Module 9

User Interface Design

Lesson 20

Basic Concepts in User Interface Design

Specific Instructional Objectives

At the end of this lesson the student will be able to:

- Identify five desirable characteristics of a user interface.
- Differentiate between user guidance and online help system.
- Differentiate between a mode-based interface and the modeless interface.
- Compare various characteristics of a GUI with those of a text-based user interface.

Characteristics of a user interface

It is very important to identify the characteristics desired of a good user interface. Because unless we are aware of these, it is very much difficult to design a good user interface. A few important characteristics of a good user interface are the following:

- **Speed of learning.** A good user interface should be easy to learn. Speed of learning is hampered by complex syntax and semantics of the command issue procedures. A good user interface should not require its users to memorize commands. Neither should the user be asked to remember information from one screen to another while performing various tasks using the interface. Besides, the following three issues are crucial to enhance the speed of learning:

    - **Use of Metaphors and intuitive command names.** Speed of learning an interface is greatly facilitated if these are based on some day-to-day real-life examples or some physical objects with which the users are familiar. The abstractions of real-life objects or concepts used in user interface design are called metaphors. If the user interface of a text editor uses concepts similar to the tools used by a writer for text editing such as cutting lines and paragraphs and pasting it at other places, users can immediately relate to it. Another popular metaphor is a shopping cart. Everyone knows how a shopping cart is used to make choices while purchasing items in a supermarket. If a user interface uses the shopping cart metaphor for designing the interaction style for a situation where similar types of choices have to be made, then the users can easily understand and learn to use the interface. Yet another example of a metaphor is the trashcan. To delete a file, the user may drag it to the trashcan. Also, learning is facilitated by intuitive command names and symbolic command issue procedures.

    - **Consistency.** Once a user learns about a command, he should be able to use the similar commands in different circumstances for carrying out similar actions. This makes it easier to learn the interface since the user can extend his knowledge about one part of the interface to the other parts. For example, in a word processor, “Control-b” is the short-cut key to embolden the selected text. The same short-cut should be used on the other parts of the interface, for example, to embolden text in graphic objects also - circle, rectangle, polygon, etc. Thus, the different commands supported by an interface should be consistent.

- **Component-based interface.** Users can learn an interface faster if the interaction style of the interface is very similar to the interface of other applications with which the user is already familiar. This can be achieved if the interfaces of different applications are developed using some standard user interface components. This, in fact, is the theme of the component-based user interface. Examples of standard user interface components are: radio button, check box, text field, slider, progress bar, etc.

The speed of learning characteristic of a user interface can be determined by measuring the training time and practice that users require before they can effectively use the software.

- **Speed of use.** Speed of use of a user interface is determined by the time and user effort necessary to initiate and execute different commands. This characteristic of the interface is some times referred to as productivity support of the interface. It indicates how fast the users can perform their intended tasks. The time and user effort necessary to initiate and execute different commands should be minimal. This can be achieved through careful design of the interface. For example, an interface that requires users to type in lengthy commands or involves mouse movements to different areas of the screen that are wide apart for issuing commands can slow down the operating speed of users. The most frequently used commands should have the smallest length or be available at the top of the menu to minimize the mouse movements necessary to issue commands.

- **Speed of recall.** Once users learn how to use an interface, the speed with which they can recall the command issue procedure should be maximized. This characteristic is very important for intermittent users. Speed of recall is improved if the interface is based on some metaphors, symbolic command issue procedures, and intuitive command names.

- **Error prevention.** A good user interface should minimize the scope of committing errors while initiating different commands. The error rate of an interface can be easily determined by monitoring the errors committed by average users while using the interface. This monitoring can be automated by instrumenting the user interface code with monitoring code which can record the frequency and types of user error and later display the statistics of various kinds of errors committed by different users.

Moreover, errors can be prevented by asking the users to confirm any potentially destructive actions specified by them, for example, deleting a group of files.

Consistency of names, issue procedures, and behavior of similar commands and the simplicity of the command issue procedures minimize error possibilities. Also, the interface should prevent the user from entering wrong values.

- **Attractiveness.** A good user interface should be attractive to use. An attractive user interface catches user attention and fancy. In this respect, graphics-based user interfaces have a definite advantage over text-based interfaces.

- **Consistency.** The commands supported by a user interface should be consistent. The basic purpose of consistency is to allow users to generalize the knowledge about aspects of the interface from one part to another. Thus, consistency facilitates speed of learning, speed of recall, and also helps in reduction of error rate.

- **Feedback.** A good user interface must provide feedback to various user actions. Especially, if any user request takes more than few seconds to process, the user should be informed about the state of the processing of his request. In the absence of any response from the computer for a long time, a novice user might even start recovery/shutdown procedures in panic. If required, the user should be periodically informed about the progress made in processing his command.

  For example, if the user specifies a file copy/file download operation, a progress bar can be displayed to display the status. This will help the user to monitor the status of the action initiated.

- **Support for multiple skill levels.** A good user interface should support multiple levels of sophistication of command issue procedure for different categories of users. This is necessary because users with different levels of experience in using an application prefer different types of user interfaces. Experienced users are more concerned about the efficiency of the command issue procedure, whereas novice users pay importance to usability aspects. Very cryptic and complex commands discourage a novice, whereas elaborate command sequences make the command issue procedure very slow and therefore put off experienced users. When someone uses an application for the first time, his primary concern is speed of learning. After using an application for extended periods of time, he becomes familiar with the operation of the software. As a user becomes more and more familiar with an interface, his focus shifts from usability aspects to speed of command issue aspects. Experienced users look for options such as “hot-keys”, “macros”, etc. Thus, the skill level of users improves as they keep using a software product and they look for commands to suit their skill levels.

- **Error recovery (undo facility).** While issuing commands, even the expert users can commit errors. Therefore, a good user interface should allow a user to undo a mistake committed by him while using the interface. Users are put to inconvenience, if they cannot recover from the errors they commit while using the software.

- **User guidance and on-line help.** Users seek guidance and on-line help when they either forget a command or are unaware of some features of the software. Whenever users need guidance or seek help from the system, they should be provided with the appropriate guidance and help.

User guidance and online help

Users may seek help about the operation of the software any time while using the software. This is provided by the on-line help system. This is different from the guidance and error messages which are flashed automatically without the user asking for them. The guidance messages prompt the user regarding the options he has regarding the next command, and the status of the last command, etc.

On-line Help System. Users expect the on-line help messages to be tailored to the context in which they invoke the “help system”. Therefore, a good on-line help system should keep track of what a user is doing while invoking the help system and provide the output message in a context-dependent way. Also, the help messages should be tailored to the user’s experience level. Further, a good on-line help system should take advantage of any graphics and animation characteristics of the screen and should not just be a copy of the user’s manual. Fig. 9.1 gives a snapshot of a typical on-line help provided by a user interface.

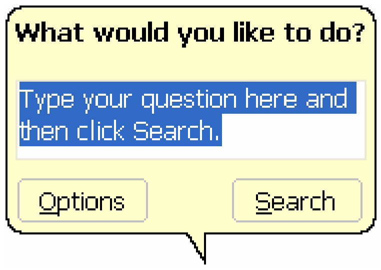

What would you like to do?
Type your question here and then click Search.
Options
Search

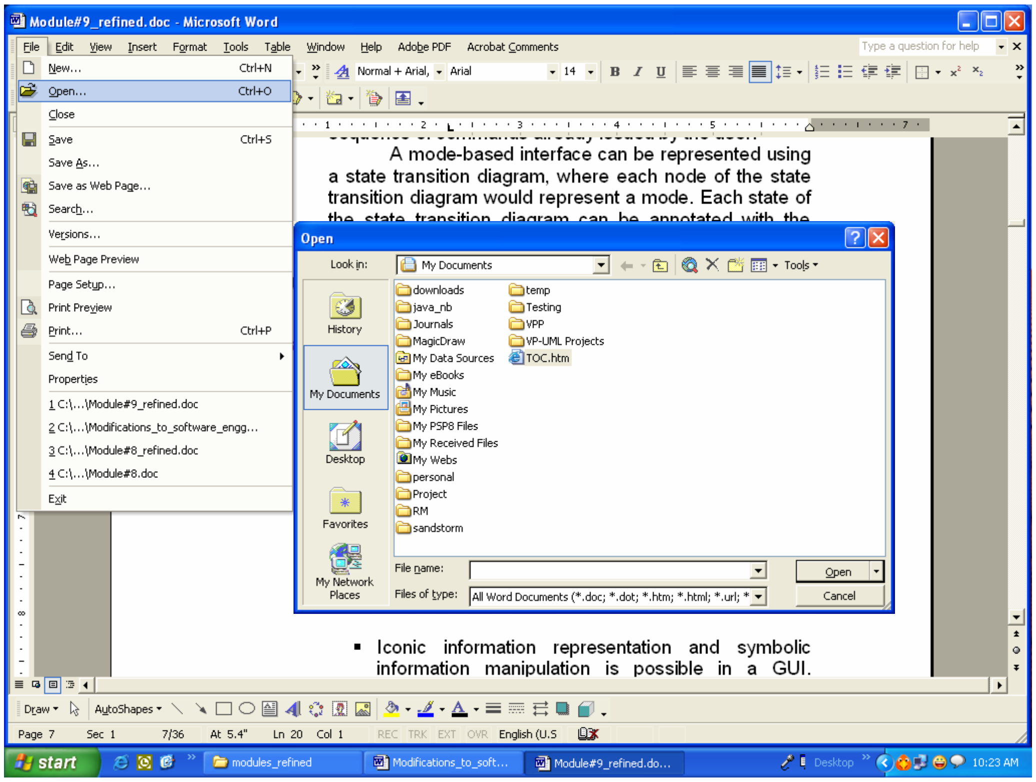
Fig. 9.1. *Example of an on-line help interface*

Guidance Messages. The guidance messages should be carefully designed to prompt the user about the next actions he might purse, the current status of the system, the progress made so far in processing his last command, etc. A good guidance system should have different levels of sophistication for different categories of users. For example, a user using a command language interface might need a different type of guidance compared to a user using a menu or iconic interface. Also, users should have an option to turn off detailed messages.

• Mode-based interface vs. modeless interface

- A mode is a state or collection of states in which only a subset of all user interaction tasks can be performed. In a modeless interface, the same set of commands can be invoked at any time during the running of the software. Thus, a modeless interface has only a single mode and all the commands are available all the time during the operation of the software. On the other hand, in a mode-based interface, different set of commands can be invoked depending on the mode in which the system is, i.e. the mode at any instant is determined by the sequence of commands already issued by the user.

  A mode-based interface can be represented using a state transition diagram, where each node of the state transition diagram would represent a mode. Each state of the state transition diagram can be annotated with the commands that are meaningful in that state.



A mode-based interface can be represented using a state transition diagram, where each node of the state transition diagram would represent a mode. Each state of the state transition diagram can be annotated with the

- Iconic information representation and symbolic information manipulation is possible in a GUI.

Fig 9.2. An example of mode-based interface

Fig 9.2 shows the interface of a word processing program. The top-level menu provides the user with a gamut of operations like file open, close, save, etc. When the user chooses the open option, another frame is popped up which limits the user to select a name from one of the folders.

Graphical User Interface vs. Text-based User Interface

- The following comparisons are based on various characteristics of a GUI with those of a text-based user interface.

    - In a GUI multiple windows with different information can simultaneously be displayed on the user screen. This is perhaps one of the biggest advantages of GUI over text-based interfaces since the user has the flexibility to simultaneously interact with several related items at any time and can have access to different system information displayed in different windows.

    - Iconic information representation and symbolic information manipulation is possible in a GUI. Symbolic information manipulation such as dragging an icon representing a file to a trash can be deleting is intuitively very appealing and the user can instantly remember it.

    - A GUI usually supports command selection using an attractive and user-friendly menu selection system.
    - In a GUI, a pointing device such as a mouse or a light pen can be used for issuing commands. The use of a pointing device increases the efficacy issue procedure.
- On the flip side, a GUI requires special terminals with graphics capabilities for running and also requires special input devices such a mouse. On the other hand, a text-based user interface can be implemented even on a cheap alphanumeric display terminal. Graphics terminals are usually much more expensive than alphanumeric terminals. However, display terminals with graphics capability with bit-mapped high-resolution displays and significant amount of local processing power have become affordable and over the years have replaced text-based terminals on all desktops. Therefore, the emphasis of this lesson is on GUI design rather than text-based user interface design.