User Interface Design
Designing effective interfaces for software systems

Importance of user interface
- System users often judge a system by its interface rather than its functionality
- A poorly designed interface can cause a user to make catastrophic errors
- Poor user interface design is the reason why so many software systems are never used

Graphical user interfaces
- Most users of business systems interact with these systems through graphical user interfaces (GUIs)
  - although, in some cases, legacy text-based interfaces are still used

GUI characteristics
- Windows
- Icons
- Menus
- Pointing Devices
- Graphics

GUI advantages
- They are easy to learn and use
  - Users without experience can learn to use the system quickly
- The user may switch quickly from one task to another and can interact with several different applications
  - Information remains visible in its own window when attention is switched
- Fast, full-screen interaction is possible with immediate access to anywhere on the screen

User-centred design
- User-centred design is an approach to UI design where the needs of the user are paramount and where the user is involved in the design process
- UI design always involves the development of prototype interfaces
**User interface design process**

1. Analyze and understand user activities
2. Produce prototype
3. Evaluate design with end-users
4. Design prototype
5. Evaluate design with end-users
6. Implement final user interface

**UI design principles**

- **User familiarity**
  - The interface should use terms and concepts which are familiar to users. For example, an office system should use concepts such as letters, documents, folders, rather than directories, file identifiers.

- **Consistency**
  - The system should display an appropriate level of consistency. Commands and menus should have the same format, command punctuation should be similar, etc.

- **Minimal surprise**
  - If a command operates in a known way, the user should be able to predict the operation of comparable commands.

- **Recoverability**
  - The system should provide resilience to user errors and allow the user to recover from errors. This might include an UNDO facility.

- **User guidance**
  - Some user guidance such as help systems, online manuals, etc. should be supplied.

- **User diversity**
  - Interaction facilities for different types of users should be supported. For example, some users may have sight difficulties and so larger text should be available.

**Design principles**

- **User familiarity**
  - The interface should be based on user-oriented terms and concepts rather than computer concepts.

- **Consistency**
  - The system should display an appropriate level of consistency. Commands and menus should have the same format, command punctuation should be similar, etc.

- **Minimal surprise**
  - If a command operates in a known way, the user should be able to predict the operation of comparable commands.

**User-system interaction**

- **Recoverability**
  - The system should provide resilience to user errors and allow the user to recover from errors. This might include an UNDO facility.

- **User guidance**
  - Some user guidance such as help systems, online manuals, etc. should be supplied.

- **User diversity**
  - Interaction facilities for different types of users should be supported. For example, some users may have sight difficulties and so larger text should be available.
## Interaction styles

- Direct manipulation
- Menu-based (Menu selection)
- Form fill-in
- Command language
- Natural language

## Advantages and disadvantages

<table>
<thead>
<tr>
<th>Interaction style</th>
<th>Main advantages</th>
<th>Main disadvantages</th>
<th>Application examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct manipulation</td>
<td>Fast and intuitive interaction, Easy to learn</td>
<td>May be hard to implement, Only suitable where there is a visual metaphor for tasks and objects</td>
<td>Video games, CAD systems</td>
</tr>
<tr>
<td>Menu selection</td>
<td>Avoids user error, Little typing required</td>
<td>Slow for experienced users, Can become complex if many menu options</td>
<td>Most general-purpose systems</td>
</tr>
<tr>
<td>Form fill-in</td>
<td>Simple data entry, Easy to learn</td>
<td>Takes up a lot of screen space</td>
<td>Stock control, Personal loan processing</td>
</tr>
<tr>
<td>Command language</td>
<td>Powerful and flexible</td>
<td>Hard to learn, Poor error management</td>
<td>Operating systems, Library information retrieval systems</td>
</tr>
<tr>
<td>Natural language</td>
<td>Accessible to casual users, Easily extended</td>
<td>Requires more typing, Natural language understanding systems are unreliable</td>
<td>Timetable systems, WWW information retrieval systems</td>
</tr>
</tbody>
</table>

## Direct manipulation advantages

- Users feel in control of the computer and are less likely to be intimidated by it
- User learning time is relatively short
- Users get immediate feedback on their actions so mistakes can be quickly detected and corrected

## Direct manipulation problems

- The derivation of an appropriate information space model can be very difficult
- Given that users have a large information space, what facilities for navigating around that space should be provided?
- Direct manipulation interfaces can be complex to program and make heavy demands on the computer system

## Menu systems

- Users make a selection from a list of possibilities presented to them by the system
- The selection may be made by pointing and clicking with a mouse, using cursor keys or by typing the name of the selection
- May make use of simple-to-use terminals such as touch-screens

## Advantages of menu systems

- Users need not remember command names as they are always presented with a list of valid commands
- Typing effort is minimal
- User errors are trapped by the interface
- Context-dependent help can be provided. The user's context is indicated by the current menu selection
**Problems with menu systems**

- Actions that involve logical conjunction (and) or disjunction (or) are awkward to represent.
- Menu systems are best suited to presenting a small number of choices. If there are many choices, some menu structuring facility must be used.
- Experienced users find menus slower than command language.

**Command interfaces**

- User types commands to give instructions to the system e.g. UNIX.
- May be implemented using cheap terminals.
- Easy to process using compiler techniques.
- Commands of arbitrary complexity can be created by command combination.
- Concise interfaces requiring minimal typing can be created.

**Command interfaces problems**

- Users have to learn and remember a command language. Command interfaces are therefore unsuitable for occasional users.
- Users make errors in commands. An error detection and recovery system is required.
- System interaction is through a keyboard so typing ability is required.

**Command languages**

- Often preferred by experienced users because they allow for faster interaction with the system.
- Not suitable for casual or inexperienced users.
- May be provided as an alternative to menu commands (keyboard shortcuts). In some cases, a command language interface and a menu-based interface are supported at the same time.

**Natural language interfaces**

- The user types a command in a natural language. Generally, the vocabulary is limited and these systems are confined to specific application domains (e.g. timetable enquiries).
- NL processing technology is now good enough to make these interfaces effective for casual users but experienced users find that they require too much typing.

**Multiple user interfaces**

[Diagram showing multiple user interfaces]
Information presentation

- Information presentation is concerned with presenting system information to system users.
- The information may be presented directly (e.g., text in a word processor) or may be transformed in some way for presentation (e.g., in some graphical form).
- The Model-View-Controller approach is a way of supporting multiple presentations of data.

Model-view-controller

Static information

- Initialized at the beginning of a session.
- It does not change during the session.

Dynamic information

- Changes during a session and the changes must be communicated to the system user.
- Both may be either numeric or textual.

Information display factors

- Is the user interested in precise information or data relationships?
- How quickly do information values change? Must the change be indicated immediately?
- Must the user take some action in response to a change?
- Is there a direct manipulation interface?
- Is the information textual or numeric? Are relative values important?
**Analog vs. digital presentation**

- **Digital presentation**
  - Compact - takes up little screen space
  - Precise values can be communicated
- **Analog presentation**
  - Easier to get an ‘at a glance’ impression of a value
  - Possible to show relative values
  - Easier to see exceptional data values

**Dynamic information display**

- Dial with needle
- Pie chart
- Thermometer
- Horizontal bar

**Displaying relative values**

<table>
<thead>
<tr>
<th>Pressure</th>
<th>Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 100 200 300 400</td>
<td>0 25 50 75 100</td>
</tr>
</tbody>
</table>

**Text highlighting**

- The filename you have chosen has been used. Please choose another name
  - Ch. 16 User interface design

**User documentation**

- As well as on-line information, paper documentation should be supplied with a system
- Documentation should be designed for a range of users from inexperienced to experienced
- As well as manuals, other easy-to-use documentation such as a quick reference card may be provided

**Document types**

- **Functional description**
  - Brief description of what the system can do
- **Introductory manual**
  - Presents an informal introduction to the system
- **System reference manual**
  - Describes all system facilities in detail
- **System installation manual**
  - Describes how to install the system
- **System administrator's manual**
  - Describes how to manage the system when it is in use
**User interface evaluation**

- Some evaluation of a user interface design should be carried out to assess its suitability.
- Full scale evaluation is very expensive and impractical for most systems.
- Ideally, an interface should be evaluated against a usability specification. However, it is rare for such specifications to be produced.

**Usability attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learnability</td>
<td>How long does it take a new user to become productive with the system?</td>
</tr>
<tr>
<td>Speed of Operation</td>
<td>How well does the system response match the user's work practice?</td>
</tr>
<tr>
<td>Robustness</td>
<td>How tolerant is the system of user error?</td>
</tr>
<tr>
<td>Recoverability</td>
<td>How good is the system at recovering from user error?</td>
</tr>
<tr>
<td>Adaptability</td>
<td>How closely is the system tied to a single model of work?</td>
</tr>
</tbody>
</table>

**Simple evaluation techniques**

- Questionnaires for user feedback
- Video recording of system use and subsequent tape evaluation
- Instrumentation of code to collect information about facility use and user errors
- The provision for on-line user feedback