CISC 839
Software Engineering of Usable Computing Systems

T.C. Nicholas Graham
http://stl.cs.queensu.ca/~graham/cisc836

Techniques for Stimulating Creative Design

- Metaphor
- Brainstorming

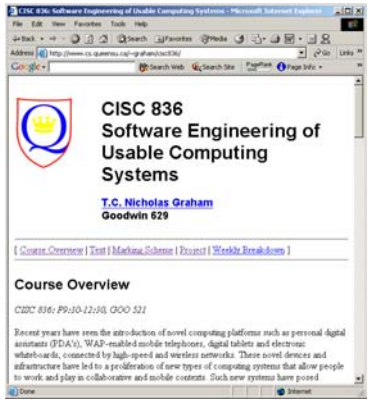
Metaphors

- Something familiar to user from real world
- Helps to structure user interface
- Helps user understand structure, predict behaviour of user interface
  - ∴ Helps in developing mental model

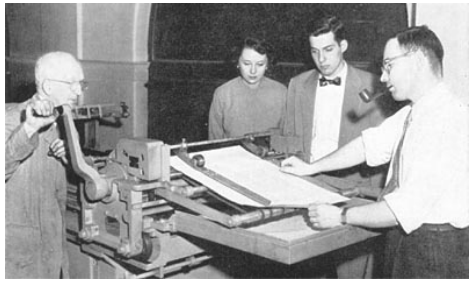
Example Metaphors

- Window
- Copy/paste
- Desktop

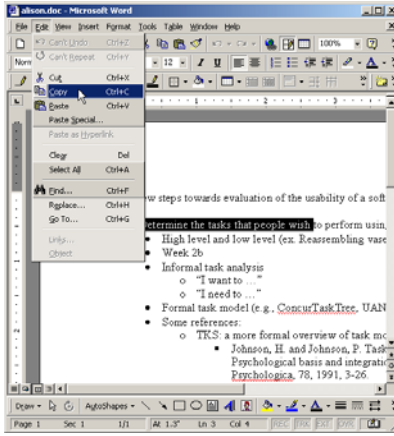
Window



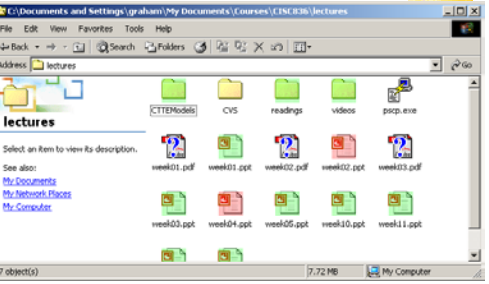
Cut/Paste



Source:
http://www.jrn.columbia.edu/alumni/history/galleries/1950-1959_gallry.asp?photo=9



Desktop



Source:
http://www.staples.ca

Metaphors

- Should help users to predict results of actions
  - Therefore important for metaphor to follow real life correctly
- Alternative approach
  - Metaphor can be used in design, then stripped out of final product

Brainstorming

- Method for group of people to come up with design ideas
- Use large blackboard, large sheets of paper, etc.

Brainstorming

- Allow people to throw out ideas
  - Record ideas
  - Most ideas will get thrown out, so don't worry about evaluating them
  - Sometimes a series of "dumb" ideas may lead to a good one
  - …so don't be afraid to have "dumb" ideas!
  - Never criticize/put down ideas during brainstorming

Brainstorming

- Can lead to quick, fluid development of ideas, creative thought

In-Class Exercise

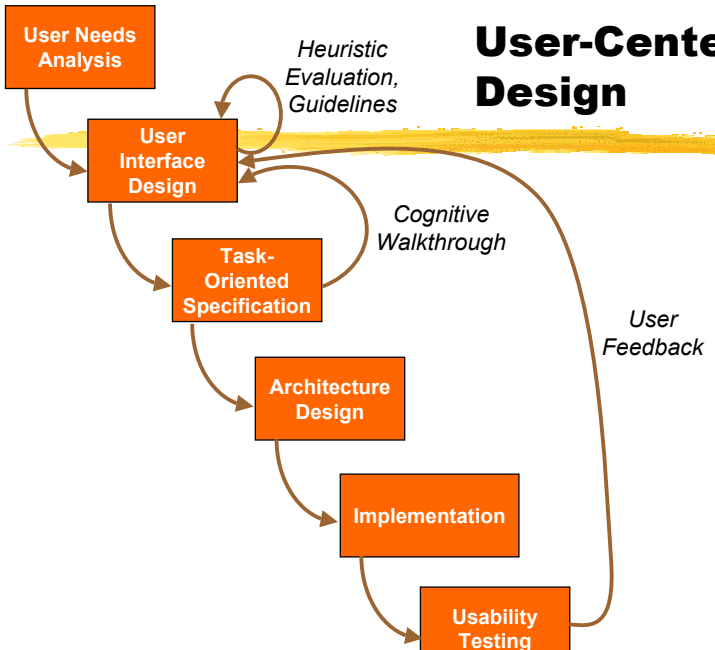
- Design a system for ordering pizza from your PC
- Metaphor is:
  - Going for a trip to the zoo

User Interface Evaluation

Evaluating User Interfaces

- Important to make evaluation early, continuous
- Techniques:
  - Heuristic evaluation
  - Guidelines
  - Cognitive walkthrough
  - Usability testing (user observation)

User-Centered Design



Heuristic Evaluation

- User interface specialists study user interface design in depth
- Looking for known problems from their experience

Guidelines

- Guidelines encode experience
- In effect, allow non-specialists to perform their own heuristic evaluation
  - Guidelines help structure system evaluation by telling designers what problems to look for

Cognitive Walkthrough

- Walk through interface in context of user trying to perform a task
  - Identify core tasks
  - Show how they would be accomplished by walking through interaction steps with user interface
- Shows difficulties with task-action mapping
  - E.g. task hard to perform, task not supported at all
- Helps identify areas where operations should be optimized

Approaches to Cognitive Walkthrough

- Informal
  - Narrative scenario
    - write out steps followed by user in narrative prose
  - Low-fidelity prototype
    - cutouts of interface elements that can be manipulated by hand
  - Mockups
    - Simulations in Macromedia Director, Shockwave etc
- Formal
  - Precise scenario
    - UML sequence diagram
  - Task-oriented specification
    - Refine ConcurTaskTree to action level
- Formal approaches are great, but significant information can be gained from cheap, informal approaches

Usability Testing

- Test usability of system by observing users
- Users given specific task to accomplish
- Observe to see what problems arise
- Can be performed with low fidelity prototype or with real system
  - Trade off quality of feedback vs how early it can be obtained

Comparison of Four Approaches

- UWE Project, HP [Jeffries et al., 1991]
- Visual interface to Unix under X/Motif
- Applied four methods separately to compare results
- Error types
  - Consistency (25% of problems)
  - Recurring -- one time or recurring?
  - General -- affecting several parts of UI or specific to single part?

Heuristic Evaluation	*Usability Testing*	*Guidelines*	*Cognitive Walkthrough*
105 core errors identified	31 core errors identified	35 core errors identified	35 core errors identified
– Identifies more problems – Identifies more serious problems – Low cost – Doesn't require implementation	– Identifies serious, recurring problems – Avoids low priority problems	– Identifies recurring, general problems – Can be used by software developers – Does not require implementation	– Helps define user' goals, assumption – Can be used by software developers – Does not require implementation
– Requires UI expertise – Requires several evaluators	– Requires UI expertise – High cost – Misses consistency problems – Requires implementation	– Misses some severe problems	– Needs task model – Tedious – Misses general, recurring problem

Kinds of Users for Testing [Booth]

- Friendly users
  - Some technical knowledge
  - Can make suggestions that naïve users perhaps could not
- Hostile users
  - People with no investment in system
  - May be useful at finding ways of breaking system
- Simulating users
  - If progress of naïve users recorded, can be later replicated by designers
  - Form of informed cognitive walkthrough
  - Saves expense of live users for incremental modifications of system

10 Steps for Observing Users [Gomoll and Nicol]

- Over all goal: observe real users using system, discover usability problems
- Important that this be carried out in a controlled setting, as opposed to asking them for their opinion after the fact, or using a questionnaire

Step 1: Set up Observation

- Write the tasks
  - Design tasks for users to work through
  - Should be real tasks, as motivated by task model
  - Should be specific, phrased in terminology of user
  - Focus on part of product under study
- Recruit the users
  - People with same experience level as user
  - Should not be people familiar with the product
- Create realistic situation
  - Natural environment, free from interruptions
  - Can use tape, video

Step 2: Describe Purpose of Observation

- Emphasize you are testing the product, not the users
  - Want them to understand that if they run into problems, the problem is with the system, not them

Step 3: Tell user they can quit any time

- Contributes to user's feeling that they are not being tested
- Lowers stress
- Basic ethical requirement on human experimentation

Step 4: Talk about, demonstrate equipment in room

- Describe any recording, monitoring equipment
- Removing mystery in process to lower anxiety of users

Step 5: Explain how to “think out loud”

- As users are carrying out their task, they should say their thoughts out loud
  - This takes some practice - requires demonstration
- Important for users to be comfortable
  - If they are nervous, they will not talk out loud

Step 6: Explain that you will not provide help

- Users should be able to solve their problems on their own
- Difficult for tester not to jump in!!
- If they are really stuck, give threshold of 3 minutes, then give just enough help to proceed
- Tell them “Even though I won’t be able to answer most of your questions, I’ll note them and answer them later”
  - And do!

Step 7: Describe tasks and introduce product

- Give them written instructions
  - Walk through them - not enough to give them a piece of paper
- Demonstrate anything that is not assumed prior knowledge
  - Do not demo the parts of the product you are trying to evaluate

Step 8: Begin observation

- Ask if they have any questions
- Begin once questions have been answered

Step 9: Conclude observation

- After the users are finished
  - explain what you were trying to find out
  - answer any remaining questions (including the ones you wrote down while they were carrying out the task)
  - discuss any interesting behaviours you want the user explain
  - ask users for over all impressions

Step 10: Use the results

- Remember to blame the system, not the user!!!