



THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

**UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office**

November 21, 2012

**THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM
THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK
OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT
APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A
FILING DATE UNDER 35 USC 111.**

**APPLICATION NUMBER: 60/162,522
FILING DATE: October 29, 1999**

**THE COUNTRY CODE AND NUMBER OF YOUR PRIORITY
APPLICATION, TO BE USED FOR FILING ABROAD UNDER THE PARIS
CONVENTION, IS US60/162,522**

**By Authority of the
Under Secretary of Commerce for Intellectual Property
and Director of the United States Patent and Trademark Office**

M. Tarver

**M. TARVER
Certifying Officer**



COVER SHEET FOR PROVISIONAL APPLICATION FOR PATENT

Assistant Commissioner for Patents
Box PROVISIONAL PATENT APPLICATION
Washington, DC 20231

Sir:

This is a request for filing a PROVISIONAL APPLICATION under 37 CFR 1.53(c).

Docket Number		9982-009-999	Type a plus sign (+) inside this box -	+
INVENTOR(s) APPLICANT(s)				
LAST NAME	FIRST NAME	MIDDLE INITIAL	RESIDENCE (CITY AND EITHER STATE OR FOREIGN COUNTRY)	
SAMUEL LAGERMANN	OVID KLAUS		London, United Kingdom Copenhagen, Denmark	
TITLE OF THE INVENTION (280 characters max)				
System and Method for Simultaneous Display of Multiple Datastreams				
CORRESPONDENCE ADDRESS:				
PENNIE & EDMONDS LLP 1155 Avenue of the Americas New York, NY 10036-2711 (650) 493-4935				
ENCLOSED APPLICATION PARTS (check all that apply)				
☒ Specification	Number of Pages	24	☐ Small Entity Statement	
☒ Drawing(s)	Number of Sheets	17	☐ Other (specify)	
METHOD OF PAYMENT (check one)				
☐ A check or money order is enclosed to cover the Provisional filing fees.			ESTIMATED PROVISIONAL FILING FEE AMOUNT	
☒ The Commissioner is hereby authorized to charge the required filing fee to Deposit Account Number 16-1150.			☒ \$150 ☐ \$ 75	

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☐ No. ☐ Yes, the name of the U.S. Government agency and the Government contract number are: _____

Respectfully submitted,

Signature

William S. Galliani
PENNIE & EDMONDS LLP

 REGISTRATION NO.
(if appropriate)

33,885

Date

October 29, 1999

☐ Additional inventors are being named on separately numbered sheets attached hereto.

Total number of cover sheet pages. 1

PROVISIONAL APPLICATION FILING ONLY

CAI - 230335.1

SurfCast Ex 2066,

Microsoft v. SurfCast IPR2013-00292, page 2

SC0001069

Patent Application

for

System and Method for Simultaneous Display of Multiple Datastreams

5

10

Inventors:

Ovid Santoro

Klaus Lagermann

15

20

25

30

Pennie & Edmonds
1155 Avenue Of The Americas
New York, New York 10036-2711
(212)-790-9090

Attorney Reference No.: 9982-009-999

35

CA1 - 220203 5

Field of the Invention

The present invention relates to methods of presenting information from a variety of sources on a display device. Specifically the present invention describes a graphical user interface for simultaneously organizing the display of data and datastreams from a multitude of sources. In particular, the present invention comprises a graphical user interface which manages the refresh rates of multiple data and datastreams. The present invention is intended to operate in a platform independent manner.

BACKGROUND OF THE INVENTION

The scope of the global communications capacity, comprising fixed link and cellular networks, continues to expand rapidly. The variety and complexity of communication devices increases and the number of users escalates. As a result, users are faced with increasingly complex systems and interfaces with which to manage multiple sources of information. At the same time, society has increased its demands on time and productivity such that users no longer have the luxury of focusing their attention on a single source of data or means of communication. Instead, the norm today is for people to carry out many tasks simultaneously.

As might be expected, these demands have exposed substantial problems in current communications technology. In particular, users are faced with insufficient resources to manage and access the volume and variety of data available to them in an efficient and productive manner. While a variety of tools designed to assist in accessing and managing these resources have been created, these tools remain unsatisfactory. Consequently, users are impeded by the myriad of data sources, each with its own login/password requirements, as well as by slow retrieval times to access the data, leading to unacceptable timescales for many operations.

A particularly acute problem relates to the limited resources available to users and the incumbent delay in accessing data resulting therefrom. In particular, both communications bandwidth and data processing resources are limited such that a user may not access data from multiple sources in real-time or near real-time. For example, under the present art, if a user were to have sufficient communications bandwidth to receive multiple data streams simultaneously, it is unlikely that the user would have sufficient data processing resources to view the data as it is received. In this regard, if a user were to receive multiple broadcast television signals on a personal computer, he would be unable to view the full data stream of

each signal simultaneously because it is unlikely that his computer would have enough television tuners to process and display more than a single signal at a time.

Under the present art, it is also quite likely that the user would lack the bandwidth resources to receive multiple video data streams simultaneously, such that if he were receiving one video data stream, he would be unable to receive a second at the same time due to bandwidth constraints. Thus, the user could not, for example, monitor multiple video data streams of sporting or news events; instead, the user could monitor only one video data stream at a time.

To address such bandwidth resource limitations, the current art only accesses data when the user requests it. As a result, there is an inevitable delay between the user's request for data and the communications device's display thereof. For example, if a user wants to monitor sources of news data on the Internet using current browser technology, the user must continuously and manually request the news data from its source to determine whether the data has been updated. Prior to requesting and subsequently receiving the data, the user has no way of knowing whether the data has been updated. In any case, the user is unlikely to want to refresh the status of each application by manual intervention himself at the respective necessary rates. Additionally, if a user wishes to view two or more webpages simultaneously, he must execute two or more copies of the web-browser program. The act of switching between alternate programs in order to ascertain which have any new material to offer is fundamentally inefficient.

Similarly, the user's access to such data is not in real-time or even near real-time because each time the user wants to view the data, he must request the data from its source and wait for the source to transmit the data. Thereafter, he must wait until his communications device has received and processed the data before it is displayed. For complex data, this can take longer than a minute to occur; and, even for simple data, this process will take many seconds. Thus, the user is denied real-time or near real-time access to the data.

Nor do present art inventions that locally store or "cache" previously accessed data to make the data available to the user more rapidly solve this problem, because this cached data is necessarily old data. The user's communications device must still verify the data's currency with the data source before the system displays the cached data. As a result, the user is denied real-time or near real-time access to updated data.

Similarly, if a user wishes to make two or more simultaneous downloads he has no control over the relative rates at which the respective downloads would occur. So-called "push technologies" attempt to address this problem by organising information from a number of related sources and periodically send it to a user. Whilst this arrangement frees a user from actively participating in the download, the price is that he has little control over the organisation of the data and can only practically handle a small number of such transmissions at any one time. Each transmission is subject to the bandwidth available.

Of course, not all tasks require the same allocation of resource and, correspondingly, not all tasks have equal priorities for a given user. In particular, a user may wish to customize his information environment in such a way that many processes are occurring synchronously yet each is communicating with the user at a rate which is acceptable. For example, a television viewer may wish to know what is being broadcast on several channels at the same time but only care to watch one of them closely. An Internet user may wish to be continually in touch with sources of data from audio, video, chat-room, video-conferencing and e-mail checker utilities but not wish all of them to update at the same frequency; the user would be satisfied merely to see at a glance a recent status of each. Some of these processes, such as chat-room activities entail very little data transmission and can, indeed, be effectively updated on a continuous basis whereas others require a great deal of bandwidth but could usefully be sampled at a lower rate. The current art lacks any technology for controlling the respective refresh rates of several simultaneous datastreams.

At the same time that users are limited by system resources, they are also finding that they have no effective way of managing the multiplicity of available data and data sources. It is difficult both to carry out and monitor two or more different types of data or computing activities because the tools available are confusing, inflexible, and/or otherwise difficult to implement. Users require immediate access to a wide variety of up to date content presented in a flexible, easily customized interface.

In addition to restrictions in the capacity of today's networks, there is very little conformity amongst the data content. A typical communication device, such as a personal computer, television or mobile telephone, comprises a display unit connected to a processing unit that can accept data from many different sources. As described above, the data and/or datastreams which available to such a device are diverse, including, for example, HTML content, e-mail, or streaming audio and video. Correspondingly, the software tools that interpret and process the different data types present each in a different way to the user. From a user's perspective, distinctions between the different data forms could usefully be removed

so that each is viewed in a similar way and such that the current display associated with any datastream gives an immediate indication of its current content. The present reality is different, however. The user must contend with a wide range of icons and program windows that may occupy space on a user's display screen. Another lack of conformity is the different mode of behavior for programs which address different types of data streams. An effort to standardize the ways in which different forms of data are presented to the user would be advantageous. Equally, unification of the way in which those data types are managed would save time and increase user productivity, for productivity is reduced when users must cope with different attributes of different programs and learn distinct paradigms for different types of data.

The nature of the application program windows and their respective icons predominant on the displays of today is restrictive. The application window typically displays the current content or output of only a single program and program icons convey nothing of the program's current state or content. Often, an icon is a static image which is merely characteristic of the program or data represented thereby rather than the program's current state or the data's content. In the present art, there is no intermediate between a window or an icon.

Thus, while a window may be resized as appropriate, it will frequently occupy the full display area, effectively limiting the user to a view of a single program. It may have active areas around its borders such as menu bars, scroll bars, or tool bars designed to allow the user to control aspects of the window's appearance or to set parameters specific to the operation of the program controlling it. Icons, in contrast, offer ease of display when multiple programs are active, but they do not permit viewing or control of the underlying program or data represented thereby. Instead, icons require user intervention, typically in the form of a mouse-click on an icon of interest, to view or control the program or data. Consequently, the user's viewing options are limited to a choice between one presenting very limited information about a multitude of programs and data and one presenting full information, but of only a single program or data source.

The fact that the GUI's of the present art are largely restricted to icons and windows diminishes our capacity to organize, manage, and access the available data. With the Internet representing an ever expanding view of currently accessible global information, the need for flexible information management tools has become crucial. Similarly, with the current expansion of television programming available, including through cable television and satellite broadcasting, the need to manage this audiovisual content becomes acute. The convergence of television programming and computers increases these management needs all the more.

Current computer based devices are centered on bookmarking schemes for managing Internet locations and on complex database technologies for managing specialist information. Neither provides visual immediacy or ease of layout. Bookmark hierarchies are presented as cascading textual menus and database technologies arrange information into rigidly defined structures. The missing capability is a visual categorisation in which an area of the display unit itself becomes the bookmark and the arrangement on the display becomes the categorisation, independent of the type of content.

While the most common way of accessing data sources is via a personal computer, present day technology exists to communicate via a television, handheld computing device, or even mobile telephones in which case Internet content and other data can be displayed as some portion of the screen. We are witnessing a convergence of technologies: televisions are beginning to find application as viewers of non-television data, (for example through use of "Vertical Blanking Interval" technology in which a data signal is inserted into the main video data stream or through set-top boxes providing limited computer and communications functionality); computers are already finding application for the display of movies, real-time data streams, and the playing of audio data; handheld computing devices and mobile telephones are also being enabled to access the Internet and other data sources.

To summarize the current state of the art, display technologies currently lack an interface which is capable of organizing any type of data, presenting such data to the user in a consistent manner and in such a way that all currently open channels are able to indicate their activity on a continual basis and which could run on any device. At its most rudimentary level, the present invention is a simple easy-to-use interface which can be a system for managing and/or otherwise organizing a user's information and interests.

SUMMARY OF THE INVENTION

The present invention provides a simple, easy to use graphical interface facilitating the organization and management of multiple data sources corresponding to a user's needs and interests. The invention comprises a uniform, graphical environment in which a user can access, operate, and/or control multiple data sources on electronic devices. According to the methods of the present invention, a uniform graphical user interface is provided. That is, the interface comprises a small number of elements which are independent of the source of the displayed data or datastream. The graphical user interface is uniform with respect to the source of accessed data and can manage multiple streams of content, entirely of the user's choice. The invention presents video clips, e-mail messages, television shows, Internet sites, application

programs, data files and folders, live video streams, music, radio shows, and any other form of electronic data to the user uniformly and simultaneously, regardless of whether the data is stored locally or available via modem, T1 line, infrared, or any other form of communication. The user's impression of the interface is also independent of the type of electronic device upon which it is implemented.

5

10 The present invention comprises a grid of tiles which resides on the user's computer desktop, Figure 1. Tiles permit "dynamic bookmarking" of information in that each tile is a viewer of a single data source—including streaming data sources—and can be customized with the user's choice of content. A tile may present data in any of a number of ways. For example, in the preferred embodiment, the tiles may present a miniaturized, "thumbnail" view of the underlying data; a "porthole" view of a portion of the underlying data as viewed at full size; a symbol indicating whether the data has been updated since it was last viewed; or a custom interface designed to allow rapid access to the underlying data. The way in which a tile displays content may be independently configured for each tile.

15 Tiles are selectable and live. When a tile is selected, whether by mouse click or otherwise, the tile instantly provides the user with access to the underlying data, whether that data be a hierarchical menuing system leading the user to a different level or tiles, a word processing file stored on a local area network, a spreadsheet stored locally on a user's computer, HTML data on the Internet, or a live television signal. The tiles are live in that each contains real-time or near real-time data.

20 A tile is different from an icon because it provides a real-time or near-real time view of the underlying data in that it contains continually refreshed content. A tile is different from a window because a tile will typically be smaller in size, allowing the user to view multiple tiles simultaneously if desired. The grid can be configured to contain any number of tiles from one to as many as can reasonably fit on the user's display.

25 Significantly, the grid manages the flow of information to the tiles. For example, the grid understands the configuration of the display device and the present allocation of resources. According to priorities that may be applied to individual tiles on a tile by tile basis if desired, the grid manages the refresh rate of each tile. For example, for locally stored word processing or spread sheet files, the user might configure the tiles to refresh only when the underlying data is written to the local hard drive. Similarly, a user might configure tiles containing infrequently updated HTML data from the Internet to refresh at a certain time each day. At the other

extreme, a user might configure an active tile to display a television channel at a refresh rate of 29 frames per second, while at the same time configuring inactive tiles to display different channels at a refresh rate of once every five seconds. In this way, a user could monitor many channels until program content of interest appeared in one of the tiles without the burden of actively refreshing each tile.

5 The grid also understands the interests of the user and acts as a repository for passwords and identifiers to subscription services. In this way, it is not necessary for a user to remember and enter a user name and password to access data requiring such a log-in. In this same vein, Internet sites of interest can be "bookmarked" and stored by the grid, each such site possessing
10 its own tile. The grid is also nestable in that a tile in one grid may point to a separate grid and tiles in the separate grid may point back to the first grid or to yet more grids. If desired, a user can impose a "theme" on a grid and thereby categorize, group, and/or otherwise manage his data sources. In this manner, the user can group tiles relating to particular subject matters, Internet sites, documents, or otherwise in a grid according to some speciality. A tile in such a
15 grid could link to another grid to provide a connection between related categories. Equally, a tile on more than one grid can point to the same information source.

Together, the grids and tile comprise the Application through which a user can simultaneously view data from a multitude of his available sources, including multiple sites on
20 the Internet, receive signals from multiple broadcast channels, and open and view multiple files. In its initial embodiment, the Application may be run through conventional computer operating systems, whereupon it overlays the user's desktop and acts as if it were a "borderless browser". Therefore the Application resides over existing applications without replacing any of them; rather it enables them to be called from the grid itself. The Application, therefore, becomes a
25 graphical file manager in which the content of continuously changing files, i.e., datastreams, is being displayed in real-time or near real-time, depending on the assigned priority. Effectively, the Application replaces the user's desktop with a more visually intuitive dynamic menuing system.

30 The Application may also run on many different devices, for example, set-top box, personal computer, hand-held device. The grid and tiles retain the same characteristics, regardless of operating device. For example, the tiles remain individually configurable and can offer near real-time views of their data content. The Application therefore affords a unification of use of a range of electronic devices.

The Application may be downloaded from a pre-determined web-site and operates in a client-server mode. Users may download preconfigured grids from the predetermined server. A grid configuration "wizard" program which guides a user through a step by step set up of a custom-grid may also be downloaded. Other web hosts are able to deliver content to end-users via the predetermined server. Some basic functions of the grid can be carried out on the predetermined server and provided to the user.

BRIEF DESCRIPTION OF THE DRAWINGS

Additional objects and features of the invention will be more readily apparent from the following detailed description and appended claims when taken in conjunction with the drawings, in which:

Fig. 1 shows a representative embodiment of the Application comprising a grid of tiles as might be depicted on a display screen.

Fig. 2 is a block diagram of a system that, in accordance with the present invention, accepts data in at least one form through at least one port and which additionally displays data to a user.

Fig. 3a shows schematic examples of an icon and a window as are commonly found in computer display systems of the background art.

Fig. 3b shows several tiles as might be found in a typical embodiment of the present invention.

Fig. 4 shows the data structure of the tile object which is controlled by the grid object within the graphical user interface of the current invention.

Fig. 5 shows one embodiment of a Tile in markup language.

Fig. 6, comprising Figs. 6 (a) - 6 (d), shows schematic representations of the GUI which forms the basis of the current invention including a demonstration of the layout of one of its members.

Fig. 7 shows the data structure of the grid object which forms part of the graphical user interface of the current invention.

Fig. 8 shows one embodiment of a grid in markup language.

Fig. 9 shows the hierarchy of software objects underlying the current invention, comprising a grid object, tile objects and files or application software.

Fig. 10 shows a sequence of windows which demonstrate how a grid might be set up for initial use by a "wizard" tool in one embodiment of the present invention.

Fig. 11 shows a schematic relationship between the server and the client device.

Fig. 12 shows a conversation diagram of a series of interactions between a client device and the predetermined server.

Fig. 13 shows how a user, the Server and 3rd party content providers communicate.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Figure 1 shows an example configuration of the graphical user interface of the present invention. A grid 100 consisting of a 3 by 3 matrix of nine tiles demonstrates some of the different contents that tiles can display. Tile 10 points to a database of stock quotes. Tile 20 displays the active folders in an electronic mail utility. Tile 30 displays a portion of an alphabetical list of quoted companies. Tiles 40, 50, 60, 70 and 80 point to websites displaying, respectively, high technology news, electronic goods for sale, categories of business news, items available by auction and the Wall Street Journal. Tile 90 points to a windows-based operating system and displays the currently accessible disc drives.

Figure 2 is a general representation of a data display system 102 within which the present invention may be implemented. System 102 comprises a central processing unit 104, an input device 106, data connection ports 108-1 through 108-N, a display 110 and main memory 112, all connected via bus 116. Residing in the memory 112 is an operating system 120, a file system 122, a cache 126 for temporary storage of data, application programs 128-1 through 128-N and a graphical user interface (GUI) program 124 that is responsible for presenting data on to display 110. Data may enter the system through any one of the ports 108-1 through 108-N which may themselves be connected to a tuner 114, or via a network interface 134 to a communications network. If a tuner 114 is employed, it may channel input from a wireless signal 130 or a cable network 132.

In one embodiment of the present invention, System 102 is a personal computer such as a desktop workstation or a portable notebook computer. In which case the input device 106 may be a keyboard, mouse, trackpad, trackball, or any combination thereof and the display 110 may be a conventional cathode ray tube (CRT) or active matrix flat-screen display. The network interface 134 may then be a connection to the internet or to a local area network.

In another embodiment of the present invention, system 102 is a mobile phone or personal digital assistant and the input device 106 may consist of several buttons on a keypad, a touch-sensitive screen with a touching device or a voice-activated means. In this embodiment, the display 110 is an LCD screen and ports 108 are a means for receiving data from radio signals.

5

In yet another embodiment of the present invention, system 102 comprises a set-top box wherein display 110 is a TV screen or monitor and tuner 114 accepts input in the form of a wireless signal 130 from broadcast transmissions or cable signals from the cable network 132. Input device 106 may be a hand-held remote control apparatus or buttons located on the set-top box or touch-sensitive areas of display 110.

10

As is known to those skilled in the art, a graphical user interface is a computer program that resides in memory 112 of some data processing system and that provides means for presenting output from application programs 128 or content of datastreams from ports 108 on an associated display. Additionally a graphical user interface allows a user to control the arrangement and displayed form of the data content. Usually a graphical user interface permits a user to specify and alter operating parameters of application programs running on the system though, at any given time, a particular application program will take priority. Typical operating parameters that may be changed depend upon the application program but include the number of buttons on a tool bar, the size of the text displayed and the colour of the background.

15

20

The graphical user interface of the current invention permits a user to control the layout of the data content and prioritize each application program running on the system. A novel feature of the present invention is that the data content of any number of the programs can vary in real time and the rate at which the display of each is updated can be controlled by the user.

25

Tile Objects

Conventional graphical user interfaces provide two distinct representations of programs, files and datastreams, as shown in figure 3a. One representation is an icon 320, the other is a window 340. An icon typically occupies only a relatively small proportion of the available display area and is an easily recognisable depiction of the program or file, either through its logo 322 or some characteristic picture with the name 324 of the program visible. An icon can be selected by, for example, a touch screen pointer, a cursor controlled by a mouse with a button or by a keyboard stroke. In response to a further selection operation on an icon, for example, a double-click of a touch screen pointer or cursor the graphical user interface will

30

35

provide a window that can be used to communicate with the program or review the file. The window may occupy a substantial percentage of available screen space. The window 340 commonly can be resized by the user and has a format which contains many active areas around its borders. Examples of active areas include a menu bar 342, a scroll bar 344 and a tool bar 346, each of which may be used to control aspects of the window's appearance or to set parameters specific to the operation of the program associated with it such as text formatting options in a word-processing package, or redirection of a web-browser to its stored home location.

In the present invention, a third representation of programs and files, herein called a tile, is introduced. A tile is associated with a program, file or datastream, in the same way that an icon and a window are. Figure 3b illustrates and some representative tiles. A tile provides an at-a-glance view of the current status of the program or file associated with it but without the large number of active areas associated with windows such as title bar, menu bar and border. Therefore tiles lead to a reduction in clutter on the display screen because many tiles may be displayed simultaneously without overlapping with one another in the way that windows would. Tiles are superior to icons because they give an immediate indication of the current state of the file or program and have a utility functions associated with them, as described below. Another advantage of using tiles is a uniformity of appearance between tiles which correspond to different programs. The display content of a tile will differ from application to application though its size and format will not.

The tile data structure 500 is shown in Figure 4. It is important to understand that the tile itself is an image that at any given instant is resident on the file system. This image is separate and distinct from the application program or file associated with the tile. The tile data structure 500 comprises two addresses: a Tile Address 502 that defines the location on the file system where the tile image is stored; and a Target Address 504 that is the location at which the file or application program associated with the tile can be found. Additionally, the tile data structure contains a Name 506 that may be displayed on the tile in certain circumstances. Tiles of the present invention may be assigned at least seven functions, including but not limited to: an initialisation function 508 that is responsible for establishing connection with the target address 504; a Refresh function 510 that handles updates to the tile image stored at the tile address 502; a Screen-size function 512 that stores the dimensions of the display area filled by the tile upon receipt of a request; an Alarm function 514 which permits the tile to display an alarm or warning when the application program associated with the tile encounters a designated event; An Onmouseover function 518 and an Onmouseout function 520 which, respectively control the behaviour of the tile when a selection tool such as a mouse-controlled cursor is

placed on and off the tile; and a Toolbar function 522 which may permit an array of special buttons to appear on or adjacent to the tile for the purposes of adjusting properties of the tile. In one embodiment, a tile is configured so that a right-click of a mouse button while the cursor is over the tile would activate the tool-bar function.

5 Referring to figure 3, there are a number of different possible representations that can be associated with a tile. Tile 302-1 displays a picture. The content of tile 302-1 is typically a miniaturised representation of a graphic or still-frame from a datastream. Tile 302-2 displays a portion of a text document; in this sense the tile functions as a transparent panel placed on top of a document, thus permitting a portion of the document to be displayed. Tile 302-3 displays another grid object. Tile 302-4 has been configured to link to an electronic mail program. The
10 alarm setting has been configured so that the tile displays the message "New Mail!" and an envelope when an unread message has been received. Tile 302-5 displays a name, "FM 101", denoting the title of a broadcast signal, in this case audio, that is associated with the tile. Tile 302-6 displays a "thumbnail" of the document viewed in window 340. A "thumbnail", as used
15 herein, is a miniaturised representation of an image which retains sufficient characteristics to permit easy recognition of the image. For example, if the document displayed in window 340 were a multipage document, tile 302-6 may display a thumbnail of the first page of the document.

20 In a preferred embodiment, clicking on a tile causes it to immediately refresh its contents. If Tile 302-1 were selected, the most recent frame of the video stream is displayed by the tile, or, if the picture were obtained from a static graphic file, the most recent version of the file is displayed by the tile. If Tile 302-2 or Tile 302-6 were selected, the most recent version of the document would be displayed.

25 In a preferred embodiment, double-clicking a tile will cause that tile to occupy a substantial fraction of the whole display area. In this embodiment, double clicking Tile 302-1 or 302-6 causes the image to be expanded to fit the resized tile. If Tile 302-2 were double-clicked, as much of the document as could be displayed in the enlarged tile becomes visible.
30 If Tile 302-4 were clicked, the mailbox window of the e-mail utility is sufficiently enlarged to allow new messages to be read. If Tile 302-5 were double clicked, the audio stream becomes audible over the appropriate channel of the system.

35 In one embodiment of the present invention, the tile is itself a document created in a markup language (Figure 5) and is suitable for display in a web-browser. In this embodiment, a tile occupies a position in a table defined with elements of mark-up language which will be

familiar to one skilled in the art. Tile-specific attributes are introduced which control the way in which a web-browser displays the tile. In the example in Figure 5, the Tile has a clickable map, i.e., separate areas of the tile produce separate results when clicked. Also, this tile has a toolbar, which in one embodiment may appear if the mouse is right-clicked when the cursor is over the tile.

5

Grid Object

In the preferred embodiment, a grid of tiles replaces the functionality of the computer "desktop" utilised by many modern computer operating systems. Whereas a desktop is typically populated by static icons and tool bars, a grid of tiles instead presents to the user an array of snapshots of current programs and files. The essence of the grid is that its content is dynamic and informative. Icons are inherently limited in the information that they can present. Each tile on the grid can show the current status of the data or datastream associated with it. The fact that a tile may refer to a separate grid permits nesting of grids and consequently a hierarchy of organised information sources.

15

As shown in Figure 6, the graphical user interface of the present invention comprises a grid object 300 divided up into a number of rectilinear tiles 302. Each tile is associated with a separate application program or file 128, any one of which may be another grid object 300. Such a hierarchical structure permits a user to organise programs and data through the graphical user interface. For example, separate categories of information can be displayed on separate grids.

20

In the preferred embodiment, the grid feature permits a regular layout of tiles on the display screen, as depicted in figures 6(a) - (d). There are no gaps between the tiles, the tiles are not permitted to overlap and the whole grid is covered by tiles. Figure 6(a) shows one embodiment in which all tiles are the same size and are presented in an array comprising M rows and N columns. There is no particular requirement that the arrangement consists of more than one row and more than one column. On a mobile telephone or smart phone, for example, a single row of tiles may be apposite.

30

Figure 6(b) shows an arrangement in which there is a unit tile size, that of tile 302-1-1, but Tile 301-1-2 and Tile 302-N-1 have been configured to occupy regions of the grid equal to exact numbers of the unit tile size. Such an arrangement may be useful and important if one or more datastreams is of particular interest.

35

Figure 6(c) shows an arrangement of tiles in which Tile 302-N-1 is associated with a further copy of the grid program. The lower half of the figure shows an enlarged perspective of that tile.

Figure 6(d) shows an arrangement of tiles in which different application programs associated with three of the tiles are depicted. Tile 302-3-2 links to a file viewer displaying a specific file; tile 302-N-1 presents streaming video; Tile 302-N-M depicts a page of information on the world wide web.

Figure 7 shows a schematic data structure of the attributes of one embodiment of grid object 400. The grid may have a parent grid whose parent grid address 402 is known. (For example, grid 400 may be the grid associated with Tile 302-N-1 of Figure 6(c), in which case its parent is Grid 300.) The grid itself has an address 404 that specifies its location within the file system of the system's main memory. The grid has associated with it several utility programs: a configuration wizard 406 that may be called by the user when setting up a new grid; a tile creation function 408 utilised by the configuration wizard when initialising new tiles; a tile annihilation function 410 utilised in case of error or when resizing the grid. Additionally, the grid object stores the number of rows 412 and the number of columns 414 of tiles that are present. The grid also stores a tile list 416 containing attributes of each respective tile. In particular, the address of each tile, its priority and its refresh rate are stored by the grid program. The grid also stores other attributes of tiles such as their respective positions on the grid as given by their column and row number. The priority of a tile may be used to determine its refresh rate in one embodiment of the present invention. A password feature may be built into each tile where permitted to access the tile's content.

By using the native attributes of a tile, a user may specify a presentation of the grid, consisting of its dimensions, (i.e., the number of tiles to display and their arrangement), and the programs or files to be associated with each tile. A single grid, composed of multiple tiles, may therefore present a number of information sources simultaneously.

When two or more tiles connect to sources of data available over a network, communication must be established in such a way that the rate at which updated data is transmitted to the grid can be controlled. In practice, for an embodiment of the Application which resides on a user's computer, a flow control protocol such as TCP is required. In this way, each tile can communicate with the remote datastream to which it is linked and a determination can be made of available bandwidth at the time of data transfer. Alternatively,

in a client-server mode, flow control is not necessary because communication with the server suffices, as is described below.

In one embodiment of the present invention, the grid is a document created in a markup language such as HTML, SGML or XML (Figure 8) and is therefore suitable for display via a web-browser. In this embodiment, the addresses of the parent grid, the grid itself and each of the tiles is then expressed as a Universal Resource Locator (URL). The various functions controlled by the grid are accessible through function calls devised according to methods familiar to one skilled in the art. For example, "dynamic HTML", java applets or simple CGI-scripts could provide the technological basis for enabling various grid utilities.

The overall hierarchy of a graphical user interface embodied by the present invention is summarised in Figure 9. The Grid 300 is the top level functionality to which application programs are subordinate. The grid replaces the functionality of a user's computer desktop and offers more features. The user can control their display in such a way that the tiles 302 present simultaneous information content from a plurality of sources. The grid controls the layout and priorities of the tiles. Each tile is associated with a data stream or application program and allows a choice of displayed images. A tile may be associated with another grid and thereby permit a "layering" of information hierarchies.

Grid Configuration Wizard

In one embodiment of the present invention, the set up of a particular grid is achieved through a grid configuration program ("wizard") that is downloaded to the display device from a remote site. The grid configuration program permits a user to define and install one or more grids on the client system. One embodiment of the grid configuration wizard is shown in Figure 10. The user is offered the choice of preconfigured, or "standard" grid configurations, selected from a list. Examples of such grids include grids themed by content or by type of data. There might be sport-related grids or grids whose content is video-based. Additionally, the user will be permitted to configure "customised" grids in which each tile can be taken from a list of predefined samples or can be initialised according to the user's wishes. In a preferred embodiment, a grid can be filled by the "drag and drop" technique in which a selected document is moved on to the display area of the grid configuration program and automatically becomes a tile.

Client-Server Interaction

In one embodiment of the present invention, figure 11, the user at client device 700 interacts with server software on a predetermined server 702. The server stores locally a profile comprising user-specific content 706 that can feed customized data to the user. For example, the user may store pre-defined grid configurations on the predetermined server. Additionally, passwords for specific web-sites might be stored along with the user's profile. A grid generator 704 on the predetermined server creates a grid of tiles according to user-specified content. Each tile has been created on the server by producing an image from the location specified. For example, Tile Creator 708-1 produces a tile from Content 710-1. Thus, when a user logs on to the predetermined server, perhaps through a conventional web-browser, a grid of tiles is downloaded to the user's system.

The client-server embodiment permits certain key features to be incorporated. Once individual users may be differentiated, so may the devices they use. Therefore, the tile-content that the predetermined server delivers can be customized according to the rendering device, Figure 12. Each time a user logs on to the predetermined server, a "session" is initiated. The server verifies the user's identity. The client registers one or more resources, such as connection bandwidth, cost per transmission unit and properties of local playback device from which the server identifies the client device type, for example, set-top box or personal computer. The server retrieves settings specific to the user, if appropriate. Details of a session, as defined by tile content and priorities can be held over from one session to another, both for the purposes of permitting a user to continue with ongoing work and in order to protect against the adverse consequences of abnormal disconnections. Additionally, targeted advertising and messaging can be delivered to subsets of users via the predetermined server. Upon completion of these steps, the Server knows the client characteristics and is able to distribute bandwidth available to the client among multiple content servers. In this example, if the client has an incoming bandwidth of say 56Kbits/second, and he is requesting datastreams from three sources with equal priority, then the Server will respond for each request that a bandwidth of $56/3 = 18.7$ Kbits/sec is available for each stream.

Having rendered a grid and delivered it to the client device, the user can request datastreams. Any number of datastreams can be requested at once, the corresponding stream request information which defines parameters for each stream is communicated to the server.

In this example, the Server component may reside locally on the client machine (in which case the Server is known as the "Resource Manager", or it may be a remote Surfcast server.

Intensive operations on each displayed tile are also channeled through the predetermined server. For example, refresh operations on a tile, generate a refresh request that is sent to the predetermined server. Similarly, requests to thumbnail a given image can be carried out, by request, on the predetermined server and the resulting compressed image transmitted to the user.

- 5 In a preferred embodiment, aspects of a user's profile are transmitted to third parties so that the third parties may then communicate tiled data directly to the user. For example, a user's custom grid may contain a tile that points to a third party web-site. The predetermined server notifies the 3rd party that the user requires tiled data. The third party then permits the tiled
10 representation of its web-site to be transmitted directly to the user, Figure 13.

15

20

25

30

35

What is Claimed is:

- 1 A computer program for organizing the simultaneous communication with a plurality
of datastreams in order to present dynamic information content, such that
any of said datastreams corresponds to any available data type;
5 said computer program displays an array of tiles;
each tile of said array is associated with one of said datastreams;
Each said tile has a refresh rate associated with it;
information associated with each said tile is updated by automatically retrieving
data from said datastream associated with said tile at a rate determined by said
10 refresh rate assigned to said tile.
- 2 The computer program of Claim 1 wherein said refresh rate assigned to said tile is
individually configurable for each said tile.
- 15 3 The computer program of Claim 2 configured to create an unrestricted number of said
tiles.
- 4 The computer program of Claim 3 such that each said tile in said array of tiles has the
20 same shape and size as any other said tile and no one of said tiles overlaps any other
said tile.
- 5 The computer program of Claim 3 in which one of said datastreams is a further instance
of said computer program.
- 25 6 The computer program of Claim 3 wherein a separate refresh rate is assigned to each
said tile from a priority associated with each said tile.
- 7 The computer program of Claim 3 configured to assign a status to each said tile where
30 said status defines one of two states, "selected" and "unselected".
- 8 The computer program of Claim 3 configured to interrupt said refresh rate associated
with one of said tiles and cause said tile to depict a visual indication of the type of said
data associated with it.
- 35

9 The computer program of Claim 3 configured to interrupt said refresh rate associated with one of said tiles and cause said tile to depict a visual indication of the title of the content of said datastream associated with it.

10 The computer program of Claim 3 configured to interrupt said datastream associated with one of said tiles and cause said tile to depict a thumbnail view of the data associated with it.

11 The computer program of Claim 3 configured to interrupt said datastream associated with one of said tiles and cause said tile to depict a static representation of a portion of the data associated with it.

12 The computer program of Claim 3 configured to associate with one of said tiles an encoded password specific to the data source associated with it.

13 The computer program of Claim 3 configured to display a frame of content of video signal in one of said tiles.

14 The computer program of Claim 3 configured to display a portion of a file of textual content in one of said tiles.

15 The computer program of Claim 3 configured to display a portion of a file containing graphical content in one of said tiles.

16 The computer program of Claim 3 configured to play an audio signal associated with one of said tiles.

17 The computer program of Claim 3 wherein said datastream is a broadcast TV or a CAT signal.

40 A method for presenting data from a plurality of sources on a visual display, said method comprising the steps of:

partitioning said visual display into an array of tiles;

associating a data source with each tile of said array of tiles;

5 assigning a refresh rate to each said tile;

updating information associated with each said tile by automatically retrieving data from said data source associated with said tile at a rate determined by said refresh rate assigned to said tile.

10 41 The method of Claim 40 in which the number of said tiles in said array of tiles is specified.

42 The method of Claim 40 in which each said tile in said array of tiles has the same shape and size as any other said tile and no one of said tiles overlaps any other said tile.

15 43 The method of Claim 40 further comprising the step of determining said refresh rate for each said tile from a priority associated with said data source.

20 44 The method of Claim 40 further comprising the step of assigning a status to each said tile wherein said status defines one of two states, "selected" and "unselected".

25

30

35

60 A system comprising a processor, display unit, input device and readable memory, a set of executable instructions residing in said memory, wherein said set of executable instructions comprises:

instructions to control simultaneous communication with a plurality of datastreams:

5 instructions to arrange for the display of an array of tiles;

instructions to associate each tile of said array of tiles with one datastream of said plurality of datastreams;

instructions to request retrieval of data associated with each said datastream;

10 instructions to assign a rate at which said retrieval of said data associated with each said datastream is requested.

61 The system of Claim 60 encompassing at least one network data connection.

62 The system of Claim 60 encompassing at least one tuner device.

15 63 The system of Claim 60 where said memory, said input device and said processor reside on a mobile telephone.

64 The system of Claim 60 where said memory, said input device and said processor
20 reside on a set-top box for use with a television.

65 The system of Claim 60 where said memory, said input device and said processor reside on a personal computer.

25 66 The system of Claim 60 where said memory, said input device and said processor reside on a personal digital assistant.

67 The system of Claim 60 in which the number of said tiles in said array of tiles is specified.

30 68 The system of Claim 60 in which each said tile in said array of tiles has the same shape and size as any other said tile and no one of said tiles overlaps any other said tile.

- 69 The system of Claim 60 in which one of said datastreams is a further instantiation of said set of executable instructions.
- 70 The system of Claim 60 wherein said set of executable instructions assigns a separate refresh rate to each said tile from a priority associated with each said tile.
- 5 71 The system of Claim 60 further comprising instructions to assign a status to each said tile where said status defines one of two states, "selected" and "unselected".
- 72 The system of Claim 60 further comprising instructions to interrupt retrieval of data associated with one of said tiles and cause said tile to depict a visual indication characteristic of said data associated with it.
- 10 73 The system of Claim 60 further comprising instructions to interrupt retrieval of data associated with one of said tiles and cause said tile to depict a visual indication of the name of said datastream associated with it.
- 15 74 The system of Claim 60 further comprising instructions to cause each said tile to depict a thumbnail view of the data associated with it.
- 20 75 The system of Claim 60 further comprising instructions to associate with one of said tiles a password specific to the data associated with it.
- 76 The system of Claim 60 further comprising instructions to display a frame of content of a video signal in one of said tiles.
- 25 77 The system of Claim 60 further comprising instructions to display a portion of a file of textual content in one of said tiles.
- 78 The system of Claim 60 further comprising instructions to transmit a datastream of audio content.
- 30 79 The system of Claim 62 further comprising instructions to display a frame of content of broadcast TV signal in one of said tiles.
- 35 80 The system of Claim 61 wherein one of said datastreams is transmitted via a server to the network data connection.

81 The system of Claim 61 wherein one of said datastreams may be a page of material on the world wide web, transmitted via the network connection.

5

10

15

20

25

30

35

- 24 -

CA1 - 220203 5

Tile 10

Tile 20

Tile 30

Tile 40

Tile 60

Tile 50

Tile 70

Tile 80

Tile 90

Grid 100

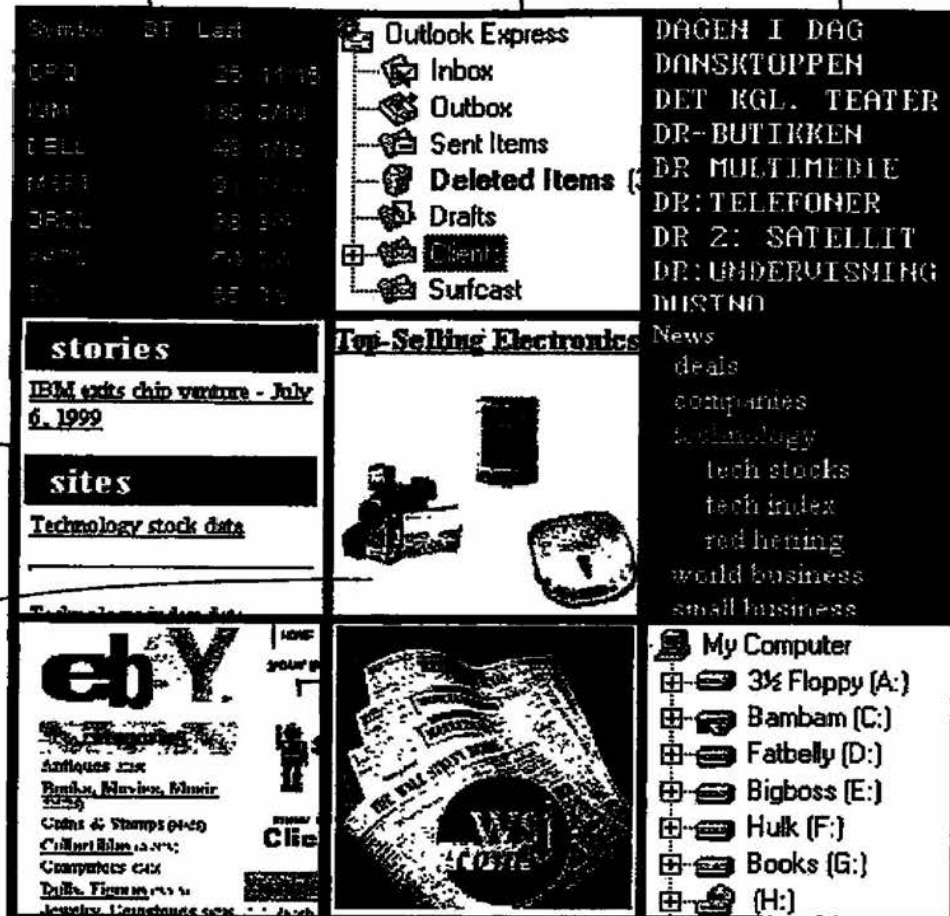


Figure 1.

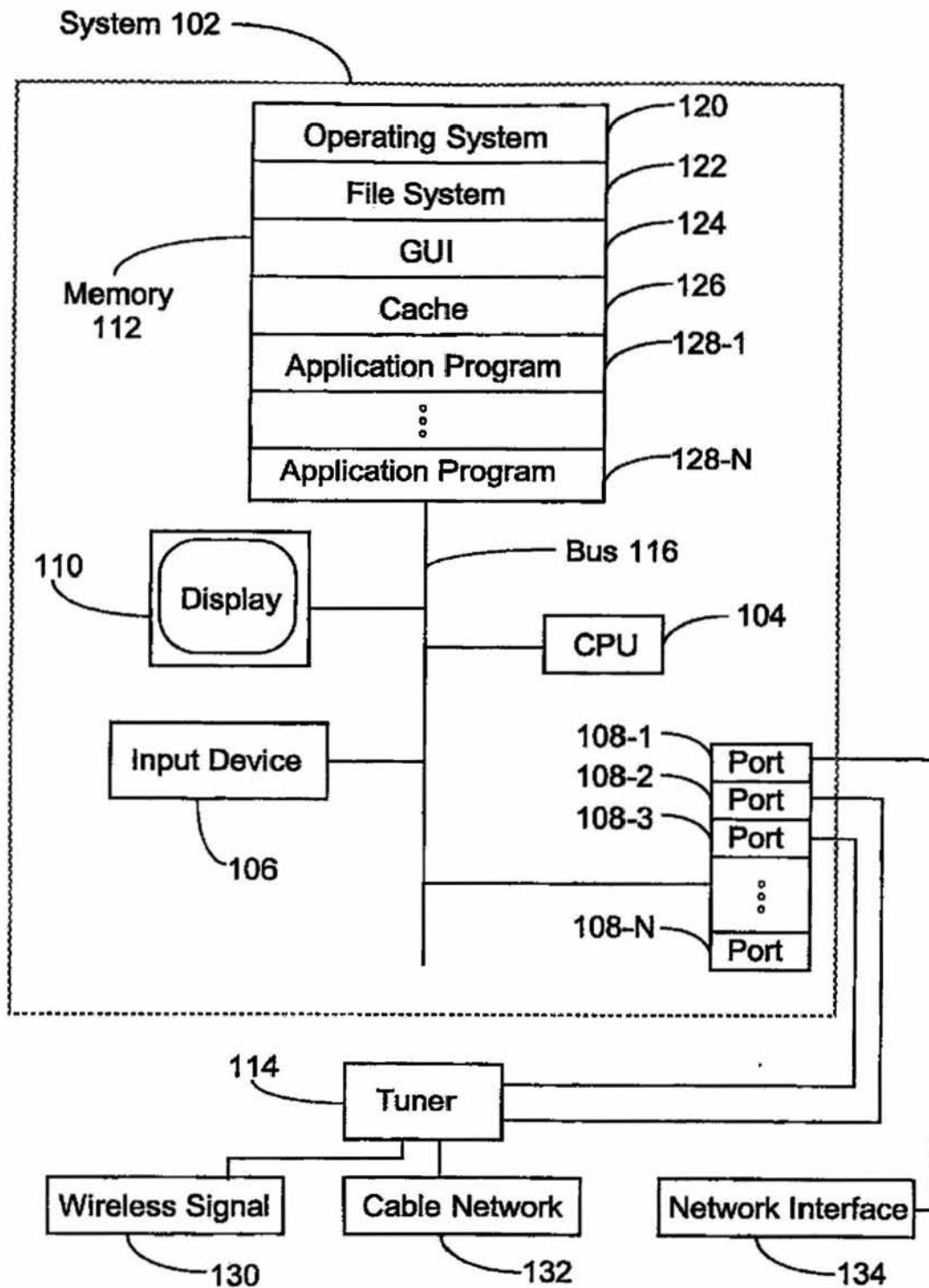
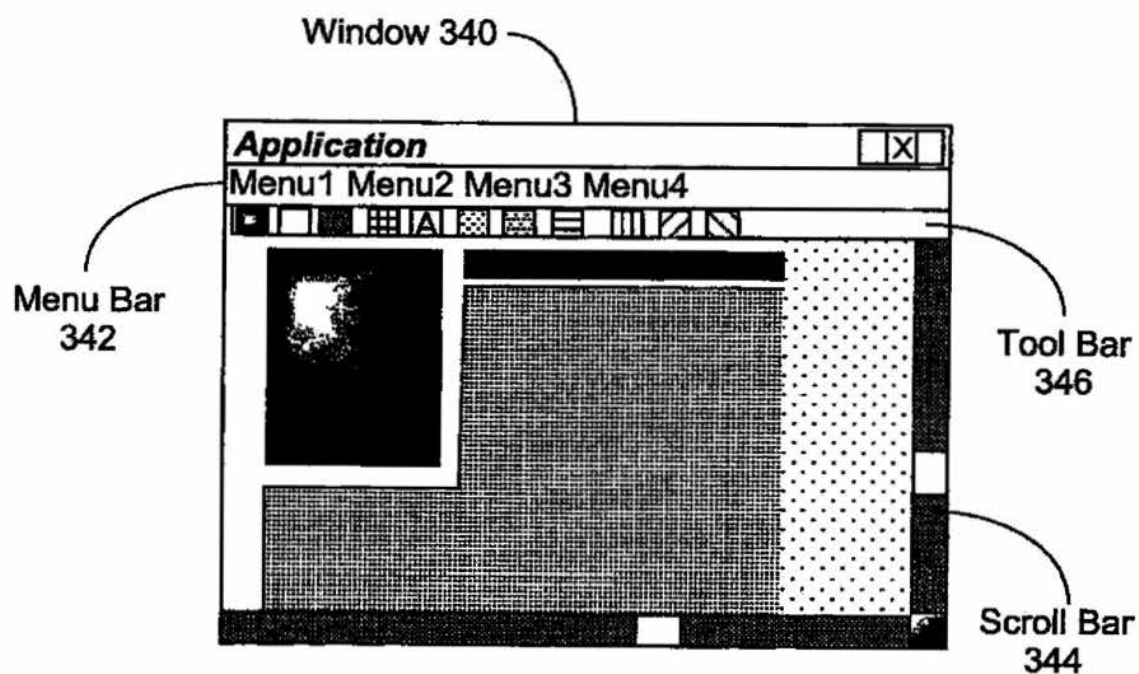


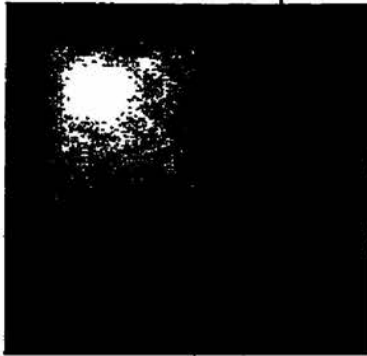
Figure 2.



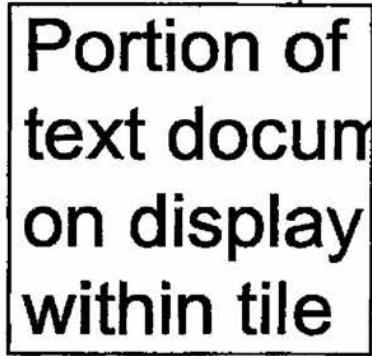
43 3a.

Tiles 302

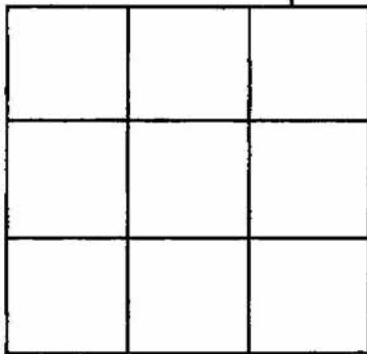
302-1



302-2



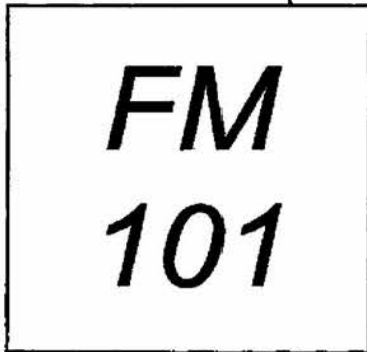
302-3



302-4



302-5



302-6



Fig. 3b

Tile data Structure 500

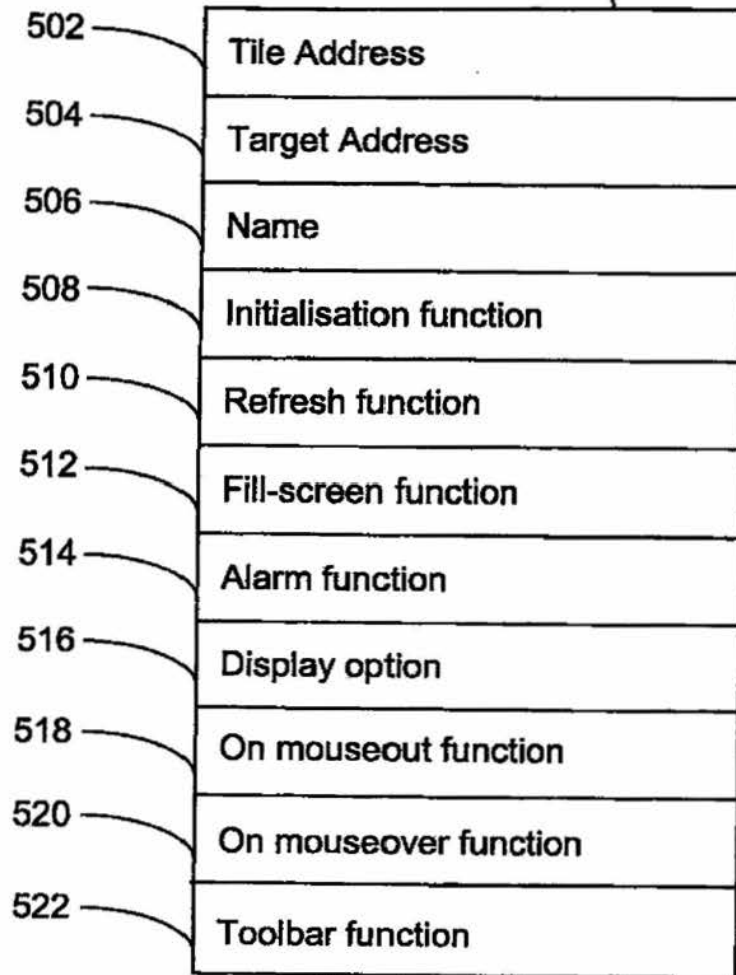


Figure 4.

```

<HTML>
<HEAD>
<TITLE>Surfcast Grid Example</TITLE>
<META NAME="Resource Manager" CONTENT="RESMGR.EXE">
<META NAME="Author" CONTENT="Surfcast, Inc.">
</HEAD>
...
<BODY BACKGROUND="surfback.gif">
<TABLE>
<TR>
<TD><A
HREF="http://www.somewhere.com/sometarget.html"
    TARGET="new"
    ONMOUSEOVER=action1(arg1)
    ONMOUSEOUT=action2(arg2)
    REFRESH=timeout(200)

SOURCE="http://www.surfcast.com/tilelibrary/tile2.til">
</TD>

<TD><A
HREF="http://www.somewhereelse.com/someothertarget.html"
    TARGET="new"
    ONMOUSEOVER=action3(arg3)
    ONMOUSEOUT=action4(arg4)
    REFRESH=timeout(500)

SOURCE="http://www.camelot.castle/swords/excalibur.til">
</TD>

<TD><A HREF="local/document.htm"
    TARGET="new"
    ONMOUSEOVER=action1(arg3)
    ONMOUSEOUT=action2(arg4)
    REFRESH=timeout(0)
    SOURCE="local/documents/somedocument.til">
</TD>

</TR>

</TABLE>
</BODY>
</HTML>

```

Figure 5.

CAI - 229840.1

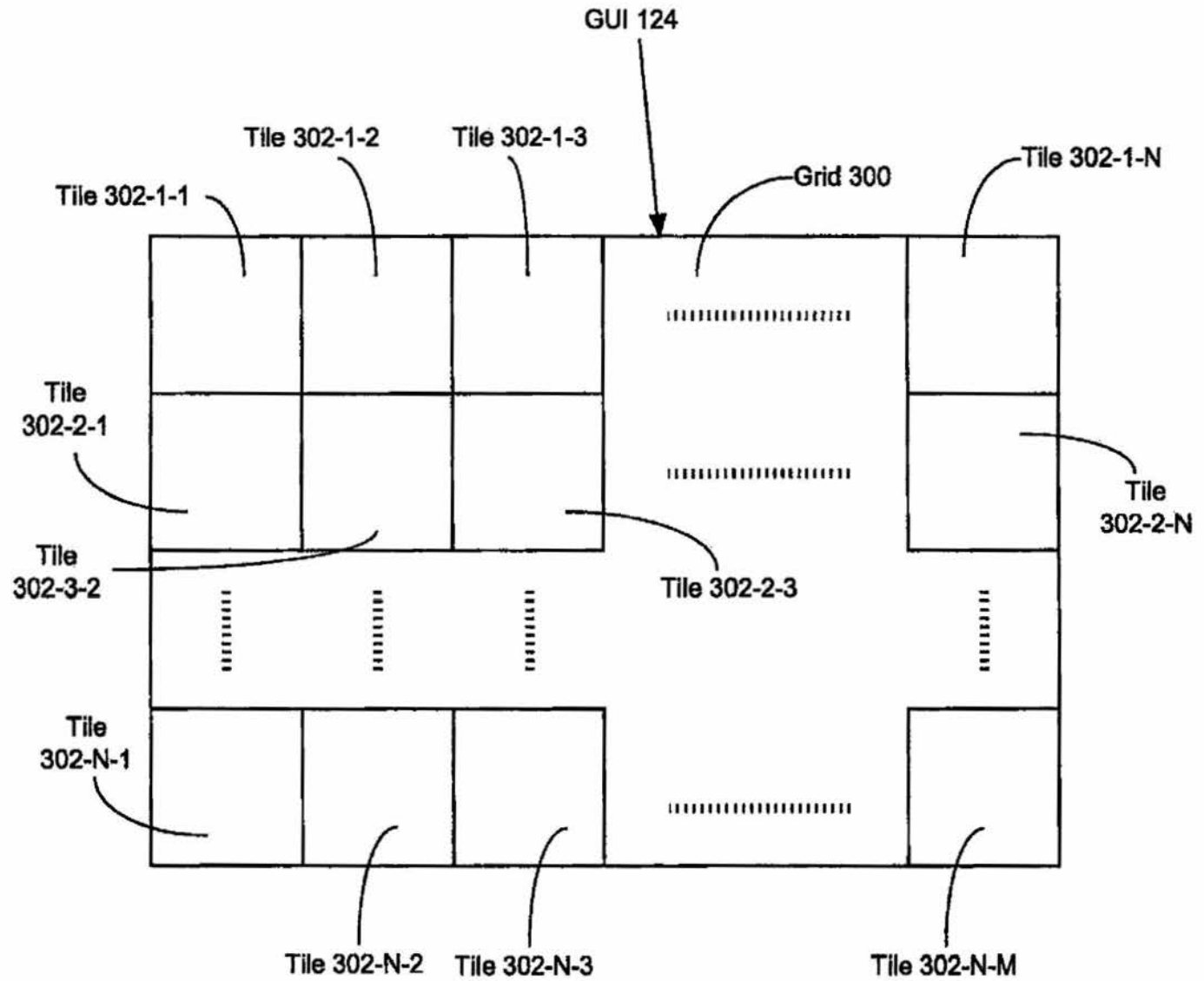


Fig. 6a)

FIG. 124

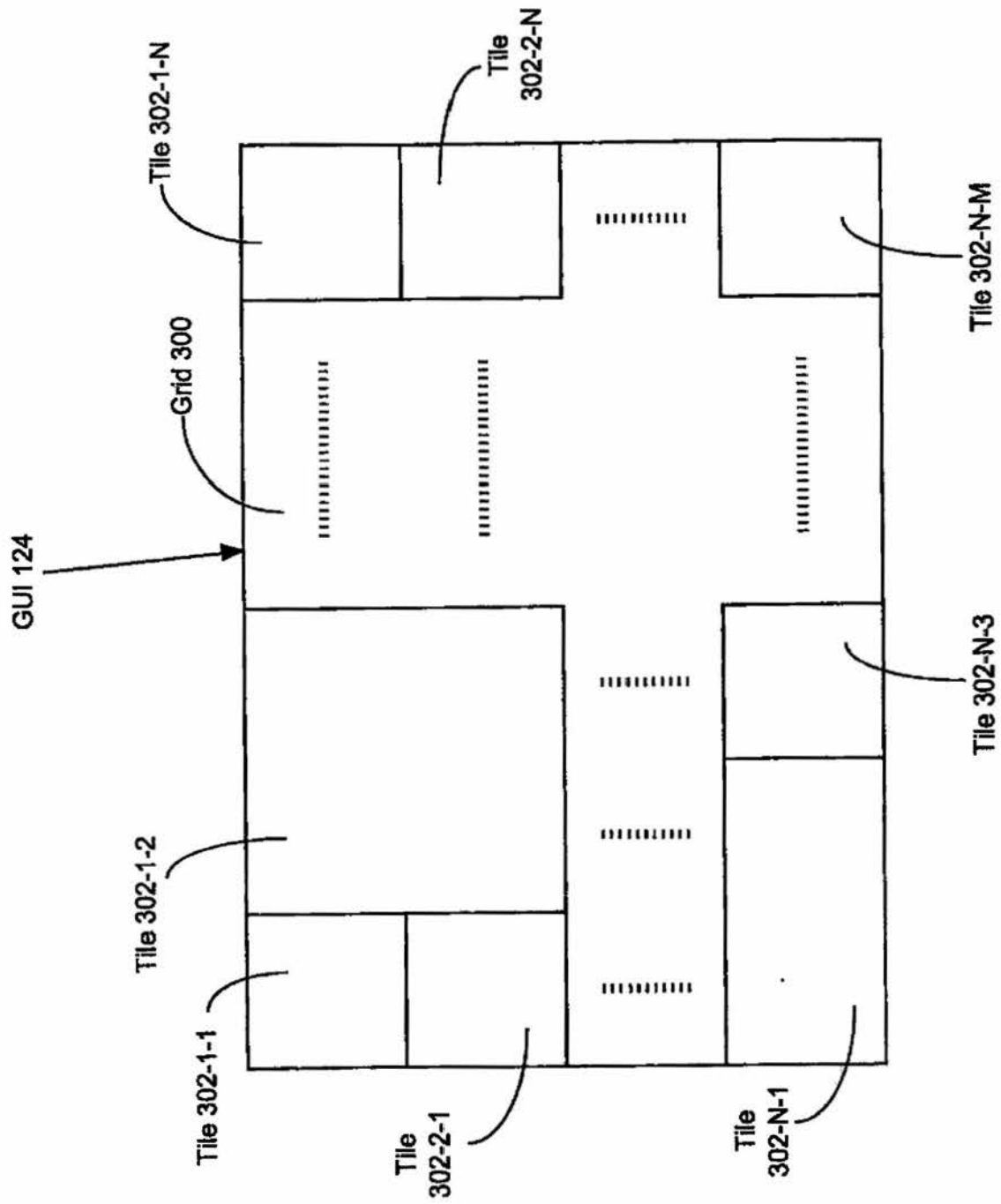


Fig. 124

60207-2563703

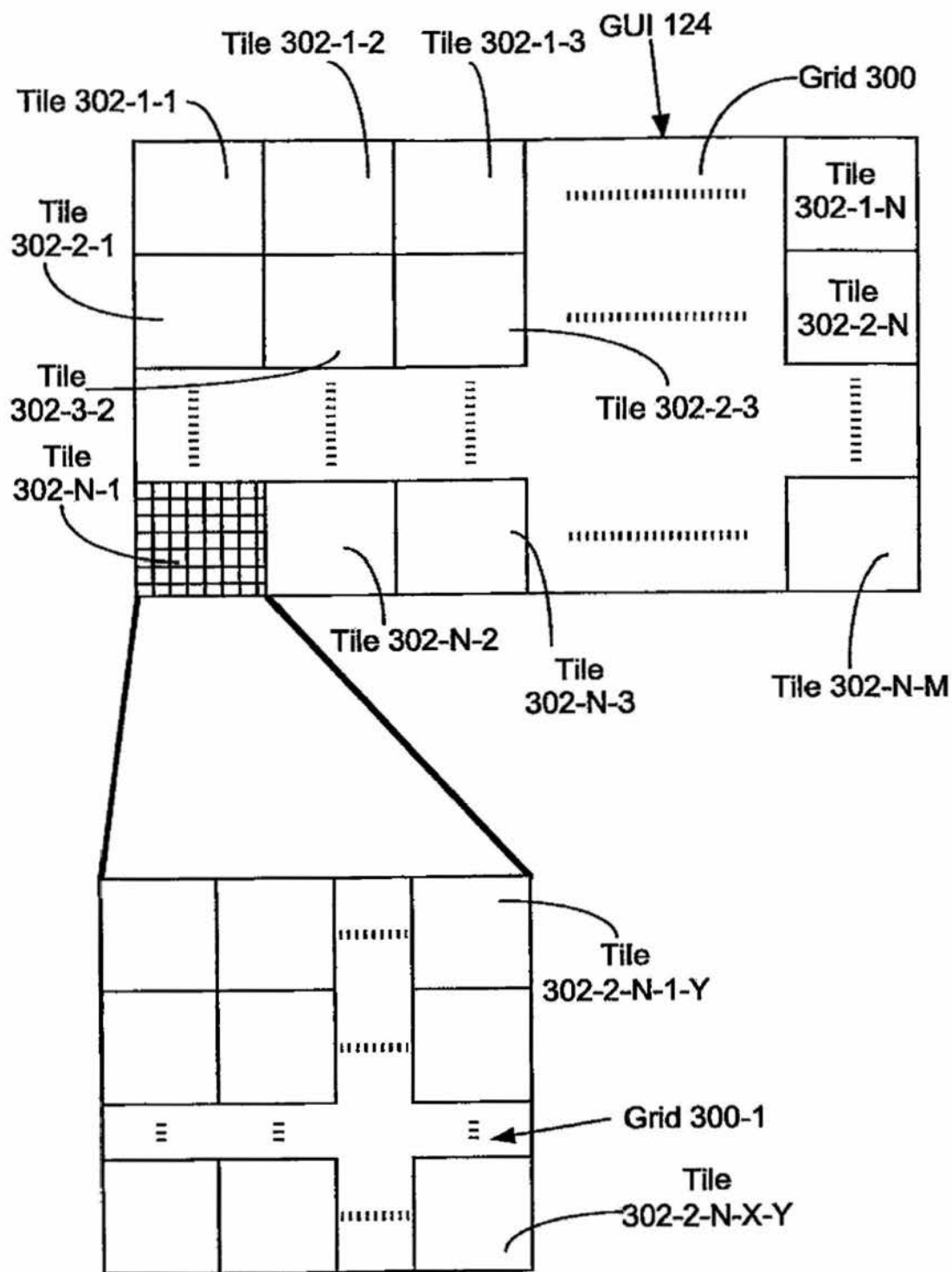


fig. 6c

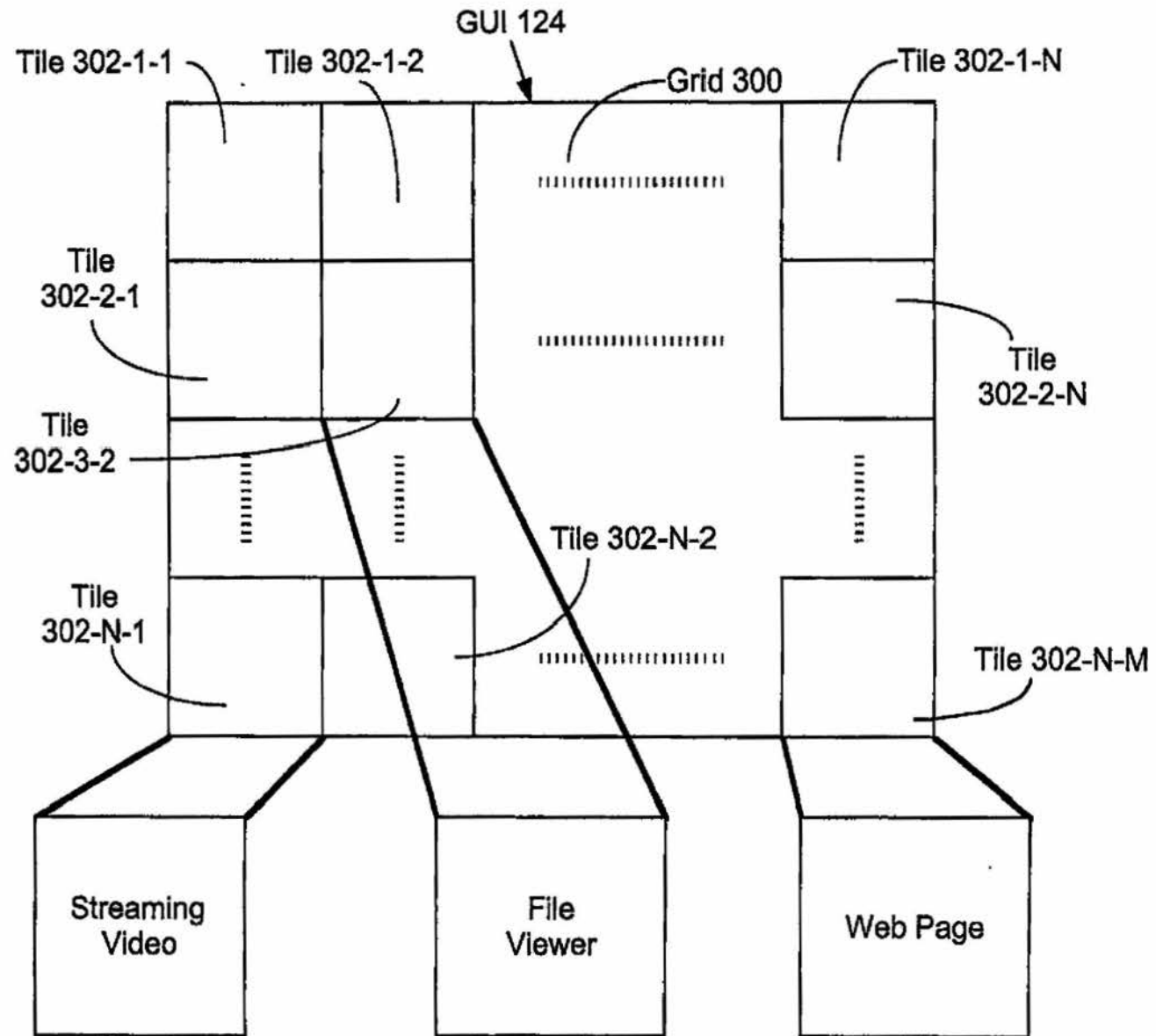


Fig. 6d)

Grid data structure 400

402	Parent Grid (address)						
404	Address of self						
406	Configuration Wizard						
408	Tile creation function						
410	Tile annihilation function						
412	Number of rows						
414	Number of columns						
416	Tile list	Address	Refresh rate	Position	Priority	Password	...
		Address	Refresh rate	Position	Priority	Password	...
		Address	Refresh rate	Position	Priority	Password	...
		Address	Refresh rate	Position	Priority	Password	...
		...	...	...	...	...	...

Fig 7.

<TD><A HREF="tile2_target.htm"
TARGET="new"
ONMOUSEOVER=action1(arg3)
ONMOUSEOUT=action2(arg4)
REFRESH=timeout(500)
SOURCE=<http://www.camelot.castle/swords/excalibur.til>
CLICKMAP= { 0, 0,50,50, clickfunction1(clickargument1),
51, 0,50,50, clickfunction2(clickargument2),
0,51,50,10, clickfunction3(clickargument3) }
TOOLBAR={"local/toolbars/radio.tool" PLACEMENT="bottom" }
ALARM="alarms/condition_rain=TRUE, condition_weekend=FALSE,
alarmaction=blow_the_horn">
</TD>

Fig. 8.

CA1 - 229839.1

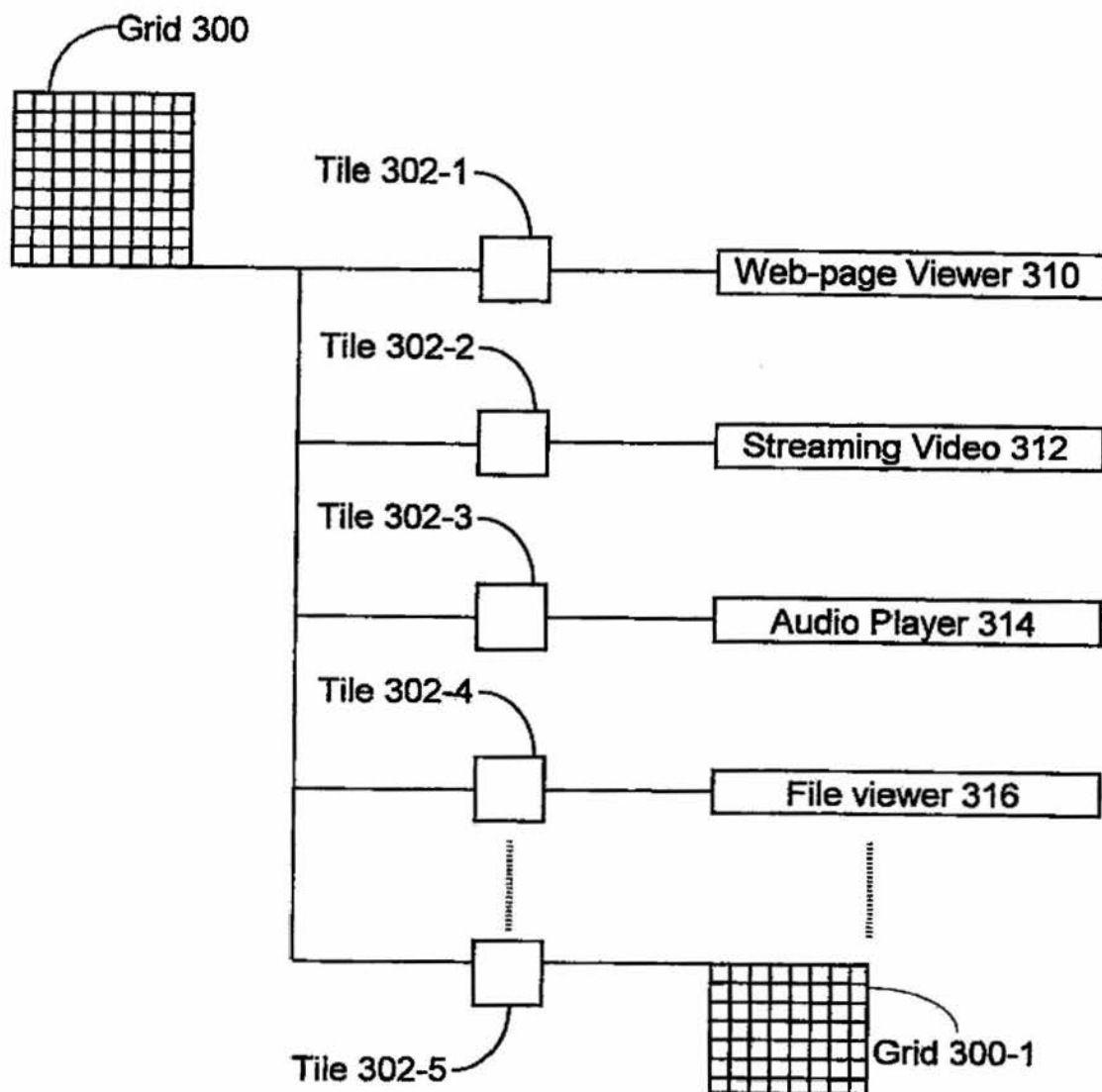


Fig. 9



Grid Wizard

☒ Customized Grid
 ☐ Minimum Grid
 ☐ Standard Grid

Next
 >>



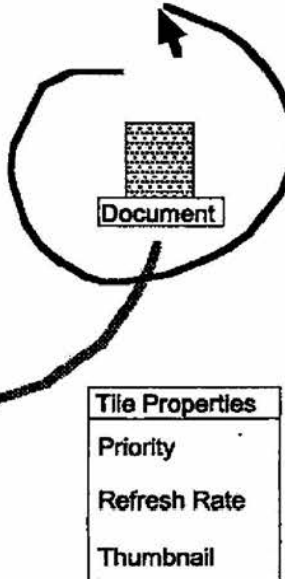
Grid Wizard

Back
 <<

Next
 >>

Weather
 Sports
 News
 Stocks

Tile Settings



Document

Tile Properties
 Priority
 Refresh Rate
 Thumbnail



Grid Wizard

Name Grid:

Store Grid

Launch Grid? Yes No

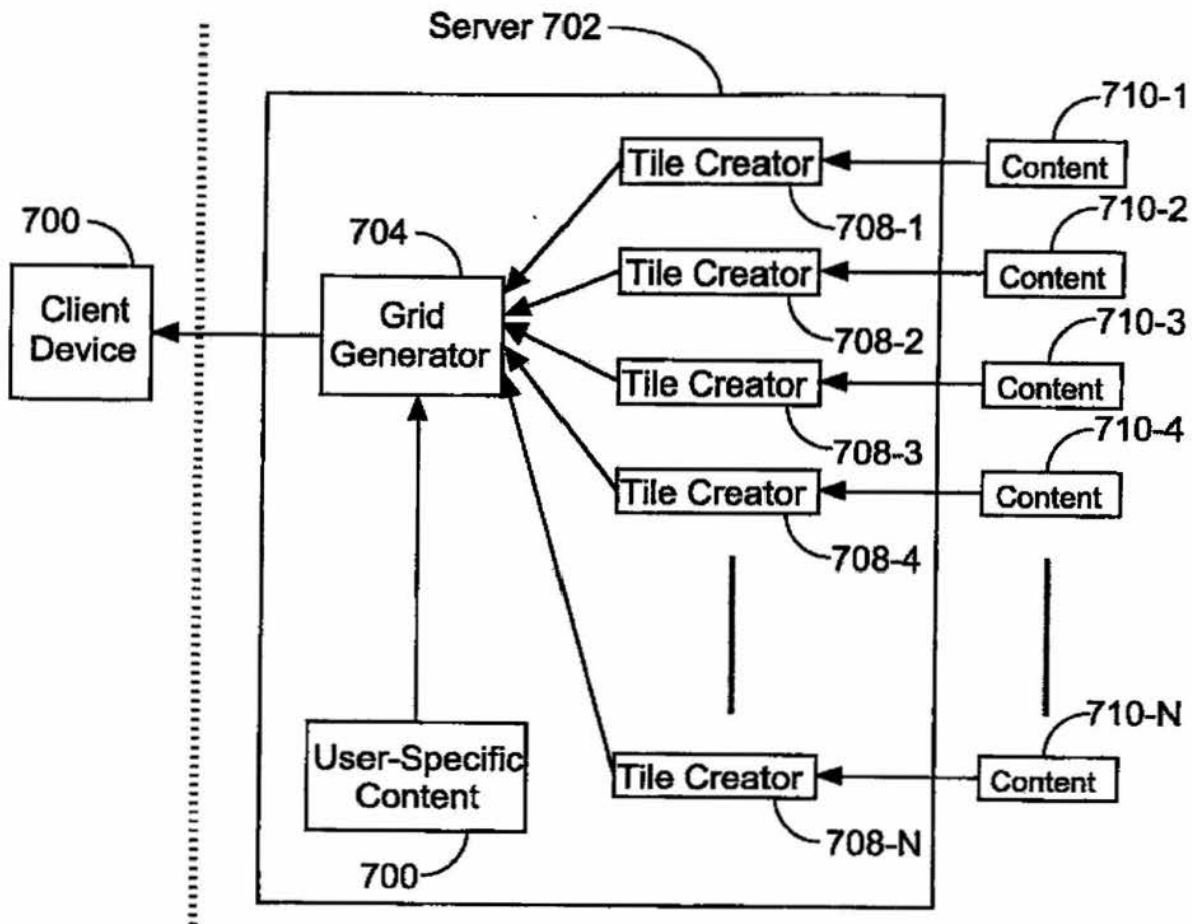
Back
 <<

Finish
 >>

27.10.

Case provided by USPTO from the IFW Image Database on 11/15/2012

SurfCast Ex. 266,
 Microsoft v. SurfCast IPR2013-00292, page 40
 SC0001107



A5-11

700

Server 702

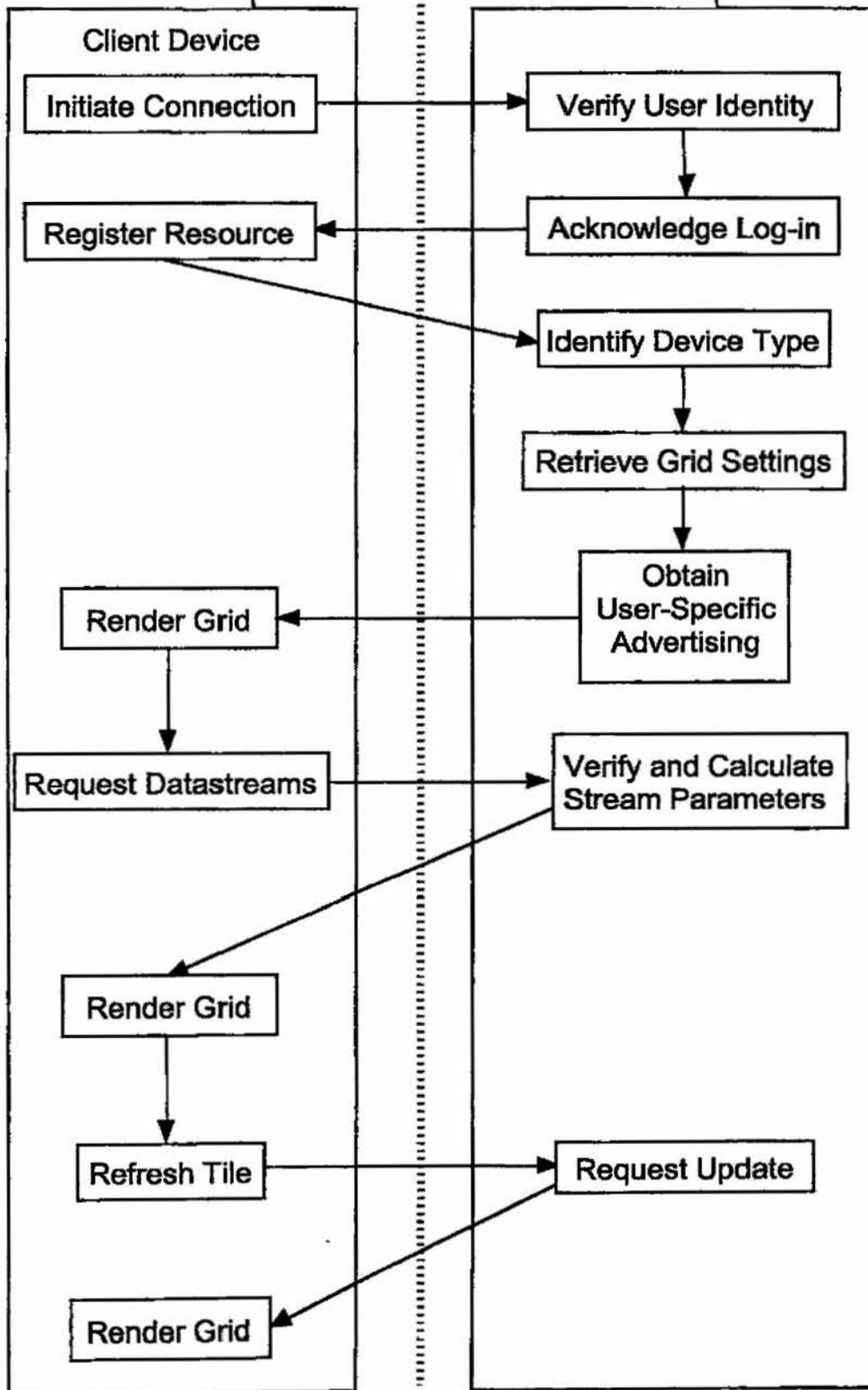


Fig. 12

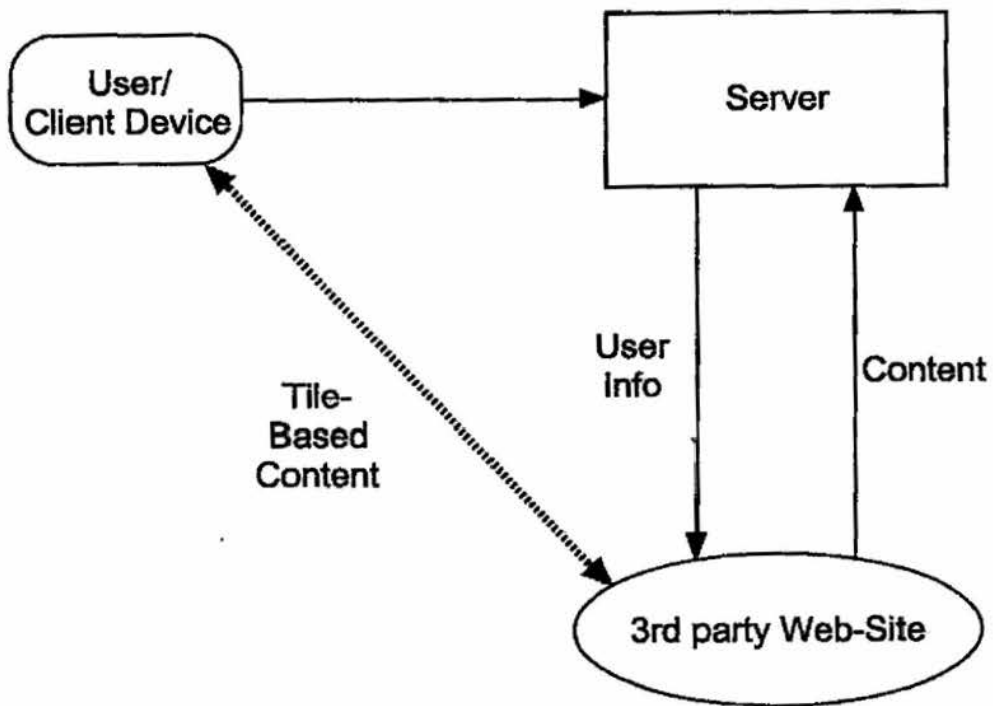


Fig-13