

---

# **Software Engineering**

# **User Interface Design**

# User interface design

---

- Designing effective interfaces for software systems

# Objectives

---

- To suggest some general design principles for user interface design
- To explain different interaction styles
- To introduce styles of information presentation
- To describe the user support which should be built-in to user interfaces
- To introduce usability attributes and system approaches to system 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**

# User Interfaces'

---

- Legacy text-based interfaces;
  - old trend
- **Graphical User Interfaces 'GUI':**
  - current trend

# Graphical User Interfaces 'GUI'

## characteristics

---

<b>Characteristic</b>	<b>Description</b>
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 others, icons represent 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 interest in a window.
Graphics	Graphical elements can be mixed with text on the same display.

# 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

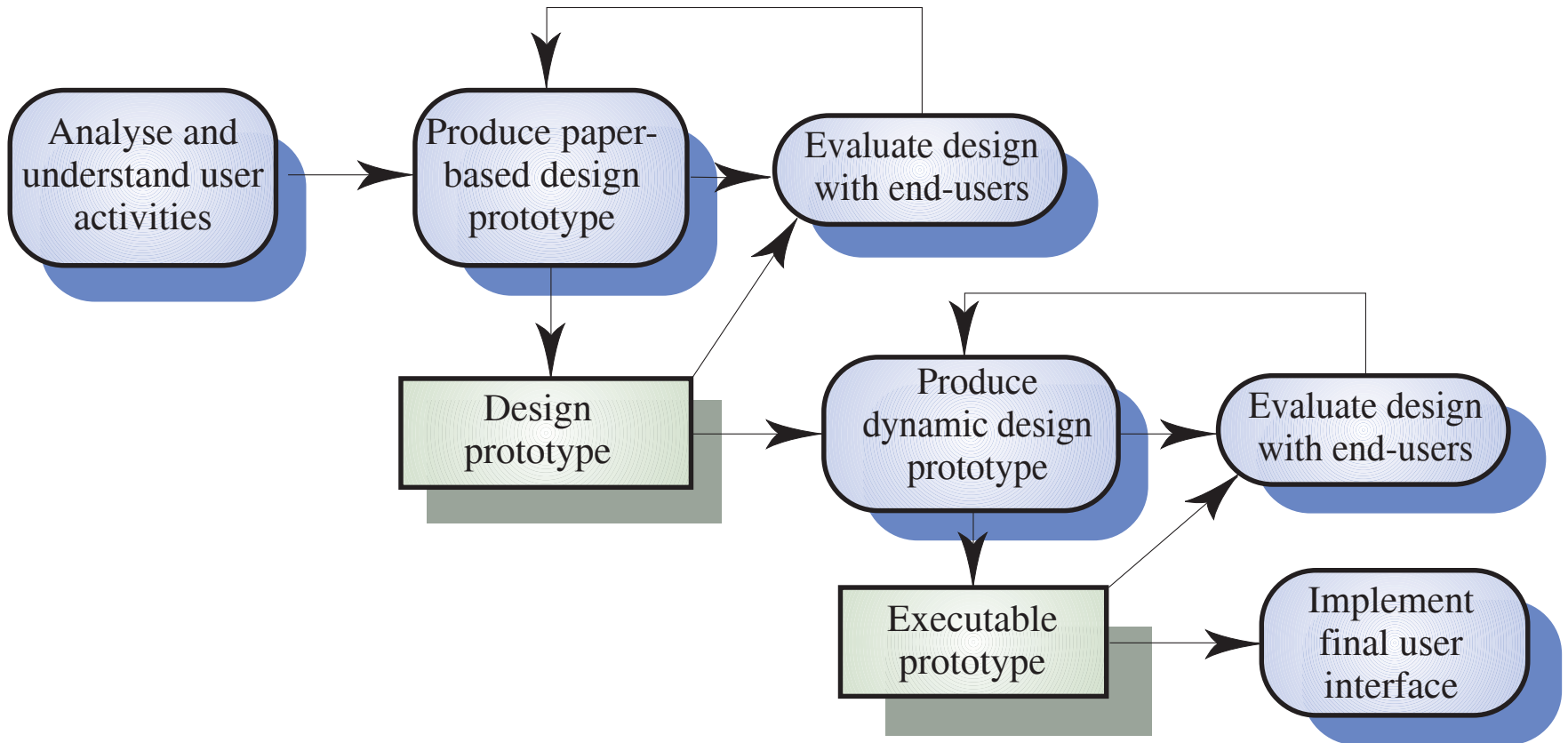
---

- The aim of this chapter is to sensitise 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 account of 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 recognise that people make mistakes
- UI design principles underlie interface designs although not all principles are applicable to all designs

# User interface design principles

---

<b>Principle</b>	<b>Description</b>
User familiarity	The interface should use terms and concepts which are drawn from the experience of the people who will make most use of the system.
Consistency	The interface should be consistent in that, wherever possible, comparable operations should be activated in the same way.
Minimal surprise	Users should never be surprised by the behaviour of a system.
Recoverability	The interface should include mechanisms to allow users to recover from errors.
User guidance	The interface should provide meaningful feedback when errors occur and provide context-sensitive user help facilities.
User diversity	The interface should provide appropriate interaction facilities for different types of system user.

# 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
  - 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
  - Allow the user to **recover from errors**.
  - 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 seeing difficulties and so **larger text** should be available

# User-system interaction

---

Two problems must be addressed in interactive systems design

- **User Input:** How should information from the user be provided to the computer system?
- **System Output:** How should information from the computer system be presented to the user?

# Interaction styles

---

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

# Advantages and disadvantages of Interaction styles

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



# Direct manipulation advantages

---

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

# Control panel interface

---

The control panel interface consists of the following elements:

- Title:** A text input field containing "JSD. example".
- Method:** A text input field containing "JSD".
- Type:** A text input field containing "Network".
- Selection:** A text input field containing "Process".
- Grid:** A checkbox with a small square icon next to the label "Grid".
- Units:** A dropdown menu currently showing "cm" with a right-pointing arrow.
- Reduce:** A dropdown menu currently showing "Full" with a right-pointing arrow.
- Busy:** A black rectangular indicator with the word "Busy" in white text.
- Buttons:** A row of five buttons labeled "NODE", "LINKS", "FONT", "LABEL", and "EDIT".
- Buttons:** Two larger buttons labeled "OUT" and "PRINT" stacked vertically on the right side.

# Menu systems

---

- Users make a selection from a list
- 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 which involve logical conjunction (and) or disjunction (or) are awkward to represent
- Experienced users find menus **slower** than command language

# Form-based interface

---

## NEW BOOK

Title

ISBN

Author

Price

Publisher

Publication  
date

Edition

Number of  
copies

Classification

Loan  
status

Date of  
purchase

Order  
status

# 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

# Problems with command interfaces

---

- Users have to **learn and remember** a command language. Command interfaces are therefore **unsuitable for occasional** users
- Users make **errors** 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

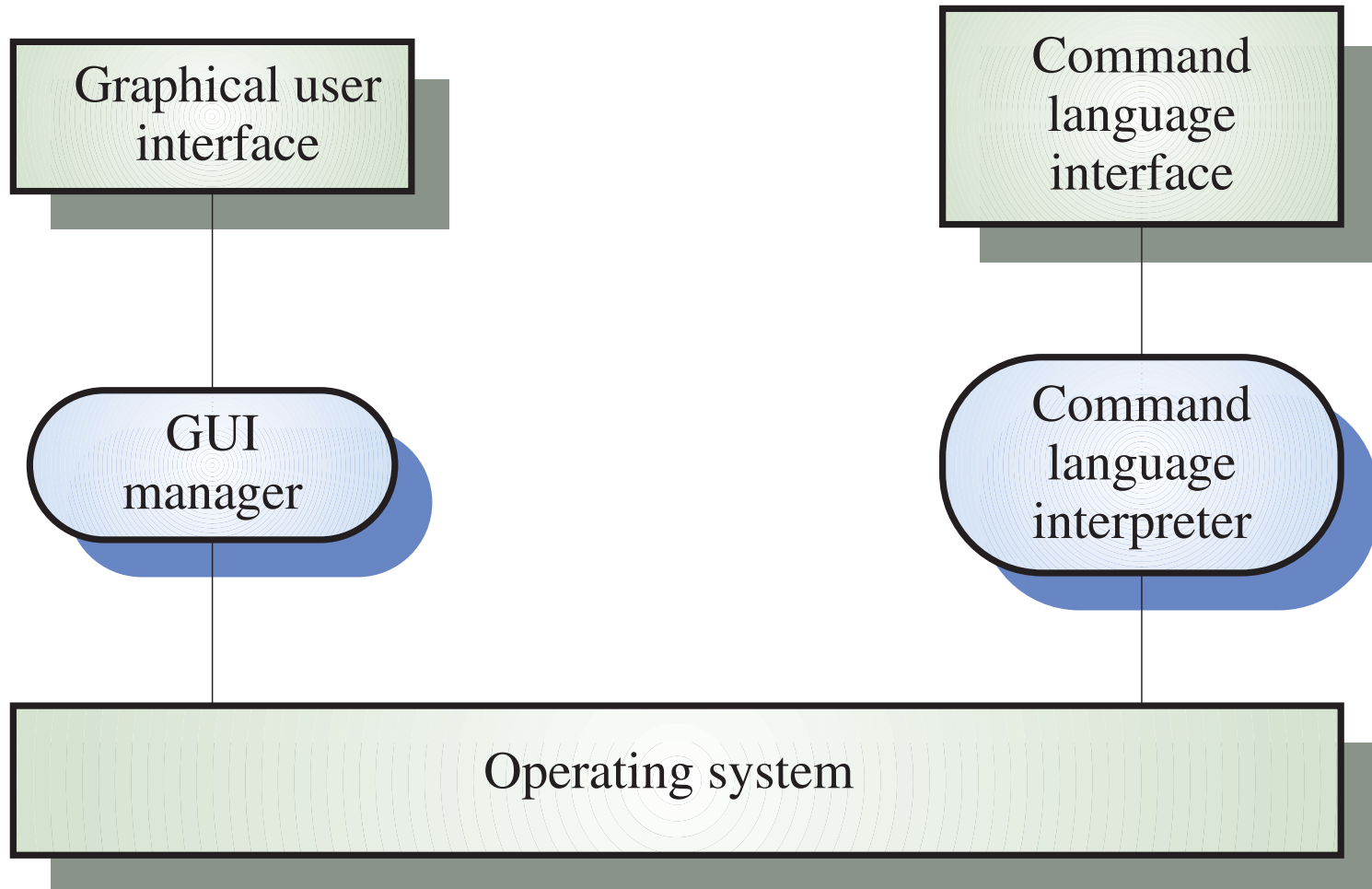
# Natural language interfaces

---

- 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

---



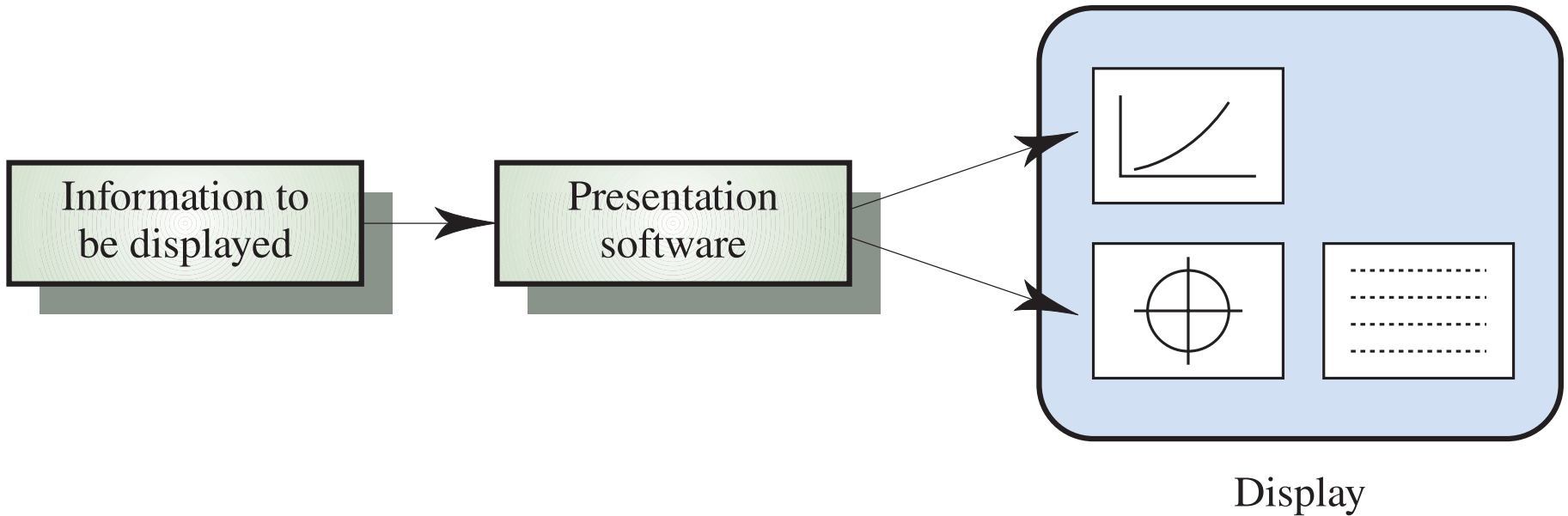
# Information presentation

---

- 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 (MVC)** approach is a way of supporting **multiple presentations of data**

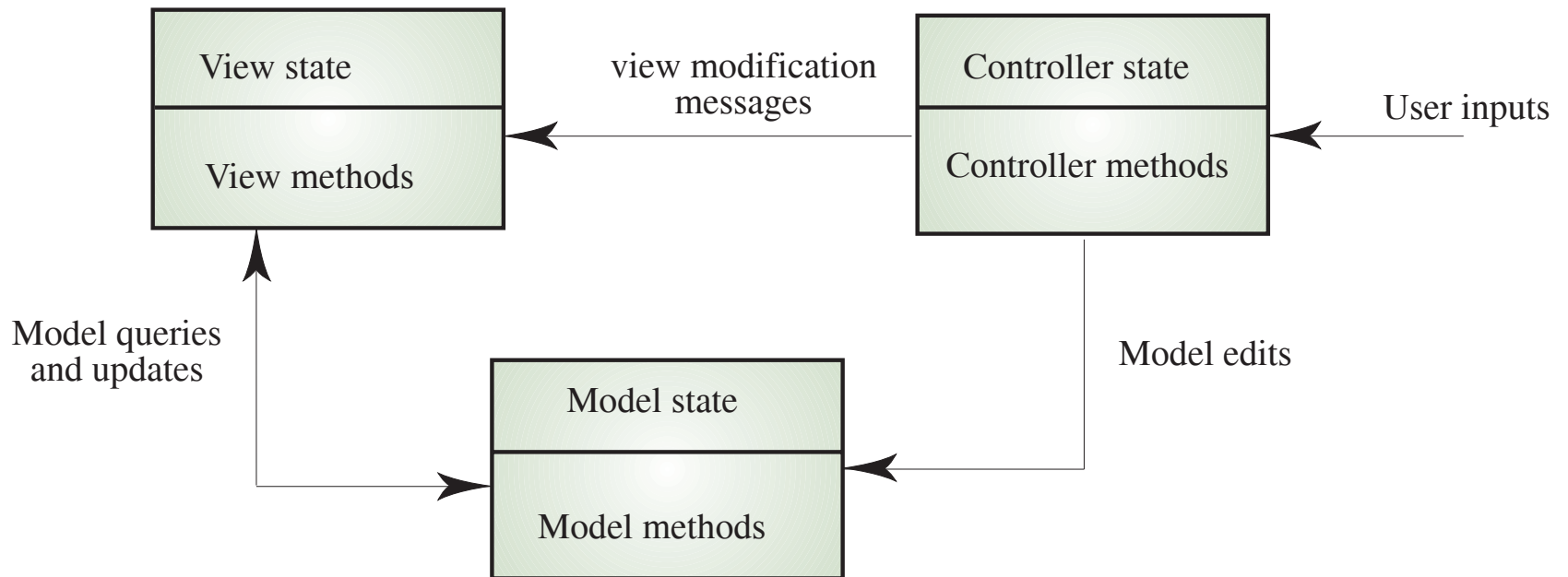
# Information presentation

---



# Model-View-Controller (MVC)

---



# Information presentation

---

- **Static information**
  - Initialised at the beginning of a session. It does not change 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

---

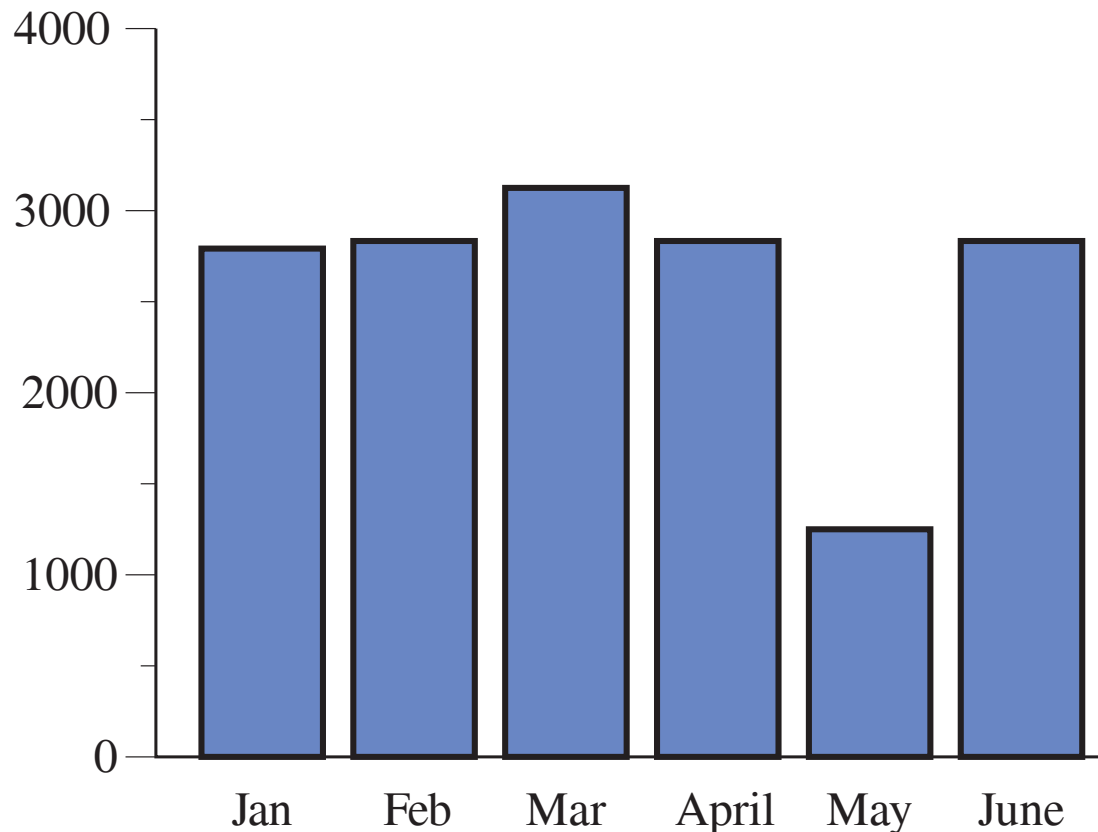
- Is the user interested in precise information or data relationships?
- How quickly do information values change? Must the change be indicated immediately?
- Must the user take some action in response to a change?
- Is there a direct manipulation interface?
- Is the information textual or numeric? Are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



# Alternative information presentations

---

Jan	Feb	Mar	April	May	June
2842	2851	3164	2789	1273	2835



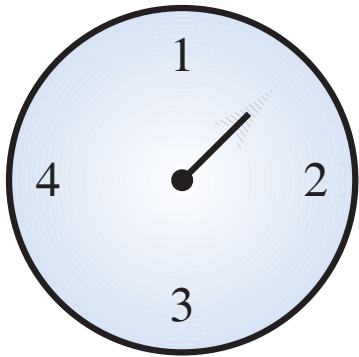
# Analogue vs. digital presentation

---

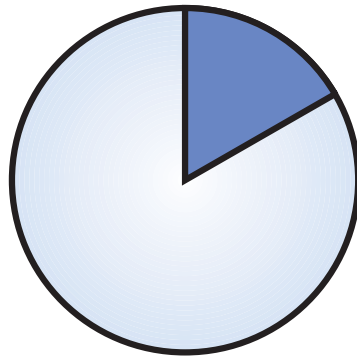
- Digital presentation
  - Compact - takes up little screen space
  - Precise values can be communicated
- Analogue presentation
  - Easier to get an 'at a glance' impression of a value
  - Possible to show relative values
  - Easier to see exceptional data values

# Dynamic information display

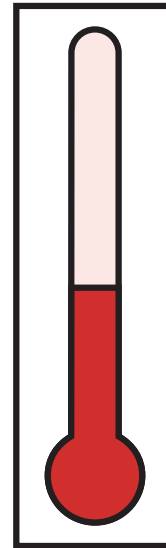
---



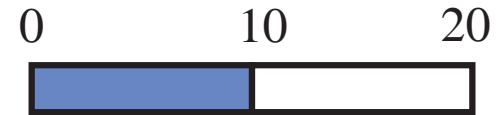
Dial with needle



Pie chart



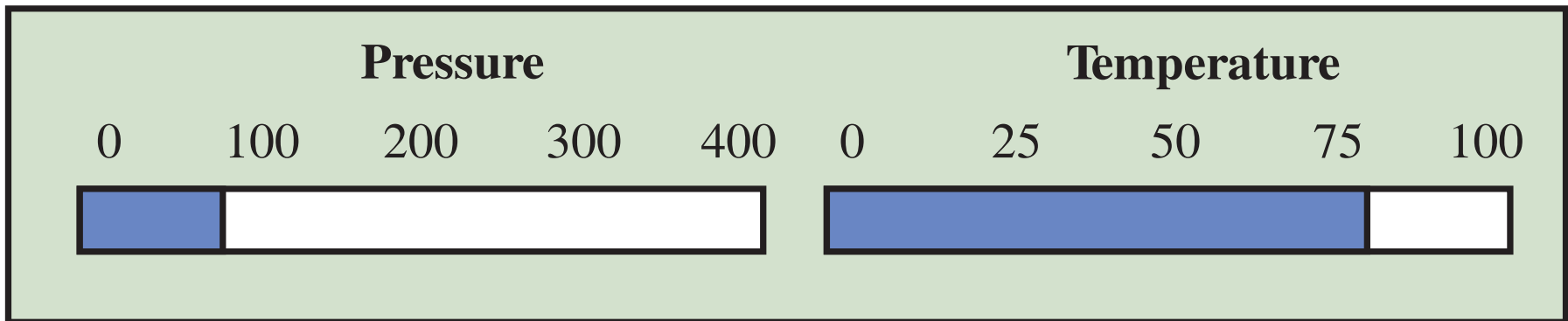
Thermometer



Horizontal bar

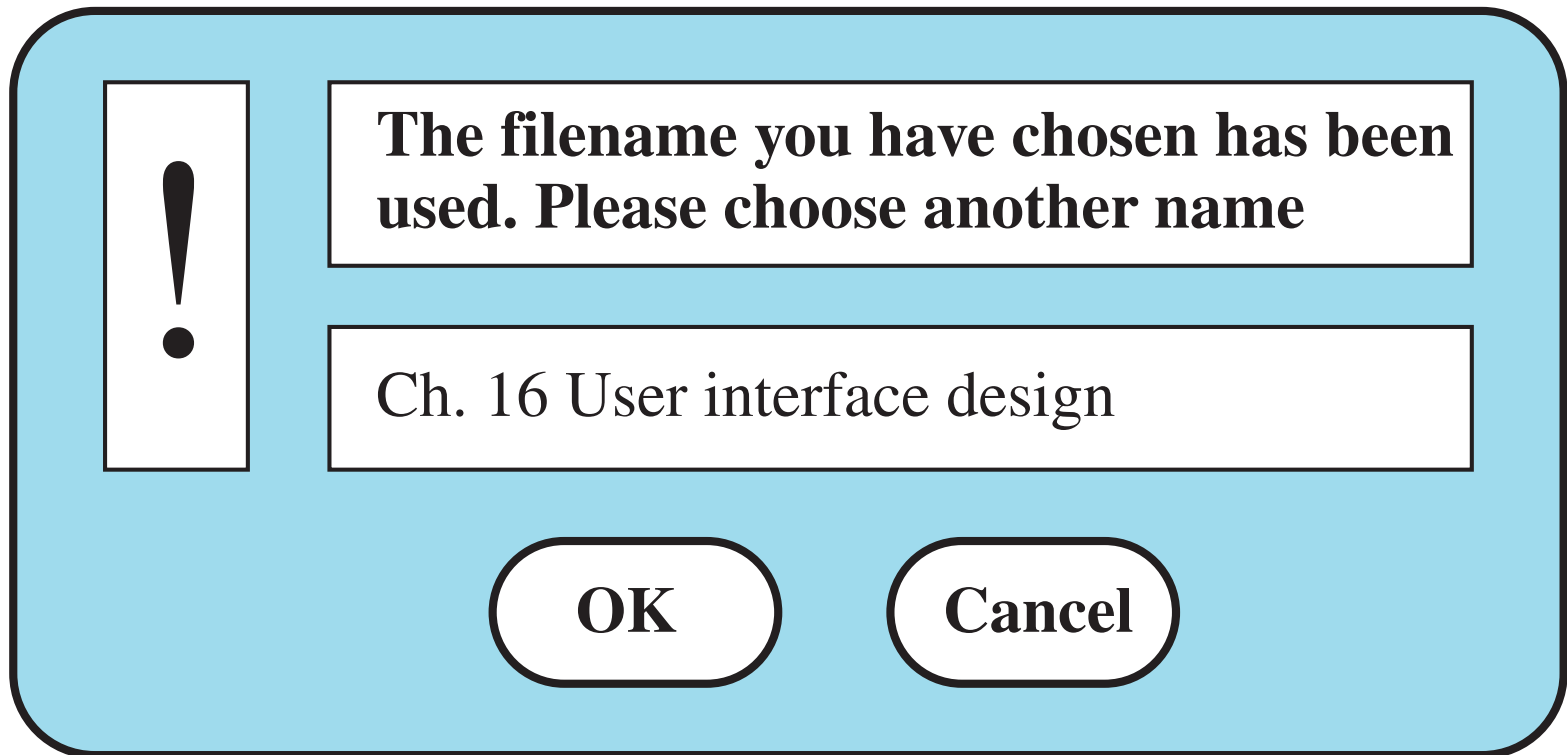
# Displaying relative values

---



# Textual highlighting

---



# 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 visualised 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

# Colour displays

---

- Colour 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 colour in interface design include:
  - The use of colour to communicate meaning
  - Over-use of colour in the display

# Colour use guidelines

---

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



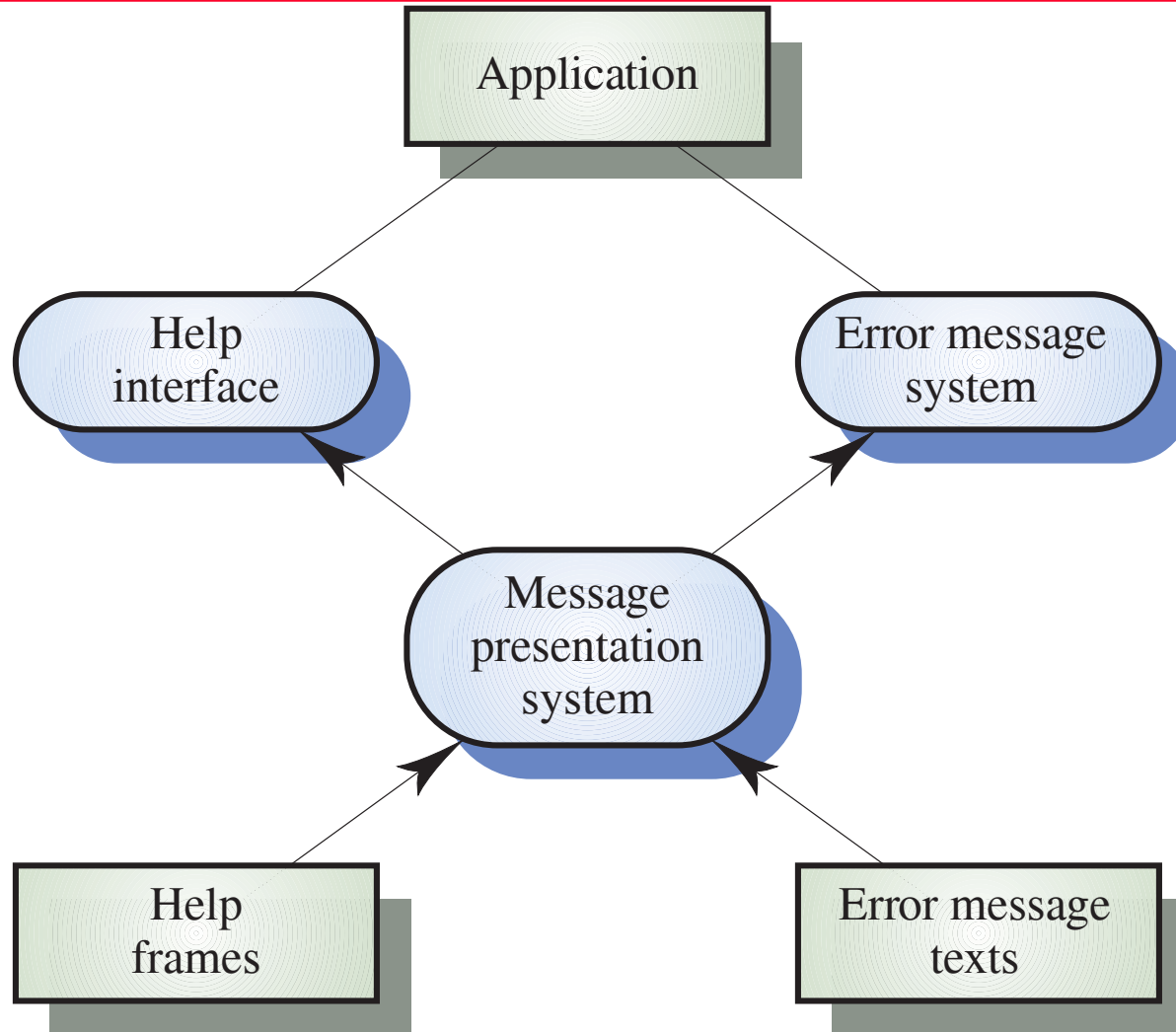
# User support

---

- 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

---



# 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

---

Context	The user guidance system should be aware of what the user is doing and should adjust the output message to the current context.
Experience	As users become familiar with a system they become irritated by long, 'meaningful' messages. However, beginners find it difficult to understand short terse statements of the problem. The user guidance system should provide both types of message and allow the user to control message conciseness.
Skill level	Messages should be tailored to the user's skills as well as their experience. Messages for the different classes of user may be expressed in different ways depending on the terminology which is familiar to the reader.
Style	Messages should be positive rather than negative. They should use the active rather than the passive mode of address. They should never be insulting or try to be funny.
Culture	Wherever possible, the designer of messages should be familiar with the culture of the country where the system is sold. There are distinct cultural differences between Europe, Asia and America. A suitable message for one culture might be unacceptable in another.

# Nurse input of a patient's name

---

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

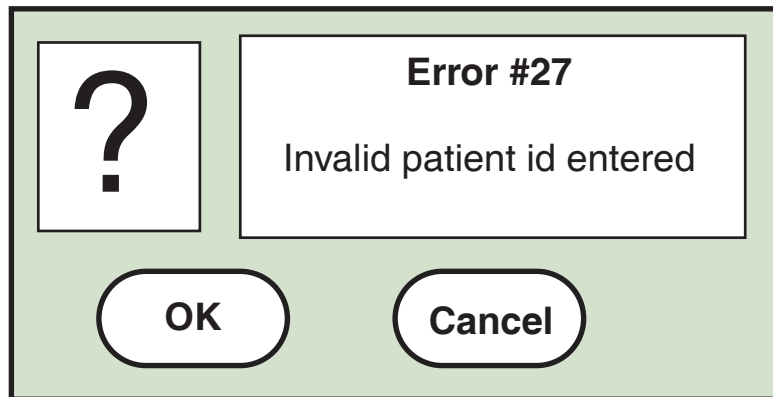
Bates, J.

OK Cancel

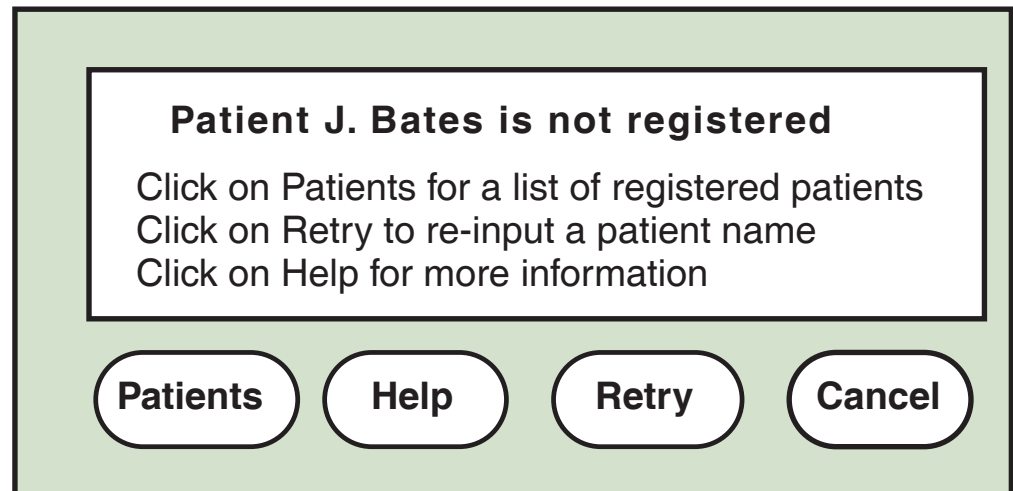
# System and user-oriented error messages

---

System-oriented error message



User-oriented error message



# 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.

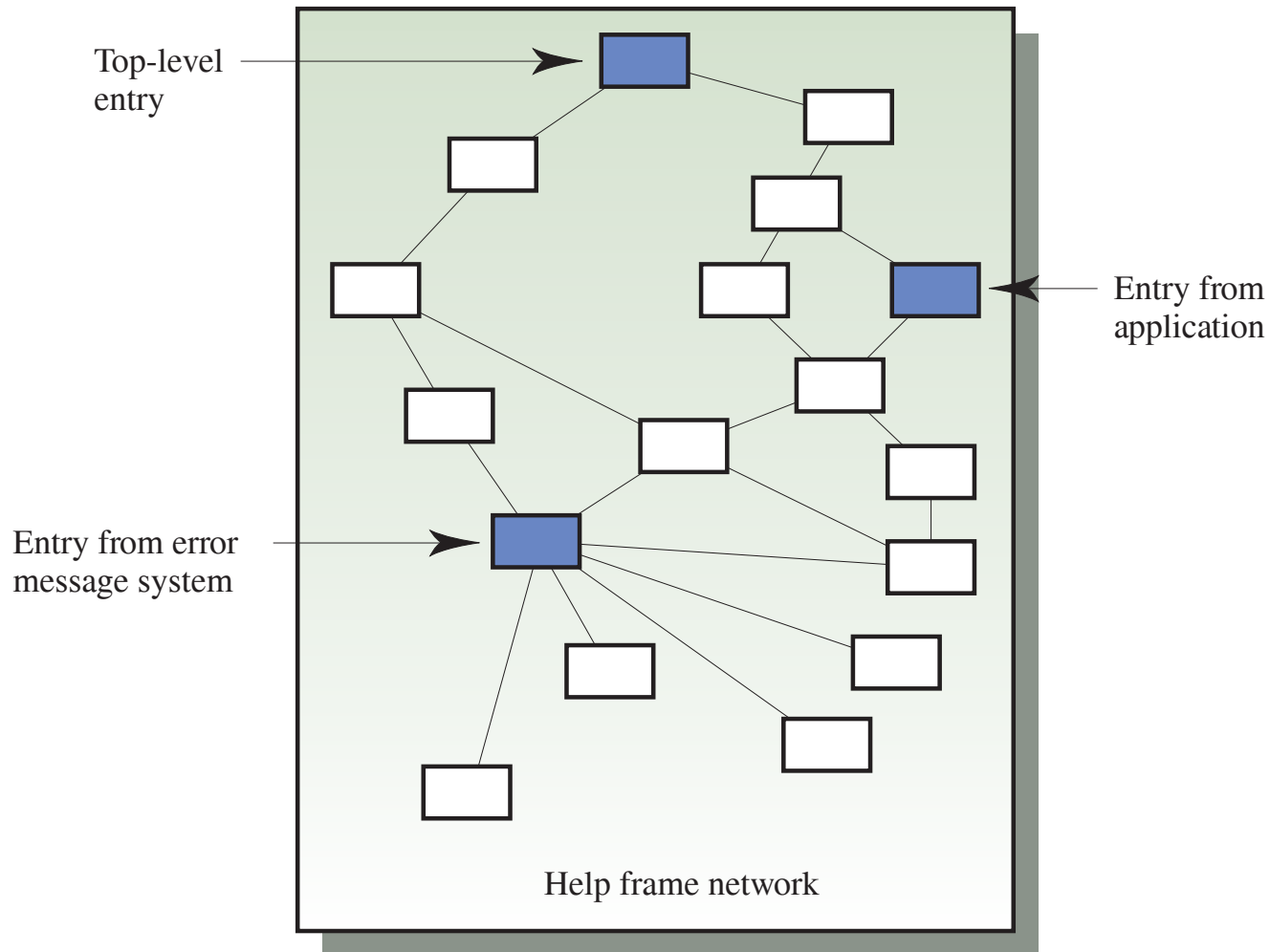


# Help system use

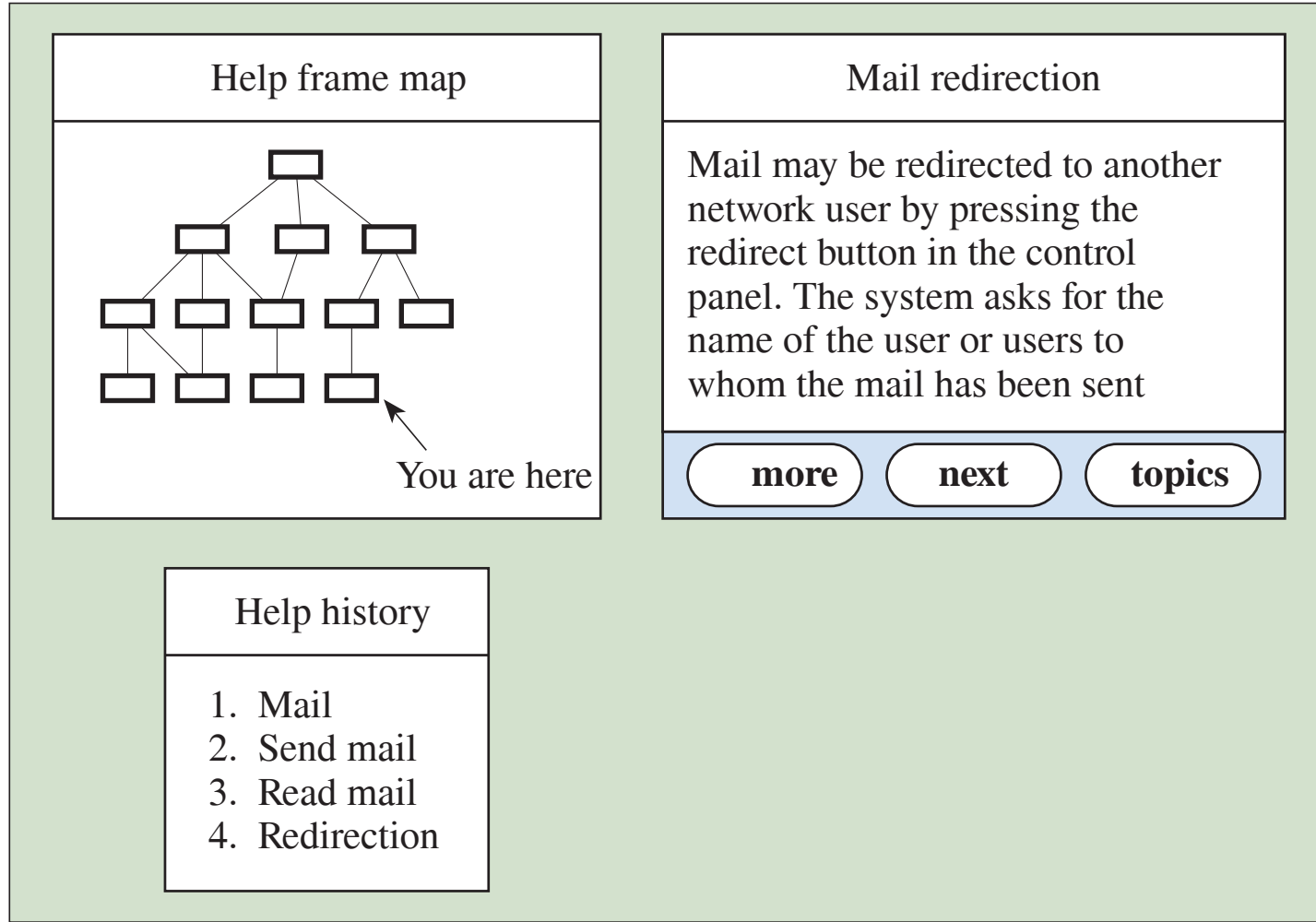
---

- 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



# Help system windows



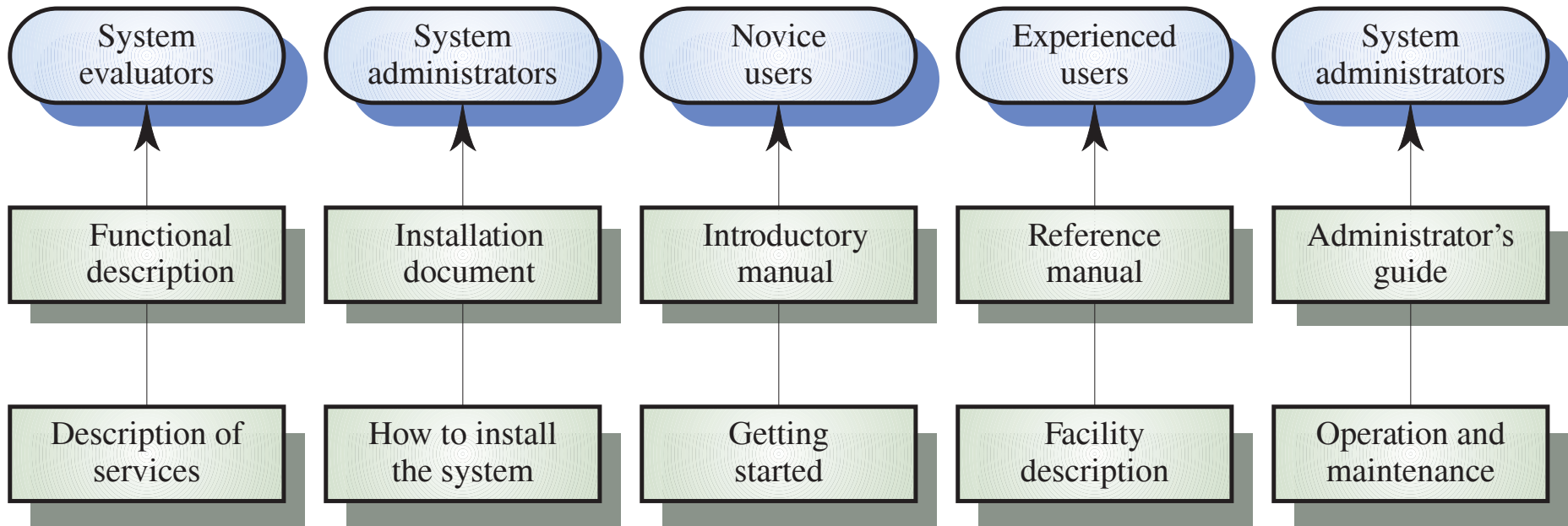
# User documentation

---

- 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

---



# 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

---

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



# 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 grip button for on-line user feedback.

# Important issues in the design of user interface

---

- 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
- Colour should be used sparingly and consistently

# Important issues in the design of user interface

---

- 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