- User Interface Design
  - Chapter 16 +.

Importance of UI Design

- System users often judge system quality by its interface rather than its functionality
  - Computers are now being used by many non-technical people
  - Implementers are often poor judges of interface acceptability and effectiveness
- Poor user interface design can decrease the effectiveness, productivity and safety of software intensive systems
  - The interface to one tool may psychologically "interfere" with the interface to another. Hence tool integration is important
- A significant percentage of application code is devoted to the user interface
  - As much as 70% for some classes of applications

Human Factors

- Limited capacity of human short-term memory; need to "chunk"
- 111100001111111110
- 111 100 001 111 111 110
- 7 4 1 7 7 6
- 7/4/1776

Human Factors

- Limited capacity of human short-term memory; need to "chunk"
- People make mistakes
- People have different interaction preferences
- People perform differently
  - novice, knowledgeable intermittent, frequent expert
- Performance improves as people learn to use a system
- Many human factor aspects of UI are measurable
- There exist some experimental results, but technology is being driven by the market rather than by research

Measurable Human Factor Goals

- Time to learn
- Speed of performance
- Rate of errors
- Subjective satisfaction
- Retention over time

Aspects of UI Design

- Menu selection formats
  - Click, letters, numbers; pulldown, popup, cascading
- Message wording, terminology, abbreviations, syntax
- Character set, fonts, highlighting, color
- Devices
  - Keyboards, displays, cursor control, virtual reality
- Modalities
  - Sound, touch, position sensor
- Response times, display rates
- On-line help, tutorials, training and documentation
- Error interaction and recovery

Structure of UI

- **Presentation:** display of computer generated or processed information
- **Dialog:** controlled solicitation of user input
- **Application interface:** the two way translation of processing requests between the user interface software and application software

Presentation Issues

- Information visualization
  - digital vs. analog, text vs. graphs…
- Animation
- Use of colors, fonts, highlights
- Text vs. graphics
- Static vs. dynamic
- Ambient vs. requested

Dialog Modeling

- Some interfaces are *modal*.
  - That is, their state affects allowable user interactions
- Modality can be dealt with differing amounts of emphasis
  - Vi vs. emacs on command entry
  - Disabled options in dialogs
  - Separate windows, colors
  - Wizards
- State transition machines/diagrams can often be used to model UI dialog

Elements of Dialog Models

- For computer application software, the user typically interacts by entering text, pressing mouse buttons, or pressing a function key
- The user may be selecting from a menu, filling in a form, or answering a question
- Hence, a model of a user interface needs to indicate not only the states and their transitions, but also the contents of the menus, the layout of the forms, and the details of the questions
- A user action often translates into a request for computation. So a dialog model must specify the requested computations

Application Interface

- User requests are initiated by UI *events*, such as keystrokes or mouse clicks
- GUI software normally features an event loop that interprets events and invokes corresponding functionality
- Parameters may be passed to the program either by push or pull
- Resultant values must be displayed by the GUI

Interaction Style

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

Other interaction styles

- Programming language
- Handwriting
- Voice / speech
- Drawing

UI services

- Macro commands; hot keys; accellerators
- Undo; redo
- Cut; copy; paste
- On-line help; tutorials
- Default values
- User confirmation; customization files
- Command files

UI integration

- There is now an abundance of applications available to typical computer users
- Similar sounding operations in separate applications may cause very different actions to be taken
- Hence, integration speeds learning and reduces errors
- It can also increase productivity by allowing separate applications to interact effectively

Aspects of integration

- **Presentation integration:** same message format, highlighting, and WIMP styles across applications
- **Dialog integration:** message and error interactions, data entry, and menu control behaving similarly
- **Semantic intergration:** identical actions in separate applications behaving identically

Integration metaphors

- One way that integration has been facilitated in the past is through the use of a consistent metaphor for all applications.
- This includes applications written by the user
- Examples of metaphors that have been used are the following:
  - Desktop
  - Window manager
  - Screen editor
  - Spreadsheet
  - Data base

UI Principles

- Know the user; support transition
- Organize application interface around tasks
- Limit reliance on human memory
- Provide easy ways to do common things
- Common look and feel; integration; use of metaphor
- Engineer for error
- Measure; evaluate; refine

More UI 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.

Error message guidelines

- Product
  - Be as specific and precise as possible
  - Be constructive: indicate what needs to be done
  - Use a positive tone: avoid condemnation
  - Choose user-centered phrasing
  - Consider multiple levels of messages
  - Keep consistent grammatical form, terminology, and abbreviations
- Process
  - Establish a message quality control group
  - Include messages in the design phase
  - Place all messages in a file
  - Review messages during development
  - Attempt to eliminate the need for messages
  - Carry out acceptance tests
  - Collect frequency data for each message
  - Review and revise messages over time

Menu selection guidelines

- Use task semantics to organize menu structure (single, linear sequence, tree structure, acyclic networks, and cyclic networks)
- Try to give position in organization by graphic design, numbering, and titles
- Items become titles in walking down a tree
- Make meaningful groupings of items in a menu
- Make meaningful sequences of items in a menu
- Items should be brief and consistent in grammatical style
- Permit type-ahead, jump-ahead, or other short-cuts
- Permit jumps to previous and main menu
- Use consistent layout and terminology
- Consider novel selection mechanisms and devices
- Consider response time and display rate impact
- Consider screen size
- Offer help facilities

Dialog design guidelines

- Consistency
- Short cuts for experts
- Feedback
- Closure
- Be aware of common errors
- Simple error handling
- Undo
- User-driven
- Reduced short-term memory load
- Log user errors
- Feedback on selection

Command language guidelines

- Create explicit model of objects and actions
- Choose meaningful, specific, distinctive names
- Try for hierarchical structure
- Provide consistent structure (hierarchy, argument order, action-object)
- Support consistent abbreviation rules (prefer truncation to one letter)
- Offer frequent users the capability to create macros
- Consider command menus on high-speed displays
- Limit number of commands and ways of accomplishing a task

UI Design Process

- UI design is an iterative process involving close liaisons between users and designers
- Core activities
  - **User analysis:** Understand who the user is and what they will do with the system
  - **System prototyping:** Develop a series of prototypes for experiment
  - **Interface evaluation:** Experiment with these prototypes with users

Analysis

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

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 userÕswork 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?

Sample 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

UI Design Technology

- WYSIWYG UI-building tools
- Widget libraries;
- UI toolkits
- Special purpose graphics hardware
- Animation scripting
- Virtual reality
- Hypertext
- Groupware Libraries for 3D presentations
- Model-based user interface generation
  - User, display, presentation, dialog, application models
- Scripted GUI testers

Direct Manipulation Interface

- Continuous representation of objects; visible feedback
- Physical actions instead of complex syntax
- Rapid, incremental, reversible operations
- Shallow learning curve