Arranging display of GUI elements- A TRIZ based analysis

Umakant Mishra
1. Introduction

A graphical user interface has a number of advantages over its predecessors. It helps user to do complex operations through simple pointer movements. The new generation GUIs provide a lot of advanced features through icons and other types of controls on a graphical user interface. The power of GUI has made it more and more popular and extends its use from computer to television, phone, photocopier, fax, camera and other types of consumer devices.
One of the biggest drawbacks of the graphical user interface is that it consumes the active display area associated with the device. Whether the display screen is a part of a television, computer, or any other consumer device, the graphical user interface appears on the active area of the display screen. As a result the GUI masks a part of the display screen and the amount of information that can be viewed on the screen is reduced.

**Ideal Arrangement (IFR)**

Ideally the GUI should offer all its features without blocking any part of the display screen. The features should be readily available and the user should not face any difficulty in searching them. Operating the GUI should require minimum amount of pointer movements.

**Different arranging solutions**

The conventional graphical user interfaces include various types of display objects. Some of them are position indicating display objects (such as a shortcut), some are selectable display objects without internal location distinctions (such as icons) and some are selectable display objects with internal location indications (such as windows and menus).

Some display objects permanently occupy screen real estate (like buttons and toolbars), some objects expand when selected and shrink when not used (like menus and windows), and some objects adjust their size according to available screen space. It is necessary to organize all the objects in such a way that they consume minimum amount of valuable screen and at the same time be easily accessible to the user with minimum pointer movement.

There are several techniques to organize the objects in a graphical user interface to get various advantages.

- Using a large virtual workspace where the screen is thought as a movable viewport onto the workspace. The user moves the screen to view the contents of other parts of the virtual workspace.

- Using multiple virtual workspaces to display a large number of visual elements. The user can switch from one workspace to another, and see a detailed view or a condensed view of the workspaces.

- Using different rooms for different operations, such as using “mailroom” for mailing. The user goes to a room through a “door” and performs only specific related activities in the room.

- The concept of “desktop” emphasizes on the accessibility of the graphical elements. As in case of a physical office desktop, the most frequently used resources are placed in the most easily accessed locations.
Arranging the items on the desktop based on different parameters such as by name, type, size, date etc.

In some methods, the graphical objects are arranged in physical sets, such as “toolbars”, “taskbars”, “control bars”, “on-screen menus” and the like. They are grouped and placed together for ease of searching.

In some systems, like televisions, a command menu or other information may be displayed at one side, overlaid on, and impinging upon, the main program being watched.

Displaying only the basic options at the startup. The advanced options will be displayed only on user request.

In many computer software applications, arrays of graphical objects are arranged around the perimeter of the display screen.

Floating options that allow toolbars, toolboxes and other components to float to different parts of the screen.

Attachable and detachable toolbars and similar other components which can be detached from the main window and placed outside.

Using drawer like components where the toolbars or other such components open up when the pointer is moved onto them.

Using a full screen mode which instantly hides / shows all the toolbars on a single keystroke.

2. Inventions on arranging graphical elements

2.1 Method and apparatus for selecting button functions and retaining selected options on a display (US Patent 5243697)

Background problem
A menu is very important in a graphical user interface as it can provide a large number of options to the user without occupying any permanent screen space. But the major drawback of the menu mechanism is that the user has to open the menu again and again each time he wants to select a particular item. There is a need to keep the frequently accessed menu items displayed on the screen so that the user can access the desired item without opening and navigating the menu every time.
Solution provided by the invention
Patent 5243697 (invented by Hoeber et al., assigned by Sun Microsystems, issued in Sep 1993) discloses a pushpin to retain the frequently used menus on the display screen.

According to the invention when the menu button is selected by the user, a menu appears in a rectangular box containing a pushpin and several menu items. If the user clicks on the pushpin then the temporary menu box is converted into a permanent window and remains on the display regardless of other display operations. The user may click again on the pushpin to release the permanent placement of the menu on the screen.

This invention provides an intuitive means for retaining frequently used menus on a graphical user interface screen.

TRIZ based analysis
The invention improves the accessibility to menu items by keeping the menu screen opened for successive use. This method avoids the user to open it again and again every time he needs (Principle-10: Prior action).

According to the invention, the user can “pin” the menu to keep it open or “release the pin” for normal use (Principle-15: Dynamize).

2.2 Method and apparatus for facilitating operator reconfiguration of a graphical user interface in a data processing system (Patent 5384910)

Background problem
The conventional graphical user interfaces are rigid in structure and displays all the items in a menu simultaneously without leaving any role for the operator in designing the menu. This causes operator confusion and makes the software inconvenient and unpopular. There is a need to streamline the interface and eliminate the arbitrary display features and minimize operator confusion.
Solution provided by the invention
Patent 5384910 (invented by Torres, assigned by IBM, issued Jan 1995) uses a menu-formatted graphical user interface, which can be configured by the operator. The initial display configuration is set by the software manufacturer, which can later be modified in response to operator input.

The invention provides (1) adding a new menu field (2) removing a particular menu field and (3) automatically providing a visual representation of the particular menu fields. The end-user operator can rearrange the (1) textual content or (2) iconographic representation format or (3) both.

![Menu Interface Diagram]

This menu-formatted graphical user interface is directly manipulated by the operator. The size of the GUI is automatically adjusted in response to the inclusion or removal of items made by the operator.

TRIZ based analysis
The invention provides a menu formatted graphical user interface, which may be configured by the end user (or operator) for inclusion or exclusion of menu items (Principle-24: Intermediary, Principle-15: Dynamize).

The sizes of the visual components are automatically adjusted in response to the inclusion or deletion of items (Principle-15: Dynamize, Principle-25: Self service).

2.3 Method and apparatus for providing conditional cascading in a computer system graphical user interface (US Patent 5425140)

Background problem
In a cascading menu system, the user is presented with a main menu including a plurality of choices, each of the choices representing a basic function again having a plurality of choices. This advance function of a menu interface is very desirable for more experienced GUI users but seen as burdening and confusing to the novice users.
Solution provided by the invention

Patent 5425140 (invented by Bloomfield, et al., assigned by International Business Machines Corporation, issued in Jun 1995) provided a means of shielding the users from advanced operations in a graphical user interface.

The invention includes an advanced function activator button adjacent to one of the user selectable action items. The advanced functions are to be displayed only if the advanced function activator option has been selected by the user.

This option enhances the ease of use of a graphical user interface by novice users, as the advanced functions are not displayed to them.

TRIZ based analysis

The GUI should display advanced functions for advanced users, but the display of advanced functions should not confuse the novice users (contradiction).

The invention isolates the advanced functions from the basic functions and does not display the advanced functions to a common user (Principle-2: Taking out).

The user can decide whether to view only basic functions or view the advanced functions by checking an advanced option check box (Principle-15: Dynamize).

The invention provides a method of dynamically classifying advanced items and basic items. For example, when the user selects a submenu item that is different from the default submenu item, the presently selected submenu item becomes the default item or behavior (Principle-15: Dynamize).
2.4 Snap control for relocating elements of a graphical user interface (US Patent 5704050)

Background problem
The desktop of a graphical user interface contains various objects such as text, hyperlinks, icons, toolbars and so on. Sometimes the limited space of desktop becomes full of large number of objects which leads to a messy desktop. How to expand the available space in the desktop to accommodate more icons and other GUI elements?

Solution provided by the invention
Redpath disclosed a method of utilizing the portion beyond the edge of the screen (Patent 5704050, Assigned to IBM, Dec 97). The invisible area beyond the screen is utilized for temporarily storing objects, which are currently not used. The invention provides a snap control through which the user can selectively move objects to a hide position, which is beyond the edge of the display screen.

This invention enables the user to intuitively use space beyond the edge of the display. This method indirectly expands the effective area of the desktop to be utilized by the user.

TRIZ based analysis
The desktop should be able to display all types of icons, windows and other GUI elements (Ideal Final Result).

We want to display buttons, icons and other GUI elements on the desktop but we don’t want to look the desktop messy with so many elements (Contradiction).

The invention does not end the GUI at the edge of the display screen. It effectively expands the area of a desktop being utilized by a user (Principle-37: Expansion) and allows the to use the space beyond the boundary of the screen (Principle-17: Another dimension).
The method moves out the unused icons to an invisible space beyond the screen while allowing to maintain a frame of reference (Principle-2: Taking out).

2.5 Grouping of computer operations (US Patent 5929851)

Background problem
The graphical user interfaces provide various graphical objects to access various functionalities of an application. It’s necessary to change the icons and tools in the interface according to the need of the user. Some applications provide mechanism for customizing toolbars and menu bars through drag and drop operations, but there are no mechanism for customizing other types of collective views such as grouping actions with icons and buttons.

Solution provided by the invention
Vanessa Donnelly invented a method of grouping actions (Patent 5929851, Assigned by IBM, Jul 99). According to the invention, the user can select an action in order to invoke a desired operation. The user is provided with an easy to use mechanism for grouping actions together as desired, so that his commonly used actions are provided in one place.

TRIZ based analysis
The invention provides customization of icons and buttons (Principle-15: Dynamize).

The user can group similar actions or buttons together, which will facilitate ease of finding and ease of execution (Principle-5: Merging).
2.6 Method and apparatus for arranging displayed graphical representations on a computer interface (US Patent 6043817)

**Background problem**

A desktop is used to display links to various computer resources (such as files, folders, application programs etc.). This arrangement helps the user to access the resources easily and reduces the burden of remembering their original locations. Typically a user is tempted to create shortcuts for all frequently used resources, which results in a cluttered desktop.

Like a physical desktop the GUI desktop should also cleaned and organized regularly. Currently there are methods to reorganize the position of icons in the desktop. There is a need for an improved method of organizing desktop objects.

**Solution provided by the invention**

Patent 6043817 (invented by Bolnick et al., assigned by Microsoft Corporation, issued in Mar 2000) discloses a method of arranging the repositionable icons on a desktop based on a selected criterion.

The invention uses frames to arrange the display of graphical elements. Each frame designates a perimeter on the desktop and a filter property that associates the frame to a particular type of computer resources. For example, all “document” type resources will be organized in a specific container.

The invention determines the size for each one of the sets of graphical user interface boxes. It organizes the GUI boxes in a front-to-back stacking order. The smaller GUI boxes are ordered towards to front of the stacking order and larger GUI boxes are ordered towards the back of the stacking order.
TRIZ based analysis
The invention automatically organizes GUI boxes in a front-to-back stacking order including creating an overlapping view wherever required (Principle-7: Nested doll).

Arranging the opened windows in various modes, viz., stack, cascade and tile (Principle-17: Another dimension).

Automatically determining the size and position for each window based on the position of other windows according to the arrangement indicated by the selected mode (Principle-25: Self Service).

2.7 Method and system for displaying graphical objects on a display screen (US Patent 6448986)

Background problem
When the GUI elements like toolbars are loaded onto the screen they cover some part of the screen. When more such toolbars are loaded there is an increased likelihood that a desired data or a desired portion of an image has been covered up by the overlying toolbars. To avoid this one has to remove the toolbars or scroll the data or image.

One solution to this is the so-called full screen mode in windows which instantly hides / shows all the toolbars on a single keystroke. Another approach is to use a floating toolbar; still another approach is to use a dropdown toolbar etc. But all of them still have some part of the screen blocked and use some cumbersome method to display and hide the toolbars. There is a need for a method to automatically hide the GUI objects without any additional cumbersome operation.

Solution provided by the invention
Kim Smith (patent 6448986, assigned by Spotware Technologies LLC, issued in Sep 2002) discloses a method of displaying the graphical user interface in such a way that all the data remains visible irrespective of whether the toolbars or graphical objects are displayed or hidden.

According to the invention the program or operating system resizes the data screen to fit in whatever size display region is available, such region being generally smaller when graphical objects are displayed on the display screen and generally larger when the graphical objects are hidden there from.
The user displays graphical objects in a corresponding tool area defined in the display screen by moving a cursor or other pointing device into a control region defined in the display screen. The control program or operating system is useful not only with personal computer systems, but also with advanced televisions, PC/TVs, office equipments, consumer electronic devices and others.

**TRIZ based analysis**

Toolbars are necessary for faster access of commands but they occupy screen space. Displaying more toolbars increase the likelihood of covering a desired portion of the data or image on the screen. It is necessary to display toolbars but without obscuring any part of the valuable data (*Contradiction*).

One option is to remove the unwanted toolbars (*Principle-2: Taking out*), another is to scroll the data or image for viewing (*Principle-15: Dynamize*).

But both these options require some physical action to be done by the user. Hence they are counter-productive. It is necessary to get the visibility of the data or image without any additional manual effort (*Contradiction*).

The invention hides the toolbar in normal operating time and displays them when the pointing device into a control region defined in the display screen (*Principle-34: Discard and recover*).

Even when the toolbars are displayed, the program or operating system resizes the data screen to fit in whatever size display region is available. Thus when the toolbar is visible the screen is zoomed out and when the toolbar is hidden the screen is zoomed in (*Principle-15: Dynamize*).
2.8 Method and system for automatically resizing and repositioning windows in response to changes in display (US Patent 6473102)

Some display systems offer the user the ability to change the resolution of the display device. But increasing or decreasing the resolution results in decreasing and increasing the size of display items correspondingly, and often shifts the location of components from their original position. But there are some components, such as tool palettes and utility windows, which are not desirable to be shifted from their original positions.

Similarly, while switching the display from a full-sized monitor to an LCD screen, the positions and sizes of the objects on the display can change significantly, because of the operating parameters of the two display devices. This may even make some items unviewable because they are positioned outside the display area. It is necessary to provide a mechanism that is capable of maintaining windows and similar objects in an accessible condition even if there are changes in the display environment.

Solution provided by the invention
Rodden et. al, disclosed a method of automatically resizing and repositioning windows (Patent 6473102, assignee Apple Computers, Oct 2002) in response to changes in display environments or display parameters. According to the invention, the window position is recalculated and redrawn within the newly available display area. A minimum size is retained to display at least the minimum required information. For example, if the window is a utility window containing buttons, the minimum size of the window requires that at least one button be visible.

If the minimum size is not present in the available display area, the window is moved into a free area to attain the minimum size. Conversely, if there is more space after displaying the minimum information, the system decides whether to display additional integral components. After the window is displayed, the control elements, such as scroll buttons etc. are redrawn at the new size and position.
TRIZ based analysis
The invention resizes and relocates the windows to be displayed in an altered display environment (Principle-15: Dynamize). Although the components are resized, the positions of the windows remain in tact regardless of changes in the size or resolution of the display device (Principle-39: Calm).

If the space is not sufficient to display the window, the window shrinks to display the minimum required information. Conversely, if there is space to display more than the minimum required information, the system will determine whether to display additional integral components (Principle-16: Partial or excessive action).

2.9 Easy method of dragging pull-down menu items onto a toolbar (US Patent 6621532)

Background
Accessing an option through navigating a menu tree is time taking. This is worse in case of a sub-menu item which needs to activate and go through several level of menu by controlling the mouse pointer. Selecting an item on a toolbar is faster as it does not require activation of any menu. On the other hand a toolbar permanently occupies some real estate on the GUI.

Example Invention:
US Patent 6621532 (invented by Mandt, assigned by IBM, Sep 03) discloses a method of dragging pull down menu items onto a toolbar. According to the invention when the user drags a menu item and drops on the toolbar, the menu is automatically converted to a toolbar button. This facilitates the user to easily access the option during later use.

TRIZ based analysis
The control functions in a GUI should be accessible in a single click (Ideal Final Result). The frequently used menu items should be accessible in a single click but without blocking the valuable screen space (Desired result).
If menu items are displayed permanently like toolbar items then they will occupy more screen space. If menu boxes are closed then the screen space is saved, but the user has to open the menu boxes every time he wants to access the option (Contradiction).

The invention provides a drag and drop option to convert the menu items to toolbar buttons. The user can drag the frequently used menu items and drop them onto the toolbar to create toolbar buttons. Thereafter the menu items can be accessed from the toolbar by a single click (Principle-15: Dynamize).

3. Summary

This article has analyzed 9 inventions from the US patent database from a TRIZ perspective. The article draws the Ideal Final Result and illustrates how each of these inventions solves some contradictions to achieve some or other aspect of the Ideal Final Result. Some interesting solutions are:

⇒ Using floating toolbars and other components to easily change their position by dragging.

⇒ Keeping a menu box open by using a pushpin.

⇒ Dynamically adjusting the number and position of visual elements based on the available screen space.

⇒ Creating frames on a desktop to organize specific type of icons within separate individual frames.

⇒ Displaying only the basic functions at startup and displaying the advanced functions on request.

⇒ Studying user behavior and adding or removing items from the display dynamically based on user behavior.

⇒ Taking out the unused icons to an invisible space beyond the screen.

⇒ Grouping buttons and other similar controls for ease of finding.

⇒ Automatic resizing of windows and toolboxes based on available screen space.

⇒ Hiding toolbars during normal operation and displaying them when the pointer is moved onto a control region.
Reference to patents:


9. US Patent 6621532, “Easy method of dragging pull-down menu items onto a toolbar”, invented by Mandt, assigned by IBM, Sep 03 (Menu Article)

Other references:


