Using TRIZ for maximizing information presentation in GUI

Umakant Mishra
1. Introduction

The size of the display screen is typically seen as a constraint in a graphical user interface. This is because:

- The modern software tools provide numerous graphic elements in their GUI demanding more and more screen space.

- The user often needs to work on multiple documents simultaneously, which requires more screen space.
A task-switching environment demands more control elements, which may not be relevant to the current task. These common controls also require screen space.

The processing and memory capability of a computer can create and manipulate very large documents, databases and images by using various software tools. But the size of the display screen is typically small to display these objects or documents.

1.1 Lack of screen space- the problem analysis

Ideally the screen should display all the information that the user needs (Ideal Final Result).

Solution: Increase the screen size by using large monitors or display devices (Principle-37: Expansion).

But large monitors or display devices are very expensive. They are also difficult to move and manage. Besides, there is a physical limitation to expand a monitor size beyond a limit (Contradiction).

Reduce the size of fonts, pictures and graphic elements (Principle-35: Parameter change).

But reducing the size of data and images reduces the clarity of viewing (Contradiction).

Physically partition the screen into a number of windows to display discrete categories of object or data (Principle-1: Segmentation).

Allow overlapping of windows so that there can be more number of windows than the physical screen size (Principle-17: Another dimension).

Bring the most active window onto the top for ease of operation as in time division multiplexing (Principle-1: Segmentation).

Use a hybrid approach of space division (to display multiple windows on the same screen space) and time division (displaying as and when required) for efficient use of virtual workspaces (Principle-40: Composite).

In a windows environment the overlapping foreground windows disrupt the visibility of background objects. If the foreground object persists for a long time, the background object is disrupted till that time. Ideally the foreground windows and toolbars should not block the background data or objects (Contradiction).
The concept of floating toolbars can move the position of toolbars to any position in the display screen. Hence they need not any more remain as a hindrance to the valuable data area on the screen (Principle-15: Dynamize).

Use a three dimensional interface to display more items in less screen space (Principle-17: Another dimension).

Expand the effective size of the display by utilizing the space beyond the edge of the display (Principle-37: Expansion).

Use different methods of displaying windows such as “cascaded”, “timed”, “locked”, “timed icon” etc. to prevent blocking the background data and objects (Principle-35: Parameter change).

Use semi-transparent windows and “see-through” objects to ensure visibility of background data (Principle-32: Color change).

1.2 Different aspects of maximizing information presentation

It is necessary to display more information in the limited display screen. The meaning of displaying more information can be better explained as follows.

✦ Maximizing number and size of windows in the limited screen space (e.g., overlapping or cascading windows).

✦ Maximizing display of buttons, toolbars, menus, indicators and other GUI elements in the limited space of a window or graphical user interface.

✦ Presenting maximum number of buttons in the limited space of a toolbar (e.g., by using drop down toolbars), presenting maximum number of items in the limited space of menu (e.g., by using drop down menu, cascading menu etc.).

✦ Maximizing help text and button captions in the limited space of a button or icon.

✦ Maximizing presentation of actual data (such as, an image or document etc.) in the available screen space (e.g., by reducing the space for GUI control elements, by using scroll bars etc.)

1.3 Various methods addressing screen space problem

Using icons for windows- one technique is used to reduce the windows to icons or small pictures reminding the user of the contents of the window. The icon takes less space but still informs about the contents.
The desktop metaphor - the allocation of screen space can be switched from one display system object to another by shrinking the windows of one task to icons and expanding the icons of another task to windows.

Large virtual workspace metaphor - in this type of interface the screen is thought as a movable viewport on the large virtual workspace. Typically all objects and data are arranged in three screens, one for an overview of the whole space, one for a detailed view of some portion of the space, and one touch-screen for control. The user can move from overview to detailed view and vice versa. But one of the difficulties of this system is that the user needs help navigating.

Multiple virtual workspaces - this technique allows more convenient access to non-displayed display objects by switching quickly from one virtual workspace to another. Like large virtual workspace, the user needs help in navigating multiple virtual workspaces.

Overlapping windows - that creates virtual display spaces. The user can select the required window to display the contents. There may be different methods to manage the overlapping windows to reduce disruption of background objects.

Sharing windows - the method of sharing the windows and icons can also reduce the cluttering in a task-switching environment.

Increasing dimensions - another method is to increase dimensions of display, for example using 3d or 4d visual space for presenting graphic elements and data.

Using scrollbars - when the data or document being displayed within the window is larger than the size of the window, some data remains hidden beyond the boundary of the window. A scrolling mechanism can selectively move the hidden portion of data inside the display area.

2. Inventions on maximizing information presentation

2.1 User interface with multiple workspaces for sharing display system objects (5072412)

Background problem
The relatively small size of a display screen is one of the major constraints of a graphical user interface as it limits the number of perceptible objects, which can be displayed to the user at any given time. There are some solutions like large virtual workspaces and multiple virtual workspaces to overcome this problem. But in these situations, the user typically needs help in navigation. There is a need for an improved method of using multiple workspaces.
Solution provided by the invention

Henderson, et al. invented a user interface (US Patent 5072412, Assignee Xerox Corporation, Issued Dec 1991) with multiple workspaces for sharing the display system objects. According to this method the same window will provide different display characteristics to different workspaces. The user can switch between workspaces by using “door” and “back door”.

![Diagram](FIG. 1A)

The objects in the window remain active even after the user leaves the workspace. When the user returns to the workspace it will have the same contents as when it disappeared.

The invention provides an overview display, which shows a representation of each workspace so that the user can navigate to any workspace from the overview.

TRIZ based analysis

The same window is shared by multiple workspaces (Principle-6: Universality).

When the user leaves the workspace the content of the window disappears and when the user returns to the workspace the content of the window reappears (Principle-34: Discard and recover).

The invention provides an overview display for workspace selection and easy navigation (Principle-8: Counterweight).

2.2 Four dimensional Graphical User Interface (5678015)

Background problem

In a typical graphical user interface, the icons are used to represent various functions or applications. When the user adds more and more icons to the desktop (or other interfaces) there is a shortage of space on the available screen. How to display more icons in the limited available screen space?
Solution provided by the invention

Eng Lim Goh invented a four dimensional interface (Patent 5678015, assigned by Silicon Graphics Inc, Oct 97) which solves the above problem. According to the invention the interface is displayed like rotating cube, which can display upto six workspaces simultaneously. The workspaces are translucent so that all six workspaces are visible to the user simultaneously. It is a combination of a three dimensional figure with a real time rotation that gives the four dimensional features.

The advantage of the present invention is that it permits a user to view a larger effective workspace than the conventional 2 dimensional windows.

TRIZ based analysis

The graphical user interface should accommodate and display as many number of icons as required (Ideal Final Result).

Solution-1: In some cases the scrollbars are used to give larger workspace than the window (Principle-15: Dynamize). The disadvantage here is that only a portion of the workspace is available at any given time.

Solution-2: Another strategy is to provide multiple windows (Principle-1: Segmentation) each containing a portion of the total set of icons. But because the windows must overlap, the user can see only a portion of the total set of icons at any given time.

Solution-3: Display the icons in a three/ four dimensional Graphic User Interface (Principle-17: Another Dimension).

This invention displays the icons in a three dimensional cube (Principle-17: Another dimension).
The interface is translucent so that the icons on the rear side are also visible (Principle-32: Color Change).

The cube is capable of rotating that makes any icon easily accessible by the user (Principle-15: Dynamize).

2.3 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).

2.4 Sliding out interface (US Patent 5914716)

Background problem
The recent application programs are providing hundreds of command buttons which are difficult to accommodate within the limited screen space. There are several solutions to display more icons in the limited screen space. One is by reducing the icon size, but this still occupies permanent screen. Another method is by grouping the related commands into folder icons, but this creates the difficulty of exploring through clicking on folder icons. There is a need for effectively displaying more buttons in less screen space.

Solution provided by the invention
Rubin et al. disclosed a sliding out interface bar (Patent 5914716, assigned by Microsoft corporation, Jun 99), which conserves valuable screen space. The graphical user interface displays a target image representing a sliding out command bar containing a set of selectable computer resources. When the user moves the display pointer to a location near the target image, the tray slides out and the command bar remains visible. When the pointer is moved away from the image, the command bar slides in and becomes invisible.

TRIZ based analysis
We want hundreds of icons to be displayed on the screen. But we don’t want them to occupy valuable screen space (Contradiction).

One solution is to reduce the size of icons which will effectively occupy less screen space (Principle-35: Parameter change). But this has two disadvantages. Firstly, small icons will suffer from the problem of visibility and identification. Secondly, the icons, even though smaller, still occupy permanent real estate.
Another solution is to group the related commands and keep them inside folder icons (Principle-5: Merging). This suffers from the problem of exploration as the user has to first click on the folder icon in order to explore the individual buttons.

This invention discloses a sliding interface which slides out when the user moves the pointer near to it and slides in when the pointer is moved away (Principle-15: Dynamize, Principle-34: Discard and recover).

2.5 Expandable and collapsible options in GUI (US Patent 5986657)

**Background problem**
A typical GUI is full of various visual tools, such as toolbars, toolboxes, menus, buttons and so on. In case of large applications it is sometimes very difficult to accommodate so many of tools within the limited available screen space. Besides displaying so many of tools occupies a lot of screen space and leaves less space for the main worksheet. There is a need to provide a more compact presentation of groups of functions in the GUI.

**Solution provided by the invention**
Berteig et al. disclosed a method of using expandable and collapsible sub-panels (Patent 5986657, Assigned by Autodesk Inc, Nov 99) to solve the above problem. According to the invention, each sub-panel may be toggled between an expanded state and a collapsed state. The user can change the appearance and functionality of the GUI by expanding and collapsing the sub-panels, but the total size of the screen used by sub-panels remain the same.
TRIZ based analysis

The invention provides expandable and collapsible sub panels. The user can expand any sub-panel containing the required functions. The expansion of one sub-panel will collapse another to keep the total screen consumption the same (Principle-15: Dynamize).

2.6 Method of displaying multiple windows (US Patent 6025841)

Background problem

In a typical windows environment, there are several windows displayed on the screen. The user clicks on the visible part of a window to bring it to the front. Thus the user has to click on different windows in order to see their contents.

There is a problem in this mechanism. When the size of the window is large, it covers other small windows. If a window is totally hidden behind another window then it becomes difficult to select that window, as the user cannot find any part of the covered window to click. There is a need to display multiple windows in a more stable and persistent fashion so that the important windows do not get covered by other windows.

Solution provided by the invention

Finkelstein et al. found a method of managing simultaneous display of multiple windows (Patent 6025841, Assigned by Microsoft, Feb 2000). The invention provides a continuous, automatic adjustment to window size, and position of a selected top most window. The invention provides three different methods to keep the target window from obscuring the content of an underlying window.

1. Move away- when the window is selected to be on the top, it is moved away to a location where there is more free screen.
2. Disappear- the target window is disappeared momentarily and reappears once again when the action (say dragging an object) is completed.
3. Reduce- the system reduces the size of the target window to smaller, which is less obscuring.
TRIZ based analysis

When the target window is hiding other useful contents of the screen (at the input of a trigger through mouse or keyboard) the system calculates a new location for the target window and redraws the window in that position (Principle-15: Dynamize).

The system displays the selected window typically in a higher z-order so that it is displayed on the top (Principle-17: Another dimension).

The windows are adjusted automatically to display the useful contents during operation based on the interaction of the user, without the user explicitly adjusting window size or position etc. (Principle-25: Self service).

The system uses three different methods to keep a target window from obscuring the contents of an underlying window, viz., Move away (Principle-15: Dynamize), Disappear (Principle-34: Discard and recover), and Reduce (Principle-35: Change parameter).

2.7 Graphical user interface with optimal transparency thresholds for maximizing user performance and system efficiency (US Patent 6118427)

Background problem
A typical problem with the GUls is to efficiently displaying a number of objects (windows, menu, toolbar etc.) within a limited area of display. The more screen space occupied by these GUI elements, the less screen space is left for the actual information.

The conventional methods of “space division” or “time division” of the screen space have some drawback, i.e., all objects behind a fully opaque window or menu are not visible to the user.

In some cases a semi-transparent or translucent GUI has solved the space problem to some extent but has created visibility problem.

Solution provided by the invention
Buxton et al. (Patent 6118427, assigned by Silicon Graphics Inc, issued in Sep 2000) provides a solution that maximizes user performance and system efficiency. The invention utilizes variable-transparency to merge layers of images on the display. For example, “see through” objects such as menus, tool palettes, windows, dialog boxes or screens are superimposed over similar objects. The variation in the transparency is determined by user selection error rates and user response time.
The invention provides hybrid windows having both variably transparent regions and opaque regions. The system efficiency is given priority than visual enhancement.

**TRIZ based analysis**

The semi-transparent (or translucent) object images solve the problem of screen space by allowing the background object images to be visible. But while solving one problem they create another problem, i.e., reduced visibility. The degree of visual interference is generally a function of transparency. The higher the transparency of the foreground object, the higher the severity of the visual interference (Contradiction).

If the degree of transparency is pre-determined it may not be efficient. On the other hand if the transparency level is user configurable then the user is additionally burdened with adjusting this difficult task of adjusting the transparency level (Contradiction).

The invention displays variable-transparent layers of images one on top of another (Principle-7: Nested doll).

The transparency of the images varies according to user response time and selection errors (Principle-15: Dynamize).

The invention provides hybrid windows having both variably-transparent regions and opaque regions (Principle-40: Composite).
2.8 Graphical user interface (US Patent 6353436)

Background problem
When the user works with various applications simultaneously or working with large documents or images it is necessary to have a large display screen to view them properly. Although the concept of windows and virtual workspaces solve this problem to some extent they need excessive use of mouse movements to select the desired objects on the screen.

Solution provided by the invention
Patent 6353436 (invented by Reichlen, assigned by Sun Microsystems Inc., issued in Mar 2002) discloses a method of rapidly navigating through windows and performing GUI operations without requiring use of a keyboard or mouse. The invention uses a video display system which includes a swiveling chair, a computer mounted at the base of the chair, a platform for supporting keyboard and mouse, a head mounted display and a position sensor, including a transmitter mechanically coupled to the head mounted display and a receiver mechanically connected to a stationary reference point. The reference point can be located above the user’s head, at the base of the chair or any other place nearby.

The virtual view space is the total image area in the video display system. The virtual view space is 360 degree and has a height of 135 degree. The virtual view is shaped like a “cylinder” which surrounds the user. The view space includes 96 million discrete points, 16 thousand in horizontal and 6 thousand in vertical, each identified by a yaw and pitch location. During operation the user may navigate the virtual view space moving and rotating his head.

TRIZ based analysis
The invention expands the display area by using virtual view space (Principle-36: Expansion, Principle-26: Copying).

The virtual view space is shaped like a cylinder that surrounds the user (Principle-14: Curve).
2.9 Maximizing information presentation on the screen (US Patent 6512529)

Background problem
It is always a problem to display the contents of many windows in the limited screen space. There are different ways of displaying windows on a screen, such as, “tiled” where the contents are totally visible but the window size is small, “cascaded” where the window size is larger but the contents are partially overlayed by other windows. There is a need for a method to display multiple windows without obscuring the main window.

Solution provided by the invention
Janssen et al. invented a method (Patent 6512529, Assigned by Gallium Software, Jan 2003) for maximizing information presentation on a screen. The invention prescribes 4 different display modes. In “normal mode”, the contents of the windows are exposed. In “timed mode” the contents are displayed for a specified period of time. In “locked mode” the contents of the window are displayed in an opaque manner. In the “timed icon mode”, the contents of the window are exposed for a specified period of time after which the window is automatically reduced to an icon.

According to the invention the windows are made invisible so that the information on the background window is not obscured. The invention consists of a user interface that provides a rapid means of displaying and hiding information in invisible windows.

TRIZ based analysis
The invention accommodates more icons and windows, which are not displayed in the normal window, rather displayed in special modes and invisible trays (Principle-17: Another dimension).

In “timed mode” the window returns to its invisible state after a specified period of time (Principle-32: Color change).

The user can change the status of any window as “normal”, “timed”, “locked” or “timed icon” to change the pattern of their visibility (Principle-15: Dynamize, Principle-35: Parameter change).
3. Summary

A graphical user interface consists of various data and control elements. As the physical size of the display device is difficult to expand, it is necessary to display all these information within the limited size of the display screen. There are many inventions on how to maximize the presentation of information on a graphical user interface. Some of the interesting methods are as follows.

⇒ Sharing the same window for multiple workspaces can help displaying the contents of multiple workspaces in the same physical location (although it can display only one workspace at a time).

⇒ Using scrollbars to scroll a larger workspace than the size of the window or even larger than the size of the display screen.

⇒ Using a three dimensional interface to display more icons. A cube has six sides and can display much more icons than a two dimensional interface.

⇒ Expanding the effective size of the display by utilizing the space beyond the edge of the display.

⇒ Grouping a number of command buttons inside a single folder button.

⇒ Using a sliding interface to slide out and display graphical objects when the user moves the pointer near to it.

⇒ Using expandable and collapsible sub-panels, where expansion of one sub-panel collapses another sub-panel to keep the total size fixed.

⇒ Adjusting the windows automatically to display the useful contents during the operation based on the interaction of the user.

⇒ Using different methods to keep a target window from obscuring the contents of underlying windows.

⇒ Displaying windows in different modes such as “normal”, “cascaded”, “timed”, “locked”, “timed icon” etc. for effective viewing of data.

⇒ Using “see through” objects such as transparent menus, tool palettes, windows, dialog boxes etc., which do not hide the visibility of others.
⇒ Using a variable transparency to build the “see through” objects, as a pre-determined degree of transparency may not be efficient in different situations.

⇒ Expanding the view screen by using projectors. Alternatively using virtual view space by using head mounts.

During the course of analysis we tried to find the Ideal Final Result or the desired result. We also find the contradictions involved in achieving the desired result and the how the Inventive Principles are applied to solve the contradictions in order to achieve the desired result.

Reference to patents:


Other references:


12. Umakant Mishra, 10 Inventions on Scrolling and Scrollbars in Graphical User Interface. (December 6, 2006), Available at SSRN: http://ssrn.com/abstract=949243


