in identifying any party that managed to compromise the security of a VDE installation or the delivered content. If a party produces an authorized clear form copy of VDE controlled content, including making unauthorized copies of an authorized clear form copy, fingerprint information would point back to that individual and/or his or her VDE installation. Such hidden information will act as a strong disincentive that should dissuade a substantial portion of potential content "pirates" from stealing other parties electronic information. Fingerprint information identifying a receiving party and/or VDE installation can be embedded into a VDE object before, or during, decryption, replication, or communication of VDE content objects to receivers. Fingerprinting electronic content before it is encrypted for transfer to a customer or other user provides information that can be very useful for identifying who received certain content which may have then been distributed or made available in

unencrypted form. This information would be useful in tracking who may have "broken" the security of a VDE installation and was illegally making certain electronic content available to others.

5 Fingerprinting may provide additional, available information such as time and/or date of the release (for example extraction) of said content information. Locations for inserting fingerprints may be specified by VDE installation and/or content container control
10 information. This information may specify that certain areas and/or precise locations within properties should be used for fingerprinting, such as one or more certain fields of information or information types. Fingerprinting information may
15 be incorporated into a property by modifying in a normally undetectable way color frequency and/or the brightness of certain image pixels, by slightly modifying certain audio signals as to frequency, by modifying font character formation, etc. Fingerprint
20 information, itself, should be encrypted so as to make it particularly difficult for tampered fingerprints to be interpreted as valid. Variations in fingerprint locations for different copies of the same property; "false" fingerprint information; and
25 multiple copies of fingerprint information within a

specific property or other content which copies
employ different fingerprinting techniques such as
information distribution patterns, frequency and/or
brightness manipulation, and encryption related
5 techniques, are features of the present invention for
increasing the difficulty of an unauthorized
individual identifying fingerprint locations and
erasing and/or modifying fingerprint information.

10 ! provide smart object agents that can carry requests,
data, and/or methods, including budgets,
authorizations, credit or currency, and content. For
example, smart objects may travel to and/or from
remote information resource locations and fulfill
15 requests for electronic information content. Smart
objects can, for example, be transmitted to a remote
location to perform a specified database search on
behalf of a user or otherwise "intelligently" search
remote one or more repositories of information for
20 user desired information. After identifying desired
information at one or more remote locations, by for
example, performing one or more database searches,
a smart object may return via communication to the
user in the form of a secure "return object"
25 containing retrieved information. A user may be

charged for the remote retrieving of information, the
returning of information to the user's VDE
installation, and/or the use of such information. In
the latter case, a user may be charged only for the
5 information in the return object that the user
actually uses. Smart objects may have the means to
request use of one or more services and/or resources.
Services include locating other services and/or
resources such as information resources, language or
10 format translation, processing, credit (or additional
credit) authorization, etc. Resources include
reference databases, networks, high powered or
specialized computing resources (the smart object
may carry information to another computer to be
15 efficiently processed and then return the information
to the sending VDE installation), remote object
repositories, etc. Smart objects can make efficient
use of remote resources (e.g. centralized databases,
super computers, etc.) while providing a secure
20 means for charging users based on information
and/or resources actually used.

! support both "translations" of VDE electronic
agreements elements into modern language printed
25 agreement elements (such as English language

agreements) and translations of electronic rights
protection/transaction management modern
language agreement elements to electronic VDE
agreement elements. This feature requires
5 maintaining a library of textual language that
corresponds to VDE load modules and/or methods
and/or component assemblies. As VDE methods are
proposed and/or employed for VDE agreements, a
listing of textual terms and conditions can be
10 produced by a VDE user application which, in a
preferred embodiment, provides phrases, sentences
and/or paragraphs that have been stored and
correspond to said methods and/or assemblies. This
feature preferably employs artificial intelligence
15 capabilities to analyze and automatically determine,
and/or assist one or more users to determine, the
proper order and relationship between the library
elements corresponding to the chosen methods
and/or assemblies so as to compose some or all
20 portions of a legal or descriptive document. One or
more users, and/or preferably an attorney (if the
document a legal, binding agreement), would review
the generated document material upon completion
and employ such additional textual information
25 and/or editing as necessary to describe non electronic

transaction elements of the agreement and make
any other improvements that may be necessary.
These features further support employing modern
language tools that allow one or more users to make
5 selections from choices and provide answers to
questions and to produce a VDE electronic
agreement from such a process. This process can be
interactive and the VDE agreement formulation
process may employ artificial intelligence expert
10 system technology that learns from responses and,
where appropriate and based at least in part on said
responses, provides further choices and/or questions
which "evolves" the desired VDE electronic
agreement.

15
! support the use of multiple VDE secure subsystems
in a single VDE installation. Various security and/or
performance advantages may be realized by
employing a distributed VDE design within a single
20 VDE installation. For example, designing a
hardware based VDE secure subsystem into an
electronic appliance VDE display device, and
designing said subsystem's integration with said
display device so that it is as close as possible to the
25 point of display, will increase the security for video

materials by making it materially more difficult to
"steal" decrypted video information as it moves from
outside to inside the video system. Ideally, for
example, a VDE secure hardware module would be
5 in the same physical package as the actual display
monitor, such as within the packaging of a video
monitor or other display device, and such device
would be designed, to the extent commercially
practical, to be as tamper resistant as reasonable.
10 As another example, embedding a VDE hardware
module into an I/O peripheral may have certain
advantages from the standpoint of overall system
throughput. If multiple VDE instances are employed
within the same VDE installation, these instances
15 will ideally share resources to the extent practical,
such as VDE instances storing certain control
information and content and/or appliance usage
information on the same mass storage device and in
the same VDE management database.

20

! requiring reporting and payment compliance by
employing exhaustion of budgets and time ageing of
keys. For example, a VDE commercial arrangement
and associated content control information may
25 involve a content provider's content and the use of

clearinghouse credit for payment for end-user usage
of said content. Control information regarding said
arrangement may be delivered to a user's (of said
content) VDE installation and/or said financial
5 clearinghouse's VDE installation. Said control
information might require said clearinghouse to
prepare and telecommunicate to said content
provider both content usage based information in a
certain form, and content usage payment in the form
10 of electronic credit (such credit might be "owned" by
the provider after receipt and used in lieu of the
availability or adequacy of electronic currency)
and/or electronic currency. This delivery of
information and payment may employ trusted VDE
15 installation secure subsystems to securely, and in
some embodiments, automatically, provide in the
manner specified by said control information, said
usage information and payment content. Features of
the present invention help ensure that a
20 requirement that a clearinghouse report such usage
information and payment content will be observed.
For example, if one participant to a VDE electronic
agreement fails to observe such information
reporting and/or paying obligation, another
25 participant can stop the delinquent party from

5 successfully participating in VDE activities related
to such agreement. For example, if required usage
information and payment was not reported as
specified by content control information, the
“injured” party can fail to provide, through failing to
securely communicate from his VDE installation
secure subsystem, one or more pieces of secure
information necessary for the continuance of one or
more critical processes. For example, failure to
10 report information and/or payment from a
clearinghouse to a content provider (as well as any
security failures or other disturbing irregularities)
can result in the content provider not providing key
and/or budget refresh information to the
15 clearinghouse, which information can be necessary
to authorize use of the clearinghouse’s credit for
usage of the provider’s content and which the
clearinghouse would communicate to end-user’s
during a content usage reporting communication
20 between the clearinghouse and end-user. As another
example, a distributor that failed to make payments
and/or report usage information to a content
provider might find that their budget for creating
permissions records to distribute the content
25 provider’s content to users, and/or a security budget

limiting one or more other aspect of their use of the provider's content, are not being refreshed by the content provider, once exhausted or timed-out (for example, at a predetermined date). In these and other cases, the offended party might decide not to refresh time ageing keys that had "aged out." Such a use of time aged keys has a similar impact as failing to refresh budgets or time-aged authorizations.

support smart card implementations of the present invention in the form of portable electronic appliances, including cards that can be employed as secure credit, banking, and/or money cards. A feature of the present invention is the use of portable VDEs as transaction cards at retail and other establishments, wherein such cards can "dock" with an establishment terminal that has a VDE secure sub-system and/or an online connection to a VDE secure and/or otherwise secure and compatible subsystem, such as a "trusted" financial clearinghouse (e.g., VISA, Mastercard). The VDE card and the terminal (and/or online connection) can securely exchange information related to a transaction, with credit and/or electronic currency being transferred to a merchant and/or

clearinghouse and transaction information flowing
back to the card. Such a card can be used for
transaction activities of all sorts. A docking station,
such as a PCMCIA connector on an electronic
5 appliance, such as a personal computer, can receive
a consumer's VDE card at home. Such a station/card
combination can be used for on-line transactions in
the same manner as a VDE installation that is
permanently installed in such an electronic
10 appliance. The card can be used as an "electronic
wallet" and contain electronic currency as well as
credit provided by a clearinghouse. The card can act
as a convergence point for financial activities of a
consumer regarding many, if not all, merchant,
15 banking, and on-line financial transactions,
including supporting home banking activities. A
consumer can receive his paycheck and/or
investment earnings and/or "authentic" VDE content
container secured detailed information on such
20 receipts, through on-line connections. A user can
send digital currency to another party with a VDE
arrangement, including giving away such currency.
A VDE card can retain details of transactions in a
highly secure and database organized fashion so that
25 financially related information is both consolidated

and very easily retrieved and/or analyzed. Because of the VDE security, including use of effective encryption, authentication, digital signaturing, and secure database structures, the records contained within a VDE card arrangement may be accepted as valid transaction records for government and/or corporate recordkeeping requirements. In some embodiments of the present invention a VDE card may employ docking station and/or electronic appliance storage means and/or share other VDE arrangement means local to said appliance and/or available across a network, to augment the information storage capacity of the VDE card, by for example, storing dated, and/or archived, backup information. Taxes relating to some or all of an individual's financial activities may be automatically computed based on "authentic" information securely stored and available to said VDE card. Said information may be stored in said card, in said docking station, in an associated electronic appliance, and/or other device operatively attached thereto, and/or remotely, such as at a remote server site. A card's data, e.g. transaction history, can be backed up to an individual's personal computer or other electronic appliance and such an appliance

may have an integrated VDE installation of its own.
A current transaction, recent transactions (for
redundancy), or all or other selected card data may
be backed up to a remote backup repository, such a
5 VDE compatible repository at a financial
clearinghouse, during each or periodic docking for a
financial transaction and/or information
communication such as a user/merchant transaction.
Backing up at least the current transaction during a
10 connection with another party's VDE installation
(for example a VDE installation that is also on a
financial or general purpose electronic network), by
posting transaction information to a remote
clearinghouse and/or bank, can ensure that
15 sufficient backup is conducted to enable complete
reconstruction of VDE card internal information in
the event of a card failure or loss.

! support certification processes that ensure
20 authorized interoperability between various VDE
installations so as to prevent VDE arrangements
and/or installations that unacceptably deviate in
specification protocols from other VDE arrangements
and/or installations from interoperating in a manner
25 that may introduce security (integrity and/or

confidentiality of VDE secured information), process control, and/or software compatibility problems.

Certification validates the identity of VDE installations and/or their components, as well as VDE users. Certification data can also serve as information that contributes to determining the decommissioning or other change related to VDE sites.

support the separation of fundamental transaction control processes through the use of event (triggered) based method control mechanisms. These event methods trigger one or more other VDE methods (which are available to a secure VDE sub-system) and are used to carry out VDE managed transaction related processing. These triggered methods include independently (separably) and securely processable component billing management methods, budgeting management methods, metering management methods, and related auditing management processes. As a result of this feature of the present invention, independent triggering of metering, auditing, billing, and budgeting methods, the present invention is able to efficiently, concurrently support multiple financial currencies (e.g. dollars,

marks, yen) and content related budgets, and/or
billing increments as well as very flexible content
distribution models.

5 ! support, complete, modular separation of the control
structures related to (1) content event triggering, (2)
auditing, (3) budgeting (including specifying no right
of use or unlimited right of use), (4) billing, and (5)
user identity (VDE installation, client name,
10 department, network, and/or user, etc.). The
independence of these VDE control structures
provides a flexible system which allows plural
relationships between two or more of these
structures, for example, the ability to associate a
15 financial budget with different event trigger
structures (that are put in place to enable controlling
content based on its logical portions). Without such
separation between these basic VDE capabilities, it
would be more difficult to efficiently maintain
20 separate metering, budgeting, identification, and/or
billing activities which involve the same, differing
(including overlapping), or entirely different,
portions of content for metering, billing, budgeting,
and user identification, for example, paying fees
25 associated with usage of content, performing home

banking, managing advertising services, etc. VDE modular separation of these basic capabilities supports the programming of plural, "arbitrary" relationships between one or differing content portions (and/or portion units) and budgeting, auditing, and/or billing control information. For example, under VDE, a budget limit of \$200 dollars or 300 German Marks a month may be enforced for decryption of a certain database and 2 U.S. Dollars or 3 German Marks may be charged for each record of said database decrypted (depending on user selected currency). Such usage can be metered while an additional audit for user profile purposes can be prepared recording the identity of each filed displayed. Additionally, further metering can be conducted regarding the number of said database bytes that have been decrypted, and a related security budget may prevent the decrypting of more than 5% of the total bytes of said database per year. The user may also, under VDE (if allowed by senior control information), collect audit information reflecting usage of database fields by different individuals and client organization departments and ensure that differing rights of access and differing budgets limiting database usage can be applied to