

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
3 April 2008 (03.04.2008)

PCT

(10) International Publication Number
WO 2008/038004 A2

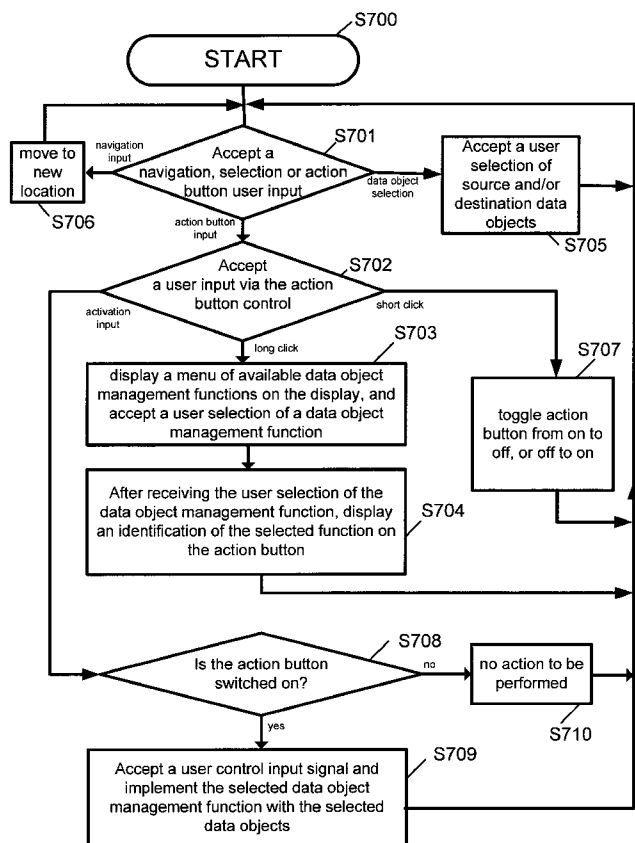
- (51) International Patent Classification:
G06F 17/30 (2006.01)
- (21) International Application Number:
PCT/GB2007/003680
- (22) International Filing Date:
27 September 2007 (27.09.2007)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
0619064.9 27 September 2006 (27.09.2006) GB
- (71) Applicant: Lee, Yisia Young Suk [KR/GB]; 4 Kersfield Road, London, SW15 3HW, United Kingdom (GB).
- (74) Agent: Collins, John David; Marks & Clerk, 90 Long Acre, London WC2E 9RA, United Kingdom (GB).
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM,

AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published: — without international search report and to be republished upon receipt of that report

(54) Title: A DATA OBJECT MANAGEMENT APPARATUS AND METHOD



(57) Abstract: A user interface apparatus for management of stored data objects, the apparatus comprising an interface for retrieving stored data objects from a data store; an input for receiving a user selection of a data object management function, a user selection of one or more data objects, and a function activation input to implement a selected data object management function; and a controller for controlling at least one display to display a representation of data objects and to display an identification of a user selected data management function, and for implementing said user selected data object management function using said user selected data objects in response to said function activation input.

WO 2008/038004 A2

A Data Object Management Apparatus and Method

The present invention relates to an apparatus and method for management of data objects, and in particular, to a user interface apparatus and method for management of data object properties and/or relationships.

It is desirable to make it easy for a user to navigate and manage stored information, e.g. to locate information of interest to the user, and to modify the stored information. The information may include data with hierarchically structured content, e.g. a database with subject matter topics and subtopics.

Data is commonly stored on computer readable media such as disks, which are formatted and indexed according to a particular disk file system, e.g. FAT, NTFS, HFS, ext2, ISO 9660, UDF. Data on the disk may be stored and organised as data files within a hierarchical directory structure. File management functions such as copy and paste, provided by a computer operating system, e.g. Windows 2000™, Mac OS™, Linux™, may be applied to selected files to change their location within the hierarchy.

Data with hierarchically structured content may be stored on the disk by using directories and subdirectories created on the disk to provide the required hierarchical structure. The hierarchical data on the disk may then be browsed by user navigation through the hierarchical directory structure, and during this navigation, the user may select desired files for display or use.

A disadvantage of using such directories and subdirectories to represent the hierarchical content relationships between data files is the amount of time and computer resources required to move large numbers of files from one directory to another. In particular, where data sources comprise multiple disks and/or multiple network locations, it is extremely slow to physically rewrite files from one location to another. Furthermore, different network locations may be owned by different information providers, each contributing a part of the data, and it may be unacceptable to transfer files between

them. The databases may be used by a large number of different users, and it may be unacceptable for any single user to move files to new locations.

An alternative method of storing data files with hierarchically organised content is independent of the hierarchy of the disk file system. Instead of using directories and subdirectories of the disk file system to organise data files into a hierarchy by content, each data file may be provided with a label indicating its position in a “content hierarchy”. The data file location in the disk filing system thus is not related to its position in the content hierarchy. This method has the advantage of allowing the positions of files within the information hierarchy to easily be changed, without moving the physical location of the files on the disk(s) or network.

Each data file may include information defining the position of that data file in the content hierarchy. For example, the filename of the data file may comprise an identification of the content hierarchy position, e.g. as a text string. Alternatively, text or binary information defining the position may be stored as part of the file contents.

In another embodiment, a separate indexing scheme may be used, which is readable independently from any of the data files of the content hierarchy. The separate indexing scheme may be a single index, e.g. comprising a single file, or it may be a plurality of files, each defining the hierarchical position of one or more data files. In some embodiments, the index may be stored in a text format, although this is not essential.

A mark-up language such as XML (eXtensible Mark-up Language) may be used to process the information about positions in the content hierarchy. The mark-up language may translate this position information into a format that is readable as a hierarchical position. In some embodiments, the mark-up language may also amend the information about positions in the content hierarchy. For example, the mark-up language may be used to “move” a file to another location in the content hierarchy by amending the stored information defining its position in the content hierarchy. It is not then necessary to physically move data on the disk or network in order to change the file’s position in the hierarchy.

The present invention provides an apparatus and corresponding method for management of stored data objects. The data objects can be stored in local or remote storage, and they may include web pages, multimedia files, text files, applets, applications, data files, executable code, upgrades to applications, links to further data objects, or any other type of storable information.

A user may select a data object function and one or more data objects via a user input device. The data object function may be a function for creating a new data object, deleting an existing data object, deleting a link to an existing data object, modifying a data object, or modifying a link to a data object. Alternatively, the data object function may allow a user to set up or modify access rights for data objects, and/or to set up or modify access rights to allow or deny permission to other users to modify links or relationships between data objects. Many other data object functions are also possible, e.g. the data object function may perform a file management role, or a file editing role. This role may be at least partly defined by the data object selection. In one embodiment, a particular function may perform a file management role when the selected data objects are files, but the same function may perform a file editing role when the selected data objects are components of files.

A data object function may be implemented by firstly selecting the function from a plurality of available functions, and subsequently, activating the function. A first user input may be received to select the function, and a second user input may be received at a later time to activate the function. Data objects may be selected before or after the function has been selected. In one embodiment, the apparatus includes a memory to store independent selections of data objects for a plurality of different functions. If a user selects a first function, selects data objects, and then selects a different function, the original selection of data objects is stored and restored the next time the first function is selected.

Activation of the function normally involves performing the function using any selected data objects. Some functions (such as "new") require only a single data object to be selected (e.g. representing the "folder" or other location where the new data will be created). However, in particular embodiments, such functions may be performed

once for each selected data object, thus several new data objects could be created by a single function activation.

The apparatus includes a display controller for controlling one or more displays to display a representation of data objects. For example, data objects may be represented by icons, hyperlinks, etc. The display controller may also display an identification of a user selected data management function, e.g. as an icon or text identifying which function is currently selected.

The apparatus includes a controller for implementing a user selected data object function, using selected data objects, when the function is activated in response to an activation input from a user.

The apparatus may include storage for storing data objects, such as a local disk drive. Alternatively or additionally, a storage device for storing data objects may be provided remote to the apparatus, and accessible via a communications link. Examples of such a communications link include a wireless network link, an infrared link, a Bluetooth link, a radio link, a microwave link. The stored data objects may be static or dynamic, for example, a data object on the storage device may be updated at times, or may comprise a link to an information stream (e.g. a television or radio broadcast).

The storage for storing data objects may be a storage device in a single physical location, or may be a plurality of storage devices in several physical locations. The display controller may also be configured to display user selectable menu items for navigation through the plurality of data objects, e.g. with individual menu items corresponding to individual data objects.

Thus the present invention provides a user interface which allows for easy navigation and management of a collection of data objects stored on one or more data storage devices.

In embodiments of the invention discussed below, the screen area identifying the data object management function is referred to as the “action button”, because it provides a

user interface control for allowing a user to initiate the operation of the data object management function.

In a preferred embodiment of the invention, a plurality of locally stored data objects each represent a data file that is stored in local or remote storage means. These locally stored data objects define a set of relationships between the data files, e.g. they may define an information hierarchy by content of the information in the data files, or they may define some other type of information structure. It is thus possible to modify relationships in this information structure without the need to modify, or even to access, the data files on the storage means, by simply modify the locally stored data objects representing the data files. If a user requests to view the contents of a data file, then at that stage, a request may be sent to the storage means to retrieve the relevant stored information.

The user input means may also be configured to receive user navigation commands for navigation through the plurality of stored data objects, and the control means may be configured to display a representation of particular data objects that are selected according to said user navigation commands. The data objects may comprise a plurality of links between data files, the links defining a virtual file system. These links may be stored in storage means as one or more data files.

The display controller may be configured to control the display to display said representation of data objects in a first display area, and to display said identification of a user selected data management function in a second display area separate to the first display area. The display controller may also be configured to display a third display area for displaying a navigation menu with menu items representing data objects, or other user navigation controls.

The second display area may show a user operable control button, and the display controller may be configured to display a menu of available data object management functions for selection of one said function when a first type of user input is received, and the control means is configured to implement the selected function when a second type of user input is received.

The data object management functions may comprise a plurality of file management and file editing functions. Some or all of the data object management functions may be configured to perform both file management operations and data file editing operations, depending on whether a file or a part of a file content is selected as a data object, e.g. a move function may either move a file to a new location in the information structure, or move text or graphics within a file to a new location, in both cases with the same user interface.

In one embodiment, the use of editing functions on a data file may have no effect on the source data file on the storage means, but instead create a new data object containing the edited text. In another embodiment, editing of data files may result in changes to the stored source data files on the storage means.

The user input means may be configured to accept selection of a plurality of destination data objects, and to perform said selected function for the plurality of destination objects in a single implementation operation.

The retrieving means may be adapted to retrieve only information corresponding to a most recently selected menu item for a current level of the hierarchy in said hierarchically arranged information source in response to a most recent user selection. The display controller may be adapted to control the display to display new menu items for a new level of the hierarchy when the user input device receives a history menu item as a user selection.

The action button may be used to perform “file management” type functions, or file editing functions. The file editing functions may be used to edit text, graphics, or other components of a data object or file.

A first type of highlighting may be used for selection of source data objects, and a second type of highlighting may be used for selection of destination data objects. For example, reverse colours may be the first type of highlighting, and a dotted border or outline may be the second type of highlighting.

Embodiments of the present invention have the advantage of allowing a common user interface for management and editing of data objects on any level. File management functions may be performed, e.g. a file or directory may be moved to a new location in the hierarchy, and/or file editing functions may be performed, e.g. moving, copying and pasting within a single file. There is thus no need to provide separate applications to handle basic editing of text or graphics files, which can save memory and disk space on a portable computing device, or provide additional functionality on a device that would not normally have any file editing functions. Both files and virtual directory structures can be edited or created with the same user interface.

A further advantage of embodiments of the present invention is to make efficient use of available screen area, making it suitable for use a device of limited size or with a limited display area, such as a portable computing device. All or most of the screen area may be dedicated to display of data objects, and minimal area of the screen is taken up by the user interface. The use of a single control button for a plurality of data object management functions saves space on the screen. Thus, embodiments of the invention allow the size of hardware to be minimised whilst available display area for the information is maximised.

A yet further advantage is provided over traditional filing systems by the ability for any data object to have associations between any two data objects. Thus, for example, a data object representing a file may also behave as a “directory” in the content hierarchy, allowing other files to be associated as child data objects in the content hierarchy. For example, a graphics file may have an association with a sound file. In another example, a particular child data object may be associated independently with two different parent data objects at different parts of the information structure.

Although some described embodiments of the invention are discussed in terms of hierarchical information content, this is not essential to the invention. The information content may have an alternative structure, for example, particular files may be accessible from multiple parts of an otherwise hierarchical structure, or cyclical relationships may be included, e.g. A leads to B, which leads to C, which leads back to

A. The information may include text, web pages, photographs, images or video images for displaying on a screen, and/or audio information, such as digital speech or music, for outputting through a speaker.

Keyboard shortcuts may be provided to allow the action button to be changed to a predetermined function on performing only a single user input action, such as pressing a key or a key combination.

Menus may be opened by clicking the pointer or rolling the pointer over a particular area of the screen. It can be seen in this diagram that a user can navigate back to any level in the hierarchy using the history menu to display new menu items for the next level in the hierarchy below the selected history menu item.

Data objects may include files (e.g. text, audio, graphics, video, other types of data files), applications, links to files, links to particular storage means (e.g. to a DVD reader), components of files (e.g. text, audio, graphics, video, or other data components), shortcuts, links, pointers, hyperlinks, directories, folders, virtual folders, temporary data in memory, etc. The data objects may be files stored in a disk filing system or data stored in a volatile memory area. File management operations may include operations to move files from one location to another, to copy and paste files, to delete files, create new files, etc. File editing operations include editing, copying, pasting or deleting elements of the content of a file, for example, text or images contained within the file.

A user may activate the action button with a designated secondary action, such as a long click with the mouse, when the pointer is located within the action button. This results in a menu appearing on the screen, giving a choice of action functions. Examples of the functions available for the action button include “New”, “Move”, “Copy”, “Paste”, “Delete” and “Undo”. The user selects an action function, and this newly selected function becomes the function of the action button.

In some embodiments, a navigation menu may be displayed to provide additional user navigation options. For example, a navigation history menu may be displayed to show

the navigation route which has been taken by the user, and allow a quick shortcut back to a selected point on that navigation route. Menu items may each be displayed in an allocated area of the display, arranged in a series, for example, in a series of boxes displayed on the screen.

In the present invention, a display may be configured to display a history list at the same time as displaying user selected information, allowing rapid and easy navigation through the directory structure, whilst simultaneously showing the selected information. The system may also be configured to allow activation of an item in the history list to allow a sub-menu of the activated item to be displayed, for further ease of user navigation.

The apparatus may include a loudspeaker, for playback of audio information. However, the present invention is not limited to such a technology or design.

The display may be a monitor, TV screen, touch screen, or any other type of screen or display for use with a computing device. The display may be configured to display a navigation panel in an area of the display, the navigation panel comprising a complete or partial mapping of the node structure for allowing alternative navigation with a further user input device. The information display may include holography.

Items may be selected using a pointer displayed on the display, for example, a pointer linked to a mouse or other input device. Alternatively, items may be selected without a pointer being displayed on the screen, e.g. by use of a touch sensitive screen, and a user touching the part of the screen corresponding to the item. Items may be selected by the positioning of the pointer on the screen, and by clicking or activating a pointer device, or by single or multiple touches to a touch sensitive screen.

The user input device may be a keyboard, keypad, mouse, pen, game controller, joystick, remote control, track pad, tracker ball or other pointer device, touch sensitive screen or any other device for allowing a user to input information to the apparatus. The user input device may be a keypad with three keys, or a keypad with two keys, where mode selection is operated by pressing both keys simultaneously, or a joystick e.g. a

mini-joystick which is operated by the fingers rather than a handheld joystick. It may be a pressure controlled device such as a track point device or touch pad. It may be a keyboard, where only a small number of the keys are used for the navigation user input.

In some embodiments, the use of a further user input device or an additional part of a current user input device (such as additional keys on a keyboard) allows shortcuts to be taken from one part of the information structure to another.

In embodiments including a touch sensitive screen, the user input may involve any of a variety of methods, such as by a person's finger or using a stylus. If the screen is a touch sensitive screen, control button areas of the screen may be configured as touch sensitive buttons. If the screen is not a touch sensitive screen, they may be operable using a user input device such as a pointer device, by moving a cursor on the screen onto the position of the buttons. Alternatively, they may be used to visually indicate when a key or button on a keypad or keyboard is activated.

The apparatus may comprise any computing device, including a handheld or portable computer, an embedded computing device, a PDA, a mobile phone, etc. The information store may be a disc such as a hard disk, floppy disc, CD, DVD, optical disc, etc, and the information store may be provided as part of the computing device, or it may be remote from the computing device. The information store may be of a read-only type or may be of a read-write type. The invention could be embodied in software, specially configured hardware, or a combination of software and specially configured hardware.

A computing device may be configured as an apparatus according to the present invention, by means of computer readable code provided on the computing device for configuring the computing device as such an apparatus. The data for user navigation may also be provided on the computing device. Alternatively some or all of the data may be provided separately or remotely from the apparatus and the apparatus may be accessed by the apparatus e.g. from a plug-in data card or disk, or over a communications link.

Additional visual information may be used to assist a user with navigation, for example, the colour of a part of the display may be changed according to which level of the hierarchy and/or which particular branch of the hierarchy is being displayed, or when the navigation proceeds to a different group of subnodes.

The virtual file system may include files physically stored on a plurality of different types of file storage system, e.g. files stored on different operating systems, on different types of computers, etc.

In some embodiments of the invention, the access rights of data objects may be set or modified to allow or refuse access permission to certain users or groups of users. For example, the access rights may include read access, write access, and execute access for data objects. An owner of a data object may choose to allocate or modify any of these types of access rights to data objects. Where an owner of a data object is specifying access rights, the owner may specify particular groups of users, and allocate access rights to all users in the group. For example, the group may be selected according to network location or network address, or according to a predetermined list of users. For example, the owner may allow all users at the same network address to modify a particular data object, but refuse permission to users at any other network address. Where a list of users is specified, this list may be set up by a system administrator, or may be set up by the owner of the data object.

In some embodiments, the information relating to links between data objects may also be given access rights. This information relating to links between data objects may or may not be treated as a data object in its own right. A user may be given permission to modify these links between data objects, independently of any access rights to the data objects themselves. This may allow users to “move” and “copy” files within a virtual file structure, but not to edit the files themselves. In other embodiments, the access rights to data objects may be linked to the access rights to modify links between these data objects, so that the owner only needs to make a single change to the access permissions to control the access to both.

In some embodiments, the action button may be used to invoke a function which changes the access rights of data objects. This function may allow a user to select one data object or a plurality of data objects and to modify the access rights on all of these at the same time. This function may also provide the user with the option of adding or deleting members to one or more of lists of other users and changing access rights for these lists of users.

The use of embodiments of the invention to change access rights to data objects is particularly useful when implemented on a network such as the internet. For example, it may allow employees from different physical locations in the same company to work more effectively together on-line. These embodiments may be used for many purposes in addition to data organising, for example, broadcasting and managing data objects, control electronics, processing, editing, design or developing of projects, or any other software implementations.

Embodiments of the invention may use web 2.0 software, such as Ajax, Macromedia Flash, Java or any programmable languages. These may be used to implement on-line applications, which may include data object functions, and may be invoked using the action button.

In some embodiments of the invention, the data objects may include data objects relating to upgrades, downgrades or other modifications of software applications or apparatus, e.g. using software plugins, new software components or applications, and hardware devices and components. Embodiments of the invention may be used to track which upgrades, downgrades or modifications have taken place. For instance, a data object may be created to represent a particular upgrade, downgrade or modification. These data objects may be stored together with information on the date of the upgrade, downgrade or modification, and any other relevant technical data. Thus, embodiments of the invention may be used to represent and manage a hardware and/or software system.

In one embodiment, a user may be able to track and manage changes to software or hardware by using an interface provided by the invention to manually “move” new

components to a data object representing particular hardware or software, to record that a change has been made. Similarly, links to data objects representing components may be removed from data objects representing software or hardware, to represent removal of that component. This interface may use the action button to allow a user to select an appropriate function (e.g. “add component”, “remove component”, and to record the modification to a single piece of hardware or software, or to several pieces of hardware or software at once. These embodiments may be configured to display the history of the upgrades, downgrades or other changes, and they may be configured to automatically determine and display any potential conflicts between different components, e.g. using a database specifying potential conflicts. For example, such a system could indicate that a particular type of graphics card was not compatible with a particular type of computer, or that a particular software upgrade would not run satisfactorily on a certain specification of computer. The above system may also be used to represent different hardware connected to a network, and allow a network administrator to record or simulate the addition or removal of components on the network. In some embodiments, the likely effects on network performance may be calculated, e.g. by simulation.

For software components, some embodiments of the invention may be configured to combine individual software modules to link or increase their functionalities. For example, if a user links a “timer” application to a “clock” application, the functionalities of these two data objects may be combined to produce an object with additional functionality as an alarm clock. This may be achieved automatically if the applications are initially written to be compatible in this way. Similarly, if a user links an “alarm clock” application to a “TV channel” application, the functionalities of these two objects may be combined to produce a TV program recording system which can record for selected times.

Embodiments of the present invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

Figure 1 is a block diagram showing the apparatus according to an embodiment of the invention;

Figure 2 is a screen layout diagram showing a computer screen layout in an embodiment of the invention;

Figure 3 is a screenshot of a user interface according to an embodiment of the invention, using the screen layout of figure 2;

Figure 4 is a schematic view demonstrating how the screenshot of figure 3 relates to data objects on different kinds of storage media;

Figure 5 is a screenshot showing an example of a file naming scheme that creates a “content hierarchy” for a plurality of data files stored in a single directory;

Figure 6 is a schematic diagram showing how the data on the storage means can be interfaced to a file navigation system via XML;

Figure 7 is a block diagram showing how different components may be integrated to produce a navigation system according to an embodiment of the invention;

Figure 8 is a schematic diagram showing the activation and deactivation process for the action button, in an embodiment of the invention;

Figure 9 shows a plurality of screen layouts, illustrating how the function of the action button may be selected by a user, in an embodiment of the invention;

Figures 10A and 10B show two examples of menus, from which the function of the action button may be selected;

Figure 11 is a flowchart showing a process for implementing a data object management function according to an embodiment of the invention;

Figure 12 is a screenshot of a file-browser user interface in an embodiment of the invention, showing an example of an activated category with two selected items;

Figure 13 is a flowchart showing the response of a user interface system to user input, in an embodiment of the invention;

Figures 14A-F show a plurality of computer screenshots demonstrating the use of the “Move” action function in an embodiment of the invention;

Figures 15A-C show a plurality of computer screenshots demonstrating the use of the “New” action function in an embodiment of the invention;

Figures 16A-D show a plurality of computer screenshots demonstrating the use of the “Select”, “Copy” and “Paste” action functions in an embodiment of the invention;

Figures 17A and 17B show a plurality of computer screenshots demonstrating the use of the “Copy” and “Paste” action function to edit a text document in an embodiment of the invention;

Figures 18A and 18B are screen layout diagrams, showing two different examples of a screen menu system and a remote control unit for navigation in an embodiment of the invention;

Figure 19 shows a further example of a screen layout, where content information and controls from data objects are displayed, rather than just icons;

Figure 20A shows a partial screenshot of the movie player interface of figure 19 with a file selection menu. Figure 20B shows a summary information screen, with a summary of file information for an MPEG-4 file.

Figure 21 is a screenshot of an alternative file browsing system, incorporating an action button according to the invention;

Figure 22 shows separate screenshots for two different users of a system according to an embodiment of the invention;

Figure 23 is a screenshot of a screen displaying the contents of a folder or virtual folder in an embodiment of the invention;

Figure 24 is a screenshot showing the use of multiple windows, in an embodiment of the invention;

Figure 25 is a computer screenshot showing how selected objects may be labelled with a checkbox to indicate whether they are selected by a user;

Figure 26 is a computer screenshot showing how a selected items list may be displayed in a bottom screen area, in one embodiment of the invention;

Figure 27 shows a screenshot of a folder system, and a corresponding remote controller with a display for displaying a list of selected items; and

Figure 28 shows a partial screen layout including forward and back navigation buttons;

Figure 29 shows the screen layout of figure 28 adapted for use with a remote controller, where the navigation buttons are not displayed on the screen;

Figure 30 shows the screen layout of figure 28, including a further navigation menu which is temporarily displayed in the middle of the screen;

Figure 31 shows a hierarchical file organisation structure, for organising files from a plurality of different file stores;

Figure 32 is a diagram showing multiple screenshots for a navigation process through the hierarchy of figure 31 using the forward and back navigation buttons;

Figure 33 is a diagram showing multiple screenshots for a navigation process, using an alternative screen layout;

Figure 34 is a diagram showing multiple screenshots for a navigation process, using another alternative screen layout; and

Figure 35 illustrates a user input method in which each of the five fingers on a person's hand controls a separate user input.

Figure 1 shows an apparatus according to a first embodiment of the invention. The apparatus includes a computer 100 connected to a display 101, a keyboard 102 and a pointing device such as a mouse 103. In this embodiment, the computer 100 includes an information store 106, a CPU 104, a display driver 107, a working memory 105, and a broadband modem 109, all of which are connected to an internal bus 108. The computer 100 is configured to receive user input from the keyboard 102 and mouse 103, select and retrieve information from the information store 106, and output the retrieved information to the display 101. In alternative embodiments, instead of a keyboard and a mouse, a user input device may be provided which is integral with the display 101, e.g. a touch sensitive mechanism provided on the display.

The memory 105 is used for loading and implementing program code. This code may be retrieved from the storage application 106 for loading into the memory 105.

The broadband modem 109 allows the apparatus to access externally located information, e.g. via the internet. In a further embodiment, an alternative interface means may be provided instead of, or in addition to, the broadband modem 109. For example, the interface means may comprise a port or connector to allow the apparatus to be connected directly to another computer apparatus, or the interface means may comprise any type of network interface. Information may be obtained via the interface means, the information store, or a combination of the two.

Figure 2 shows a screen layout for a computer screen 400 in a first embodiment of the invention. The computer 100 is configured to control two display areas on the display 101. The screen 400 is divided into a contents window 401 and a border area 402. The contents window 401 is used for displaying data objects, and the border area 402 contains an area for a button that may be user activated to perform a predetermined

function, and is referred to as an “action button” 403. The action button 403 is a defined screen area which can display an icon or text indicating a user selected function, and which may be user activated by means of a user input device such as a mouse, thus providing an interface to allow a user to select a particular function, e.g. a file management function.

Figure 3 shows a screenshot having the screen layout shown in figure 2. Again, the action button 403 is located in the bottom left corner of the screen. The contents window 401 shows icons representing several different data objects, including a document, a folder labelled “Folder A”, a music file, an application, a link to a DVD ROM drive, a movie file, and a message file which may be a saved email message.

The icons shown in figure 3 do not directly represent the physical locations of data files, but instead, are data object links, which are more similar to MS Windows “shortcuts”. Moving an icon from one location to another has no effect on the location of the corresponding data file on the disk or network. Instead, only the data object link is moved to a new location. A data object link may act as a data object in its own right. Data object links may be generated manually, or may be dynamically and/or automatically generated by the user navigation system. Data object links may be generated according to the available data files at a local and/or remote data source. They may be added to the user navigation system as a result of selection of data files by a user.

Data object links are preferably stored locally on the user navigation system apparatus, because this improves the speed of reading and writing to these data object links. They may link to local files, remote files, or to other local or remote data objects. For example, they may provide a link to a DVD ROM drive, an application, or to a directory in a disk file system. Data object links may each be stored as individual files, or a plurality of data object links may be stored within a single file, e.g. a text file. A data object link may include details of any associated files, and user customised file parameters. Data object links may include links to any type of file, e.g. any type of data file, applications, folders, etc.

The use of data object links in the navigation system allows a plurality of users to each maintain their own personalised information hierarchy, using a shared set of source data files. Thus, users may move, copy, delete, or create new data links, and may set up individual data links to store customised parameters. For example, a link to an application file may contain user customised start-up parameters.

One benefit of using such data links to represent data files is that each data file can have associated “sub-files” by storing the appropriate information in the shortcut. Thus, each data file can appear to the user to behave like a folder as well as a file. This is not a feature that is provided by MS Windows shortcuts.

A top part of the contents window 401 in figure 3 contains a plurality of tabs 410, which provide a navigation menu to allow a user to quickly navigate to other information accessible by the system. In this example, only the first tab (labelled as “System”) can be activated, and the others are blank. The “System” tab is the currently selected tab, and this is indicated by the tab area joining continuously with the contents window area 420. In this embodiment, selection of a tab results in a display showing the data objects that are associated with the current data object, and that appear to the user as though they are contained in a folder represented by the current data object.

In some embodiments, the navigation menu takes the form of a history list, showing a history of recent items that a user has displayed. The history list can allow a user to easily navigate back to a location which they have previously visited. Each item in the history list is allocated a separate box or area of space within the upper section of the contents window. For example, previously displayed tabs in the history menu may be displayed to the left of the currently displayed tab. In alternative embodiments, the navigation menu may provide links to different steps of a path through a hierarchical information structure.

Figure 4 is a schematic view showing how the screenshot of figure 3 relates to data objects on different kinds of storage media. The top part of the figure shows a simplified view of the screenshot of figure 3, including category tabs, each having an associated contents window which shows items associated with that particular category.

Selected items are shown in reverse colours. The bottom part of figure 4 shows a perspective view of the screenshot in the top part of the figure, with arrows linking the data items in the contents view to a plurality of different possible data sources. These data sources include a hard disk drive, flash memory unit, a holographic data storage system, a database accessible via a web server, a USB memory card, a flash memory card, a CD ROM disk, a DVD ROM disk, and a holographic data storage system ROM disk. It is not essential to have all, or even any, of these data sources present, although it is advantageous to use embodiments of the invention to navigate information on such data sources. Any alternative types of data sources could also be used, e.g. other types of computer readable cards, chips or disks, a DVB-H source (digital video broadcasting) and data from any type of medium via any type of network. Embodiments of the invention serve the purpose of managing data syndication, including hierarchical or inheritance relations, for data from a wide range of possible sources.

Figure 5 is a screenshot showing an example of a file naming scheme that defines a hierarchy for a plurality of data files. The hierarchy is completely independent of the disk filing system, as all of the files are located in a single directory of the disk filing system. In this example, the hierarchical information relates to ecology, and each of the plurality of file names begins with the word “ecology” followed by a dash. The dash acts as a separator between levels of the hierarchy, although an alternative separator character could equally well be used. The first listed file has file name “ecology-amphibians-image.swf”. Thus, the first file relates to an image in the amphibians part of the ecology information hierarchy. The second, third and fifth listed files have file names “ecology-birds-image.swf”, “ecology-fishes-image.swf” and “ecology-insect-image.swf”, and relate to images in the birds section, fishes section and insects section respectively of the ecology information hierarchy.

The fourth listed item has file name “ecology-image.swf”, and simply relates to an image in the upper level of the hierarchy. The sixth to the thirty fifth files have files names beginning with “ecology-mammal-”, indicating that they are in the mammal section of the hierarchy. These are further divided by filename into several sub-categories of mammal, including “ecology-mammal-carnivora-image.swf”, “ecology-mammal-cetacea-image.swf” and “ecology-mammal-primates-image.swf”. Several

further sublevels are also defined for some of the items. Thus, a complete hierarchical structure may be defined by a text string within the file name of each item in the information hierarchy.

Figure 6 is a schematic diagram showing how the data on the storage means can be interfaced to a file navigation system via XML. The navigation system 152 may send instructions to an XML application 151, and may receive data from the XML application 151. This is indicated by arrows in both directions between navigation 152 and XML 151. However, in this embodiment, the XML 151 does not modify the data 150 held in the storage means, but only receives a copy of this data 150 and processes it in a way that the navigation system 152 can make use of it. Thus, only a single arrow is shown between the XML 151 and the data 150, indicating data flow out of the information store. The XML application may generate data filenames using an information request from the navigation system that specifies a hierarchical location. The XML application may also receive a data file and translate the text string of the file name into a hierarchical location, and pass this information to the navigation system to inform it of the file's position in the hierarchy.

In one example of a navigation system according to the invention, data object links in the user interface may be automatically generated from hierarchical data filenames, via XML or another suitable text processing application. The XML application may read a hierarchical data filename, and automatically process it to obtain the location of the data file within the hierarchy, and to generate a data object link corresponding to that location in the user navigation system.

Figure 7 is a block diagram showing links between different software components in a navigation system according to an embodiment of the invention. Each component is indicated as a box, and the boxes are arranged in an interlinked structure indicating the data flow between them. Box R702 indicates the user navigation operating system, which can send and receive information to the users R701 of the system, as indicated by the arrows in both directions between them. Two way data flow also may occur between the navigation operating system R702 and each of several installed software application, which include communication/terminal emulation software R703, WWW browsers

R704 (e.g. Netscape or internet explorer), FTP clients R705, and other applications R706 (e.g. email, word, Photoshop, video games). Bidirectional data flow may also occur between the navigation operating system R702 and external interactive software R707 which may be located on an external DVD disk or USB memory.

Each of the software applications may interact with network components, as follows. The communication/terminal emulation software R703 may interact with a 3G, WAP or GPRS network component R708. The WWW browser R704 and the applications R706 may interact with WWW servers R709 on the network. The ftp clients R705 may interact with ftp servers R710 on the network. The applications R706, the external interactive software R707, and the navigation operating system R702 itself may all transfer data to and from an internal or external storage means containing a database, e.g. containing hierarchical information. In the diagram, the links are all shown as bidirectional, apart from the external interactive software R707, which reads data but cannot be written to. However, in other embodiments, it may be possible to write data to this software, e.g. if the software is provided on a re-writable medium.

The WWW servers R709 may exchange data with further applications and documents R712, such as HTML documents, DHMTL, XML, RSS, Java and Macromedia Flash. In turn, these may exchange data via scripting language applications R713 such as Perl, PHP, Python, MySQL and ASP. Data may be exchanged between the scripting language applications R713 and a database R714 on the network, which may include hierarchical information, and/or an object file system. Data may also be exchanged between the database R714 and the ftp servers R710.

Software relating to the navigation methodology in an embodiment of the invention may be present as part of the operating system R702, any of the software applications R703, R704, R705, R706 or R707, or as a component in application R712, which may be HTML docs, DHTML, XML, RSS, Java, Multimedia Flash, etc. This is indicated by a shaded box with the words "Navigation Methodology" shown in each of these locations in the figure.

Figure 8 is a flow diagram, illustrating how the Action button 403 is activated and deactivated. When the action button is inactive, no function is shown in the border area 402, and instead, a blank square is displayed. The action button may be activated by a user clicking on it using a mouse or activating it using some other user interface. When activated, the appearance of the square changes, and text or an icon may be shown to indicate a currently selected function of the action button. In the example of figure 6, the currently selected function is a “Move” function, which is discussed in more detail below. To deactivate the action button, a user may perform the selection operation again. In some embodiments, the action button may toggle between an “off” state and an “on” state, when a user selects it using a first selection process, such as a single mouse click.

Figure 9 shows four different screenshots, indicating four different stages of a process for selecting an action button function. Arrows are shown connecting the screen layouts, to illustrate progression from one screen layout to another.

The first screenshot shows the activated action button, displaying the text “Move”. The action button may be activated by clicking on it using a pointer device, pressing a key on a keypad or keyboard, etc. To change this selected function of the action button, a user may use a second kind of user input differing from that used to switch the action button on and off. The second kind of user input may involve, for example, pressing the button for an extended period, such as 1 second, or pressing a second key on a keyboard or keypad, etc.

When the user input to select a new action function is received by the system, the screen display is updated to the second of the screenshots in figure 9. This shows a function menu in the middle part of the screen, containing a list of user selectable functions, with the current selection, which is a “Move” function, being highlighted. To select a different function, a pointer device may be used, or keys on a keypad or keyboard may be used to scroll through the options. The third screenshot of figure 9 shows the “Edit” function being selected in place of the “Move” function. Finally, the fourth screenshot of figure 9 shows the screen after the new selection has been made and the menu has

disappeared. The Action button now shows the text “Edit” instead of “Move”, and selection of it will activate the edit function.

Two different examples of the action button function menu are shown in figure 10. The first menu shows the options “New” 501, “Move” 502, “Copy” 503, “Paste” 504, “Delete” 505 and “Undo” 506. This menu thus provides file management type functions. However, in preferred embodiments of the invention, these functions may be used for both file management and file editing, with the same user interface.

In some embodiments of the invention, a larger number of options may be provided on the function menu by the use of submenus. On selection of a submenu option, the submenu may be displayed, showing a different set of options from the original menu. This may be displayed in place of the original menu. One option in the submenu may be to return again to the previous menu. Additional submenus may be provided as selectable options from the submenus.

The second function menu shown in figure 8 has the options “Colours” 511, “Save as different name” 512, “Slice” 513, “Scale” 514, “Cook” 515 and “Disable” 516. This function menu may be appropriate for graphics editing and analysis functions. The slice function may be configured as a function for dividing up an image into parts. The scale function may alter the image size by a uniform factor. The cook function may be used to apply a pre-selected plug-in management process to applications/apparatus. “Colours” may be selected to alter the image colours, and disable may be used to disable this particular menu, and revert back to a master menu. Further options for the action button menu include options to change the font, size, style, colour, etc. of text or of another type of data object.

Figure 11 is a flowchart of a process of selection and activation of the action button in an embodiment of the invention. The process begins at step S600. At step S601, the apparatus presents the user with a list of data object management functions, and the user selects one of these functions. The apparatus receives the user selection, and updates the second display area of the screen accordingly. At step S602, the apparatus receives a user selection of one or more data objects. These may include both source data objects

and destination data objects, e.g. objects to “move” and a location in which to move them. Alternatively, these may include only source data objects, e.g. objects to “copy”, or only destination data objects, e.g. for creating a new file or folder. The apparatus updates the display to indicate which data objects have been selected by the user, and may indicate whether they have been selected as source objects or destination objects. At step S603, the apparatus receives a user input configured to trigger an implementation of the selected function. The function is then performed by the apparatus, using the selected data objects.

This user interface has the considerable advantage that a plurality of similar operations may be performed in a single batch, with fewer user inputs (e.g. keystrokes or mouse clicks) required than in a presently known file management system. For example, to copy a batch of files to several different locations, each of the locations may be selected as a destination data object, and a single implementation of the action button is sufficient to make the multiple copies.

In an alternative embodiment, selection of data objects may be performed prior to selection of the data object management function, or some data objects may be selected before selection of the function and others may be selected after selection of the function.

Figure 12 is a screenshot showing an example of highlighting used to indicate selected items, in an embodiment of the invention. In figure 12, two data objects have been selected. These are “Document A1” and “Image A1”. The icons representing each of these data objects are shown in inverted colours, to indicate their selection. In addition, the action button is shown as light text on a dark background, to match the highlighting scheme of the data objects, and indicate that it is active. In alternative embodiments, a different highlighting scheme may be used for data objects and/or the action button.

The dashed line around the “Document A3” icon indicates that this icon is being activated, in order to activate the category tab of Document A3.

Figure 13 is a flowchart showing a process of navigation through a hierarchical directory structure to obtain and display the information required by the user. The process starts at step S700. At step S701, the user interface accepts a user input either via the action control button, as a navigation selection, or as a selection of one or more data object. If the user input is via the action button, the process goes on to step S702. At this step, if the user action is a short click, the action button is toggled from off to on, or from on to off at step S707, and then returns to step S701 to await further user input.

If the user input at step S702 is a long click, then the process moves on to step S703, where a menu of data object management functions is displayed on the screen. The system accepts a user input comprising a selection of a function from this menu. Then, at step S704, after receiving the user selection, the action button is switched on and an identification of the selected function is shown on the action button, e.g. using text or a suitable image. The process then returns to step S701.

If the user input is an action button activation input, then the process goes from step S702 to step S708. If the action button is switched off at step S708, the process continues to step S710, where no action is performed, and then back to step S701. However, if the action button is switched on at step S708, the process continues to step S709, where the selected data object management function is implemented on the selected data objects. The process then returns to step S701.

If, at step S701, the user input is not an action button input, but a navigation input, the process continues to step S706. A navigation input may be a selection of a navigation menu, or a navigation selection to a selection of child objects of a selected data object, or the use of navigation controls such as forward and back navigation buttons. At step S706, the system updates the current location in the information structure to the newly selected location, and returns to step S701.

If, at step S701, the user input is a data object selection, then the process continues to step S705, where the selection of source data objects and/or destination data objects is accepted by the system. In some embodiments, there may be one type of user input set up to select a source data object, and a different type of user input set up to select a

destination data object. Not all functions require a source data object and a destination data object, but some do, for example, a copy function requires an item to copy, and a destination location to copy the item to. After the user selection of data objects is made, the process returns to step S701, from which further selections of data objects may be made, or other user inputs may be received.

Figures 14A-F show a series of screenshots occurring during an example use of the “Move” action button function. Each of figures 14A-F shows two screenshots. Figure 14A shows a first screenshot, which may be a start-up screen when the user interface is initially started up. The text “System” is displayed in the contents window 401. Only one of the menu tabs is in use, and this also shows the text “System”. The “System” tab is separated from the contents window by a dividing line. The action button is inactive. In this example, “System” is a data object which behaves like a folder in the traditional hierarchical directory based filing systems.

The second screenshot in figure 14A is the screen that is displayed if the user selects the “System” tab from the menu tabs. The contents window is now shown as integral with the “System” tab. The contents window includes icons representing seven files. These are a document entitled “Document”, a folder entitled “Folder A”, a music file entitled “Music”, an application entitled “Application”, a link to a DVD ROM entitled “DVD ROM”, a movie file entitled “Movie”, and a message file entitled “Message”. These seven data objects are all associated with the “System” data object, as child objects.

A user may then generate a user input signal to open the action button function menu. Figure 14A shows an arrow leading to reference letter “Y1”, to denote that the next stage of the process is shown in figure 14B, which begins with reference letter “Y1”.

The first screenshot in figure 14B shows the same contents window as before, but the action button has been activated with the “Move” function. This activation process corresponds to step S707 in figure 13. The next screenshot shows the same screen, but with the document file icon for “Document” being selected and highlighted. The selection process corresponds to step S705 in figure 13. The process then continues at “Y2” in figure 14C.

The first screenshot of figure 14C shows the same screen as previously, but with the application icon for “Application” and the movie icon for “Movie” also highlighted. The selection process again corresponds to step S705 in figure 13. The second screenshot of figure 10C also shows the folder icon for “Folder A” being highlighted with a dashed line, instead of the reverse colour highlighting of the document, movie and application. This is to indicate that “Folder A” is being selected to activate its category tab. The category tab can be activated, for example, by double clicking on the icon, pressing the button or key for a prolonged period such as 1 second, using a second button, or using a different finger than would be used to select data objects for operation by an action button function. The process then continues at step Y3, in figure 14D.

The first screenshot of figure 14D shows the activated “Folder A” category tab. This corresponds to navigation step S706 of figure 13. The contents of folder A are represented as icons, consisting of document icons for “Document A1”, “Document A2” and “Document A3”, an image icon for “Image A1” and a folder icon for “Folder B”. In the next screenshot, three icons have been selected and highlighted, i.e. Document A1, Image A1, and Document A3. The folder B icon has dashed lines surrounding it, representing activation of the “Folder B” category tab. The process then continues at step Y4 in figure 14E.

The first screenshot of figure 14E shows the contents of Folder B in the contents window. This comprises icons for three movies, “Movie B1”, “Movie B2” and “Movie B3”, and an icon for “Folder C”. In the next screenshot, two of the icons (Movie B2 and Movie B3) have been selected and highlighted. The movie B1 icon has dashed lines surrounding it, representing activation of the “Movie B1” category tab. The process then continues at step Y5 in figure 14F.

The first screenshot of figure 14F shows the contents of the “Movie B1” tab. This comprises an application “Application C” and a folder “Folder D”. The action button “Move” function is then activated to move all of the highlighted files to the Movie B1 tab. This may be done, for example, by double clicking or clicking on the category menu or clicking on the category tab, pressing a second button, or using a different

finger to drop the selected items into the Movie B1 category. The next screenshot shows all of these icons, now located within the Movie B1 category. The action button has been switched off.

Figures 15A-C show an example of an alternative process, starting from the point labelled X2 which follows the first screenshot in figure 14E. At point X2, the folder B category tab is activated, showing the “Movie B1”, “Movie B2”, “Movie B3” and “Folder C” icons, and the action button is set to the “Move” function.

Figure 15A shows first and second screenshots of the function of the action button being changed. In the first screenshot, the function menu has been activated, and is shown in the middle of the screen. The options are New, Move, Copy, Paste, Delete and Undo. The Move function is highlighted. In the second screenshot, the user has selected “New” instead of “Move”, and “New” is now the highlighted function. The process then continues at step Y6 in figure 15B.

Figure 15B also shows two screenshots in the process. The first screenshot shows the contents of folder B, including a new folder, named as “New Folder”. This has been created by the user activating the “New” function on the action button. The “New” function may be configured to create a new directory by the operating system, or just a new “shortcut” within the user interface which represents a folder and creates a relationship between other data objects. Alternatively, the “New” function may be configured to create a new document for a particular application, or a new topic (e.g. for an editorial website). The “New” function may behave differently depending on the activated category, and/or depending on other factors implemented by the object orientated programming. In the second screenshot of figure 15B, the category tab of the New Folder is selected, and the action button function menu is once again activated. The selected function is changed from “New” to “Paste”. The process then continues at step Y7 in figure 15C.

Figure 15C shows a single screenshot of the category tab of the “New Folder”, after the “Paste” function has been implemented. All of the data objects that were selected prior to setting up the new folder are now pasted into the new folder category.

Figures 16A-C show an example of an alternative process, starting from the point labelled X1 which follows the first screenshot in figure 14D. At point X1, the folder A category tab is activated, showing the “Document A1”, “Document A2”, “Image A1”, “Folder B” and “Document A3” icons, and the action button is set to the “Move” function.

Figure 16A shows two screenshots, and the first screenshot shows “Document A2” as highlighted. The action button has been switched off, so that Document A2 can be opened, run or played (depending on file type). In this case document A2 is a text file. The second screenshot of figure 16A shows the open text file, displaying part of the text, with a scroll bar at the right hand side of the screen to scroll through the rest of the text. The action button has now been switched on and set to the “Select” function. The process continues in step Y8 of figure 16B.

Figure 16B shows a first screenshot in which a part of the text of document A2 has been selected and highlighted. The action button function has now been changed from “Select” to “Copy”. The second screenshot of figure 16B shows the category tab of document A2 having been activated. This results in an empty selection, as document A2 currently has no associated files or objects in its category tab. The action function has now been changed from “Copy” to “Paste”. The process continues in step Y9 of figure 16C.

Figure 16C shows the copied part of document A2 pasted into the category tab of document A2 as a separate data object. This data object has then been selected and highlighted in the second screenshot of figure 16C. The process continues at step Y10 of figure 16D.

Figure 16D shows a single screenshot, following on from figure 16C. The text data object “changes to the ...” created from document A2 has now been opened, run or played to allow editing. It is now displayed with a scrollbar at the side of the screen, and the category tab menu contains a final item entitled “changes to the ...” to represent the category created by this object.

Figures 17A-C show an example of an alternative process, starting from the point labelled X3 which follows the first screenshot in figure 16C. At point X3, the text data object “changes to the ...” is displayed in the document A2 category tab. The action button is switched off.

Figure 17A shows two screenshots, and the first screenshot shows the re-opening of document “Document A2” to display its text. The second screenshot shows the “changes to the ...” object in the category tab of Document A2 being highlighted, and the action function set to “Copy”. The process then continues at step Y11 in figure 17B.

Figure 17B shows a first screenshot with the text of document A2 opened again. A location is selected within the text and marked by a cursor. The category tab is switched off before selecting the location within the text. The second screenshot of figure 17B shows the action function set to the “Paste” function, and the copied text object “changes to the ...” inserted and highlighted at the selected location with the text of document A2.

Figures 18A and B show two different menu systems that may be used with embodiments of the present invention. These menu systems may be used to provide navigation shortcuts to different parts of the stored information, or they may be used to provide options for functions available for the action button.

Figure 18A shows a menu area on the left hand side of the screen. In this embodiment, the whole height of the screen is used for the menu area. In some embodiments, this screen region may be permanently used as a menu area, and in other embodiments, the menu may only be displayed on activation, when it is required. In figure 14A, one of the menu items is highlighted. A submenu is also displayed, to the right of and slightly overlapping the menu. This submenu corresponds to the selected item from the menu. One item from the submenu has also been selected.

The menu may allow easier navigation of a hierarchical information structure. Thus, the menu may show each step in a path from the top of the hierarchy to the currently

selected information in the hierarchy, and allow a user to navigate straight to any of these steps by clicking on the appropriate menu item. Each submenu may show links to the information directly below the selected menu item in the information hierarchy.

Figure 18A also shows a remote control unit, which may be used as a user input device in embodiments of the invention. The remote control unit shown in the figure has two individual buttons at the top, and a direction controller pad at the bottom. The direction controller pad allows a user to move a cursor on the display screen to a desired position, e.g. to select a particular item. The two control buttons may be configured to switch the action button on and off, and to activate the action button function menu. In other embodiments, three or more buttons may be provided.

Figure 18B shows a second type of menu system which may be used in embodiments of the invention. This has a menu arranged in a horizontal direction at the top of the screen. This menu may correspond to the data object tabs described above, or may correspond to a history of the user's navigation or to a path through the hierarchy of information to the currently selected position. Below the horizontal menu at the top of the screen, is a two dimensional grid of boxes extending across the width of the screen. Each box may contain a submenu option, where the submenu corresponds with the selected item in the menu. In the figure, the second from left menu option has been highlighted, and a submenu grid of five boxes wide by two boxes high is displayed. The fourth from left, bottom submenu box has been selected and highlighted.

Built-in applications to open and edit files may be provided within the user navigation system. Alternatively or additionally, links to external applications may be provided.

The navigation system may be configured to save any changes implemented by a user in the source data file, or it may be configured to generate a new file if any changes are made by a user. A third option is that an "update" file may be saved and used to record changes to the source file for a particular user. Different users may own different update files.

A further advantage of embodiments of the invention is the use of a single user interface for both file management and file editing.

In some embodiments of the invention, two different data object selection types may be provided, to select source and destination data objects respectively. These selections may be indicated on the screen by different types of highlighting. For some data object management functions, it may be possible to select multiple destination files, as well as multiple source files, e.g. for the “copy” function.

Figure 19 is a screenshot in a further embodiment of the invention. Where figure 16 discusses the use of built-in applications to edit files, figure 19 shows several of these built-in applications open simultaneously for several different types of file. In this embodiment, the basic screen layout is similar to that described previously. A tab corresponding to a data object entitled “My Web” is open, and its contents are shown in the contents window. However, a higher degree of functionality is provided within the tab, than in the previously described embodiments. Instead of the data files just being shown as icons, the data files are loaded into appropriate media players. For example, the content of a text file is displayed in a small notepad display R1001 on the top left hand side part of the content window. Below that, an internet link R1002 is shown in hypertext format, so that it can immediately be clicked by a user to open the associated URL, without a requirement to open a web browser first. Below the internet link, a date and time is displayed within a calendar application R1003.

On the top right hand side of the screen, a movie file is shown as loaded into a movie player application R1004. User controls are provided within the movie application, e.g. to allow a user to play the movie file. Below the movie file, a sound file is shown as loaded into a sound player application R1005. User controls are provided within this sound player application to allow a user to play, rewind, fast-forward, etc. Below the sound application, a reminder application is provided, which displays a user set reminder message and has a button for a user to click to indicate that the reminder is no longer needed.

Figure 20A shows an enlarged view of the movie player interface of figure 19, with a menu open for selection of a movie file. Figure 20B shows a summary information screen for an MPEG-4 file, giving a summary of the file information for the selected file. The video format, audio format, and various other properties are shown. This summary screen may be activated when a predetermined user input is received by the navigation system in an embodiment of the invention.

Figure 21 shows a further option for a navigation browser to be included in embodiments of the invention. The screen is divided into a left hand side part and a right hand side part. Above these two parts, a menu bar is shown, which includes an action button according to an embodiment of the invention. The right hand side part of the screen shows a hierarchical tree structure indicating the structure of the information hierarchy, and showing the user's currently selected position in the hierarchy as a highlighted node in the tree structure. The left hand side part of the screen shows the contents of the currently selected folder, and in some embodiments, may also be used to display the contents of individual documents if the documents are opened for editing.

User interfaces according to the present invention may be customisable for particular users. Figure 22 shows such an example for two users, "User A" and "User B". When User B has logged in, the user interface allows User B to navigate to the User B folder and displays User B's files and data objects in that folder, as shown in the top screenshot in figure 22. The User B folder may be set up as the default starting folder for navigation, whenever User B logs into the system. However, if User A logs in instead, then the user interface allows User A to navigate to the User A folder and displays User A's files and data objects in that folder, as shown in the top screenshot in figure 22. The User A folder may be set up as the default starting folder for navigation, whenever User A logs into the system.

One advantage of the present invention is that in some embodiments, each user has their own personal information hierarchy structure. Thus, if one user "moves" a file from one folder to another, this only affects that user. This is because it is not the file that moves on the disk, but simply a "shortcut" or list that is updated to change the position of the file in the virtual hierarchy. Thus, if each user has their own personal list or set of

shortcuts, then each user may have a personalised hierarchal “file system”, which is independent of other user’s “file systems”. The user’s personalised hierarchy may allow them to navigate to both shared files and personal files of that user.

Figure 23 shows an alternative screen layout for some embodiments of the invention. Instead of the tab menu described above in the top part of the screen, an address bar is provided. This comprises a box containing the hierarchical path of the presently selected folder, and by clicking on the arrow at the side of this box, the box may be expanded to provide menu options for navigation to parts of the hierarchy above and below the current location.

Figure 24 shows a further variation of the screen layout used in some embodiments of the invention. A layout similar to that of figure 23, apart from the action button, is shown in an individual window. Other windows are also shown, which may have different locations of the hierarchy selected and displayed. At the bottom of the screen, a task bar is shown. The task bar contains a “start” button similar to that used in the Microsoft Windows operating system, and an action button is located in the task bar.

Figure 25 shows yet another variation on the screen layout. In the first screenshot of figure 25, the screen layout is similar to that of figure 3, but a “play” button is provided at the bottom of the screen, in addition to the action button. The play button allows files to be opened, run, played, according to their file type. The action button is switched off in this first screenshot. However, in the second screenshot of figure 25, the action button has been switched on to show the “Move” function. In this embodiment, when the action button is switched on, each icon representing a data object is provided with a check box to indicate selection or non-selection of this icon. Thus, instead of selected items being highlighted by a colour change of their icon, the status of the check box indicates the selection status.

In one embodiment, a check box may indicate selected source objects by showing a check in the check box, and inverted icon colours may show a destination object.

Figure 26 shows a further variation on the user interface, in which icons representing all of the selected items (both in the current folder and in other folders) are shown in a row at the bottom of the screen. This provides a user with a quick visual indication of their selection of data objects.

Figure 27 shows another form of the user interface, in which the selected items are displayed on the screen of a separate handheld controller, e.g. a mobile phone or PDA. This controller has a direction control pad, control buttons and a screen, and may communicate with the navigation system using wireless technology such as Bluetooth or infra-red. The controller may be configured to display other elements of the user interface, such as the action button and/or the menus and/or the data objects available for selection. Figure 27 doesn't show an action button. This is not necessary in all embodiments of the invention. However, when present, an action button may be displayed on the main screen or on the screen of a remote controller. The later option has the advantage of freeing up more screen area for information display on the main screen.

The top part of figure 28 shows a screen layout used in another embodiment of the invention. The action button is not shown in this embodiment, but the screenshot includes an information display area and forward and back navigation buttons. A menu can be activated by clicking on the information display area. The bottom part of figure 28 shows the activated menu, which consists of a plurality of links to data objects, displayed as icons. The links include an address link, a TV link, a DVD ROM link, a music link, a photo link and a message link. The TV link has been selected and highlighted. Again, the back and forward navigation buttons are shown at the bottom of the screen.

Figure 29 shows a full screen view of the information display of figure 28. This is particularly useful for displaying images or videos with the maximum possible size. Figure 29 also shows a remote controller that may be used for user input to the navigation system. The remote controller has back and forward buttons, a menu buttons, and four further navigation or control buttons. In this embodiment, it is unnecessary to

display the back and forward buttons on the screen, because they can simply be activated on the remote control unit.

The first screenshot of figure 30 shows the screen layout of figure 28, just prior to the forward navigation button being pressed and held for a 1 second period. This long click button activation results in a menu display as shown in the bottom screenshot of figure 28. The menu allows a user to set up a priority order for playing or opening files. In this example, the priority order has been set to “My Favourite”, “Live Episode”, “Recent Play”, “AI”. In the fifth priority position, an input field is displayed for user input, to allow a user to select a new item to add to the priority list. User selectable up and down buttons are provided, to allow a user to increase or decrease the priority of a selected item.

Figure 31 shows a hierarchical file organisation structure, for organising files from a plurality of different file stores. A single top level of the hierarchy is not shown, but instead, six main topics are shown on the first level of the hierarchy. These are “Addresses” 1001, “TV” 1004, “DVD ROM” 1029, “Music” 1033, “Photo” 1043 and “Message” 1051. The address topic has two sub topics, i.e. “Boy Friends” 1002 and “Girl Friends” 1003. The TV topic has sub topics of “News” 1005, “Documentary” 1011, “Travel” 1012, “Dramas” 1013, “Sports” 1025 and “Movie” 1026. Some of these TV subtopics have individual channels or programs as sub-sub topics, and individual episodes as sub topics of the channels or programs.

The DVD Rom topic 1029 has individual scenes as sub topics. The Music topic 1033 has individual artists as sub topics, Albums or tracks as sub-sub topics, and albums may have tracks as further sub topics. The Photo topic 1045 has subtopics labelled according to content, e.g. “Party” 1049 and “Landscape” 1050. The Message topic 1051 has the subtopics “Inbox” 1052, “Outbox” 1053 and “Editor” 1054.

Figure 32 shows a series of screenshots for a navigation process through the hierarchy of figure 31 using the forward and back navigation buttons. The first screenshot, located in the top left part of the screen, is the same as the first screenshot of figure 28 discussed above. The four screenshots shown below this first screenshot are similar to the first,

but each showing a different item in the hierarchy. These different items are selected according to the priority setup, as discussed with reference to figure 30. A user may navigate between them by use of the forward and back buttons, which navigate through the priority list.

The screenshots shown on the right hand side of figure 32 are all menu screenshots, similar to that of the bottom part of figure 28, but showing menus for different parts of the information hierarchy. The first menu shows the six main topics of “Addresses” 1001, “TV” 1004, “DVD ROM” 1029, “Music” 1033, “Photo” 1043 and “Message” 1051. This is the default menu from a start point at which no navigation path has been selected. The user can obtain this menu by clicking on the screen from the top left screenshot of figure 32. If the user deactivates this menu, the next priority screen will be displayed, i.e. the second top screenshot on the left of figure 32. If the user selects the “TV” option in the menu, the screen display will change to the second top screenshot on the right of figure 32, which shows the six subtopic for “TV”. A further selection of “Drama” from this menu results in the screenshot below being displayed, giving menu options for “Drama”. When “Jinny” is selected, the screenshot on the bottom right of figure 32 is shown, giving the options of episodes 1 to 4. On selection of episode 2, the screen display shows the screenshot at the bottom left of figure 32, with episode 2 of the Jinny program.

Figure 33 is a diagram showing multiple screenshots for a navigation process, using an alternative screen layout. The left side of the figure shows screenshots of video clip introduction screens. The right hand side of the figure shows screenshots of menu displays. The difference between the menus of figure 33 and those of figure 32 is that the menus in figure 33 are roll-down menus which are activated by selection of an item in a menu bar at the top of the screen. The menus each consist of a plurality of menu options displayed on a grid, which is displayed on top of the existing screen display.

The display may show a line-by-line format when program lists rather than program categories are shown. This line-by-line format may show further information on each line, such as the date and time of original broadcast.

Figure 34 is a diagram showing multiple screenshots for a navigation process, using another alternative screen layout. The menu screenshots on the right hand side of the figure are similar in layout to those of figure 33, except that they show icons in addition to the names of menu items in the grid. On the left hand side information display screenshots, a large icon and filename may be displayed in place of the initial video image discussed for figures 32 and 33.

Figure 35 shows an example of a user input scheme which may be used in the present invention. A person's hand is shown, where each finger controls a different function of the user interface. In this example, the forefinger controls the select option, the middle finger controls the action function, the ring finger controls the undo function, the thumb controls a first user defined option, and the little finger controls a second user defined option. This user input scheme may be implemented by the use of touch sensitive pads positioned against a person's fingers. In one embodiment, this comprises a glove with touch sensitive pads in the fingers. In another embodiment, a keypad is provided with five buttons arranged in a shape that all five fingers may simultaneously touch a different button. In a yet further embodiment, touch sensitive pads are not required, and a camera detects the user's hand movements and sends the hand movement information to a control system where it is interpreted as particular user inputs. Other embodiments may use any other type of suitable implementation of the hand input method.

The present invention may be combined with a navigation system configured with additional user-operable navigation controls. Icons may be provided on the screen, e.g. in the area next to the action button, to navigate through an information structure according to a selected navigation mode. Navigation modes may include scrolling through all items in a single level of the information structure, or following a path downwards through the information structure, etc. For example, a back button, a forward button and a mode switch button may be displayed at the bottom part of the screen. These control buttons relate to back navigation, forward navigation, and mode selection respectively. The first two control button areas 3, 4 show left and right arrows to indicate a "back" 6 and a "forward" 7 control respectively. The third control button area 5 shows a navigation mode symbol 8 to pictorially indicate the current navigation mode.

The user input device may be adapted to move a pointer, and the display controller may be adapted to control the display to display new sub-menu items when a pointer is moved over a menu item. The display controller may be adapted to control the display to display menu items in allocated areas of the display arranged in a series. The display controller may be adapted to control the display to display menu items arranged along an upper area of the display. The display controller may be adapted to control the display to display menu items overlaying the information displayed on the display. These menu items may temporarily replace the information displayed on the display.

The present invention can be implemented in dedicated hardware, using a programmable digital controller suitably programmed, or using a combination of hardware and software. Thus the present invention can be embodied by any suitable carrier medium for carrying machine readable instructions for controlling one or more programmable controllers. The carrier medium can comprise any storage medium such as a floppy disk, hard disk, CD ROM, magnetic tape, or programmable memory device, or a transient medium such as an electrical, optical, microwave, RF, electromagnetic, acoustic or magnetic signal. An example of such a signal is an encoded signal carrying computer code over a communications network, e.g. a TCP/IP signal carrying computer code over an IP network such as the Internet, an intranet, or a local area network.

Although specific embodiments of the invention have been described, further modifications are also possible. The code for each process in the methods according to the invention may be modular, or may be arranged in an alternative way to perform the same function. The methods and apparatus according to the invention are applicable to any computer with display means, including a handheld device.

Although embodiments of the invention are discussed in terms of a filing system in which data objects are not physically moved, but an index or label is updated to define their position in an information structure, the invention can also have utility in other types of filing system, e.g. a filing system composed of directories and subdirectories defining an information structure. Thus, embodiments of the invention include systems of this type.

While the invention has been described in terms of what are at present its preferred embodiments, it will be apparent to those skilled in the art that various changes can be made to the preferred embodiments without departing from the scope of the invention, which is defined by the claims.

CLAIMS:

1. A user interface apparatus for management of stored data objects, the apparatus comprising:

retrieving means for retrieving stored data objects from storage means;

user input receiving means for receiving a user selection of a data object management function, a user selection of one or more data objects, and a function activation user input to implement a selected data object management function; and

control means for controlling at least one display to display a representation of data objects and to display an identification of a user selected data management function; and for implementing said user selected data object management function using said user selected data objects in response to said function activation user input.

2. The apparatus of claim 1, further comprising storage means for storing associations or links between data objects, the associations or links defining a virtual file system.

3. The apparatus of claim 2, wherein said storage means for storing associations or links is configured to store associations or links of a data file with other data objects, to allow said data file to represent a folder in the virtual file system.

4. The apparatus of any previous claim, wherein the control means is configured to control the display to display said representation of data objects in a first display area, and to display said identification of a user selected data management function in a second display area separate to the first display area.

5. The apparatus of claim 4, wherein the control means is configured to control the display to display a third display area for displaying a navigation menu with menu items representing data objects.

6. The apparatus of claim 4 or claim 5, wherein the control means is configured to control the display to display, in response to a first user selection of a data object

management function, a menu of available data object management functions for user selection.

7. The apparatus of any previous claim, wherein the control means is configured to implement a data object management function as either of a file management operation and a file editing operation, according to the data objects that are selected when the data object management function is implemented .

8. The apparatus of any previous claim, wherein the control means is configured to implement a file editing function using a selected data object, with no effect on a data file in said data store corresponding to said selected data object, and the controller is configured to generate a new data object comprising an edited version of said data file in response to implementation of said function.

9. The apparatus of any previous claim, wherein the control means is configured to implement a file management function using a selected data object, with no effect on a data file in said data store corresponding to said selected data object, and the controller is configured to generate or modify a second data object comprising links or associations defining relationships between the selected data object and additional data objects.

10. The apparatus of any previous claim, wherein said input is configured to receive a user selection of a plurality of destination data objects representing destination locations in an information structure, and the control means is configured to implement said selected function for each of the plurality of destination data objects in response to a single function activation user input.

11. The apparatus of any previous claim, wherein said interface is configured to receive data objects from a data store comprising data objects organised in a hierarchical information structure.

12. The apparatus of any previous claim, wherein said input is configured to receive user navigation commands for navigation through a plurality of stored data objects, and

wherein the controller is configured to control said display to display a representation of the data objects that are navigated to, using said user navigation commands.

13. The apparatus of claim 12, wherein the control means is configured to control the display to display a history menu comprising a representation of a plurality of previously selected data objects, for user selection of one or more of said previously selected data objects.

14. The apparatus of claim 12 or claim 13, further comprising a data store for storing a user editable navigation priority list defining a predetermined order for navigation through the plurality of data objects.

15. The apparatus of any one of claims 12 to 14, wherein the control means is configured to select one of a plurality of navigation strategies according to a user navigation mode selection received at said input, each said navigation strategy defining an order for navigation through the plurality of data objects.

16. Apparatus as claimed in any previous claim, wherein the control means is configured to control the display to display a first part of the displayed information on a first display, and a second part of the displayed information on a second display, wherein the second display is smaller than the first display, and the second display is provided on a remote controller, said remote controller providing user input signals to said input.

17. A user interface apparatus for management of stored data objects, the apparatus comprising:

an interface for retrieving stored data objects from at least one data store;

an input device for receiving a user selection of a data object management function, a user selection of one or more data objects, and a function activation user input to implement a selected data object management function; and

a controller connected to the interface and the input device for controlling at least one display to display a representation of data objects and to display an identification of a user selected data management function; and for implementing said

user selected data object management function using said user selected data objects in response to said function activation user input.

18. The apparatus of claim 17, further comprising a data store for storing associations or links between data objects, the associations or links defining a virtual file system.

19. The apparatus of claim 18, wherein said data store for storing associations or links is configured to store associations or links of a data file with other data objects, to allow said data file to represent a folder in the virtual file system.

20. The apparatus of any one of claims 17 to 19, wherein the controller is configured to implement a data object management function as either of a file management operation and a file editing operation, according to the data objects that are selected when the data object management function is implemented .

21. The apparatus of any one of claims 17 to 20, wherein the controller is configured to implement a file editing function using a selected data object, with no effect on a data file in said data store corresponding to said selected data object, and the controller is configured to generate a new data object comprising an edited version of said data file in response to implementation of said function.

22. The apparatus of any one of claims 17 to 21, wherein the controller is configured to implement a file management function using a selected data object, with no effect on a data file in said data store corresponding to said selected data object, and the controller is configured to generate or modify a second data object comprising links or associations defining relationships between the selected data object and additional data objects.

23. The apparatus of any one of claims 17 to 22, wherein said input is configured to receive a user selection of a plurality of destination data objects representing destination locations in an information structure, and the controller is configured to implement said

selected function for each of the plurality of destination data objects in response to a single function activation user input.

24. The apparatus of any one claims 17 to 23, further comprising a data store for storing a user editable navigation priority list defining a predetermined order for navigation through the plurality of data objects.

25. The apparatus of any previous claim, wherein the data objects comprise representations of hardware components, and the data object management function comprises a function for representing a plurality of data objects as a combination of hardware components in a hardware device.

26. The apparatus of claim 25, wherein the control means is configured to store compatibility information on combinations of hardware components, and wherein the data object management function is configured to indicate the compatibility of a selected combination of components.

27. The apparatus of claim 25 or claim 26, wherein the control means is configured to store historical information on the addition and removal of hardware components, and wherein a data object management function is provided to allow a user to view the historical information.

28. Apparatus for navigation of information, the apparatus comprising
storage means for storing data objects, wherein each data object represents a data file and is configured to store associations between the data file and one or more other data files;

means for receiving user navigation commands from a user input device, said user navigation commands for user navigation through a plurality of stored data objects, said user navigation commands including data object selection, and

control means for displaying, in response to a received data object selection, representations of other data files associated with said data file that is represented by the selected data object.

29. Apparatus for navigation of information, the apparatus comprising
a data store for storing data objects, wherein each data object represents a data file and is configured to store associations between the data file and one or more other data files;
an input device for receiving user navigation commands from a user input device, for user navigation through a plurality of stored data objects, said user navigation commands including data object selection, and
a controller connected to the data store and the input device for displaying, in response to a data object selection received at the input, representations of other data files associated with said data file that is represented by the selected data object.
30. Apparatus for navigation and browsing of information, the apparatus comprising
processing means for generating a hierarchical structure of data file shortcuts, at least some of the shortcuts including links to data files at another level of the hierarchy;
and
storage means for storing said hierarchical structure of data file shortcuts.
31. Apparatus for navigation and browsing of information, the apparatus comprising
a processor for generating a hierarchical structure of data file shortcuts, at least some of the shortcuts including links to data files at another level of the hierarchy; and
a data store for storing said hierarchical structure of data file shortcuts.
32. Apparatus for navigation and browsing of information, the apparatus comprising
retrieving means for retrieving stored data objects from storage means;
user input receiving means for receiving a user selection of a data object management function, and a user selection of one or more data objects; and
control means for controlling at least one display to display a representation of data objects and to display an identification of a user selected data management function; and for implementing said user selected data object management function using said user selected data objects;
wherein at least one of the data object management functions is configured to perform either of a file management operation and a file editing operation, according to which data objects are currently selected.

33. Apparatus for navigation and browsing of information, the apparatus comprising an interface for retrieving stored data objects from at least one data store; an input device for receiving a user selection of a data object management function, and a user selection of one or more data objects; and a controller connected to the interface and the input device for controlling at least one display to display a representation of data objects and to display an identification of a user selected data management function; and for implementing said user selected data object management function using said user selected data objects; wherein at least one of the data object management functions is configured to perform either of a file management operation and a file editing operation, according to which data objects are currently selected.
34. A user interface apparatus for management of stored data objects, the apparatus comprising:
an data input for retrieving stored data objects from data storage;
a user input device for receiving a user selection of a data object management function, a user selection of one or more data objects, and a function activation user input to implement a selected data object management function;
a display controller for controlling at least one display to display a representation of data objects and to display an identification of a user selected data management function; and
a processor for implementing said user selected data object management function using said user selected data objects in response to said function activation user input.
35. A handheld device comprising a screen, a user input device, and the apparatus of any one of the previous claims.
36. A carrier medium carrying code for configuring a programmable apparatus as the apparatus of any preceding claim.

37. A method of operating a user interface apparatus for management of stored data objects, the method comprising:

controlling at least one display to display a representation of data objects for selection by a user;

receiving a user selection of one or more data objects and a user selection of a data object management function;

controlling said at least one display to display an identification of the user selected data management function;

receiving a function activation user input for implementing a selected data object management function; and

operating on said user selected data objects with said user selected data object management function in response to said function activation user input.

38. The method of claim 37, further comprising storing associations or links between data objects, the associations or links defining a virtual file system.

39. The method of claim 38, wherein storing associations or links includes storing associations or links of a data file with other data objects, to allow said data file to represent a folder in the virtual file system.

40. The method of any one of claims 37 to 39, further comprising displaying said representation of data objects in a first display area, and displaying said identification of a user selected data management function in a second display area separate to the first display area.

41. The method of claim 40, further comprising displaying a navigation menu with menu items representing data objects, in third display area.

42. The method of claim 40 or claim 41, further comprising displaying, in response to a first user input for selection of a data object management function, a menu of available data object management functions for user selection.

43. The method of any one of claims 37 to 42, wherein at least one of the data object management functions is configured to perform either of a file management operation and a file editing operation, according to which data objects are currently selected.

44. The method of any one of claims 37 to 43, wherein implementation of a file editing function on a data object has no effect on the corresponding stored data file, but creates a new data object comprising an edited version of said data file in response to implementation of said function.

45. The method of any one of claims 37 to 44, wherein implementation of a file management function on a selected data object has no effect on the corresponding stored data object, but generates or modifies a second data object comprising links or associations between the selected data object and additional data objects.

46. The method of any one of claims 37 to 45, further comprising receiving a user selection of a plurality of destination data objects representing destination locations within an information structure, and implementing said selected function for each of the plurality of destination data objects in response to a single function activation user input.

47. The method of any one of claims 37 to 46, wherein said data objects are organised in a hierarchical information structure.

48. The method of any one of claims 37 to 47, further comprising receiving user navigation commands for navigation through the plurality of stored data objects, and displaying a representation of data objects that are selected by said user navigation commands.

49. The method of claim 48, further comprising displaying a history menu comprising a representation of a plurality of previously selected data objects, for user selection of one or more of said previously selected data objects.

50. The method of claim 48 or claim 49, further comprising storing a user editable navigation priority list, defining an order for navigation through the plurality of data objects.

51. The method of any one of claims 48 to 50, further comprising selecting one of a plurality of navigation strategies according to a user navigation mode selection, each said navigation strategy defining an order for navigation through the plurality of data objects.

52. The method as claimed in any one of claims 37 to 51, further comprising sending first display control information to a first display and sending second display control information to a second display, wherein the second display is smaller than the first display, and the second display is provided on a remote controller.

53. The method of any one of claims 37 to 52, wherein the data objects comprise representations of hardware components, and the data object management function comprises a function for representing a plurality of data objects as a combination of hardware components in a hardware device.

54. The method of claim 53, wherein the control means stores compatibility information on combinations of hardware components, and wherein the data object management function is configured to indicate the compatibility of a selected combination of components.

55. The method of claim 53 or claim 54, wherein the control means stores historical information about the addition and removal of hardware components, and wherein a data object management function is provided to allow a user to view the historical information.

56. A method of navigation of information, comprising
storing data objects, wherein each data object represents a data file and includes associations between the data file and one or more other data files;

receiving user navigation commands from a user input device, for user navigation through a plurality of stored data objects, said user navigation commands including data object selection, and

displaying, in response to a received data object selection, representations of other data files associated with a data file represented by the selected data object.

57. A method of navigation and browsing of information, comprising generating a hierarchical structure of data file shortcuts, at least some of the shortcuts including links to data files at another level of the hierarchy; and storing said data file shortcuts in at least one data store.

58. A method of navigation and browsing of information, comprising receiving a user selection of a data object management function, and a user selection of one or more data objects; controlling at least one display to display a representation of data objects and to display an identification of a user selected data management function; and implementing said user selected data object management function using said user selected data objects; wherein said implementation of the data object management function comprises performing either of a file management operation and a file editing operation, according to which data objects are selected when the data object management function is implemented.

59. A carrier medium carrying code for configuring a programmable apparatus to perform the method of any one of claims 37 to 58.

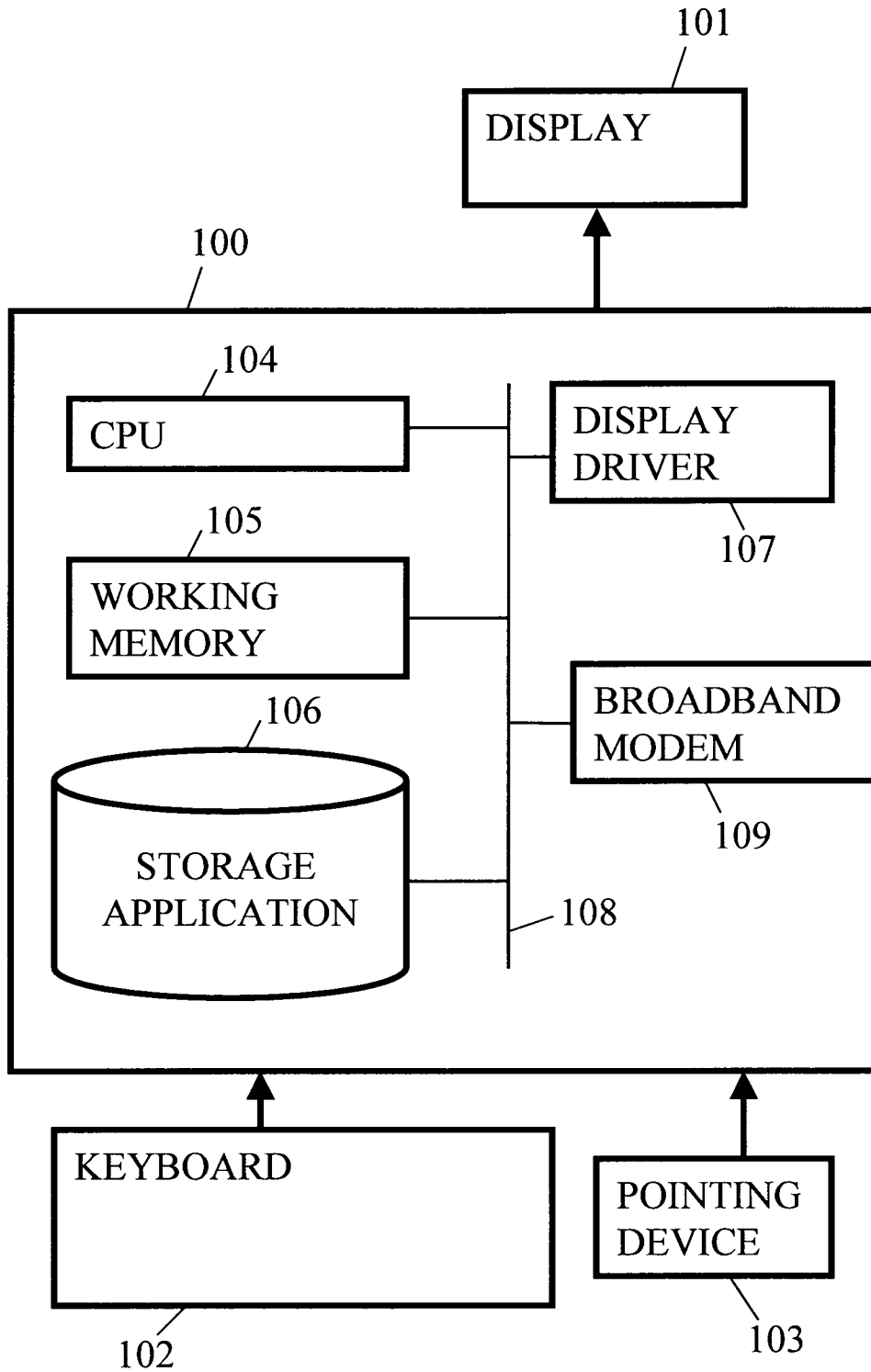


Figure 1

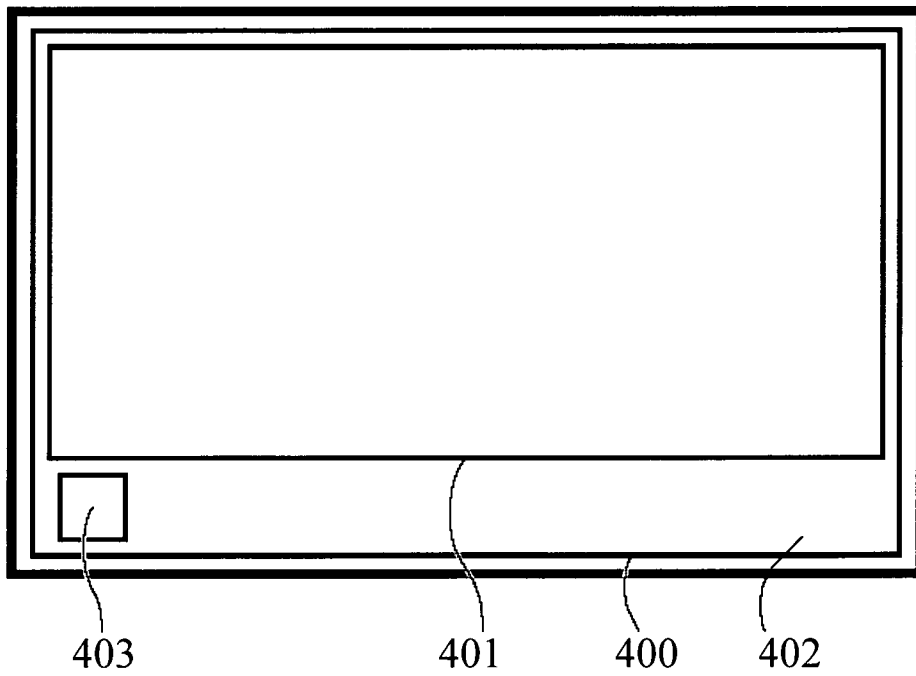


Figure 2

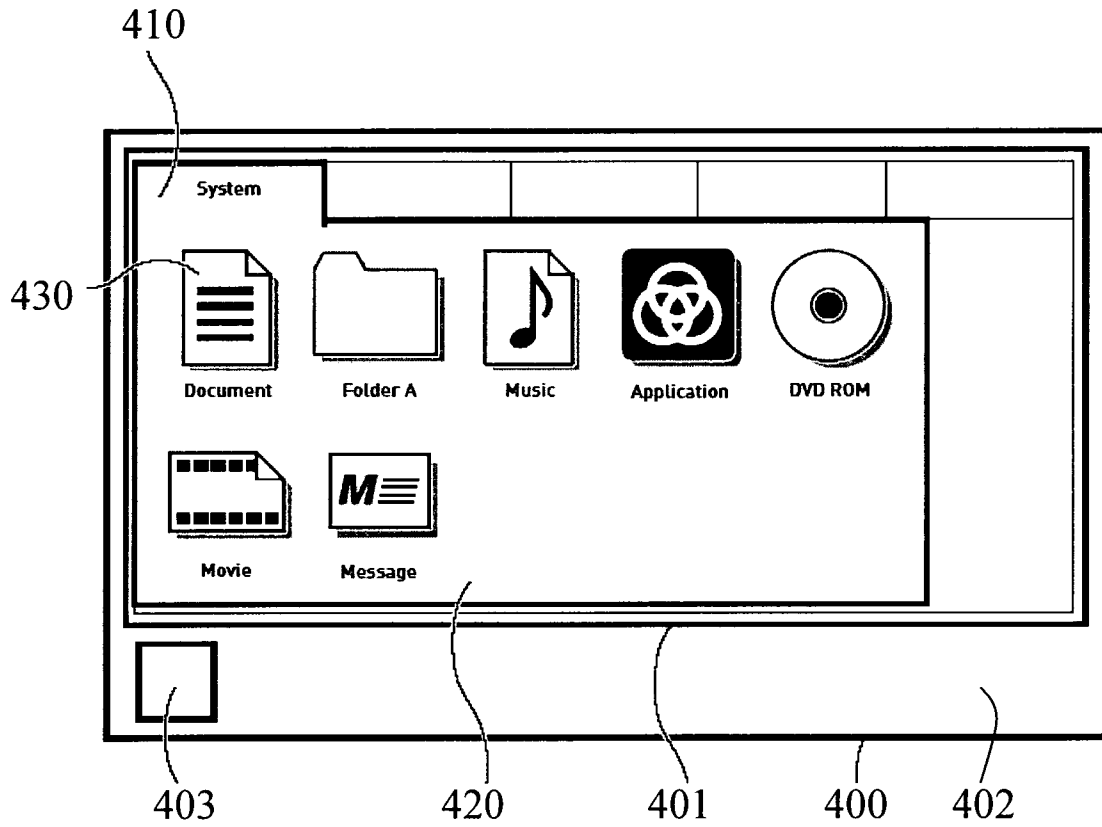


Figure 3

4/46

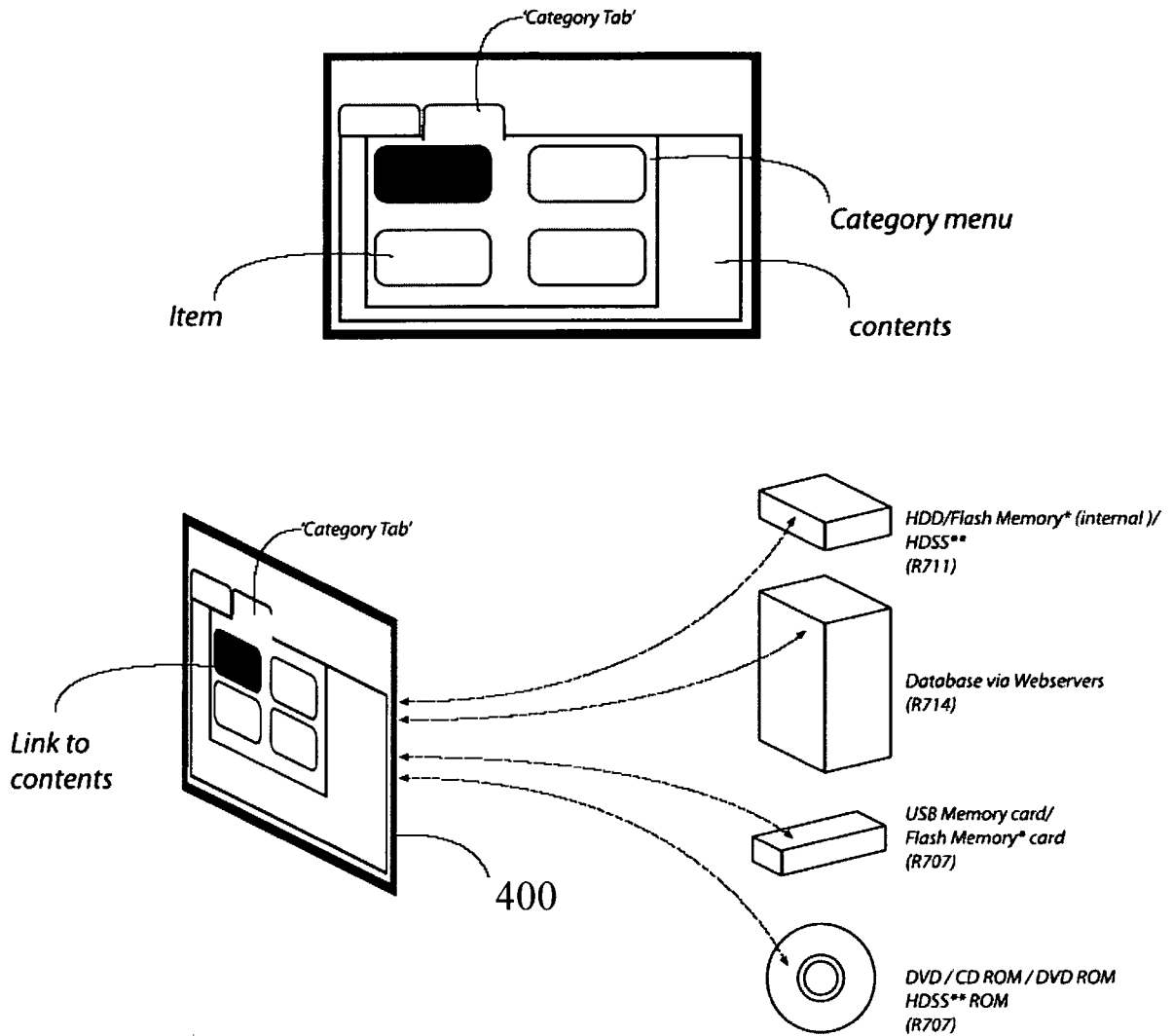


Figure 4

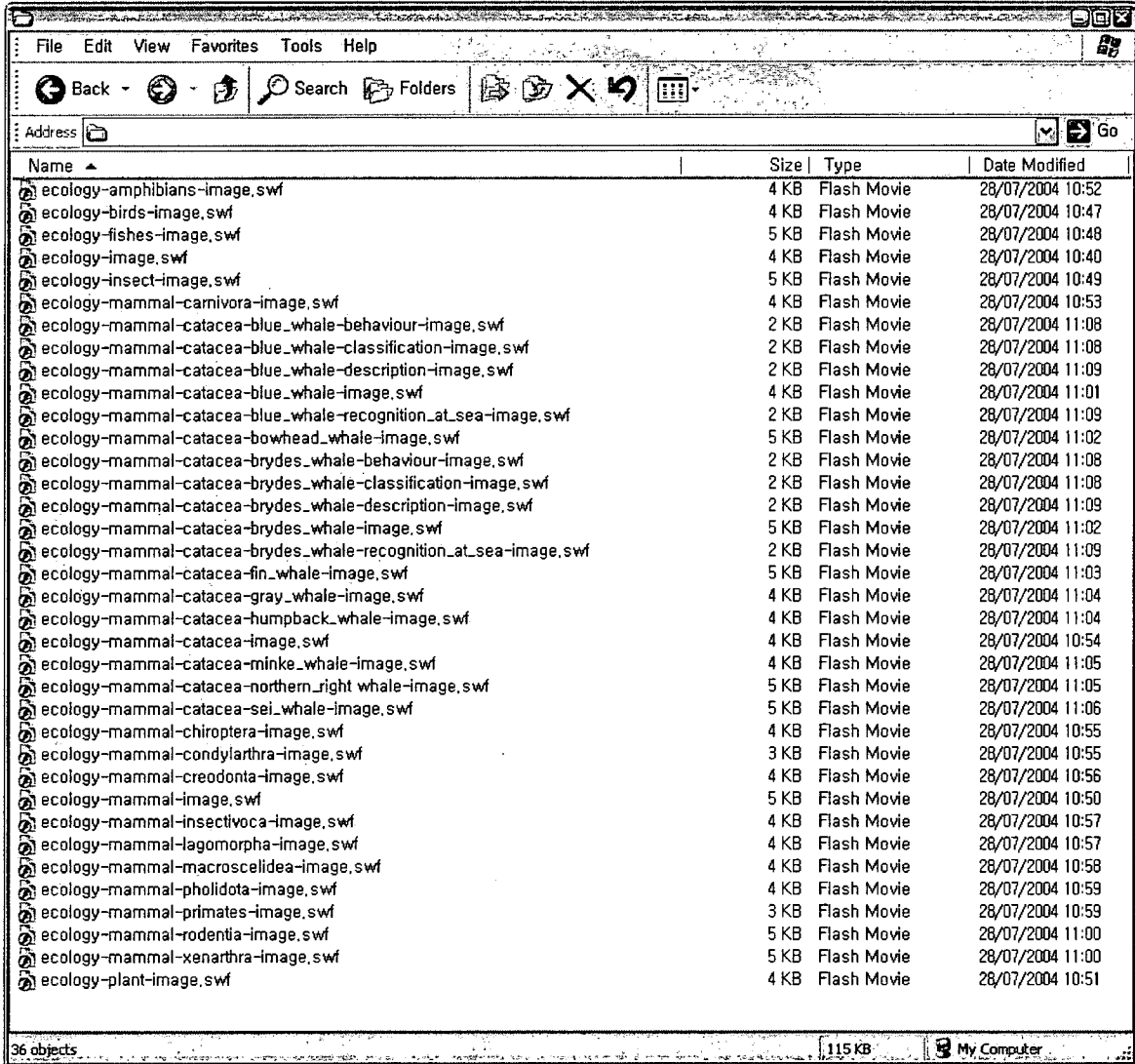


Figure 5

6/46

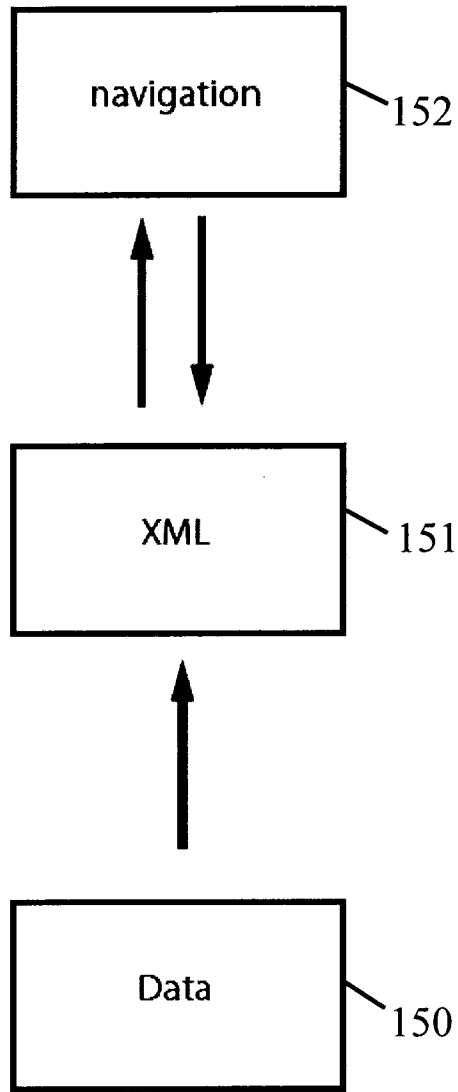


Figure 6

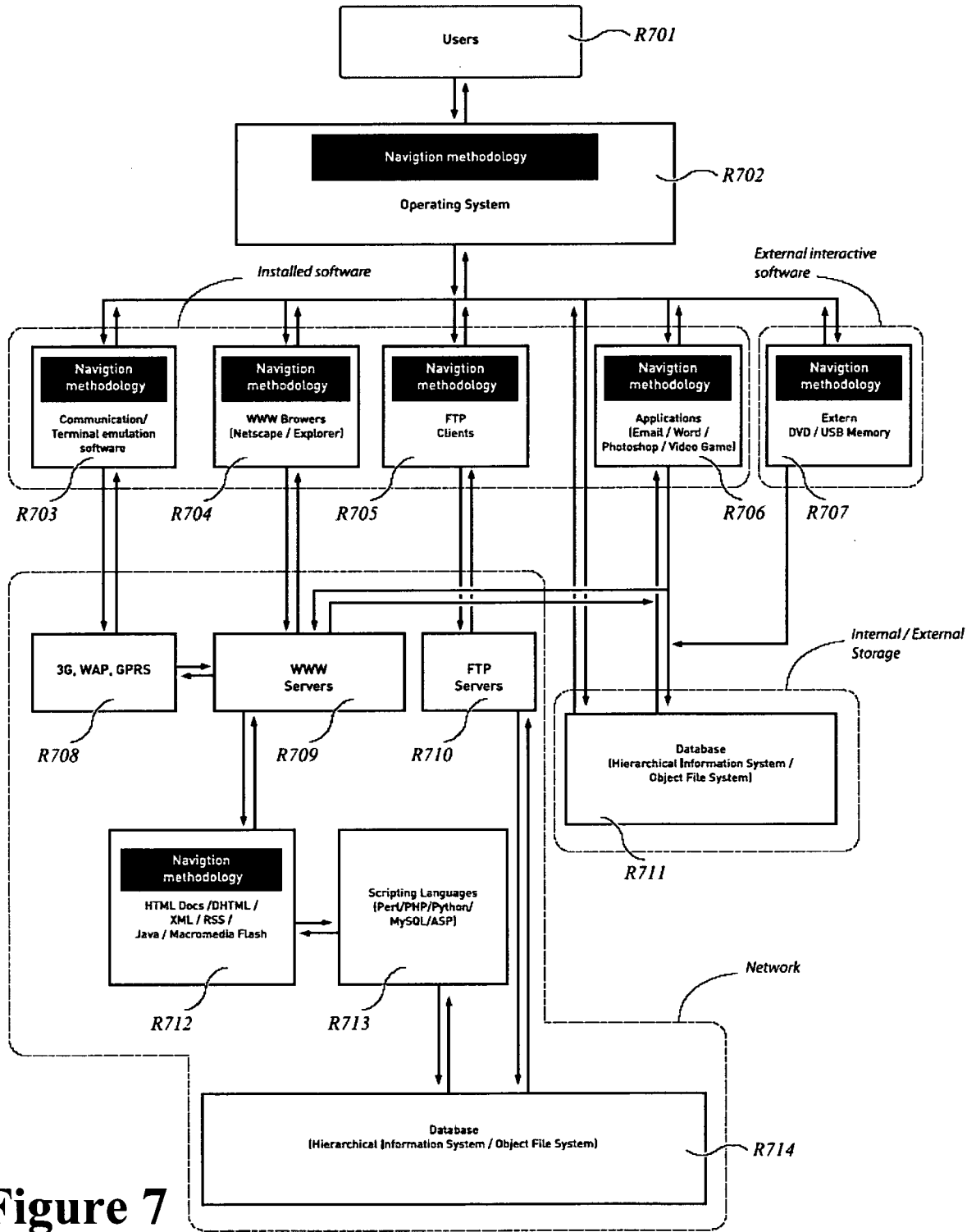


Figure 7

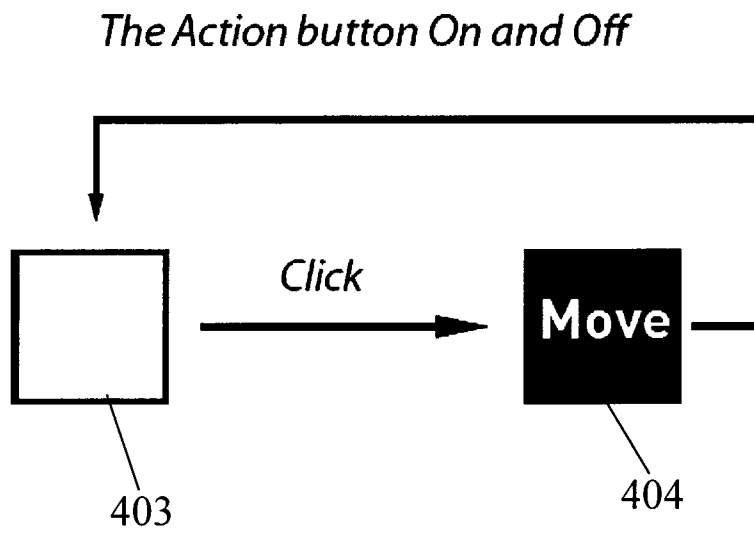


Figure 8

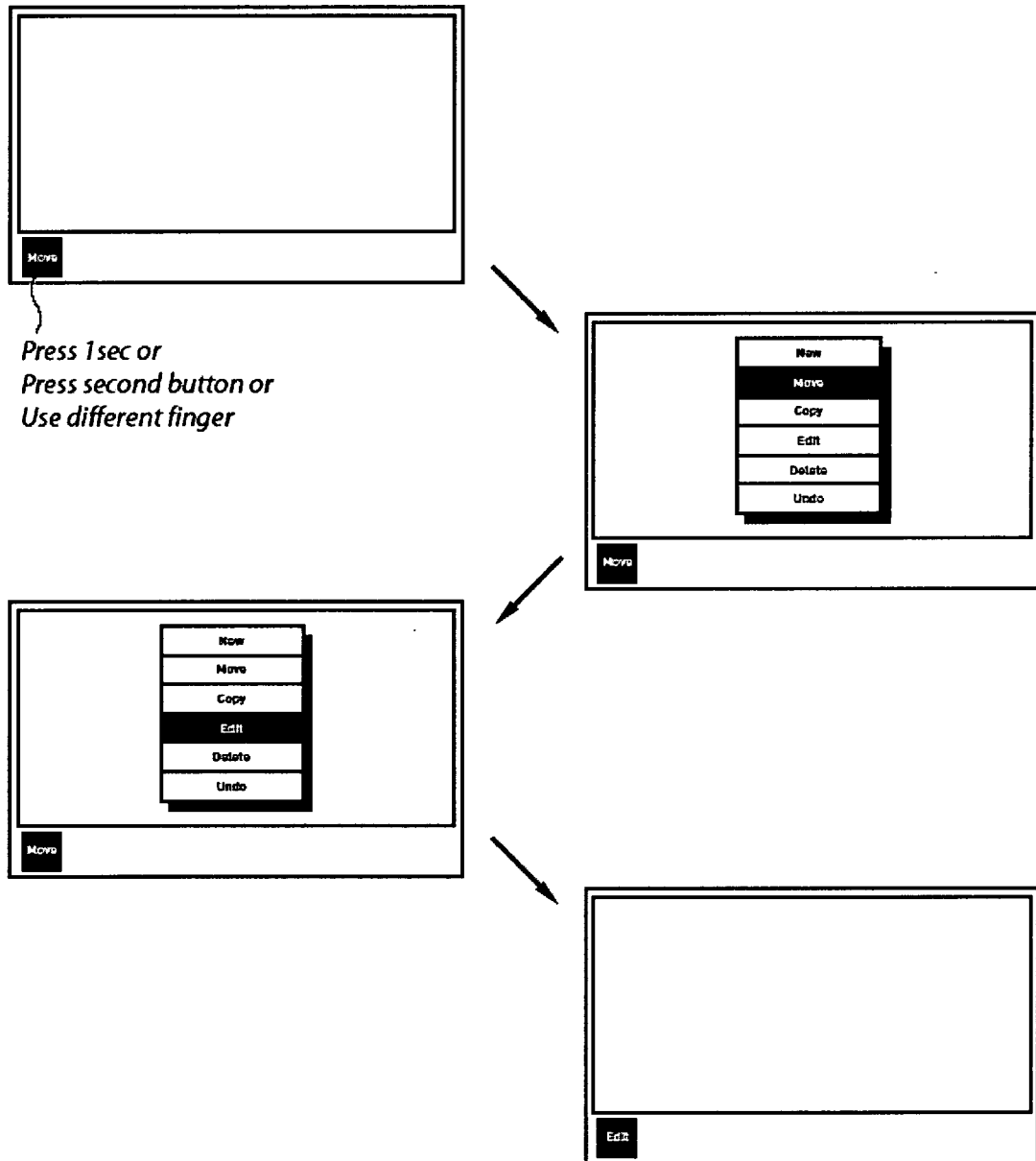


Figure 9

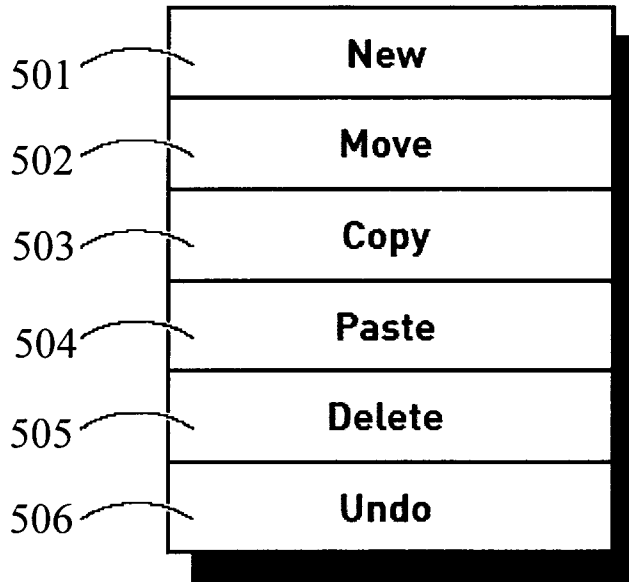


Figure 10A

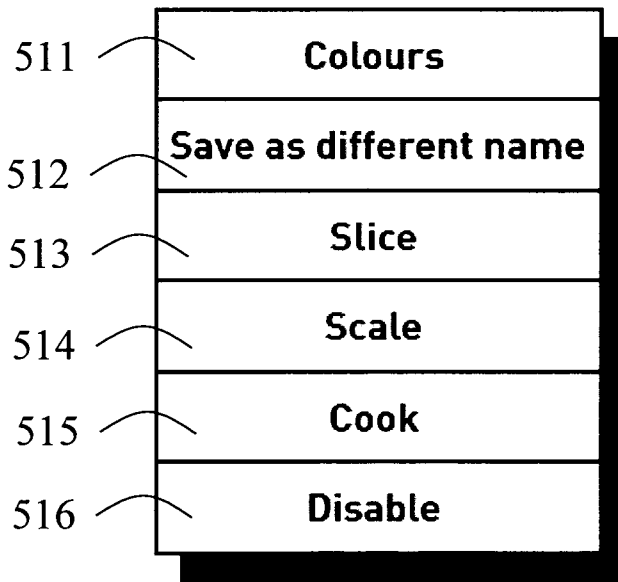
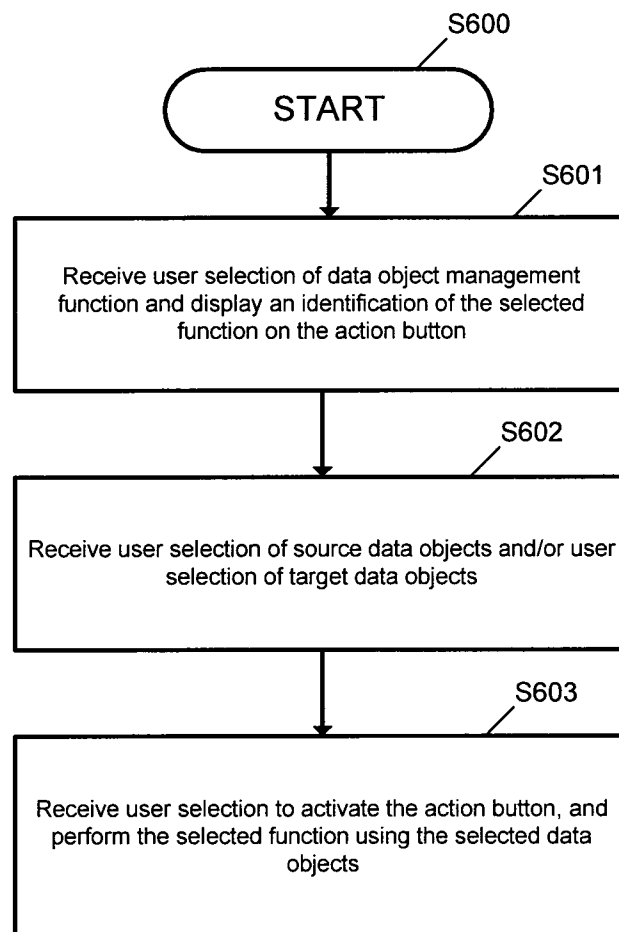


Figure 10B

11/46

**Figure 11**

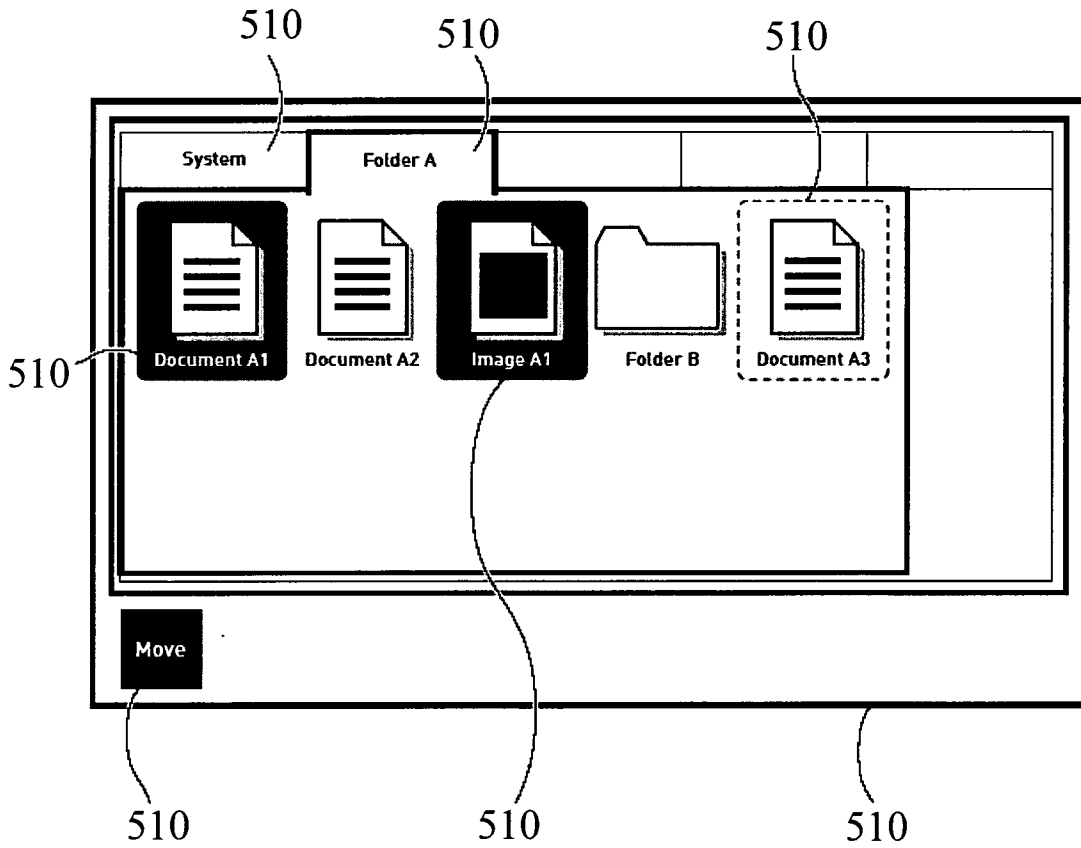


Figure 12

13/46

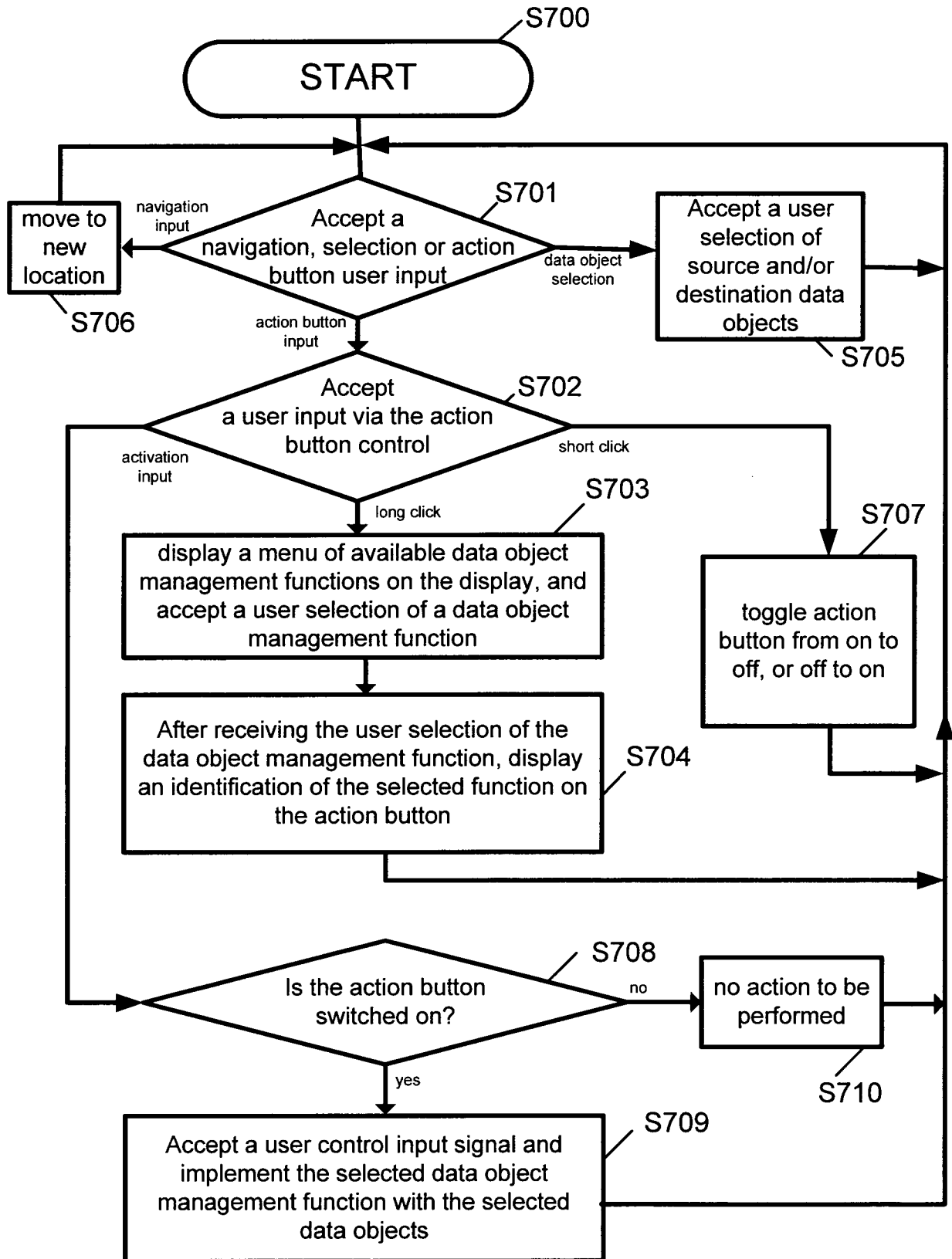
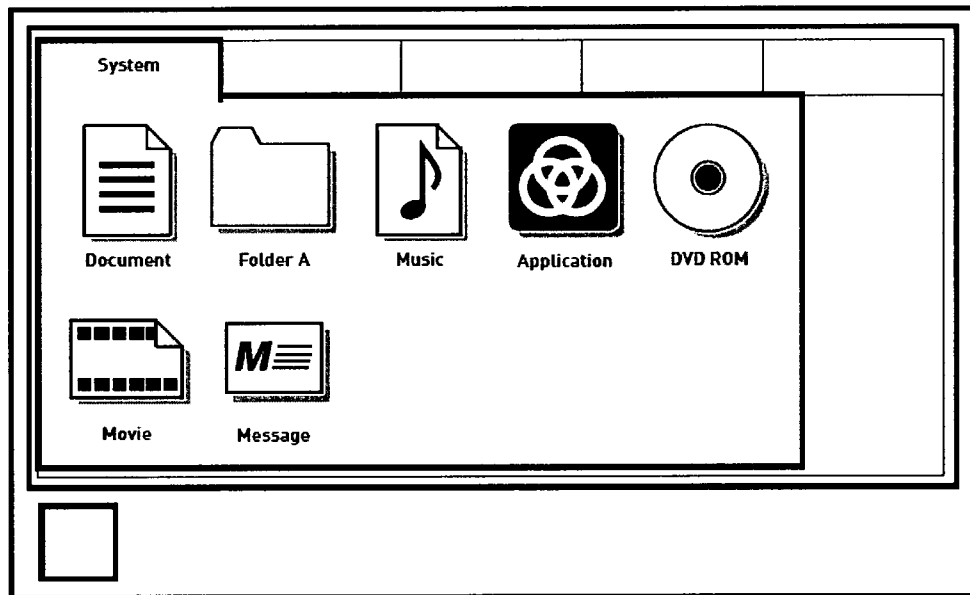
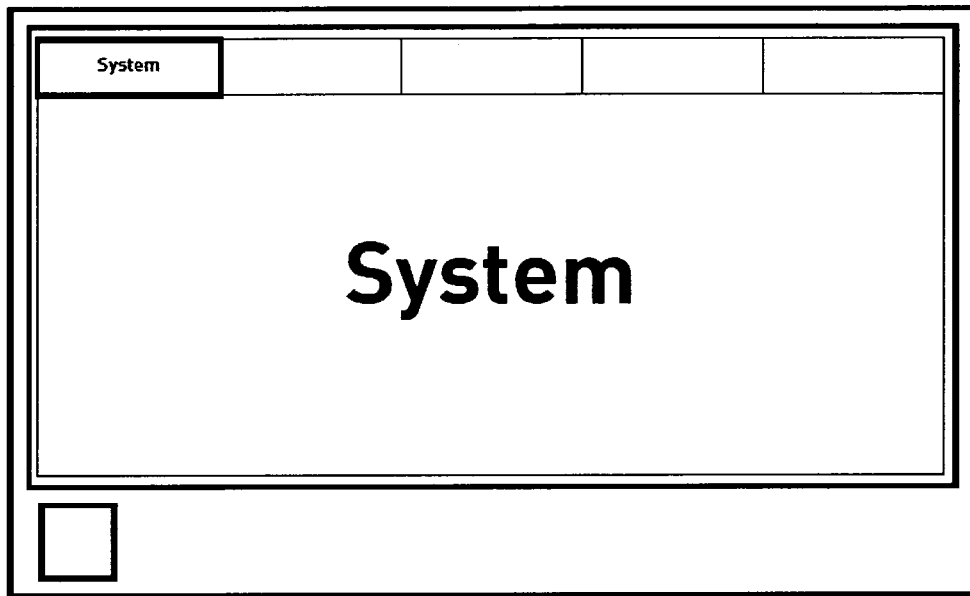


Figure 13

14/46

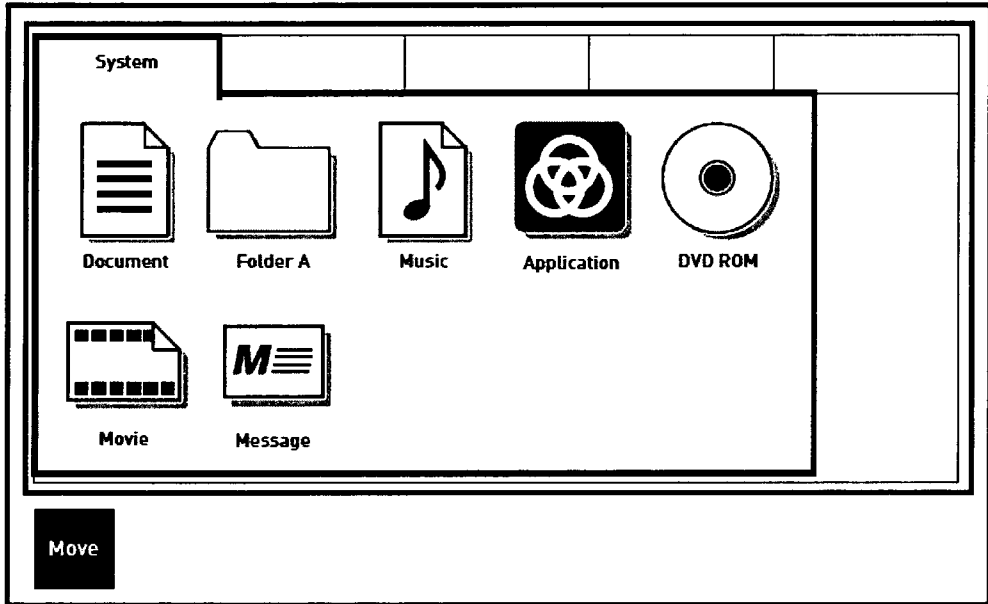


Open the category menu

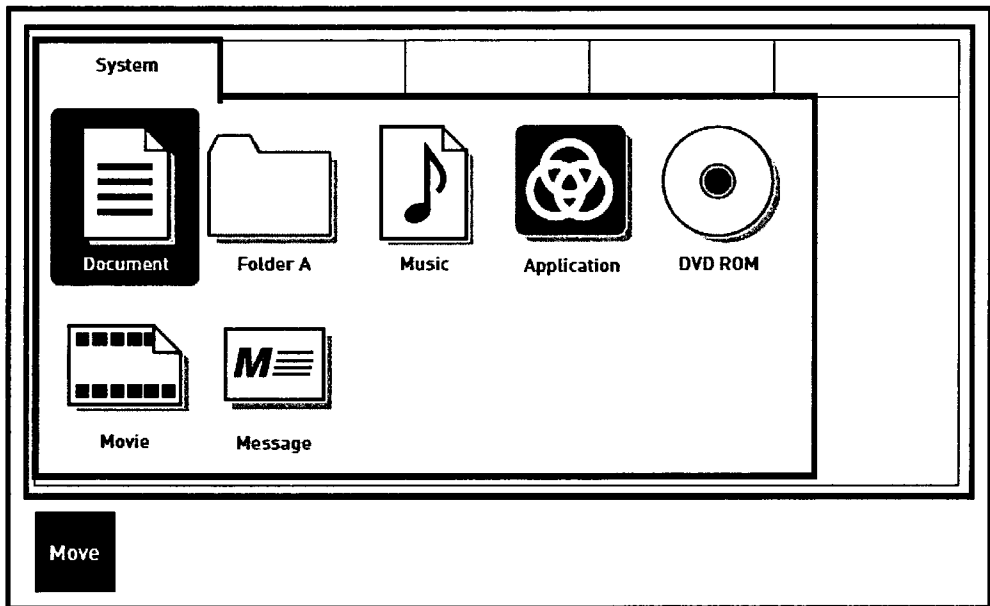


Figure 14A

15/46



Turn on the Action button

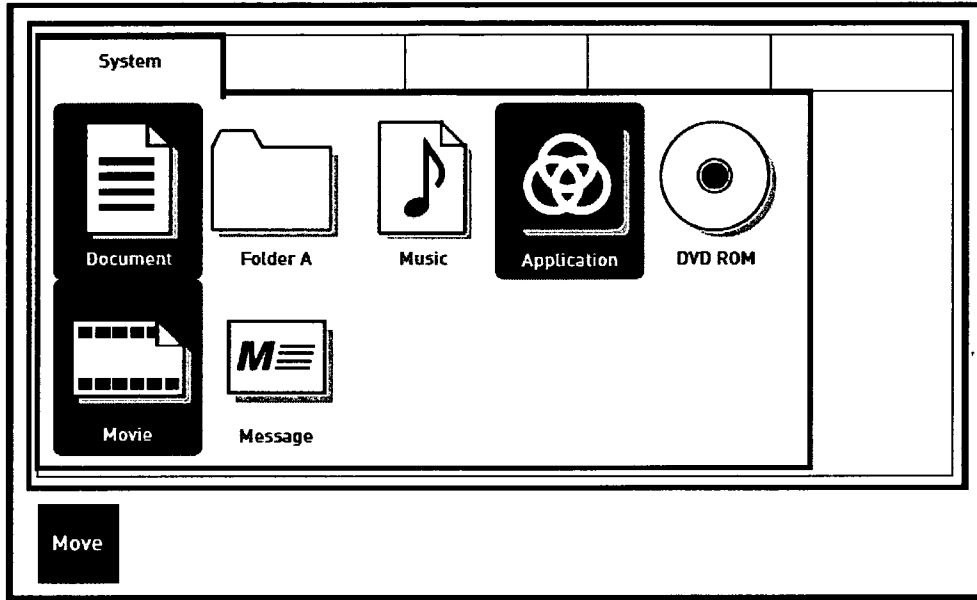


Select item

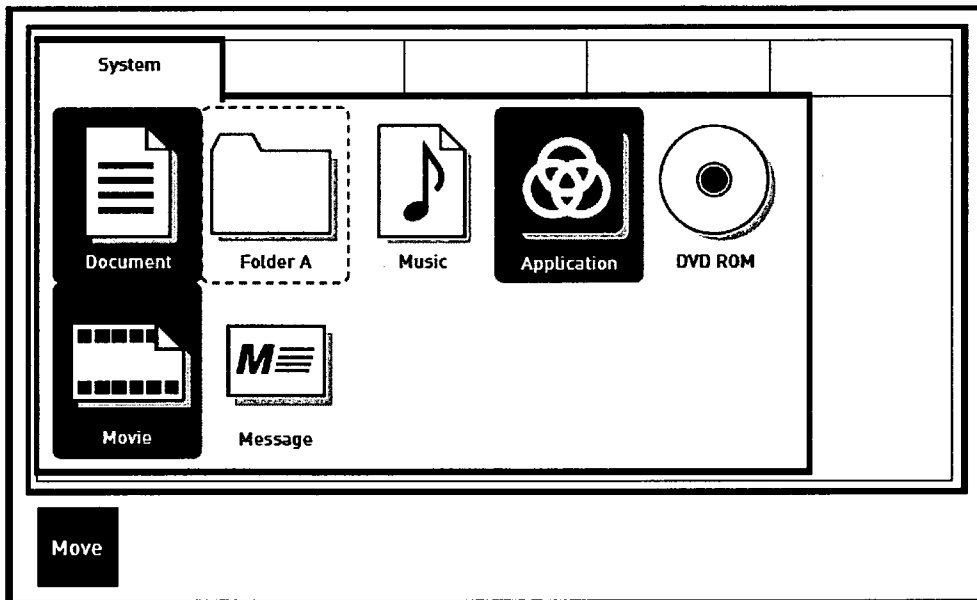


Figure 14B

16/46



Select additional items

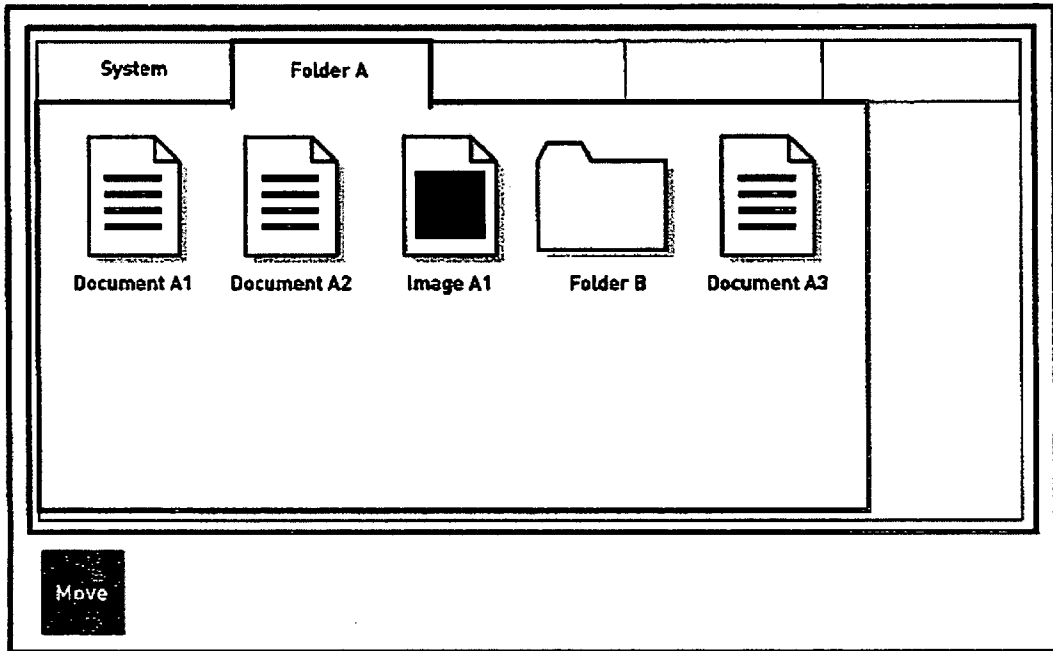


Double click or
Press 1 sec or
Press second button or
Use different finger to
Activate category tab of Folder A



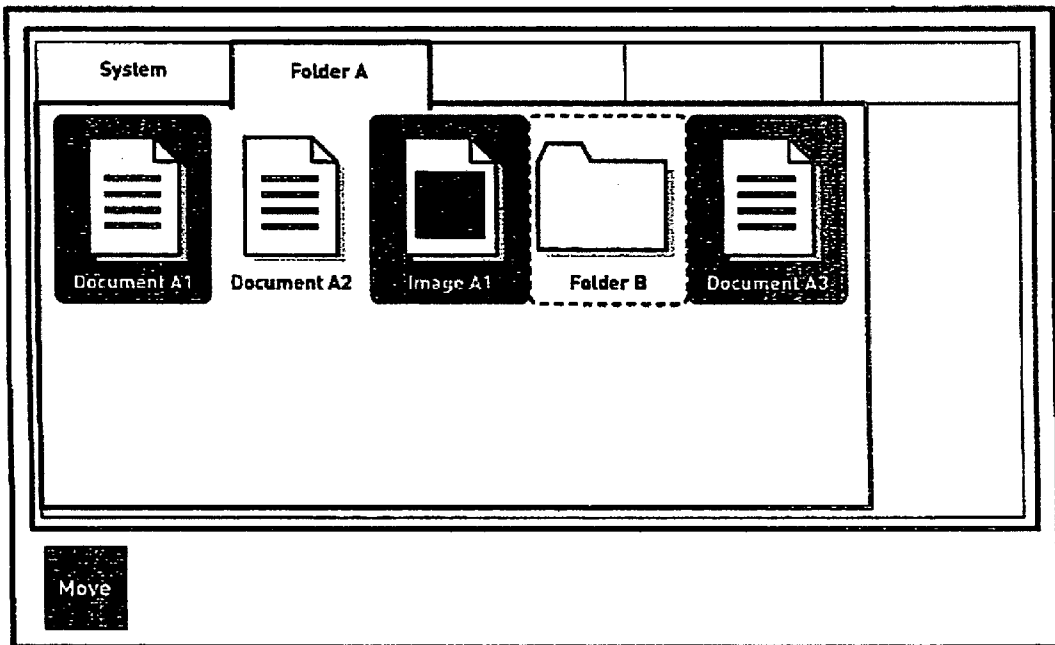
Figure 14C

17/46 Y3
↓



Open the category menu of Folder A

↓ → X1



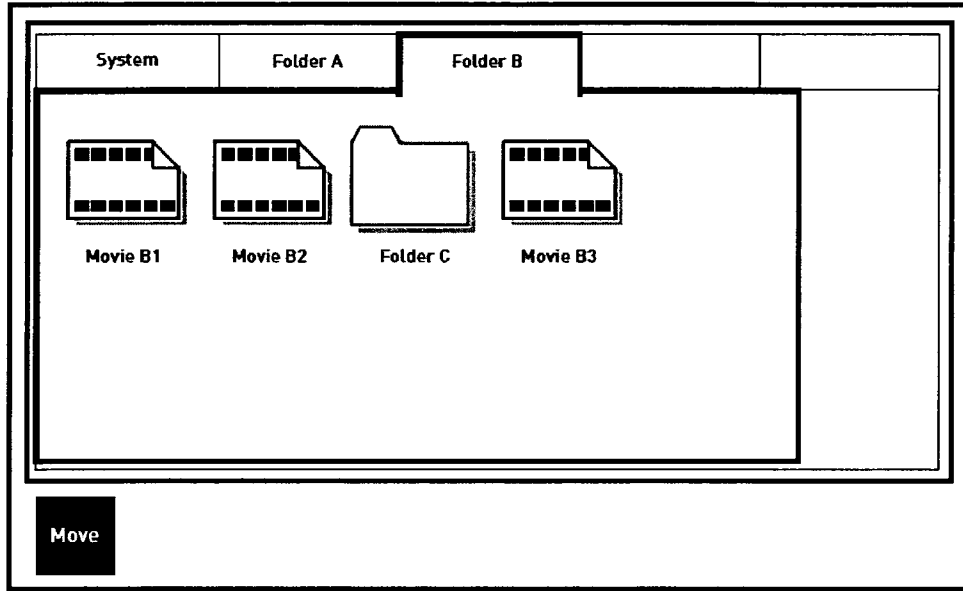
Select additional items in category of Folder A

and

↓ Y4

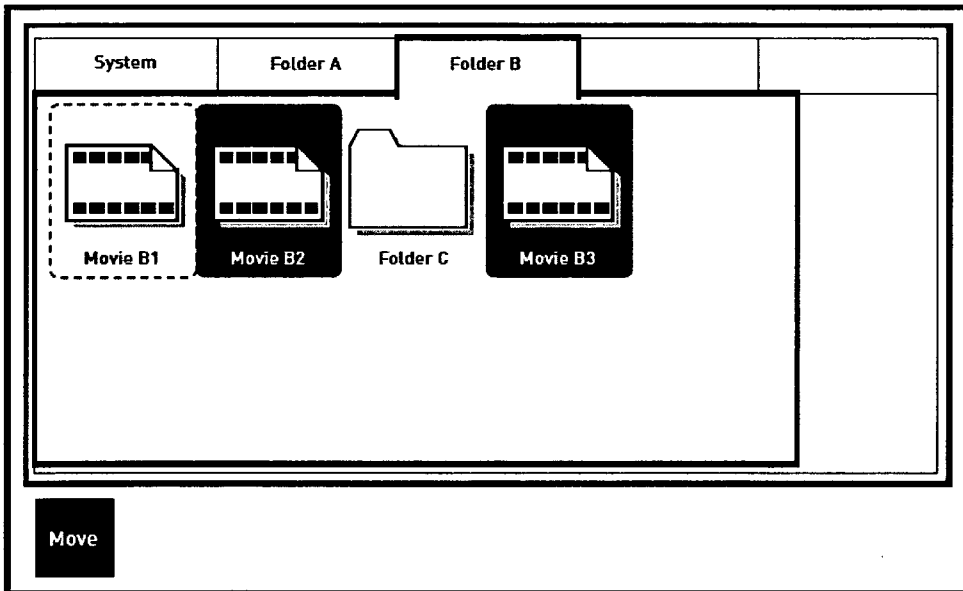
Figure 14D

18/46 Y4
↓



Open the category menu of Folder B

↓ → X2



Select additional items in category of Folder B

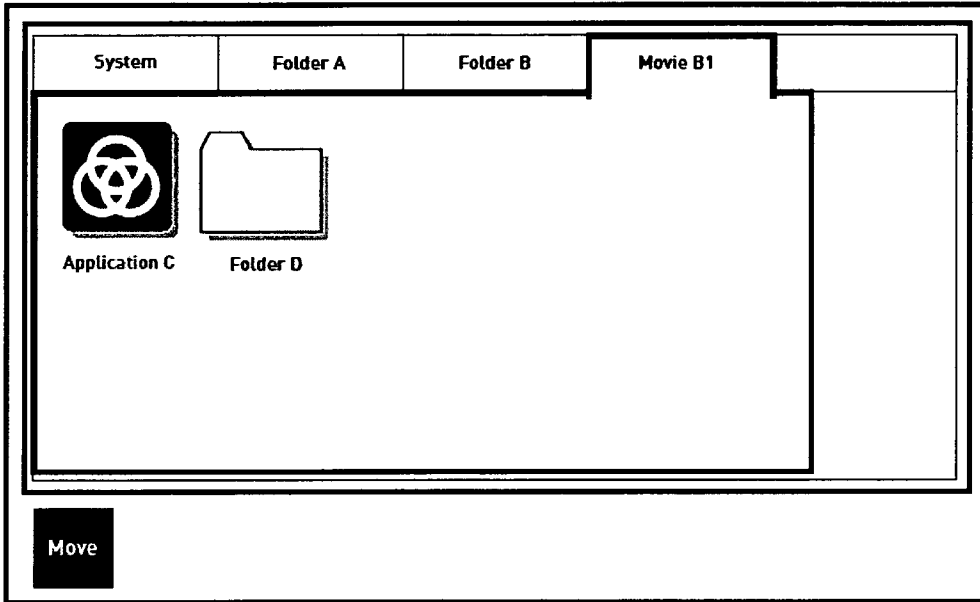
and

*Activate category tab of Movie B1
(User can attach any type of files in the Document)*

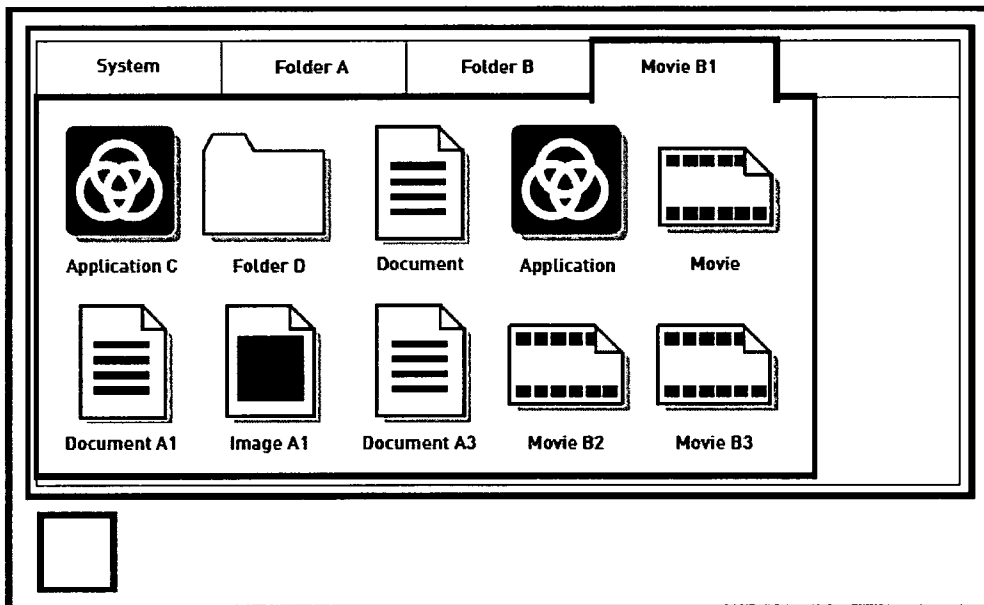
↓ Y5

Figure 14E

19/46 (Y5)



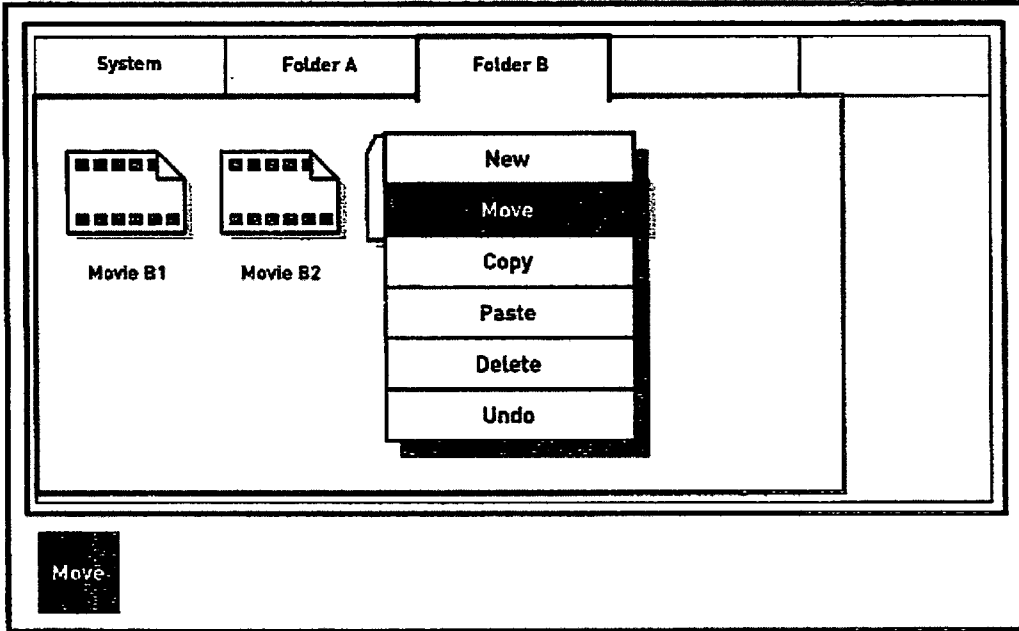
Open the category menu of Movie B1



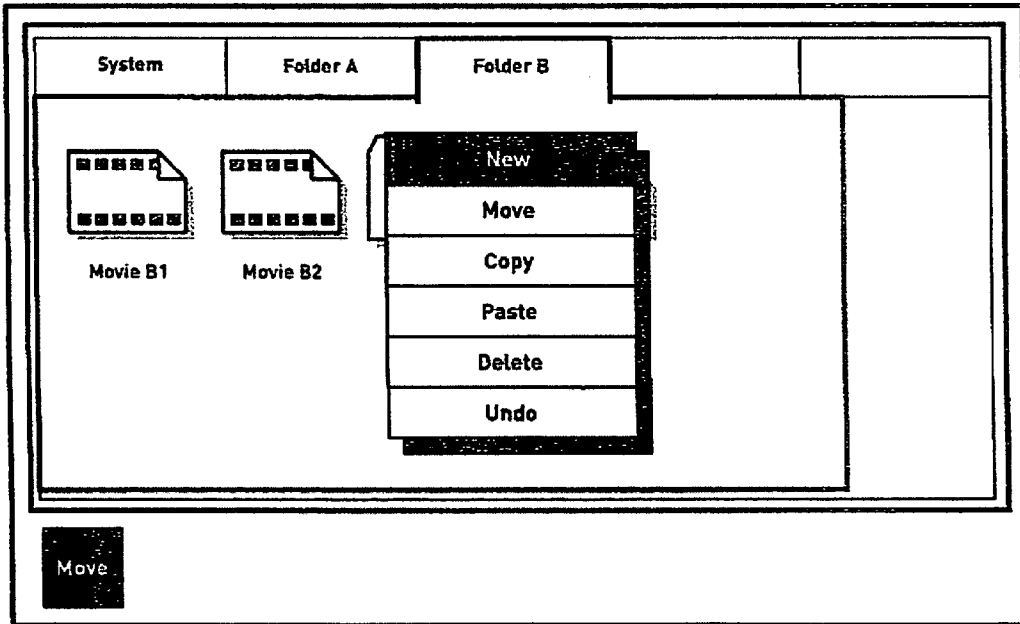
Double click or
Click on the category menu or
Click on the category tab or
Press second button or
Use different finger
to
Drop the selected items in category of Movie B1

Figure 14F

20/46 (X2)



Change action mode

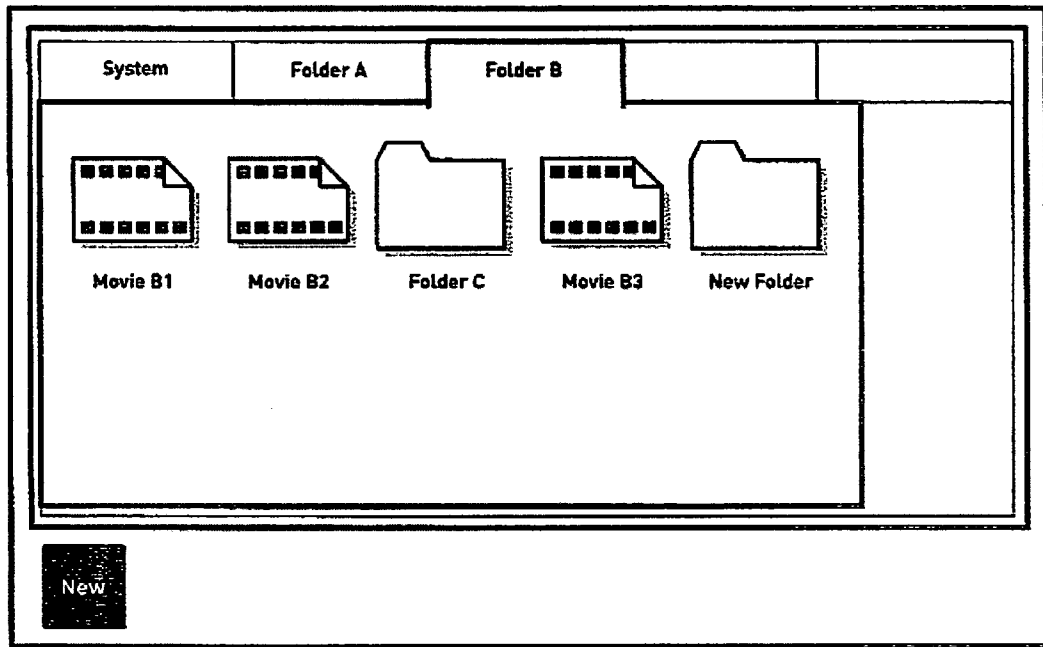


Select New

(Y6)

Figure 15A

21/46



*Created
New folder (for OS) or
New document (for Applications) or
New topic (for Editorial website).
Depending on the category and OOP*.*

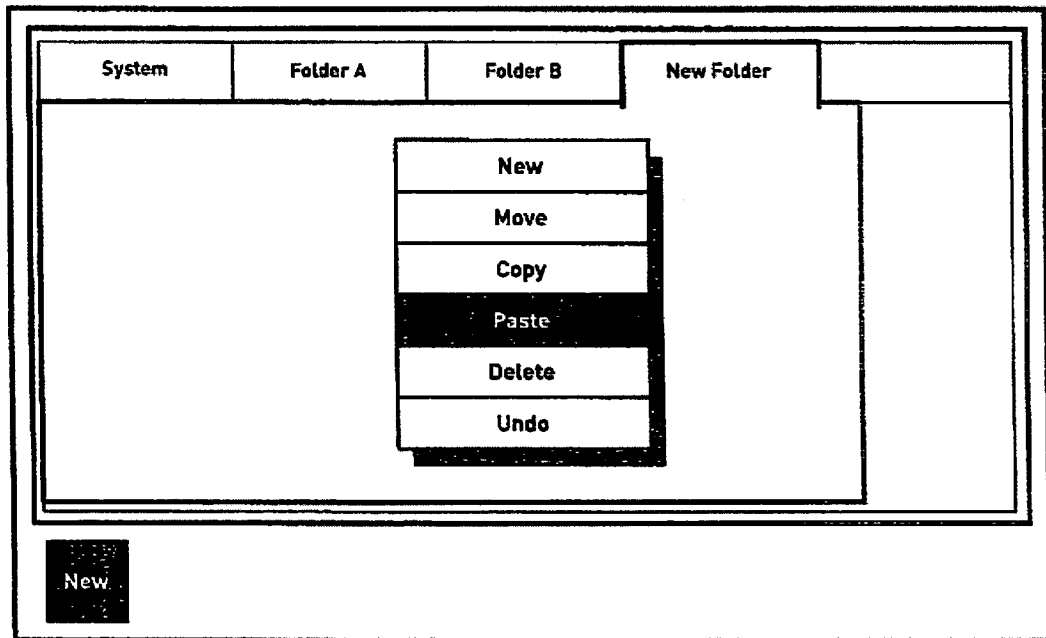
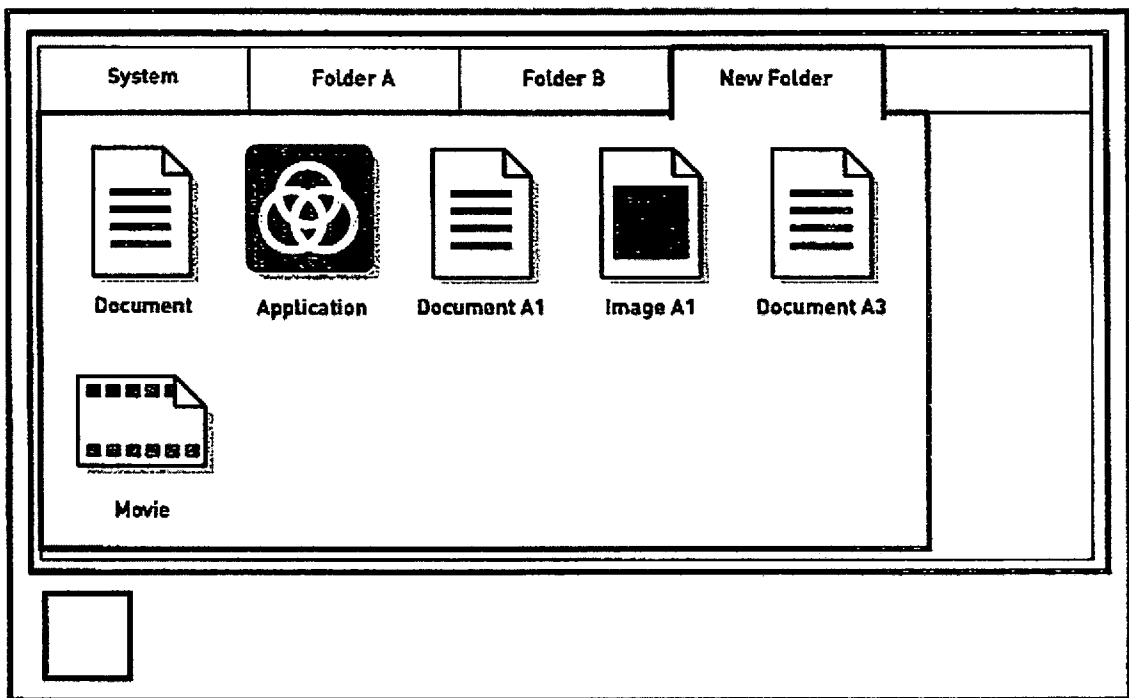


Figure 15B

22/46

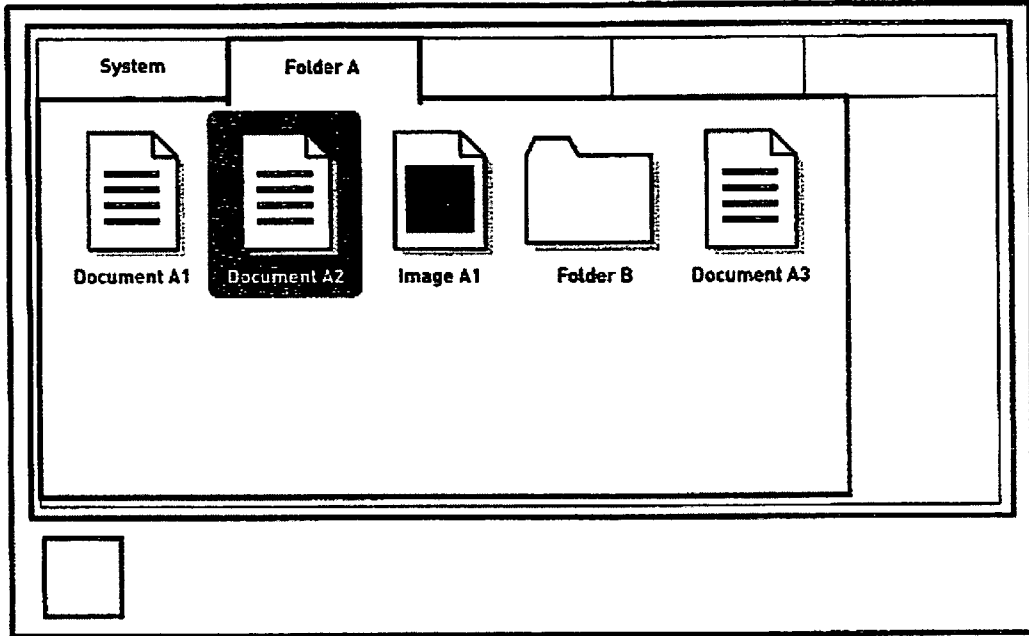


Selected items moved in New Folder

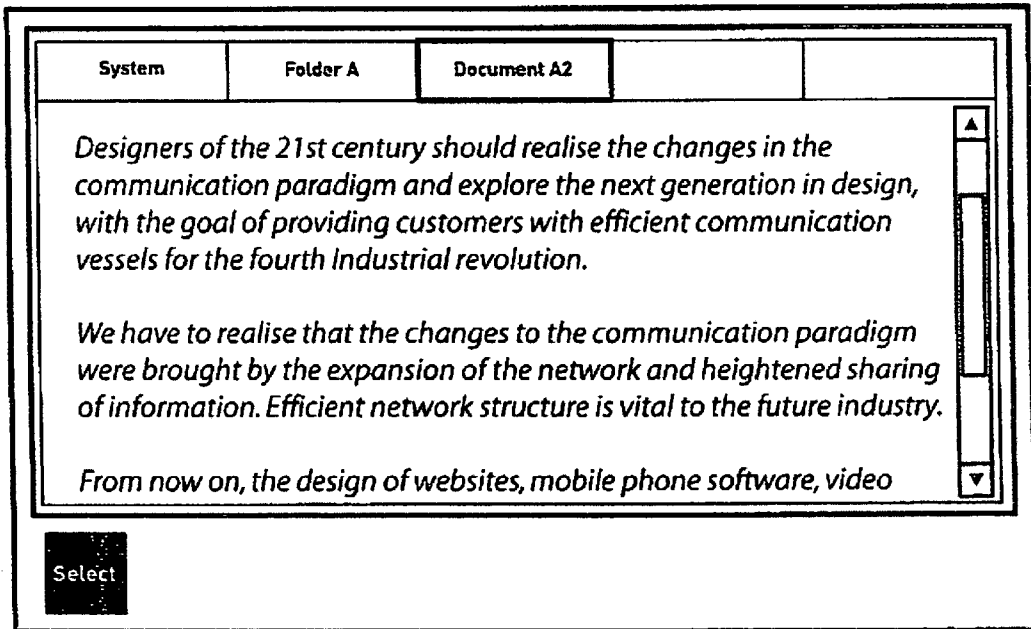
Figure 15C

23/46

X1



Switch off the Action and Run(Play) the Document A2 (file)

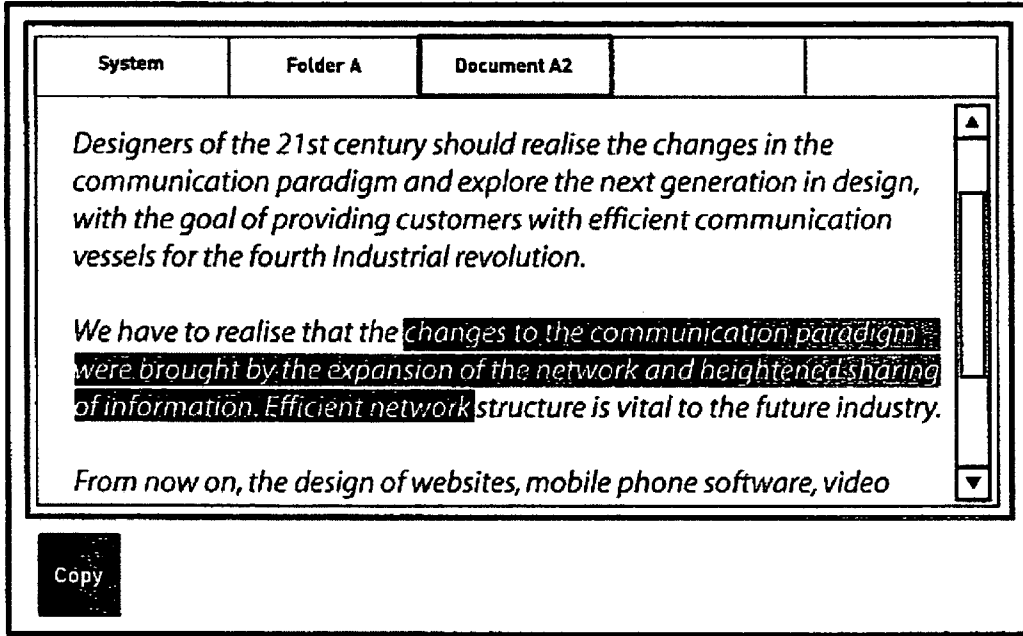


Change the action mode to select the text

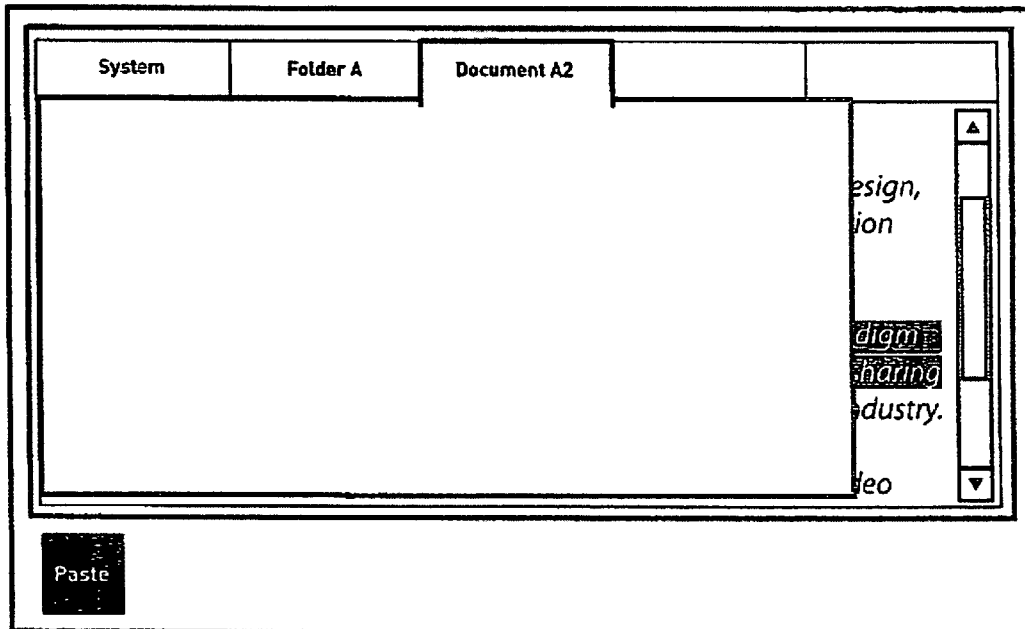
Y8

Figure 16A

24/46 Y8
↓



copy the selected text

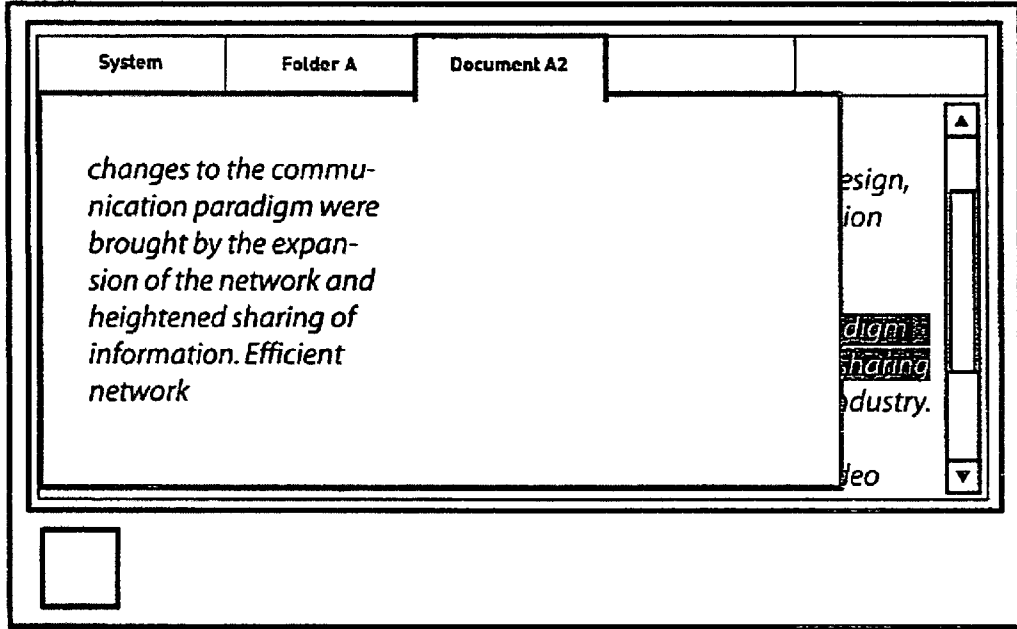


Activate category tab of Document A2

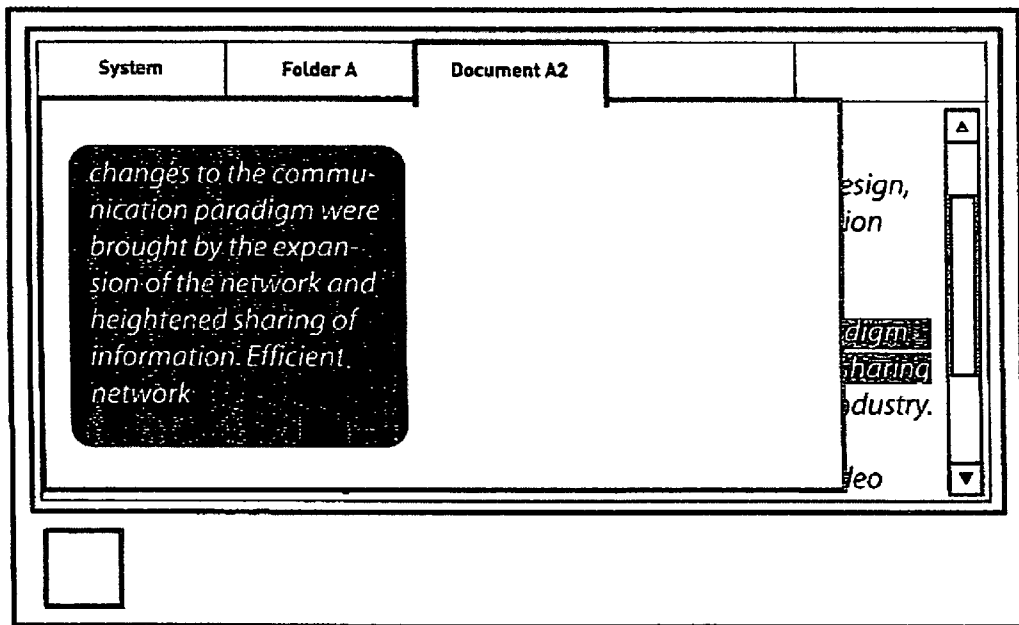
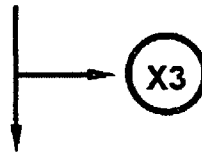
↓
Y9

Figure 16B

25/46



Paste the copied text in to the Category Menu



Run(Play) the copied text (file)



Figure 16C

26/46

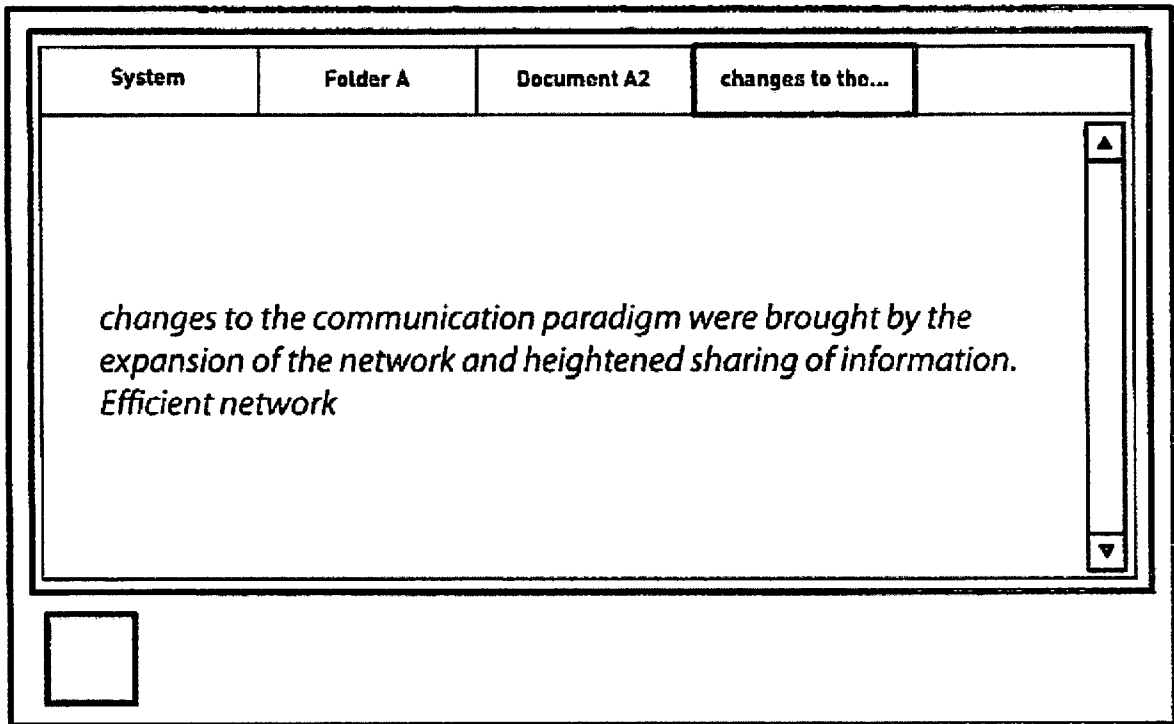
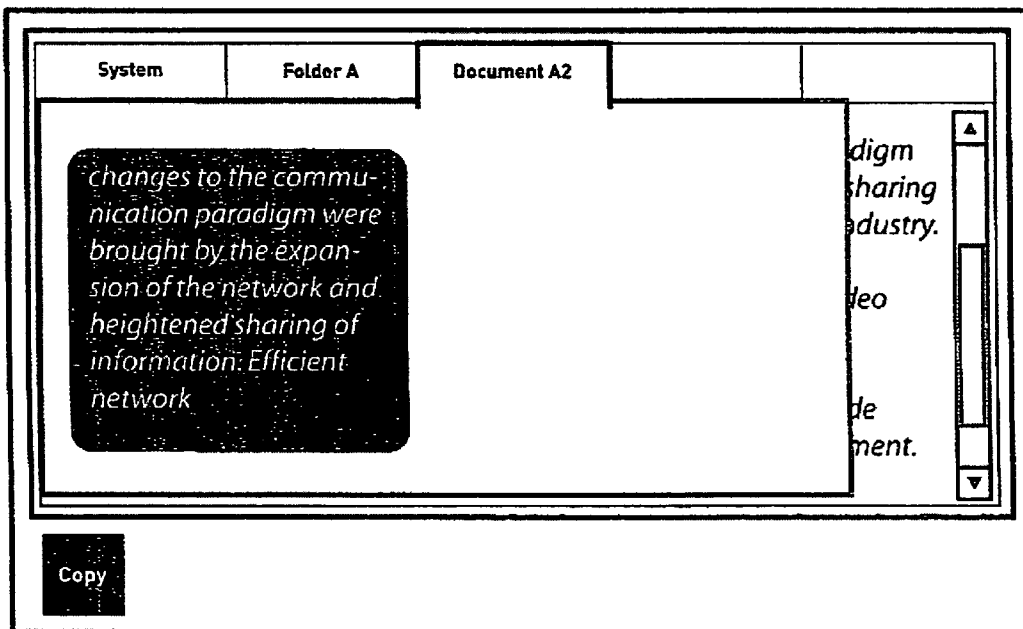
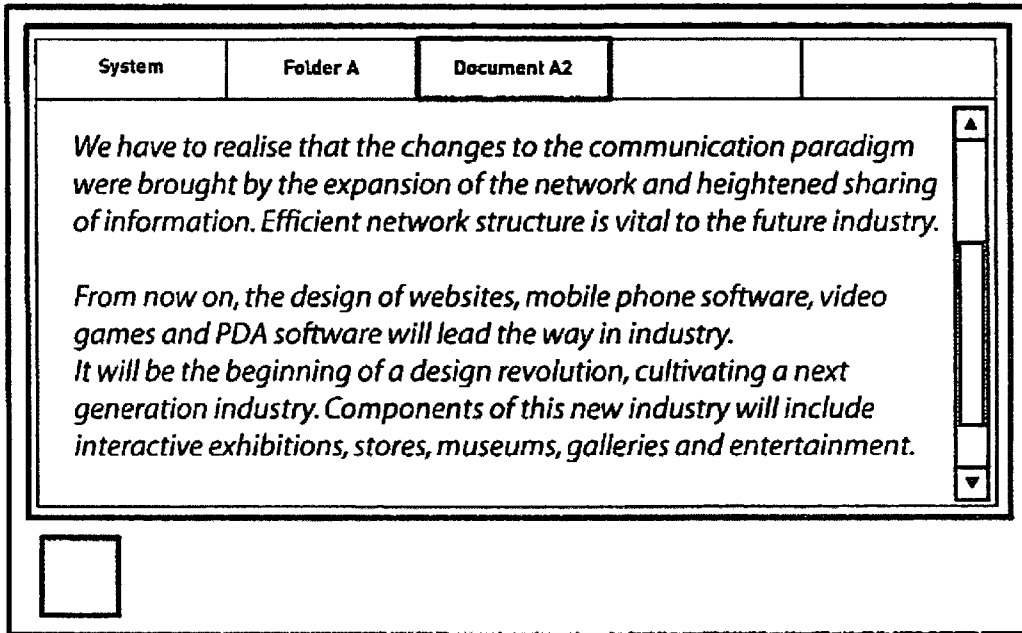


Figure 16D

27/46 **X3**



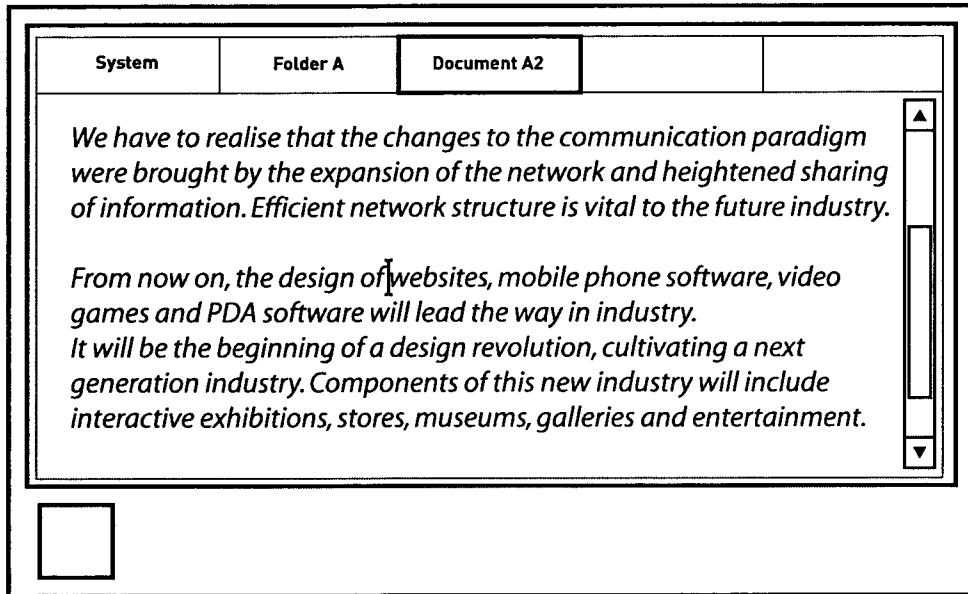
Activate category tab of Document A2
and
Copy the text

Y11

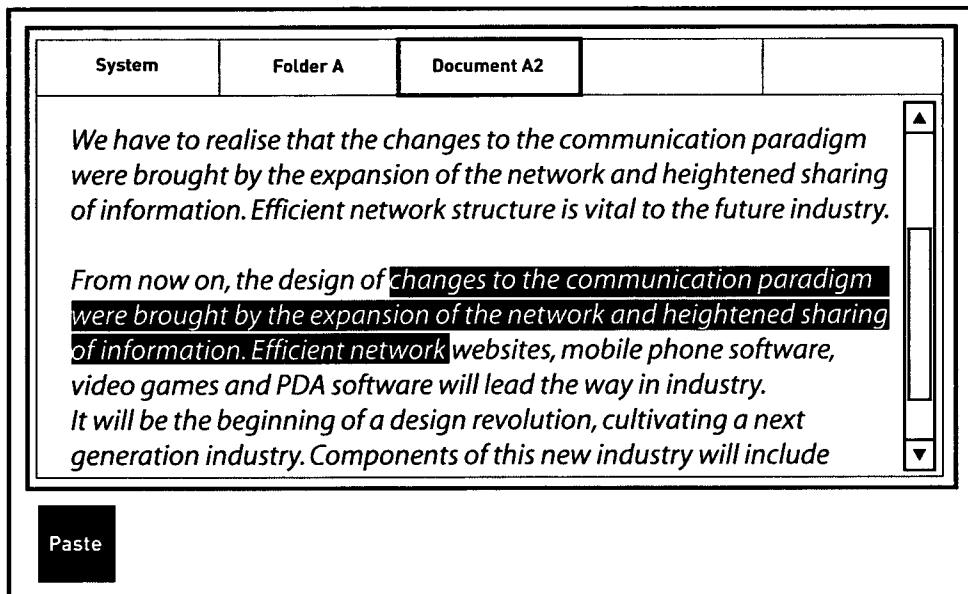
Figure 17A

28/46

Y11



Deactivate category tap and select the location



Paste the copied text in to the location

Figure 17B

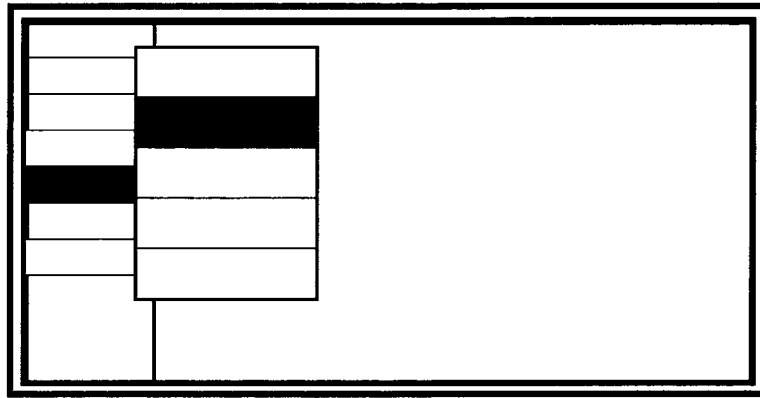


Figure 18A

510

510

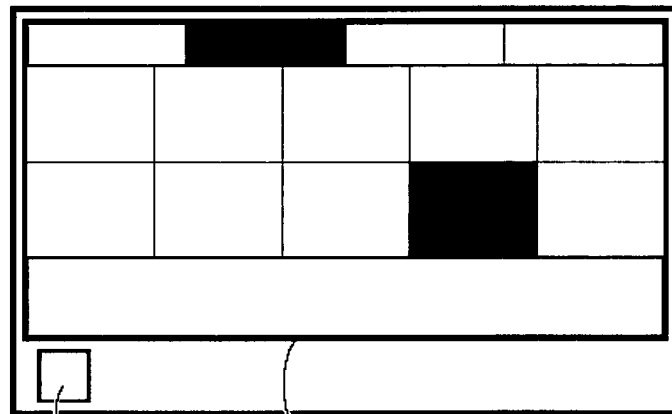


Figure 18B

510

510

510

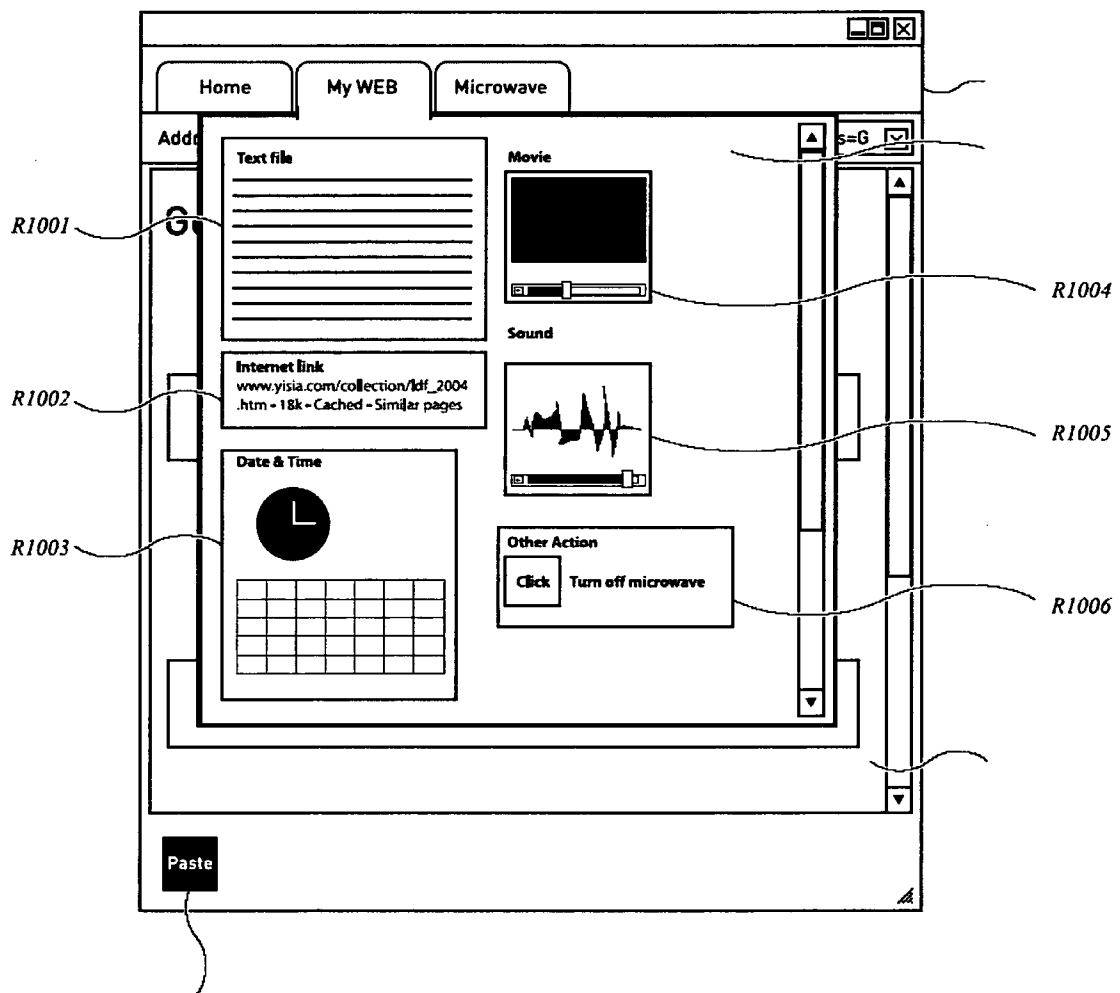


Figure 19

Figure 20A

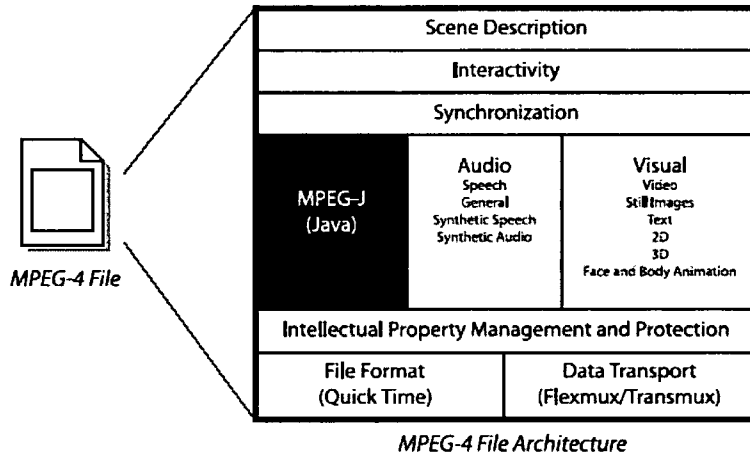
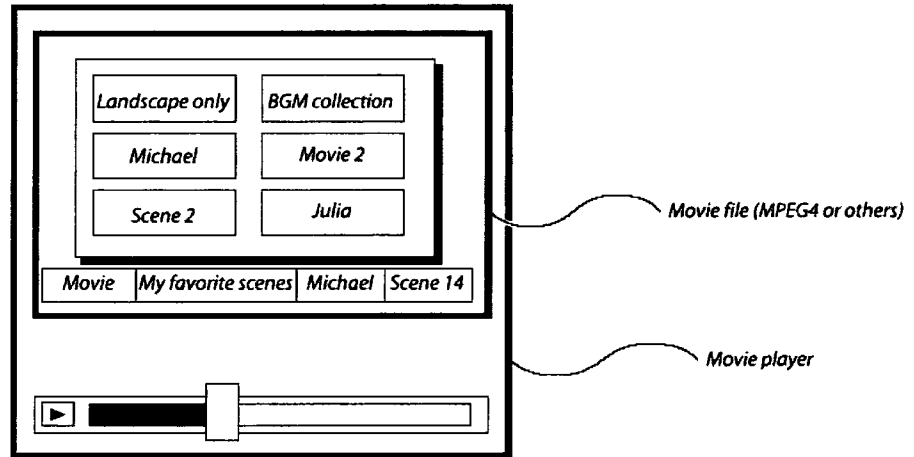


Figure 20B

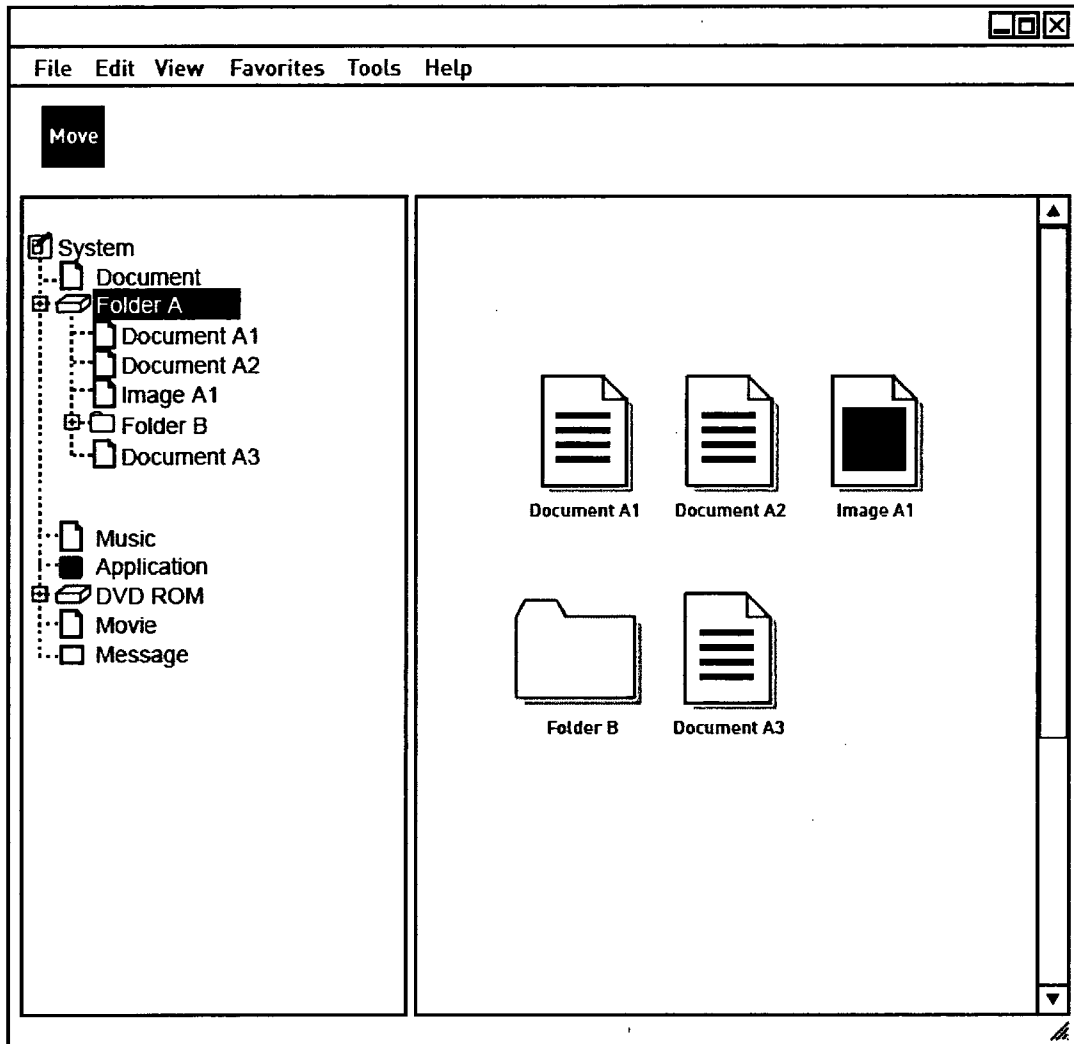
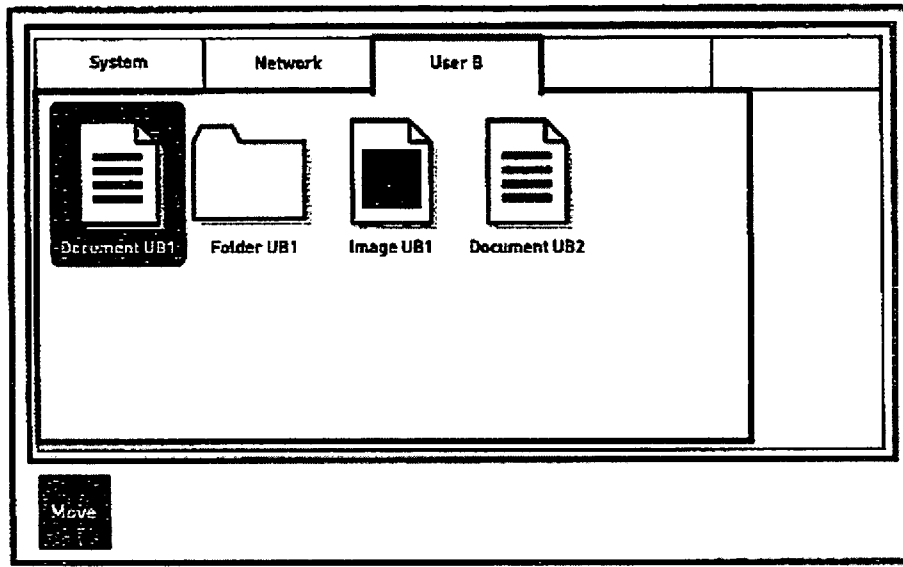
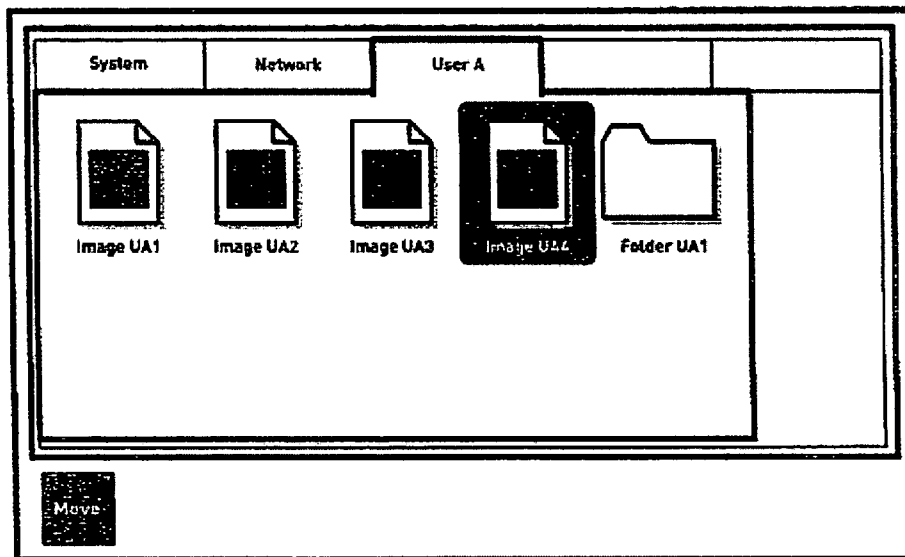


Figure 21

33/46



User A



UserB

Figure 22

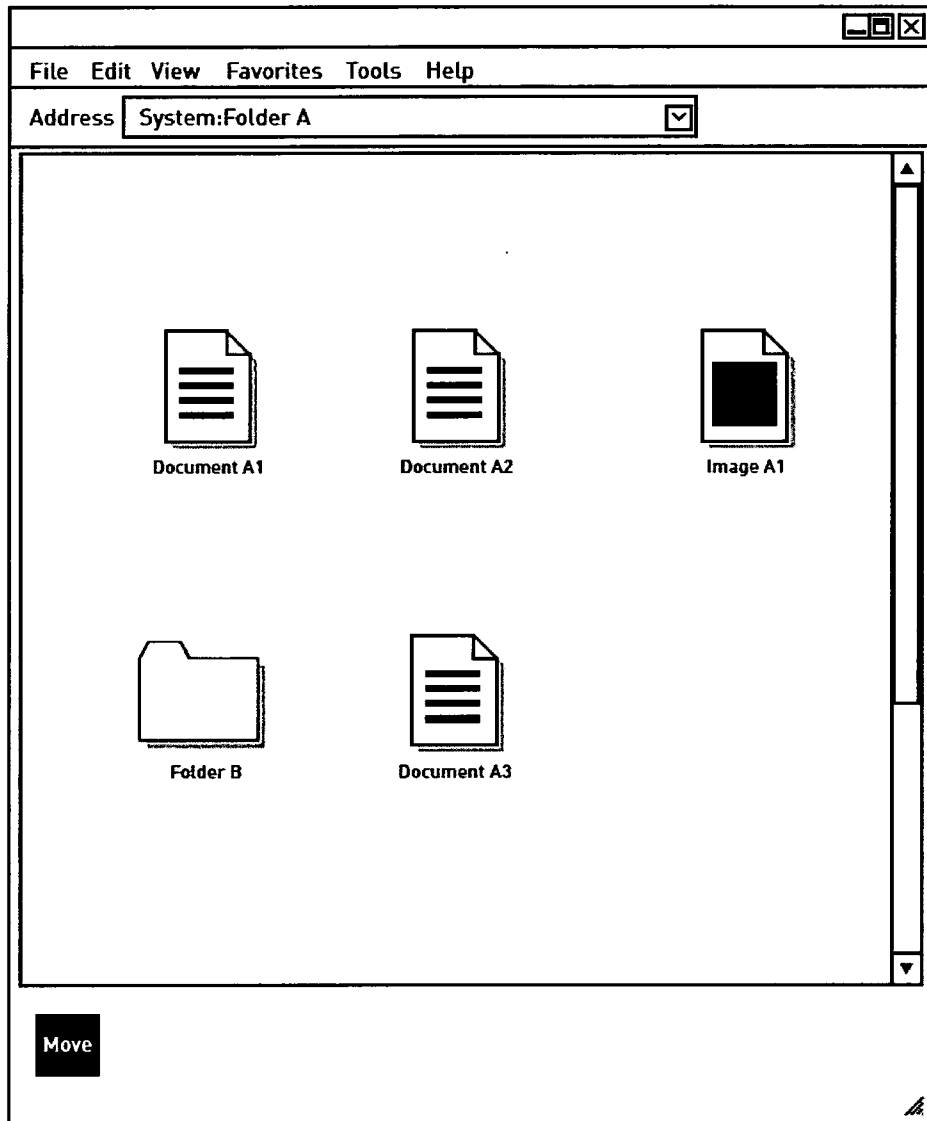


Figure 23

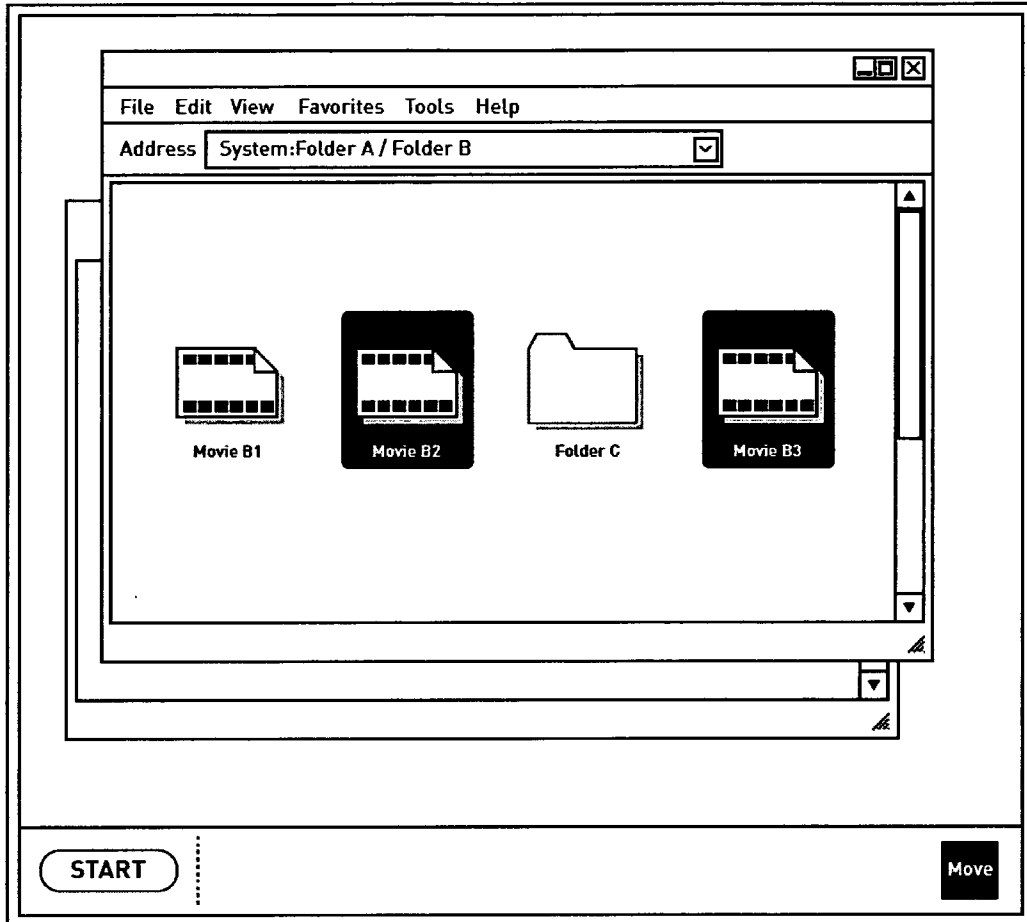


Figure 24

36/46

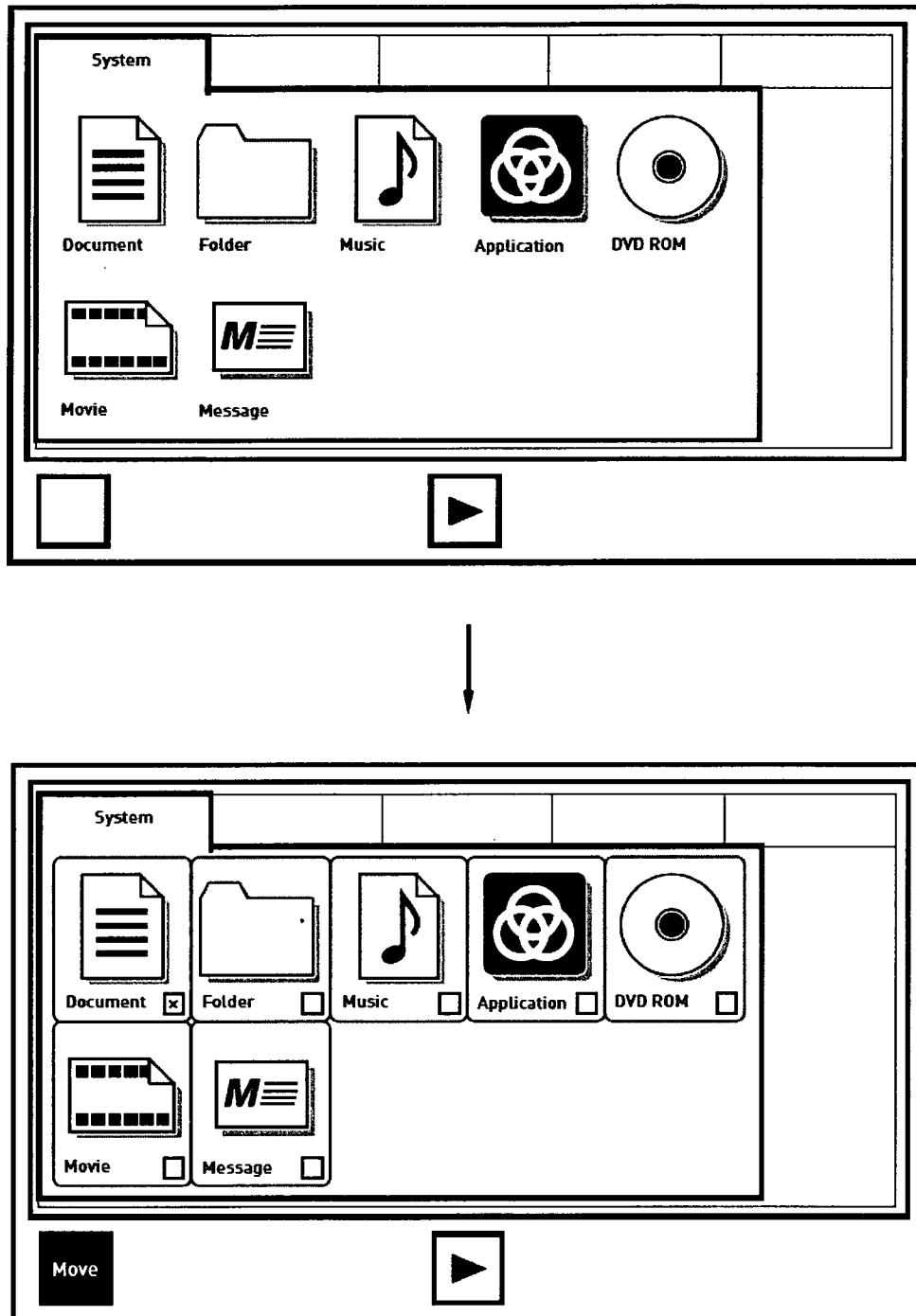


Figure 25

37/46

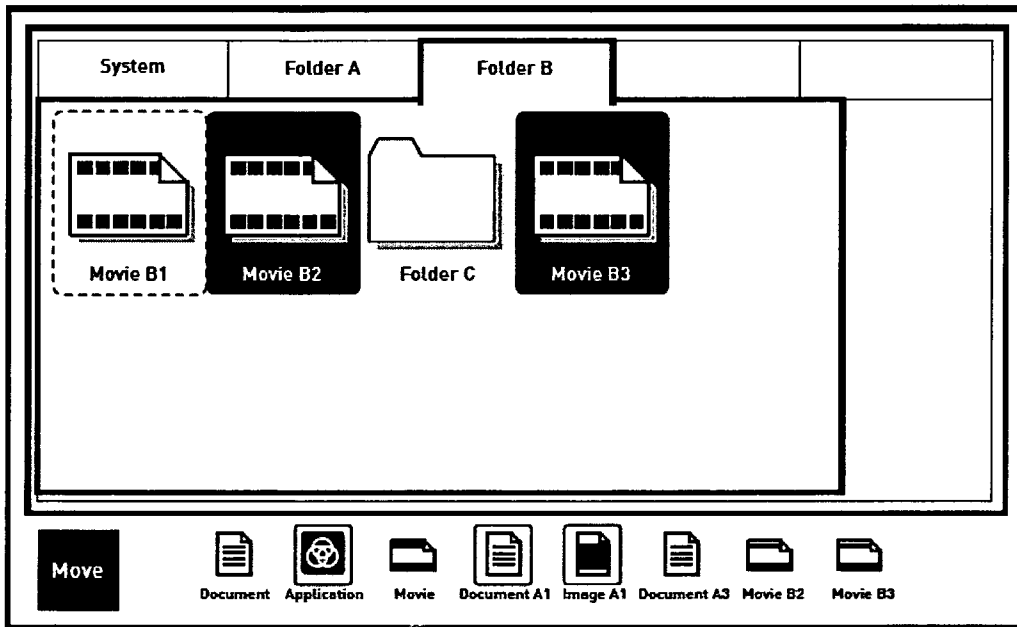


Figure 26

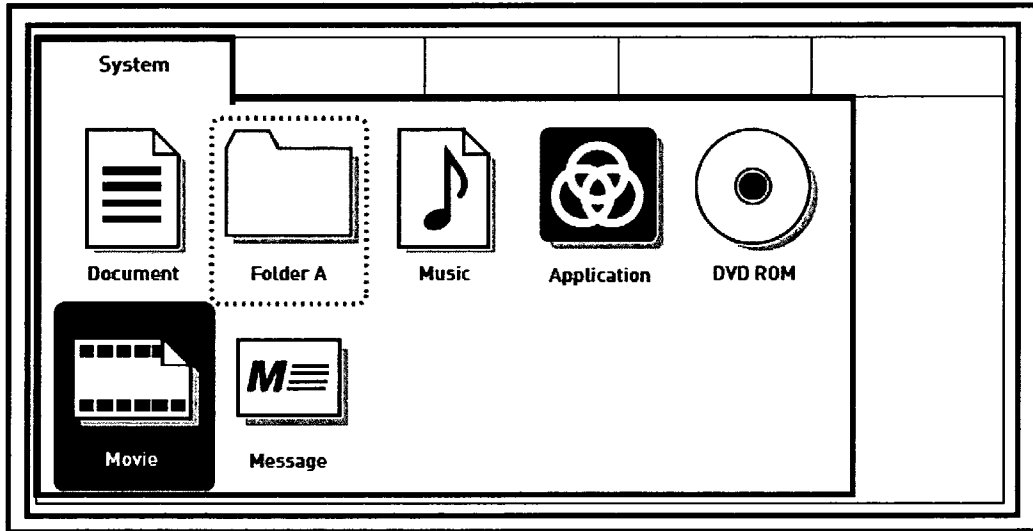
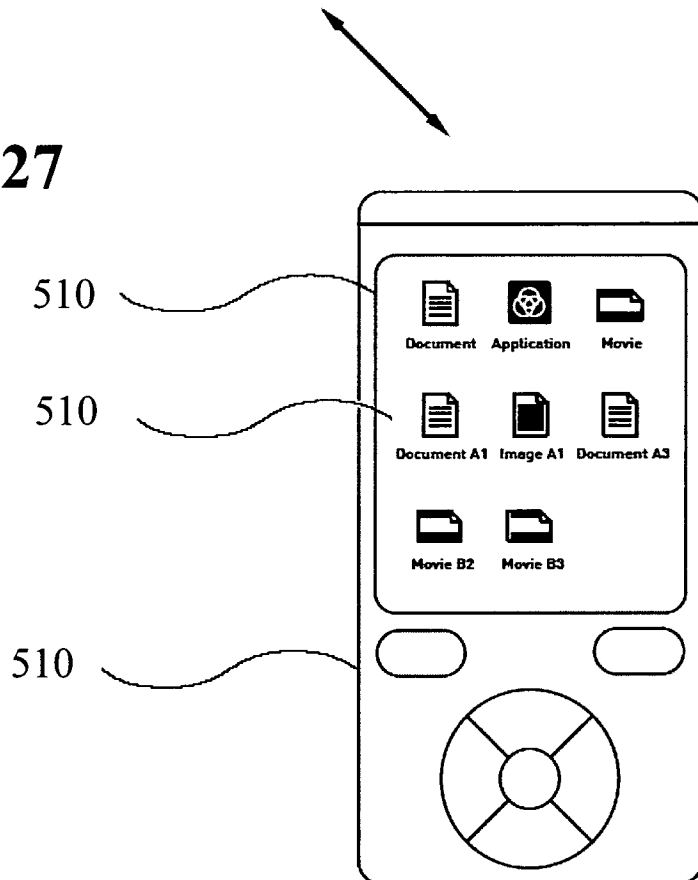


Figure 27



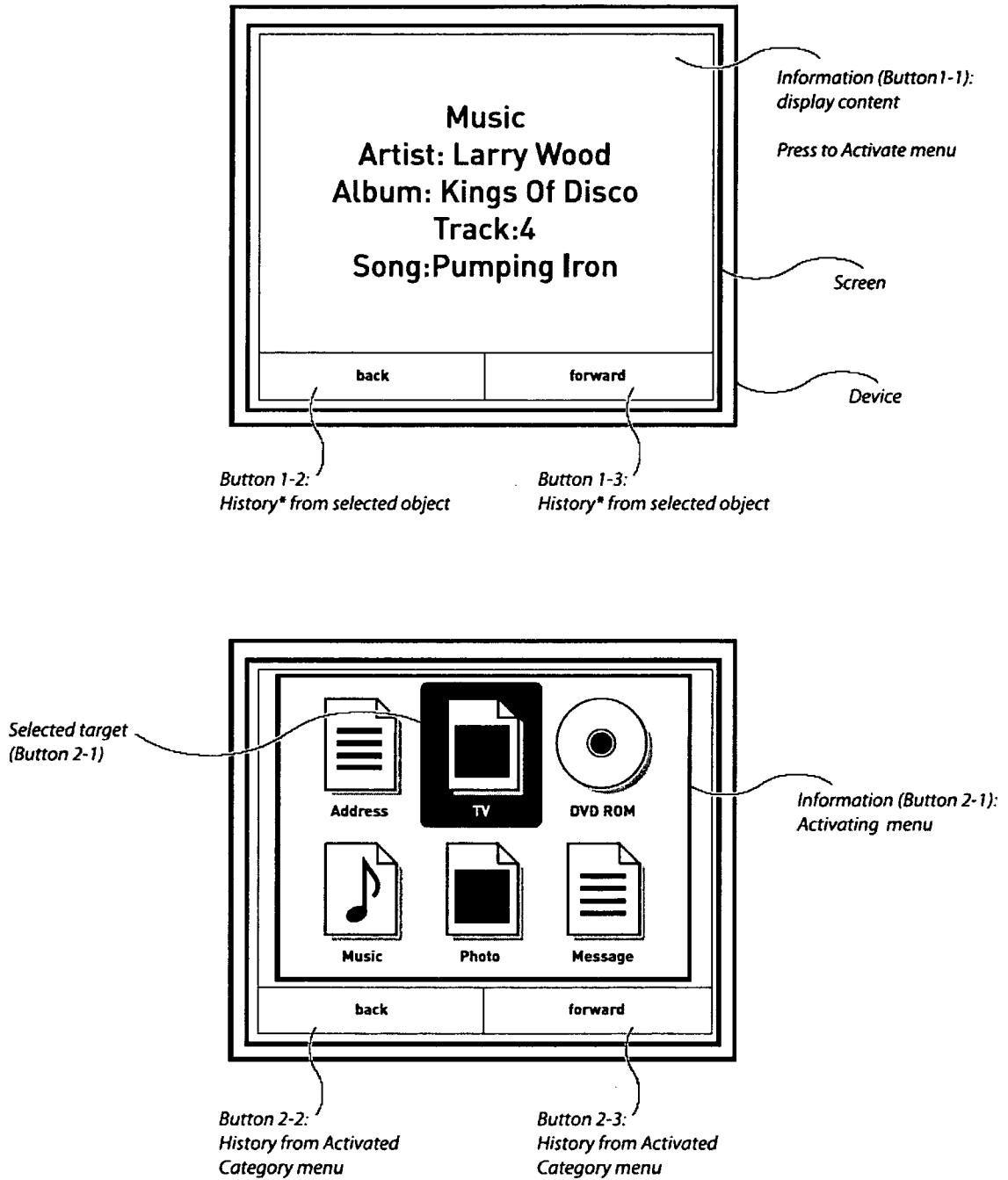


Figure 28

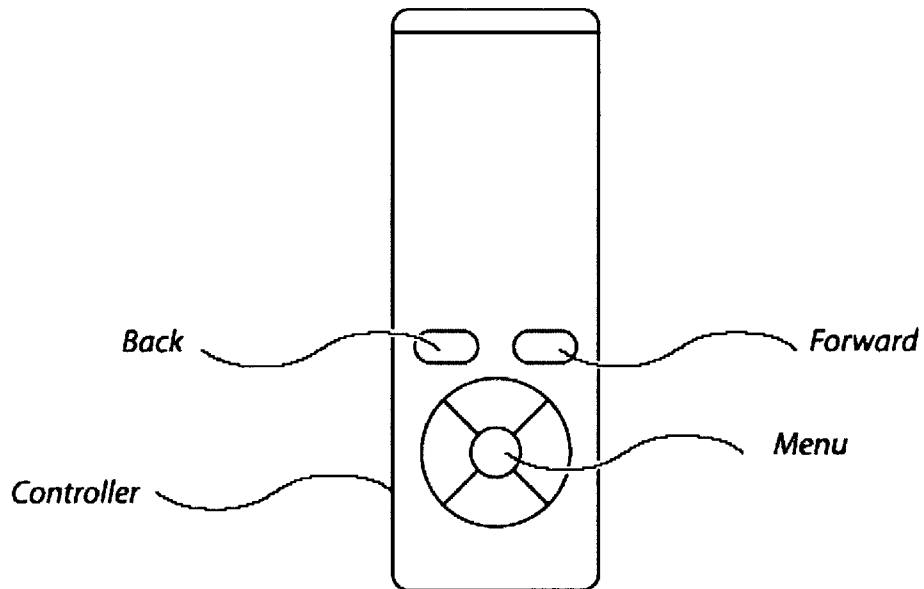
40/46



Display



Figure 29

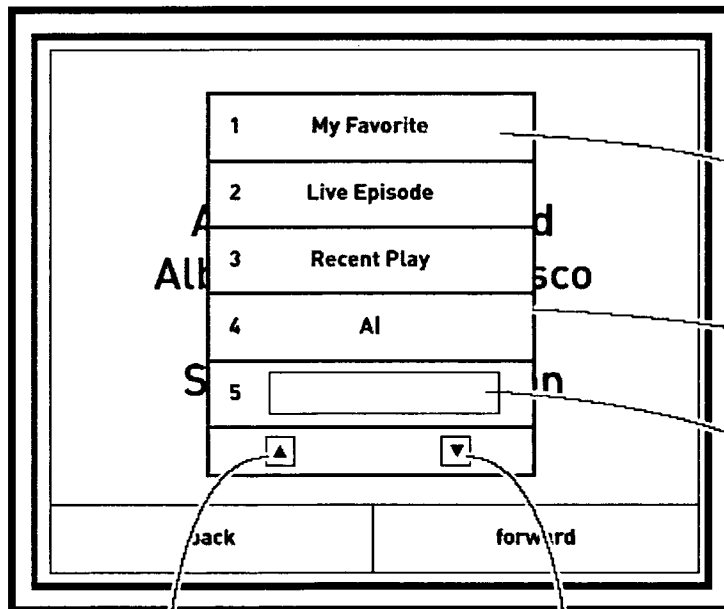


41/46



Press and hold 1Sec

Figure 30



Button

Menu

User Input

Up the priority of selected

Down the priority of selected

Display mode change

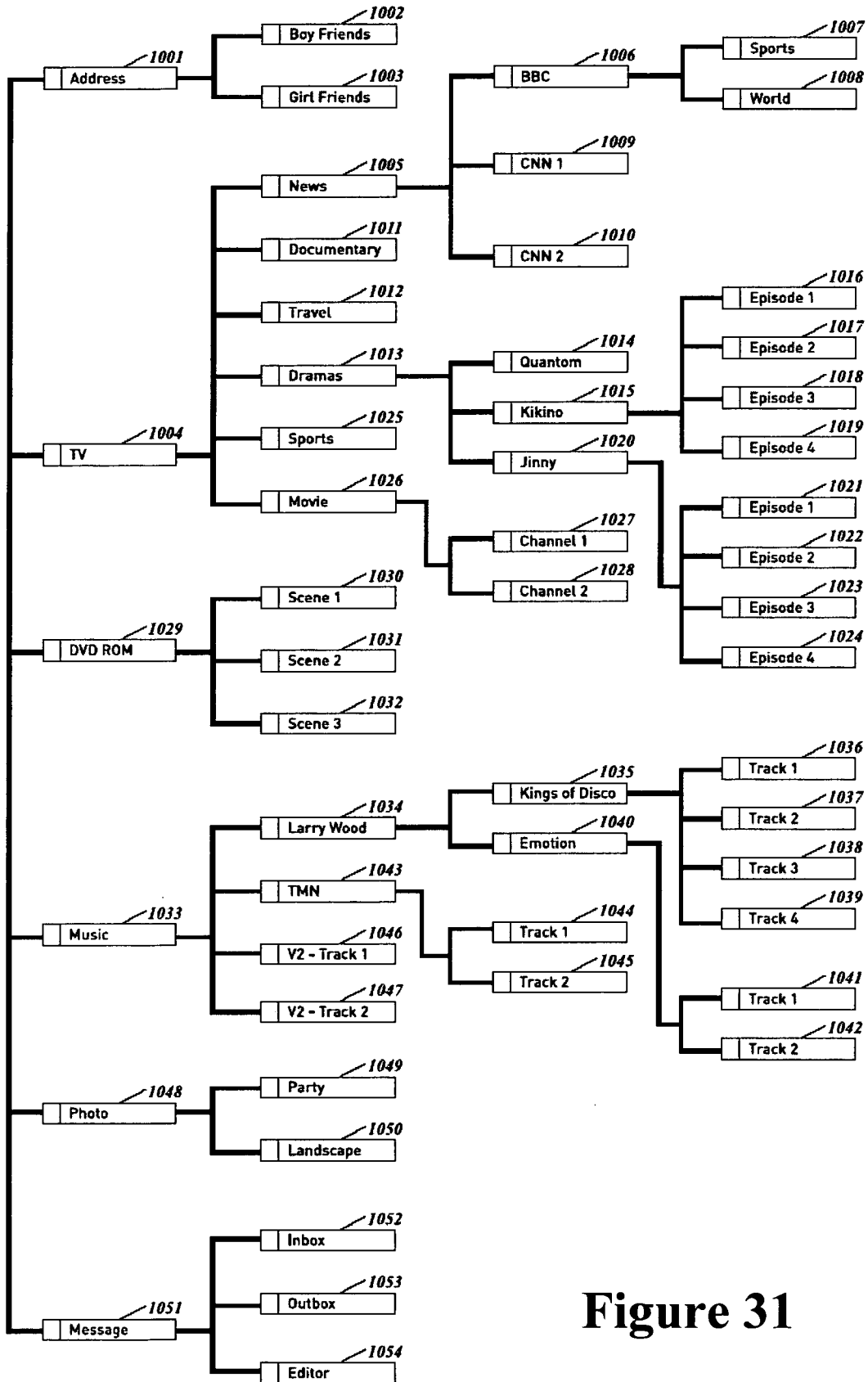


Figure 31

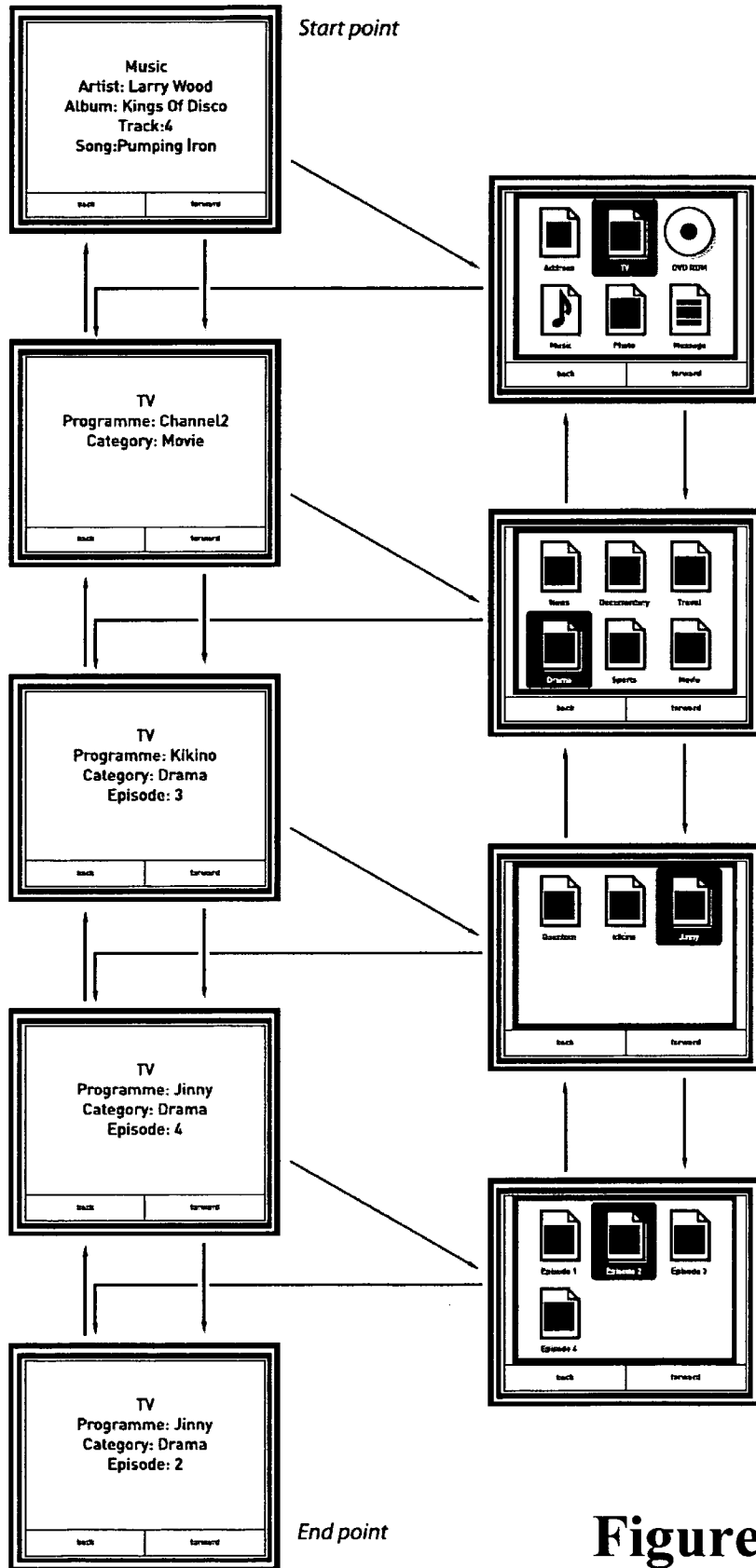


Figure 32

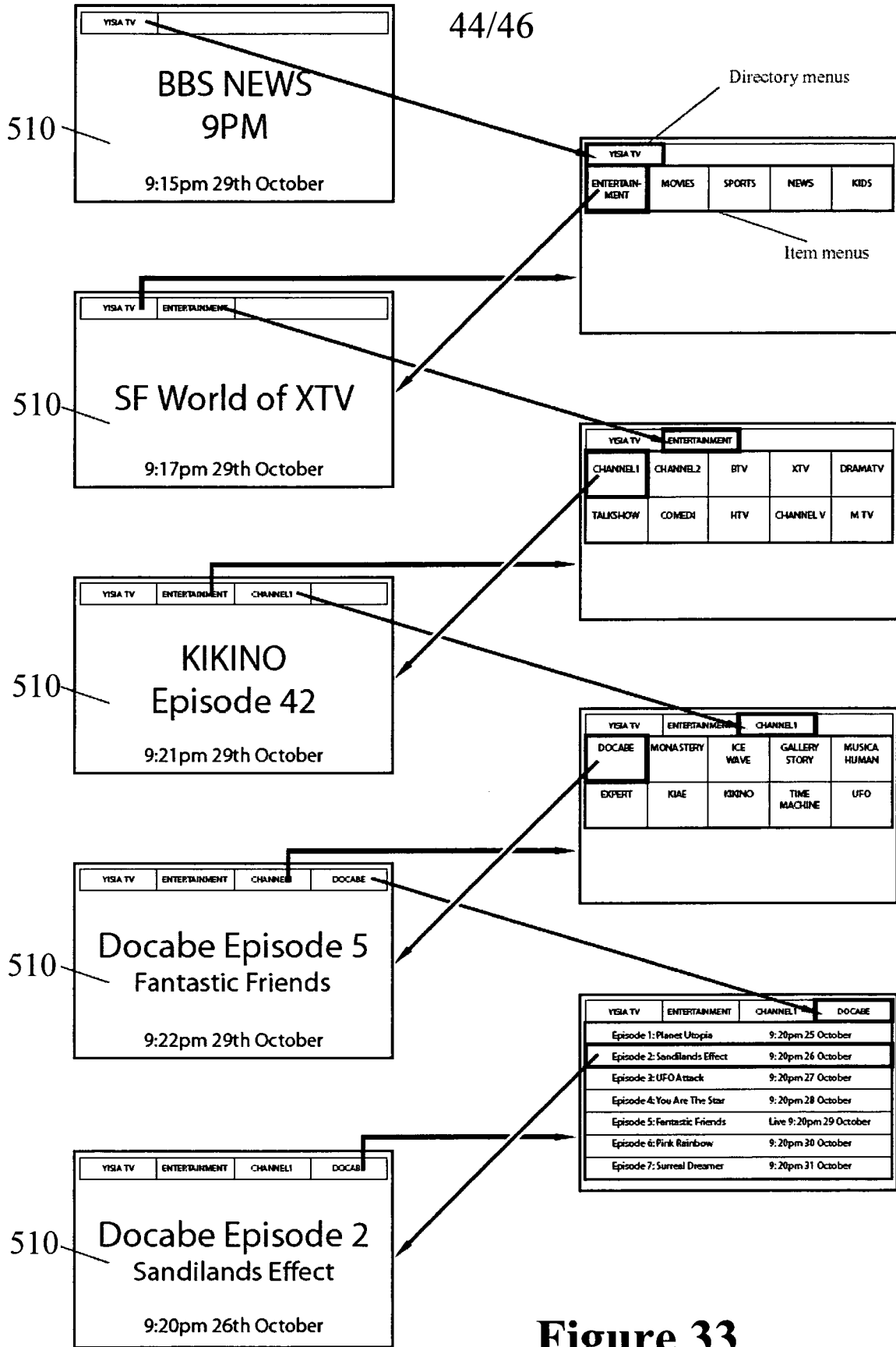


Figure 33

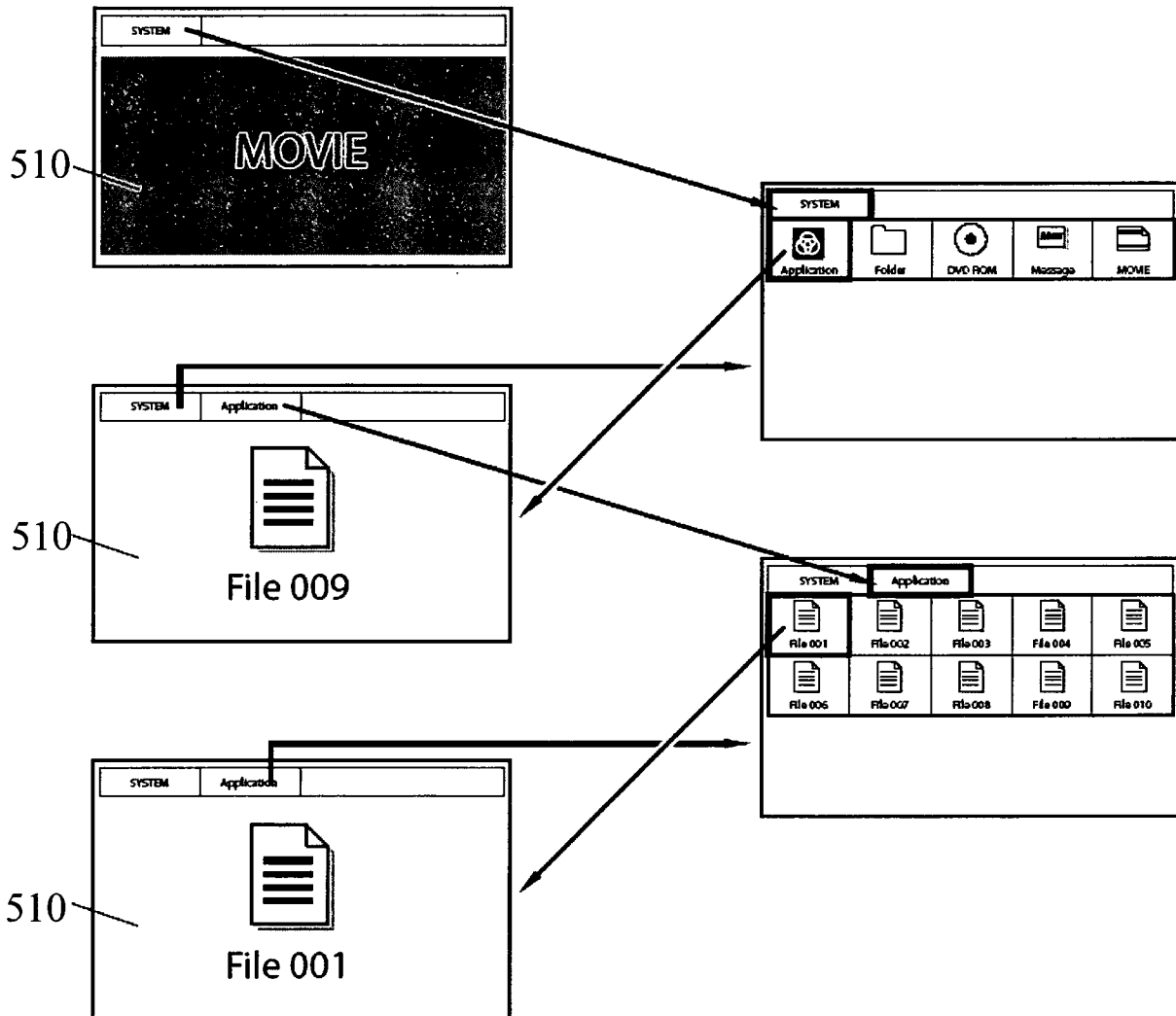


Figure 34

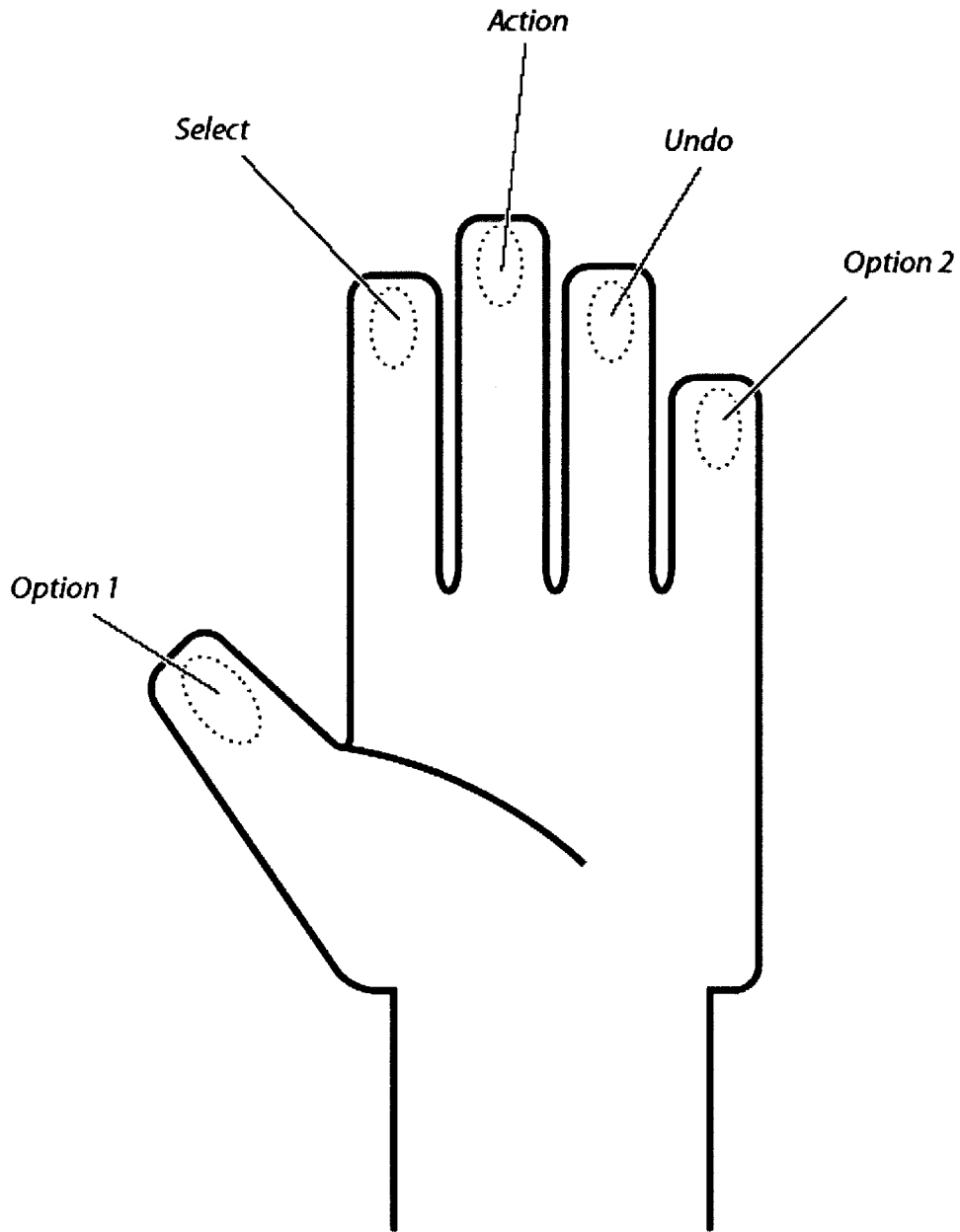


Figure 35