ICT 2402 Software Engineering

User Interface Design

Topics covered

- Design issues
- The user interface design process
- User analysis
- User interface prototyping
- Interface evaluation

User Interface Design

User → User → Computer
← Interface ← System

A system needs:

- Hardware design
- Software design
- Interface design

UI Design contd...

In most organizations:

- Hardware design ← specialist HW designers
- Software design
- Interface design

} ← software engineers

But, interface design requires:

- Software engineers
- Human factor specialists
  - HCI specialists
  - Interface Architects

The user interface

- User interfaces should be designed to match the skills, experience and expectations of its anticipated users.
- 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 Interface

Keep it simple!

Low usability → → user errors & misunderstandings → → data corruption

→ → users reject the system

UIs:

- Command line
- GUI – more popular for interactive systems

GUI Characteristics

- Windows
  - Display different information simultaneously
- Icons
  - Represent different data types
- Menus
  - Select commands
- Pointing devices
  - Select items
- Graphics
  - (mixed with text) – give information

Advantages of GUIs

- Easy to learn / use
- Windows help to switch tasks
- Pointing device gives access to any point on screen

Interface Design

Paper-based mock-up

↓

Screen-based design

- Simulates user interaction

User-centered design:
End-user involvement

- Evaluators
- Members of design team

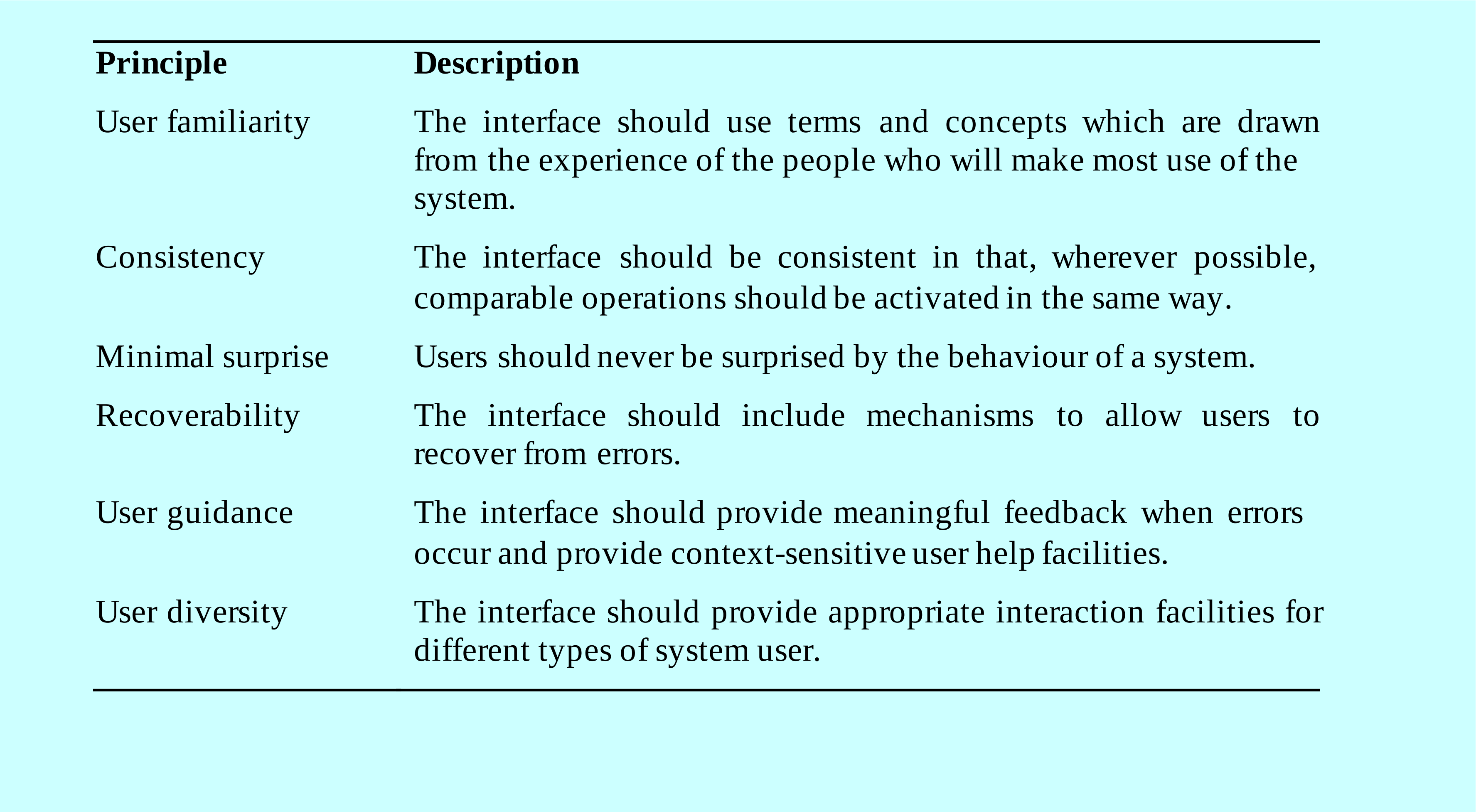
Human factors in interface design

- 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

- 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 recognize that people make mistakes.
- UI design principles underlie interface designs although not all principles are applicable to all designs.

User interface design principles



Principle	Description
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
  - 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

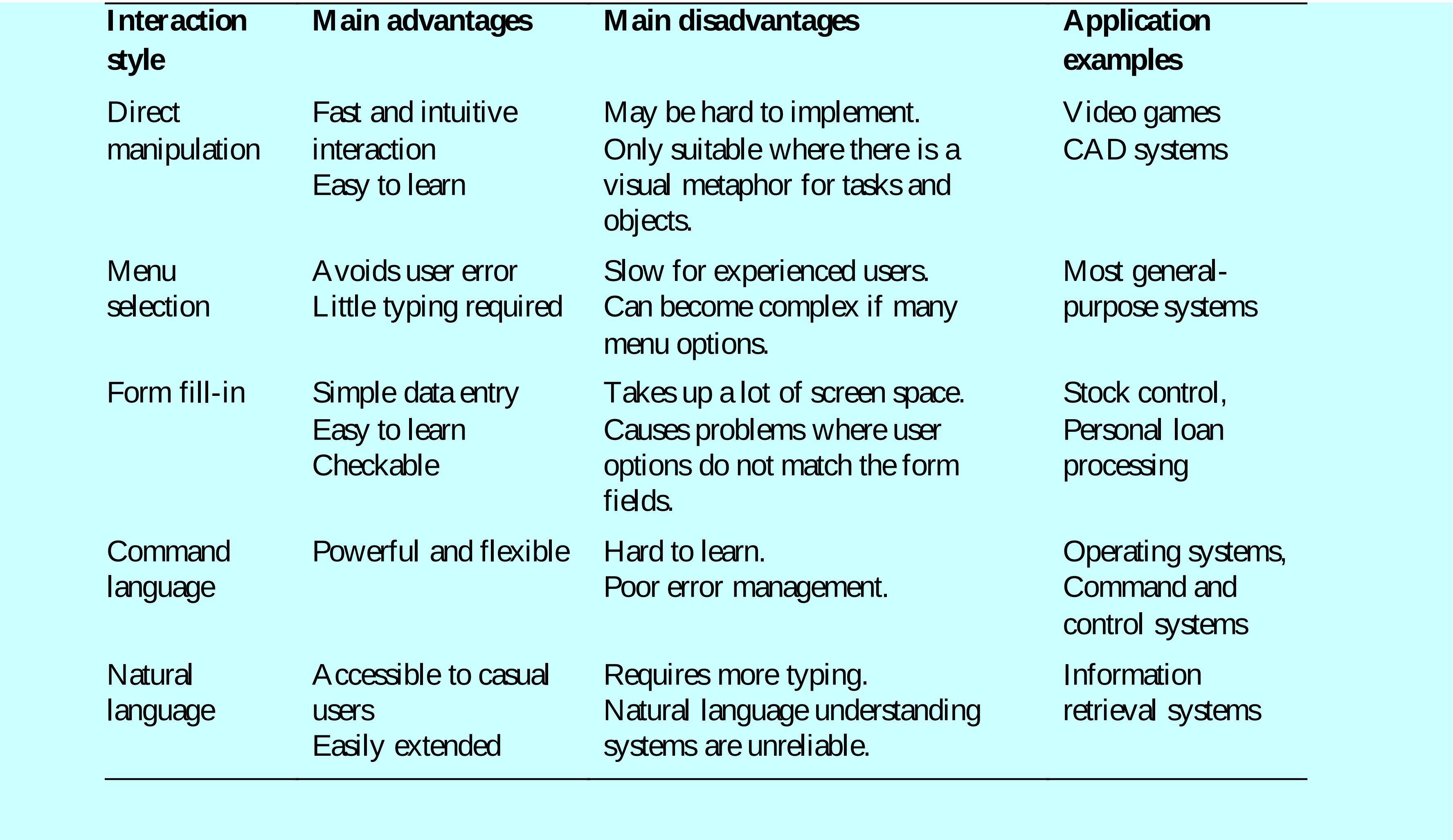
Design principles(Cont…)

- 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, checkpointing, 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

Interaction styles

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

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 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 Checkable	Takes up a lot of screen space. Causes problems where user options do not match the form fields.	Stock control, Personal loan processing
Command language	Powerful and flexible	Hard to learn. Poor error management.	Operating systems, Command and control systems
Natural language	Accessible to casual users Easily extended	Requires more typing. Natural language understanding systems are unreliable.	Information retrieval systems

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.

Information Presentation(contd...)

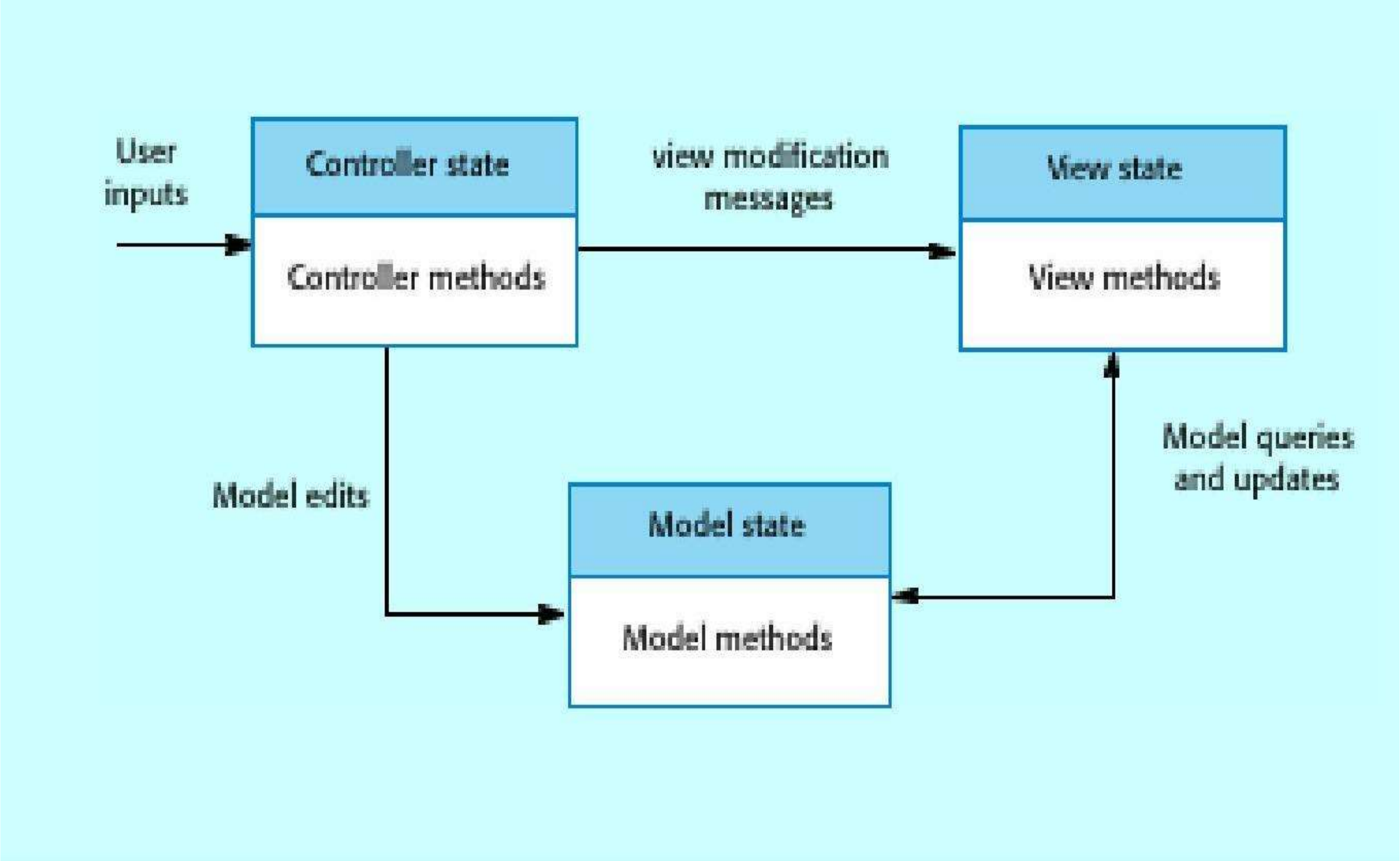
- Direct representation of input
  - E.g.: word processor

- Graphical representation

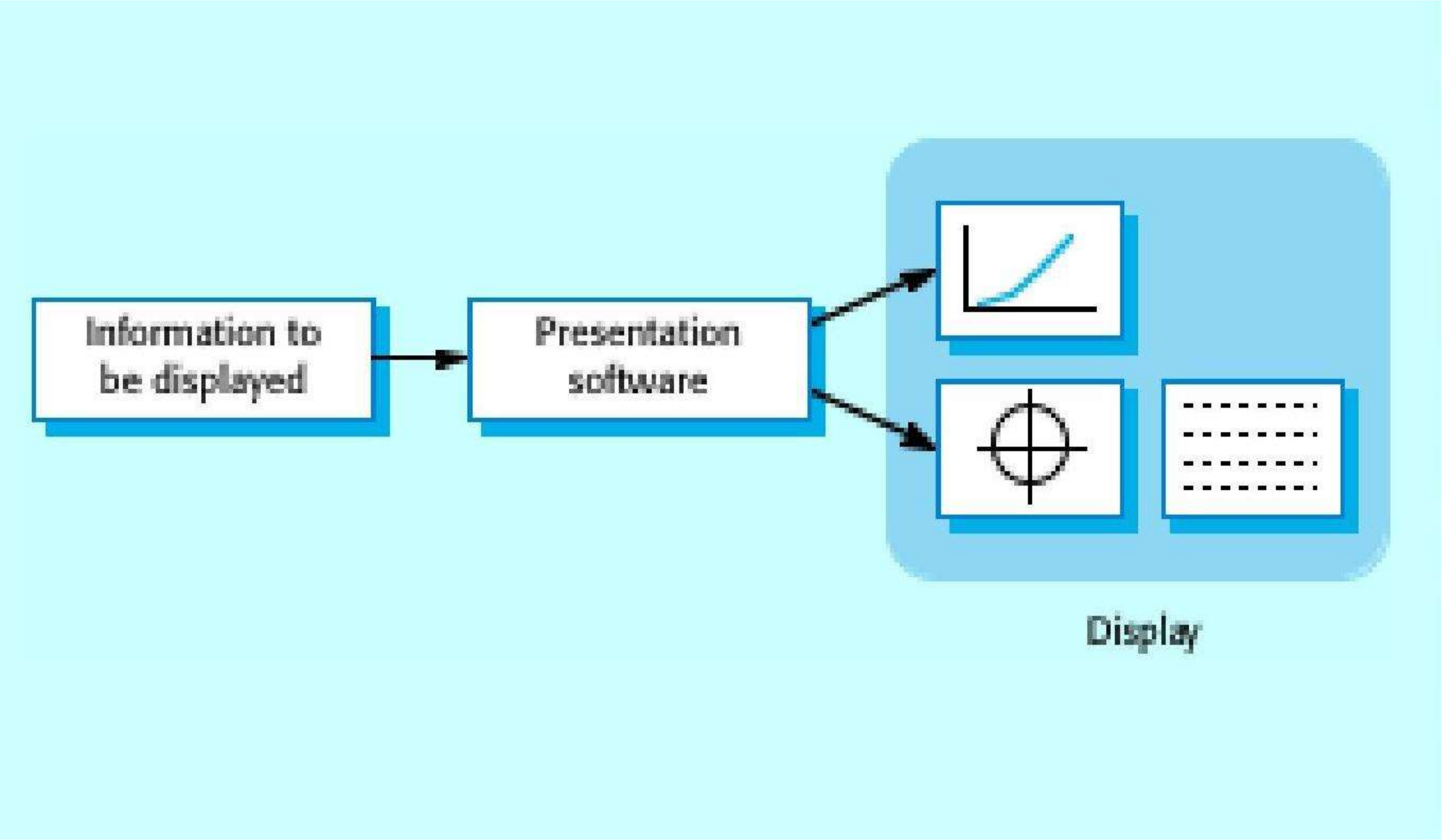
Consider:

  - Precise information or relationships
  - Dynamicity of values
  - User response to information changes
  - Textual or numerical information

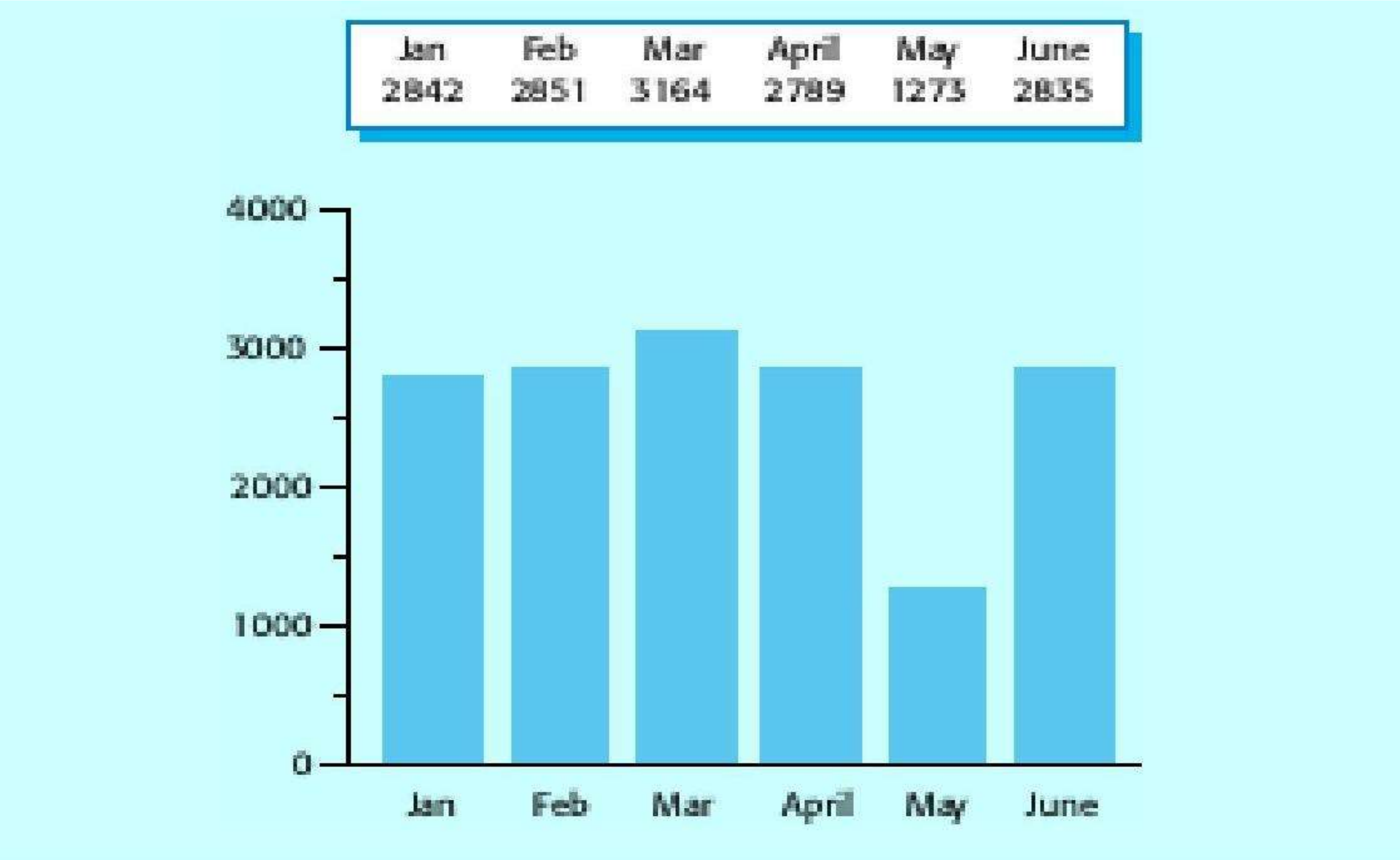
Model-View-Controller



Information presentation(contd…)



Alternative information presentations



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

Information Presentation (contd...)

Textual information:

- Less space
- Cannot read at-a-glance
- Not efficient for dynamic data
  - Bars and charts are more effective

Information presentation(contd…)

- Static information
  - Initialized 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 Presentation (contd...)

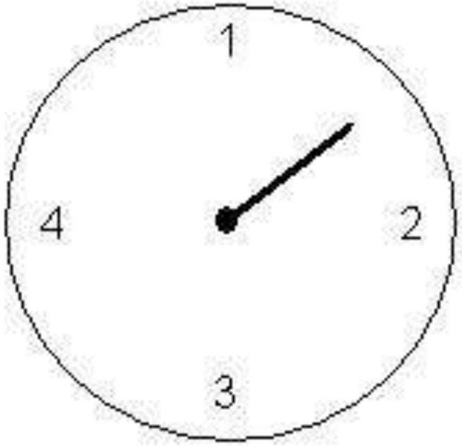
Static data:

- Associated icon
- Highlights
- Red lettering

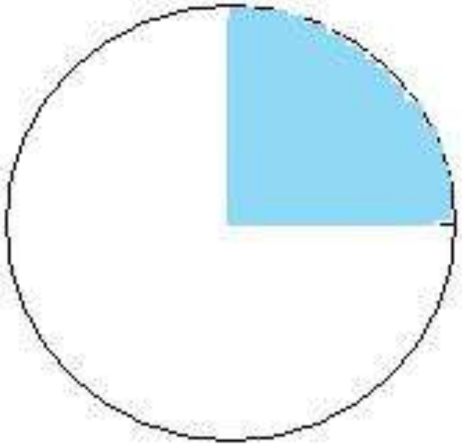
Dynamic data:

- Best represented graphically
- Analogue representation better than digital to show relative values

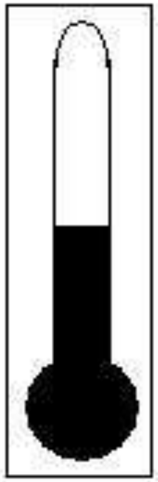
Representing Dynamic Information



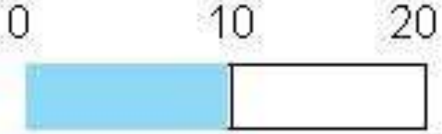
Dial with needle



Pie chart

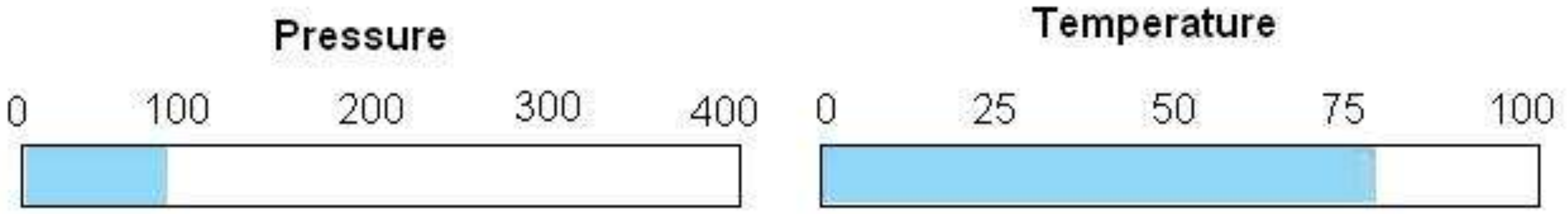


Thermometer

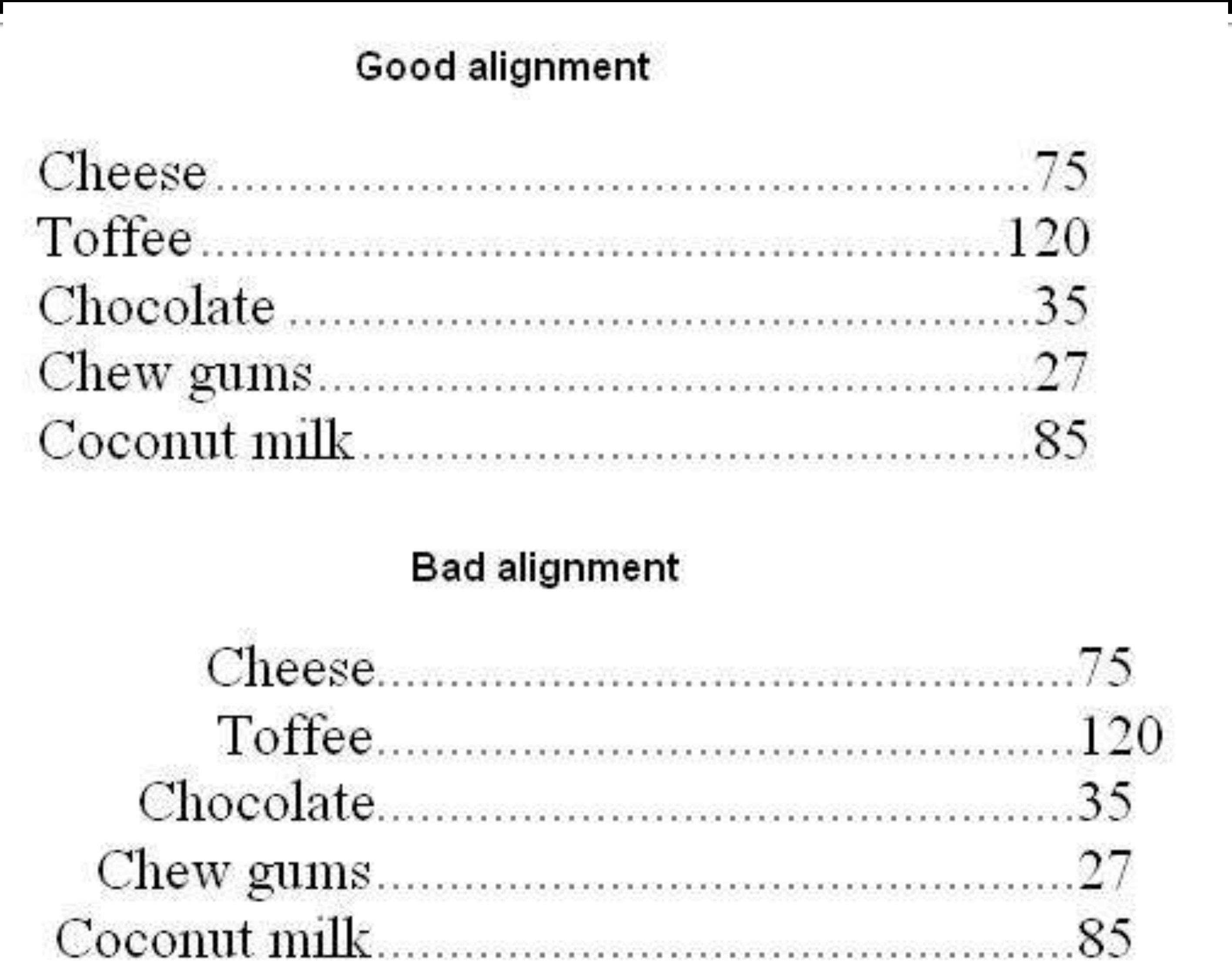


Horizontal Bar

Analogue Representation of Relative Values



Representing Numerical Values



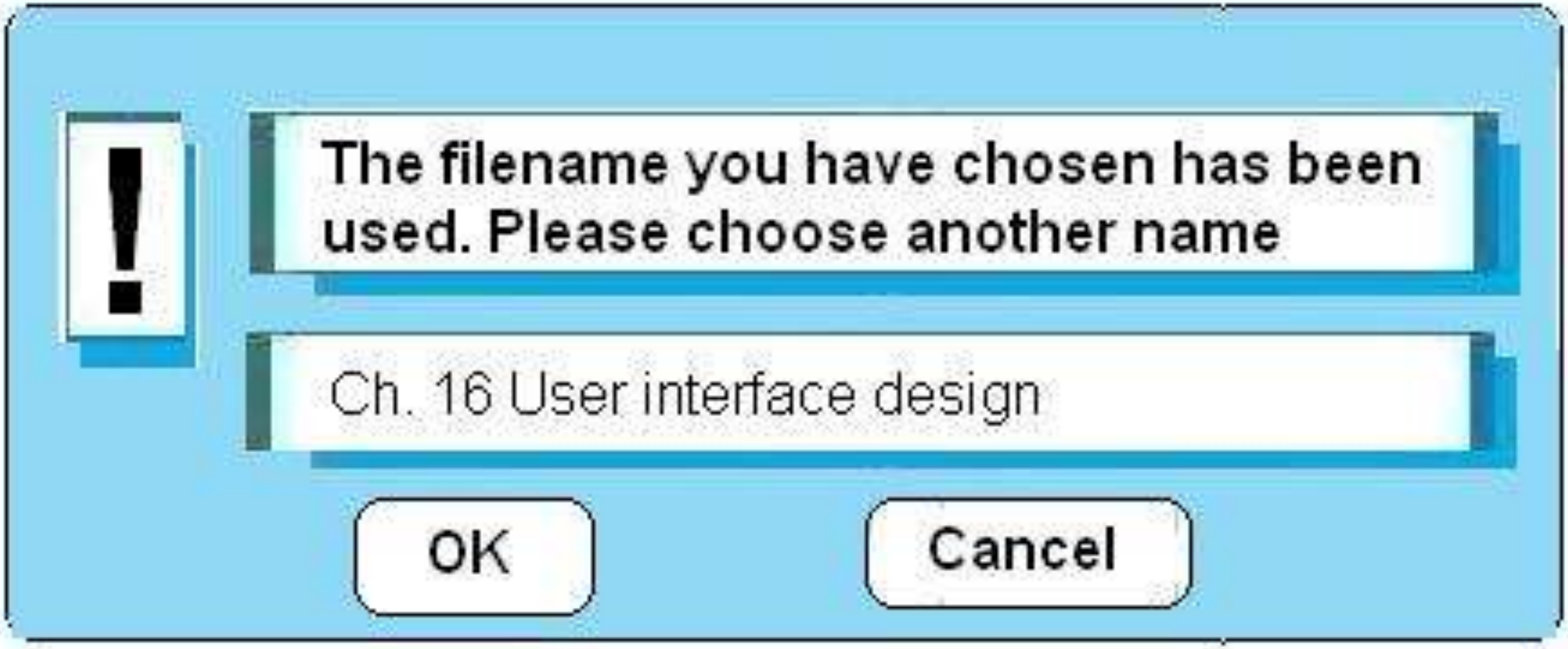
Good alignment

Cheese	75
Toffee	120
Chocolate	35
Chew gums	27
Coconut milk	85

Bad alignment

Cheese	75
Toffee	120
Chocolate	35
Chew gums	27
Coconut milk	85

Representing Alphanumerical Values on Graphical Backgrounds



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

Ch. 16 User interface design

Color

Used for:

- Highlighting
- Showing different layers

Guidelines:

- Limit colors
- Show state changes
- Color coding – For anomalies & similarities
- Consistency
- Avoid poor color pairing

Do not use colors to represent meanings
10% of men & 2% of women are color blind!

Fonts

- Pitch
  - Fixed pitch (same width)
    - E.g.: `Courier`
  - Variable pitch
    - E.g.: Times New Roman
- Serif or Sans Serif
  - Serif (splayed ends) → good for reading
    - E.g.: Times New Roman
  - Sans Serif (squared ends) → good for presenting
    - E.g.: Arial

User Support

Help!

→ I need help, I'm stuck!

Help?

→ I need more information.

User Help & Guidance:

- System's messages responding to user actions
- Online help
- System documentation

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.

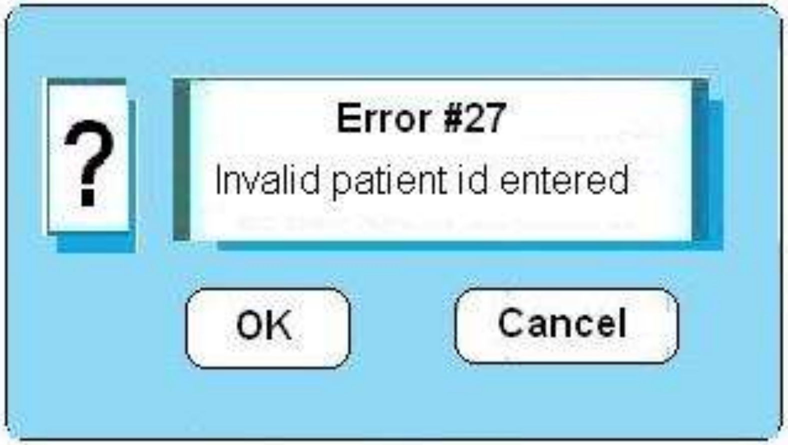
Error messages(contd…)

Consider:

- Context – Use user-oriented language, not system-specific terms
- Experience – User controls message conciseness
- Skill level
- Style – Positive, not negative
  Active, not passive
  Not insulting or funny
  - Be polite, concise, consistent & constructive
  - Never be abusive – Avoid beeps & noises
  - Suggest how to correct error

User-Oriented vs. System-Specific Error Messages

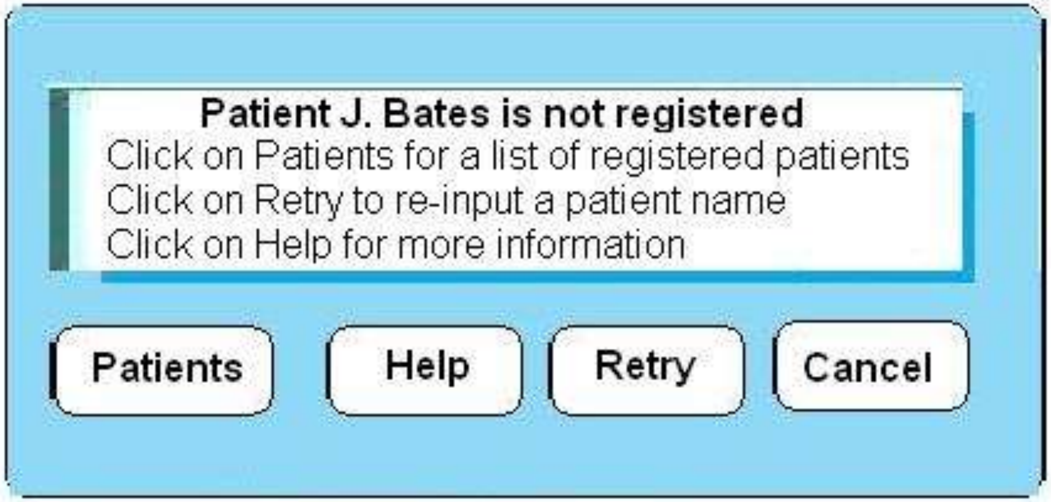
System-oriented error message



Error #27

Invalid patient id entered

User-oriented error message



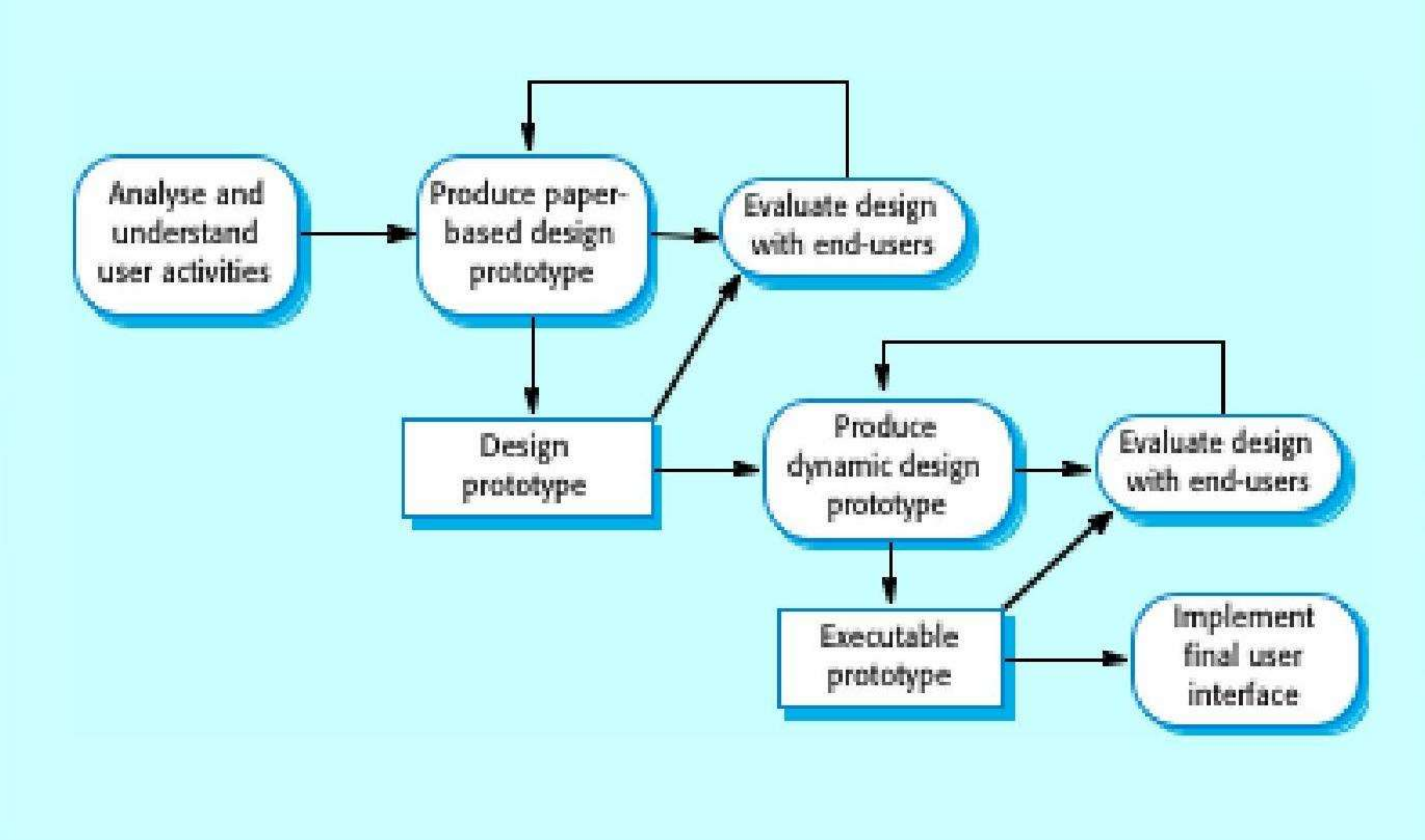
Patient J. Bates is not registered

Click on Patients for a list of registered patients
Click on Retry to re-input a patient name
Click on Help for more information

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 UI design process


Analyse and understand user activities
Produce paper-based design prototype
Evaluate design with end-users
Design prototype
Produce dynamic design prototype
Evaluate design with end-users
Executable prototype
Implement final user interface

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.

User interaction scenario

Jane is a student of Religious Studies and is working on an essay on Indian architecture and how it has been influenced by religious practices. To help her understand this, she would like to access some pictures of details on notable buildings but can’t find anything in her local library.

She approaches the subject librarian to discuss her needs and he suggests some search terms that might be used. He also suggests some libraries in New Delhi and London that might have this material so they log on to the library catalogues and do some searching using these terms. They find some source material and place a request for photocopies of the pictures with architectural detail to be posted directly to Jane.

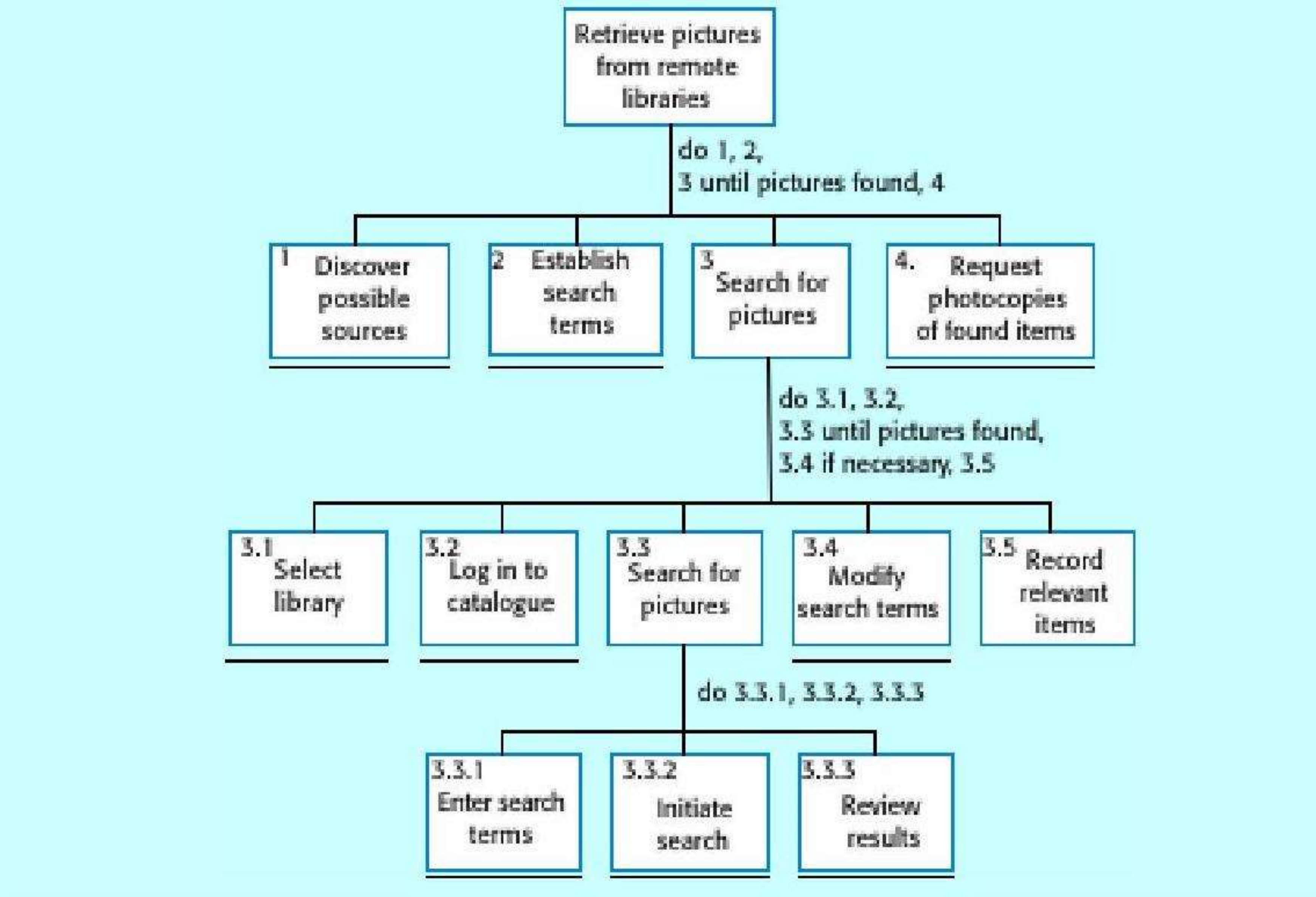
Requirements from the scenario

- Users may not be aware of appropriate search terms so need a way of helping them choose terms.
- Users have to be able to select collections to search.
- Users need to be able to carry out searches and request copies of relevant material.

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


Retrieve pictures from remote libraries
do 1, 2, 3 until pictures found, 4
1 Discover possible sources
2 Establish search terms
3 Search for pictures
4. Request photocopies of found items
do 3.1, 3.2, 3.3 until pictures found, 3.4 if necessary, 3.5
3.1 Select library
3.2 Log in to catalogue
3.3 Search for pictures
3.4 Modify search terms
3.5 Record relevant items
do 3.3.1, 3.3.2, 3.3.3
3.3.1 Enter search terms
3.3.2 Initiate search
3.3.3 Review results

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 organizational 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;
  - The design is then refined and increasingly sophisticated automated prototypes are then developed.

Paper prototyping

- Work through scenarios using sketches of the interface.
- Use a storyboard to present a series of interactions with the system.
- Paper prototyping is an effective way of getting user reactions to a design proposal.

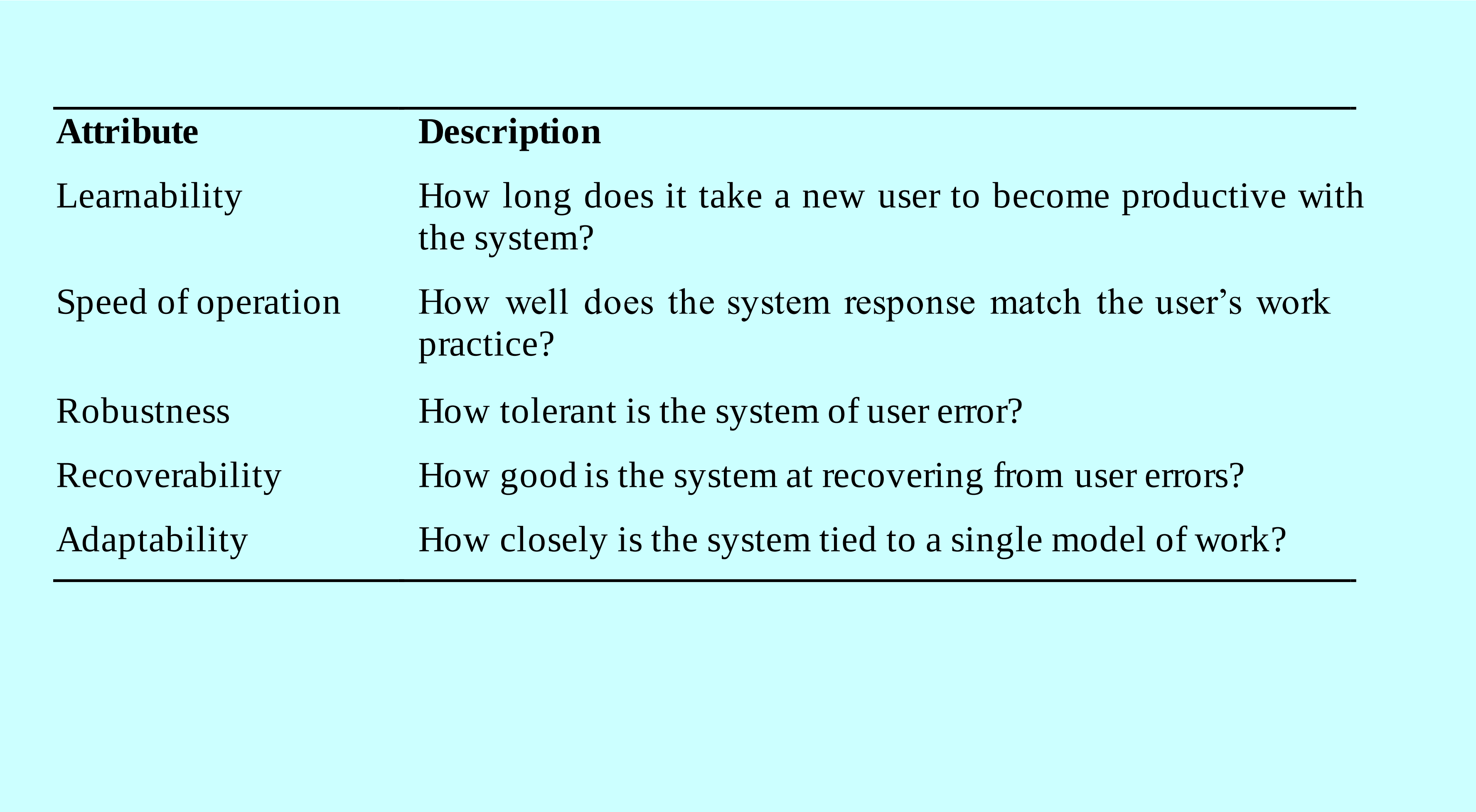
Prototyping techniques

- 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.
- Internet-based prototyping
  - Use a web browser and associated scripts.

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



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 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.
- Observation of users work with the system
- The provision of code in the software to collect on-line user feedback.

Questions?