

Name of Faculty : Prof. Chanchal Lohi

Designation : Assistant Professor

Department : Information Technology

Subject : Software Engineering

Subject Code : IT-604 (B)

Unit : III

Topic : User Interface Design

"WORKING TOWARDS BEING THE BEST"

User Interface Design

The visual part of a computer application or operating system through which a client interacts with a computer or software. It determines how commands are given to the computer or the program and how data is displayed on the screen

1. TYPES OF USER INTERFACE

There are two main types of User Interface:

1. Text-Based User Interface or Command Line Interface.
2. Graphical User Interface (GUI)

1.1 Text-Based User Interface: This method relies primarily on the keyboard. A typical example of this is UNIX.

Advantages:

- Many and easier to customize options.
- Typically capable of more important task.

Disadvantages

- Relies heavily on recall rather than recognition.
- Navigation is often more difficult.

1.2 Graphical User Interface (GUI): GUI relies much more heavily on the mouse. A typical example of this type of interface is any versions of the Windows operating systems.

GUI Characteristics:

Characteristics	Descriptions
Windows	Multiple windows allow different information to be displayed simultaneously on the user's screen.

Icons	icons different types of information. On some systems, icons represent files. On other icons describes processes.
Menus	Commands are selected from a menu rather than typed in a command language.
Pointing	A pointing device such as a mouse is used for selecting choices from a menu or indicating items of interests in a window.
Graphics	Graphics elements can be mixed with text or the same display

Advantages:

- Less expert knowledge is required to use it.
- Easier to Navigate and can look through folders quickly in a guess and check manner.
- The user may switch quickly from one task to another and can interact with several different applications.

Disadvantages:

- Typically decreased options.
- Usually less customizable. Not easy to use one button for tons of different variations.

2. UI DESIGN PRINCIPLES

- a) **Structure:** Design should organize the user interface purposefully, in the meaningful and usual based on precise, consistent models that are apparent and recognizable to users, putting related things together and separating unrelated things, differentiating dissimilar things and making similar things resemble one another. The structure principle is concerned with overall user interface architecture.
- b) **Simplicity:** The design should make the simple, common task easy, communicating clearly and directly in the user's language, and providing good shortcuts that are meaningfully related to longer procedures.

- c) **Visibility:** The design should make all required options and materials for a given function visible without distracting the user with extraneous or redundant data.
- d) **Feedback:** The design should keep users informed of actions or interpretation, changes of state or condition, and bugs or exceptions that are relevant and of interest to the user through clear, concise, and unambiguous language familiar to users.
- e) **Tolerance:** The design should be flexible and tolerant, decreasing the cost of errors and misuse by allowing undoing and redoing while also preventing bugs wherever possible by tolerating varied inputs and sequences and by interpreting all reasonable actions.

3. GUI DESIGN LEVELS

Proper or good UI design works from the user's capabilities and limitations not the machines. While designing the UI, knowledge of the nature of the user's work and environment is also critical.

The task to be performed can then be divided which are assigned to the user or machine, based on knowledge of the capabilities and limitations of each. The design of a user interface is often divided into four different levels –

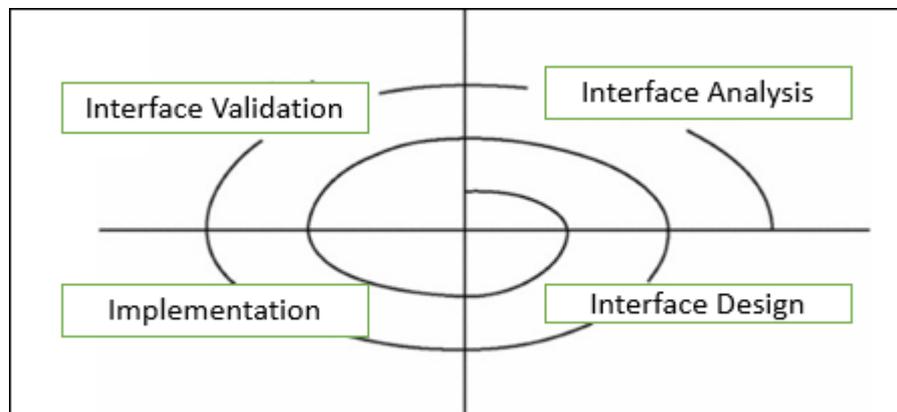
1. **The conceptual level** – It describes the basic entities considering the user's view of the system and the actions possible upon them.
2. **The semantic level** – It describes the functions performed by the system i.e. description of the functional requirements of the system, but does not address how the user will invoke the functions.
3. **The syntactic level** – It describes the sequences of inputs and outputs required to invoke the functions described.
4. **The lexical level** – It determines how the inputs and outputs are actually formed from primitive hardware operations.

User interface design is an iterative process, where all the iteration explains and refines the information developed in the preceding steps. General steps for user interface design

- Defines user interface objects and actions (operations).
- Defines events (user actions) that will cause the state of the user interface to change.
- Indicates how the user interprets the state of the system from information provided through the interface.
- Describe each interface state as it will actually look to the end user.

4. USER INTERFACE DEVELOPMENT PROCESS

It follows a spiral process as shown in the following diagram –



Interface analysis: It concentrates or focuses on users, tasks, content, and work environment who will interact with the system. Defines the human - and computer-oriented tasks that are required to achieve system function.

Interface design: It defines a set of interface objects, actions, and their screen representations that enable a user to perform all defined tasks in a manner that meets every usability objective defined for the system.

Interface construction: It starts with a prototype that enables usage scenarios to be evaluated and continues with development tools to complete the construction.

Interface validation: It focuses on the ability of the interface to implement every user task correctly, accommodate all task variations, to achieve all general user requirements, and the degree to which the interface is easy to use and easy to learn.

5. USER INTERFACE MODELS

When a user interface is analyzed and designed following four models are used –

User profile model:

- Created by a user or software engineer, which establishes the profile of the end-users of the system based on age, gender, physical abilities, education, motivation, goals, and personality.
- Considers syntactic and semantic knowledge of the user and classifies users as novices, knowledgeable intermittent, and knowledgeable frequent users.

Design model:

- Created by a software engineer which incorporates data, architectural, interface, and procedural representations of the software.
- Derived from the analysis model of the requirements and controlled by the information in the requirements specification which helps in defining the user of the system.

Implementation model:

- Created by the software implementers who work on look and feel of the interface combined with all supporting information (books, videos, help files) that describes system syntax and semantics.
- Serves as a translation of the design model and attempts to agree with the user's mental model so that users then feel comfortable with the software and use it effectively.

User's mental model:

- Created by the user when interacting with the application. It contains the image of the system that users carry in their heads.
- Often called the user's system perception and correctness of the description depends upon the user's profile and overall familiarity with the software in the application domain.