

---

# User interface design

---

# The user interface

---

- Should be designed to match:
  - Skills, experience and expectations of its anticipated users.
- 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.

# Human factors in interface design

---

- Limited short-term memory
  - People can instantaneously remember about 7 items of information. If you present more than this, they are more liable to make mistakes.
- People make mistakes
  - When people make mistakes and systems go wrong, inappropriate alarms and messages can increase stress and hence the likelihood of more mistakes.
- People are different
  - People have a wide range of physical capabilities. Designers should not just design for their own capabilities.
- People have different interaction preferences
  - Some like pictures, some like text.

# UI design principles

---

- Take in account:
  - the needs, experience and capabilities of the system users.
- Be aware of people's physical and mental limitations (e.g. limited short-term memory)
- Recognise that people make mistakes.
- Note that: not all principles are applicable to all designs.

# User interface design principles

---

<b>Principle</b>	<b>Description</b>
<b>Minimize the number of steps</b>	The interface should use terms and concepts which are known from the experience of the people who will make most use of the system.
<b>Consistency</b>	The interface should be consistent in that, whenever possible, comparable operations should be conducted in the same way.
<b>Minimize errors</b>	Errors should never be required by the behaviour of a system.
<b>Recoverability</b>	The interface should include mechanisms to allow users to recover from errors.
<b>Minimize the number of choices</b>	The interface should provide meaningful feedback when users make and provide context-sensitive error help facilities.
<b>Minimize the number of screens</b>	The interface should provide appropriate interaction facilities for different types of system use.

# Design issues in UIs

---

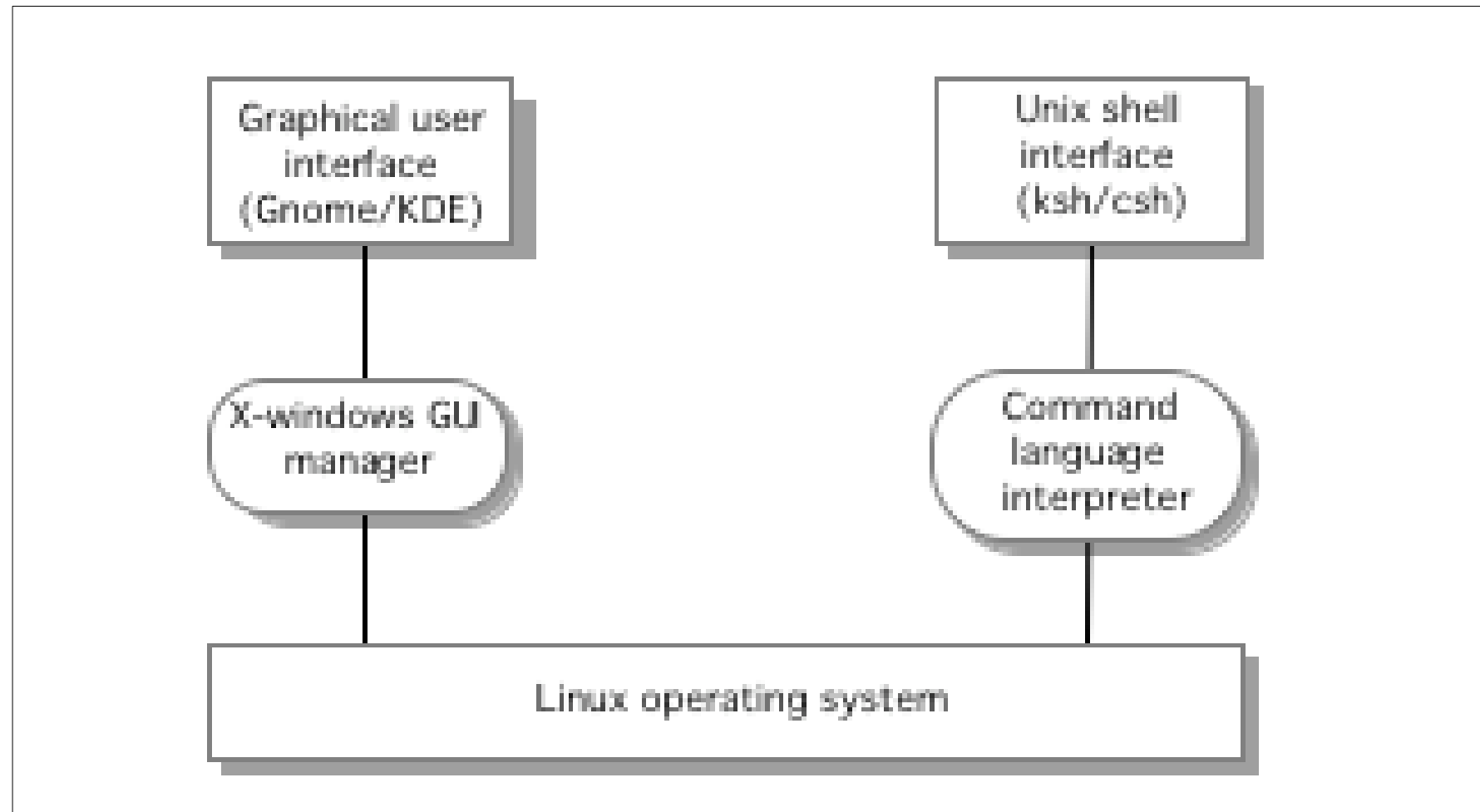
- Two problems must be addressed in interactive systems design
  - How should information from the user be provided to the computer system?
  - How should information from the computer system be presented to the user?

# Interaction styles

Interaction style	Main advantages	Main disadvantages	Application examples
Direct manipulation	Fast and intuitive interaction Easy to learn	May be hard to implement. Only suitable where there is a clear metaphor for tasks and objects.	Office games WWW systems
Menu selection	Simple user error Little typing required	Slow for experienced users. More screen real estate is used by menu systems.	Most general-purpose systems
Point-and-click	Simple data entry Easy to learn Flexible	Slower up a lot of screen space. Screen real estate where menu systems do not control the focus shift.	Small control, focused text processing
Command language	Command and flexible	Hard to learn. More error management.	Operating systems, command and control systems
Natural language	Flexible to novel users Easily extended	Requires more typing. Natural language understanding systems are available.	Interactive natural systems

# Multiple user interfaces

---





# Information presentation

---

- Information presentation is concerned with presenting information to users.
  - Direct presentation  
(e.g. text in a word processor)
  - Indirect presentation  
(e.g. in some graphical form).
- The Model-View-Controller approach is a way of supporting multiple presentations of data.

# 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.

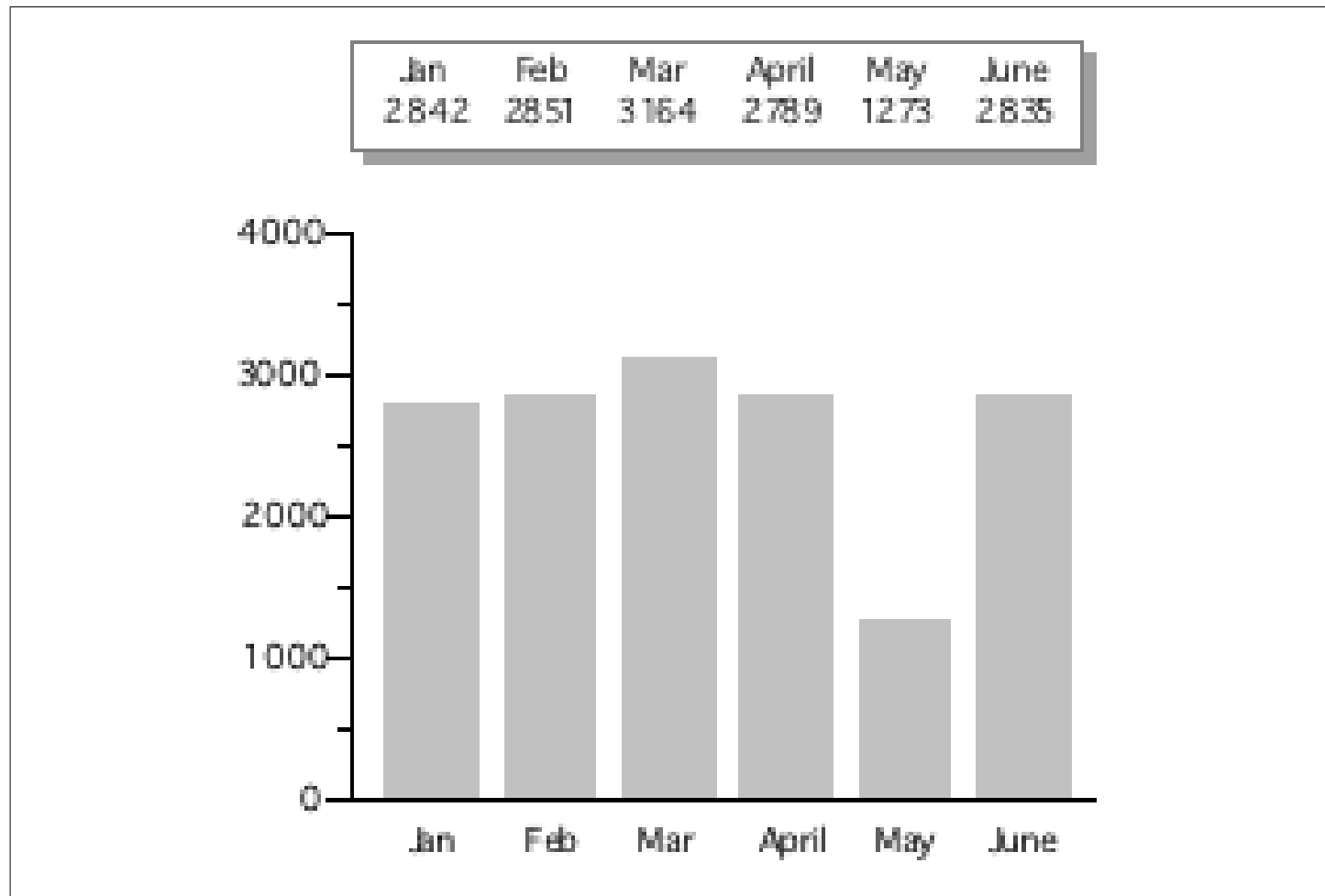
# Information display factors

---

- 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

---



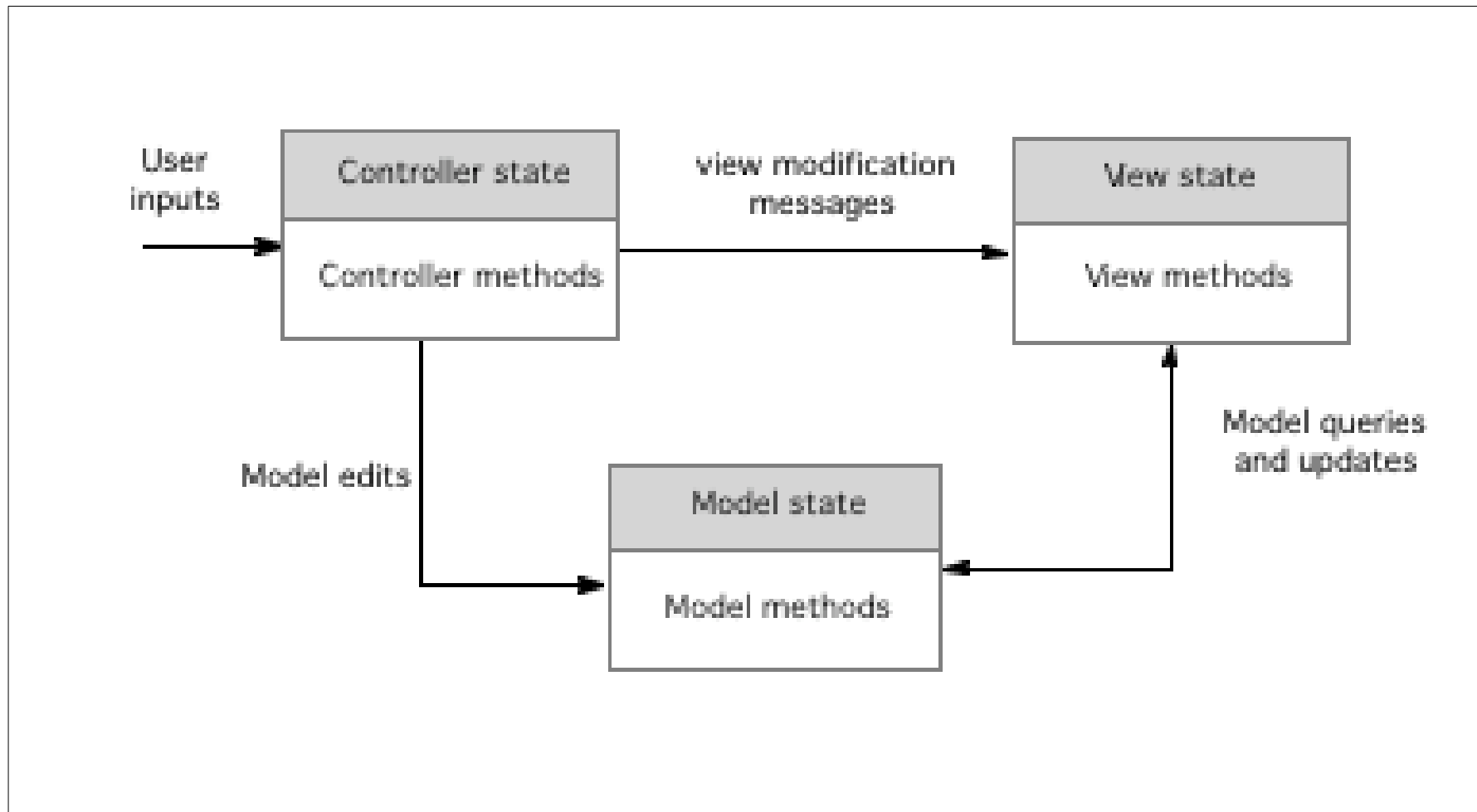
# Analogue or digital presentation?

---

- 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.

# Model-view-controller

---



# Data visualisation

---

- 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.
- Colour 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;
  - The over-use of colour in the display.



# Colour use guidelines

---

- Limit the number of colours used and be conservative in their use.
- Use colour change to show a change in system status.
- Use colour coding to support the task that users are trying to perform.
- Use colour coding in a thoughtful and consistent way.
- Be careful about colour pairings.

# 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

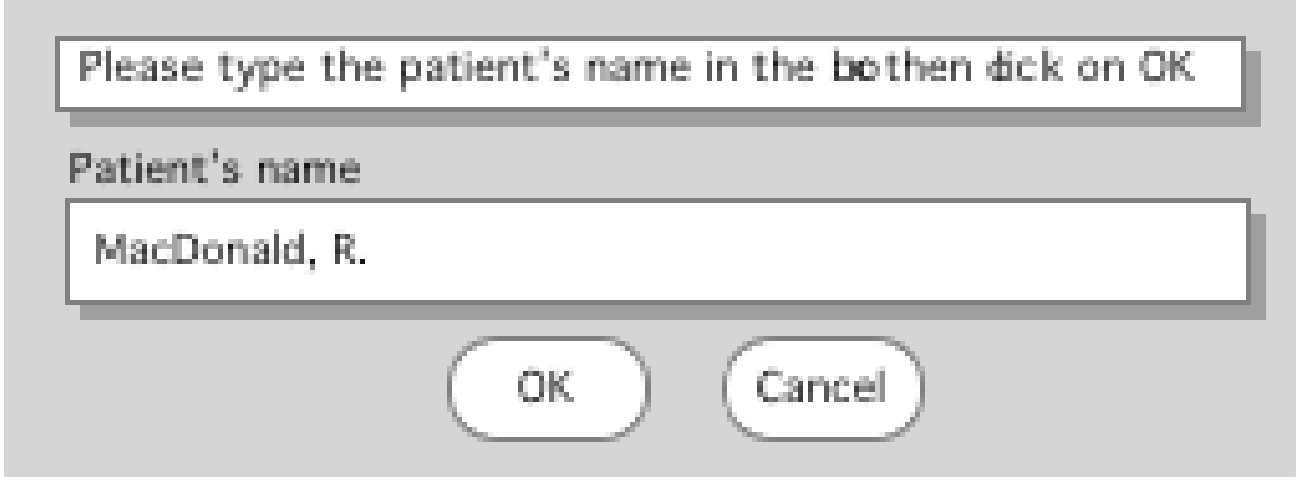
---

Factor	Description
Content	Whenever possible, the messages generated by the system should reflect the current user content. As far as is possible, the system should be aware of what the user is doing and should generate messages that are relevant to their current activity.
Regulation	As a user becomes familiar with a system they become habituated by long, 'uninterrupted' messages. However, designers need to think about what these statements are a problem. One should provide both types of message and allow the user to control message presentation.
Visual 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 technology that is familiar to the user.
Style	Messages should be positive rather than negative. They should use the active rather than the passive voice as often. They should never be leading or try to be funny.
Custom	Whenever possible, the design of messages should be familiar with the culture of the country where the system is used. There are distinct cultural differences between Europe, Asia and America. It is often messages for one culture might be unacceptable in another.

# User error

---

- Assume that a nurse misspells the name of a patient whose records he is trying to retrieve.



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

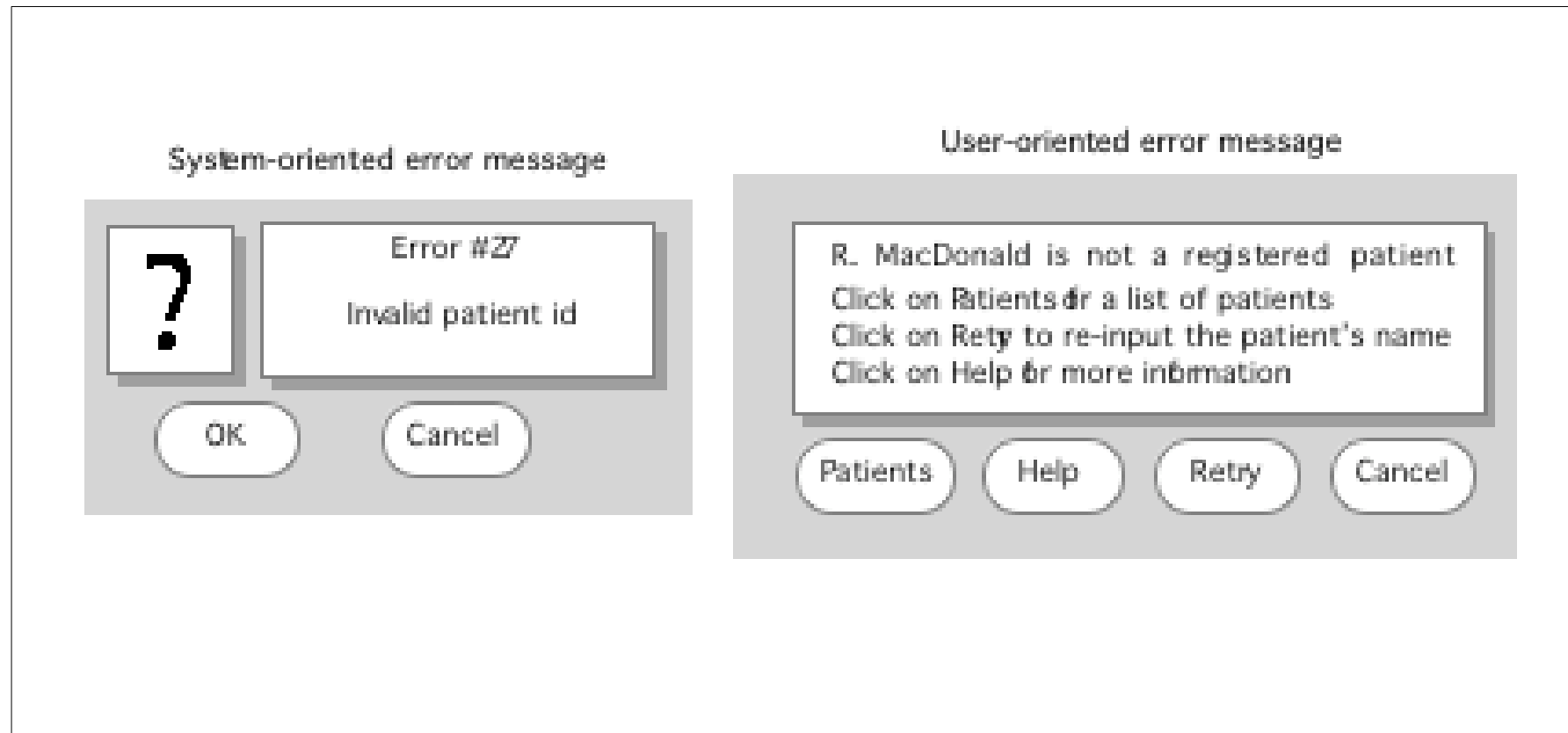
Patient's name

MacDonald, R.

OK Cancel

# Good and bad message design

---



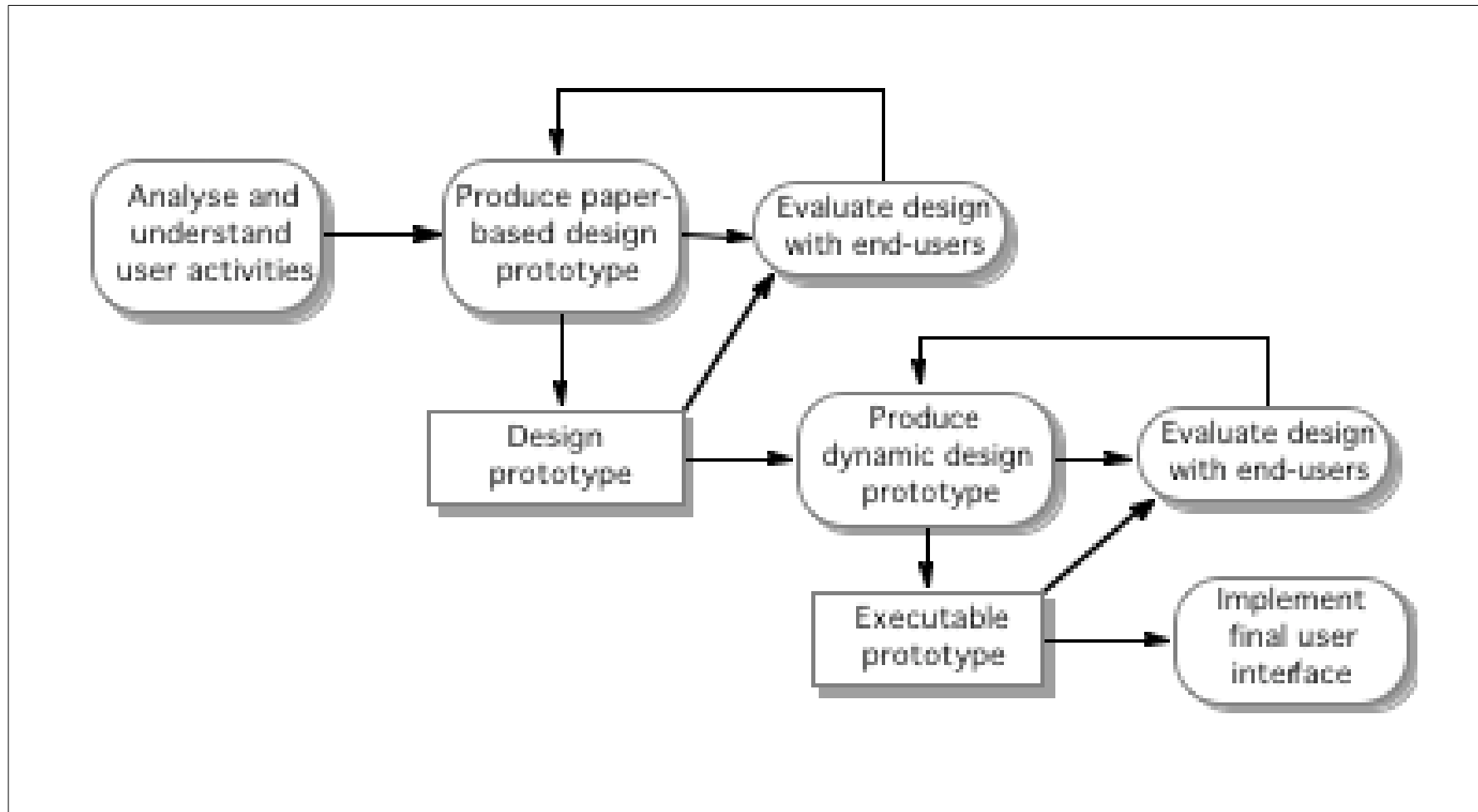
# The UI design process

---

- UI design is an iterative process involving close liaisons between users and designers.
- The 3 core activities in this process are:
  - User analysis. Understand what the users will do with the system;
  - System prototyping. Develop a series of prototypes for experiment;
  - Interface evaluation. Experiment with these prototypes with users.

# The design process

---



# User analysis

---

- If you don't understand what the users want to do with a system, you have no realistic prospect of designing an effective interface.
- User analyses have to be described in terms that users and other designers can understand.
- Scenarios where you describe typical episodes of use, are one way of describing these analyses.



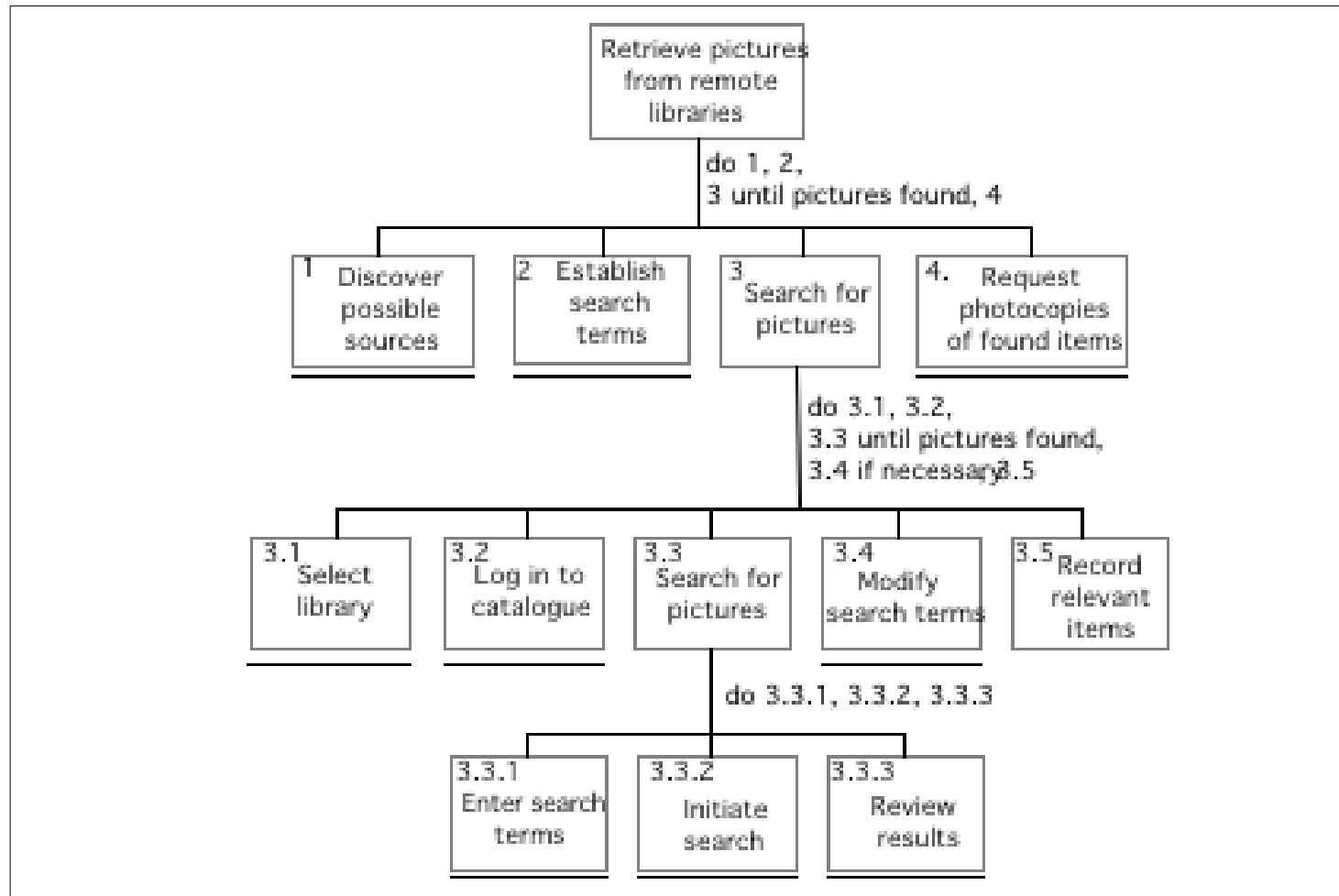
# Analysis techniques

---

- Task analysis
  - Models the steps involved in completing a task.
- Interviewing and questionnaires
  - Asks the users about the work they do.
- Ethnography
  - Observes the user at work.

# Hierarchical task analysis

---



# Interviewing

---

- Design semi-structured interviews based on open-ended questions.
- Users can then provide information that they think is essential; not just information that you have thought of collecting.
- Group interviews or focus groups allow users to discuss with each other what they do.

# Ethnography

---

- Involves an external observer watching users at work and questioning them in an unscripted way about their work.
- Valuable because many user tasks are intuitive and they find these very difficult to describe and explain.
- Also helps understand the role of social and organisational influences on work.

# User interface prototyping

---

- The aim of prototyping is to allow users to gain direct experience with the interface.
- Without such direct experience, it is impossible to judge the usability of an interface.
- Prototyping may be a two-stage process:
  - Early in the process, paper prototypes may be used; Work through scenarios using sketches of the interface
  - The design is then refined and increasingly sophisticated automated prototypes are then developed.

# Prototyping techniques

---

- Script-driven prototyping
  - Develop a set of scripts and screens using a tool such as Macromedia Director. When the user interacts with these, the screen changes to the next display.
- Visual programming
  - Use a language designed for rapid development such as Visual Basic. See Chapter 17.
- Internet-based prototyping
  - Use a web browser and associated scripts.

# User interface evaluation

---

- Full scale evaluation is very expensive and impractical for most systems.
- 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 code in the software to collect on-line user feedback.

# Usability attributes

---

Attribute	Description
Learnability	How long does it take a new user to become productive with the system?
Speed of operation	How well does the system support what the user's work requires?
Flexibility	How tolerant is the system of user error?
Memorability	How good is the system at supporting those who return?
Helpfulness	How clearly is the system itself a simple model of itself?