Inventions on user friendliness of a GUI- A TRIZ based analysis

Umakant Mishra
Inventions on User-Friendliness of a GUI
A TRIZ based analysis

Umakant Mishra, Bangalore, India
http://umakantm.blogspot.in

Contents

1. Introduction .................................................................................................................. 1
  1.1 Ideal Final Result .................................................................................................... 2

2. Inventions on increasing user friendliness ................................................................. 3
  2.1 Status indicators of an improved graphical user interface ........................................ 3
  2.2 Intelligent window user interface for computers ..................................................... 4
  2.3 Method and system for efficient organization of selectable elements on a graphical user interface ......................................................................................... 5
  2.4 Graphical user interface system and methods for improved user feedback .......... 6
  2.5 Method and system in a graphical user interface for facilitating cursor object movement for physically challenged computer users ........................................... 7
  2.6 Providing a preview capability to a graphical user interface dialog ..................... 9

3. Summary ...................................................................................................................... 10

Reference to patents ........................................................................................................ 10
Other references .............................................................................................................. 11

1. Introduction

A GUI is a revolutionary development in the evolution of man-machine interfacing technology. It facilitates the user in performing complex operations by merely clicking on visual buttons or dragging and dropping visual elements.

Although a GUI is considered to be convenient it still has certain shortcomings like ambiguity of pictorial symbols, difficulty of cursor movements, difficulty of interacting with smaller graphic elements, likeliness of making mistakes by slip of fingers, demand of more screen space, difficulty of searching in nested containers etc. Some of these problems are discussed in separate individual articles. This article deals with increasing user friendliness.
Conventionally many users fail to operate a computer satisfactorily because of the special skill or training required to operate it. In order to avoid this problem, it is necessary to design the user interface in such a way that an untrained user should also be able to operate the computer effectively and efficiently. In other words, the GUI should assist the user in operating the computer. The user friendliness is very important especially for the operators having less training, less experience, less intelligence, less patience, less vision, less hand movements and other challenges or constraints.

1.1 Ideal Final Result

Ideally the user should require no training before operating a GUI and require no assistance while operating a GUI (Ideal Final Result).

The user need not search for the desired graphic element. The desired graphic element should automatically be searched for the user (Ideal Final Result).

The user need not move the cursor to the desired icon or other GUI element. The cursor should automatically move onto the desired icon or other GUI element (Ideal Final Result).

Although it is difficult to achieve the Ideal Final Results, we can draw a list of desirable features of a GUI moving towards the Ideal Final Results.

› The graphic elements should be self-explanatory about their meanings, purpose and method of interaction.

› The graphic elements should be easily accessible. In other words, the user need not navigate much to search the desired graphic elements.

› The graphic elements should be easily visible even to the users with poor vision. (The inventions on visibility related assistive features are analyzed in a separate article.)

› The GUI should provide ALL TYPES OF INFORMATION about ALL TYPES OF GUI ELEMENTS.

› The cursor should automatically move onto the desired icon or other GUI element. (The inventions on minimizing cursor movements are analyzed in a separate article.)

› The GUI should guide the user as and when he requires assistance during his course of interaction with the computer.
2. Inventions on increasing user-friendliness

2.1 Status indicators of an improved graphical user interface (US Patent 5644334)

Background problem
While developing software programs, a programmer must keep track of various state attributes for different objects. Typically these attributes are represented as bits of state associated with the object. In some graphical user interface systems, these attributes are available from menus accessible from a window. But opening the menus to examine the states is a time consuming and inefficient process. Hence there is a need for an improved graphical user interface having capability of positionally displaying state attributes for more than one object at a time.

Solution provided by the invention
Patent 5644334 (invented by Jones et al., assigned by apple Computer Inc., issued in July 1997) discloses the concept of status indicators, which can display object attributes on a window panel. The attributes can be like whether the source code of an object has been saved or compiled, whether the protection status is read only etc.

According to the invention each type of indicator has a different color and a different location along the width of the panel for ease of visibility and identification.

TRIZ based analysis
Ideally the user need not do anything to know the status of various objects. All the objects themselves should reveal their status to the user (Ideal Final Result).

The invention displays different types of status through status indicators. Different status indicators are distinguished by different colors, shapes and spatial positions (Principle-35: Parameter change).

The invention provides a customizable browser framework to select or deselect any of the available status indicators on the list (Principle-15: Dynamize).
2.2 Intelligent window user interface for computers (US Patent 5737557)

Background problem
The Windows 95 and Macintosh GUI provide visual representation of files and folders. However the groups of files comprise merely the collection of individual files, but the collection has no knowledge of files comprising the collection. For example, the program manager does not maintain contextual information about the items in the collection.

Under this circumstance, when the user deletes a program icon the files remain resident on the storage media, although the user may believe that the files have been removed. Similarly, when a program is de-installed, all visual clues for the program are removed from the GUI and the user must independently keep track of all de-installed programs.

There is a need for an intelligent user interface that can maintain the contextual information regarding various application programs and keep track of the de-installed programs.

Solution provided by the invention
Gary Sullivan invented a compound display object (Patent 5737557, Assigned by AST Research Inc, Apr 98) that provides a single integrated visual representation via a “software suite window”. The interface facilitates execution of various operations that applies simultaneously to the collective properties of all the items in the software suite.

The software suite is associated with the installation-de-installation information. When a file is de-installed, the user may choose to change the appearance of the icon to be de-emphasized or “grayed” or deleted altogether. This integrated management of the software suite helps the user knowing the status of each individual application in the suite.
TRIZ based analysis
The GUI should keep track of files associated with programs and inform user about the status and history of installed and de-installed programs (IFR).

The present invention provides a single integrated visual representation of multiple application programs via a “software suite window” (Principle-40: Composite).

The de-installed files/applications of the software suite are displayed differently (such as grayed or de-emphasized) to indicate about their de-installed status (Principle-32: Color change).

2.3 Method and system for efficient organization of selectable elements on a graphical user interface (US Patent 6005567)

Background problem
If the selectable elements on a GUI are placed in difficult positions the user takes long time and faces inconvenience in selecting the elements. Hence, typically the most frequently used elements are located in an easily accessible area for faster and convenient access.

But the conventional GUIs are static, so they do not change according to the usage pattern of the user. Secondly, the usage patterns of users of different skill levels are different. There is a need for automatically organizing the selectable elements on a GUI according to changes in the user need.

Solution provided by the invention
The invention keeps a statistics of access frequency of the elements, and organizes the elements depending on their access frequency. The elements of higher access frequency are placed in a first area and the elements of a lower access frequency are placed in a second area.

The user can select the selectable elements on a GUI easier when they are organized according to the principles of this invention. (This patent was extended further in patent 6069625 issued later in May 2000).

**TRIZ based analysis**

The GUI elements should be easily accessible by the user (desired result).

The method puts the more frequently used elements in a location that is easily accessible so that the user can search those elements conveniently in less time and effort (Principle-10: Prior action).

The invention moves the elements dynamically from first area to second area (or vice versa) according to their frequency of access to cope with the changing behavior of the user (Principle-15: Dynamize).

### 2.4 Graphical user interface system and methods for improved user feedback (US Patent 6005570)

**Background problem**

Typically a GUI has several icons/buttons, which are internally connected to specific functions. When the user clicks on an icon, the system performs the associated function. When the number of buttons or icons increases it is not possible to display the name of the function on the button itself. A smart icon provides a solution to this problem. When a user moves the cursor on to a smart icon, it displays balloon help and informs its function.

Although the approach of smart icon and balloon help is good, it has a major drawback that it obscures the screen by popping up a message at a screen location that is proximate to the cursor location. Secondly, the balloons are resource intensive to frequently repaint the user image.

**Solution provided by the invention**

Patent 6005570 (invented by Gayraud et al., assigned by Inprise Corporation, issued in Dec 1999) discloses an improved method of displaying icon descriptors without obscuring the active screen. According to the present invention, the “hints” for the icons are displayed along one side of the screen (preferably at the bottom of the screen), which does not block any valuable portion of the screen.
When the cursor enters into the boundary of an object, a corresponding descriptor or “hint” is displayed at a status frame of the window, and when the cursor moves out of the boundary of that object the descriptor is cleared.

**TRIZ based analysis**

The icons themselves should explain their functions (Desired result).

The icons should be small in size to occupy less space on the display screen but should provide textual descriptions about their functions (Contradiction).

The balloon help can describe their functions in a pop-up display (Principle-17: Another dimension) when the user moves cursor onto the icon (Principle-15: Dynamize).

Moving of cursor on to the icon should describe the function of the icon but the pop-up display should not obscure the valuable workspace (Contradiction).

The invention isolates the descriptor display area from the main working area and displays the descriptions in a static frame at the bottom of the screen, which do not obscure the valuable screen items (Principle-2: Taking out).

Similar to balloon help, the invention also displays the descriptor when the cursor moves on to the icon and clears the descriptor when the cursor moves out of the icon (Principle-15: Dynamize).

**2.5 Method and system in a graphical user interface for facilitating cursor object movement for physically challenged computer users (US Patent 6031531)**

**Background problem**

Most features of the graphical user interfaces are designed keeping the skills and requirements of the common people in mind. But the physically challenged computer users often face difficulties in precisely moving the cursor onto specific icons on the screen. It is desirable to include features that would allow physically challenged users to access such graphical objects with greater easiness.
Solution provided by the invention

Patent 6031531 (invented by Kimble, assigned by IBM, Issued in Feb 2000) discloses a method of facilitating cursor movement for physically challenged users. According to the invention each icon will have an icon domain, which is larger than the icon itself. When the cursor enters into an icon domain, the cursor is automatically positioned at the center of that icon.

According to this method, the icons are graphically magnetized such that the cursor object is immediately snapped towards a desired icon when entering a graphically magnetized surrounding of the desired icon.

This method assists physically challenged users in manipulating icons and other graphically displayed objects within a graphical user interface.

TRIZ based analysis

Ideally the cursor itself should automatically move onto the desired icon (Ideal Final Result).

When the physically challenged user fails to put the cursor exactly on the icon the GUI should be able to understand the intention of the user (Desired result).

The icon’s click area can be enlarged so that the cursor can enter into the click area with less effort (Principle-37: Expansion). Interpreting differently, the icon can be sensitive to accept a selection when the user just moves the cursor close to the icon (Principle-16: Partial or excessive action).

According to the invention, the icons are graphically magnetized to pull the cursors entering into the icon domain. This assists the user by eliminating the need for positioning the cursor object precisely on the icon to select that object (Principle-10: Prior Action, Principle-8: Counterweight).
2.6 Providing a preview capability to a graphical user interface dialog (US Patent 6061059)

Background problem
The dialog boxes are used to display specific messages to the user and get the user feedback to the program control. The program does the relevant operation depending on the feedback from the dialog box and the operation is done permanently. It will be nice to have previewable dialogs, which allows the user to see the effects of the dialog control before finally closing the dialog.

Solution provided by the invention
Taylor et al. disclosed a method (Patent 6061059, assigned by Adobe Systems Incorporated, May 2000) of providing a preview capability to dialogs. The previewable dialog will process the commands produced by the dialog’s control to provide a preview capability.

According to the invention, the preview mechanism includes a “do and undo” mechanism kept in a dialog command buffer. The method bundles the commands in the dialog command queue into a single macro command, so that the whole set of commands can be done and undone.

![Image of a dialog box with preview option]

The previewable dialog has the advantage that the developer can just add and remove the preview capability to a dialog by simply adding and removing the preview control. The developer need not be aware of how the preview control is implemented.

TRIZ based analysis
The invention provides a preview before the user finally decides to select an option (Principle-27: Cheap and disposable).

This functionality prevents user from taking poor/improper decisions (Principle-8: Counterweight).

The preview mechanism includes a series of “do and undo” functions related to the specific dialog box operations. The functions are done for preview and undone again to bring the status to normal (Principle-13: Reversing).
3. Summary

The user-friendliness of a GUI is important to motivate and attract the users to operate the computer successfully and more effectively. These features assist even the untrained, inexperienced and physically or mentally challenged user to operate a computer with greater success. They reduce the requirement of user training, reduce user frustration and increase acceptance of the software.

Many inventions on assistive features like “screen zooming features”, “improving navigation”, “tactile interfacing”, “speech and hearing”, “features for visually impaired” are discussed in other articles. The six patents analyzed in this article try to improve the user friendliness of the GUI through the following means.

⇛ Making the meaning and function of graphic elements self-explanatory.
⇛ Informing status of objects through status indicators.
⇛ Maintaining and displaying various types of information on collection of files, interdependent programs, de-installed programs etc.
⇛ Placing more frequently accessed graphic elements on easily accessible locations for easier and faster access.
⇛ Displaying the meaning and function of an icon (or other graphic object) upon moving the cursor onto the icon (or other graphic object).
⇛ Displaying preview features before closing unknown operations.

All these features try to achieve the Ideal Final Results (described above in 1.1) to some extent. Keeping the IFRs in mind there can be many more inventions in the same directions.

Reference to patents:


**Other references:**


Inventions on user friendliness of a GUI, by Umakant Mishra