



(51) **International Patent Classification:**
G06F 9/06 (2006.01) **G06F 3/14** (2006.01)
G06F 17/24 (2006.01)

Garry [US/US]; 255 King St., Apt. 518, San Francisco, California 94107 (US).

(21) **International Application Number:**
PCT/US2009/056703

(74) **Agents:** REES, Karl T. et al.; 2055 Gateway Place, Suite 550, San Jose, California 95110 (US).

(22) **International Filing Date:**
11 September 2009 (11.09.2009)

(81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, 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, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
61/097,059 15 September 2008 (15.09.2008) US
61/122,695 15 December 2008 (15.12.2008) US

(71) **Applicant** (for all designated States except US):
PALANTIR TECHNOLOGIES, INC. [US/US]; 100 Hamilton Avenue, Suite 300, Palo Alto, California 94301 (US).

(72) **Inventors; and**

(75) **Inventors/Applicants** (for US only): **AYMELOGLU, Andrew** [US/US]; 432 High Street, Palo Alto, California 94301 (US). **MIYAKE, Nicholas** [US/US]; 460 Channing Ave., Apt. A, Palo Alto, California 94301 (US). **BURR, Brandon** [US/US]; 569 Addison Ave., Apt. 4, Palo Alto, California 94301 (US). **CICERONE, Derek** [US/US]; 145 Hawthorne Ave., Apt. A, Palo Alto, California 94301 (US). **SIMLER, Kevin** [US/US]; 3035 Emerson St., Palo Alto, California 94306 (US). **TAN,**

(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, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

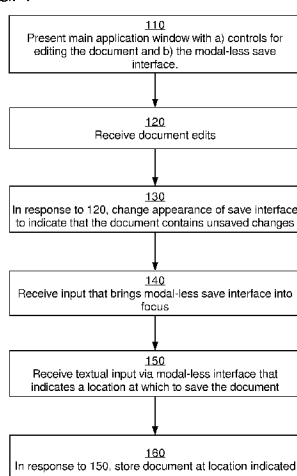
Published:

— without international search report and to be republished upon receipt of that report (Rule 48.2(g))

(54) **Title:** MODAL-LESS INTERFACE ENHANCEMENTS

FIG. 1

100



(57) **Abstract:** A modal-less save interface enables a user to provide a filename under which to save a document at any time during operation of the application, without interrupting other user tasks of the application. The user inputs a filename and performs a save operation on a document without navigating menu items or launching a modal window. The modal-less save interface may be provided in a portion of the main application window. For example, the modal-less save interface may be presented in a toolbar, taskbar, status bar, window pane, or sidebar. An application may feature a modal-less access control interface. The modal-less access control interface may be provided in a portion of the main application window. The access control interface may comprise a control, such as link, button, checkbox, text box, or pull-down menu. Upon selection of the control, the application sets one or more access control levels for the current document.

MODAL-LESS INTERFACE ENHANCEMENTS

FIELD OF THE INVENTION

[0001] Embodiments of the invention described herein relate generally to document management, and, more specifically, to techniques for providing save and access control interfaces in a document authoring application.

BACKGROUND

[0002] The approaches described in this section are approaches that could be pursued, but not necessarily approaches that have been previously conceived or pursued. Therefore, unless otherwise indicated, it should not be assumed that any of the approaches described in this section qualify as prior art merely by virtue of their inclusion in this section.

[0003] Graphical interface design techniques traditionally rely upon modal windows to provide document saving functionality within applications. A “modal window” is a child window of another application window that is generated when the application enters a particular operating mode. Modal windows are often referred to as modal dialogs in that they are used to display dialogs that solicit user input, such as a save dialog.

[0004] A problem with using a modal window to facilitate document saving functionality is that modal windows interrupt application workflow. For example, to save a document in an application such as Microsoft Word, a user must navigate to and select a menu item in a toolbar that launches the modal window for the save dialog. The user must then wait for the application to launch the modal window. The modal window for the save dialog then appears over the main application window, and the user cannot edit a document or perform any other function with the application until the save operation is complete. To complete the operation, the user must select a folder in which to save the document and a filename for the document, then click a button to trigger the save operation.

[0005] Because save operations interrupt an application’s workflow, it is inefficient for users to save their work. This inefficiency is especially pronounced when new or different filenames are frequently required.

[0006] In one approach, once a filename has been identified, save dialogs typically “remember” the filename, so as to cut down on the amount of work required of a user. However, the application must still launch a modal window for the save dialog. The delay between a user selecting the menu item and the modal window appearing can often be one of the biggest sources of inefficiency in the save process. Furthermore, the modal window still interrupts the workflow of the main application window.

[0007] In another approach, shortcut keys facilitate rapid saving of a document under the same filename. However, the user must still launch a save dialog in a modal window when the document does not have a name, or when the user wants to change the name.

[0008] A similar problem exists in interfaces for managing access rights for a document. Document access control operations in graphical interfaces are inefficient in that they traditionally require a user to navigate through complex menus and modal windows. Furthermore, it is often not apparent to a user that it is even possible to control the level of access that other users will have to their documents. Therefore, many users fail to properly protect access to their documents.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

[0010] FIG. 1 is a flow chart illustrating a method for providing modal-less save functionality;

[0011] FIG. 2 is a flow chart illustrating a method for performing a save operation that is supported by modal-less save functionality;

[0012] FIG. 3 is a flow chart illustrating a method for providing modal-less access control management for an object;

[0013] FIG. 4 depicts a modal-less save interface and access control interface within a main application window;

[0014] FIG. 5 illustrates how a software application may assign a different appearance to a modal-less save interface to indicate the save status of a document being edited within a main application window; and

[0015] FIG. 6 is block diagram of a computer system upon which embodiments of the invention may be implemented.

DETAILED DESCRIPTION

[0016] In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

[0017] Embodiments are described herein according to the following outline:

1.0. General Overview

- 1.1. Terms and Definitions
- 2.0. Functional Overview
 - 2.1. Modal-Less Save Interface
 - 2.2. Modal-Less Access Control Interface
- 3.0. Implementation Examples
 - 3.1. Example Modal-less Save Interface
 - 3.2. Example Modal-less Access Control Interface
 - 3.3. Save Status Feedback
 - 3.4. Access Levels
- 4.0. Implementation Mechanism—Hardware Overview
- 5.0. Extensions and Alternatives
- 6.0. Conclusion

1.0. GENERAL OVERVIEW

[0018] Approaches, techniques, and mechanisms are disclosed for modal-less save and access control operations. According to an embodiment, a modal-less save interface enables a user to provide a filename under which to save a document at any time during operation of the application, without interrupting other user tasks of the application. Various controls allow a user to input a filename and perform a save operation on a document without the user having to navigate menu items or launch a modal window. The modal-less save interface may be provided in a portion of the main application window. For example, the modal-less save interface may be presented in a toolbar, taskbar, status bar, window pane, sidebar, or any other component of the application window.

[0019] According to an embodiment, an application may feature a modal-less access control interface. The modal-less access control interface may be provided in a portion of the main application window. The access control interface may comprise a control, such as link, button, checkbox, text box, or pull-down menu. Upon selection, the control sets the access control level for the document currently open within the application window. The access control interface may be accessed at any time during a user's interaction with the main application window.

[0020] In other aspects, the invention encompasses a computer apparatus and a computer-readable medium configured to carry out the foregoing steps.

1.1. TERMS AND DEFINITIONS

[0021] For convenience, various embodiments of the techniques described herein are described with respect to “documents.” As used herein, a document is any type of object,

including files and other resources, that embodies, encapsulates, or encodes information in structured data. Specifically, the data for a document is structured in such a manner so as to be understood by one or more document authoring applications. An application understands structured data when the application, upon reading the data, may perform various actions with the information represented by the object, such as presenting the information to a user. Documents may include word processing files, PDF files, images, and any other form of structured data. Moreover, a document authoring application is any application that may interpret a document in such a manner so as to deliver one or more visual presentations of the information embodied, encapsulated, or encoded within the document.

2.0. FUNCTIONAL OVERVIEW

2.1. MODAL-LESS SAVE INTERFACE

[0022] FIG. 1 is a flow chart 100 illustrating a method for providing modal-less save functionality, according to an embodiment. The illustrated embodiment is exemplary only—other embodiments may implement more or fewer steps, in different orders.

[0023] At step 110, an application executing at a computing device presents to a user an application window for working with a document. This application window includes at least: a) one or more controls for editing the document and b) a modal-less save interface. For example, the application may be a Palantir Finance client, and the document may be a financial analysis document that the user has newly created. The one or more controls may be any controls suitable for accepting user input for editing the document, including text fields, buttons, pull-down menus, and drag-and-drop elements. The save interface includes an editable text box. The save interface may optionally include a save button. An example application window is described in section 3.1.

[0024] At step 120, the application receives, via the one or more controls, one or more commands to add information to or modify information in the document. For example, a user may select and modify text displayed for the object in the application window.

[0025] At step 130, in response to step 120, the application changes the appearance of the save interface to indicate that the opened document contains unsaved changes. For example, the application may change a background color or font in the save interface. Other techniques for indicating the save status of a document are discussed in section 3.3.

[0026] At step 140, the application receives input that brings the save interface into focus. For example, a user may use a mouse or enter a keystroke combination to bring the editable text box into focus.

[0027] At step 150, while the save interface is in focus, the application receives textual input that indicates a location at which to save the object. For example, the user may then enter a name, such as “object1”, in the editable text box.

[0028] In an embodiment, the input of step 150 may not necessarily specify an exact file system location at which the document is to be saved; rather, the entered text may be used to determine the location at which to save the document. For example, the application may transparently append extensions or other identifiers to the file name. The application may also assume that the entered file name is relative to some path—for example, a root directory on a server at which the application stores its documents. In another embodiment, the entered file name may instead exactly specify the save location with, for example, an absolute path.

[0029] In some embodiments, a file name may already be present in the editable text box. For example, in the case of a newly created document, the application may pre-populate the text box with a generic name. In the case of a previously saved document, the application may have already filled in the text box with the name under which the document was last saved. When a file name is already present in the text box, the application may allow a user to simply edit that file name. Or, the application may let the user type in an entirely new file name.

[0030] In an embodiment, the user terminates the input by pressing an Enter key, thereby indicating that the file name has been fully entered. In another embodiment, the user terminates the input by clicking on a save button in the save interface. In an embodiment, the user terminates the input by using the mouse or keyboard to switch focus away from the save interface. Yet other embodiments may rely on other or additional techniques for terminating the input of step 140.

[0031] At step 160, in response to the input of step 150, the application stores the object at the location indicated by the input of step 150.

[0032] FIG. 2 is a flow chart 200 illustrating a method for performing a save operation that is supported by modal-less save functionality, according to an embodiment of the invention. The method takes advantage of the fact that the modal-less save interface does not restrict the application’s workflow, even when assigning a new name to the saved object. Thus, unlike many conventional save interfaces, save operations facilitated by a modal-less save interface may be interrupted while the user performs other document-related tasks. Again, the illustrated embodiment is exemplary only—other embodiments may implement more or fewer steps, in different orders.

[0033] At step 210, an application receives, in a save interface, input specifying a first set of one or more characters of a name under which a currently open object is to be saved. For

example, suppose that the user intended to name the currently open document “object1.” The user may enter the first three characters “obj,” and thereby accomplish step 210.

[0034] At step 220, subsequent to step 210, the application receives input shifting focus from the save interface and to one or more controls for editing the object. For example, while typing a name under which to save the object, the user may have noticed additional modifications that needed to be made to the object. The user may have then immediately clicked on an editing control so as to make the change.

[0035] At step 230, subsequent to step 220, the application receives input via the one or more editing controls. The input indicates one or more edits to the currently open object.

[0036] At step 240, upon completion of the edits, the application receives input shifting focus back to the save interface.

[0037] At step 250, while focused again at the save interface, the application may receive input specifying a second set of one or more characters of the name. For example, the user may enter “ect” to complete the document’s name. In other words, the document’s name comprises the second set of characters (“ect”) added to the first set of characters (“obj”).

[0038] At step 260, subsequent to step 250, the application may receive input indicating that the name has been fully entered. For example, the user may click on a save button or press an Enter key.

[0039] At step 270, in response to the input of step 260, the application may save the object at a location based upon the specified document name.

2.2. MODAL-LESS ACCESS CONTROL INTERFACE

[0040] FIG. 3 is a flow chart 300 illustrating a method for providing modal-less access control management for an object, according to an embodiment. The illustrated embodiment is exemplary only—other embodiments may implement more or fewer steps, in different orders.

[0041] At step 310, an application executing at a computing device presents to a user a single application window for working with a document. This single application window includes at least: a) one or more controls for editing the document and b) a modal-less save interface. The one or more controls may be any controls suitable for accepting user input for editing the document, including text fields, buttons, pull-down menus, and drag-and-drop elements. The access control interface may include one or more controls for configuring access permissions for the document. For example, the access control interface may include one or more of: a pull-down box listing two or more access control levels, a set of one or more links specifying access control levels, or a set of one or more checkboxes specifying access control levels. An example application window is described in section 3.2.

[0042] At step 320, the application receives, via the one or more controls, one or more commands to add information to or modify information in the document. For example, a user may select and modify text displayed for the object in the application window.

[0043] At step 330, the application receives input that brings the access control interface into focus. For example, a user may use a mouse or enter a keystroke combination to bring the editable text box into focus.

[0044] At step 340, while the access control interface is in focus, the application receives input that specifies one or more access levels for the document. For example, the user may click on a pull-down menu and select one of the listed access control levels. Example access levels are discussed in section 3.4.

[0045] At step 350, the application changes permission metadata for the object in accordance with the input of step 340.

3.0. IMPLEMENTATION EXAMPLES

3.1. EXAMPLE MODAL-LESS SAVE INTERFACE

[0046] FIG. 4 depicts a modal-less save interface 410 within a main application window 400, according to an embodiment of the invention. Main application window 400 is the main graphical interface for a software application executing on a computer. Main application window 400 features a variety of controls that allow a user to edit a document. Although FIG. 4 depicts the editing of a financial report document, the document edited in main application window 400 may be any type of document, including a word processing document, spreadsheet, web page, data file, email, and so on.

[0047] A portion of main application 400 is occupied by save interface 410. Save interface 410 comprises a text entry control 420 and a save control 430. While interacting with application window 400, a user may at any time input a filename in text entry box 420. In response to receiving input via text entry control 420, the software application may save the document with the designated filename. For example, a user may type the filename. Then, user may signal that the filename has been fully entered by pressing the ENTER key or clicking on a button of a pointing device.

[0048] Text entry control 420 may also be configured to receive input designating a folder or directory for saving the document.

[0049] Save control 430 is a button that, when selected, causes the software application to save the currently opened document with the filename designated in text entry control 420. A user may select save control 430 at any time when working within main application window 400 by, for instance, clicking on save control 430.

[0050] The user may enter a different name in text entry control 420 at any time. When the user enters a new name in the text entry control 420 and selects save control 430, the current document is saved using the newly entered name. No modal window or save dialog is displayed and the user's work is not interrupted. For example, as illustrated by the method of FIG. 2, it is possible for a user to enter just one or two characters of a new filename in the text entry control 420, then interact with other features of the application without completing a save operation, then complete the new filename, then select the save control 430. In this way, the user can initiate a save operation, perform other application tasks, and then later complete the save operation at a convenient time.

[0051] Moreover, saving different versions of the document under different filenames becomes as simple as entering each different filename in the text entry control 420 and selecting the save control 430. The user can interrupt any interaction with the text entry control 420 at any time without having to close a modal window or suffer other interruptions.

3.2. EXAMPLE MODAL-LESS ACCESS CONTROL INTERFACE

[0052] The main application window 400 of FIG. 4 also comprises an access control interface 440, according to an embodiment of the invention. Access control interface 440 is a link. When a user clicks on access control interface 440, the application sets the access control level for the currently open document to "Public." As an example, the access control interface 440 is depicted as part of save interface 410. However, access control interface 440 may also be entirely separate from save interface 410.

3.3. SAVE STATUS FEEDBACK

[0053] According to an embodiment of the invention, an application may utilize a modal-less save interface to provide a user with feedback indicating the status of saving a current document. The application may assign different appearances to the save interface depending on the save status of the document. In this manner, the application may indicate to a user that a document has not been saved within a certain time period or that the document has changed since it was last saved.

[0054] For example, the save interface may be displayed using a new color, font size or style, or interface size. To more visibly call attention to the save status of a current document, the appearance of the save interface may oscillate between two different states. For example, the controls of the save interface may feature a distinctive color that periodically becomes more saturated and less saturated, thus appearing to fade in and out or appearing to "glow."

[0055] FIG. 5 illustrates how a software application may assign a different appearance to modal-less save interface 410 to indicate the save status of a document being edited within main application window 400, according to an embodiment of the invention. In contrast with text

entry control 420 of FIG. 4, the border of text entry control 520 of FIG. 5 is depicted as “glowing,” indicating that the current financial document contains information that should be saved. Because of the resulting different appearance of save control 530, a user of the application is more likely to notice save control 530 and perform a save operation.

3.4. ACCESS LEVELS

[0056] In some embodiments, the access control interface may only allow a user to designate the currently open document as public or private. In other embodiments, the access control interface may allow a user to set the access control level for a document to one of a variety of access control levels. Each access control level may permit or prohibit various levels of document access to different users or user groups. The various levels of document access may include, for example, permissions to read, write, or modify the document, as well as permissions to be aware that the document exists.

[0057] For example, an access control interface may feature a pull-down menu listing access control levels of “Public,” meaning that the document may be accessed by all users, “Restricted to Group,” meaning that the document may only be accessed by the user’s group, and “Restricted to User,” meaning that the document may only be accessed by the user. While editing a document in the main application window, the user may select one of these levels from the pull-down menu. In response, the application may instantly set the access control level for the document to correspond to the selected access control level.

[0058] Access control levels may be built-in to the application, or they may be configured on a server and then imported into the application. Data indicating access control levels for a document may be stored as metadata within the document, as metadata on a server at which the document is stored, or as metadata in a file system.

4.0. IMPLEMENTATION MECHANISM—HARDWARE OVERVIEW

[0059] According to one embodiment, the techniques described herein are implemented by one or more special-purpose computing devices. The special-purpose computing devices may be hard-wired to perform the techniques, or may include digital electronic devices such as one or more application-specific integrated circuits (ASICs) or field programmable gate arrays (FPGAs) that are persistently programmed to perform the techniques, or may include one or more general purpose hardware processors programmed to perform the techniques pursuant to program instructions in firmware, memory, other storage, or a combination. Such special-purpose computing devices may also combine custom hard-wired logic, ASICs, or FPGAs with custom programming to accomplish the techniques. The special-purpose computing devices may be desktop computer systems, portable computer systems, handheld devices, networking

devices or any other device that incorporates hard-wired and/or program logic to implement the techniques.

[0060] For example, FIG. 6 is a block diagram that illustrates a computer system 600 upon which an embodiment of the invention may be implemented. Computer system 600 includes a bus 602 or other communication mechanism for communicating information, and a hardware processor 604 coupled with bus 602 for processing information. Hardware processor 604 may be, for example, a general purpose microprocessor.

[0061] Computer system 600 also includes a main memory 606, such as a random access memory (RAM) or other dynamic storage device, coupled to bus 602 for storing information and instructions to be executed by processor 604. Main memory 606 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 604. Such instructions, when stored in storage media accessible to processor 604, render computer system 600 into a special-purpose machine that is customized to perform the operations specified in the instructions.

[0062] Computer system 600 further includes a read only memory (ROM) 608 or other static storage device coupled to bus 602 for storing static information and instructions for processor 604. A storage device 610, such as a magnetic disk or optical disk, is provided and coupled to bus 602 for storing information and instructions.

[0063] Computer system 600 may be coupled via bus 602 to a display 612, such as a cathode ray tube (CRT), for displaying information to a computer user. An input device 614, including alphanumeric and other keys, is coupled to bus 602 for communicating information and command selections to processor 604. Another type of user input device is cursor control 616, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 604 and for controlling cursor movement on display 612. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), that allows the device to specify positions in a plane.

[0064] Computer system 600 may implement the techniques described herein using customized hard-wired logic, one or more ASICs or FPGAs, firmware and/or program logic which in combination with the computer system causes or programs computer system 600 to be a special-purpose machine. According to one embodiment, the techniques herein are performed by computer system 600 in response to processor 604 executing one or more sequences of one or more instructions contained in main memory 606. Such instructions may be read into main memory 606 from another storage medium, such as storage device 610. Execution of the sequences of instructions contained in main memory 606 causes processor 604 to perform the

process steps described herein. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions.

[0065] The term “storage media” as used herein refers to any media that store data and/or instructions that cause a machine to operation in a specific fashion. Such storage media may comprise non-volatile media and/or volatile media. Non-volatile media includes, for example, optical or magnetic disks, such as storage device 610. Volatile media includes dynamic memory, such as main memory 606. Common forms of storage media include, for example, a floppy disk, a flexible disk, hard disk, solid state drive, magnetic tape, or any other magnetic data storage medium, a CD-ROM, any other optical data storage medium, any physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, NVRAM, any other memory chip or cartridge.

[0066] Storage media is distinct from but may be used in conjunction with transmission media. Transmission media participates in transferring information between storage media. For example, transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 602. Transmission media can also take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0067] Various forms of media may be involved in carrying one or more sequences of one or more instructions to processor 604 for execution. For example, the instructions may initially be carried on a magnetic disk or solid state drive of a remote computer. The remote computer can load the instructions into its dynamic memory and send the instructions over a telephone line using a modem. A modem local to computer system 600 can receive the data on the telephone line and use an infra-red transmitter to convert the data to an infra-red signal. An infra-red detector can receive the data carried in the infra-red signal and appropriate circuitry can place the data on bus 602. Bus 602 carries the data to main memory 606, from which processor 604 retrieves and executes the instructions. The instructions received by main memory 606 may optionally be stored on storage device 610 either before or after execution by processor 604.

[0068] Computer system 600 also includes a communication interface 618 coupled to bus 602. Communication interface 618 provides a two-way data communication coupling to a network link 620 that is connected to a local network 622. For example, communication interface 618 may be an integrated services digital network (ISDN) card, cable modem, satellite modem, or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 618 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 618 sends and

receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

[0069] Network link 620 typically provides data communication through one or more networks to other data devices. For example, network link 620 may provide a connection through local network 622 to a host computer 624 or to data equipment operated by an Internet Service Provider (ISP) 626. ISP 626 in turn provides data communication services through the world wide packet data communication network now commonly referred to as the “Internet” 628. Local network 622 and Internet 628 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 620 and through communication interface 618, which carry the digital data to and from computer system 600, are example forms of transmission media.

[0070] Computer system 600 can send messages and receive data, including program code, through the network(s), network link 620 and communication interface 618. In the Internet example, a server 630 might transmit a requested code for an application program through Internet 628, ISP 626, local network 622 and communication interface 618.

[0071] The received code may be executed by processor 604 as it is received, and/or stored in storage device 610, or other non-volatile storage for later execution.

5.0. EXTENSIONS AND ALTERNATIVES

[0072] In the foregoing specification, embodiments of the invention have been described with reference to numerous specific details that may vary from implementation to implementation. Thus, the sole and exclusive indicator of what is the invention, and is intended by the applicants to be the invention, is the set of claims that issue from this application, in the specific form in which such claims issue, including any subsequent correction. Any definitions expressly set forth herein for terms contained in such claims shall govern the meaning of such terms as used in the claims. Hence, no limitation, element, property, feature, advantage or attribute that is not expressly recited in a claim should limit the scope of such claim in any way. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

6.0. CONCLUSION

[0073] The above described techniques provide an application user with more efficient access to save functionality. Because a save operation may be triggered without switching to a different window, the above-described techniques avoid the overhead associated with the modal windows used in traditional save operations. Nor do the techniques require a user to memorize a correct combination of keystrokes to trigger a save operation. Furthermore, by placing a

filename entry box directly within the main application window, the above described techniques allow a user to rapidly save a document under different filenames, thereby allowing a user to create different versions of a document with relative ease.

[0074] The above-described techniques also greatly simplify the process of controlling access to documents. By providing a user with a modal-less access control interface, an application allows a user rapid access to access control features, without the confusion and inefficiency of navigating through menu items and modal dialogs.

CLAIMS

What is claimed is:

1. A method comprising:
presenting to a user a window comprising at least:
 - a) one or more controls for editing an object; and
 - b) a save interface;receiving, via the one or more controls, one or more commands to add to or modify data inside the object;
receiving, via the save interface, input that indicates a location at which to save the object;
in response to the input that specifies a location at which to save the object, storing the object at the location;
wherein the method is performed by one or more computing devices.
2. The method of Claim 1, wherein the save interface includes an editable text box, and wherein the input that specifies a location at which to save the object comprises textual input into the editable text box.
3. The method of Claim 2, wherein the input that specifies a location at which to save the object is terminated upon one of: receipt of an Enter keystroke; a mouse click to a save button in the save interface; or a change in focus away from the editable text box.
4. The method of Claim 1, further comprising, in response to modifying data inside of the object, prior to receiving the input that specifies the location, changing the appearance of the save interface to indicate to the user that the object contains unsaved modifications.
5. The method of Claim 4, wherein changing the appearance of the save interface comprises rendering a glow effect around an editable text box in the save interface.
6. The method of Claim 4, wherein changing the appearance of the save interface comprises oscillating between two different appearances for the save interface.
7. A method comprising:
receiving, in a save interface, first input specifying a first portion of a name;
subsequent to the first input, receiving second input specifying one or more changes to an object;

subsequent to the second input, receiving, in the save interface, third input specifying a second portion of the name;

wherein the name comprises the second portion added to the first portion;

subsequent to the third input, receiving fourth input indicating that the name has been fully entered;

in response to the fourth input, saving the object at a location based upon the name;

wherein the method is performed by one or more computing devices.

8. The method of Claim 7, wherein the second input is received in an application window, and wherein the save interface is a modal-less interface inside the application window.

9. A method comprising:

presenting to a user a window comprising at least:

- a) one or more controls for editing an object; and
- b) an access control interface;

receiving, via the one or more controls, one or more commands to add to or modify data inside the object;

receiving, via the access control interface, input that specifies one or more access control levels for the object;

in response to the input that specifies one or more access control levels for the object, changing permission metadata for the object;

wherein the method is performed by one or more computing devices.

10. The method of Claim 9, wherein the access control interface is one of: a pull-down box listing two or more access control levels, a set of one or more links specifying access control levels, or a set of one or more checkboxes specifying access control levels.

11. One or more storage media storing instructions which, when executed by one or more processors, cause:

presenting to a user a window comprising at least:

- a) one or more controls for editing an object; and
- b) a save interface;

receiving, via the one or more controls, one or more commands to add to or modify data inside the object;

receiving, via the save interface, input that indicates a location at which to save the object;

in response to the input that specifies a location at which to save the object, storing the object at the location.

12. The one or more storage media of Claim 11, wherein the save interface includes an editable text box, and wherein the input that specifies a location at which to save the object comprises textual input into the editable text box.
13. The one or more storage media of Claim 12, wherein the input that specifies a location at which to save the object is terminated upon one of: receipt of an Enter keystroke; a mouse click to a save button in the save interface; or a change in focus away from the editable text box.
14. The one or more storage media of Claim 11, wherein the instructions, when executed by the one or more processors, further cause, in response to modifying data inside of the object, prior to receiving the input that specifies the location, changing the appearance of the save interface to indicate to the user that the object contains unsaved modifications.
15. The one or more storage media of Claim 14, wherein changing the appearance of the save interface comprises rendering a glow effect around an editable text box in the save interface.
16. The one or more storage media of Claim 14, wherein changing the appearance of the save interface comprises oscillating between two different appearances for the save interface.
17. One or more storage media storing instructions which, when executed by one or more processors, cause:
 - receiving, in a save interface, first input specifying a first portion of a name;
 - subsequent to the first input, receiving second input specifying one or more changes to an object;
 - subsequent to the second input, receiving, in the save interface, third input specifying a second portion of the name;
 - wherein the name comprises the second portion added to the first portion;
 - subsequent to the third input, receiving fourth input indicating that the name has been fully entered;
 - in response to the fourth input, saving the object at a location based upon the name.
18. The one or more storage media of Claim 17, wherein the second input is received in an application window, and wherein the save interface is a modal-less interface inside the application window.

19. One or more storage media storing instructions which, when executed by one or more processors, cause:

presenting to a user a window comprising at least:

- a) one or more controls for editing an object; and
- b) an access control interface;

receiving, via the one or more controls, one or more commands to add to or modify data inside the object;

receiving, via the access control interface, input that specifies one or more access control levels for the object;

in response to the input that specifies one or more access control levels for the object, changing permission metadata for the object.

20. The one or more storage media of Claim 19, wherein the access control interface is one of: a pull-down box listing two or more access control levels, a set of one or more links specifying access control levels, or a set of one or more checkboxes specifying access control levels.

FIG. 1

100

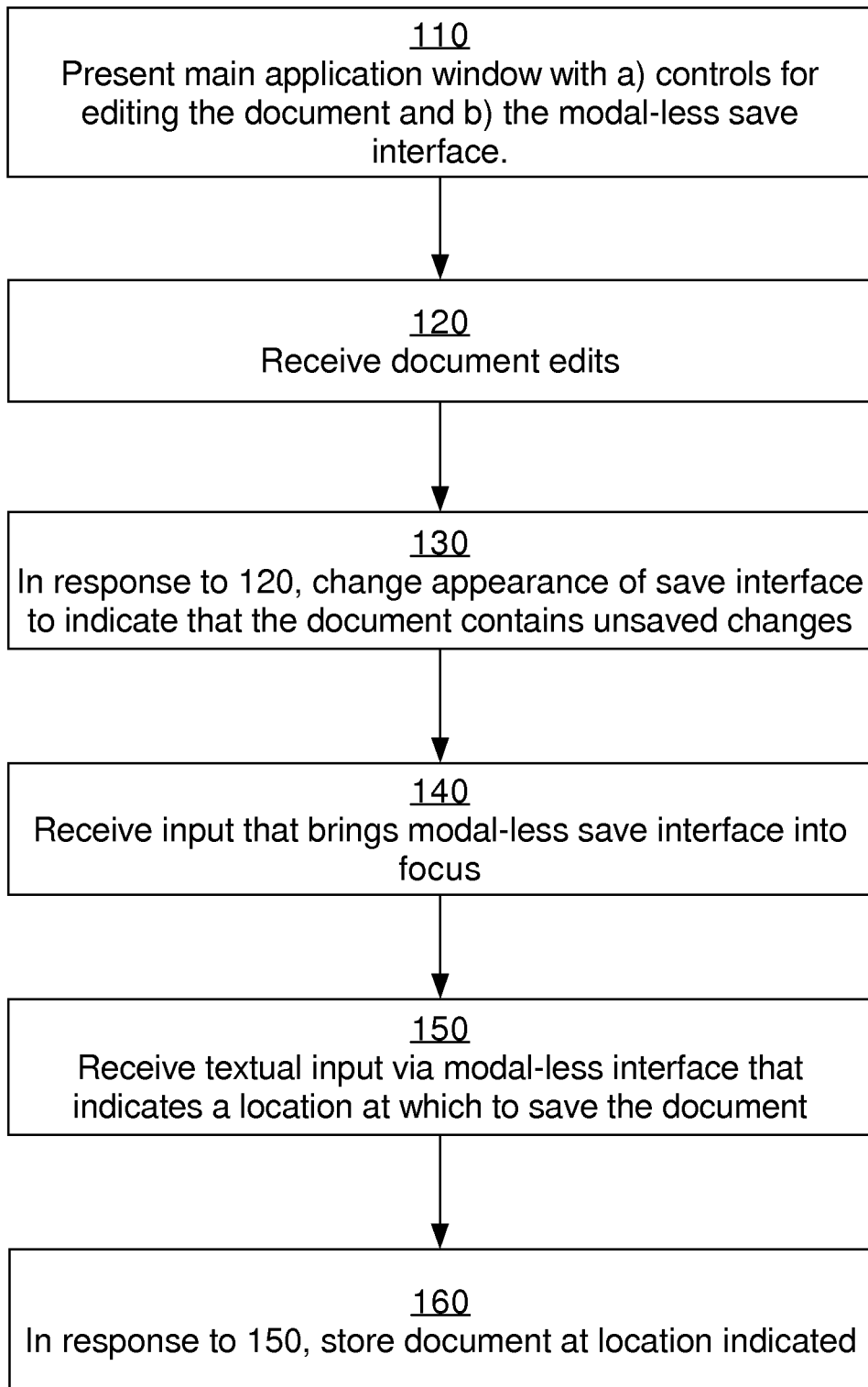


FIG. 2

200

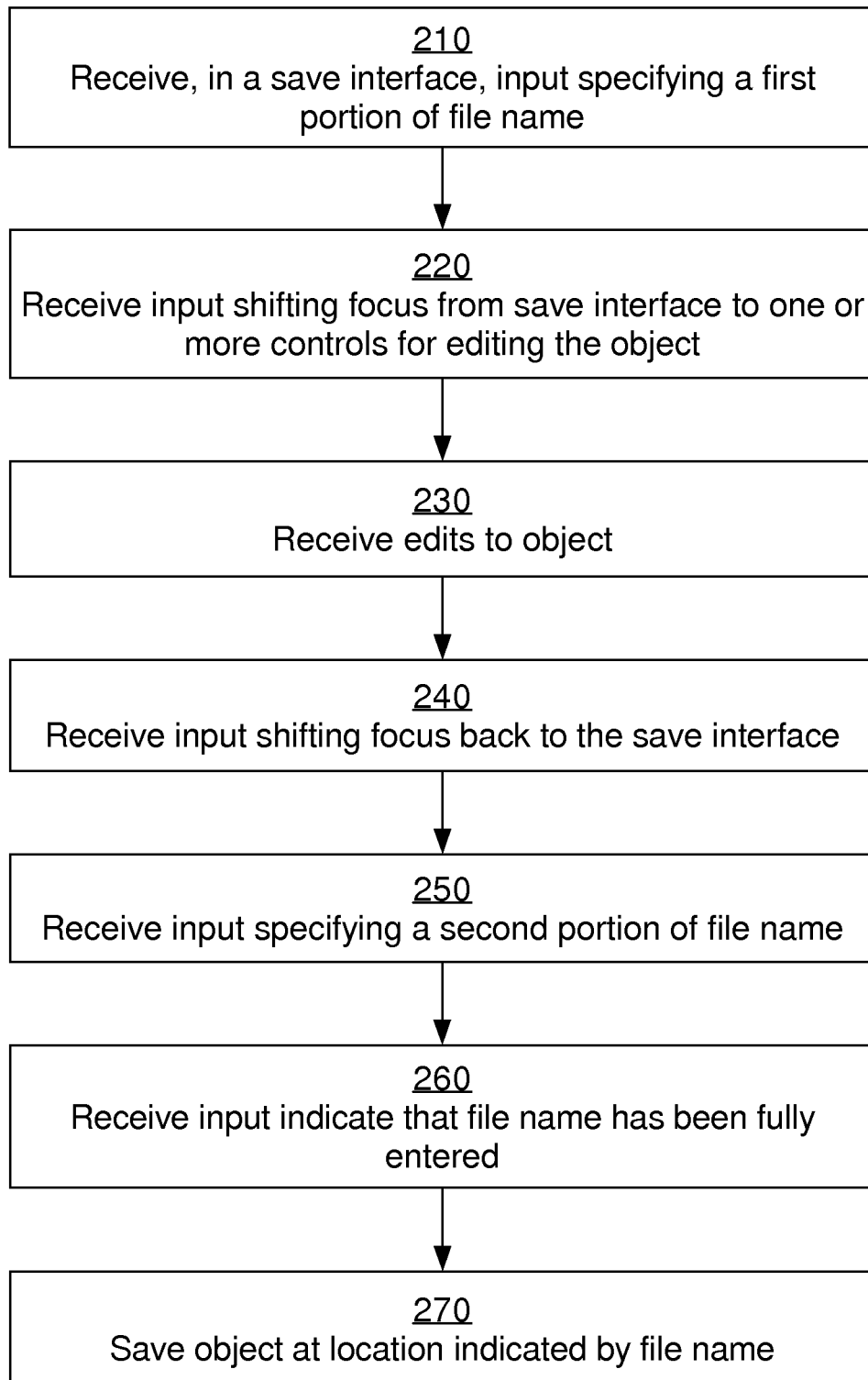
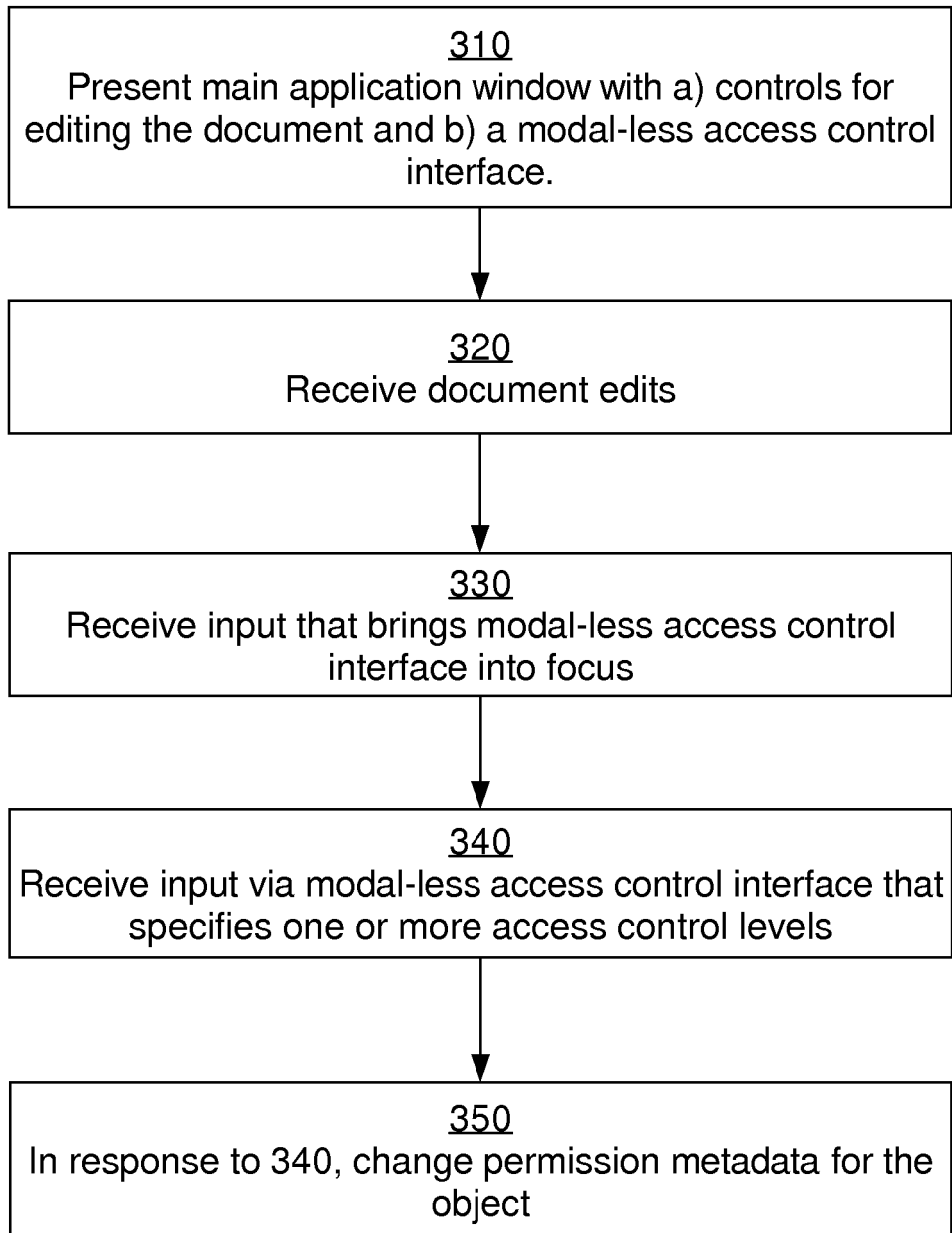
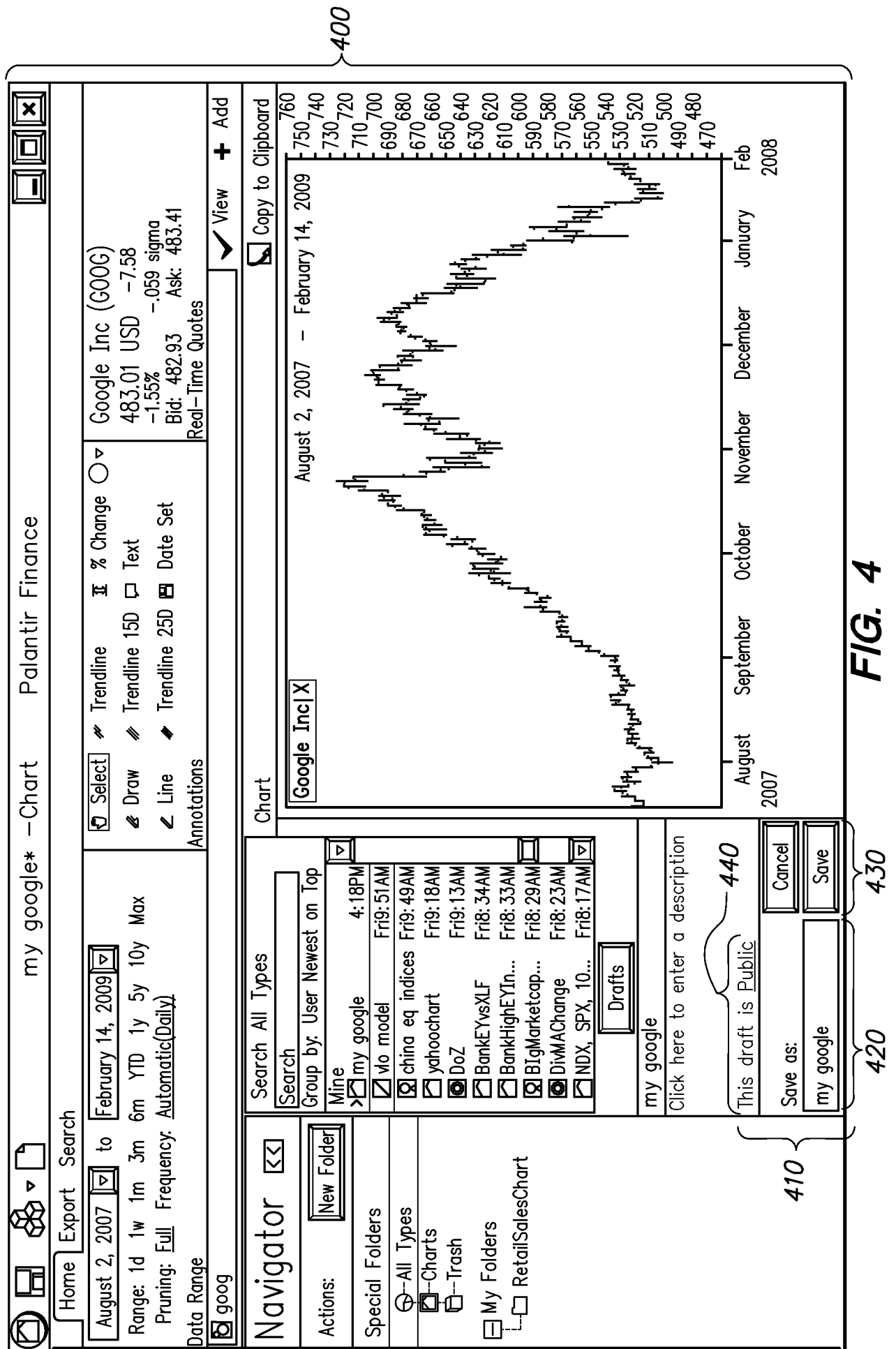


FIG. 3

300





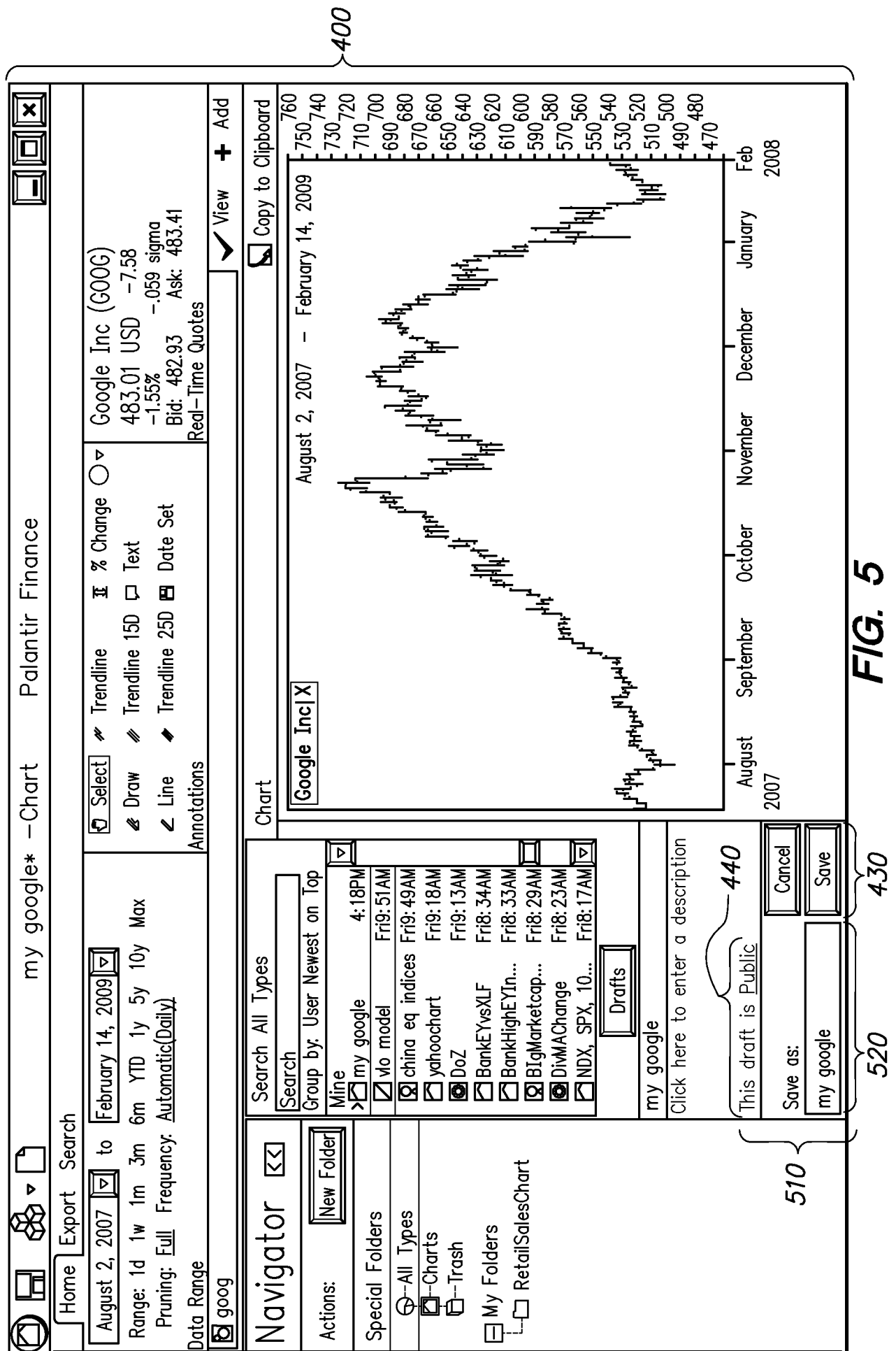


FIG. 6

