



(19) **United States**

(12) **Patent Application Publication**  
**Kong et al.**

(10) **Pub. No.: US 2013/0326403 A1**

(43) **Pub. Date: Dec. 5, 2013**

(54) **METHOD OF SETTING ALTERNATE STYLE ASSIGNMENTS TO MENU ELEMENTS OF AN APPLICATION**

**Publication Classification**

(76) Inventors: **Sarah Kong**, Cupertino, CA (US); **Scott Byer**, Cupertino, CA (US)

(51) **Int. Cl.**  
**G06F 3/048** (2006.01)  
(52) **U.S. Cl.**  
USPC ..... **715/810**

(21) Appl. No.: **12/780,027**

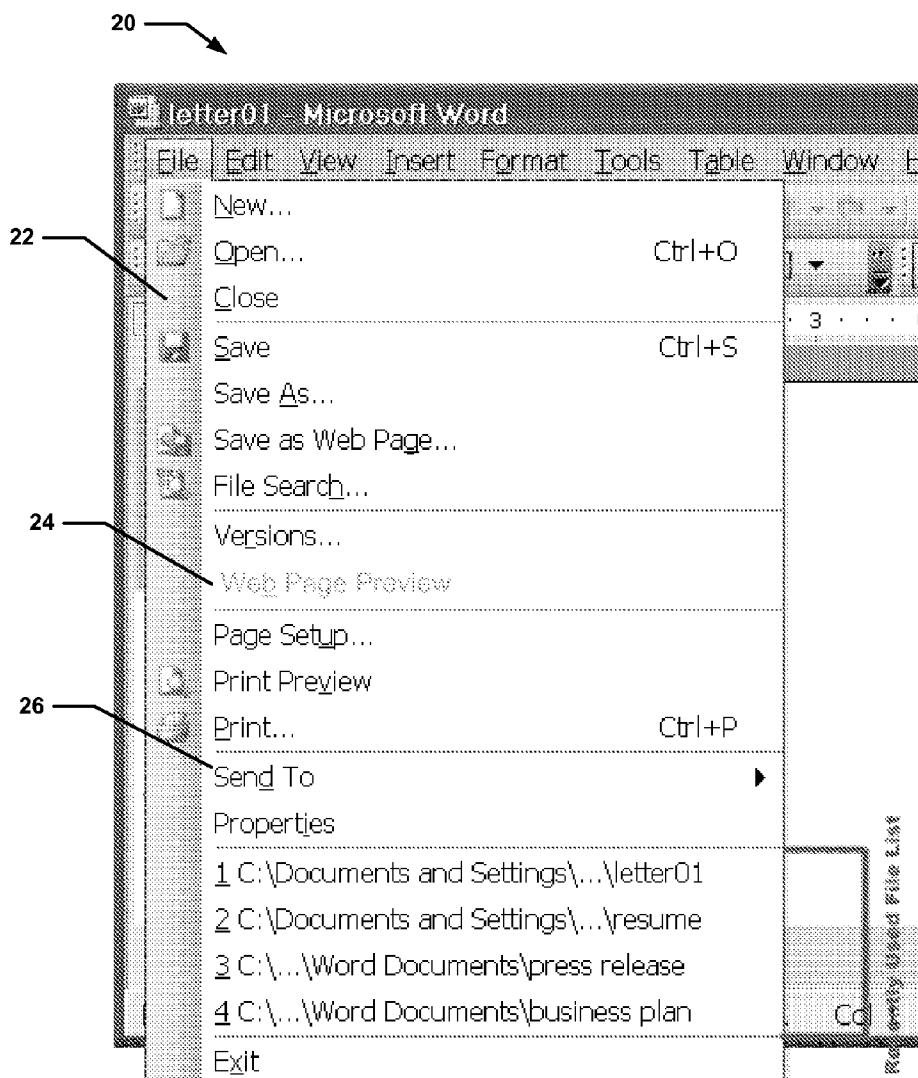
(57) **ABSTRACT**

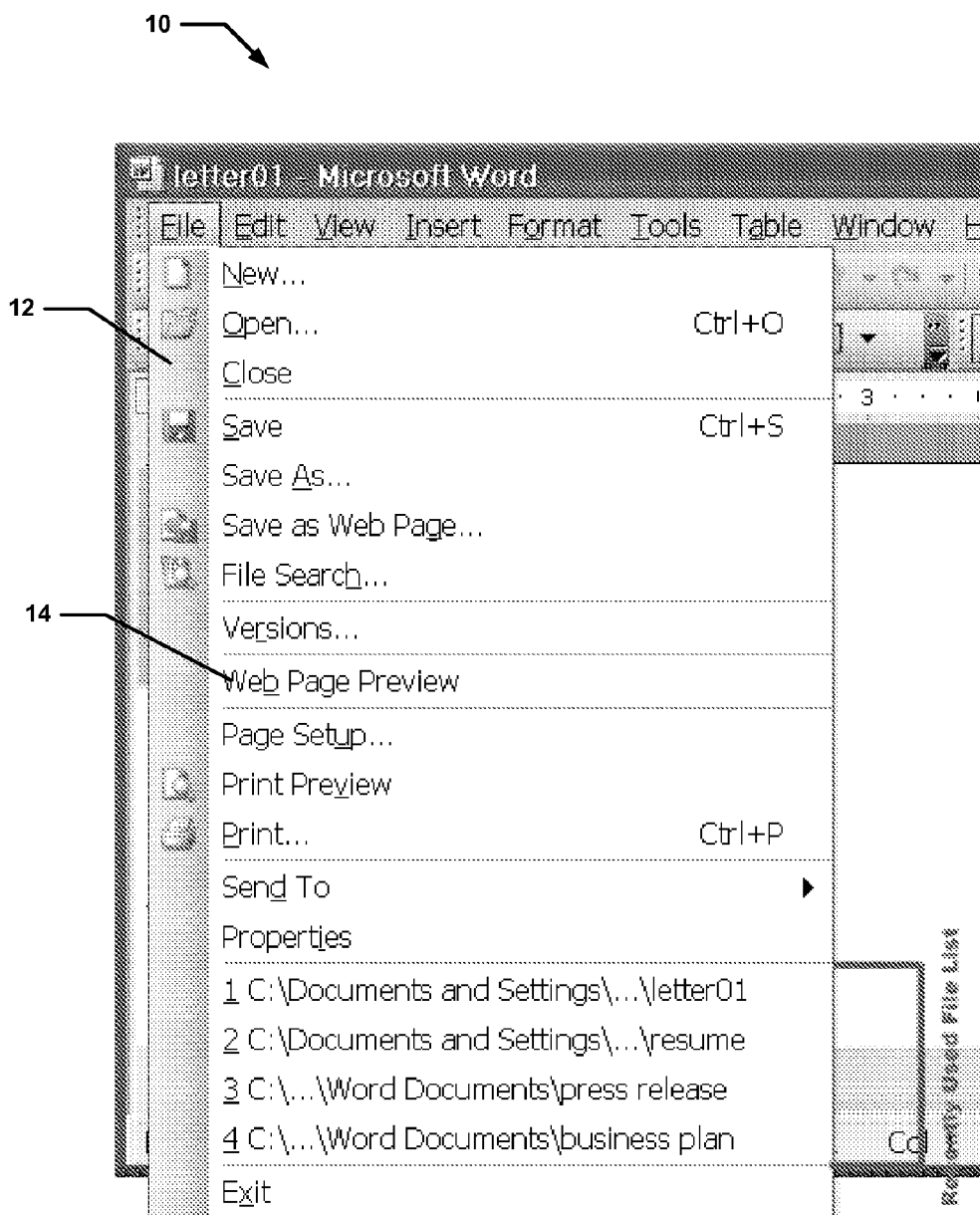
(22) Filed: **May 14, 2010**

A method, apparatus and computer program product for setting alternate style assignments to menu elements of an application are presented. At least one menu element of an application is selected. An alternate style is assigned to the at least one menu element. The at least one menu elements of the application is displayed, wherein the at least one menu element is presented in the alternate style assigned to the element.

**Related U.S. Application Data**

(63) Continuation of application No. 11/238,668, filed on Sep. 29, 2005, now Pat. No. 7,739,620.





PRIOR ART

FIGURE 1

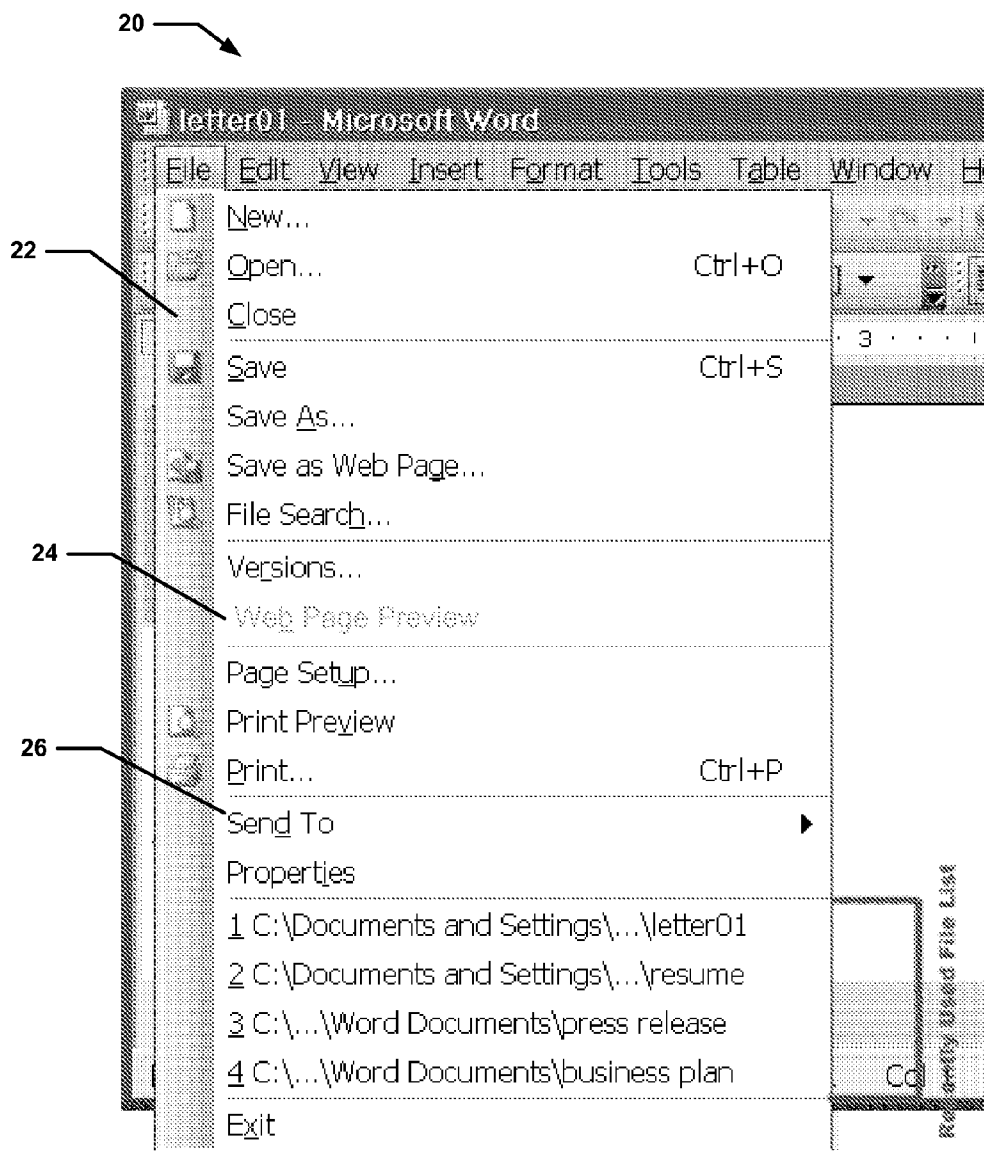
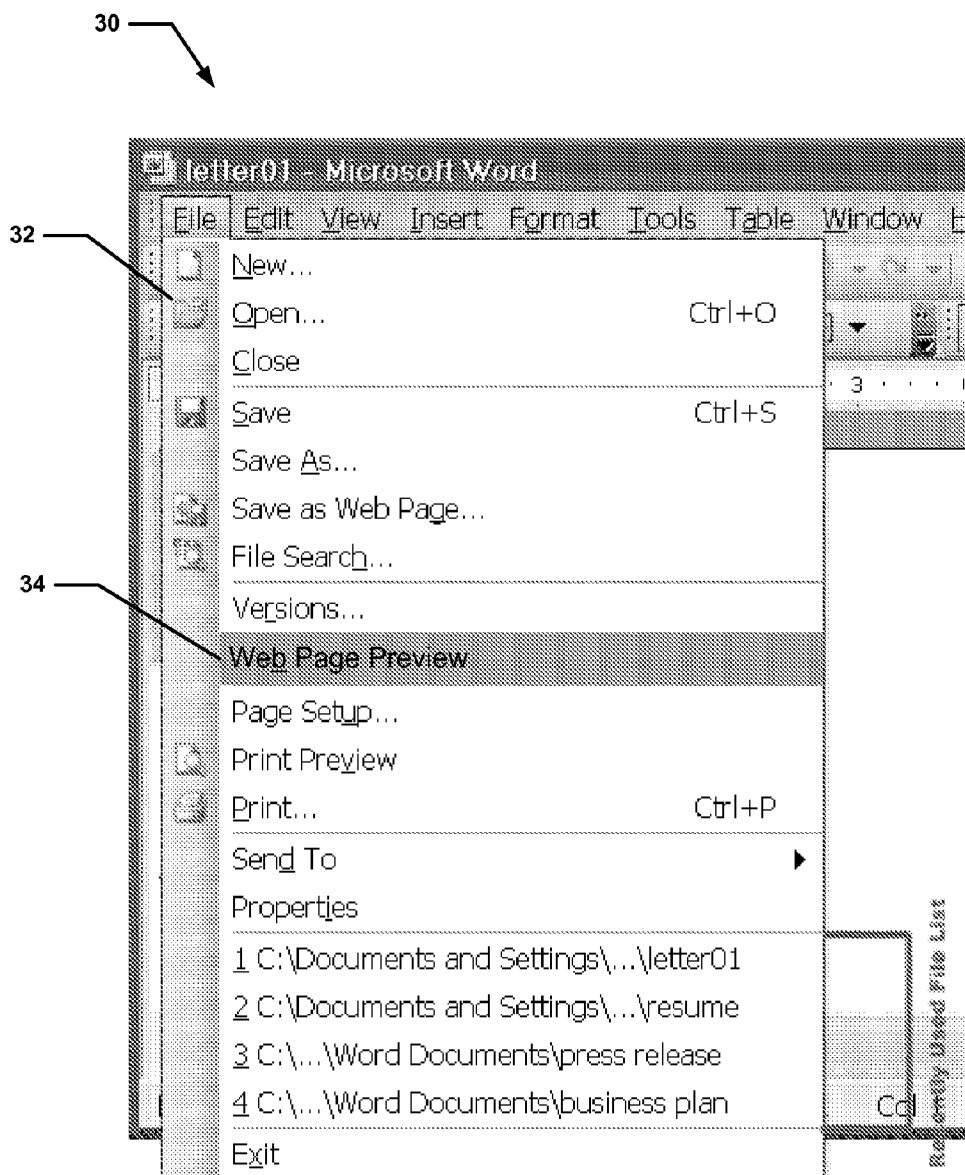
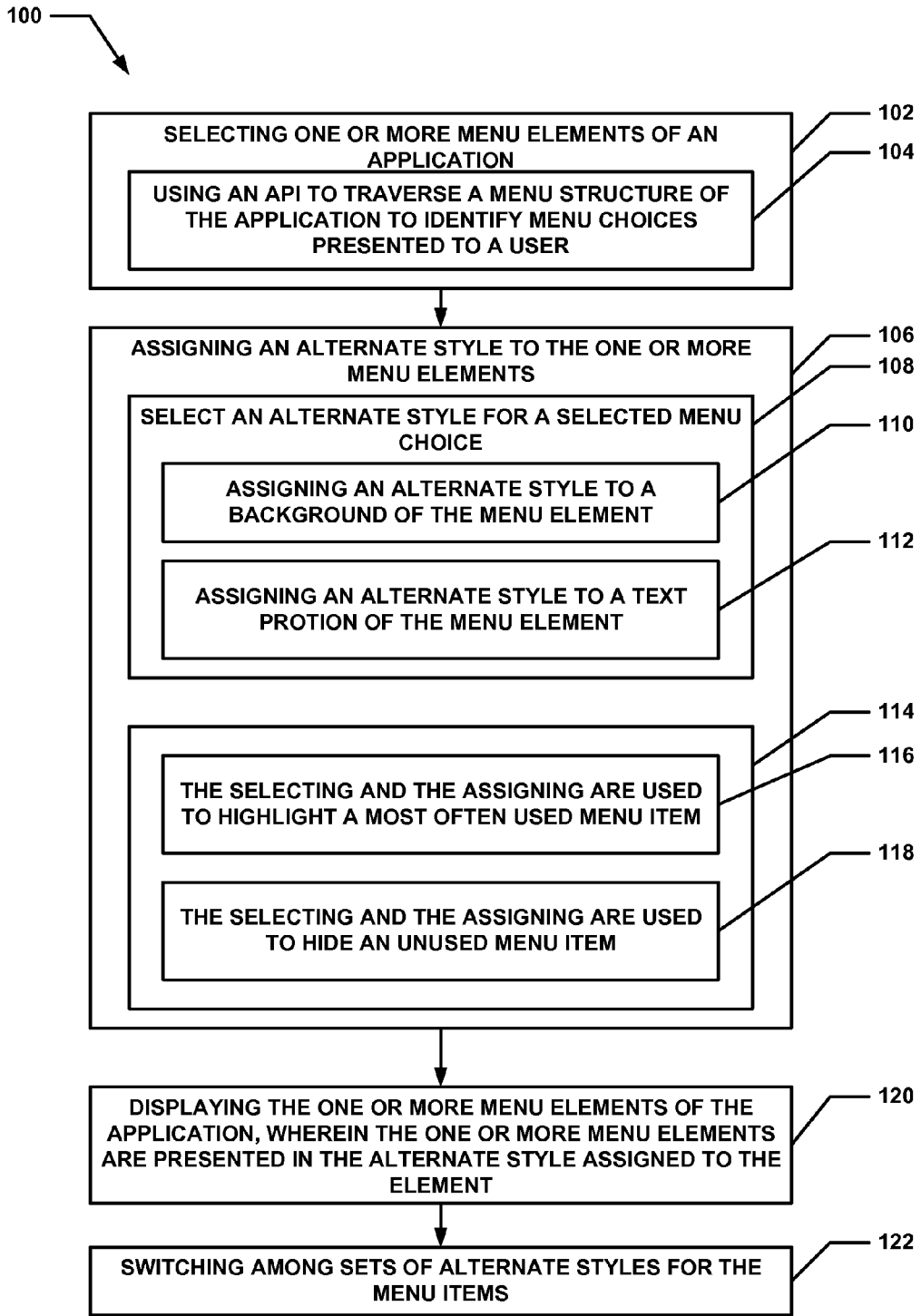


FIGURE 2



**FIGURE 3**



**FIGURE 4**

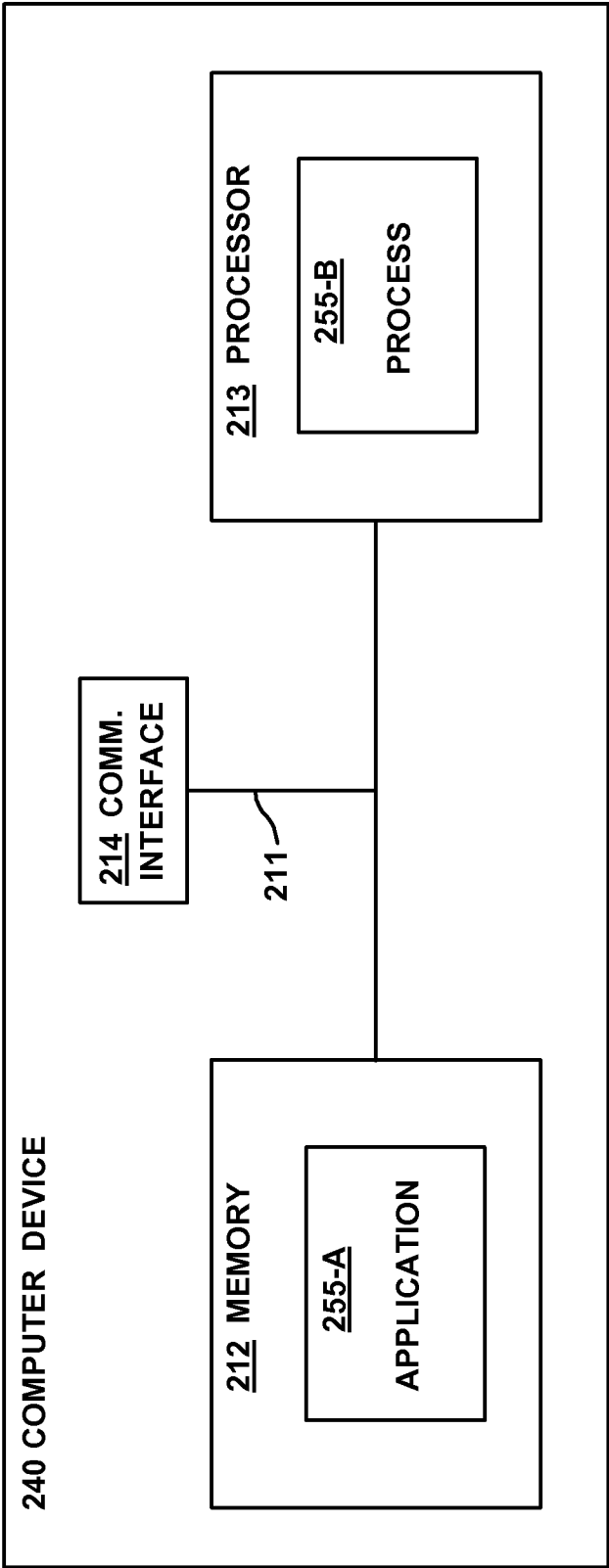


Figure 5

## METHOD OF SETTING ALTERNATE STYLE ASSIGNMENTS TO MENU ELEMENTS OF AN APPLICATION

### RELATED APPLICATIONS

**[0001]** This patent application is a continuation under 35 U.S.C. §120 of U.S. patent application Ser. No. 11/238,668 filed on Sep. 29, 2005, entitled, “A METHOD OF SETTING ALTERNATE STYLE ASSIGNMENTS TO MENU ELEMENTS OF AN APPLICATION,” the contents and teachings of which are hereby incorporated by reference in their entirety.

### BACKGROUND

**[0002]** A Graphical User Interface (GUI) is a program interface that takes advantage of a computer’s graphics capabilities to make an application easier to use. GUIs can free a user from learning complex command languages. A typical GUI includes a pointer controlled by a pointing device, icons, a desktop area, windows and menus.

**[0003]** A pointer is a symbol that appears on the display screen and that a user moves to select objects and commands. Usually, the pointer appears as a small angled arrow, although different applications may incorporate different pointers. For example, many text-processing applications use a pointer that is shaped like a capital I. The pointer is controlled by way of a pointing device such as a mouse or trackball, which enables the user to select and manipulate objects on the display screen.

**[0004]** Icons are small pictures that represent commands, files, or windows. By moving the pointer to the icon and pressing a mouse button, the user can execute a command or convert the icon into a window. The user can also move the icons around the display screen. The area on the display screen where icons are grouped is often referred to as the desktop. The icons are intended to represent real objects on a real desktop.

**[0005]** A user can divide the display screen into different areas called windows. In each window, a user can run a different program or display a different file. Typically a user can move windows around the display screen, and change their shape and size at will.

**[0006]** Most applications now have a menu-driven component. A menu comprises a graphical list of operations (menu elements) a user can perform. Most graphical user interfaces let a user execute commands by selecting a choice from a menu. A user can choose an item from the menu by highlighting it and then pressing the Enter or Return key, or by simply pointing to the item with a mouse and clicking one of the mouse buttons.

**[0007]** In addition to their visual components, GUIs also make it easier to move data from one application to another. A GUI includes standard formats for representing text and graphics. Because the formats are well-defined, different programs that run under a common GUI can share data. This makes it possible, for example, to copy a graph created by a spreadsheet program into a document created by a word processor.

**[0008]** There are several different types of menus. One type is known as a pop-up menu. A pop-up menu appears temporarily when a mouse button is clicked on a selection. Once you make a selection from a pop-up menu, the menu usually disappears. A pull-down menu is a special type of pop-up

menu that appears directly beneath the command selected by the user. A cascading menu is a submenu that opens when a user selects a choice from another menu.

**[0009]** Another type of menu is known as a moving-bar menu. In a moving bar menu, options are highlighted by a bar that a user can move from one item to another. Most menus are moving-bar menus. A menu bar is a menu arranged horizontally. Each menu option is generally associated with another pull-down menu that appears when a user makes a selection. A tear-off menu is a pop-up menu that a user can move around the screen like a window.

### SUMMARY

**[0010]** Menus, when used as part of a GUI, have proven to be very useful and are ubiquitous. Conventional mechanisms for menus such as those explained above suffer from a variety of deficiencies. One such deficiency is that conventional menus provide a series of menu elements having a similar look. The menu elements typically have the same color and the same background. Thus, the menus do not provide additional help or information to a user.

**[0011]** Embodiments of the invention significantly overcome such deficiencies and provide mechanisms and techniques that provide a method of setting alternate style assignments for menu elements of an application.

**[0012]** In a particular embodiment of a method for setting alternate style assignments to menu elements of an application, the method includes selecting at least one menu element of an application. The method further includes assigning an alternate style to the at least one menu element. The method additionally includes displaying the at least one menu element of the application, wherein the at least one menu element is presented in the alternate style assigned to the element.

**[0013]** Other embodiments include a computer readable medium having computer readable code thereon for setting alternate style assignments to menu elements of an application. The medium includes instructions for selecting at least one menu element of an application. The medium also includes instructions for assigning an alternate style to the at least one menu element. The medium further includes instructions for displaying the at least one menu element of the application, wherein the at least one menu element is presented in the alternate style assigned to the element.

**[0014]** Still other embodiments include a computerized device, configured to process all the method operations disclosed herein as embodiments of the invention. In such embodiments, the computerized device includes a memory system, a processor, communications interface in an interconnection mechanism connecting these components. The memory system is encoded with a process that provides setting alternate style assignments to menu elements of an application as explained herein that when performed (e.g. when executing) on the processor, operates as explained herein within the computerized device to perform all of the method embodiments and operations explained herein as embodiments of the invention. Thus any computerized device that performs or is programmed to perform up processing explained herein is an embodiment of the invention.

**[0015]** Other arrangements of embodiments of the invention that are disclosed herein include software programs to perform the method embodiment steps and operations summarized above and disclosed in detail below. More particularly, a computer program product is one embodiment that has

a computer-readable medium including computer program logic encoded thereon that when performed in a computerized device provides associated operations setting alternate style assignments to menu elements of an application as explained herein. The computer program logic, when executed on at least one processor with a computing system, causes the processor to perform the operations (e.g., the methods) indicated herein as embodiments of the invention. Such arrangements of the invention are typically provided as software, code and/or other data structures arranged or encoded on a computer readable medium such as an optical medium (e.g., CD-ROM), floppy or hard disk or other a medium such as firmware or microcode in one or more ROM or RAM or PROM chips or as an Application Specific Integrated Circuit (ASIC) or as downloadable software images in one or more modules, shared libraries, etc. The software or firmware or other such configurations can be installed onto a computerized device to cause one or more processors in the computerized device to perform the techniques explained herein as embodiments of the invention. Software processes that operate in a collection of computerized devices, such as in a group of data communications devices or other entities can also provide the system of the invention. The system of the invention can be distributed between many software processes on several data communications devices, or all processes could run on a small set of dedicated computers, or on one computer alone.

**[0016]** It is to be understood that the embodiments of the invention can be embodied strictly as a software program, as software and hardware, or as hardware and/or circuitry alone, such as within a computer device.

#### BRIEF DESCRIPTION OF THE DRAWINGS

**[0017]** The foregoing and other objects, features and advantages of the invention will be apparent from the following more particular description of preferred embodiments of the invention, as illustrated in the accompanying drawings in which like reference characters refer to the same parts throughout the different views. The drawings are not necessarily to scale, emphasis instead being placed upon illustrating the principles of the invention.

**[0018]** FIG. 1 comprises a prior art screen shot of an example pull-down menu showing a list of menu elements;

**[0019]** FIG. 2 comprises a screen shot of an example pull-down menu showing a list of menu elements wherein a selected menu element is provided having an alternate style (text) in accordance with embodiments of the present invention;

**[0020]** FIG. 3 comprises a screen shot of an example pull-down menu showing a list of menu elements wherein a selected menu element is provided having an alternate style (background) in accordance with embodiments of the present invention;

**[0021]** FIG. 4 illustrates a flow diagram of a process setting alternate style assignments to menu elements of an application in accordance with embodiments of the invention; and

**[0022]** FIG. 5 illustrates an example computer system architecture for a computer system that provides the setting of alternate style assignments to menu elements of an application in accordance with embodiments of the invention.

#### DETAILED DESCRIPTION

**[0023]** Referring to FIG. 1, a GUI 10 for a particular application is shown. Here the GUI is for a Microsoft Word appli-

cation, however it should be understood that the concepts explained herein apply to any GUI and any application. GUI 10 includes a menu 12. Here, menu 12 has been provided in response to user selecting the “File” element of the application toolbar. It should be understood that the concepts explained herein apply to any menu element and to any menu. The resulting menu 12 provides the user with different options relating to the file, presented as a series of menu elements. A particular menu element 14 shows that one of the options the user has is to perform a “Web Page Preview” for the document. The menu element 14 includes a text portion 16 and a background portion 18. All the menu elements have a similar look in that the text for each menu element is the same color and size, and the backgrounds for each of the menu elements is the same color.

**[0024]** Referring now to FIG. 2, a menu 22 of GUI 20 is shown. In this menu, a menu item is presented in a manner that makes the menu element different from the other menu elements of menu 22. Here the menu element 24 is provided wherein the text portion is in an alternate style than the text portions of other menu elements. The background color has not changed. This changing of the menu element text is accomplished by the present invention. The basic configuration allows a dialog to allow a user to define what alternate style menu choices are to be set to. Alternately, the system may apply the different alternate styles for menu elements in response to certain user inputs (e.g. selecting help for a particular function, as will be described in detail below). The application can apply menu alternate styles to any application, since it can use a windows API to traverse the menu structure of that application to identify what menu choices the application shows to the user. The dialog then allows the user to custom select one or more of those choices to allow the user to set an alternate style for that choice. Thereafter, when the application displays that menu, the dialog can instruct windows to render that choice as per the user setting. In such a manner the menu element is presented such that the menu element is different than other menu elements of the same menu.

**[0025]** Referring now to FIG. 3, a menu 32 of GUI 30 is shown. In this menu, a menu item is presented in a manner that makes the menu element different from the other menu elements of menu 32. Here the menu element 34 is provided wherein the background portion is in an alternate style as compared to the background portions of other menu elements. The text color has not changed. This changing of the menu element background is accomplished by the present invention. The basic configuration allows a dialog to allow a user to define what alternate style menu choices are to be set to. The application can apply menu alternate styles to any application, since it can use a windows API to traverse the menu structure of that application to identify what menu choices the application shows to the user. The dialog then allows the user to custom select one or more of those choices to allow the user to set an alternate style for that choice. Thereafter, when the application displays that menu, the dialog can instruct the operating system to render that choice as per the user setting. In such a manner the menu element is presented such that the menu element is different than other menu elements of the same menu.

**[0026]** In one configuration, the menu items can be color coded to show differences in software versions. The alternate style may be realized as different font types, different colors, or the like. Thus, if a new release of a product comes out, the



new menu choices can be color coded to correspond to the new version of software (e.g., all new features in yellow background have been added in new version). Also, each version of the software could have alternate style choices, so that version one items are all a first color (e.g. yellow), version two additions are all another color (e.g. blue), version three items are another color (e.g. red) and so forth.

**[0027]** Another use of alternate style menu items is to provide a tutorial for users in the form of following alternate style choices in drill down menus. Thus, if a user chooses help on a topic, the help might direct the user to the first top level menu and direct the user to follow all menu choices in yellow to get to where the user wants to be. For example, if a user wanted help regarding inserting a filename and path in a header in a

**[0028]** Microsoft Word document, the insert menu would be presented. The menu element Autotext of the Insert menu would be highlighted, directing the user to click that menu element for further help information. Clicking on the highlighted Autotext menu element results in a Autotext menu wherein a Header/Footer menu element is highlighted. Clicking on the Header/Footer menu element results in a presentation of a Header/Footer menu wherein the Filename and Path element is highlighted.

**[0029]** The present invention may further be used to hide commands in a menu that are seldom or never used. The present invention may also be used to highlight (in color) commands in a menu that are used most often. A user further has the ability to switch among sets of edited menus on the fly.

**[0030]** In a particular embodiment, an Application Programming Interface (API) is used to traverse a menu structure of the application to identify menu choices presented to a user. The user, by way of the GUI, can assign a selected alternate style for a selected menu choice. For example, the user can select a text alternate style and/or a background alternate style. Thereafter, when the application displays that menu, that alternate style choice defined by the user is used for that menu element.

**[0031]** A flow chart of the presently disclosed method is depicted in FIG. 4. The rectangular elements are herein denoted "processing blocks" and represent computer software instructions or groups of instructions. Alternatively, the processing blocks represent steps performed by functionally equivalent circuits such as a digital signal processor circuit or an application specific integrated circuit (ASIC). The flow diagrams do not depict the syntax of any particular programming language. Rather, the flow diagrams illustrate the functional information one of ordinary skill in the art requires to fabricate circuits or to generate computer software to perform the processing required in accordance with the present invention. It should be noted that many routine program elements, such as initialization of loops and variables and the use of temporary variables are not shown. It will be appreciated by those of ordinary skill in the art that unless otherwise indicated herein, the particular sequence of steps described is illustrative only and can be varied without departing from the spirit of the invention. Thus, unless otherwise stated the steps described below are unordered meaning that, when possible, the steps can be performed in any convenient or desirable order.

**[0032]** Referring now to FIG. 4, a particular embodiment of a method 100 of setting alternate style assignments to menu elements of an application is shown. The method starts with processing block 102 wherein at least one menu element of an

application is selected. The user thus determines that a particular menu element of a menu of an application should be changed, and selects the menu element. An example of this is Microsoft Word application, File menu wherein the Web Page Preview menu element 24 has been selected as shown in FIG. 2. In processing block 104 an API is used to traverse a menu structure of the application to identify menu choices presented to the user.

**[0033]** As recited in processing block 106 and 108 an alternate style is assigned to at least one menu element. This may include assigning an alternate style to a background portion of a menu element as recited in processing block 110 and/or can include assigning an alternate style to a text portion of a menu element, as recited in processing block 112. The selecting and assigning of alternate styles to menu elements can be done to highlight a most often used menu item, can be used to hide an unused menu element, show differences in software versions and provide a tutorial by directing a user to follow alternate style choices in at least one drilldown menu.

**[0034]** In processing block 114, the at least one menu element of the application is displayed, wherein the at least one menu element is presented in the alternate style assigned to the element. In processing block 116, the user can switch among sets of alternate styles for the menu items.

**[0035]** FIG. 5 illustrates example architectures of a computer system that is configured as a host computer system 240. The computer system 240 may be any type of computerized system such as a personal computer, workstation, portable computing device, mainframe, server or the like. In this example, the system includes an interconnection mechanism 211 that couples a memory system 212, a processor 213, and a communications interface 214. The communications interface 214 allows the computer system 240 to communicate with external devices or systems.

**[0036]** The memory system 212 may be any type of computer readable medium that is encoded with an application 255-A that represents software code such as data and/or logic instructions (e.g., stored in the memory or on another computer readable medium such as a disk) that embody the processing functionality of embodiments of the invention as explained above. The processor 213 can access the memory system 212 via the interconnection mechanism 211 in order to launch, run, execute, interpret or otherwise perform the logic instructions of the applications 255-A for the host in order to produce a corresponding process 255-B. In other words, the process 255-B represents one or more portions of the application 255-A performing within or upon the processor 213 in the computer system.

**[0037]** It is to be understood that embodiments of the invention include the applications (i.e., the un-executed or non-performing logic instructions and/or data) encoded within a computer readable medium such as a floppy disk, hard disk or in an optical medium, or in a memory type system such as in firmware, read only memory (ROM), or, as in this example, as executable code within the memory system 212 (e.g., within random access memory or RAM). It is also to be understood that other embodiments of the invention can provide the applications operating within the processor 213 as the processes. While not shown in this example, those skilled in the art will understand that the computer system may include other processes and/or software and hardware components, such as an operating system, which have been left out of this illustration for ease of description of the invention.

[0038] Having described preferred embodiments of the invention it will now become apparent to those of ordinary skill in the art that other embodiments incorporating these concepts may be used. Additionally, the software included as part of the invention may be embodied in a computer program product that includes a computer useable medium. For example, such a computer usable medium can include a readable memory device, such as a hard drive device, a CD-ROM, a DVD-ROM, or a computer diskette, having computer readable program code segments stored thereon. The computer readable medium can also include a communications link, either optical, wired, or wireless, having program code segments carried thereon as digital or analog signals. Accordingly, it is submitted that that the invention should not be limited to the described embodiments but rather should be limited only by the spirit and scope of the appended claims.

- 1. A computer-implemented method comprising:
  - assigning an alternate style to at least one menu element of a new version of an application, wherein the at least one menu element is for a new feature of the new version of the application, the new feature being included in the new version of the application and is not included in a prior version of the application; and
  - displaying, in the new version of the application executed on a computer system, the at least one menu element of the application, wherein the at least one menu element is presented in the alternate style, wherein the alternative style indicates that the new feature is new in the new version by distinguishing the at least one menu element for the new feature from menu elements for features provided by the prior version
 wherein the new feature is used in a tutorial that provides guidance regarding making a sequence of menu selections, wherein the tutorial provides guidance by applying an alternate style to the at least one menu element to be selected during the sequence of menu selections.
- 2. (canceled)
- 3. (canceled)
- 4. The method of claim 1 wherein assigning the alternate style comprises assigning the alternate style to a background of the at least one menu element.
- 5. The method of claim 1 wherein assigning the alternate style comprises assigning the alternate style to a text portion of the at least one menu element.

- 6-9. (canceled)
- 10. A computer-implemented method comprising:
  - assigning an alternate style to at least one menu element of an application, wherein the at least one menu element is for at least one feature of the application used in a tutorial; and
  - displaying, in the application executed on a computer system, the at least one menu element of the application,

wherein the at least one menu element is presented in the alternate style, wherein the alternative style indicates that the at least one feature is used in the tutorial by distinguishing the at least one menu element from menu elements for features not used in the tutorial,

wherein the tutorial provides guidance regarding making menu selections, wherein the tutorial provides guidance upon the indication of a help request for a particular function in the application.

- 11. (canceled)
- 12. A non-transitory computer readable medium having computer readable code thereon the medium comprising:
  - instructions for assigning an alternate style to at least one menu element of a new version of an application, wherein the at least one menu element is for a new feature of the new version of the application, the new feature being included in the new version of the application and is not included in a prior version of the application; and
  - instructions for displaying, in the new version of the application executed on a computer system, the at least one menu element of the application, wherein the at least one menu element is presented in the alternate style, wherein the alternative style indicates that the new feature is new in the new version by distinguishing the at least one menu element for the new feature from menu elements for features provided by the prior version, wherein the new feature is used in a tutorial that provides guidance regarding making a sequence of menu selections, wherein the tutorial provides guidance by applying an alternate style to the at least one menu element to be selected during the sequence of menu selections.
- 13. (canceled)
- 14. (canceled)
- 15. The non-transitory computer readable medium of claim 12 wherein the instructions for assigning the alternate style comprise instructions for assigning the alternate style to a background of the at least one menu element
- 16. The non-transitory computer readable medium of claim 12 wherein the instructions for assigning the alternate style comprise instructions for assigning the alternate style to a text portion of the at least one menu element.
- 17-20. (canceled)
- 21. The method of claim 1 wherein the new version of the application is a new software product release with respect to the prior version of the application.
- 22. The computer-implemented method of claim 10 wherein the tutorial provides guidance regarding making a sequence of menu selections.

\* \* \* \* \*