

Principles of User Interface Design

Common User Access

A user interface is the software application's interface with a user. A graphical user interface (GUI) is a user interface that uses text and images and involves direct manipulation of the individual graphical elements by the user.

Contents

Background	1	<i>Windows</i>	4
Design Principles	2	<i>Views</i>	5
<i>Principles to Put the User in Control</i>	2	<i>Window Organization</i>	5
<i>Principles to Reduce the User's Memory Load</i>	2	<i>Menu Hierarchy</i>	6
<i>Principles that Support Consistency</i>	2	<i>Controls</i>	8
<i>Principles that Create Aesthetic Appeal</i>	3	<i>Navigation</i>	10
Design Concepts	3	<i>Selection</i>	11
Application—Editorial Comments	4	<i>User Information</i>	11
Elements of the CUA User Interface	4		

Background

IBM developed a conceptual framework for the design of graphical user interfaces beginning in 1989. They called it Common User Access or CUA. The third and current version is CUA 91, published in 1991. CUA was documented in two IBM publications:

- *Systems Application Architecture Common User Access Guide to User Interface Design* (SC34-4289-00) and
- *Systems Application Architecture Common User Access Advanced Interface Design Reference* (SC34-4290-00).

These two documents presented design principles to underlie the graphical user interface, specified the various elements of a GUI, and described the uses and interoperability of those elements. CUA became the basis for the operating systems like Windows, OS/2, and Mac OS.

At the time CUA was developed, user interfaces were limited to a command-line interface, IBM 3270 terminals, and early microprocessor operating systems like MS-DOS. A command-line interface is text-based where commands are typed on the keyboard. 3270 terminals provided a “thin client” to a mainframe application “server,” to use modern terminology, and displayed text only in 24 rows of 80 columns each; the text was uppercase in one font, with some limited variation in color, and the screen resolution was coarse. Each MS-DOS application had a totally different user interface. CUA was intended to rationalize and harmonize all user interfaces, from mainframes to microprocessors. CUA 91 was adopted by Microsoft in the 1995 version of Windows.

The CUA user interface is “graphical,” including the use of a mouse (generally referred to as a pointing device), and incorporates elements of object orientation. Its developers claimed that “a user can focus more closely on the task at hand without having to be so conscious of the tools involved.”

When reading these two documents it is sometimes difficult to separate marketing hype from semantics from a truly different paradigm. However, the idea that applications become invisible and that users be able to perform the same actions in the same way on any object, regardless if it is a purchase order or memo or spreadsheet, is a useful one. After all, how many unique applications can we master at the same time?

Principles of User Interface Design

Common User Access

CUA focuses on the quality and effectiveness of the user's operation of an application. While this had been a design issue for 3270-based applications all along, the possibilities for workstation-based* applications greatly increased the effort required to design such a user interface.

CUA extends the concept of ergonomic into the mental realm. Not only must an application be easy to operate manually, but it must also be easy to comprehend. As the operation of a computer should be as much as possible a natural extension of a person's manual dexterity, so too should the organization and meaning of the application be a natural extension of the person's conceptual model.

I believe that Common User Access is as valuable today as when it was conceived.

Design Principles

CUA is based on principles about the nature of the user's control, the quantity and quality of what the user has to memorize, the consistency of operations and appearance, and the satisfaction and acceptance to be gained from an aesthetic appearance.

Principles to Put the User in Control

- Enable the user's control by providing alternative actions in addition to not imposing a fixed sequence of actions.
- Incorporate messages that describe the condition objectively, are clear and descriptive yet concise, and provide a way for the user to correct the error.
- Provide immediate results of user actions, and feedback if that is not possible.
- Provide a way to reverse the user's actions or a warning message when an action cannot be reversed.
- Accommodate users with different business and technical skill levels: Unskilled users need direction while skilled users need shortcuts.
- Accommodate users with differing manual dexterity by providing both mouse and keyboard access to all screen elements. (The mouse is not universally ergonomic. The level of hand and eye coordination it requires is unnatural, even unachievable, for some people.)
- Enable the user to customize their interface by providing mechanisms to change the volume and duration of sounds, the intensity and hue of colors, the arrangement of choices on menus, the sequence of steps in a process, etc.

Principles to Reduce the User's Memory Load

- Provide alternatives from which a user can choose: lists of items, list of actions, etc.
- Provide visual signals to remind the user of the task in progress: emphasis, messages, progress indicators, etc.
- Provide default settings and a mechanism to save changed settings.
- Make objects concrete and recognizable.
- Incorporate visual metaphors when possible.

Principles that Support Consistency

- Provide for identical implementation of common functions throughout the application.

* Modern workstations have monitors capable of displaying upper and lower case text, images, lots of colors, all in fine resolution.

Principles of User Interface Design

Common User Access

- Provide visual signals to establish continuity and points of reference.
- Enable the user to predict the result of an action by providing consistent responses and labeling actions with appropriate terms.
- Maintain continuity between applications and between versions of the same applications.

Principles that Create Aesthetic Appeal

- Make screen layouts consistent with each other.
- Arrange the screen elements in a way that is sensible and logical and consistent with the user's conceptual model.
- Incorporate into the composition of the screen the elements of graphic design which contribute to the application's attractiveness: white space, color, overlap, size and shape differences, pattern, contrast, balance, etc.

Design Concepts

The following terms describe concepts that are central to CUA. They are listed here in alphabetical order.

Choice: In a window, the text or image that a user selects to modify or manipulate the corresponding object. CUA identifies three kinds of choices: An *action choice* immediately performs some task, e.g., Print. A *routing choice* displays a menu or a secondary window. A *settings choice* allows the user to display or change the characteristics or properties of the corresponding object.

Control: In a window, a mechanism used to interact with data—enter it, display it, and change it—or provide access to an action or routing choice. Some of the many kinds of controls are menu, list box, and push button.

Interface orientation: A user interface may be object-action oriented or it may be action-object oriented. The former refers to an approach where the user first selects an object, then selects an action that pertains to that object. The latter refers to an approach where the user first selects an action, then selects the object of that action.

Metaphor: An object used in place of another to suggest a likeness or analogy between them. More broadly: figurative visual language.

Model: A descriptive representation of a person's conceptual and operational understanding of something. There are three kinds of models of user interfaces: the user's conceptual model, the designer's model, and the programmer's model.

Object: An object is an item used to perform a task. In the CUA environment, there are three classes of objects: container objects, data objects, and device objects. Objects can be grouped, and groups can be organized hierarchically.

Transparent: An interface that does not distract the user from the task at hand.

Principles of User Interface Design

Common User Access

Application—Editorial Comments

The fifteen years following the explication of CUA have seen the mass adoption of computers, monitors capable of high resolution of text and images, and the internet. The vocabulary CUA introduced is now well known to many. But the design principles have had an uneven application.

The crux of the design problem is reflecting the user's conceptual model. That model is the result of the confluence of the user's needs, their experience, and the identity, the essence, of the object. This is complicated by two things:

1. Recently science realized that for some scientific research the mere existence of the observer influences the observed object. Similarly the design of a user interface influences the user's conceptual model. The user is both observer and operator. Their needs may not change, but their understanding of the identity of the object can change. Whether this change is for the better or for the worse is irrelevant.
2. Each user may have a different conceptual model.

It is not enough for a user interface to use labels the user can understand correctly and immediately. It must also decompose the object and use methods and properties that match and extend the user's model. And it must accommodate the inevitable evolution of the user's model. This is where the principle about accommodating users with different business and technical skills comes into play.

A famous architect is often quoted as saying "form follows function." In UI design form follows both function—what the user needs to do with the object—and the innate structure of the object. A deep understanding of the object and the uses to which it will be put is essential.

Know your audience, lead don't follow, look to the future.

Elements of the CUA User Interface

The key elements of a GUI are windows, pointers and cursors, views, menus, choices and controls, navigation, the selection process, and user information. These are described in the following sections as a combination of specification and guidelines. This represents a standardized vocabulary critical to effective communication among designers, developers, and users.

Windows

A window is an area with visible boundaries within which information is displayed. Windows are used to present objects, action and routing choices, and messages and to conduct a dialog with the user.

CUA 91 identifies two types of windows:

- A **primary window** is the principal focus of the user's activity. The primary window remains for the duration of the subject program, although the views may change as the user's dialog progresses.
- A **secondary window** contains auxiliary objects and action choices.

Window elements can be divided into two groups. In one group are the elements that relate to the window as a whole: border and title bar, the latter comprising the system menu icon, window title,

Principles of User Interface Design

Common User Access

and window sizing (min-max) buttons. In the second group are the elements that relate to the subject of the window: menu bar, view, and message.

Views

A view is the contents of a window—one way of looking at an object's information. A view comprises text and/or images and lines that portray information about the subject of the window. When the amount of data exceeds the space available, scroll bars—horizontal and/or vertical—are used. A view may also contain action and routing choices.

CUA identifies four kinds of views:

- A **composed view** utilizes an arrangement of data which conveys meaning, like a graph.
- A **contents view** is a list of the object's components, as either icons or details.
- A **settings view** displays the object's characteristics, attributes, or properties and provides mechanisms by which they can be changed.
- A **help view** displays information that can assist the user in working with an object.

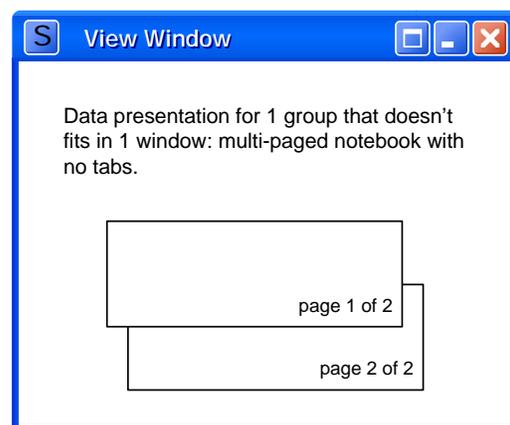
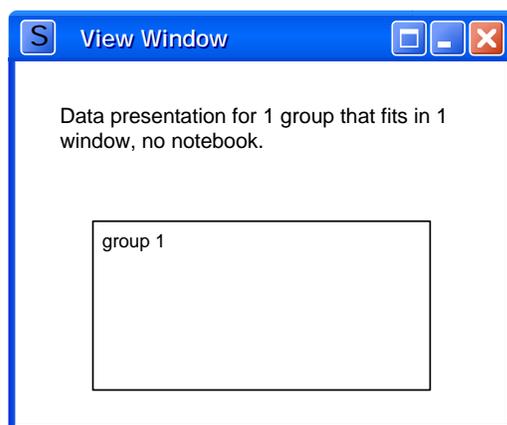
Window Organization

Windows have a parent-child structure. Primary windows can have one or more secondary windows. Secondary windows can be nested.

A window and its content are not necessarily permanently tied together. The user can completely change the content of a window by selecting a different view or by changing the window's size. However, a box used for a dialog or to present a message is fixed with its content.

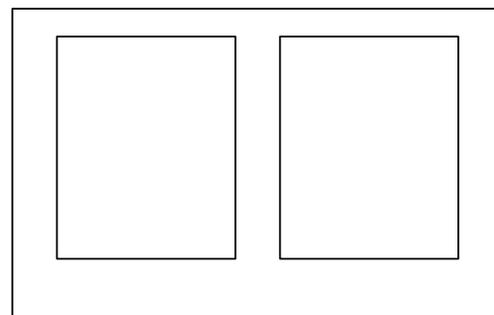
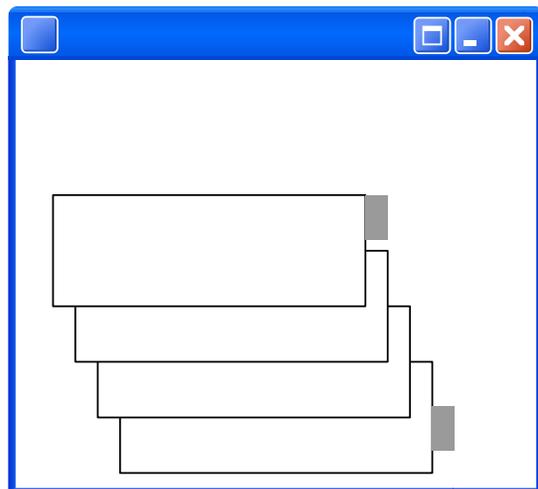
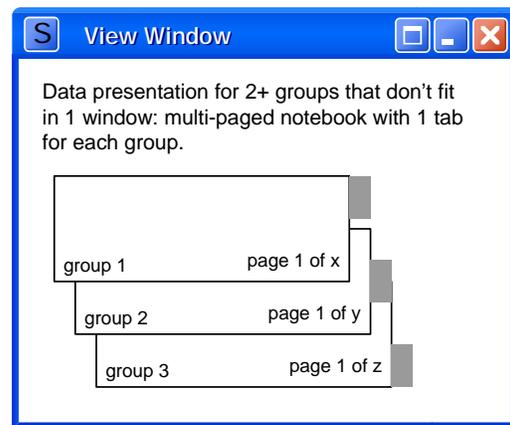
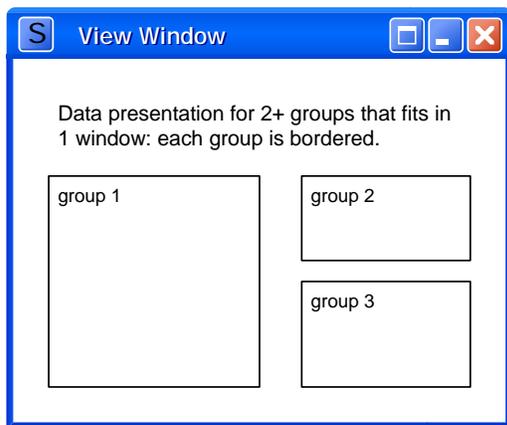
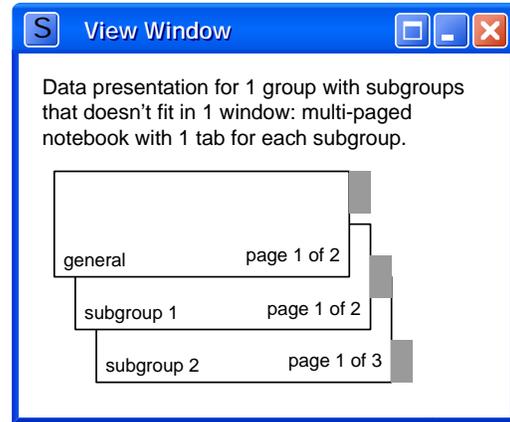
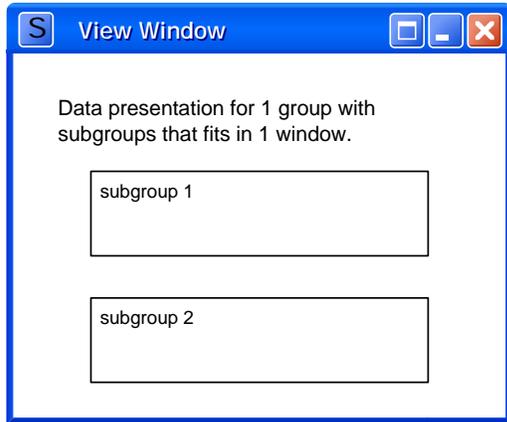
The diagrams that follow illustrate how data about an object can be presented within a window. Different conditions related to the structure and volume of the data are addressed:

- The data exists as one group and fits within one window.
- The data exists as one group and doesn't fit within one window.
- The data exists as one group with subgroups and fits within one window.
- The data exists as one group with subgroups and exceeds the capacity of one window.
- The data exists as two or more groups and fits within one window.
- The data exists as two or more groups and exceeds the capacity of one window.
- The data exists as two or more groups, some of which can be subdivided, and exceeds the capacity of one window.



Principles of User Interface Design

Common User Access



Menu Hierarchy

Menu items exist in a hierarchy that is reflected as a cascade as each is selected.

A three-level menu hierarchy is employed to present choices that can be applied to an object. The menu bar—the second line of a primary and, optionally, secondary window—is the top level. There are seven common menu bar choices:

Principles of User Interface Design

Common User Access

File, Edit, View, Selected, Options, Windows, Help

In addition there is a system menu that contains window-specific choices. It is accessed through the system menu icon in the upper left corner of a window.

There are standardized menu choices for the pull-down menus (second level) that are associated with each menu bar choice. A perusal of these will give an idea of the approach that CUA offers to applications.

File	Edit	View	Selected	Options	Window	Help
New	Copy	Include	Print		<window list>	General Help
Open	Create	Refresh				Using Help
Open As	Cut	Refresh Now				Tutorial
Print	Delete	Sort				Product
Save	Find					Information
Save As	Paste					
	Select All					
	Deselect All					
	Undo					
	Redo					

<i>Second-level Menu Choice</i>	<i>Menu</i>	<i>Action</i>
Close	System	Removes a window and all its subsidiary windows.
Copy	Edit	Copies a selected object onto the Clipboard.
Create	Edit	Creates a new object on the Clipboard.
Cut	Edit	Removes a selected object and places it on the Clipboard.
Delete	Edit	Removes a selected object or group of objects.
Find	Edit	Displays a window that allows the user to search an object displayed in the window in which Find was selected. Search criteria can be specified.
General Help	Help	Displays a window containing a brief overview of the function of the window in which General Help was selected.
Help Index	Help	Displays an alphabetic listing of all help topics within the application.
Hide	System	Removes the window and all associated windows from the Desktop. The window is placed in a window list.
Include	View	Displays a window that allows the user to specify characteristics of a subset of objects so that only the subset is displayed in the window.
Maximize	System	Enlarges a window to the largest size possible for that view or to the size of the Desktop.
Minimize	System	Removes a window and all its subsidiary windows from the Desktop and replaces it by a minimized-window icon.
Move	System	Allows the user to move the window to a different location.
New	File	Allows the user to create a new object.

Principles of User Interface Design

Common User Access

<i>Second-level Menu Choice</i>	<i>Menu</i>	<i>Action</i>
Open	File	Displays a window that allows the user to specify an object to be opened in the window from which Open was selected.
Open As	File	Displays a window containing the named view or, when several views are available, displays a cascaded menu containing the names of the view choices.
Paste	Edit	Places a copy of the contents of the Clipboard into an indicated object (may be a different context).
Print	File, Selected	Causes an object to be printed.
Product Information	Help	Displays a window containing product information, such as the product name, copyright notice, logo, version number and date, and developer names.
Refresh (Now)	View	Causes the current view to be changed to match the underlying data either immediately or at a later time.
Restore	System	Returns a window to the size and position it had previously before it was minimized or maximized.
Save	File	Stores the object in the window on a data storage device, such as a disk.
Save As	File	Displays a window that allows the user to save a copy of an object with a different name.
Select All, Deselect All	Edit	Allows the user to select all the objects in a view or deselect all the selected objects in the view in which the choice was selected.
Size	System	Allows the user to change the size of the window.
Sort	View	Displays a window or cascaded menu that allows the user to arrange the objects in a view in a specified order.
Split	System	Allows the user to divide a window for displaying a view into more than one pane, and then change the sizes of the panes.
Tutorial	Help	Provides access to online educational information.
Undo, Redo	Edit	Undo reverses the effect of the last applied operation on an object—returning it to the state it was in before the operation was applied. Redo reverses the effect of the last applied undo.
Using Help	Help	Displays a window containing information about how to use the help function.
Window List	System	Displays a window containing a list of all the active applications or windows, depending on the environment.
	Windows	Displays a list of associated primary windows for subsequent access. For example, a Close All choice could be provided.

Controls

Controls differ in appearance, operation, and the kinds of choices for which they are suitable. A control is usually identified by a text label such as a field prompt (which identifies the field), column heading, group heading, or title. Controls can be grouped into four types: those relating to text,

Principles of User Interface Design

Common User Access

those related to non-text data, those relating to groups of data, and those related to action or routing choices.

The following tables describe each control and its recommended usage. This enumeration is worthwhile because, while at this time (2007) these controls have become common, they are sometimes misused.

Text Controls

These controls are used for settings choices involving text entry.

<i>Control</i>	<i>Description and Usage</i>
Entry field	Lets user type one or more lines of text into a rectangular area of fixed size. The number of characters can be limited. Scrolling can be used when data exceeds visible space.
Combination box	Combination entry field and list box (see <i>Non-Text Data Controls</i>). User can select one item from list or type directly in the entry field. Typed text can differ from list items.
Drop-down combination box	Combination box with the list hidden until user activates it.
Spin button with entry field	Displays a sequential ring of possible choices. User can select one choice or type a valid choice. Typically used for numbers like the month of the year. Limit choices to 20.

Non-Text Data Controls

These controls are used for settings choices that do not involve text entry.

<i>Control</i>	<i>Description and Usage</i>
Check box	A small square that represents a settings choice with two clearly distinguishable states, like on and off. Use a group of check boxes for settings choices that are not mutually exclusive and can each be set on or off (limit to 6 per field).
Drop-down list	A variation of a list that displays only one item until the user causes the other items to be displayed. One item can be selected from any number of items.
List box	A vertical list of objects or settings from which the user can select one or more items. Scrolling can be done when the list exceeds the visible space.
Radio button	A small circle that represents a fixed settings choice. Used in a group of two or more to display mutually exclusive choices for a field. Limit to 6 choices.
Slider	A linear device that represents a quantity within its range of possible values. Typically used with an incremental scale. User can change the value with buttons or the arm. Limit visible increments to 60.
Spin button without entry field	Same as spin button, but without the entry field.
Value set	A set of images or short text in squares that represent mutually exclusive settings choices. Limit choices to 20.

Data Group Controls

These controls handle the needs for grouping data that exceeds the capacity of a window.

Principles of User Interface Design

Common User Access

<i>Control</i>	<i>Description and Usage</i>
Notebook	Resembles a bound notebook containing pages separated into sections by tabbed dividers. User turns the pages or selects a tab to move to the first page in that section. Used to present data that can be logically organized into groups. Can contain unlimited controls except another notebook.
Container	A container that holds other objects. An example is a folder. Allows extended selection of unlimited objects.

Controls for Action and Routing Choices

These controls are used for action and routing choices.

<i>Control</i>	<i>Description and Usage</i>
Cascaded menu	A menu that appears to the side of the pull-down menu from which it was selected. Contains 10 or less action or routing choices in a vertical list from which only one item can be selected.
Menu bar	A list just below the title bar (of a window) containing routing choices in a horizontal list, each of which displays a pull-down menu when selected; usually horizontal. Limit to 6 choices.
Pop-up menu	A menu displayed next to an object. It contains choices, in a vertical list, appropriate to the given object, or set of objects, in their current context. Limit to 10 action or routing choices. (Today this is commonly called a context menu.)
Pull-down menu	A menu that extends below the selected choice on a menu bar or from the system-menu icon. Limit to 10 action or routing choices.
Push button	A rectangle containing text or an image that represents an action or routing choice. Use for all action and routing choices in windows (which display a view of an object) without menu bars.

Leaving theory aside for a moment, how controls work—what they can do—is specific to their implementation in a programming language. Typically controls are available to developers as DLL files and are manipulated with program code. Controls available from different vendors may vary in their appearance and operation. It is up to the designer to choose to use them in a way that is CUA-compliant.

Navigation

Navigation refers to movement of the cursor within a window and the access from a keyboard to any part of a window. Navigation is used to access controls. It is accomplished in several ways:

- **mouse:** Use for every choice and entry field; position pointer symbol at any position within an entry field.
- **first-letter navigation:** Use in all lists of objects or settings choices.
- **mnemonic:** Use for each predefined choice. It moves the cursor to that choice and also selects it.
- **shortcut key:** Use for each predefined menu choice or for each frequently-used menu choice. It moves the cursor to that choice (even when it is overlaid by other windows) and also selects it.

Cursor-movement keys:

Principles of User Interface Design

Common User Access

- **[Tab], [Shift+Tab]:** Move cursor to the next field (all controls are fields), left to right, top to bottom. Wrap cursor from bottom right to top left. Because a group of radio buttons is one field, [Tab] moves to the default or first button; an arrow key is used to move between buttons.
- **[Home]:** Move cursor to leftmost choice in a group of choices, or to the beginning of the current line in an entry field.
- **[End]:** Move cursor to the rightmost choice in a group of choices, or to the end of the current line in an entry field.
- **arrow keys:** Move cursor within entry field or multiple choices for one field.
- **[Page Down]:** Scroll down; displays content below the currently visible area.
- **[Page Up]:** Scroll up; displays content above the currently visible area.

Selection

Selection causes an action. For example:

- Selecting an **menu item** activates that choice, like printing a document.
- Selecting an **entry field** allows the user to type text in it.
- Selecting a **radio button** sets its corresponding choice and unsets a previous choice for the field.
- Selecting an item in a **combination box** list causes it to be placed in the entry field, overlaying any previous value.
- Selecting a **notebook tab** causes the first page of that section to be displayed.
- Selecting a **push button** activates its choice.
- Selecting a **window** activates it and causes it to be placed on top of all the other open windows. The previously active window, now inactive, is moved to the background.

User Information

Users are kept informed of the state of objects, processes, and other elements of their interaction with the application.

- **Feedback** is provided for processes that last longer than five seconds. This is typically done with clocks and sliders which provide progress information.
- **Information** can be explanative or descriptive, and can apply to a single object or a process or a view of an object. It is conveyed by text in a special area of the window.
- **Messages** are displayed in windows and are used for information that is particularly important or urgent, especially exceptions. Message windows include controls, typically push buttons, with which the user can respond. The message includes directions for possible responses, such as how an exception can be corrected.
- **Visual signals** use emphasis to indicate states like selected, open (in use), unavailable, and inappropriate text entered in a field.
- **Audible signals** are sounds used to draw the user's attention to significant conditions.
- **Help** describes how the application is to be used and the significance of key elements and conditions. It may include a tutorial to describe the business concepts and rules.

The key point is that too much information is an oxymoron.