

9905774-

9905774-7

1

INTERFACE CONTROL FOR SWITCHING BETWEEN DISPLAY REGIONS
ON A DISPLAY

FIELD OF THE INVENTION

The present invention is directed to the field of displays. It is more specifically directed to display interface control.

BACKGROUND OF THE INVENTION

Limited display size on many devices make it hard to effectively display controls for switching between multiple display regions. One approach to the problem of displaying display region controls on a limited display space is to use tabs with text labels as a visual indicator of the display regions and their contents, and a click on the tab as the means to switch between the display regions. The size of these tabs is either predetermined by the programmer, in which case the display region labels must fit within the predetermined tab size; or is determined by the length of the label to be displayed on the tab. In cases where the number of display regions is greater than the number of tabs and their labels which will fit on the display, the tabs are made to scroll by clicking on a left or right arrow, which scrolls the tabs to the left

or the right. In this case, not all the tabs and their labels representing the display regions are visible to the user at the same time. An advantage of tabs is that it offers a clear, well understood user interface metaphor for switching between different places. In particular, in the physical world, it enables switching between different documents or files in a same way as in tabs on the files of a physical file system.

An alternative approach to this problem of providing controls for switching between display regions, is to use icons. Icons provide a pictorial representation of an object and can also be labeled. An icon together with a corresponding label makes a strong connection between the icon and the thing it represents. Clicking on an icon displayed on a screen takes the user to the display region represented by said icon and label. Icons can be placed in an icon bar and be in a fixed location, this is important as it allows experienced users of the system to learn the spatial location of an icon and thus enables rapid navigation. The advantage of icons is that they are small and so maximize use of display space. The disadvantage of this solution is that the icons are generally used for switching between applications, rather than between display regions within an application. There is not the strong metaphor of switching between different documents as is the case with the tab representation. A further disadvantage of icons is that to make them more meaningful, they are often displayed along with a label. This label can take a lot of room on the display. To reduce the amount of space taken by the icons and their labels, the labels

can be made to appear only when the cursor is over the icon. A disadvantage of this approach is that the label can only be made to appear momentarily when the cursor is over the icon, it does not remain visible once the cursor moves away. Thus, the label is not visible once the display region has been displayed on the screen and the user is working within that display region.

Yet another approach is to use menus which contain items whose label and/or icon represents different display regions. The disadvantage of this approach is that the labels and/or icons representing the display regions are not visible until the menu has first been selected. However, menus have the advantage of being fixed in order so as a user becomes more familiar with the system they learn the location of a menu item. They can then easily navigate to the item they want to select.

SUMMARY OF THE INVENTION

The present invention provides a method, apparatus and system to overcome problems with devices that have limited display size which make it difficult to display controls such as labeled icons for switching between multiple display regions. One aspect of the present invention provides a resizable tab control that makes optimal use of a minimal display. The resizable tab control is especially advantageous for the small displays of pervasive computing devices, but is not

limited to use on only small devices. This control uses the metaphor of tabs to indicate different display regions.

In a particular embodiment, tabs are displayed across the top, side or bottom of the screen in a consistent order. An icon and a label are visually displayed within the tab. The tab icon is a visual representation that indicates the content of the display region. The tab label is a textual representation that indicates the content of the display region. We define an icon to always be smaller than a label. Only one display region can be shown on the screen at a time. The display region that is currently shown on the screen is the selected region. When a display region is not the currently selected region, its tab is sized to show only the icon. When a display region is the selected region, the tab is sized to show both the icon and the label.

BRIEF DESCRIPTION OF FIGURES

These and other objects, features, and advantages of the present invention will become apparent upon further consideration of the following detailed description of the invention when read in conjunction with the drawing figures, in which:

FIG. 1 shows a method for an example embodiment in accordance with the present invention;

FIG. 2 shows an example of a method for initialization of a tab bar in accordance with the present invention;

FIG. 3 shows an example of a tab creation method in accordance with the present invention;

FIG. 4 shows an example of a method for adding a tab to a tab bar in accordance with the present invention;

FIG. 5 shows an example of a method for drawing the tabs in accordance with the present invention;

FIG. 6 shows an example of a method for handling a pen down within the tab bar in accordance with the present invention;

FIG. 7 shows an example of a the computer system in accordance with the present invention;

FIG. 8 shows an example of an embodiment of the current invention.

DETAILED DESCRIPTION OF THE INVENTION

Limited display size on pervasive computing devices such as the IBM Workpad make it hard to display controls for switching between multiple display

regions. The present invention provides a resizable tab control that makes efficient use of a small display screen area although the resizable tab control is especially advantageous for the small displays of pervasive computing devices, but it is not limited to use on only small devices. The invention is also advantageously employed by display devices which have no space and/or tab size problem.

The resizable tab control uses a collection of tabs to visually represent different display regions. In a particular embodiment the tabs are advantageously displayed across the top, side and/or bottom of the screen in a consistent order. An icon and a label are visually displayed within the tab. The tab icon is a visual representation that indicates the content of the display region. The tab label is a textual representation that indicates the content of the display region. A tab icon size is by definition generally smaller or equal to the size of its corresponding tab label. In general it is assumed that an icon is always smaller than its corresponding label. When the label is smaller, the label may be interchanged with and take the place of the icon in implementing the present invention. Although in some embodiments two or more display regions are shown on the display, employing display area splitting, windowing etc., generally only one display region is usually shown on the screen at a time. The

display region that is currently shown on the screen is the currently selected region. When a display region is not the currently selected region, its tab is sized to show only its corresponding icon. When a display region is the selected region, the tab is sized to show both the icon and the label of that selected region besides the region information.

Resizable tabs maintains the advantages of tabs as a user interface metaphor for switching between display regions, thus having an advantage over a purely icon-based representation. It also overcomes the disadvantages of tabs and their labels taking up too much space on a limited sized display. It does this by using only icons to represent the contents of display regions that are not currently selected. In a particular embodiment it also has the advantage of allowing the maintenance of a constant order to the visual representation of the contents of the display region. This facilitates navigation as a user becomes more familiar with the application that is utilizing the tabs. Finally, because the label is always visible for the selected display region, resizable tabs are better than icons with labels which are shown only when the cursor is over the icon.

In an embodiment of the present invention the method is included as part of the operating system of a pen-based computer system to allow for incorporation into multiple application programs. In an alternate

embodiment the invention is incorporated directly into an application on either a multipurpose or single purpose device. Although a general embodiment places the tabs horizontally across the top of screen, the tabs could be placed in any location. It's advantageous for the tabs to maintain an order and for the associated label and icon to be displayed so that it is at least in part within the bounds of the tab. For example, the tabs may be displayed on the side of the screen and the label may be vertically displayed.

The applications in the pen-based system utilized in an example embodiment consist of forms. In the embodiment, we define a tab bar to be that portion of the screen containing the visual representation of the tabs. (See 811 in Figure 8). Referring to Figure 1, when a form that has a tab bar is created, our method starts 100. In block 101, the tab bar region is initialized. After initialization, the system monitors for input 102. When the application form receives input, it processes the event as in block 103. Pen down events are handled by the tab bar event handler 103.

Figure 2 shows an example of the initialization starting with "initialize" 200 the tab bar. First the tabs are created (201). The currently selected tab are set (202), and finally the tabs are drawn (203).

Figure 3 describes an example of the creation of the tabs starting with "create tabs" 300. The tab bar object is created (301). The tab bar object contains

information pertaining to the collection of tab objects including: a pointer to the linked list of tab objects, a pointer to the first and last tab objects, a pointer to the first and last visible tab objects, and a pointer to the currently expanded tab object. It also contains information linking the tab bar object to its outer container, namely, the coordinates of its top left corner and the extent of its enclosing rectangle. After creating the tab bar object, each tab ranging from "1" to "n" is added to the tab list maintained by the tab bar object (302).

Figure 4 shows an example scheme to add a tab to the tab bar. It starts with "add tab" 400. A tab object is created and initialized (401). A tab object contains information about an individual tab including the tab label, the label font, the bitmap identifier (used to access the tab icon), the message id (used to indicate to the containing application which tab was clicked), and the current extent of the tab object's containing rectangle. This latter value varies as a function of which tab is currently the selected tab and is set when the tab bar object is rendered to the screen (203). Once the tab bar has been created and initialized, the tab label and icon are set (402). This also includes specifying the tab label font. Next in block 403, the message id is set. Finally, the tab object is added to the tab list maintained by the tab bar object (404).

The final step draws the tabs to the screen. An example of a 'draw tab' procedure is described in Figure 5.

This procedure begins with "draw tabs" 500. It then clears the region of the display where the tab bar is going to appear (501). Then it checks 502 if there are any more tabs to be drawn. If there are more tabs to be drawn it gets the id of the next tab (503). It then computes the size of the icon bitmap for that tab (504). In block 505 it checks to see if the tab to be drawn is marked as the selected one. If it is the selected tab the size of the label is calculated in block 508. If it isn't the selected one, the label size is set to 0 (506). Then, before drawing the tab, the procedure first checks that there is room on the screen to display it (508). If there isn't room on the screen, block 509 draws an indicator to show that there are more tabs than those currently viewable on the screen. If there is room on the screen, block 510 calculates width, height and origin of the rectangle bounding the tab. Once these coordinates are set, block 511 draws the tab outline, and then block 512 draws the tab contents consisting of the tab icon, and if it is the selected tab, the tab label.

In an embodiment, when a user taps on the screen, the application initiates a pen down event. A tab bar event handler scheme shown in Figure 6, is then called. It starts a "tab bar handler event" 600, by determining 601 that the pen down is within the tab bar region of the display. If it isn't, this procedure is done 614, and the pen down event is sent to the application. If the pen down is within the tab bar display region, 602

checks to see if the event is in the currently selected tab. If it is in the currently selected tab then it does nothing. If it isn't in the currently selected tab, then block 603 determines which tab the pen is in, and saves the id of that tab. The tab that the pen is within we refer to here as the down tab. The event handler then checks in 604 to see whether the pen is still down. If the pen is down then in block 607 it gets the coordinates of the pen, checks to see if they are still within the down tab in 608 and if they are and the down tab is not yet highlighted 611, it highlights the down tab 612, and then returns to 604 to check again if the pen is still down. If the pen coordinates are not within the bounds of the down tab, then if the down tab is highlighted (609) it unhighlights the down tab in block 610 and then returns to 604 to check again to see if the pen is still down. If the pen coordinates are not within the bounds of the down tab (608), and the down tab is not highlighted, then the procedure returns to 604 to check if the pen is still down. If the pen is not still down when checked in 604, and the up point of the pen was inside the bounds of the down tab 605, a tab select event notification is sent to the application. This event typically initiates the drawing of the contents of the display region associated with the tab and the redrawing of the tab bar as outlined in Figure 5. If the up point is not within the bounds of the down tab then nothing happens and the handler exits.

A computer system embodiment of the present invention is shown in Figure 7. The invention may be implemented on

any general purpose or dedicated pointer-based computer system. This includes for example pen-based, mouse, track-ball, and track-pad computer systems. For the purposes of illustration, the invention is described in connection with a pen-based system.

A computer system in the example embodiment described generally includes:

- a bus or other communication device 701 for communicating information; and coupled with bus 701,

- a processor 702 for processing information;

- a read only memory (ROM) or other static storage device 703 for storing information and instructions for processor 702;

- a random access memory (RAM) or other storage/memory device 704 for storing information and instructions for processor 702;

- a display device 705, such as a cathode ray tube, liquid crystal display, etc., for displaying information to the computer user;

- and a pointing device 706, such as a mouse, pen, track-ball, or track-pad, coupled with at least one user-signal-detecting device such as a button

on a mouse for pointing at locations on the screen of display device 705 and for communicating information and command selections to processor 702.

In an alternate embodiment the display device is a touch-sensitive display device 705 which receives location input from a users finger or other pointing device and for communicating information and command selections to processor 702.

A software architecture embodiment may utilize a graphical user interface (GUI) environment. GUI is well known in the art. The present invention may be implemented as part of an application program, as part of a user interface of a single purpose system, or in any other suitable manner.

The GUI system utilized by the preferred embodiment is based on a message-driven architecture. Message-driven architectures are well-known in the art and employed in most GUI systems. In the architecture, any user input or system event is translated into a message, which is sent to objects that are relevant to the input or event.

Thus the present invention provides a mechanism for switching between display regions and their contents. Consideration is now given to objects being displayed in a display embodiment in accordance with the present inventor. Referring to Figure 8, a tab 802 has an

associated display region 803 containing application specific data or contents such as lists, buttons, text, and other user interface objects. There is also a tab bar display region 811. The tab bar display region 811 contains multiple tabs 801, 802, 807, 809. A tab 802 has an associated tab label 806 and a tab icon 805. Typically, only one display region can be visible at a time. The tab associated with the visible display region is known as the selected tab. In the present embodiment, the selected tab is sized so that both the tab label and the tab icon is displayed within its frame. In general however, the tab icon and tab label are displayed on the screen in a manner that associates them with the currently selected tab, and the selected tab sized accordingly. The tabs associated with display regions currently not shown on the screen are displayed in a collapsed form showing only their tab icon. For example, tab 801 shows icon 804, tab 807 shows icon 808, tab 809 shows icon 810. When there is more than one tab, they are advantageously displayed in a fixed order. This order may be predetermined or determined algorithmically. In the present embodiment, collapsed tabs are displayed so that they appear set back or deeper than the current tab in a manner known to those familiar with the art.

If there are more tabs than fit across the screen, then one or more, visual indicators such as an arrow icon or a "more" button, is placed on the screen. Tapping on the visual indicator causes all, or a subset of the tabs currently not displayed on the screen to be displayed on the screen.

In the embodiment being described when a tab is selected, as by tapping on it with a stylus, the display region currently displayed on the screen is removed from the screen, and its associated tab is reduced in size to show only its associated tab icon. The display region associated with the selected tab is displayed on the screen, its associated tab is expanded to a size that will show both the tab icon and the tab label.

A current procedural implementation example centers on two data structures: the TabBarObject and the TabObject.

The TabBarObject contains information pertaining to the collection of TabObjects including a pointer to a linked list of TabObjects, a pointer to the first and last TabObjects, a pointer to the first and last visible TabObjects, and a pointer to the currently expanded TabObject. It also includes information linking the TabBarObject to its outer container, namely, the coordinates of its top left corner and the extent of its enclosing rectangle.

The TabObject contains information about an individual tab including the tab label, the label font, the bitmap identifier, the control id (used to indicate to the containing application which tab was clicked), and the current extent of the tab's containing rectangle. This

latter value generally varies as a function of which tab is currently expanded and is set when the TabBarObject is rendered to the screen.

Associated with these two data structures are several procedures:

CreateTabBarObject takes a pointer to a containing user interface object (a form) and the identifier of a resource describing the placement and extent of the tab bar in the form. It returns a pointer to the created TabBarObject which is used in subsequent calls. This procedure is called by the code responsible for creating and rendering the containing form.

AddTabObject takes a pointer to a TabBarObject, a label string, a bitmap id, and a control id. It creates an initialized TabObject and links it into the list of TabObjects in the TabBarObject. It returns a pointer to the new TabObject. This procedure is called by the code responsible for creating and rendering the containing form. It is called once per tab.

SelectTabObject takes a pointer to a TabBarObject and a control id. It sets TabBarObject's currently expanded tab pointer. This procedure is called after the tabs have been added to the tab bar and before the tab is rendered.

`UpdateTabObject` takes a pointer to a `TabBarObject`. It computes the bounding rectangles for all the visible tabs and displays the tabs in either their collapsed or expanded form as indicated by the `TabBarObject`. The bounding rectangles are saved in the individual `TabBarObjects` to allow the determination of which tab was clicked in a subsequent user input event. This procedure is called as the final step in the code responsible for creating and rendering the containing form.

`TabObjectHandleInput` takes a pointer a `TabBarObject`, a pointer to the containing form, and an event descriptor. This procedure determines which (if any) of the tabs contains the point in the event descriptor (typically, the down point of a click or a pen down). It captures the mouse or pen input stream, tracking whether the events continue to be within the tab. During this event tracking, changes in containment are indicated visually within the tab itself. Once an up event is received (typically, the up point of a click or a pen up), this procedure throws an event containing the control id of the "clicked" tab. This event is caught by the containing application and will typically cause the current form to be closed and a new form to be opened (which begins the process of tab bar creation with `CreateTabBarObject`).

`DeleteTabBarObject` takes a pointer to a `TabBarObject` and performs memory deallocation and cleanup. This procedure is called when the containing form is closed.

Thus one aspect of the present invention, is to provide a method comprising the step of displaying controls employed for switching between multiple display regions in a display system by: generating a collection of tabs to visually represent at least a subset of the multiple display regions on the display module; providing each tab with a tab icon and a tab label; displaying the tab icon, tab label and data content for a selected region of said subset; and displaying only tab icons of all non selected regions of said subset.

In an embodiment the method further comprises the step of providing said tab data content with a plurality of user selectable objects; and/or a user selecting one of the selectable objects by clicking on said one of the selectable objects; and/or forming each tab icon to have an icon display size which is less than or equal to a label display size for each tab label.

In some embodiments of the method the step of providing includes ordering the tab icon with respect to the tab label in a particular set order; and/or the step of ordering displays the tab icon ahead of the tab label ; and/or the step of generating includes furnishing a capability to

change particular display regions that are included in the subset ; and/or the step of furnishing includes providing directional tab movement arrows ; and/or the step of displaying includes visually showing tab icons and tab labels in proximity to at least one edge of the display system ; and/or the edge is a top edge of the display system ; and/or the step of displaying includes limiting the display content to include information related to only one tab icon.

In another aspect the present invention provides an apparatus to switch between multiple display regions in a display module. The apparatus comprising: a collection of tabs; a generator to generate at least a subset of the multiple display regions on the display module and to provide each tab of the collection of tabs with a tab icon and a tab label; a display controller to control the display of the tab icon, tab label and tab data content for at least one selected region taken from said subset, and the tab icon of all non selected regions of said subset.

In some embodiments of the apparatus further comprises the step of a user interface to provide said tab data content with a plurality of user selectable objects; and/or the user interface enables a user to select one of the selectable

objects by clicking an object and/or the display controller includes a sizing module to size each tab icon to have an icon display size which is less than or equal to a label display size for each tab label and/or the display controller includes an ordering module to order the tab icon with respect to said tab label in a particular set order and/or the ordering module orders displays of the tab icon ahead of displays of a corresponding tab label; and/or the generator furnishes a capability to change particular display regions that are included in the subset; and/or wherein the display controller includes a subset selector which provides directional tab movement arrows to change the subset; and/or the display controller visually shows tab icons and tab labels in proximity to at least one edge of the display module ; and/or the edge is a side edge of the display module ; and/or the display controller limits the display content to include information related to only one tab icon.

In another aspect of the present invention, a computer program product is provided which comprises a computer usable medium having computer readable program code means embodied therein for displaying controls employed for switching between multiple display regions in a display system, the computer readable program code means in said computer program product comprising computer readable program code

means for causing a computer to effect: generating a collection of tabs to visually represent at least a subset of the multiple display regions on the display module; providing each tab with a tab icon and a tab label; displaying the tab icon, tab label and data content for a selected region of said subset; and displaying only tab icons of all non selected regions of said subset.

In some embodiments of the computer program product, the computer readable program code means in said computer program product further comprising the step of providing said tab data content with a plurality of user selectable objects; and/or the computer readable program code means in said computer program product further comprising computer readable program code means for causing a computer to effect a user selecting one of the selectable objects by clicking on said one of the selectable objects ; and/or the computer readable program code means in said computer program product further comprising computer readable program code means for causing a computer to effect forming each tab icon to have an icon display size which is less than or equal to a label display size for each tab label ; and/or the step of providing includes ordering the tab icon with respect to said tab label in a particular set order ; and/or the step of ordering displays the tab icon ahead of the tab label ;

and/or the computer readable program code means in said computer program product further comprising computer readable program code means for causing a computer to effect generating includes furnishing a capability to change particular display regions that are included in the subset; and/or the step of furnishing includes providing directional tab movement arrows; and/or the step of displaying includes visually showing tab icons and tab labels in proximity to at least one edge of the display module ; and/or the edge is a bottom edge of the display module ; and/or the step of displaying the tab icon includes limiting the display content to include information related to only one tab icon.

In another aspect of the present invention, a method is provided for switching between display regions on a device. The method comprising: providing a plurality of associated display regions having a set of interface objects and content fields; generating a tab icon and a tab label for each of the display regions; formulating an ordered tab list of all display regions; selecting a starting region from the display regions as a first region to be displayed; displaying the starting region; and displaying tab icons for at least a subset of all other display regions.

In an embodiment the method further comprises selecting another region to be a new displayed region; and/or the steps of displaying the starting region includes showing the tab icon, the tab label and the contents of the starting region ; and/or of displaying the tab icons includes employing the ordered tab list for arranging the tab icons ; and/or displaying the tab icon, the tab label and the contents associated with said another region.

In some embodiments of the method, the subset includes all icons neighboring the displayed region that fit on a display area of the device; and/or the subset includes all the display regions; and/or the step of providing an ability for changing the subset displayed ; and/or the device is a hand-held computer.

In still another aspect of the present invention, the invention provides an apparatus for switching between display regions on a device. The apparatus comprising: a plurality of associated display regions having a set of interface objects and content fields, each of the display regions having a corresponding tab icon and tab label; a generator to generate a starting region from the display regions as a first region to be displayed; a display controller to display the starting region, and to display the tab icons for at least a subset of all

other regions; and/or the display controller selects another region to be a new displayed region ; and/or the display controller controls the device to show the tab icon, the tab label and the contents of the starting region ; and/or the display controller forms and employs an ordered tab list for arranging the tab icons ; and/or the display controller causes the display of the tab icon, the tab label and the contents associated with said another region ; and/or the subset includes all icons neighboring the displayed region that fit on a display area ; and/or the subset includes all the display regions ; and/or further comprises a means for changing the subset to be displayed ; and/or the device is a hand held computer.

In another aspect of the present invention, a computer program product is provided which comprising a computer usable medium having computer readable program code means embodied therein for switching between display regions on a device by: providing a plurality of associated display regions having a set of interface objects and content fields; generating a tab icon and a tab label for each display region; selecting a starting region from the display regions as a first region to be displayed; displaying the starting region; and displaying tab icons for a subset of all other regions.

In still another aspect of the present invention, the computer readable program code means in the computer program product further comprising the step of selecting another region to be a new displayed region ; and/or the step of displaying the starting regions includes showing the tab icon, the tab label and the contents of the starting region ; and/or the computer readable program code means in said computer program product further comprising the steps of formulating an ordered tab list of all display regions, and displaying the tab icons includes employing the ordered tab list for arranging the tab icons ; and/or the step of displaying the tab icon, the tab label and the contents associated with said another region.

In some embodiments of the computer program product the subset includes all icons neighboring the displayed region that fit on a display area ; and/or the subset includes all the display regions ; and/or the computer readable program code means in said computer program product further comprising the step of providing a means for changing the subset displayed ; and/or the device is a hand-held computer.

It is noted that this invention may also be provided as an apparatus or a computer product. It may be implemented as an article of manufacture comprising

a computer usable medium having computer readable program code means embodied therein for causing a computer to perform the methods of the present invention.

It is further noted that although the description of the invention is made for particular arrangements of steps, the intent and concept of the present invention are suitable and applicable to other arrangements. It will be clear to those skilled in the art that other modifications to the disclosed embodiments can be effected without departing from the spirit and scope of the invention.

**CLAIMS**

We claim:

1. A method comprising:

displaying controls employed for switching between multiple display regions in a display system by:

generating a collection of tabs to visually represent at least a subset of the multiple display regions on the display module;

providing each tab with a tab icon and a tab label;

displaying the tab icon, tab label and data content for a selected region of said subset; and

displaying only tab icons of all non selected regions of said subset.



2. A method as recited in claim 1, further comprising the step of providing said tab data content with a plurality of user selectable objects.

3. A method as recited in step 2 further comprising a user selecting one of the selectable objects by clicking on said one of the selectable objects.

4. A method as recited in claim 1, further comprising forming each tab icon to have an icon display size which is less than or equal to a label display size for each tab label.

5. A method as recited in claim 1, wherein the step of providing includes ordering the tab icon with respect to the tab label in a particular set order.

6. A method as recited in claim 5, wherein the step of ordering displays the tab icon ahead of the tab label.

7. A method as recited in claim 1, wherein the step of generating includes furnishing a capability to change particular display regions that are included in the subset.



8. A method as recited in claim 7, wherein the step of furnishing includes providing directional tab movement arrows.

9. A method as in claim 1, wherein the step of displaying includes visually showing tab icons and tab labels in proximity to at least one edge of the display system.

10. A method as in claim 9, wherein the edge is a top edge of the display system.

11. A method as in claim 1, wherein the step of displaying includes limiting the display content to include information related to only one tab icon.

12. An apparatus to switch between multiple display regions in a display module, the apparatus comprising:

a collection of tabs,

a generator to generate at least a subset of the multiple display regions on the display module



and to provide each tab of the collection of tabs with a tab icon and a tab label;

a display controller to control the display of the tab icon, tab label and tab data content for at least one selected region taken from said subset, and the tab icon of all non selected regions of said subset.

13. An apparatus as recited in claim 12, further comprising a user interface to provide said tab data content with a plurality of user selectable objects.

14. An apparatus as recited in step 13, wherein the user interface enables a user to select one of the selectable objects by clicking an object.

15. A apparatus as recited in claim 13, wherein the display controller includes a sizing module to size each tab icon to have an icon display size which is less than or equal to a label display size for each tab label.

16. An apparatus as recited in claim 12, wherein the display controller includes an ordering



module to order the tab icon with respect to said tab label in a particular set order.

17. An apparatus as recited in claim 16, wherein the ordering module orders displays of the tab icon ahead of displays of a corresponding tab label.

18. An apparatus as recited in claim 12, wherein the generator furnishes a capability to change particular display regions that are included in the subset.

19. An apparatus as recited in claim 12, wherein the display controller includes a subset selector which provides directional tab movement arrows to change the subset.

20. An apparatus in claim 12, wherein the display controller visually shows tab icons and tab labels in proximity to at least one edge of the display module.

21. An apparatus as in claim 20, wherein the edge is a side edge of the display module.



22. An apparatus as in claim 12, wherein the display controller limits the display content to include information related to only one tab icon.

23. A computer program product comprising a computer usable medium having computer readable program code means embodied therein for displaying controls employed for switching between multiple display regions in a display system, the computer readable program code means in said computer program product comprising computer readable program code means for causing a computer to effect:

generating a collection of tabs to visually represent at least a subset of the multiple display regions on the display module;

providing each tab with a tab icon and a tab label;

displaying the tab icon, tab label and data content for a selected region of said subset; and

displaying only tab icons of all non selected regions of said subset.



24. A computer program product as recited in claim 23, the computer readable program code means in said computer program product further comprising the step of providing said tab data content with a plurality of user selectable objects.

25. A computer program product as recited in claim 24, the computer readable program code means in said computer program product further comprising computer readable program code means for causing a computer to effect a user selecting one of the selectable objects by clicking on said one of the selectable objects.

26. A computer program product as recited in claim 23, the computer readable program code means in said computer program product further comprising computer readable program code means for causing a computer to effect forming each tab icon to have an icon display size which is less than or equal to a label display size for each tab label.

27. A computer program product as recited in claim 23, wherein the step of providing includes ordering the tab icon with respect to said tab label in a particular set order.



28. A computer program product as recited in claim 27, wherein the step of ordering displays the tab icon ahead of the tab label.

29. A computer program product as recited in claim 23, the computer readable program code means in said computer program product further comprising computer readable program code means for causing a computer to effect generating includes furnishing a capability to change particular display regions that are included in the subset.

30. A computer program product as recited in claim 29, wherein the step of furnishing includes providing directional tab movement arrows.

31. A computer program product as recited in claim 23, wherein the step of displaying includes visually showing tab icons and tab labels in proximity to at least one edge of the display module.

32. A computer program product as recited in claim 31, wherein the edge is a bottom edge of the display module



33. A computer program product as recited in claim 23, wherein the step of displaying the tab icon includes limiting the display content to include information related to only one tab icon.

34. A method for switching between display regions on a device, the method comprising:

providing a plurality of associated display regions having a set of interface objects and content fields;

generating a tab icon and a tab label for each of the display regions;

formulating an ordered tab list of all display regions;

selecting a starting region from the display regions as a first region to be displayed;

displaying the starting region; and

displaying tab icons for at least a subset of all other display regions.

35. A method as in 34, further comprising selecting another region to be a new displayed region.

36. A method as in 34, wherein the steps of displaying the starting region includes showing



the tab icon, the tab label and the contents of the starting region.

37. A method as in 34, wherein the steps of displaying the tab icons includes employing the ordered tab list for arranging the tab icons.
38. A method as in 35, further comprising displaying the tab icon, the tab label and the contents associated with said another region.
39. A method as in 34, wherein the subset includes all icons neighboring the displayed region that fit on a display area of the device.
40. A method as in 39, wherein the subset includes all the display regions.
41. A method as in 39, further comprises the step of providing an ability for changing the subset displayed.
42. A method as in 34, wherein the device is a hand-held computer.



43. An apparatus for switching between display regions on a device, the apparatus comprising:
- a plurality of associated display regions having a set of interface objects and content fields, each of the display regions having a corresponding tab icon and tab label;
 - a generator to generate a starting region from the display regions as a first region to be displayed;
 - a display controller to display the starting region, and to display the tab icons for at least a subset of all other regions.
44. An apparatus as recited in claim 43, wherein the display controller selects another region to be a new displayed region.
45. An apparatus as in claim 43, wherein the display controller controls the device to show the tab icon, the tab label and the contents of the starting region.
46. An apparatus as in claim 43, wherein the display controller forms and employs an ordered tab list for arranging the tab icons.



47. An apparatus as in claim 44, wherein the display controller causes the display of the tab icon, the tab label and the contents associated with said another region.
48. An apparatus as in claim 43, wherein the subset includes all icons neighboring the displayed region that fit on a display area.
49. An apparatus as in claim 48, wherein the subset includes all the display regions.
50. An apparatus as in claim 48, further comprising a means for changing the subset to be displayed.
51. An apparatus as in claim 43, wherein the device is a hand held computer.
52. A computer program product comprising a computer usable medium having computer readable program code means embodied therein for switching between display regions on a device by:
providing a plurality of associated display regions having a set of interface objects and content fields;



generating a tab icon and a tab label for each display region;

selecting a starting region from the display regions as a first region to be displayed;

displaying the starting region; and

displaying tab icons for a subset of all other regions.

53. A computer program product as recited in claim 52, the computer readable program code means in said computer program product further comprising the step of selecting another region to be a new displayed region.

54. A computer program product as recited in claim 52, wherein the step of displaying the starting regions includes showing the tab icon, the tab label and the contents of the starting region.

55. A computer program product as recited in claim 52, the computer readable program code means in said computer program product further comprising the steps of formulating an ordered tab list of all display regions, and displaying the tab icons includes employing the ordered tab list for arranging the tab icons.



56. A computer program product as recited in claim 53, the computer readable program code means in said computer program product further comprising the step of displaying the tab icon, the tab label and the contents associated with said another region.
57. A computer program product as recited in claim 52, wherein the subset includes all icons neighboring the displayed region that fit on a display area.
58. A computer program product as recited in claim 57, wherein the subset includes all the display regions.
59. A computer program product as recited in claim 57, the computer readable program code means in said computer program product further comprising the step of providing a means for changing the subset displayed.
60. A computer program product as recited in claim 52, wherein the device is a hand-held computer.