US 20070174782A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2007/0174782 A1

(43) Pub. Date: Jul. 26, 2007

Russo

(54) SMART ICON PLACEMENT ACROSS DESKTOP SIZE CHANGES

(75) Inventor: **Thomas P. Russo**, Seattle, WA (US)

Correspondence Address:
BANNER & WITCOFF, LTD.
ATTORNEYS FOR CLIENT NOS. 003797 & 013797
1100 13th STREET, N.W.
SUITE 1200
WASHINGTON, DC 20005-4051 (US)

(73) Assignee: **Microsoft Corporation**, Redmond, WA

(21) Appl. No.: **11/338,851**

(22) Filed: **Jan. 25, 2006**

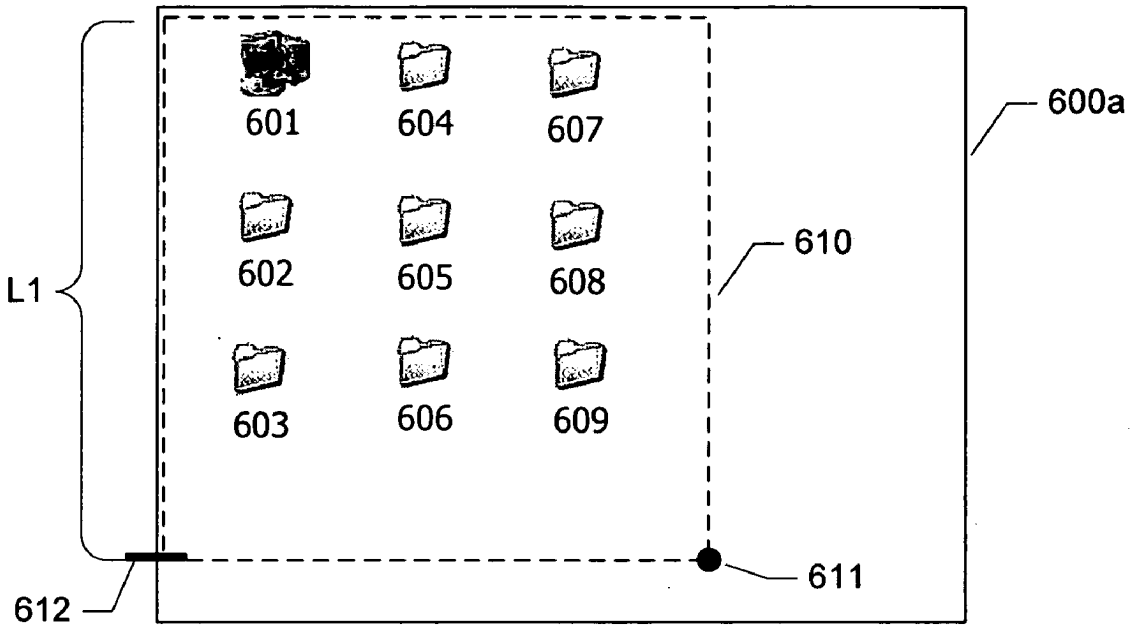
Publication Classification

(51) **Int. Cl.**
G06F 9/00 (2006.01)

(52) **U.S. Cl.** **715/781**; 715/835; 715/810; 715/800; 715/799

(57) **ABSTRACT**

Intelligent rearrangement of icons on a desktop in response to changes in desktop size, such as orientation, may be provided using a number of approaches. Desktop area in multiple orientations may be divided into a grid having equal numbers of rows and columns, and placement of icons in each orientation may use common grid coordinates. Icons may alternatively be reflected along a diagonal of desktops in different orientations, such that some icon positional relationships may be maintained. Placement of icons on a desktop may be restricted to a predefined subset region of the desktop, where the subset region will fit in multiple orientations without requiring adjustment. Additionally, areas of a desktop in multiple orientations may be reserved for predefined types of icons, and changes in orientation may result in placement of icons in corresponding reserved areas based on icon attribute.



SAP AMERICA, INC.
EXHIBIT 1020

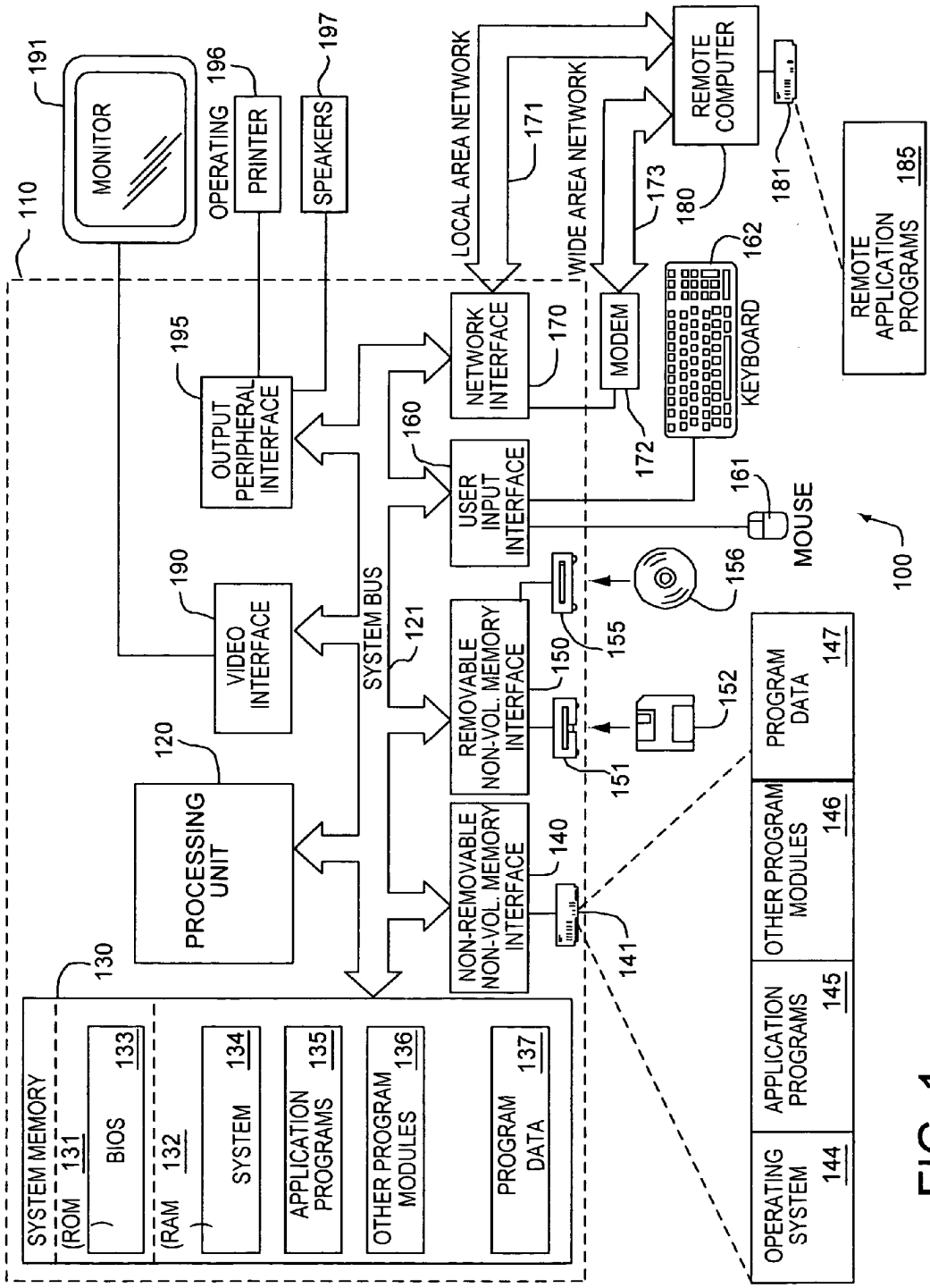

110
MONITOR
191
OPERATING
PRINTER
196
SPEAKERS
197
LOCAL AREA NETWORK
171
WIDE AREA NETWORK
173
REMOTE COMPUTER
180
181
REMOTE APPLICATION PROGRAMS
185
162
KEYBOARD
172
MODEM
NETWORK INTERFACE
170
195
OUTPUT PERIPHERAL INTERFACE
160
USER INPUT INTERFACE
161
MOUSE
156
100
190
VIDEO INTERFACE
SYSTEM BUS
121
REMOVABLE NON-VOL. MEMORY INTERFACE
150
155
152
151
120
PROCESSING UNIT
NON-REMOVABLE NON-VOL. MEMORY INTERFACE
140
141
130
SYSTEM MEMORY
(ROM 131
BIOS
133
(RAM 132
SYSTEM
134
APPLICATION PROGRAMS
135
OTHER PROGRAM MODULES
136
PROGRAM DATA
137
OPERATING SYSTEM
144
APPLICATION PROGRAMS
145
OTHER PROGRAM MODULES
146
PROGRAM DATA
147

FIG. 1

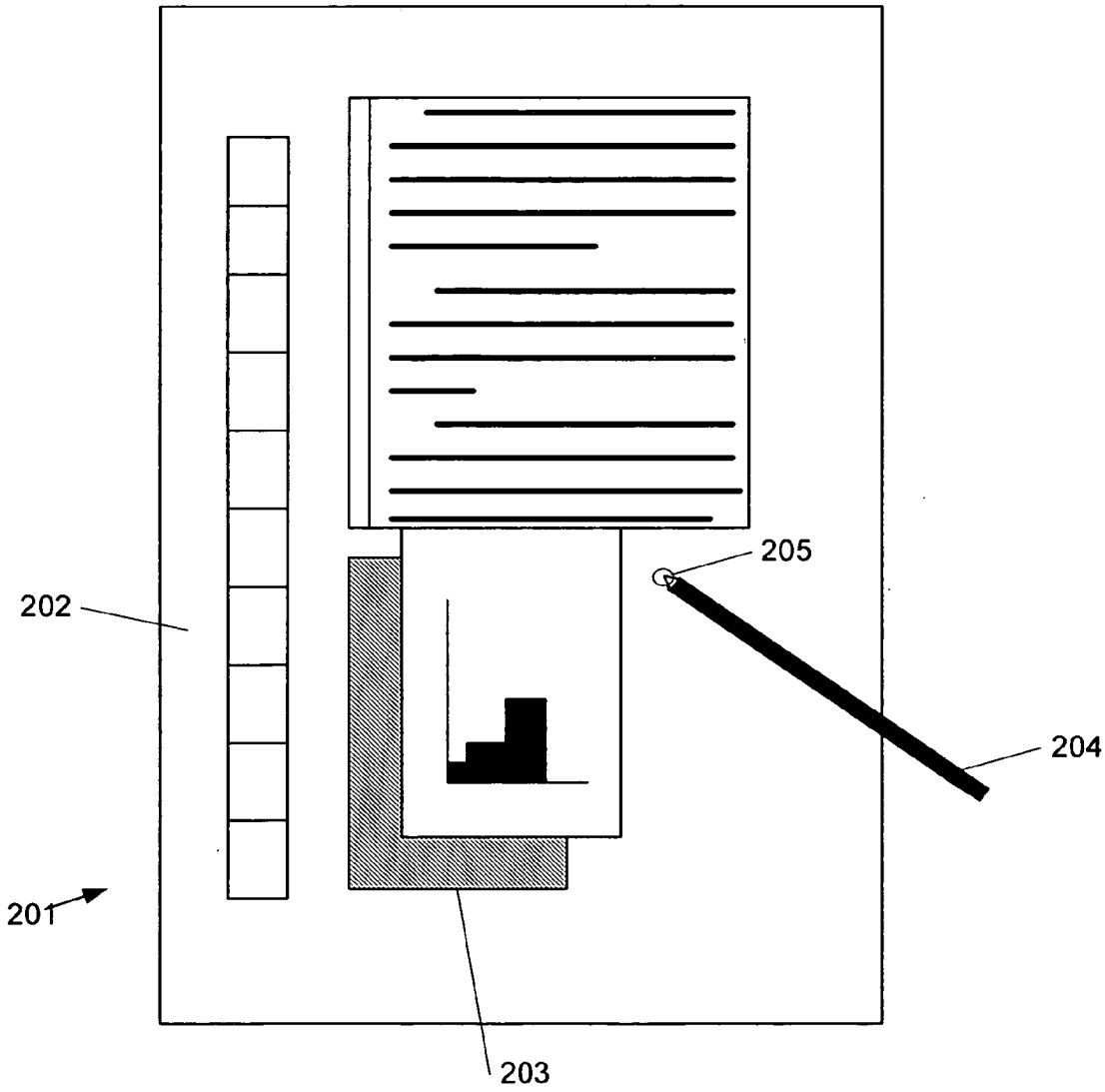


FIG. 2

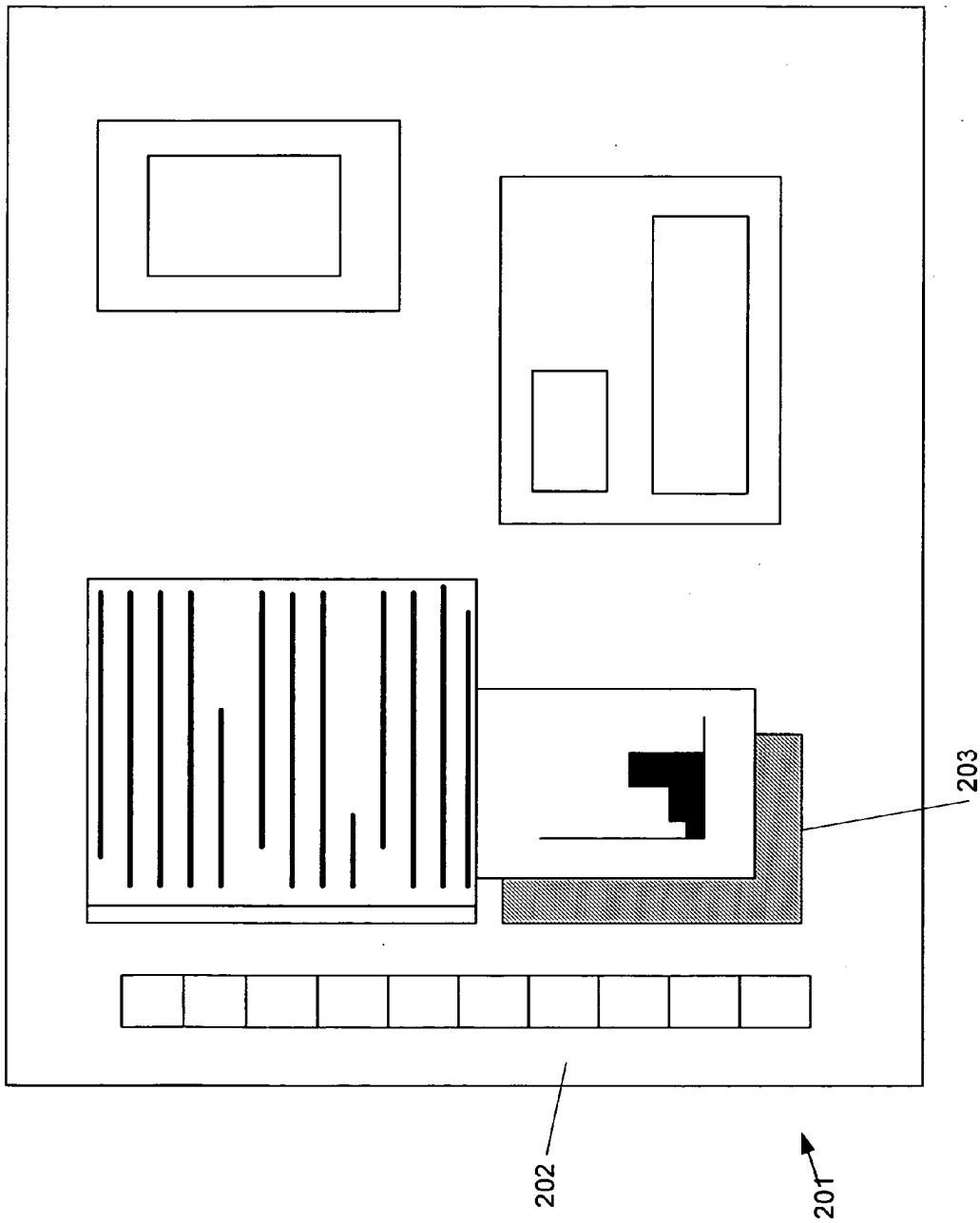


FIG. 3

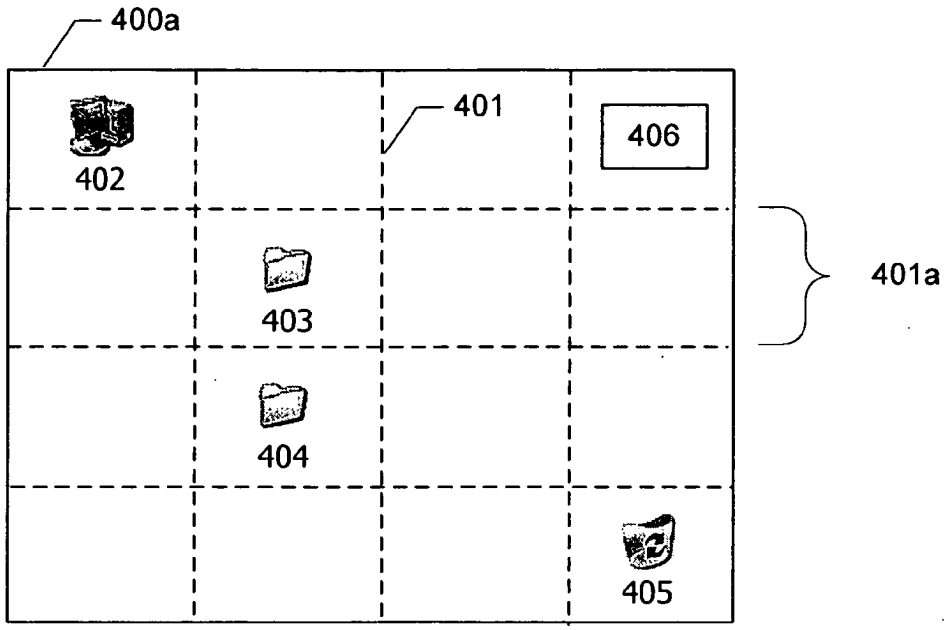


FIG. 4A

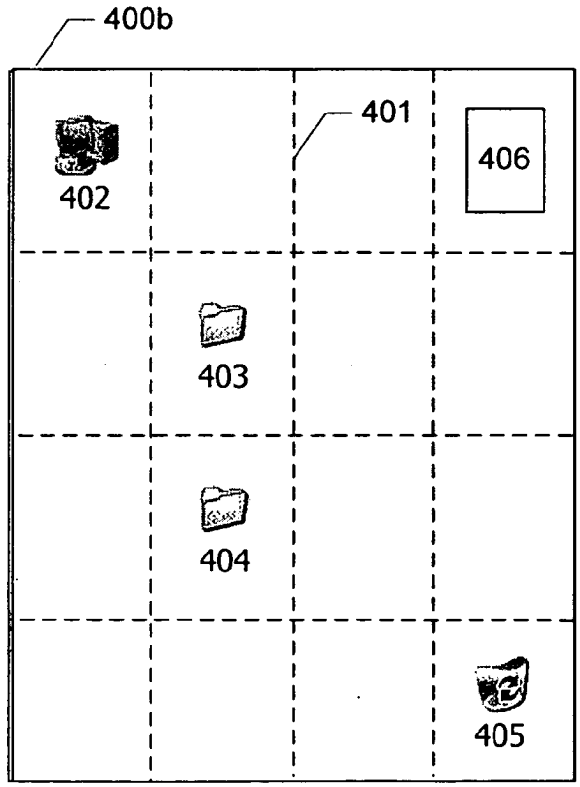


FIG. 4B

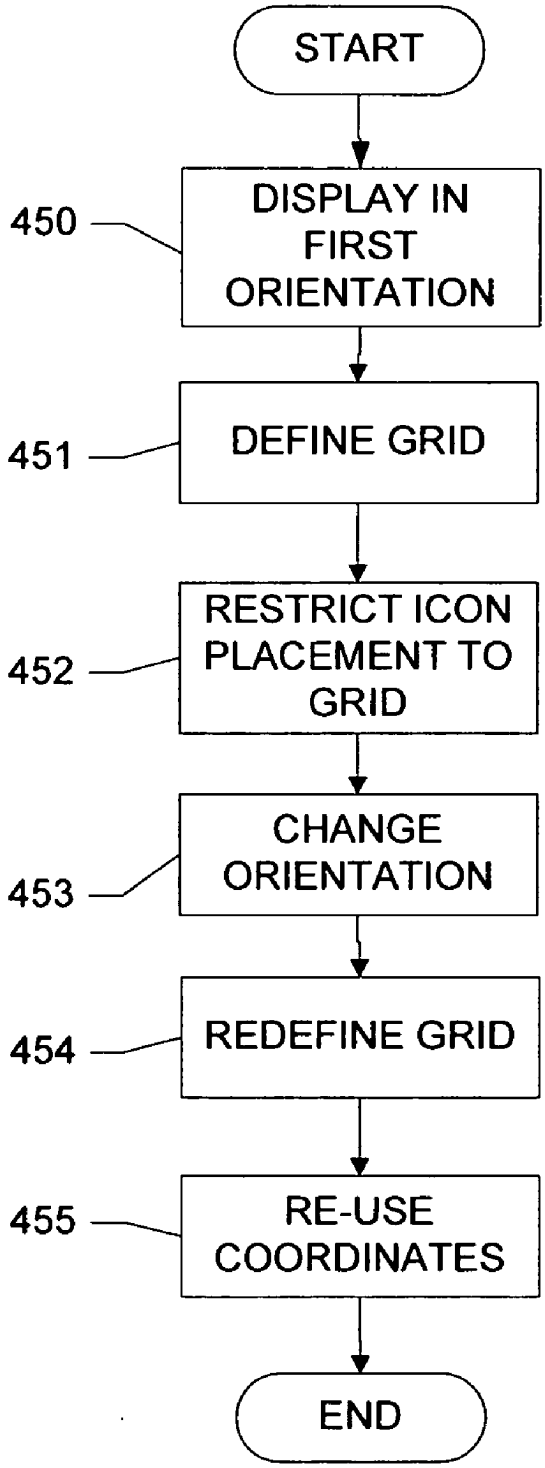

START
450
DISPLAY IN FIRST ORIENTATION
451
DEFINE GRID
452
RESTRICT ICON PLACEMENT TO GRID
453
CHANGE ORIENTATION
454
REDEFINE GRID
455
RE-USE COORDINATES
END

FIG. 4C

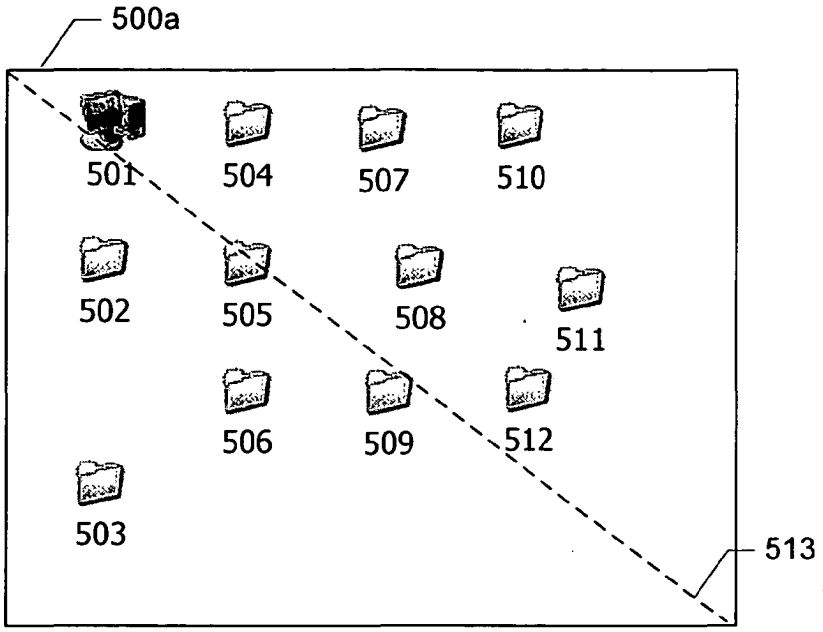

500a
501
504
507
510
502
505
508
511
506
509
512
503
513

FIG. 5A

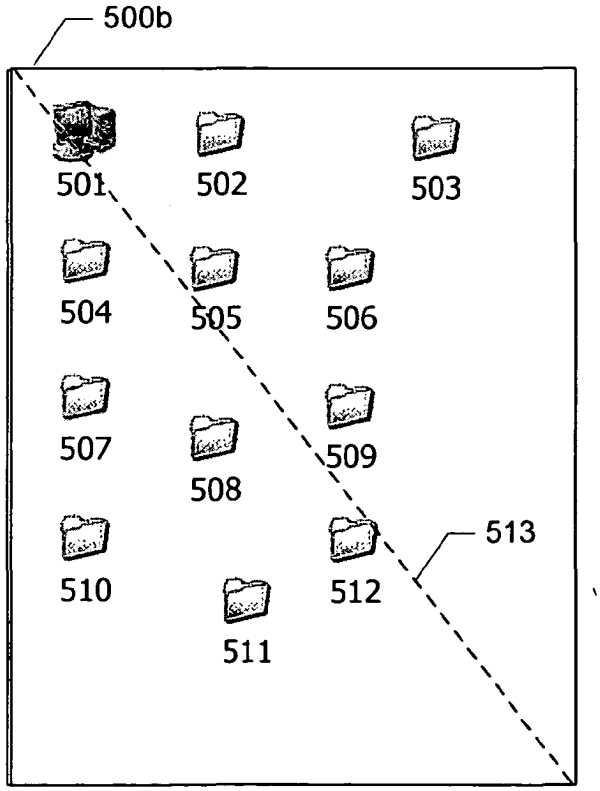

500b
501
502
503
504
505
506
507
508
509
510
511
512
513

FIG. 5B

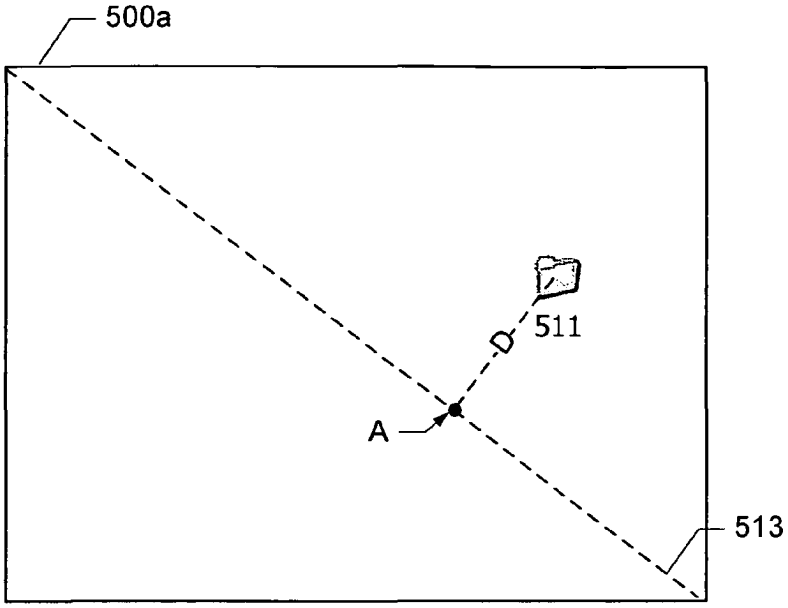


FIG. 5C

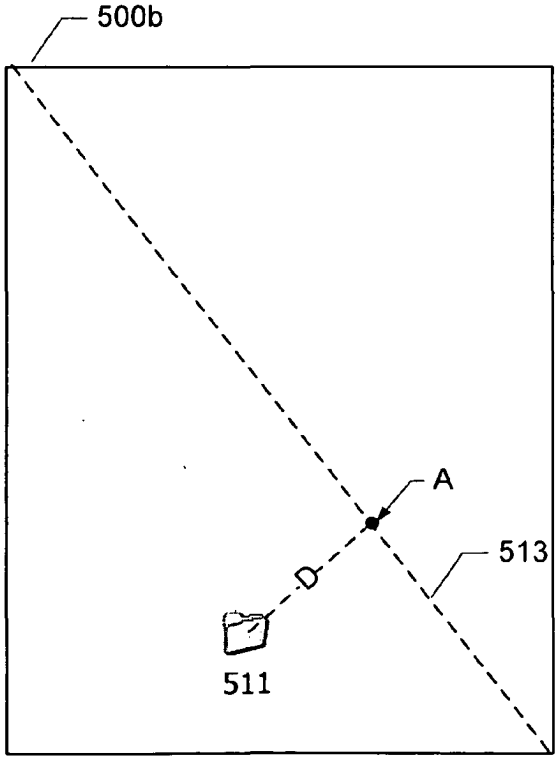


FIG. 5D

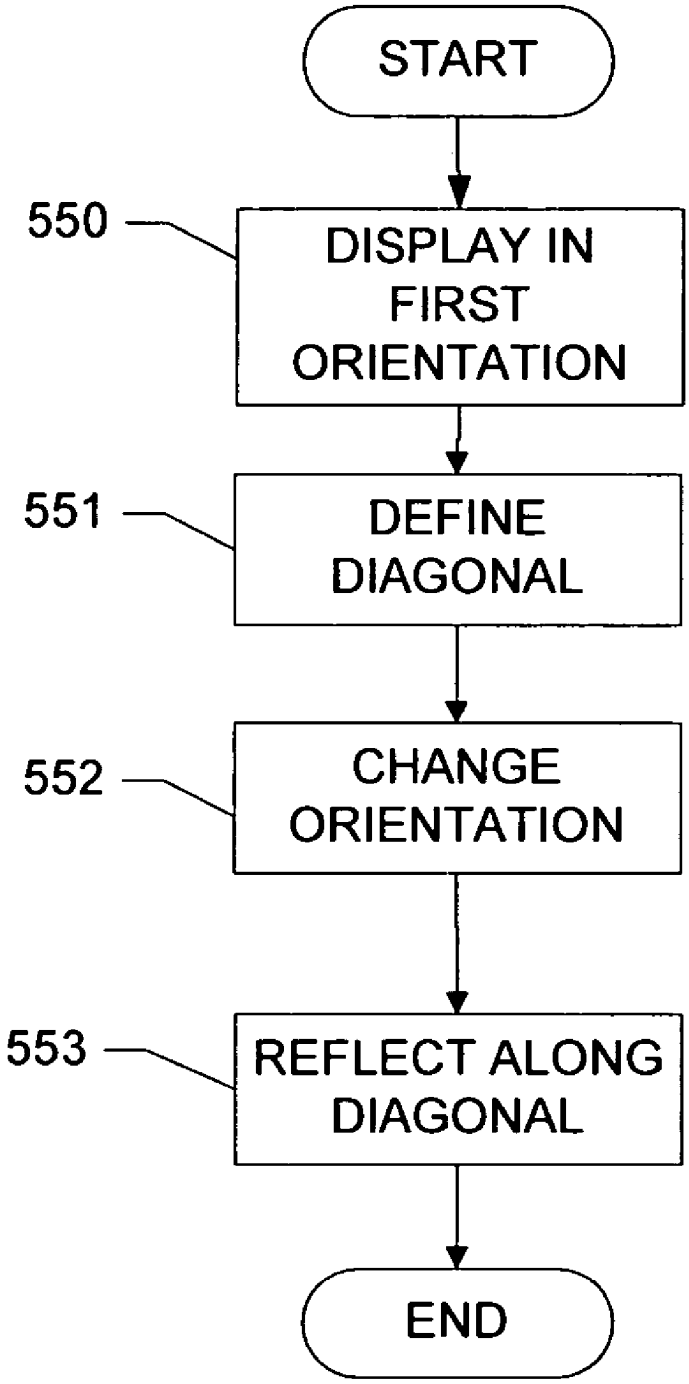

START
550
DISPLAY IN FIRST ORIENTATION
551
DEFINE DIAGONAL
552
CHANGE ORIENTATION
553
REFLECT ALONG DIAGONAL
END

FIG. 5E

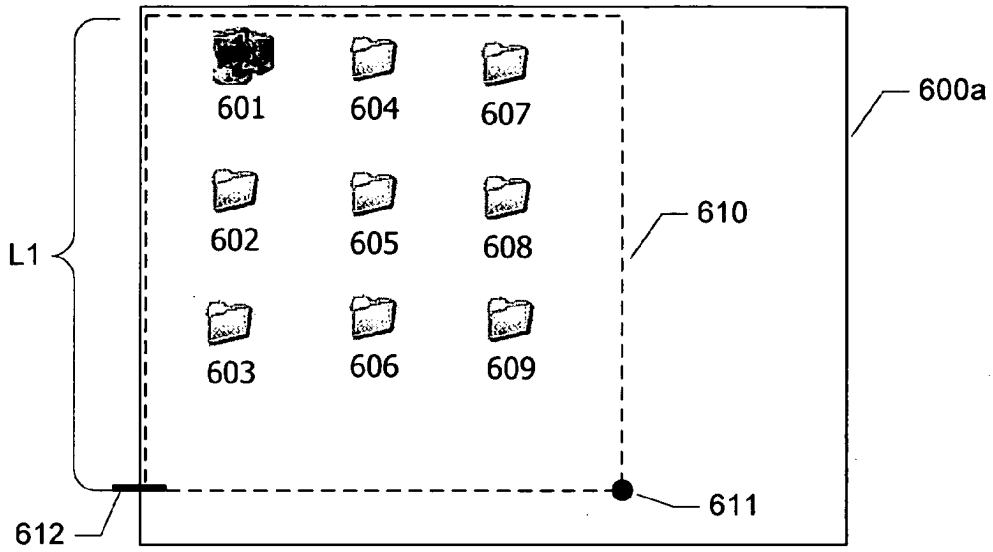

601
604
607
602
605
608
603
606
609
600a
610
L1
611
612

FIG. 6A

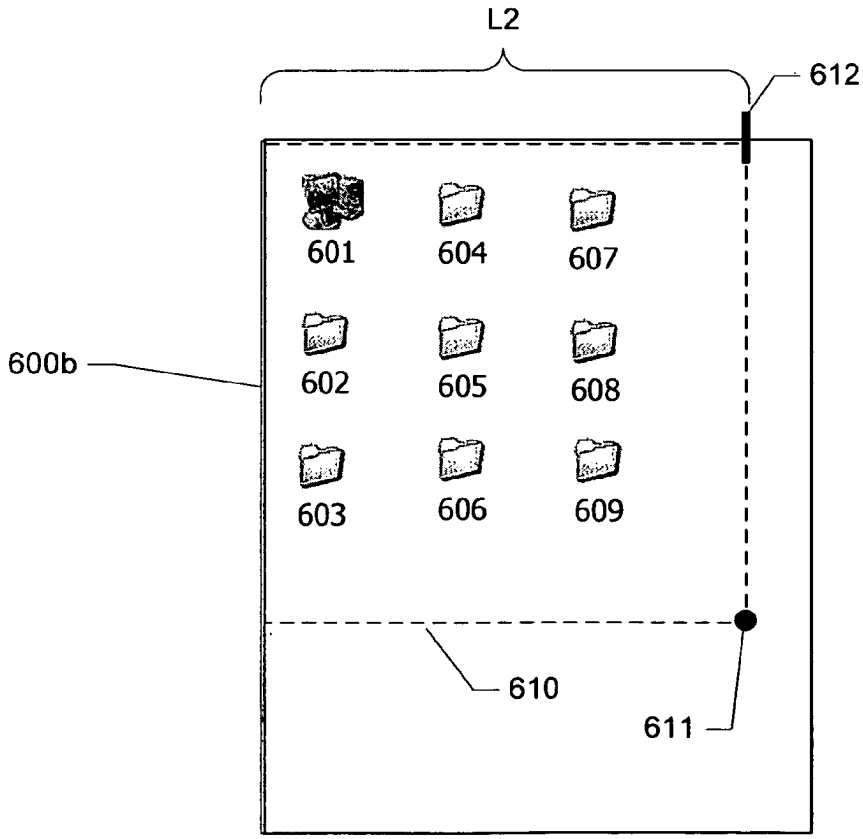

L2
612
601
604
607
600b
602
605
608
603
606
609
610
611

FIG. 6B

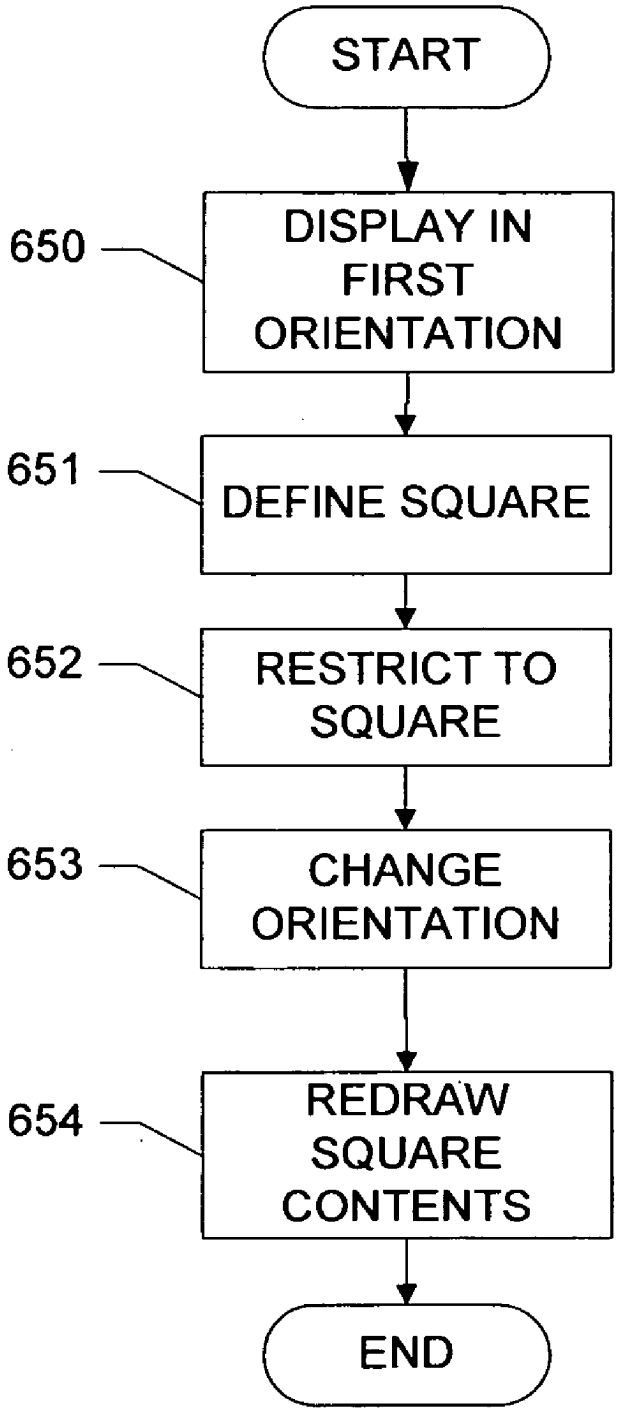

START
650
DISPLAY IN FIRST ORIENTATION
651
DEFINE SQUARE
652
RESTRICT TO SQUARE
653
CHANGE ORIENTATION
654
REDRAW SQUARE CONTENTS
END

FIG. 6C

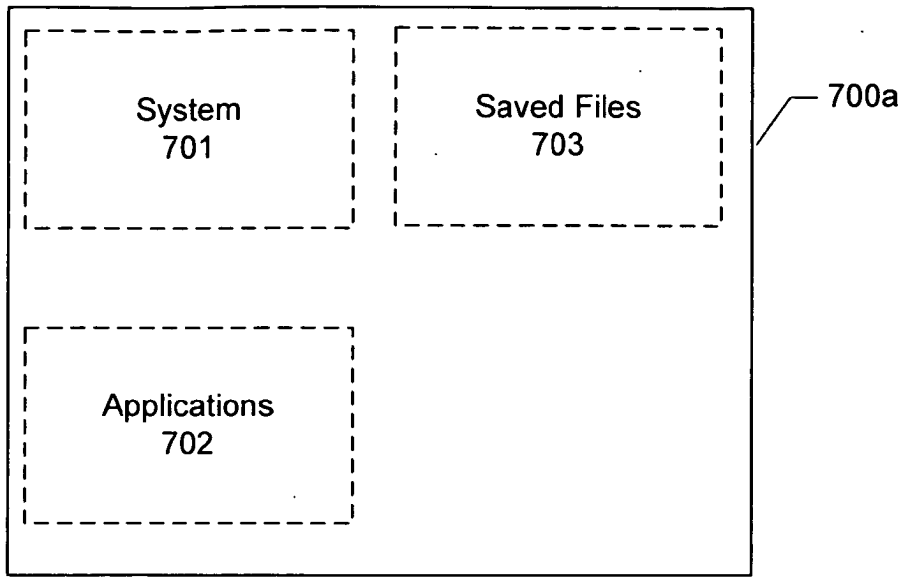

System
701
Saved Files
703
700a
Applications
702

FIG. 7A

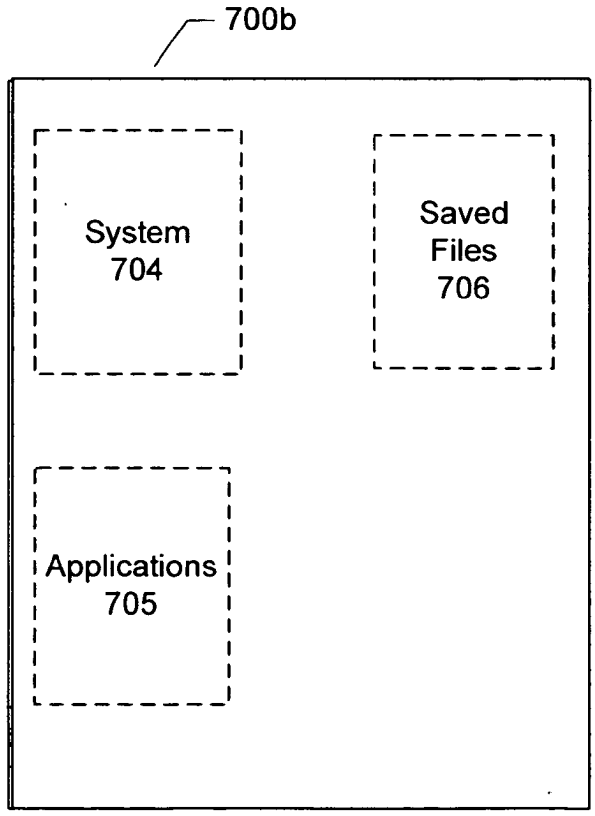

700b
System
704
Saved
Files
706
Applications
705

FIG. 7B

SMART ICON PLACEMENT ACROSS DESKTOP SIZE CHANGES

BACKGROUND

[0001] In today's digital world, the use of graphical user interfaces (GUIs) to display and manage computer information has become ubiquitous. For example, the WINDOWS™ (Microsoft Corporation, Redmond, Wash.) operating systems used in many personal computers employs a GUI desktop with various icons or indicia representing objects on the computer system. The icons may represent applications, data files, hardware components, shortcuts to Internet addresses, and any variety of other data.

[0002] For many users, the GUI desktop is a convenient location for temporary (or even permanent) storage of icons representing applications and data that the users wish to have available as they go about their daily business. Managing the arrangement of these icons, however, can become difficult if a large number of icons are displayed, as individual icons can quickly become lost in a sea of other icons.

[0003] The problem becomes even worse when the user makes changes to the GUI desktop. For example, today's display monitors often allow changes in orientation to switch from landscape to portrait mode, or vice versa. Making such size changes alters the orientation of the usable space on the GUI desktop, and can disrupt the user's icon organization, making it more difficult for the user to function. Accordingly, there is a need for an intelligent management of desktop items that can account for such changes.

SUMMARY

[0004] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

[0005] A desktop may be configured to intelligently allocate placement of icons when changing orientations. The desktop may be divided in a first orientation into a first grid having an equal number of rows and columns of cells, and placement of icons may be restricted to the cells. When the desktop orientation is changed, such as by rotating 90 degrees between landscape and portrait modes, the desktop in the second orientation may be divided into a similarly-dimensioned grid, and the icons in the desktop may have the same grid coordinates in the first and second orientations.

[0006] The rows and columns may be adjusted in spacing and/or size based on the change in orientation. For example, icons may be centered in their respective cells. Similarly, the icons themselves may be adjusted in appearance, such as by changing their height and/or width, or modifying text accompanying the icon.

[0007] Alternatively, a diagonal line may be defined across desktops in two orientations, and when switching from one orientation to another, icons on the desktop may be reflected about this diagonal. The reflection may be accomplished by exchanging X- and Y-icon coordinates, such that rows become columns and vice versa.

[0008] As another alternative, a subregion may be defined on the desktop, such that the dimensions of the subregion fit within the smallest edge of each of the orientations. For example, a square subregion may be defined, having an edge length that fits within the height of the desktop when in portrait mode. Icon placement may be restricted to within this subregion, and upon a change in orientation, the subregion may be reproduced without alteration in the new orientation.

[0009] As yet another alternative, regions on a desktop may be reserved for one or more icon types, such as icons for different file types. Corresponding regions may be defined for multiple orientations, and changing between orientations may result in moving icons between corresponding regions.

[0010] These and other features will be described in greater detail below.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] FIG. **1** illustrates an example of a computing system environment.

[0012] FIG. **2** illustrates an example of a tablet-type computing environment and interface.

[0013] FIG. **3** illustrates another example of a tablet-type computing environment and interface, having a different orientation from that shown in FIG. **2**.

[0014] FIGS. **4***a-c* illustrate GUI desktops and an example process for an approach to handling changes in desktop orientation.

[0015] FIGS. **5***a-e* illustrate GUI desktops and an example process for another approach to handling changes in desktop orientation.

[0016] FIGS. **6***a-c* illustrate GUI desktops and an example process for another approach to handling changes in desktop orientation.

[0017] FIGS. **7***a-b* illustrate GUI desktops for another approach to handling icon placement among different desktop orientations.

DETAILED DESCRIPTION

[0018] FIG. **1** illustrates an example of a suitable computing system environment **100** on which the features herein 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 features described herein. 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**.

[0019] The features herein are described in the general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the features may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, mainframe computers, and the like. The

features 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.

[0020] With reference to FIG. **1**, the exemplary system **100** for implementing features described herein includes a general purpose-computing device in the form of a computer **110** including 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**.

[0021] Computer **110** may include a variety of computer readable media. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. The system memory **130** may include 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, may be stored in ROM **131**: RAM **132** may contain 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**.

[0022] The computer **110** may also include other removable/nonremovable, volatile/nonvolatile computer storage media. By way of example only, FIG. **1** illustrates a hard disk drive **141** that reads from or writes to nonremovable, nonvolatile 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/nonremovable, 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** may be 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** may be connected to the system bus **121** by a removable memory interface, such as interface **150**.

[0023] The drives and their associated computer storage media discussed above and illustrated in FIG. **1** may 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 **110** 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 **195**.

[0024] 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, 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.

[0025] When used in a LAN networking environment, the computer **110** may be connected to the LAN **171** through a network interface or adapter **170**. When used in a WAN networking environment, the computer **110** may include 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.

[0026] FIG. **2** illustrates an illustrative tablet PC **201** that can be used in accordance with various aspects described herein. Any or all of the features, subsystems, and functions in the system of FIG. **1** can be included in the computer of FIG. **2**. Tablet PC **201** includes a large display surface **202**, e.g., a digitizing flat panel display, preferably, a liquid crystal display (LCD) screen, on which a plurality of windows **203** is displayed. Using stylus **204**, a user can select, highlight, and/or write on the digitizing display surface **202**. Examples of suitable digitizing display surfaces **202** include electromagnetic pen digitizers, such as Mutoh or Wacom pen digitizers. Other types of pen digitizers, e.g., resistive, capacitive, or optical digitizers, may also be used. Tablet PC **201** interprets gestures made using stylus **204** in order to manipulate data, enter text, create drawings, and/or execute conventional computer application tasks such as spreadsheets, word processing programs, and the like.

[0027] The stylus **204** may be equipped with one or more buttons or other features to augment its selection capabilities. In one embodiment, the stylus **204** could be imple-

mented as a "pencil" or "pen", in which one end constitutes a writing portion and the other end constitutes an "eraser" end, and which, when moved across the display, indicates portions of the display are to be erased. Other types of input devices, such as a mouse, trackball, or the like could be used. Additionally, a user's own finger could be the stylus **204** and used for selecting or indicating portions of the displayed image on a touch-sensitive or proximity-sensitive display. Consequently, the term "user input device", as used herein, is intended to have a broad definition and encompasses many variations on well-known input devices such as stylus **204**. Region **205** shows a feedback region or contact region permitting the user to determine where the stylus **204** as contacted the display surface **202**.

[**0028**] In various embodiments, the system provides an ink platform as a set of COM (component object model) services that an application can use to capture, manipulate, and store ink. One service enables an application to read and write ink using the disclosed representations of ink. The ink platform may also include a mark-up language including a language like the extensible markup language (XML). Further, the system may use DCOM as another implementation. Yet further implementations may be used including the Win**32** programming model and the .Net programming model from Microsoft Corporation.

[**0029**] FIG. **3** illustrates a second orientation for the tablet PC shown in FIG. **2**. In the FIG. **3** orientation, a rectangular landscape presentation is used, as opposed to the rectangular portrait orientation shown in FIG. **2**. A landscape presentation may offer a GUI desktop that is wider, but shorter, than the presentation offered in the portrait mode.

[**0030**] FIGS. **4***a* and **4***b* illustrate steps that may be taken to help manage changes between display sizes or orientations, such as between landscape and portrait modes. These steps are also shown in the flow diagram of FIG. **4***c*, all of which will be described below. First, as shown in FIG. **4***a* and in step **450** of FIG. **4***c*, a GUI desktop **400***a* is shown in a first landscape mode. To manage placement of icons on the desktop **400***a*, a grid **401** of cells (areas or spaces) **401***a* on the desktop **400***a* may be defined in step **451**. The grid **401** may divide the desktop area into cells **401***a*, and may have an equal number of rows and columns of cells **401***a*. For example, the grid **401** is shown having 4 rows and 4 columns of cells. Although 4 are shown, any other number of rows and columns may be used, depending on user preference.

[**0031**] The actual creation of the grid **401** (which can be displayed, or can be kept invisible) can be accomplished in a number of ways. For example, the grid may be defined through user-entered criteria. The user may be prompted to enter the number and/or size of cells, rows, columns, etc., to define a corresponding grid. Alternatively, the grid may be automatically defined using a predetermined number (e.g., 4, 5, 6, 7, 8, 9, 10) of rows and columns. For example, the system may automatically define the number of rows and/or columns based on the resolution of the display, the number of icons on the desktop, the size of icons, etc.

[**0032**] With the grid **401** defined, the grid may then be used in step **452** to restrict the placement of icons on the desktop. For example, the system may only permit icons to be placed in cells **401***a* of the grid, and may only permit a certain number of icons (e.g., 1, 2, 3, etc.) per cell. Accordingly, the system may prohibit placement of an icon in a location outside of the grid **401**, such that all desktop icons **402-406** are placed in the cells **401***a* of the grid **401**.

[**0033**] As icons **402-406** are added to the desktop in step **452**, the computer system may maintain in memory a data table (or any desired type of data structure) of values identifying coordinates in grid **401** for the placement of various icons. For example, the system icon **402** may be given coordinates of (1,1) to indicate that it is in the first position of the first row of the grid **401**. These coordinates may be used when the orientation changes, as discussed below.

[**0034**] In step **453**, the desktop orientation may be changed to a different orientation, such as a portrait orientation. In response to this change in orientation, and as shown in FIG. **4***b*, when the desktop orientation is changed, e.g., by rotating 90 degrees, the square grid **401** is also redefined in step **454** as being rotated along with the desktop such that the grid columns become rows and the grid rows become columns. As part of this grid rotation, various cells within the grid may have their heights and widths swapped (e.g., former heights become current widths) to reflect the rotation of the grid. However, the rotated square grid **401** still defines a grid having X columns and rows, so when redrawing the screen in step **455** after the change in orientation, the placement of the icons **402-406** may still use the same coordinates defined earlier. In this way, the icons have common grid coordinates between the orientations.

[**0035**] When using the square grid **401** to place icons in step **455**, the spacing of the icons **402-406** may be adjusted based on dimensions of the grid cells **401***a* to present a more uniform appearance in the display mode. For example, the icons may each be centered in their respective cell **401***a*. Accordingly, while icons may be spaced farther apart horizontally in landscape mode (e.g., FIG. **4***a*), the icons may be spaced farther apart vertically in portrait mode (e.g., FIG. **4***b*). Icons themselves may also change in response to the change in modes. For example, an icon **406** may have dimensions that are proportionate to the dimensions of the desktop, such that in landscape mode, an icon is wide and short, while in portrait mode an icon is tall and skinnier. Other aspects of the icons may change as well, such as choice of image, choice of text, placement of text, orientation of text, etc. In some instances, an application on the computer system may define how its representative icon will appear in the different orientations.

[**0036**] FIGS. **5***a-e* illustrate an alternative approach to rearranging icons in response to a change in orientation, with FIG. **5***e* containing an example process flow diagram for this approach. As illustrated in FIG. **5***a* and in the process of FIG. **5***e*, a GUI desktop **500***a* may first be displayed in a first orientation (e.g., landscape) at step **550** with a number of icons **501-512**. The icons **501-512** need not be restricted in their placement, so they need not be placed in any particular order, and need not be aligned to any grid. In placing the icons **501-512**, according to this example or in any other aspect described herein, the user's computer system may maintain in memory a table or other data structure identifying coordinates of the various icons, and other information regarding icon status and/or appearance. These coordinates may be Cartesian, radial, or any other desired coordinate system.

[**0037**] To prepare for, or in response to, a change in the orientation, the computer system may define a diagonal on

the desktop in step **551**. The diagonal may be internally determined by the system, and may be displayed if desired. As shown in FIG. **5***a*, the diagonal **513** extends from an upper-left corner of the desktop to the lower-right corner. Other types of diagonals may be used as well, such as running between the upper-right and lower-left corners, or running between two other points along the periphery of the desktop.

[**0038**] When the desktop orientation is changed in step **552**, such as from the landscape orientation **500***a* to the portrait orientation **500***b* shown in FIG. **5***b*, the system may use this diagonal **513** to adjust the location of the icons in step **553**. When the desktop orientation is changed, the diagonal **513** placement may remain the same (e.g., the diagonal **513** is shown extending between the upper-left and lower-right corners of the desktop in both orientations), and the icons **501-512** may be relocated according to a reflection along diagonal **513** from the original orientation in FIG. **5***a*.

[**0039**] The reflection may be done in a variety of ways. For example, if the system employs a Cartesian coordinate system and stores X- and Y-values for placement of the icons **501-512**, the reflection may be accomplished by converting (if needed) the coordinates to have the coordinate system origin (0,0) located at one endpoint of the diagonal **513** (e.g., the upper-left corner), and then exchanging the X- and Y-values when in the new display mode. So for example, if icon **511** happens to be located at coordinates (400,−200) (e.g., 400 pixels to the right of the upper-left corner, 200 pixels down from that corner), the new placement may be located at coordinates (−200,400) in the new orientation (e.g., 200 pixels to the right of the upper-left corner, 400 pixels down from that corner). Accordingly, a row of icons in the first orientation may become a column of icons in the second orientation, and a column of icons in the first orientation may become a row of icons in the second orientation, where rows are horizontal and columns are vertical.

[**0040**] This reflection may also be accomplished through other geometric approaches. For example, and as illustrated in FIGS. **5***c* and **5***d*, the system may identify a point A along the diagonal **513** from which a perpendicular line intersects the icon being reflected **511**, and the distance D between that point and the icon. In the new orientation, as shown in FIG. **5***d*, the system may place the icon **511** along the same point A on the diagonal, but at a distance D going the opposite direction.

[**0041**] FIGS. **6***a-c* illustrate another example approach to managing icons in response to changes in desktop orientation, with FIG. **6***c* illustrating example process steps. At first, the desktop and icons **601-609** may be displayed in a first orientation in step **650**, such as the landscape orientation in FIG. **6***a*. The user, in step **651**, may then define a square region **610** of the desktop that will be preserved upon changing the desktop orientation. This may be done in any of a variety of ways, such as by a user clicking and dragging a pointer to define a square, or by designating one or more corner points **611** for the square (with a predetermined corner point, such as an upper-left corner of the screen, being automatically defined when only a single corner point **611** is designated), or activating a mode selection that uses a predetermined region **610**.

[**0042**] Alternatively, the square **610** may be defined using a slidebar **612** that is movable along the shorter edge of the rectangular landscape screen. The user may move the slidebar **612** using a pointer, cursor keys, or the like, and in response to the positioning of the slidebar **612**, the system may automatically designate square region **610** based on a distance between the slidebar **612** and a predetermined edge of the desktop. For example, the top edge of the desktop may be designated as an origin for the slidebar **612** in landscape mode, with the square region **610** having edges of length L**1** measured from this origin, and the left edge as an origin in portrait mode, with the square region **610** having edges of length L**2**.

[**0043**] As a further alternative, the square **610** may be defined automatically by the system using a predetermined size. For example, the system may automatically define square region **610** to be a square having an edge length equal to the shorter side of a rectangular display. Or, the user may be prompted to enter a numerical value (e.g., pixels, inches, etc.) identifying a size of the square region.

[**0044**] With the square region **610** defined, the system may then, in step **652**, restrict placement of icons **601-609** to within the region **610**. This restriction, as with other restrictions described herein, may be accomplished in a variety of ways. For example, a user may simply be prohibited from placing a new icon in a restricted area of the screen, and attempts to do so may result in an automatic placement of the new icon in a permitted area of the screen (e.g., the new icon may appear at a location in the permitted area that is closest to the restricted location originally designated by the user), or not at all.

[**0045**] The system may maintain coordinate information for the icons **601-609** that are located within the region **610**, indicating the icons' relative positions within the region **610**. This coordinate information may, for example, identify positions with respect to an origin associated with the region **610** (such as a Cartesian coordinate using the region's lower-left corner as the origin). When, in step **653**, the orientation is changed (e.g., rotated 90 degrees), the system may redraw the same region **610** and icons **601**, without rotation, in step **654**. Alternatively, the region **610** need not be rendered onscreen, and may instead be an internal structure defined by the system.

[**0046**] FIGS. **7***a-b* illustrate examples of a rules-based approach to managing changes in desktop orientation. In this approach, icons may be divided into categories, with one or more rules assigning specific categories to different regions on the desktop. For example, the categories can be based on an object type represented by the icon. Some icons may represent system objects, such as a hardware configuration tool, performance tools, shortcuts, etc.; some icons may represent software applications loaded onto the computer system; and other icons may represent data files saved by the user, such as word document files, pictures, etc. The categories can be defined based on other criteria, as well. For example, icons may be categorized based on other attributes, such as file size, age, creation date of the objects they represent, etc.

[**0047**] The system may permit the user to designate areas of the desktop as being reserved for icons of certain types. For example, the user could use a mouse pointer to define a custom area of the desktop, and may enter a command (e.g., a menu command, a keyboard command, etc.) to assign the custom area to one or more icon types. As illustrated in FIG.

7*a*, this example desktop includes a system icon area **701**, applications icon area **702**, and saved files icon area **703**. The areas may be displayed on the desktop, such as with a different color, appearance, and/or outline, or they may simply be internal areas that are not displayed as part of the desktop.

[**0048**] When new icons are added to the desktop, the system may automatically place icons in a predetermined area based on the icon type and the area criteria. So in the illustrated example, a new icon for a data file may automatically be placed in, or restricted to, area **703**.

[**0049**] To support this rules-based approach, the system may store information identifying one or more icon types that are permitted for a particular area, and the area of the screen, and the system may compare a new icon's type with that criteria to determine where the icon should appear.

[**0050**] Each region identified in a first orientation may also have a corresponding region defined for one or more additional orientations, such that a matched pair of regions exists between the different desktop orientations. The pairs of regions may be identical in dimension and/or size, or they may be adjusted to have different sizes based on user preference or the desktop orientation. As shown in FIG. 7*b*, the user may designate regions **704**, **705** and **706** in portrait view **700***b* as being reserved for system icons, application icons and saved file icons, respectively, which correspond to regions **701**, **702** and **703** in landscape view **700***a*. The criteria and rules defining these regions may all be stored in a common data structure. For example, the system may associate a single icon category (e.g., application files) with multiple areas across different desktop orientations (e.g., area **702** when in landscape mode, and area **705** when in portrait mode).

[**0051**] When the system switches between orientations, it may access the stored rules to identify where the various icons need to be placed in the new orientation, moving icons between matched pairs of regions for the different orientations. In this manner, the user can customize the layout and appearance of the desktop in the different orientations, and can rest assured that they will be able to find icons in the different desktop orientations.

[**0052**] As a further alternative, icons placed on a GUI desktop may be pinned to a predetermined location on a desktop. For example, certain icons may be placed at one or more predetermined locations on a desktop (e.g., one of the corners, a center of an edge, a center of the desktop, etc.), and may be kept at the same location upon a change in orientation. So, for example, an icon pinned to the upper-right corner of a landscape GUI desktop may once again appear at the upper-right corner of the desktop when rotated to portrait mode. If such a pinned icon is used, and a change in orientation results in an attempt to place an icon overlapping the pinned icon, the pinned icon may remain in its place, and the other icon may be automatically relocated to another location, such as a nearest available open area on the desktop.

[**0053**] Using one or more of the features and approaches described above, a user's experience with changing desktop orientations can be improved. Although the description above provides illustrative examples and sequences of actions, it should be understood that the various examples and sequences may be rearranged, divided, combined and subcombined as desired. Accordingly, although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing the claims.

I claim the following:

1. One or more computer-readable media having stored thereon computer-executable instructions for performing a method for placing icons on a graphical desktop when changing orientations, comprising the steps of:

placing a plurality of icons on said desktop in a first orientation, wherein said placing restricts placement of said icons to a first grid in a first orientation of said desktop, said first grid having an equal number of rows and columns of cells;

changing an orientation of said desktop from said first orientation to a second orientation;

in response to changing the orientation, re-placing said plurality of icons on said desktop in said second orientation, wherein said step of re-placing places said icons in accordance with a second grid of said desktop in said second orientation, wherein said second grid has the same number of rows and columns as the first grid, and said icons maintain common grid coordinates in said first and second grids.

2. The one or more computer-readable media of claim 1, wherein said step of placing places each of said plurality of icons in a center of a corresponding one of said plurality of cells.

3. The one or more computer-readable media of claim 1, wherein said step of re-placing further comprises the step of changing a size, color, image, or text of one of said icons.

4. The one or more computer-readable media of claim 3, wherein said step of changing a size, color or text of one of said icons includes the step of modifying a dimension of said one of said icons.

5. The one or more computer-readable media of claim 3, wherein said step of changing a size, color or text of one of said icons includes the step of modifying text associated with said one of said icons.

6. The one or more computer-readable media of claim 1, further comprising the step of receiving user input defining the equal number of rows and columns of cells in said first and second grids.

7. One or more computer readable media having stored thereon computer-executable instructions for performing a method for placing icons on a graphical desktop when changing orientations, comprising the steps of:

displaying a plurality of icons on a desktop in a first orientation;

changing said desktop from said first orientation to a second orientation; and

reflecting locations of said plurality of icons across a diagonal region of said desktop for display in said second orientation.

8. The one or more computer-readable media of claim 7, wherein said diagonal region is defined as a line connecting two opposite corners of said desktop.

9. The one or more computer-readable media of claim 8, wherein said step of reflecting further includes the steps of:

identifying X- and Y-coordinate values for said icons in said first orientation; and

using said X-coordinate values for said icons in said first orientation as Y-coordinate values for said icons in said second orientation, and using said Y-coordinate values for said icons in said first orientation as X-coordinate values for said icons in said second orientation.

10. The one or more computer-readable media of claim 7, wherein said plurality of icons are arranged in a plurality of rows and columns, and wherein said step of reflecting causes rows in said first orientation to become columns in said second orientation.

11. The one or more computer-readable media of claim 8, wherein said step of reflecting further includes the steps of:

identifying a point along said diagonal from which a perpendicular line intersects one of said icons;

identifying a distance between said identified point and said icon; and

identifying a new point along said perpendicular line, said new point being said distance away from said point along said diagonal, and wherein said new point and said icon are on opposite sides of said diagonal.

12. One or more computer readable media having stored thereon computer-executable instructions for performing a method for placing icons on a graphical desktop when changing orientations, comprising the steps of:

displaying a desktop having a plurality of icons in a first orientation, said desktop being rectangular and having a long side and a short side, wherein said icons are restricted to a subset region having a length and width both equal to or shorter than a length of said short side of said desktop;

changing said desktop from said first orientation to a second orientation;

in response to said changing, redrawing said subset region on said desktop in said second orientation, wherein icons in said subset region maintain the same spatial relationship between said first and second orientations.

13. The one or more computer-readable media of claim 12, wherein said step of defining further comprises the step of receiving user input to define dimensions of said subset region.

14. The one or more computer-readable media of claim 12, wherein said subset region is a square region.

15. The one or more computer-readable media of claim 14, wherein said step of defining further comprises the step of receiving user input to move a graphical element along an edge of said desktop.

16. The one or more computer-readable media of claim 12, wherein said step of restricting further comprises the step of, in response to receiving a request to place a new icon in a first area of said desktop outside of said subset region, automatically displaying said new icon in a second area of said desktop inside of said subset region.

17. The one or more computer-readable media of claim 16, wherein said step of automatically displaying further comprises the step of selecting said second area based on proximity to said first area.

18. One or more computer readable media having stored thereon computer-executable instructions for performing a method of placing icons on a GUI desktop, comprising the steps of:

defining a first plurality of regions on a GUI desktop in a first desktop orientation;

assigning at least one icon type to each of said first plurality of regions; and

restricting the placement of new icons on said desktop based on a type of said icons and said assigned icon types.

19. The method of claim 18, further comprising the steps of:

defining a second plurality of regions on said desktop in a second desktop orientation, wherein each of said second plurality of regions corresponds in appearance to one of said first plurality of regions; and

in response to changing said desktop from said first orientation to said second orientation, moving icons between corresponding pairs of regions in said first and second orientations.

20. The method of claim 18, wherein said icon type is based on an attribute of an object represented by said icon.

* * * * *