CS160 Midterm Review

Matthew Kam

Mar 3, 2003

Assignments Concerns

Administrivia

- Thursday (Mar 6, 2003) office hours shifted to Tuesday (Mar 4, 2003)
  - 6:30-7:30pm, couch and exhibit area outside 306 Soda
  - **Note time and venue change!**
  - Run as last-minute Q&A session
  - Covered by Hesham Kamel
  - I'm out of town at U. of Washington, Seattle from Tuesday to Saturday for user study

Design Question

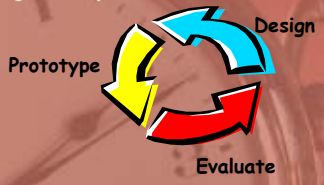
- Review assignment handouts
- Understand why each section has to be part of assignment report
- Understand how to carry out task for each section
- Easy if each group member had contributed to *every* part of project

Human-Centered Design

- Who is going to use the system?
- What are their characteristics, goals and desires?
- Choose representative tasks and analyze them
- Rough out a design (plagiarize as needed)
- Rethink the design
- Create a prototype
- Test it with users
- Iterate
- Build a production version (and ship it!)
- Track use
- Evolve the design

Iterate!

- Testing will expose problems with various severity
- You can then attack those problems in order of severity - and work on features in order of value
- Beware of interactions between design elements - fixing one may break another



Personas

- Personas are ***concrete**** representations of the user group as individuals.
- Things to strive for in a good persona:
  - Attributes (age, gender, occupation)
  - Likes, dislikes
  - Values and desires (or life's goals)
- A good persona is generative (of ideas) – a good fictional character.

* Concrete representation is the opposite of abstract representation – it widens the designer's perspective while abstraction narrows it.

Personas

- Why use personas?
  - Avoids the "elastic" user
    - Programmers bend, stretch and adapt the software for the user, not user bending and adapting to software
    - Makes it difficult for programmers to distort the users' goals and needs
  - Communication within team
    - End feature debates
  - Negative personas
    - Someone you explicitly don't want to design for

Personas

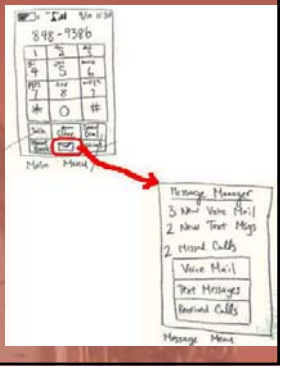
- "The essence of good interaction design is devising interactions that let users achieve their practical goals without violating their personal goals."
- Goals vs. tasks
  - A goal is an end condition
  - A task is an intermediate process required to achieve the goal
  - Tasks change as technology changes, but goals tend to remain stable
  - Programmers do task-directed design

What Should Tasks Look Like?

- Say what the user wants to do, but not how the user would do it
  - allows comparing different design alternatives
- They should be very specific
  - forces us to fill out description w/ relevant details
  - say who the users are (use personas or profiles)
    - design can really differ depending on who
    - name names (allows getting more info as necessary)
    - characteristics of the users (job, expertise, etc.)
- Some should describe a complete job
  - forces us to consider how features work together

Scenarios

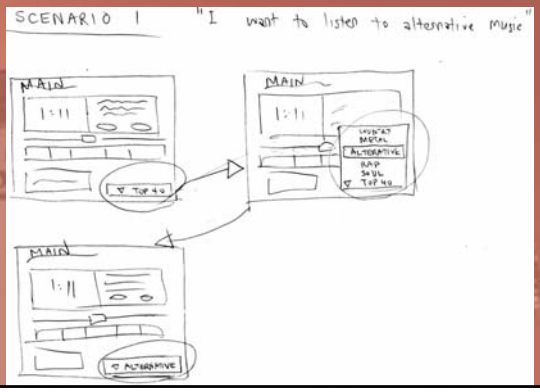
- Produce scenarios for each task
  - what user has to do & what they would see
  - step-by-step performance of task
- Scenarios are **design specific**, tasks aren't
- Scenarios force us to
  - show how various features will work together
  - settle design arguments by seeing examples
    - only examples –> sometimes need to look beyond
- Show users storyboards
  - sequences of sketches showing screens
  - actions users can take



Task ≠ Scenario

- Task e.g.:
  "It's 1 PM, you just had lunch, and you are now at your Political Science 2 lecture. The professor is explaining about the cult of personality, and because you were phasing out for a few seconds, you missed his main points, and you are now confused. Hoping he will clarify what he means by cult of personality, you decide to send feedback that this part of lecture is hard to understand."
- Scenario e.g.:
  "It's 1 PM, Mark has just had lunch, and is now at Political Science 2 lecture. ... Hoping that the professor will clarify what he means by cult of personality, Mark decides to inform the professor that this part of lecture is hard to understand. He looks for the drop-down box for lecture feedback on the upper right-hand corner of his Tablet PC screen, and taps on it to reveal a list of options. He scans them visually until he finds "confused," and taps on it to send his feedback."

Storyboard



Contextual Inquiry

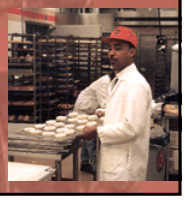
- Way of understanding users' needs and work practices
- Design happens in teams
  - design team: programmer, marketing, quality assurance, producer, more..
  - user teams: the customers are also part of a team that does something

Principles: Context

- Go to the workplace & see the work as it unfolds
- People summarize, but we want details
- Keep it concrete when people start to abstract
  - "We usually get reports by email", ask "Can I see one?"
- Look for skipped steps, ask the user to fill them in.

Master-Apprentice model

- Master – Apprentice model allows customer to teach us what they do!
  - Master does the work & talks about it while working
  - We interrupt to ask questions as they go
  - Each step reminds the user of the next
  - Skill knowledge is usually tacit (cant put it in books)
  - Studying many tasks, the designer can abstract away
  - Sometimes literal apprenticeship works (Matsushita "Home Bakery")!



Principles: Partnership

- Stick with master-apprentice relationship; avoid lapsing into other models, i.e.
  - avoid interviewer/interviewee (stops work), expert/novice (set expectations at the start)
  - partnership allows more apprentice interaction: its OK to be a designer and interrupt!
  - … but go back "in role":
  - alternate between watching & probing (*withdrawal & return*)

Principles: interpretation

- Good facts are only the starting point
  - designs based on interpretations
- Validate & rephrase
  - run interpretations by the user to see if you are right
  - share ideas to check your reasoning (walk the chain back)
  - people will be uncomfortable until the phrasing is right
  - need to be committed to hearing what the customer is really saying ("Huh?", "Umm…", "Yes, but…")

Principles: Focus

- Interviewer needs data about specific kind of work
  - "steer" conversation to stay on useful topics
- Respect triggers (flags to change focus – changing understanding)
  - shift attention (some one walks in)
  - surprises (you know it is "wrong")
  - treat every utterance by the customer as a potential clue to something important

Caveats of User-Centered Design Techniques

- Politics
  - "agents of change" can cause controversy
  - get a sense of the organization & bond w/ interviewee
  - important to get buy-in from all those involved
- Users are not always right
  - cannot anticipate new technology accurately
  - your job is to build system users will want
    - not system users say they want
    - be very careful about this (you are outsider)
      - if you can't get users interested in your hot idea, you're probably missing something
- Design forever without prototyping
  - rapid prototyping, evaluation, & iteration is key

Fidelity in Prototyping

- Fidelity refers to the level of detail
- High fidelity
  - prototypes look like the final product
- Low fidelity
  - artists renditions with many details missing

Why Do We Prototype?

- Get feedback on our design faster
  - saves money
- Experiment with alternative designs
- Fix problems before code is written
- Keep the design centered on the user

Why Use Low-fi Prototypes?

- Traditional methods take too long
  - sketches -> prototype -> evaluate -> iterate
- Can simulate the prototype
  - sketches -> evaluate -> iterate
  - sketches act as prototypes
    - designer "plays computer"
    - other design team members observe & record
- Kindergarten implementation skills
  - allows non-programmers to participate

Hi-fi Prototypes Warp

- Perceptions of the tester/reviewer?
  - formal representation indicates "finished" nature
    - comments on color, fonts, and alignment
- Time?
  - encourage precision
    - specifying details takes more time
- Creativity?
  - lose track of the big picture

Advantages of Low-fi Prototyping

- Takes only a few hours
  - no expensive equipment needed
- Can test multiple alternatives
  - fast iterations
    - number of iterations is tied to final quality
- Almost all interaction can be faked

Wizard of Oz Technique

- Faking the interaction. Comes from?
  - from the film "The Wizard of OZ"
    - "the man behind the curtain"
- Long tradition in computer industry
  - prototype of a PC w/ a VAX behind the curtain
- Much more important for hard to implement features
  - Speech & handwriting recognition

Teams

- *"A team is a small number of people with complementary skills who are committed to a common purpose, set of performance goals, and approach for which they hold themselves mutually accountable."* - K&S
- Unpacking this statement...

Common purpose

- Common purpose is helped by group affinity (people liking each other), but that is not necessary.
- Common purpose can also be achieved by interdependence (group members evaluated and rewarded together)

Goal setting

- Defines specific work products
- Facilitates communication and constructive conflict
- Attainable: maintain focus
- Leveling effect: focus on task rather than status
- Defines small wins as part of the larger purpose
- Goals are compelling

Building teams: Urgency

- Establish Urgency
- Purpose is worthwhile
- There is a clear way to move ahead

Building teams: Select for Skill

- Manager should choose team based on skills of members, and potential skills.
- Should personality be a factor?… stay tuned.

Setting rules of behavior

- E.g. no phone calls in meetings
- no sacred cows
- one conversation at a time
- encourage wild ideas
- no finger-pointing...

Set a few immediate goals

- Make them performance-oriented
- When results occur, the team starts feeling like a team

Bring in fresh facts and ideas

- Fact: teams do not share enough information (Hinds 199x).
- Regular updates and exchanges are much more valuable than they seem.
- This builds a sense of community and common knowledge.

Spend time together

- Casual or “unstructured” interactions are very important for building shared context.
- Putting people in the same space is the best way to do that.
- Recreating this online is a bit of a challenge.

Positive Feedback

- Don’t miss an opportunity to reward or encourage legitimate effort.
- Positive reinforcement encourages more effort and performance beyond expectations.

Discount Usability Engineering

- Cheap
  - no special labs or equipment needed
  - the more careful you are, the better it gets
- Fast
  - on order of 1 day to apply
  - standard usability testing may take a week
- Easy to use
  - can be taught in 2-4 hours

Discount Usability Engineering

- Based on:
  - Scenarios
  - Simplified thinking aloud
  - Heuristic Evaluation

Other budget methods

- Walkthroughs
  - put yourself in the shoes of a user
  - like a code walkthrough
- Low-fi prototyping
- Action analysis
  - GOMS (add times to formal action analysis)
- On-line, remote usability tests
- Heuristic evaluation

Heuristic Evaluation

- Developed by Jakob Nielsen
- Helps find usability problems in a UI design
- Small set (3-5) of evaluators examine UI
  - independently check for compliance with usability principles ("heuristics")
  - different evaluators will find different problems
  - evaluators only communicate afterwards
    - findings are then aggregated
- Can perform on working UI or on sketches

Heuristic Evaluation Process

- Evaluators go through UI several times
  - inspect various dialogue elements
  - compare with list of usability principles
  - consider other principles/results that come to mind
- Usability principles
  - Nielsen's "heuristics"
  - supplementary list of category-specific heuristics
    - competitive analysis & user testing of existing products
- Use violations to redesign/fix problems

How to Perform Evaluation

- At least two passes for each evaluator
  - first to get feel for flow and scope of system
  - second to focus on specific elements
- If system is walk-up-and-use or evaluators are domain experts, no assistance needed
  - otherwise might supply evaluators with scenarios
- Each evaluator produces list of problems
  - explain why with reference to heuristic or other information
  - be specific and list each problem separately

Revised Heuristics

- **H2-1:** visibility of system status
- **H2-2:** Match system and real world
- **H2-3:** User control and freedom
- **H2-4:** Consistency and standards
- **H2-5:** Error prevention
- **H2-6:** Recognition rather than recall
- **H2-7:** Flexibility and efficiency of use
- **H2-8:** Aesthetic and minimalist design
- **H2-9:** Help users recognize, diagnose and recover from errors
- **H2-10:** Help and documentation

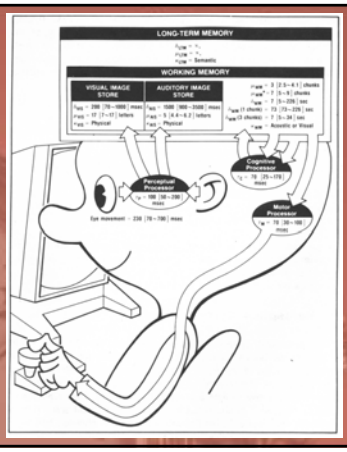
Results of Using HE (cont.)

- Single evaluator achieves poor results
  - only finds 35% of usability problems
  - 5 evaluators find ~ 75% of usability problems
  - why not more evaluators???? 10? 20?
    - adding evaluators costs more
    - many evaluators won't find many more problems

Why Model Human Performance?

- To test understanding
- To predict influence of new technology

Model Human Processor



MHP Basics

- Based on empirical data
  - years of basic psychology experiments in the literature
- Three interacting subsystems
  - perceptual, motor, cognitive
- Sometimes serial, sometimes parallel
  - serial in action & parallel in recognition
    - pressing key in response to light
    - driving, reading signs, & hearing at once
- Parameters
  - processors have cycle time (T) ~ 100-200 ms
  - memories have capacity, decay time, & type

Principles of Operation (cont.)

- Variable Cog. Processor Rate Principle
  - CP cycle time T_c is shorter when greater effort
  - induced by increased task demands/information
  - decreases with practice

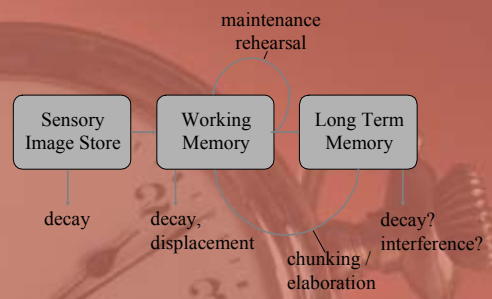
Principles of Operation (cont.)

- Fitts' Law
  - moving hand is a series of microcorrections
    - correction takes $T_p + T_c + T_m$ = 240 msec
  - time T_{pos} to move the hand to target size S which is distance D away is given by:
    - $T_{pos} = a + b \log_2 (D/S + 1)$
  - summary
    - time to move the hand depends only on the ***relative precision*** required

Principles of Operation (cont.)

- Power Law of Practice
  - task time on the nth trial follows a power law
    - $T_n = T_1 n^{-a} + c$, where a = .4, c = limiting constant
    - i.e., you get faster the more times you do it!
    - applies to skilled behavior (sensory & motor)
    - does not apply to knowledge acquisition or quality

Stage Theory



Memory

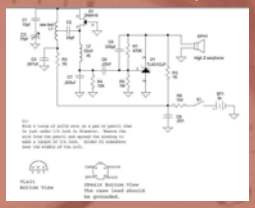
- Interference
  - two strong cues in working memory
  - link to different chunks in long term memory
- Why learn about memory?
  - know what's behind many HCI techniques
  - helps you understand what users will "get"
  - aging population of users

Recognition over Recall

- Recall
  - info reproduced from memory
- Recognition
  - presentation of info provides knowledge that info has been seen before
  - easier because of cues to retrieval
- We want to design UIs that rely on recognition!

Structural vs. Functional Models

- A structural model explains what the system does independent of use (it's a system-centered model).
- A functional model explains what the system does to assist a user's task (it's a user-centered model)



Metaphor

- Since functional models draw on past experience and not everyone has computer experience, its useful to draw on the real world.

- Hence the “desktop metaphor”:
  - Directories are like folders
  - Files are like sheets of paper
  - Windows are like ?:
  - Menus are like menus
  - Deleting is like putting in the trash
  - Running an application program is like opening the doc.
  - Copy to buffer and restore is like cut-and-paste...

Metaphor: Difficulties

- The metaphor may create expectations that are false along with the true ones:
  - Can I shred this file instead of putting in the trash can?
- Our understanding is “functional” rather than “structural”. That means understanding is relative to how *we* do things.
- For instance “work” has many meanings:
  - Something we enjoy, or dislike
  - Something that is primarily physical, vs. intellectual
  - Something that leads to a goal, vs. something we just do

Composite Metaphors

- People are usually happy stepping out of the metaphor:
  - Scroll bars
  - Resizing
  - Iconifying
- Users can use multiple metaphors at once, or other models based on familiar practice
- Over time, the original metaphor becomes redundant and the user has a new concept and set of skills.

Conceptual Models

- Because of the difficulties with metaphors, the goal of interface design is typically to come up with a clean “conceptual model”.
- A conceptual model is the user’s model of what happens. A good one acknowledges human ability:
  - Simplicity, how much to learn, how easy to apply?
  - Limited short-term memory
  - Expensive long-term memory
  - Stimulus-action fusion

Good luck