The Design of Interactive Computational Media

Class 5: 4 Jan. 2003

Activity Design

Hour 1:
Interactive Computational Media
Design Methodologies and Principles

Hour 2:
Activity Design; Metaphors and Mental Models

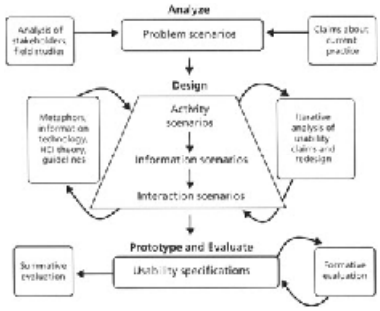
Interactive Computational Media
Design Methodologies and Principles

- Design methodologies
- Examples
  - Rosson and Carroll book
  - Baecker Grudin Buxton Greenberg (BGBG) book
- Design principles and guidelines
- Examples
- One set of design principles with illustrations

Design Methodologies

- Systematic procedures for organizing design processes for interactive computational media

Rosson and Carroll Design Methodology



BGBG Design Methodology

- A user-centred, iterative, design philosophy
  - Not intended as a rigid formula
  - Illustration of a philosophy
  - Examples of how to proceed
- Design —>Prototype- —>Evaluate —>Redesign —>Implement —>Evaluate —>Redesign —>Revise implementation —>Evaluate —>etc..

BGBG design process example (MAD)

	DESIGN	IMPLEMENT	ANALYZE AND EVALUATE
Information collection and requirements analysis	Reflections, studies, classes on filmmaking	No "Problem Scenario" developed (weakness)	Some contact with real filmmakers (should have had more contact)
Activity, information & interaction design	Initial design concepts	Design sketches, Director prototypes, small C programs	Feedback only from research group (weakness)
Prototyping and prototype system	System functionality and look-&-feel	Critical mass C prototype	Demos, first real projects, observations, filmmaker interviews
Production prototype and its evolution	Complete system, incorporating evaluation insights	Implementation of significantly useable C++ system	More demos, real projects, observations, interviews, multimedia summer camps
Production system and its evolution	Deliverable system, incorporating evaluation insights	Java implementation	Intensive internal use, beta testing, client use

BGBG design process (general case)

	DESIGN	IMPLEMENT	ANALYZE AND EVALUATE
Information collection and requirements analysis	Questionnaires, interviews, observation of potential users	Task analyses, artifact analyses, "day in the life" "problem" scenarios	e.g., interviews with users to get reactions to scenarios
Activity, information & interaction design	Initial design concepts	Design mockups, prototypes, activity scenarios	e.g., interviews with users to get reactions to prototypes, heuristic evaluations of prototypes
Prototyping and prototype system	System functionality and look-&-feel	"Smoke and mirrors" prototype, partially working system	e.g., usability tests
Production prototype and its evolution	Complete system, incorporating evaluation insights	Real working system, implemented and installed	e.g., heuristic evaulation, usability tests, reports from beta sites
Production system and its evolution	Deliverable system, monitoring and feedback system	Production system, including monitoring and feedback system	e.g., interviews, surveys of real users

Design principles (guidelines)

- Even if you follow a methodology, how do you know that you are proceeding uphill rather than downhill?
- Design principles or guidelines: statements which advise a designer on how to proceed
- Example (Hansen, 1971)
  - Know thy user
  - Minimize memorization
  - Optimize operations
  - Engineer for errors

Macintosh Human Interface Guidelines

- "...describes the way to create products that optimize the interaction between people and Macintosh computers" (Apple Computer, Addision-Wesley, 1992)
  - Ch. 1: Human Interface Principles
  - Ch. 2: General Design Considerations
  - Ch. 3: Human Interface Design and the Development Process
  - Ch. 4: Menus
  - Ch. 5: Windows
  - Ch. 6: Dialog Boxes
  - Ch. 7: Controls
  - Ch. 8: Icons
  - Ch. 9: Colour
  - Ch. 10: Behaviours
  - Ch. 11: Language

Tog On Interface

- Bruce "Tog" Tognazzini (Addison-Wesley, 1992) answers specific questions about user interface design for the Macintosh, and presents 200 guidelines dealing with, e.g.:
  - The Design Process
  - Positively Determining System Behaviour
  - Positively Influencing User Perceptions and User Behaviour
  - Promoting Consistency
  - Making the Interface "Visible"
  - Reducing or Eliminating Navigation
  - Conceptual Models and the System Image
  - Human-Computer Conversation, Vocabulary
  - Screen Objects, Menus, Icons, Fonts, Error Messages
  - User Testing
  - Minimizing Impact of New Releases on Old Users

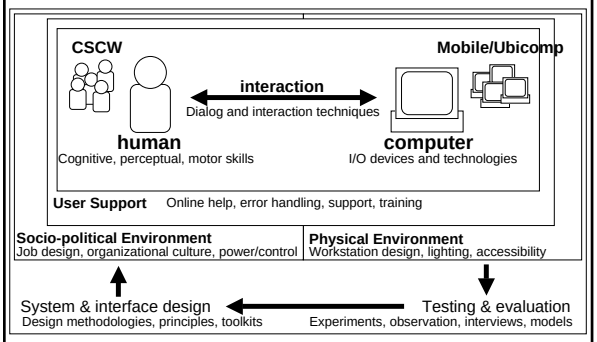
Guidelines

- Pros of guidelines
  - Stimulate ideas and insights
  - Good checklists giving helpful advice
  - Use in *heuristic evaluation*
- Cons of guidelines
  - Occasionally incorrect
  - Usually vague
  - Sometimes contradictory (need for tradeoffs)
  - Very often not at the appropriate level of specificity
  - Often difficult to apply to real design problems
  - Can be too numerous
    - Example: Smith and Mosier: 679 (!!) guidelines (1984)

My Design Guidelines (Principles)

- Heavily influenced by 30 "design elements" in P. Heckel, *The Elements of Friendly Software Design*, The New Edition, Sybex, 1991
- Twenty principles organized as follows:
  - The design and the design process
  - The user
  - The technology and the interaction
  - User support
  - The computational medium
- Illustrations
  - MAD (Expresto Creator) system design
  - Interface examples courtesy of Aaron Marcus and Associates, www.amanda.com

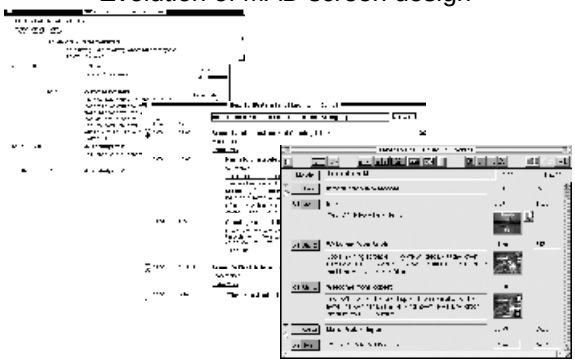
HCI and Computational Media Design Process



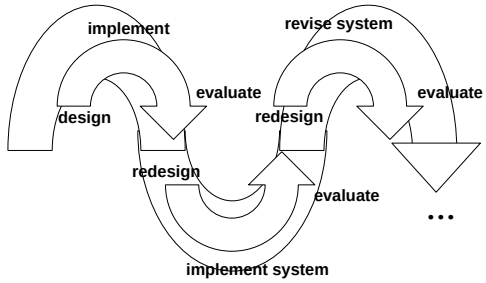
Design principles: The designer and the design process

- 1. Be humble, and iterate often
  - You won't get it right the first time …
  - Or the second time either :-)
  - MAD [C] —> Cinekit [C++] —> Expresto Creator [Java]
- 2. Follow a user-centred design process
  - Study work practice
  - Observe, "test" users as they use a system
  - Ask users with surveys, questionnaires, interviews
  - Try to put user(s) on the design team
  - *Also ask external experts in a systematic way*
  - MAD: Lots of internal and external use, multimedia summer camps with kids, interviews with filmmakers

Evolution of MAD screen design



The iterative design process



Design principles: The designer and the design process

- 3. Use multidisciplinary design teams
  - Software
  - User interface design
  - Social/behavioural science
  - Visual/graphic design
  - Domain expertise
  - MAD: Backgrounds in computer science, linguistics, HCI, architecture and design, music, psychology, …
- 4. Really know the subject matter
  - Deep domain expertise required
  - MAD: Good expertise, but could have used more

Design principles: The designer and the design process

- 5. Consider the physical environment
  - Example: Workstation ergonomics
  - Example: Mobile use
  - MAD: The Internet for distributing digital movies suggests web publishing (implemented in Expresto Creator)
- 6. Consider the work (or school or play) environment
  - Example: IBM Speech Filing System
  - MAD: The existing filmmaking production process

Design principles: The user

- 7. Really know “the user”
  - Who is a typical user? Who are all the users?
  - Observe, ask, have users participate on design team
  - Creator: Needed a filmmaker on the team
- 8. Employ the user’s knowledge
  - Communicate with appropriate metaphors
    - Example: Virtual museum
    - Example: Electronic book
    - MAD: Script, storyboard, timeline
  - Speak the user’s language
    - The user’s jargon, not computer jargon
    - Example: Points, picas, em dashes for typographers
    - MAD: NTSC Timecode — HH:MM:SS:FF

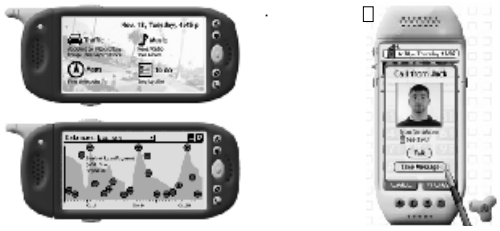
Design principles: The user

- 9. Build a mental model in the user’s mind
  - Begin with the metaphor
  - Example: Images made of pixels (painting program) or lines (drawing program)
  - Example: Financial data on a gridded worksheet with rows and columns (spreadsheet)
  - MAD: Hierarchically organized scripts —> acts, scenes, shots
- 10. Design for varieties of user expertise
  - Example: novice and experienced users
  - Example: the role of user tailorability (McGrenere)
  - MAD: Primarily for novices, needed more for experts, e.g., timeline editor

Design principles: The technology and the interaction

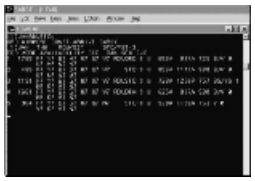
- 11. Exploit new hardware paradigms
  - Example: mobile devices linked at high-bandwidth
  - Example: speech I/O, non-speech audio
  - MAD: Portable filmmaker’s unit
- 12. Communicate visually and articulately
  - Focus the user’s attention
    - Key information at the tracking symbol
    - Graceful methods to grab the user’s attention
  - Structure the user’s interface
    - A frame of reference, a mental map
    - The role of design grids
  - MAD: Various visual representations and interfaces

Prototypes: Samsung Advanced Mobile Device Concepts

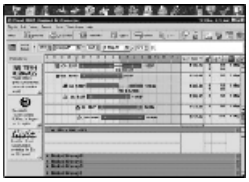


Prototypes: Message Manager For a Wrist-top Device

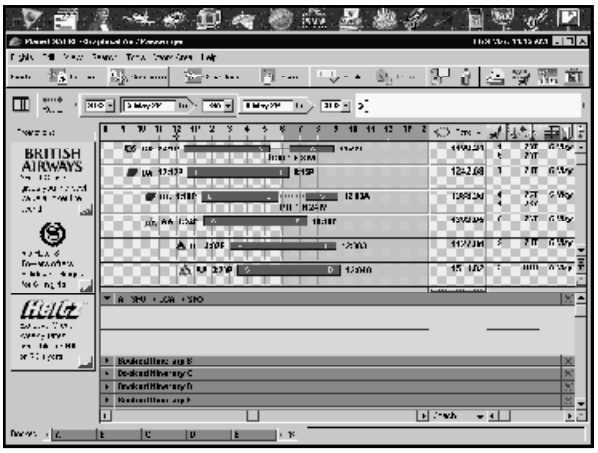
Application Example: Sabre Travel Booking Development



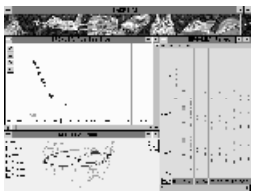
Before



After



Sabre: Information-Visualization and User-Interface Design



Information Visualization

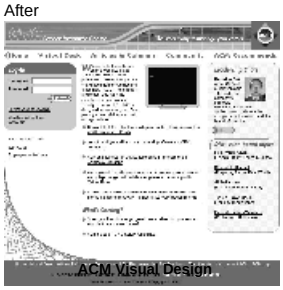
Interactive UI Guidelines

Website: ACM.org Portal Career Resource Centre Development

Before

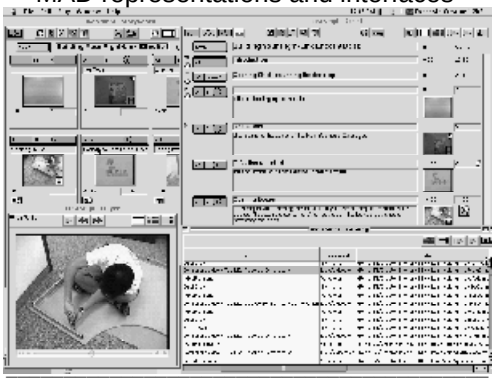
ACM Information Design

After



ACM Visual Design

MAD representations and interfaces



Design principles: Technology & interaction

- 13. Respond articulately to the user's actions
  - Speed and predictability of response
  - Complete, terse, comprehensible feedback
  - Hidden system state (modes) kept to a minimum
    - Example: My TV Zapper
  - MAD: Immediate real-time playback at any time
- 14. Orient the user in the world
  - Where am I? Where have I been? Where can I go?
  - Example: Web sites with and without site maps
- 15. Enable articulate expression by users
  - Example: Widgets for controlling rectangular areas
  - MAD: Consistent family of editors for still images and titles, sound tracks and video sequences

Widget for controlling rectangular areas

Design principles: User support

- 16. Anticipate that users will have "problems"
  - Huge varieties of users, tasks, contexts
  - Need to anticipate and if possible prevent "errors"
  - Need for online help, error handling, training, support
  - MAD: "World's first video manual for software"
- 17. Minimize user frustration
  - Consistency whenever possible
  - Error message language to reduce defensiveness
  - MAD: Does the right things, except where Java, Quicktime, or Quicktime for Java do the wrong things

Design principles: User support

- 18. Support collaborative and individual use
  - Example: Technical support via a knowledge base *and* via links to experts
- 19. Make your product reliable
  - If it doesn't work well, the interface won't save it!
  - MAD: Deeply flawed due to Java bugs
- 20. Make your design simple
  - Elegance and simplicity aid everyone involved — the designer, the implementer, *and* the user
  - MAD: Nothing added without a good reason!

Summary

- Methodologies: Systematic design processes
- Principles or guidelines: "Rules" of design advice
  - Issues of tradeoffs between conflicting guidelines

Questions and Discussion

Break

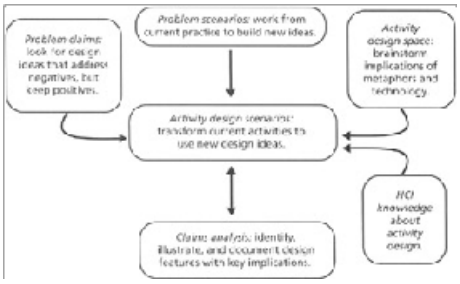
Activity Design; Metaphors and Mental Models

- Activity design
- Design process from Rosson and Carroll
- Metaphors
- Examples
- Mental models

Activity Design

- "Problem scenario" illustrates target users doing tasks using artifacts in their natural environment
  - Encapsulates current practice
  - Highlights issues and problems
- Designer has a concept for a solution to the problems
- *Activity design* articulates functionality for the solution
- Goal is to make activities
  - Effective
  - Satisfying
  - Comprehensible
- *Activity scenarios* illustrate use of the solution
- We can make claims about features of these scenarios

Activity Design (Rosson & Carroll)



Activity Design: Effectiveness

- How to we know we are solving the right problem?
- Careful analysis of results of requirements analysis
- Collaborative (participatory) design
  - *User-centred —> user-involved —> user-directed*
  - Origins in Scandinavia
  - Developers and users: equal partners on design team
  - Mutual knowledge
  - See Reading #8, *Situated Design*, Greenbaum & Kyng

Activity Design: Satisfaction

- We want computer use to be
  - Productive
  - Comfortable and satisfying
- It should not be
  - Counter-intuitive
  - Frustrating
  - Stressful
- Role of task analysis
  - Current practice (useful for "problem scenarios")
  - Desired practice (useful for "activity scenarios")

Class Exercise: Task Analysis for Electronic Classroom

- *Setting up*
  - *Lower screen*
  - *Plug in laptop*
  - *Turn laptop on*
  - *…*
- *Teaching*
  - *Advance to next slide*
  - *Present the slide*
  - *Annotate and discuss the slide*
  - *...*

Activity Design: Comprehensibility

- User needs, task analysis, and user work practices — How users think about their work — their "conceptual models"
- Example of a familiar users' conceptual model
  - The Xerox Star — The origins of the electronic desktop and the GUI — Predecessor to Lisa, Macintosh, and Windows
  - A "simulated desktop" with electronic equivalents of *paper, filefolder, file cabinets, mailboxes*
  - Key cognitive issues
    - Users' conceptual (mental) model
    - Building parallels to office concepts, objects, and operations
    - A computer system **is** an electronic office (a metaphor)

Metaphors

- What are these mental models? What are they like? How can they be conceptualized?
- Very often they relate presumed structure and function of a system to that of another, "simpler", familiar system
- The goal is to exploit the specific prior knowledge that users have of this other domain
- We use *metaphor* (an X is a Y) meaning X is like Y in certain (many) respects
- See Erickson, *Working with Interface Metaphors*, Reading #9

Examples of Metaphors

- Football is (like) war
- War is (like) football
- Text editor is (like) a typewriter
- Memory is (like) a set of pigeonholes
- Screen is (like) a television
- Screen is (like) a desktop in an office
- LOGO procedures are (like) cooperating “little people”

Examples of Metaphors

Application area	Metaphor	Exploits knowledge of
Word processing (e.g., Word)	Typewriting	Typewriting, typing paper, keyboard
Spreadsheets (e.g., Excel)	Ledger sheet	Numerical data and calculations in rows and columns
Personal financial management software (e.g., Quicken)	Checkbook, financial register	Working with a checkbook
Shared electronic workspaces (e.g., Smart Technologies)	Chalkboard, whiteboard	Writing, sketching, and collaborating on a chalkboard or a whiteboard
Idea processors, outline processors (e.g., More)	Outline	Organizing, decomposing, combining, and rearranging ideas and concepts
Virtual science fair exhibit	Lab notebook	Taking notes, recording “work in progress”
Virtual science fair exhibit	Documentary	Telling a story (in text, voice, film)
Virtual science fair exhibit	Web site	Constructing a story out of separate pages, linking them together

Metaphors are not Identities

- An X is a Y
- But X ? Y, else we would have identity, not metaphor
- Interesting aspects are areas of mismatch, breakdown
- Divide into metaphor {works, doesn't work, doesn’t apply}
- Example: Text editor is a typewriter
  - Works: Input of text, form of text, appending text
  - Does not work: Rather than *type over,* we have *insert* or *change*
  - Doen’t apply: *Block move* (e.g., cutting/pasting pieces of paper)

Recommendations regarding metaphors

- Find appropriate metaphors for teaching system to novice user
- Given choice between two metaphors, favour one based on:
  - Congruence to system (Isomorphism between entities and relationships in system and in metaphor)
  - Coverage of system's objects, features, operations
- Use related metaphors where appropriate, ideally from similar real-world domains (e.g., filing cabinet, storage boxes)
- Choose the emotional tone of the metaphor appropriately (e.g., war vs. peace, work vs. play, science vs., art, writing vs. drawing)

Recommendations regarding metaphors

- Choose metaphors that have distinctive visual and auditory representations (icons, auditory icons as a goal)
- Think through likely consequences of metaphor to users
  - Worry about apparently small details, e.g., objects using book metaphor should have page numbers, tables of contents, indices
- Point out limitations of metaphors
- Look for sequences of metaphors or models – replace one by the next when the first begins to break down, e.g., IBM Speech Filing System – Audio Distribution System
  - Telephone Answering Machine
  - Telephone Answering Machine w. Remote Control Playback

Mental models

- Metaphors and mental models
  - "Metaphors function as natural models, allowing us to take our knowledge of familiar, concrete objects and experiences and use it **to give structure to more abstract concepts**." (Erickson, in Laurel book, p. 66)
- Definition of mental models (Carroll, 1984):
  - "...structures and processes imputed to a person's mind in order to account for that person's behaviour and experience."
- More generally (Carroll & Olson, 1988):
  - "...all of what a user knows about using a particular piece of software, including *how to use it*, and how it works."

Role of mental model

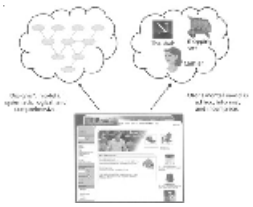
- To answer questions like:
  - What is X?
  - What happens when you do Y?
  - Why do Z?
- Example: Mental model of simple line drawing system
  - *Objects*: Page, line, point
  - *Relations*
    - Page contains 0 or more lines
    - Line connects 2 points
  - *Actions on objects*
    - Clear a Page
    - Create, delete, move points and lines
  - *Attributes of objects*
    - Color, style, weight of lines
    - Type of point
  - *Actions on attributes*: Change these attributes

Examples: Prototyping tools (later in term)

- HyperCard
  - Card, stack of cards
- Director, Flash
  - Animation, sequencing images through time
- Visual Basic
  - Set of active elements on a page with associated code
- Dreamweaver
  - Web site, collection of web pages

Kinds of models

- Designer=====> System <======> User
- Need to distinguish among the system and
  - *Designers' Conceptual Model* of the System
  - Users' Image of the System – *System Image*
  - *Users' Mental Model* of the System
  - Scientist's *Conceptualization* of that Mental Model



Relationships of models

- System built by designer
- Designers' conceptual model
  - Coherent structure behind the design
  - Goal is logic, unity, consistency
- System Image –view of system seen by user
  - Objects, commands, options, states, etc.
  - Not necessarily coherent, logic may not be apparent
  - For learners, a view through a peephole, system emerges little by little through training, use, exploration
- Users' mental model
  - Eventually, if structure is there, user may discover it, induce a coherent model of the system
  - If design is appropriate, if learning environment works, users' mental model will reflect designers' conceptual model
  - If not, mental model may be incomplete, have errors, be based on accidents enshrined as superstition

Scientific study of models

- Researchers attempt to build more and more complete, formal, and precise models of:
  - Cognitive processes of user
  - Their mental models
  - Methods such as metaphor that assist in the development of mental models
- More on this in advanced HCI courses, e.g., CSC428

Metaphors, models, and learning

- Carroll and Mack (Reading #19) description of how users learn a computer system
- Learning by *doing*
  - Desire to try things out
  - Tendency to jump the gun
  - Difficult in following written sequences of instructions
- Learning by *thinking*
  - Attempting to construct reasonable interpretations, proper *mental models* (sense-making)
  - Purposeful problem solving activity
- Learning by *knowing*
  - Making use of prior knowledge, from *metaphors* and work experience

Summary

- Designing activities so that they are
  - Effective
  - Satisfying
  - Comprehensible
- Metaphors
- Mental models

Questions and Discussion