Chapter 15
User Interface Design
Designing effective interfaces for software systems

Objectives are to
- suggest some general design principles for user interface design,
- explain different interaction styles,
- introduce styles of information presentation,
- describe the user support which should be built into user interfaces, and
- introduce usability attributes and system approaches to system evaluation.

Topics covered include
- user interface design principles
- user interaction
- information presentation
- user support
- interface evaluation

The 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 interfaces although, in some cases, legacy text-based interfaces are still used.

GUI characteristics
<table>
<thead>
<tr>
<th>characteristic</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>windows</td>
<td>Multiple windows allow different information to be displayed simultaneously</td>
</tr>
<tr>
<td>icons</td>
<td>Icons represent different types of entities such as files, processes, etc.</td>
</tr>
<tr>
<td>menus</td>
<td>Commands are selected from a list instead of typed</td>
</tr>
<tr>
<td>pointing</td>
<td>A pointing device is used to make selection</td>
</tr>
<tr>
<td>graphics</td>
<td>Graphical elements and text can be mixed on the same display</td>
</tr>
</tbody>
</table>
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

- The aim of this chapter is to sensitize software engineers to key issues underlying the design rather than the implementation of user interfaces.
- 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

UI design principles

- UI design must take into account the needs, experience, and capabilities of the system users.
- Designers should be aware of people’s physical and mental limitations (e.g., limited short-term memory) and should recognize that people make mistakes.
- UI design principles underlie interface designs although not all principles are applicable to all designs.

User interface design principles

<table>
<thead>
<tr>
<th>Principle</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User familiarity</td>
<td>Use terms and concepts drawn from those who are going to make most use of the system</td>
</tr>
<tr>
<td>Consistency</td>
<td>Comparable operations should be activated in the same way wherever possible</td>
</tr>
<tr>
<td>Minimum surprise</td>
<td>User should never be surprised by the behavior of the system</td>
</tr>
<tr>
<td>Recoverability</td>
<td>Include mechanism to recover from errors</td>
</tr>
<tr>
<td>User guidance</td>
<td>Provide context sensitive user help</td>
</tr>
<tr>
<td>User diversity</td>
<td>Provide appropriate interaction facilities for different types of user</td>
</tr>
</tbody>
</table>

Design principles

- User familiarity
  The interface should be based on user-oriented terms and concepts rather than computer concepts. For example, an office system should use concepts such as letters, documents, folders etc. rather than directories, file identifiers, etc.
- 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
Design principles

- **Recoverability**
  The system should provide some resilience to user errors and allow the user to recover from errors. This might include an undo facility, confirmation of destructive actions, 'soft' deletes, etc.

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

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

User-system interaction

- **Two problems must be addressed in interactive systems design**
  - How should information from the user be provided to the computer system?
  - How should information from the computer system be presented to the user?

- **User interaction and information presentation may be integrated through a coherent framework such as a user interface metaphor**

Interaction styles

- **Direct manipulation**
- **Menu selection**
- **Form fill-in**
- **Command language**
- **Natural language**

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

Control panel interface
Menu selection

- 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 selection

- 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 selection

- Actions which involve logical conjunction (and) or disjunction (or) are awkward to represent.
- Menu selection is 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.

Form-based interface

<table>
<thead>
<tr>
<th>Title</th>
<th>ISBN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author</td>
<td>Price</td>
</tr>
<tr>
<td>Publisher</td>
<td>Publication date</td>
</tr>
<tr>
<td>Edition</td>
<td>Number of copies</td>
</tr>
<tr>
<td>Classification</td>
<td>Loan status</td>
</tr>
<tr>
<td>Date of purchase</td>
<td>Order status</td>
</tr>
</tbody>
</table>

Command interfaces

- User types commands to give instructions to the system.
- 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.

Problems with command interfaces

- Users have to learn and remember a command language, and thus unsuitable for occasional users.
- Users often make mistakes in command. 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

Information presentation

- It 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.

Information presentation

- View state
- View methods
- View modification messages
- Controller state
- Controller methods
- Model state
- Model methods
- User input
- Model queries and updates
- Model edits

Model-view-controller
Information presentation

- **Static information**
  - Initialized at the beginning of a session and remained unchanged during the session
  - May be either numeric or textual

- **Dynamic information**
  - Changes during a session and the changes must be communicated to the system user
  - 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?

Alternative information presentations

Analogue vs. digital presentation

- **Digital presentation**
  - Compact - takes up little screen space
  - Precise values can be communicated

- **Analogue 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

Displaying relative values
Textual highlighting

The filename you have chosen has been used. Please choose another name.

Ch. 16 User interface design

OK  Cancel

Data visualisation

- Concerned with techniques for displaying large amounts of information
- Visualisation can reveal relationships between entities and trends in the data
- Possible data visualisations are:
  - Weather information collected from a number of sources
  - The state of a telephone network as a linked set of nodes
  - Chemical plant visualized by showing pressures and temperatures in a linked set of tanks and pipes
  - A model of a molecule displayed in 3 dimensions
  - Web pages displayed as a hyperbolic tree

Color displays

- Color adds an extra dimension to an interface and can help the user understand complex information structures
- Can be used to highlight exceptional events
- Common mistakes in the use of color in interface design include:
  - The use of color to communicate meaning
  - Over-use of color in the display

Color use guidelines

- Don't use too many colors
- Use color coding to support user tasks
- Allow users to control color coding
- Design for monochrome then add color
- Use color coding consistently
- Avoid color pairings which clash
- Use color change to show status change
- Be aware that color displays are usually lower resolution

User support

- User guidance covers all system facilities to support users, including on-line help, error messages, manuals etc.
- The user guidance system should be integrated with the user interface to help users when they need information about the system or when they make some kind of error
- The help and message system should, if possible, be integrated

Help and message system

- Application
- Error message system
- Help interface
- Message presentation system
- Help frames
- Error message texts
Error messages

- Error message design is critically important. Poor error messages can mean that a user rejects rather than accepts a system.
- Messages should be polite, concise, consistent, and constructive.
- The background and experience of users should be the determining factor in message design.

Design factors in message wording

<table>
<thead>
<tr>
<th>factor</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context</td>
<td>The system should be aware of what the user is doing and adjust the output message to the current context.</td>
</tr>
<tr>
<td>Experience</td>
<td>Provide different messages for users with different levels of experience.</td>
</tr>
<tr>
<td>Skill level</td>
<td>Provide different messages for users with different skill levels.</td>
</tr>
<tr>
<td>Style</td>
<td>Messages should be positive, proactive, and professional.</td>
</tr>
<tr>
<td>Culture</td>
<td>Messages must be suitable for the culture of the users.</td>
</tr>
</tbody>
</table>

Nurse input of a patient’s name

Please type the patient name in the box then click on OK.

Bates, J.

OK Cancel

System and user-oriented error messages

System-oriented error message

Error #27
Invalid patient id entered

OK Cancel

User-oriented error message

Patient J. Bates is not registered
Click on Patients for a list of registered patients.
Click on Retry to re-input a patient name.
Click on Help for more information.

Help system design

- Help? means “help I want information”.
- Help! means “HELP. I’m in trouble.”
- Both of these requirements have to be taken into account in help system design.
- Different facilities in the help system may be required.

Help information

- Should not simply be an on-line manual.
- Screens or windows don’t map well onto paper pages.
- The dynamic characteristics of the display can improve information presentation.
- People are not so good at reading screen as they are text.
**Help system use**

- Multiple entry points should be provided so that the user can get into the help system from different places.
- Some indication of where the user is positioned in the help system is valuable.
- Facilities should be provided to allow the user to navigate and traverse the help system.

**Entry points to a help system**

- Top-level entry
- Entry from application
- Entry from error message system

**Help system windows**

- Help frame map
- Mail redirection

**Mail redirection**

Mail may be redirected to another network user by pressing the redirect button in the control panel. The system asks for the name of the user or users to whom the mail has been sent.

**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.

**User document types**

- System evaluator's manual
- System administrator's manual
- Novice user's manual
- Experienced user's manual

**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>attributes</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 errors?</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 of a "gripe button" for on-line user feedback.

Key points

- Interface design should be user-centred.
- An interface should be logical and consistent and help users recover from errors.
- Interaction styles include direct manipulation, menu systems, form fill-in, command languages, and natural language.
- Graphical displays should be used to present trends and approximate values. Digital displays when precision is required.
- Color should be used sparingly and consistently.

Key points (continued)

- Systems should provide on-line help. This should include “help, I’m in trouble” and “help, I want information”.
- Error messages should be positive rather than negative.
- A range of different types of user documents should be provided.
- Ideally, a user interface should be evaluated against a usability specification.