cursor device button **83**. While selected, the icon **42** and cursor pointer **50** are dragged down until the cursor pointer **50** reaches the active region of handle bar segment **98** which represents drawer space **38**. When segment **98** is reached, simultaneously drawer space **36** closes and drawer space **38** opens. Refer to FIG. **9***c*. Icon **42** is now effectively removed from drawer space **34** because that drawer space has closed and is not accessible. Since icon **42** was selected, it will remain selected and visible after drawer space **38** opens. The cursor pointer **50** and icon **42** are then dragged to the appropriate position of the currently open drawer space **38** and then dropped by releasing the depressed cursor device button **83**. The icon **42** will then become part of the drawer space of drawer space **38**; the transfer is complete of the icon from drawer space **36** to drawer space **38**.

FIG. **10***a* and FIG. **10***b* illustrate a basic block diagram of the operational computer flow of the present invention. The flow chart serves to illustrate the basic computer "engine" of the present invention. Note that routine cursor movement and Desk Drawer option command access, as shown in FIG. **3**, are well known techniques in the art and accordingly are not shown on the flow chart.

Referring to FIG. **10***a,* the flow chart begins at state **200** with the computer generating Desk Drawer in the closed state and the processing logic displaying and activating the handle bar segments **18**, called the first triggered display region. While the drawer is closed, the processing logic at state **202** checks if the pointer is over a handle bar segment. If true then state **203** is entered and if not then the processing moves to state **206** where the processing logic allows the cursor pointer **50** to freely move about the screen location and select **207**, drop **235**, or drag **209** various icons as shown by states **206**, **207**, **208**, **230**, **235** and **209**. State **206** checks if the pointer is activated. If not activated then the processing is directed to state **202**. If true, then at state **230** the computer checks if pointer is on an icon. If false, then the present icon is deselected at state **235** and processing returns to state **202**. If true, the pointer is on the icon and activated, then state **207** is entered. If the icon, was not already selected then the icon would become presently selected at **207**. At state **208**, when the icon is selected while the cursor pointer is moved then the processing is directed to state **209** where the icon is dragged on the display screen. The same is true for state **220**. The processing returns to state **202**.

Referring still to FIG. **10***a*. The computer checks if the pointer **50** reaches a bar segment **18** at state **202** (a segment in this form is also called a first triggered display region). If false, the computer returns to state **206**. If true, the computer goes to state **203** and opens Desk Drawer to display a drawer region corresponding to the bar segment, first triggered display region, selected. The computer next checks to see if the pointer **50** ventures off the drawer region at state **204**. If true, then the computer closes the Desk Drawer by state **205** and returns to state **202**. If false, the computer moves to state **210**. At state **210** the computer checks to see if the pointer **50** has selected another active bar segment, alternate first triggered display region, from handle bar **18**. If true, then the user wants to open another drawer space and the computer simultaneously closes the active drawer space and opens the new drawer space by states **211** and **212** respectively. Then the computer returns to state **204**.

Refer to FIG. **10***b*. If state **210** was false, the computer goes to state **242**. If the pointer is active the computer returns to state **204**, if not then the computer goes to state **213**. At state **213** if the pointer **50** is on the current open handle bar, or also referred to as second triggered display region (on the top usually), then the computer goes to state **214** to check if the pointer is newly activated. If not activated the computer returns state **204**. If activated at **214** the computer then adjusts the drawer space size according to the cursor movement by state **215**, adjusts the location of individual icons within the drawer as necessary **243**, then checks at state **216** to see if the cursor is deactivated. If deactivated, the computer sets the new drawer size and returns to state **204**. If not deactivated the computer returns to state **215**. At state **213**, if the pointer is not on the current open handle bar, then the computer goes to state **217** to perform the routine cursor icon select **220**, deselect **241**, and drag **220** functions (**217**, **218**, **219**, **220**, **240**, **241**, see discussion of similar processing for states **206**, **230**, **235**, **207**, **208** and **209** above). Then the computer returns to state **204**.

FIG. **10***c* shows a more detailed flow diagram of states **203** and **212**. Referring to expanded state **203**, this state operates under both modes of single or multiple drawer spaces. At state **305**, the screen positions of the icons, including icon lateral position, are calculated from their relative positions and placed on the display screen concurrently with the display of state **301**. The processing logic at state **301** opens the selected drawer space and displays that portion on the display screen. At state **302** multiple drawer operation is determined. If multiple then state **303** is entered, else state **304** is entered. At state **303** is multiple drawer spaces have been selected then alternate segments are displayed within the current drawer space so that other drawer spaces may be selected. These alternate segments are called alternate first triggered display regions. Also within state **303**, the current first triggered display region is removed since the current drawer space is displayed. At state **304** the handle bar region of the selected drawer space is displayed at the upper portion of the display region. This handle bar region is called the second triggered display region. The processing then exits expanded state **203**.

Referring still to FIG. **10***c,* expanded state **212** operates under multiple drawer space mode. State **315** calculates the screen icon positions, including icon lateral position, from their relative positions and displays the icons on the display screen. Next, state **307** opens the new drawer space and places it on the screen. State **309** removes the current first triggered display region, the segment that triggered the new drawer space to open. State **311** displays the alternate segments representing the other drawer spaces, alternate first triggered display regions. State **313** displays the second triggered display region, or handle bar used to adjust the size of the drawer space for each drawer. The processing then exists expanded state **212**.

The preferred embodiment of the present invention, a computer controlled software user interface providing a ready access secondary display region, is thus described. While the present invention has been described in one particular embodiment, it should be appreciated that the present invention should not be construed as limited by such embodiment.

What is claimed is:

1. A computer controlled information management system including a display screen containing a first display region which allows the display of a plurality of overlapping windows at a given time, the system comprising:

means for displaying a second display region covering at least a portion of said first display region and any windows being displayed within said portion of the first display region, said second display region, when present in a mode, being completely viewable within said display screen such that, while said second display region is displayed it is always displayed over and