



US 20100005138A1

(19) **United States**(12) **Patent Application Publication**
Manzano(10) **Pub. No.: US 2010/0005138 A1**(43) **Pub. Date: Jan. 7, 2010**(54) **ELECTRONIC FILE SHARING****Publication Classification**(75) Inventor: **Michael R. Manzano**, Seattle, WA
(US)(51) **Int. Cl.**
G06F 15/16 (2006.01)

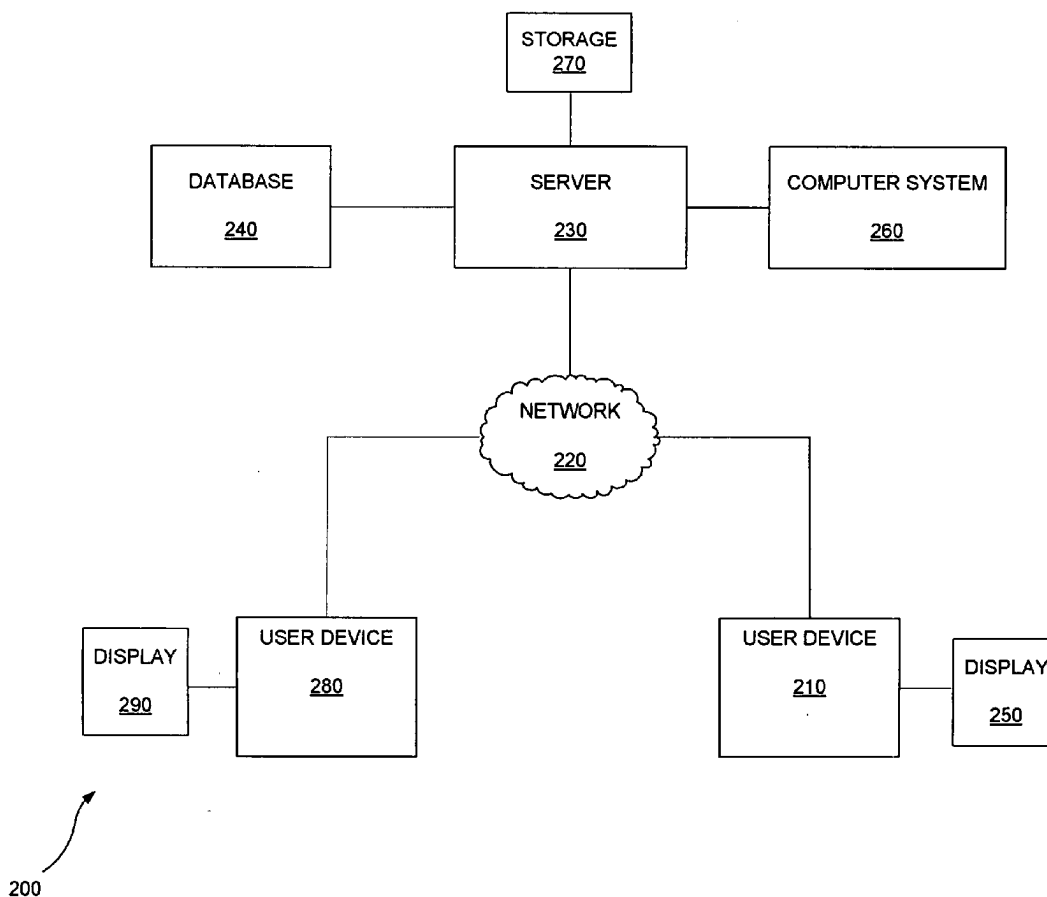
Correspondence Address:

BLACK LOWE & GRAHAM, PLLC
701 FIFTH AVENUE, SUITE 4800
SEATTLE, WA 98104 (US)(52) **U.S. Cl.** **709/203; 709/219**(73) Assignee: **Topia Technology**, Tacoma, WA
(US)(21) Appl. No.: **12/427,540**(22) Filed: **Apr. 21, 2009**(57) **ABSTRACT**

A system includes a server and a source client executable on a first electronic device in communication with the server. The source client is operable to make available to the server a set of files modifiable by a user of the first electronic device. The system further includes a first recipient client executable on a second electronic device in communication with the server. The first recipient client is configurable by a user of the second electronic device to automatically receive from the server only a subset of the files having a predetermined characteristic.

Related U.S. Application Data

(60) Provisional application No. 61/046,725, filed on Apr. 21, 2008.



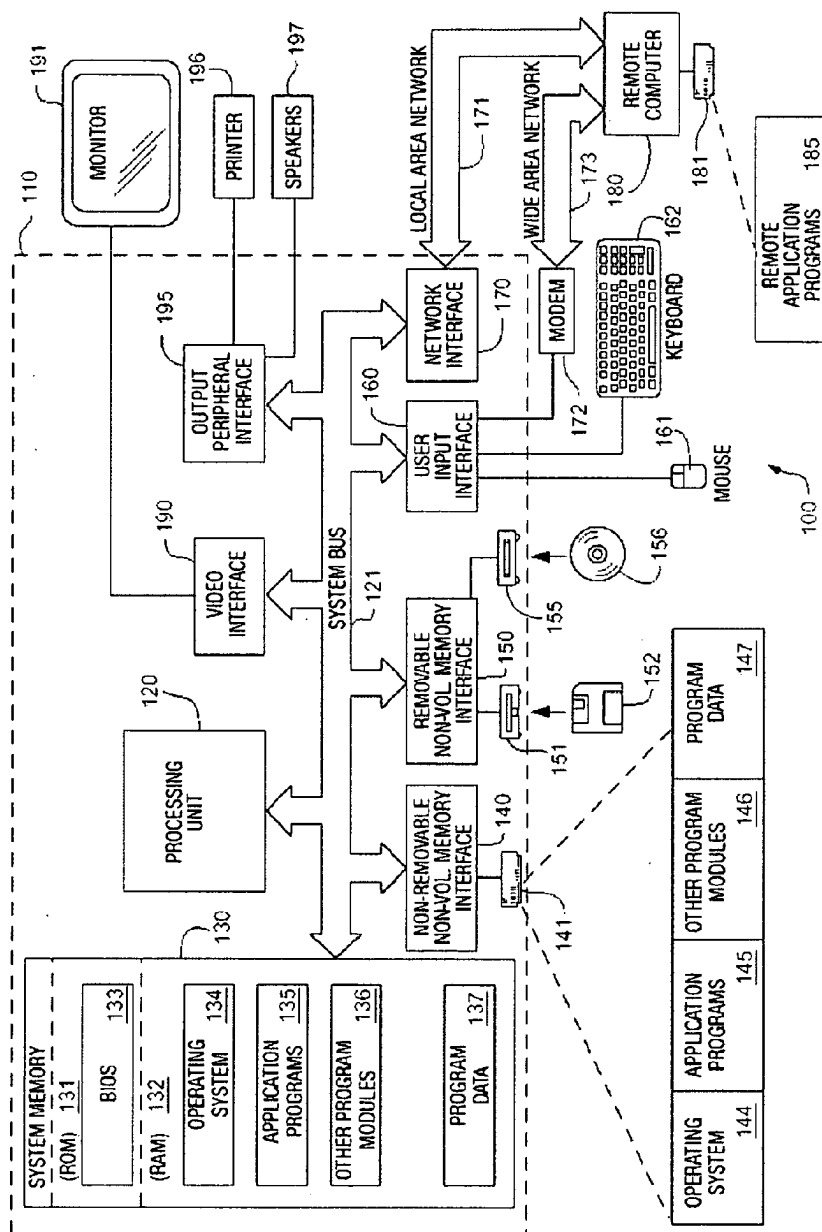


FIG. 1

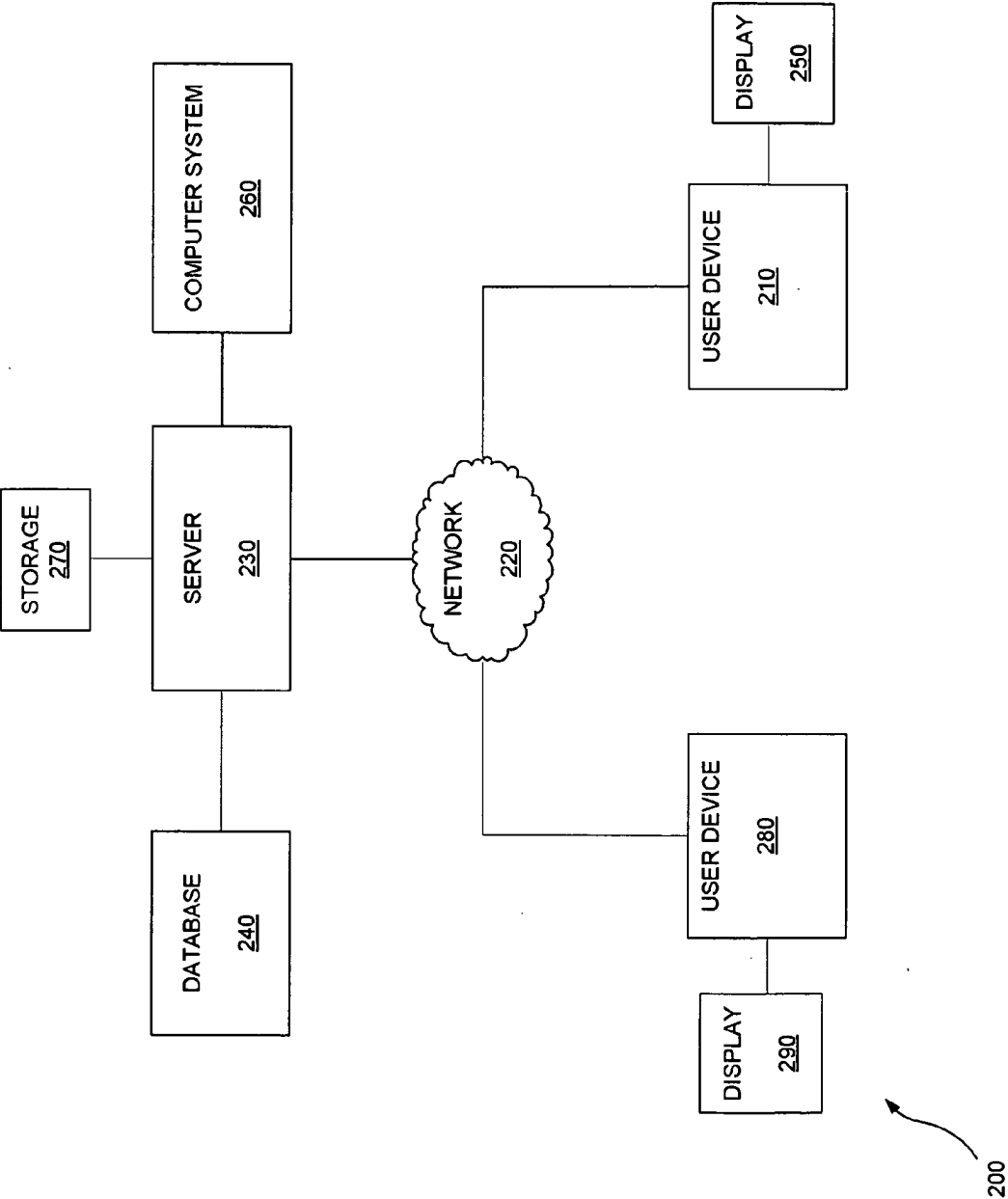


FIG. 2

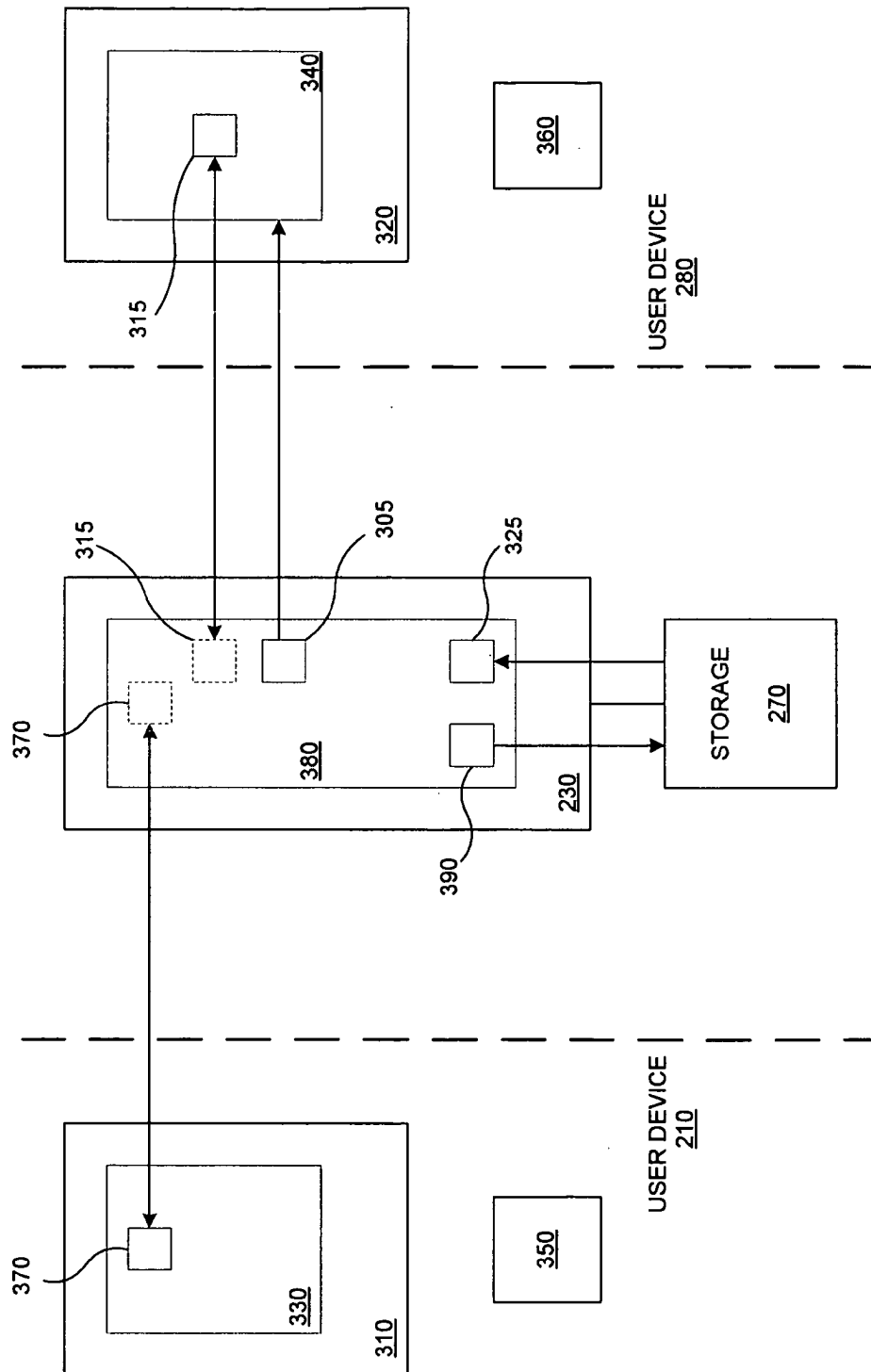


FIG. 3

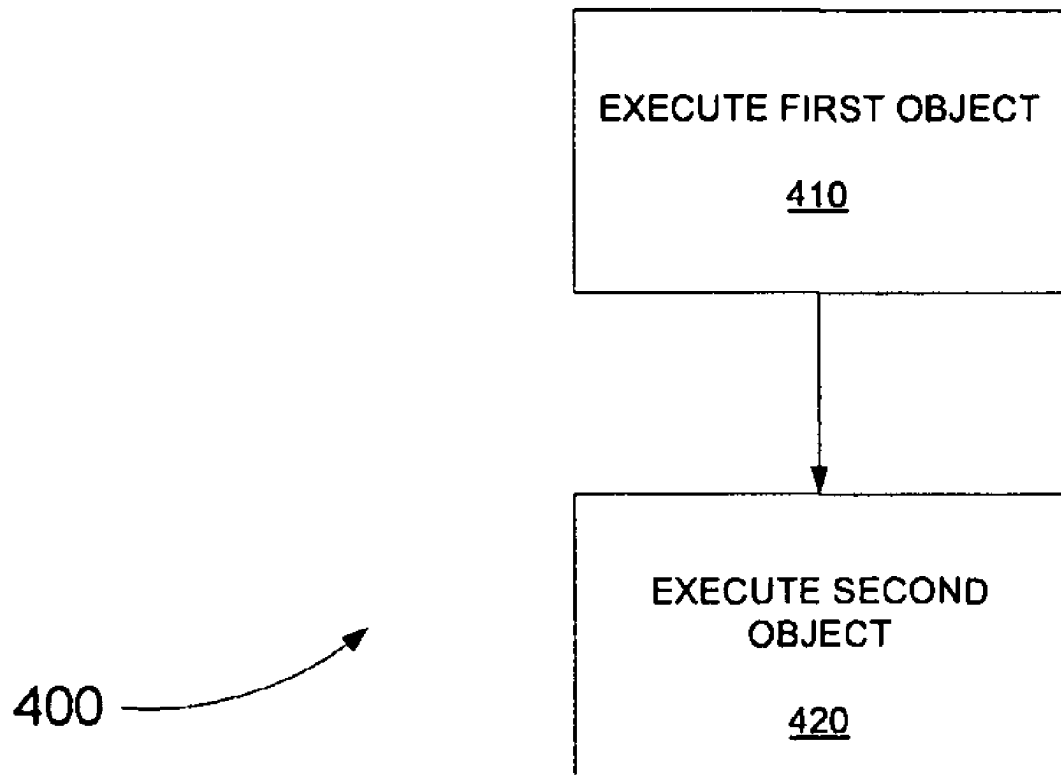


FIG. 4

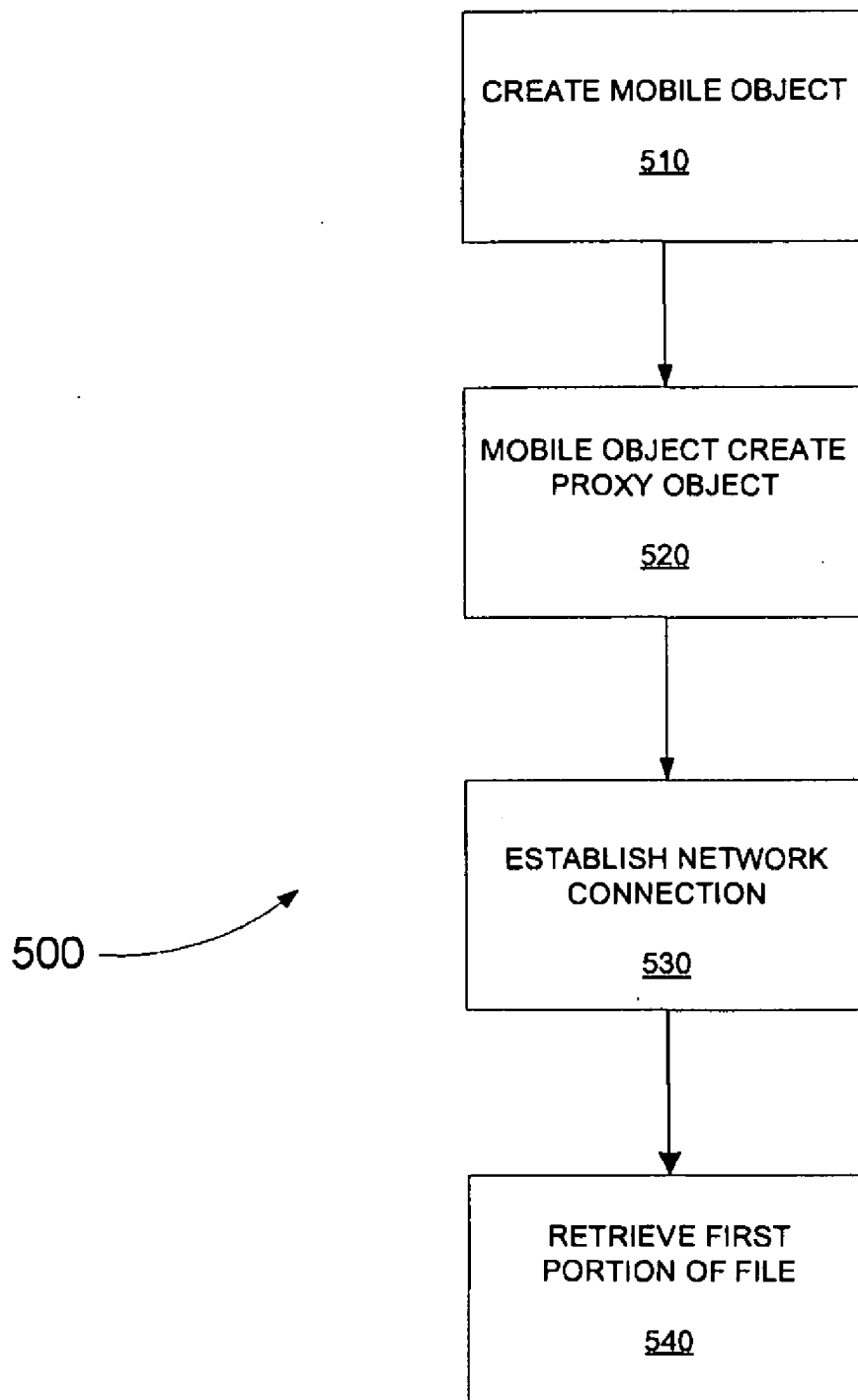


FIG. 5

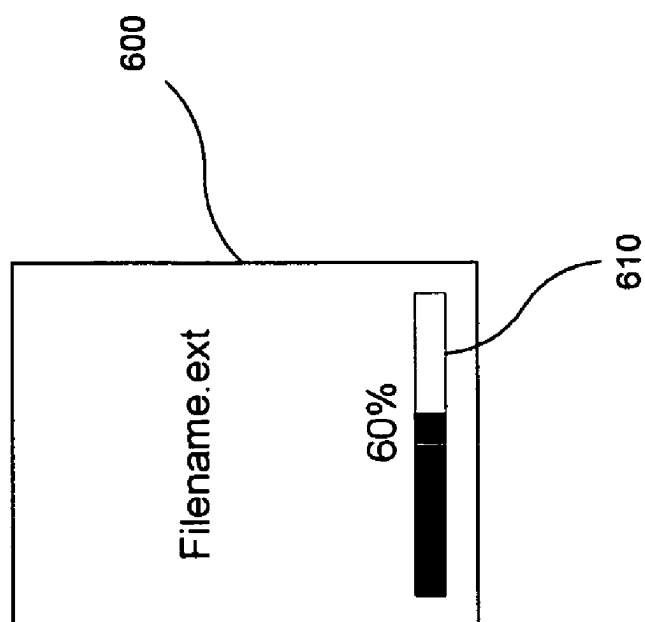


FIG. 6

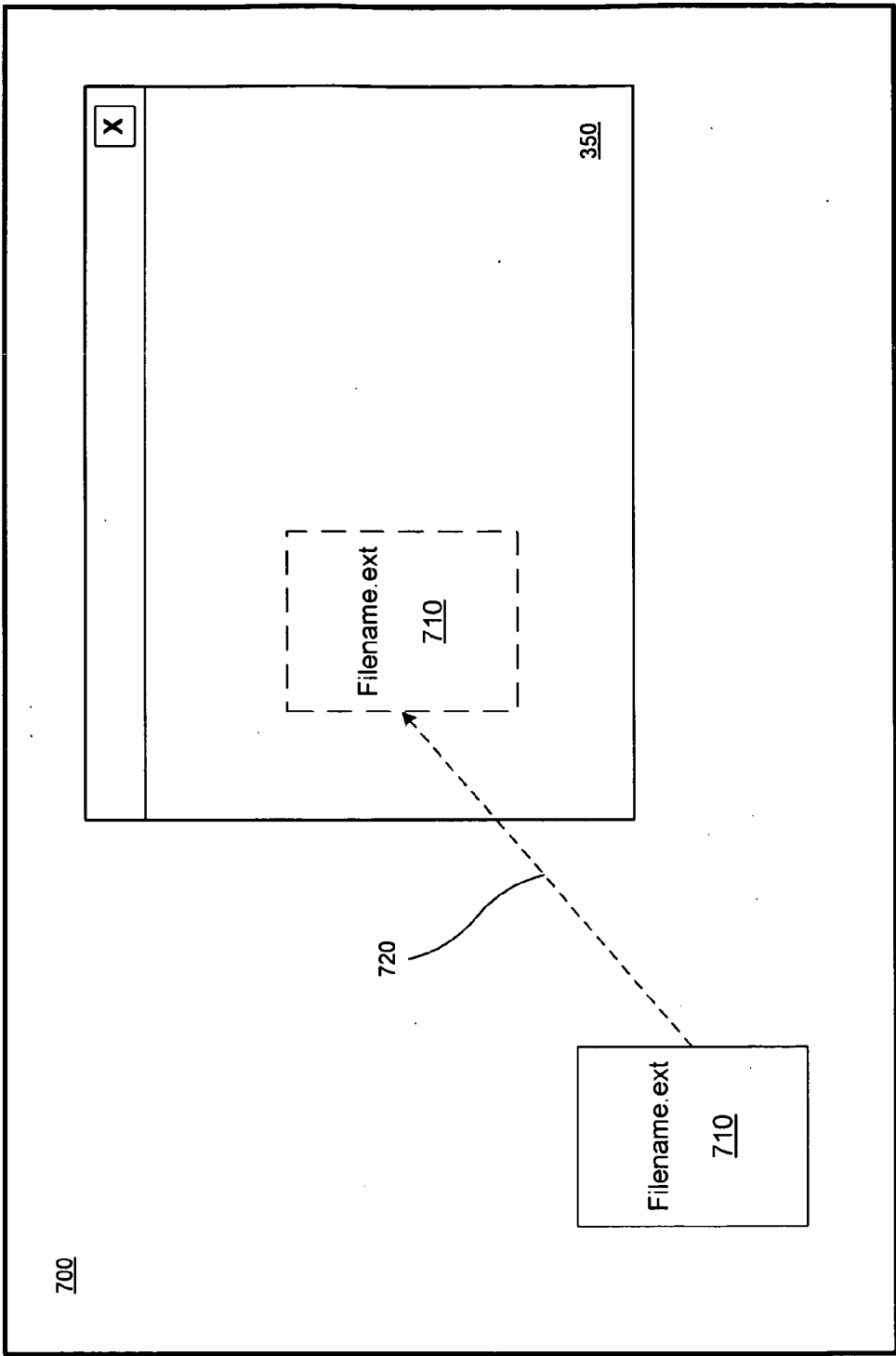


FIG. 7

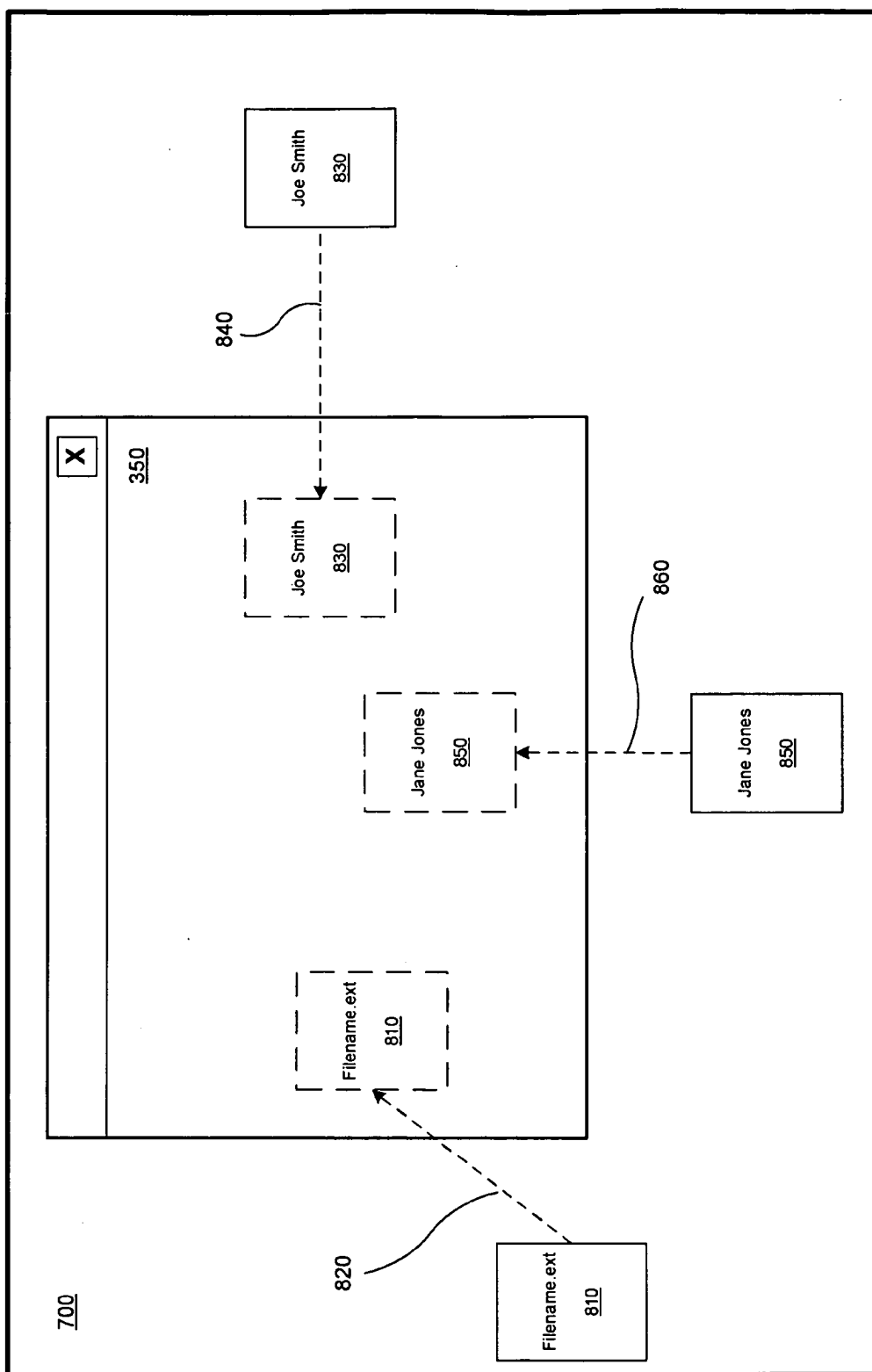


FIG. 8

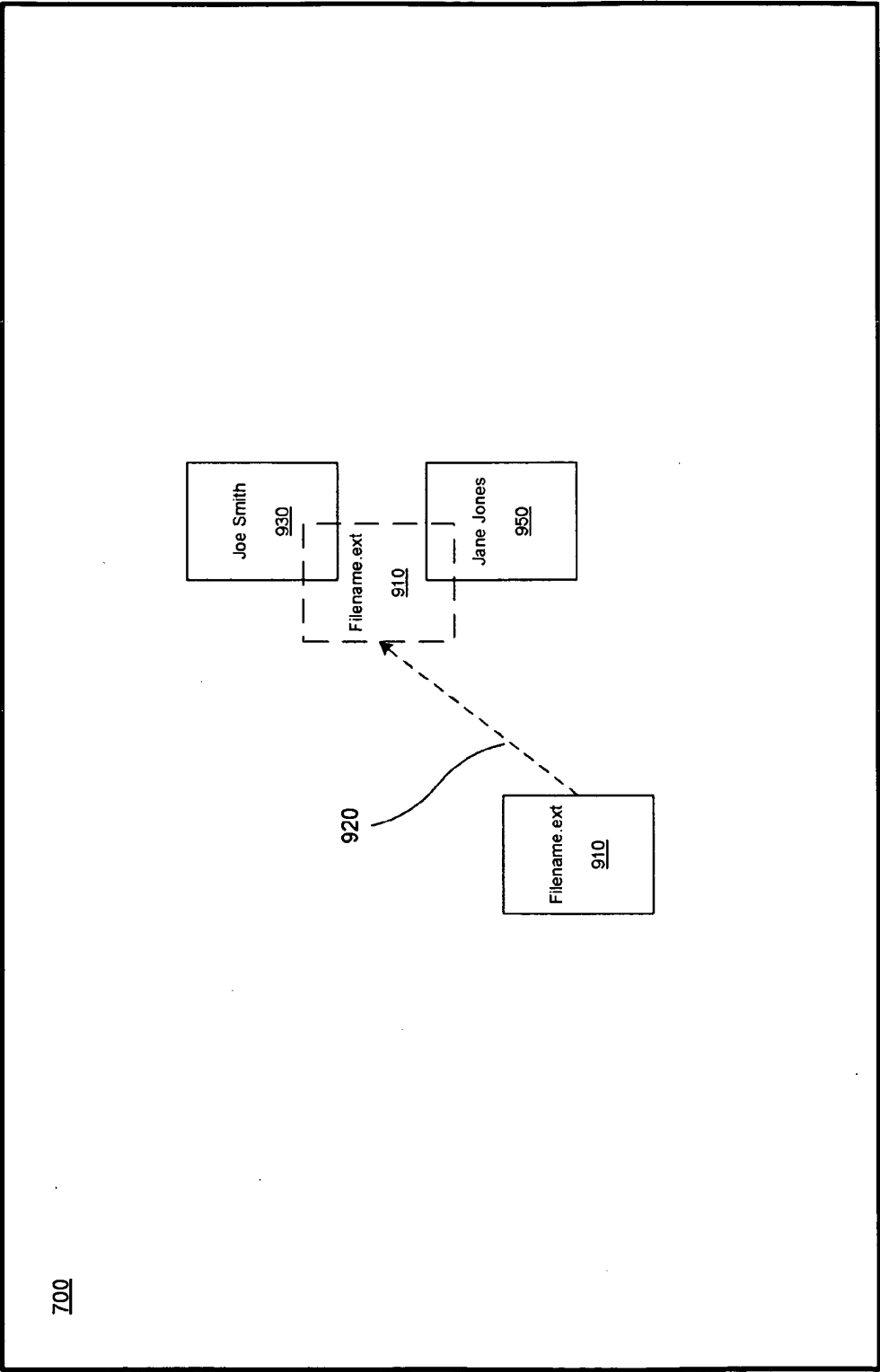


FIG. 9

ELECTRONIC FILE SHARING

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to U.S. Provisional Appl. No. 61/046,725 entitled "GRAPHICAL USER INTERFACE FOR ELECTRONIC FILE SHARING" and filed Apr. 21, 2008, which is hereby incorporated by reference in its entirety.

FIELD OF THE INVENTION

[0002] This invention relates generally to computer-implemented processes and, more specifically, to sharing of electronic files among electronic devices.

BACKGROUND OF THE INVENTION

[0003] Computer-network users typically exchange large files using awkward prior-art methods such as e-mail attachments or file transfer protocol (FTP) sites. Most email systems limit the size of attachments, and, as a result, users are forced to send multiple e-mails with discrete files attached to each. FTP sites are typically only moderately secure and their use is often difficult to manage for inexperienced users.

[0004] Other problems with the prior art not described above can also be overcome using the teachings of embodiments of the present invention, as would be readily apparent to one of ordinary skill in the art after reading this disclosure.

SUMMARY OF THE INVENTION

[0005] In an embodiment, a system includes a server and a source client executable on a first electronic device in communication with the server. The source client is operable to make available to the server a set of files modifiable by a user of the first electronic device. The system further includes a first recipient client executable on a second electronic device in communication with the server. The first recipient client is configurable by a user of the second electronic device to automatically receive from the server only a subset of the files having a predetermined characteristic.

BRIEF DESCRIPTION OF THE DRAWING

[0006] Preferred and alternative embodiments of the present invention are described in detail below with reference to the following drawings.

[0007] FIG. 1 is a schematic view of an exemplary operating environment in which an embodiment of the invention can be implemented;

[0008] FIG. 2 is a functional block diagram of an exemplary operating environment in which an embodiment of the invention can be implemented;

[0009] FIG. 3 is a functional block diagram illustrating file sharing according to an embodiment of the invention;

[0010] FIG. 4 is a flow diagram illustrating a first method according to an embodiment of the invention;

[0011] FIG. 5 is a flow diagram illustrating a second method according to an embodiment of the invention;

[0012] FIG. 6 is an illustration of a badged icon according to an embodiment of the invention;

[0013] FIG. 7 is an illustration of a graphical data-sharing technique according to a first embodiment of the invention;

[0014] FIG. 8 is an illustration of a graphical data-sharing technique according to a second embodiment of the invention; and

[0015] FIG. 9 is an illustration of a graphical data-sharing technique according to a third embodiment of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0016] An embodiment of the invention leverages remote programming concepts by utilizing processes called mobile agents (sometimes referred to as mobile objects or agent objects). Generally speaking, these concepts provide the ability for an object (the mobile agent object) existing on a first ("host") computer system to transplant itself to a second ("remote host") computer system while preserving its current execution state. The operation of a mobile agent object is described briefly below.

[0017] The instructions of the mobile agent object, its preserved execution state, and other objects owned by the mobile agent object are packaged, or "encoded," to generate a string of data that is configured so that the string of data can be transported by all standard means of communication over a computer network. Once transported to the remote host, the string of data is decoded to generate a computer process, still called the mobile agent object, within the remote host system. The decoded mobile agent object includes those objects encoded as described above and remains in its preserved execution state. The remote host computer system resumes execution of the mobile agent object which is now operating in the remote host environment.

[0018] While now operating in the new environment, the instructions of the mobile agent object are executed by the remote host to perform operations of any complexity, including defining, creating, and manipulating data objects and interacting with other remote host computer objects.

[0019] FIG. 1 illustrates an example of a suitable computing system environment 100 on which the invention may be implemented. The computing system environment 100 is only one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention. Neither should the computing environment 100 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment 100.

[0020] Embodiments of the invention are operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the invention include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

[0021] Embodiments of the invention may be described in the general context of computer-executable instructions, such as program modules, being executed by a computer and/or by computer-readable media on which such instructions or modules can be stored. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. The invention may also be practiced in distributed computing environments where tasks are performed by

remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote computer storage media including memory storage devices.

[0022] With reference to FIG. 1, an exemplary system for implementing the invention includes a general purpose computing device in the form of a computer 110. Components of computer 110 may include, but are not limited to, a processing unit 120, a system memory 130, and a system bus 121 that couples various system components including the system memory to the processing unit 120. The system bus 121 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus also known as Mezzanine bus.

[0023] Computer 110 typically includes a variety of computer readable media. Computer readable media can be any available media that can be accessed by computer 110 and includes both volatile and nonvolatile media, removable and non-removable media. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. Computer storage media includes both volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computer 110. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of the any of the above should also be included within the scope of computer readable media.

[0024] The system memory 130 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 131 and random access memory (RAM) 132. A basic input/output system 133 (BIOS), containing the basic routines that help to transfer information between elements within computer 110, such as during start-up, is typically stored in ROM 131. RAM 132 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 120. By way of example, and not limitation, FIG. 1 illustrates operating system 134, application programs 135, other program modules 136, and program data 137.

[0025] The computer 110 may also include other removable/non-removable, volatile/nonvolatile computer storage

media. By way of example only, FIG. 1 illustrates a hard disk drive 140 that reads from or writes to non-removable, non-volatile magnetic media, a magnetic disk drive 151 that reads from or writes to a removable, nonvolatile magnetic disk 152, and an optical disk drive 155 that reads from or writes to a removable, nonvolatile optical disk 156 such as a CD ROM or other optical media. Other removable/non-removable, volatile/nonvolatile computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 141 is typically connected to the system bus 121 through a non-removable memory interface such as interface 140, and magnetic disk drive 151 and optical disk drive 155 are typically connected to the system bus 121 by a removable memory interface, such as interface 150.

[0026] The drives and their associated computer storage media discussed above and illustrated in FIG. 1, provide storage of computer readable instructions, data structures, program modules and other data for the computer 110. In FIG. 1, for example, hard disk drive 141 is illustrated as storing operating system 144, application programs 145, other program modules 146, and program data 147. Note that these components can either be the same as or different from operating system 134, application programs 135, other program modules 136, and program data 137. Operating system 144, application programs 145, other program modules 146, and program data 147 are given different numbers here to illustrate that, at a minimum, they are different copies. A user may enter commands and information into the computer 20 through input devices such as a keyboard 162 and pointing device 161, commonly referred to as a mouse, trackball or touch pad. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 120 through a user input interface 160 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A monitor 191 or other type of display device is also connected to the system bus 121 via an interface, such as a video interface 190. In addition to the monitor, computers may also include other peripheral output devices such as speakers 197 and printer 196, which may be connected through an output peripheral interface 190.

[0027] The computer 110 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 180. The remote computer 180 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 110, although only a memory storage device 181 has been illustrated in FIG. 1. The logical connections depicted in FIG. 1 include a local area network (LAN) 171 and a wide area network (WAN) 173, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

[0028] When used in a LAN networking environment, the computer 110 is connected to the LAN 171 through a network interface or adapter 170. When used in a WAN networking environment, the computer 110 typically includes a modem 172 or other means for establishing communications over the WAN 173, such as the Internet. The modem 172, which may

be internal or external, may be connected to the system bus 121 via the user input interface 160, or other appropriate mechanism. In a networked environment, program modules depicted relative to the computer 110, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, FIG. 1 illustrates remote application programs 185 as residing on memory device 181. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

[0029] Referring now to FIG. 2, an embodiment of the present invention can be described in the context of an exemplary computer network system 200 as illustrated. System 200 includes electronic user devices 210, 280, such as personal computers or workstations, that are linked via a communication medium, such as a network 220 (e.g., the Internet), to an electronic device or system, such as a server 230. The server 230 may further be coupled, or otherwise have access, to a database 240, electronic storage 270 and a computer system 260. Although the embodiment illustrated in FIG. 2 includes one server 230 coupled to two user devices 210, 280 via the network 220, it should be recognized that embodiments of the invention may be implemented using two or more such user devices coupled to one or more such servers.

[0030] In an embodiment, each of the user devices 210, 280 and server 230 may include all or fewer than all of the features associated with the computer 110 illustrated in and discussed with reference to FIG. 1. User devices 210, 280 include or are otherwise coupled to a computer screen or display 250, 290, respectively. User devices 210, 280 can be used for various purposes including both network- and local-computing processes.

[0031] The user devices 210, 280 are linked via the network 220 to server 230 so that computer programs, such as, for example, a browser or other applications, running on the user devices 210, 280 can cooperate in two-way communication with server 230. Server 230 may be coupled to database 240 and/or electronic storage 270 to retrieve information therefrom and to store information thereto. Additionally, the server 230 may be coupled to the computer system 260 in a manner allowing the server to delegate certain processing functions to the computer system.

[0032] Referring now to FIG. 3, illustrated is functionality of an embodiment of the invention allowing a user (not shown) of the user device 210 to transfer a file to the user device 280. In an embodiment, an administrator (not shown) of the server 230 or other appropriate electronic device transfers a file-transfer application to the user devices 210, 280 for installation thereon. Once installed on the user devices 210, 280, the file-transfer application provides file-transfer clients 310, 320 executable by the user devices 210, 280, respectively. Each of the file-transfer clients 310, 320 includes a respective mobile-agent runtime environment 330, 340. The mobile-agent runtime environment 330, 340 include portions of memory of the user devices 210, 280 dedicated to allowing a mobile object the ability to perform operations that the mobile object is programmed to carry out. Also included in the file-transfer application are user interfaces 350, 360 that are displayable on the displays 250, 290, respectively. In an embodiment, the interfaces 350, 360 correspond to respective electronic folders or windows into which files may be placed.

Each such electronic folder may be assigned a user group defining users having permission to receive and/or send files via a particular folder.

[0033] To initiate sharing of a file with a user of the user device 280, the user of the user device 210 may place a file into the folder corresponding to the interface 350 and with which the user device 280 has been associated as a recipient. The user may do so through conventional actions such as, for example, “dragging and dropping” an icon representing the file into the interface 350 or opening the file using commands associated with the interface 350. The source client 310 monitors the folder corresponding to the interface 350. Upon noting the placement of the file in the interface 350, the source client 310 polls the server 230 to determine the recipient group including, in this case, the user device 280 and places a request with the server to share the file.

[0034] In response to the file-sharing request, the server 230 returns a transfer token (not shown) the source client 310. In an embodiment, the transfer token includes a data string that functions as a transaction identifier corresponding to the file. Upon receiving the transfer token, the source client 310 creates a transfer mobile object (TMO) 370 that includes the token and information characterizing the file (e.g., recipient identification, sender identification, file name, file workspace (i.e., name of folder corresponding to the interface 350 and into which the file was placed for sharing), file size, etc.) (collectively, “file-transfer information”).

[0035] The TMO 370 migrates to a mobile-agent runtime environment 380 executing on the server 230 and creates a proxy object 390 that functions as the TMO’s server-side agent for the file transfer. In an embodiment, the proxy 390 is, itself, a mobile object. After creating the proxy 390, the TMO 370 returns to the source client 310.

[0036] Once the TMO 370 returns to the user device 210, it establishes a network connection with the proxy 390. Subsequently, the TMO 370 reads the file and serially transfers discrete portions of the file to the proxy 390. In an embodiment, the file portions are approximately 64 kilobytes in size, although small or larger portions may be similarly employed. Upon receiving a file portion, the proxy 390 provides an electronic acknowledgement receipt to the TMO 370. The TMO 370 provides a successive file portion only after receiving an acknowledgement receipt from the proxy 390. Additionally, the TMO 370 periodically informs the source client 310 of the file-transfer status, which may be viewable by the user on the display 250. Once the entire file has been transferred to the proxy 390, the TMO 370 informs the source client 310 of the file-transfer completion and is then terminated.

[0037] In an embodiment, the network connection between the TMO 370 and proxy 390 has a failure-recovery mode. For example, if the network connection is lost, the proxy 390 is terminated, the TMO 370 reinitiates the above procedure (i.e. set up proxy and network connection therewith), and informs the new proxy of the file-transfer status at the time of connection failure. Advantageously, in the event of such connection failure, the TMO 370 is configured to initiate the failure-recovery process itself, no instruction from the source client 310 or server 230 is necessary.

[0038] Upon creation of the proxy 390, the proxy requests allocation of space on the storage 270 into which the proxy may write the file data it receives. Once the entire file has been written to the storage 270, the proxy 390 informs the server 230 of the file-storage completion and is then terminated.

[0039] At or around the time that the proxy 390 is created, the TMO 370 may create a messaging mobile object (MMO) 305 that migrates to the recipient client 320 to inform the recipient client of the file being transferred from the source client 310. In doing so, the MMO provides the file-transfer information described above to the recipient client 320. In response to being so informed, the recipient client 320 creates a transfer mobile object (RTMO) 315 that includes the file-transfer information.

[0040] The RTMO 315 migrates to the mobile-agent runtime environment 380 executing on the server 230 and creates a proxy object 325 that functions as the RTMO's server-side agent for the file transfer. In an embodiment, the proxy 325 is, itself, a mobile object. After creating the proxy 325, the RTMO 315 returns to the recipient client 320.

[0041] Once the RTMO 315 returns to the user device 280, it establishes a network connection with the proxy 325. Additionally, the RTMO 315 instructs the recipient client 320 to set up an electronic file folder into which the transferred file will be placed.

[0042] In an embodiment, the proxy 325 may poll the server 230 to determine if the source proxy 390 has placed file data from the source device 210 into the storage 270. The proxy 325 reads from storage 270 each file portion stored therein by the source proxy 390 and provides each such file portion or a copy thereof to the recipient client 320. In an embodiment, the proxy 325 reads and provides the file portions in a first-in-first-out (FIFO) manner. As such, the user device 280 may receive at least one portion of the file before the user device 210 has completed transferring the file to the server 230. Moreover, if the recipient client 320 is active at the time the source client 310 commences transfer of the file, the proxy 325 may read and provide the first file portion to the recipient client before the server 230 receives a subsequent file portion from the source client. If the recipient client 320 is not active (e.g. the user device 280 is off) at the time the source client 310 commences transfer of the file, the proxy 325 may read and provide the first file portion to the recipient client after the server 230 receives a subsequent file portion from the source client.

[0043] Once the entire file has been provided to the RTMO 315, the proxy 325 informs the server 230 of the file-share completion and is then terminated. Additionally, once the entire file has been transferred to its new file folder on the user device 280, the RTMO 315 informs the recipient client 320 of the file-transfer completion and is then terminated. Upon receipt of the file data by all designated recipients, the server 230 may purge the file portions from the storage 270.

[0044] In an embodiment, each portion of the transferred file is encrypted. In alternative embodiments, file encryption can be applied by the server 230 as file portions are received from the source client 310, or by the source device 210. In the case of encryption by the server 230, the server may have its own key applicable to the entire file data set. In the case of encryption by the source device 210, encryption and corresponding respective keys may be applied to each workspace, to each file or to each file portion as desired.

[0045] Referring to FIG. 6, in an embodiment of the invention, the user may drag or otherwise place into the interface 350, or other user interface, an icon 600 displayable on the display device 250 and representing a file to be transferred. Dragging the icon 600 into the interface enables the file associated with the icon to be transferred to one or more recipients that the user or other party has associated with the

interface (e.g., a recipient group as discussed above) in what may be termed a synchronization task. Once file transfer commences, a badge 610 in the form of, for example, a progress meter is displayed on or otherwise in association with the icon 600. The badge 610 informs the user of the progress toward completion of the synchronization task. In an embodiment, completion of the synchronization task is achieved at the time that all recipients have successfully received the file.

[0046] Still referring to FIG. 6, in an embodiment of the invention, the user may drag or otherwise place into the interface 350, or other user interface, an icon 600 displayable on the display device 250 and representing a file to be transferred. Dragging the icon 600 into the interface enables the file associated with the icon to be transferred to a recipient that the user or other party has associated with the interface (e.g., a recipient group as discussed above) in what may be termed a send task. Once file transfer commences, a badge 610 in the form of, for example, a progress meter is displayed on or otherwise in association with the icon 600. The badge 610 informs the user of the progress toward completion of the file transfer to the recipient.

[0047] Referring to FIG. 7, in an embodiment of the invention, the user may drag or otherwise place into the interface 350, or other user interface, such as an electronically generated graphical window, an icon 710 displayable on a screen 700 of the display device 250 and representing a file to be transferred. In an embodiment, the icon 710 may represent a copy of the file, rather than, or in addition to, the file itself, thereby allowing transfer of the file copy rather than the file itself. Dragging, or otherwise placing, the icon 710 into the interface 350 (as indicated by the dashed line 720) using a pointer or other selection device, such as mouse 161, automatically causes the file, and/or copy thereof, associated with the icon 710 to be transferred to one or more recipients that the user or other party has associated with the interface 350 (e.g., a recipient group as discussed above). Such file transfer described with reference to FIG. 7 may be effected using techniques described with reference to FIGS. 1-5 elsewhere herein.

[0048] Referring to FIG. 8, in an embodiment of the invention, the user may drag (as indicated by the dashed line 820), using a pointer or other selection device, such as mouse 161, or otherwise place into the interface 350, or other user interface, such as an electronically generated graphical window, a first icon 810 displayable on a screen 700 of the display device 250 and representing a file to be transferred. In an embodiment, the icon 810 may represent a copy of the file, rather than, or in addition to, the file itself, thereby allowing transfer of the file copy rather than the file itself. The user may additionally drag or otherwise place (as indicated by the dashed line 840) into the interface 350 a second icon 830 (e.g., an icon of a "v-card" including an email address) displayable on the display device 250 and representing a recipient. Dragging the first and second icons 810, 830 into the interface 350 automatically causes the file associated with the first icon 810 to be transferred to the recipient associated with the second icon 830. Similarly, dragging a third icon 850 (as indicated by the dashed line 860) representing an additional recipient into the interface 350 automatically causes the file associated with the first icon 810 to be transferred to the recipient associated with the third icon 850. Such file transfers described with reference to FIG. 8 may be effected using techniques described with reference to FIGS. 1-5 elsewhere herein.

[0049] Referring to FIG. 9, in an embodiment of the invention, the user may drag or otherwise place (as indicated by the dashed line 920) a first icon 910 on or partially coinciding with (e.g., superimposed on) a second icon 930 displayable on a screen 700 of the display device 250 and, perhaps, located in the interface 350. The first icon 910 may represent a file to be transferred. In an embodiment, the first icon 910 may represent a copy of the file, rather than, or in addition to, the file itself, thereby allowing transfer of the file copy rather than the file itself. The second icon 930 may represent a recipient (i.e., may be an icon of a “v-card” including an email address). At least partially superimposing one of the first and second icons 910, 930 with the other automatically causes the file associated with the first icon 910 to be transferred to the recipient associated with the second icon 930. This transfer operation may similarly be performed with respect to a third icon 950 also representing a recipient. For example, the second and third icons 930, 950 may be simultaneously selected, placed in close proximity to one another, or otherwise coupled using known graphical-user-interface techniques. Subsequently, at least partially superimposing the first icon 910 on the second and/or third icon 930, 950 automatically causes the file associated with the first icon to be transferred to the recipients associated with the second and third icons. Such file transfers described with reference to FIG. 9 may be effected using techniques described with reference to FIGS. 1-5 elsewhere herein.

[0050] In an embodiment, file icons, such as the icon 600, in a workspace displayed on display device 290 can visually indicate whether a corresponding file (such as a file, the transfer of which may be as is discussed herein with reference to FIG. 3) has been downloaded to or is otherwise stored on (i.e., is local to), or has otherwise been made available (e.g., sent or in transit) to the user device 280. For example, if the source client 310 has initiated, or has otherwise attempted to initiate, the sharing of a file with the recipient client 320, the interface 360 associated with the user device 280 may display an icon, such as icon 600, identifying or otherwise representing the file to be shared by source client 310. If the corresponding file (i.e., file to be shared by source client 310) is not local to user device 280, the icon 600 may be ghosted (i.e., semi-transparent) or otherwise be visually configured to indicate the non-local status of the file. As such, the icon 600 can appear in the workspace in some manner whether or not the data associated with the icon 600 has actually been downloaded to, or otherwise received by, the user device 280.

[0051] In such a situation, and in an embodiment, a user may right-click or otherwise indicate the icon 600, and from a resultant pop-up contextual menu (not shown), for example, that appears on the display 290, make a selection from the menu to request receipt or download of the associated file, at which time the icon 600 will become unghosted or will otherwise be visually configured to indicate the pending-local or local status of the file. In an embodiment, there may also be included a selection feature, accessible from a pop-up contextual menu that may be generated to display 290, to allow the user to, with a single selection from the menu, similarly download all files that are not local to user device 280.

[0052] In an embodiment, files can be downloaded to a user device 280 according to a user-defined “subscription.” With such a subscription, users can elect, on a per-workspace basis, to always download all files that have been made available to the user, or never download all such files. For purposes of this description, the “workspace” corresponds to a respective

electronic folder assigned to a particular user group, as discussed in further detail above herein with reference to FIG. 3. As such, access to each workspace is limited to a predefined and finite set of users on a given network. Files, when created or modified on, for example, user device 210 can be categorized according to particular characteristics of each file such as, by way of non-limiting example, by author, by source, or by subject matter. Users can also be able to specify a subset of the files (e.g., by author, by source, by subject matter, etc.) to always download. In this context, the term “always download” means that when someone remotely updates a file (e.g., modifies or creates a file), such file will be downloaded to a user-specified machine (e.g., user device 280) without specific instruction by the recipient. As such, an embodiment provides a mechanism for specifying the manner in which files stored on a remote computing device are automatically shared or otherwise synchronized with a local computing device, and vice-versa.

[0053] In an embodiment, if a user logs into a new computing device (i.e., a computing device on which the user hasn’t previously logged) with the user’s account information, all of the user’s workspaces will be available on the new computing device. Whether the files associated with such workspaces are downloaded to the new machine depends on the subscription parameters, described above, the user has set.

[0054] Multiple versions of every file in any workspace can be stored on the server 230. As such, in an embodiment, a user can right-click on, or otherwise indicate, a file icon associated with a particular file in the workspace and select from among multiple versions of the associated file to download to, for example, the user device 280. When a user selects a version of the associated file different from the most-recently modified version of the file, the user’s subscription may be modified so that future updated versions of that file are not automatically downloaded to the user device 280 or other device on which the user can access the associated workspace. When the user is finished looking at the selected version, for example, the user can invoke a menu to select a “newest version” option that resets the subscription so that the user resumes periodically and automatically receiving the newest version of a file as created or modified. File icons can be presented in a format, as similarly discussed elsewhere herein, to indicate whether a file is the newest version.

[0055] FIG. 4 illustrates a process 400 according to an embodiment of the invention. The process 400 is implementable in an electronic system coupled to or including a storage device. The process 400 is illustrated as a set of operations shown as discrete blocks. The process 400 may be implemented in any suitable hardware, software, firmware, or combination thereof. The order in which the operations are described is not to be necessarily construed as a limitation.

[0056] At a block 410, a first object is executed. The first object is operable to receive a plurality of portions of a file from a source device. Additionally, the first object is operable to store the portions to the storage device. In an embodiment, the first object is created by a mobile object created on the source device. For example, the TMO 370 creates a proxy object 390 that functions as the TMO’s server-side agent for the file transfer. Subsequently, the TMO 370 reads the file and serially transfers discrete portions of the file to the proxy 390. The proxy 390 writes the file data it receives to the storage 270.

[0057] At a block 420, a second object is executed. The second object is operable to retrieve from the storage device

a first portion of the file. The second objection is further operable to provide the first portion to a recipient device before the first object has received the entirety of the file from the source device. In an embodiment, the second object is created by a mobile object created on the recipient device. For example, The RTMO 315 creates a proxy object 325 that functions as the RTMO's server-side agent for the file transfer. The proxy 325 reads from storage 270 each file portion stored therein by the source proxy 390 and provides each such file portion to the recipient client 320. In an embodiment, the proxy 325 reads and provides the file portions in a first-in-first-out (FIFO) manner.

[0058] FIG. 5 illustrates a process 500 according to an embodiment of the invention. The process 500 is implementable in an electronic device coupled to an electronic system operable to receive a file from a source device. The process 500 is illustrated as a set of operations shown as discrete blocks. The process 500 may be implemented in any suitable hardware, software, firmware, or combination thereof. The order in which the operations are described is not to be necessarily construed as a limitation.

[0059] At a block 510, the electronic device creates a mobile object. For example, the recipient client 320 creates the RTMO 315.

[0060] At a block 520, the mobile object creates a proxy object at the electronic system. For example, the RTMO 315 migrates to the mobile-agent runtime environment 380 executing on the server 230 and creates a proxy object 325 that functions as the RTMO's server-side agent for the file transfer.

[0061] At a block 530, a network connection with the electronic system is established. For example, once the RTMO 315 returns to the user device 280, it establishes a network connection with the proxy 325.

[0062] At a block 540, a first portion of a plurality of portions of the file is retrieved from the electronic system. In an embodiment, the proxy object is operable to retrieve the file portions from a storage device coupled to the electronic system and provide the file portions to the electronic device. The first portion is received by the electronic device before the electronic system has received the entirety of the file from the source device. For example, The RTMO 315 creates a proxy object 325 that functions as the RTMO's server-side agent for the file transfer. The proxy 325 reads from storage 270 each file portion stored therein by the source proxy 390 and provides each such file portion to the recipient client 320. In an embodiment, the proxy 325 reads and provides the file portions in a first-in-first-out (FIFO) manner.

[0063] While a preferred embodiment of the invention has been illustrated and described, as noted above, many changes can be made without departing from the spirit and scope of the invention. For example, more than one recipient device may receive a shared file from the source device 210 in the manner described above with reference to the user device 280 and FIG. 3. In an embodiment, these multiple recipient devices can simultaneously receive the shared file by each deploying a respective proxy object to retrieve file portions from the server 230. Instead, the invention should be determined entirely by reference to the claims that follow.

The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

1. A computer-readable medium having computer-executable instructions that, when executed in a system coupled to a

network and a first electronic device coupled to a display device, enable the system to perform a method comprising at least the steps of:

determining that a source client executable on a second electronic device in communication with the network has made available to the first electronic device a data set not yet received by the first electronic device;

in response to determining the availability of the file not yet received by the first electronic device, generating in a first region of a screen of the display device a first icon representing the data set, the first icon having a first display format; and

in response to an action of a user of the first electronic device indicating that the user desires the first electronic device to receive the data set, generating in the first region of the screen a second icon representing the data set, the second icon having a second display format different from the first format.

2. The medium of claim 1, wherein the source client is operable to create a first mobile object, the first mobile object being operable to create a second mobile object operable to notify the first electronic device of the availability of the file.

3. The medium of claim 1, wherein the first region comprises an electronically generated graphical window.

4. The medium of claim 1, wherein the first display format comprises semi-transparency.

5. The medium of claim 1 wherein the method further comprises generating a mobile-agent object configured to carry information characterizing the data set over the network.

6. A system, comprising:

a server;

a source client executable on a first electronic device in communication with the server, the source client operable to make available to the server a set of files modifiable by a user of the first electronic device; and

a first recipient client executable on a second electronic device in communication with the server, the first recipient client being configurable by a user of the second electronic device to automatically receive from the server only a subset of the files having a predetermined characteristic.

7. The system of claim 6, wherein the predetermined characteristic comprises existing files that have been modified by the user of the first electronic device.

8. The system of claim 6, wherein the predetermined characteristic comprises files that have been created by the user of the first electronic device.

9. The system of claim 6, further comprising a second recipient client executable on a third electronic device in communication with the server, the second recipient client being configurable by a user of the third electronic device to automatically receive only a subset of the files having a predetermined characteristic.

10. The system of claim 6, wherein:

the source client is further operable to provide to the server multiple versions of a first file of the set of files;

the second electronic device is configured to generate a graphical user interface (GUI) to a display device, the GUI being configured to enable the user to select a first version of the first file to be received from the server by the second electronic device; and

the first recipient client is further configured to discontinue receiving, in response to the user selection of the first

version, the subset of the files having a predetermined characteristic.

11. The system of claim **10**, wherein the first recipient client is further configured to resume automatically receiving, in response to an instruction from the user, the subset of the files having a predetermined characteristic.

12. The system of claim **6**, wherein the first recipient client is operable to retrieve from the server a first portion of a plurality of portions of a file of the subset before the server has received the entirety of the file from the source client.

13. The system of claim **12**, wherein the source client is operable to create a first mobile object, the first mobile object being operable to create a proxy object at the server.

14. The system of claim **13**, wherein the first mobile object is operable to provide the plurality of file portions to the proxy object.

15. The system of claim **13**, wherein the proxy object is operable to store the file portions in a storage device coupled to the server.

16. The system of claim **13**, wherein the first mobile object is operable to create a second mobile object operable to notify the first recipient client of a file of the subset being sent to the server.

* * * * *