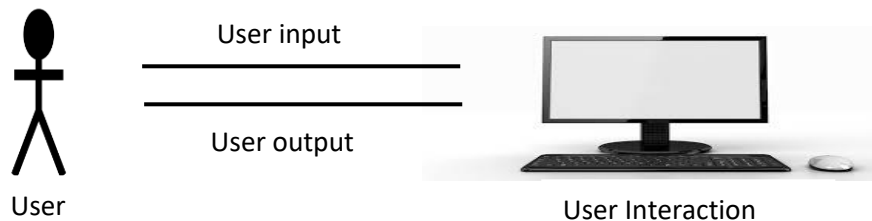


SOFTWARE ENGINEERING UNIT-III PART-2

User Interface Design: Characteristics of Good User Interface, Basic Concepts, Types of User Interfaces, Fundamentals of Component-based GUI Development, A User Interface Design Methodology.

INTRODUCTION TO USER INTERFACE DESIGN

User Interface is the front-end application view of the computer system with which user can interact in order to use the software and achieve the goals. Nowadays, humans are coming into contact with wide range of computer-based technologies and devices.



User Interface bridges the gap between the user and computer. The goodness and effective of software can be rated based on its Graphical User interface (GUI). However the user experience plays an important role in the use of software.

CHARACTERISTICS OF GOOD USER INTERFACE

A good user interface design encourages user interaction with computer system in easier and natural way. It allows the user to perform the intended tasks efficiently and effectively. The following are some important characteristics of a good user interface.

- ❖ **Usability:** It is a measure of how easy it is to use a product to perform an intended task. It is an outcome of a user-centered design process, which incorporates user concerns. A usable product can be characterized by the following quality attributes.
 - ✓ Effectiveness
 - ✓ Efficiency
 - ✓ User Satisfaction
 - ✓ Easy to Learn
 - ✓ Error Tolerant
 - ✓ Flexibility
 - ✓ Simplicity
 - ✓ Portability
- ❖ **Feedback:** It plays an important role in user interface design. A user expects feedback for each action while interacting with system. It improves user satisfaction. A feedback explains what is happening currently, what is going to happen next, what happens after submission and whether user is doing things correctly or not.

- ❖ **Intuitive and Consistent:** A good user interface presents and acquires information in a consistent manner. Through consistency, user can easily understand the next part of the software while working with current part.
- ❖ **Attractive and Aesthetic:** A good interface is attractive to use, looks pleasing and fancy. It should be designed to promote aesthetic perception and engaging interaction to the user.
- ❖ **Forgiveness:** An interface should tolerate errors that come up in the system. A forgiving interface allows users to easily recall or undo actions if done incorrectly. Where a system should generate error messages and resolve common mistakes to avoid trouble to the users.
- ❖ **Responsive:** A good interface responds quickly to the user. The fast response improves the user experience. The quick responsive application interface improves the experience and scope of usability by the users.
- ❖ **Maintainability:** As the technology and its usage changes, there should have the capacity for changes to be integrated without causing a conflict of interest. There should be a scope of incorporating the additional features.
- ❖ **Compatibility:** A good interface is compatible with users and the platform. The designer should understand the user needs and expectations.
- ❖ **User Guidance:** Some user guidance such as help systems, on-line manuals, comments, notes etc should be supplied with the interface. Should also provide context-sensitive user help facilities.
- ❖ **Robustness:** A good interface supports different categories of known and unknown users. There should be recovery tips at any state of the system. A user should be able to navigate from any given state to other with good responsive and recoverable features.

BASIC CONCEPTS OF USER INTERFACE

User Interface design is a key aspect to the successful use of a software product. There are some basic concepts in designing of effective user interface such as interface design elements and considerations, visual designs, help system and user centered design.

❖ **Interface Design Elements:**

Interface design is a way of user interaction. User interacts through input design, navigation, feedback, containers and output designs. An effective design of elements increases efficiency and user satisfaction.

The following are the elements of interface design like

- Input design controls
- Navigation
- Feedback
- Containers
- Output design

❖ **Visual Design:**

It is the use of typography, symbols, colors and graphics to represent information to the user. It is a kind of art of keeping things to look good and usable. It should be information oriented to understand complex information. Aesthetic issues of users are considered during visual design.

❖ **Help System:**

The user guidance affects the usability of the application. Users need to know how to work with the application. The functionality of the application should work properly. There should be proper feedback or message after performing the action. Online help system allows user to explore the alternative way of system operation.

❖ **User Centered Design:**

It is an approach to design an application with the involvement of user in the process. It aims to improve the interaction between people and systems to make technology effectively accessible. Where user needs are the centre of focus in designing user interface. The User centered design is becoming popular in today's web development which focus on usability principles, user characteristics, environment, features, and workflow in the design of an interface.

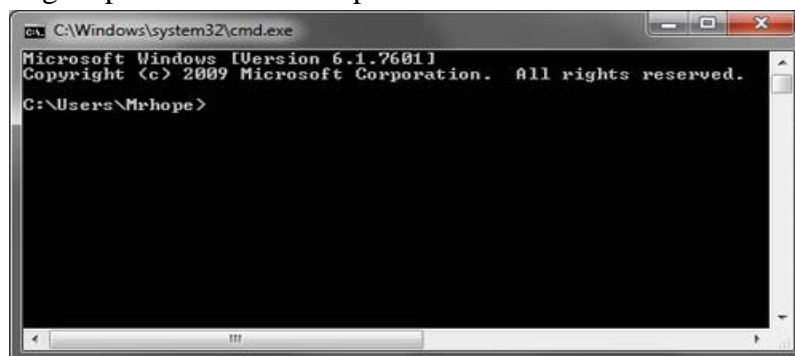
TYPES OF USER INTERFACES

User interaction with computer has now been trending in various ways. The following are the different categories of interfaces.

❖ **Command Language Interface**

The user provides the input by typing a command string with the computer keyboard and the system provides output by printing text on the computer monitor. Most commonly, command line interface is used in DOS computers. User interaction is possible through running commands correctly. The command prompt indicated the current position on command line.

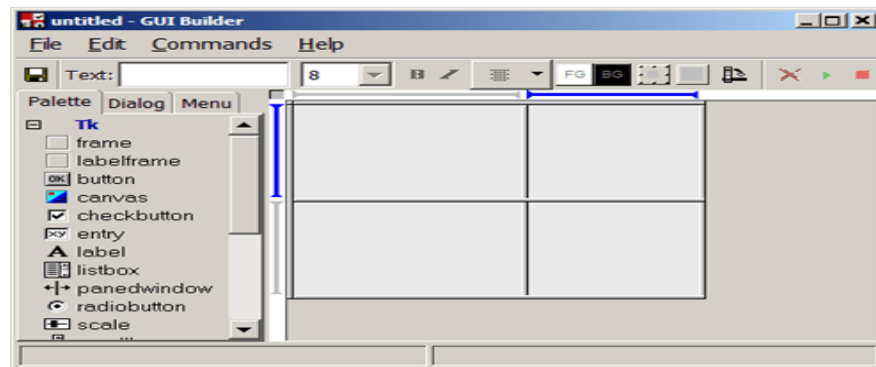
Certain advantage of command line interface is quick to use depending upon the awareness about commands which is faster in execution and response as compared to other types of user interface. It can be very difficult and unfriendly to use for those who are not computer-savvy. If there are mistakes in typing or running commands, it may result in data loss or illegal operation in the computer.



❖ **Graphical User Interface**

The use of pictures rather than just words to represent the input and output of a program. Input is accepted via devices such as keyboard and mouse. Is the most commonly used user interface. It makes use of pictures, graphics and icons. Windows Operating system is an example of GUI software. It is useful in many respects. It can be

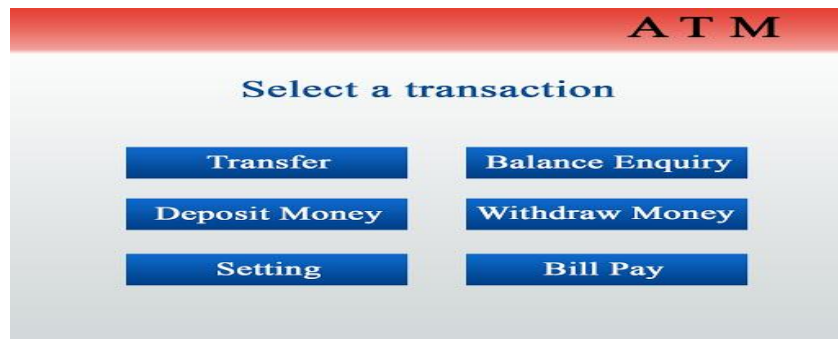
easily explore and find the information through GUI. Where GUI takes up higher space in memory than other interfaces, which uses more CPU processing power than other types of interfaces.



❖ Menu Driven Interface

It provides the user with a list of available selections through screens or menus. Here the user must know which task they desire to perform. User selects options in menu by using keyboard, light, pen, touch screen or mouse. The menu driven depends upon nature of the application. An example of menu driven is ATM machine with menu options.

This interface is very easy to use. There is no difficult command to learn or remember to operate the system. However, improper design of menu interface may be confusing or slow to use.



❖ Direct Manipulation Interface

It allows a user to directly interact with interface objects. Here the objects of interest are represented as distinguishable objects in the user interface and are manipulated in a direct fashion. For example, moving files in windows explorer by dragging file names from one folder to another. This interface is visible and continuous representation of the task objects and their actions. There is no need of extra efforts to learn and remember the commands. The operations are performed in rapid and incremental way. Users get immediate feedback on their actions. So mistakes can be quickly detected and corrected. Direct manipulation interfaces can be complex to program and make heavy demands on the computer system.



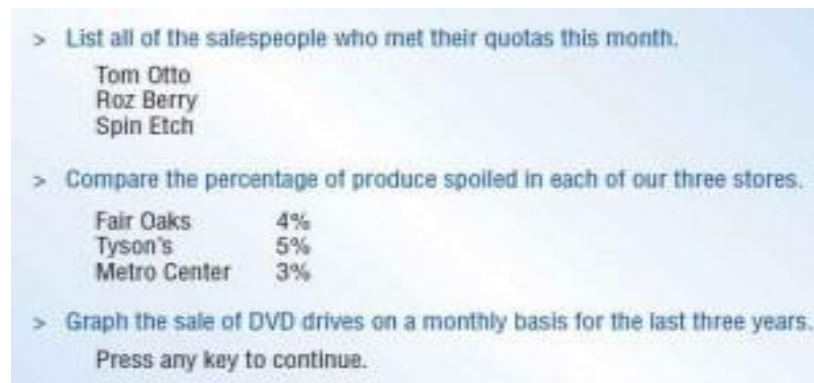
❖ Form fill-in Interface

Form fill-in interface consist of onscreen forms or Web-based forms displaying fields containing data items or parameters that need to be communicated to the user. This interface technique is also known as a form-based method and input/output forms. Forms for display screens are set up to show what information should be input and where. This arrangement allows movement one field backward or forward by clicking the appropriate arrow key. It provides the user good control over data entry. The chief advantage of the input/output form interface is that It shows field labels as well as the context for entries. In addition, Web forms can return incomplete forms to the user with an explanation of what data must be entered to complete the transaction. The main drawback is that users experienced with the system or application might become impatient with input/output forms and might want more efficient ways to enter data.

❖ Natural Language Interface

Natural-language interfaces are perhaps the dream and ideal of inexperienced users, because they permit them to interact with the computer in their everyday, or natural, language. No special skills are required of the user, who interfaces with the computer using natural language. Implementation problems and extraordinary demand on computing resources have so far kept natural-language interfaces to a minimum. The demand exists,

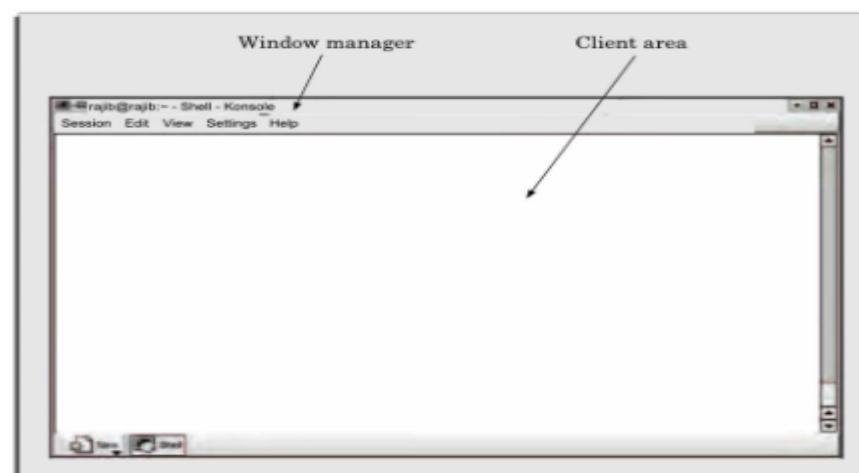
though, and many programmers and researchers are working diligently on such interfaces. It is a growth area, and it therefore merits continued monitoring.



FUNDAMENTALS OF COMPONENT-BASED GUI DEVELOPMENT

Component-based architecture focuses on the decomposition of the design into individual functional or logical components that represent well-defined communication interfaces containing methods, events, and properties. The primary objective of component-based architecture is to ensure **component reusability**. A component encapsulates functionality and behaviors of a software element into a reusable and self-deployable binary unit. There are many standard component frameworks such as COM/DCOM, JavaBean, EJB, CORBA, .NET, web services, and grid services.

The fundamentals of component GUI development window can be considered to be a virtual screen, in the sense that it provides an interface to the user for carrying out independent activities. A window can be divided into two parts: client part, and non-client part.



Window Management System (WMS):

Most GUI environments do this through a window management system (WMS). A window management system is primarily a resource manager. It keeps track of the screen area resource and allocates it to the different windows that seek to use the screen.

A WMS consists of two parts

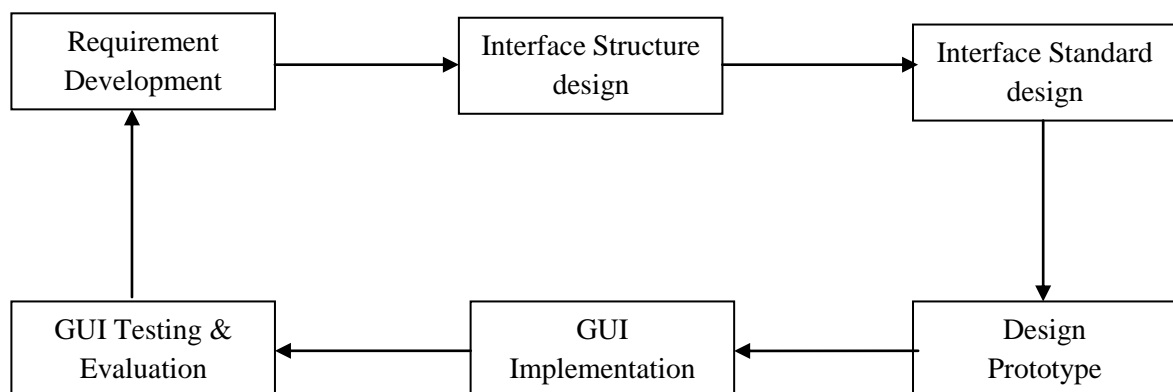
1. Window manager
2. Window system

Window manager is the component of WMS with which the end user interacts to do various window-related operations such as window repositioning, window resizing, iconification, etc. Different interface programming packages support different widget sets. However, a surprising number of them contain similar kinds of widgets, so that one can think of a generic widget set which is applicable to most interfaces.

USER INTERFACE DESIGN METHODOLOGY

User interface design is the design of websites, computers, applications and likewise. As the name says it's an interface design in which users are of prime importance. While designing the first and foremost thing that needs to be kept in mind is the user experience and interaction.

There are a number of activities performed for designing user interface. The process of GUI design and design implementation is alike SDLC. This process starts with Requirement identification to GUI evaluation through user interaction. GUI design methodologies



❖ Requirement Development

The purpose of this phase is to understand user needs and functional requirements in the interface. It begins with identifying customers and end-users who will use the system. A document of usability analysis is prepared outlining potential user capabilities of the product. These requirements are developed through use case model.

❖ Interface Structure Design

This phase in the organisation of various objects or elements together that perform user's task. It is also known as task analysis. A task is an action performed by the user on an object such as forms, reports, menus, screens etc. The simplest way of interface structural design is to use of task decomposition.

❖ **Interface Standard Design**

It is the common setting and considerations across the individual elements. These can be different set of interface standards for different parts of the system. The interface metaphor and interface templates are used in this design.

❖ **Design Prototype**

A prototype is a rough layout or simulation of final user interface involving controls such as forms, screens, reports. It gives the user a view to the application and its working. It can add or remove elements in the interface. It improves efficiency of software development process. Web prototype is created through web based languages such as Microsoft front page, HTML, PHP etc.

❖ **GUI Implementation**

It is finally developed with front end application development language or tools. Modern programming frameworks such as .NET, Java, Python etc provide inbuilt controls and components which can be embedded into the application.

❖ **GUI Testing & Evaluation**

It is performed in various ways. Some of the widely used methods are in-house inspection, direct involvement of user, and release of beta version. Here testing mainly includes usability, compatibility, consistency, and user acceptance. GUI testing is performed in order to evaluate interface to improve the quality of design interface and prevent errors to optimize designs.