Human-Computer Interaction
Round 7

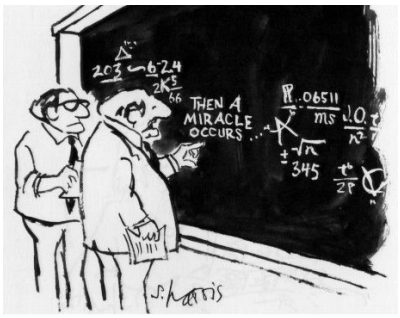
I6: Heuristic Evaluation

- (Organized) comprehensive list
- Due next week
- Shared with class

T5: Paper Prototyping #2

- Big deal … Get going!

UI Design



Why is UI Design Hard?

- Infinite possibilities
- Many, many published heuristics, guidelines, rules
- Everything comes together
  - Desired functionality
  - User abilities, knowledge
  - Aesthetics
  - Conventions
  - ...

Best solution approach

- Try (& evaluate) lots of stuff
- Parallel design
  - Generate several options at once, by different designers
- Cyclic design
  - Generate, evaluate, repeat

Golden rules and heuristics

- "Broad brush" design rules
- Useful check list for good design
- Better design using these than using nothing!
- Different collections e.g.
  - Nielsen's 10 Heuristics (see Chapter 9)
  - Shneiderman's 8 Golden Rules
  - Norman's 7 Principles

Shneiderman's 8 Golden Rules

1. *Strive for consistency*
2. *Enable frequent users to use shortcuts*
3. *Offer informative feedback*
4. *Design dialogs to yield closure*
5. *Offer error prevention and simple error handling*
6. *Permit easy reversal of actions*
7. *Support internal locus of control*
8. *Reduce short-term memory load*

Norman's 7 Principles

1. *Use both knowledge in the world and knowledge in the head.*
2. *Simplify the structure of tasks.*
3. *Make things visible: bridge the gulfs of Execution and Evaluation.*
4. *Get the mappings right.*
5. *Exploit the power of constraints, both natural and artificial.*
6. *Design for error.*
7. *When all else fails, standardize.*

HCI design patterns

- An approach to reusing knowledge about successful design solutions
- Originated in architecture: Alexander
- A pattern is an invariant solution to a recurrent problem within a specific context.
- Examples
  - Light on Two Sides of Every Room (architecture)
  - Go back to a safe place (HCI)
- Patterns do not exist in isolation but are linked to other patterns in *languages* which enable complete designs to be generated

HCI design patterns (cont.)

- Characteristics of patterns
  - capture design practice not theory
  - capture the essential common properties of good examples of design
  - represent design knowledge at varying levels: social, organisational, conceptual, detailed
  - embody values and can express what is humane in interface design
  - are intuitive and readable and can therefore be used for communication between all stakeholders
  - a pattern language should be generative and assist in the development of complete designs.

Expert analysis

- **Cognitive walkthrough**
- **Heuristic evaluation**
- Use of models (Ch 12, e.g. GOMS)
- Previous work

Cognitive walkthrough

- Start with
  - Specifications
  - Task descriptions
  - Actions needed to complete tasks
  - Indication of who users are

(Similar to what you have been doing)

Cognitive walkthrough

- For each action, step through and "try to tell a believable story" about:
  - Do actions match goal at any point?
  - Will users see action is available?
  - Will users know action is one they need?
  - If action taken, will users understand feedback they get?

Cognitive walkthrough

- Discount technique
- 5 evaluators find 75% of problems
- Use Neilson's ten heuristics
- Note severity of problems

Usability heuristics

- Many to choose from
  - Neilsen's 10 principles
  - Shneiderman's 8 golden rules
  - Tognazzini's 16 principles
  - Norman's rules from Design of Everyday Things
  - Mac, Windows, Gnome, KDE, Java guidelines
- Help designers choose design alternatives
- Help evaluators find problems in interfaces ("heuristic evaluation")

Some we've already discussed

- User-centered design
  - Know your users
  - Understand their tasks
- Fitts's Law
  - Size and proximity of controls should relate to their importance
  - Tiny controls are hard to hit
  - Screen edges are precious
- Memory
  - Use chunking to simplify information presentation
  - Minimize working memory
  - Rely more on recognition than recall
- Color guidelines
  - Don't depend solely on color distinctions (color blindness)
- Principles of direct manipulation
  - Affordances
  - Feedback
- Grouping

Nielsen's Heuristics (some) 1. Match the Real World

- aka "Speak the User's Language"
- Use common words, not techie jargon
  - But use domain-specific terms where appropriate
- Don't put limits on user defined names
- Allow aliases/synonyms in command languages
- Metaphors are useful but may mislead

Nielsen's Heuristics (some)
2. Consistency and Standards

- Principle of Least Surprise
  - Similar things should look and act similar
  - Different things should look different
- Other properties
  - Size, location, color, wording, ordering, …
- Follow platform standards
- Kinds of Consistency
  - Internal
  - External
  - Metaphorical

Nielsen's Heuristics (some)
4. User Control and Freedom

- Provide undo
- Long operations should be cancelable
- All dialogs should have a cancel button

Nielsen's Heuristics (some)
5. Visibility of System Status

- Keep user informed of system state
  - Cursor change
  - Selection highlight
  - Status bar
- Response time
  - < 0.1 s: seems instantaneous
  - 0.1-1 s: user notices, but no feedback needed
  - 1-5 s: display busy cursor
  - > 1-5 s: display progress bar

Nielsen's Heuristics (some) 6. Flexibility and Efficiency

- Provide easily-learned shortcuts for frequent operations
  - Keyboard accelerators
  - Command abbreviations
  - Bookmarks
  - History

Nielsen's Heuristics (some) 7. Error Prevention

- Selection is less error-prone than typing
- Disable illegal commands
- Description Error
  - when two actions are too similar
  - e.g., case sensitivity
  - different things should look and act different
- Mode Error
  - Eliminate modes
  - Visibility of mode
  - Spring-loaded or temporary modes

Nielsen's Heuristics (some) 8. Recognition, Not Recall

- Use menus, not command languages
- Use combo boxes, not textboxes
- Use generic commands where possible (Open, Save, Copy Paste)
- All needed information should be visible

Nielsen's Heuristics (some)
9. Error Reporting, Diagnosis, Recovery

- Be precise; restate user's input
  - Not "Cannot open file", but "Cannot open file named paper.doc"
- Give constructive help
  - why error occurred and how to fix it
- Be polite and non-blaming
  - Not "fatal error", not "illegal"
- Hide technical details (stack trace) until requested

Nielsen's Heuristics (some)
10. Aesthetic and Minimalist Design

- "Less is More" / KISS
  - Omit extraneous info, graphics, features

Action

- My rule: action oriented
- Buttons = verbs!

Tog's 16 Principles

http://www.asktog.com/basics/firstPrinciples.html

- **Anticipation**
- Autonomy
- Color blindness
- Consistency
- **Defaults**
- Efficiency
- **Explorable interfaces**
- Fitts's Law
- Human interface objects
- Latency reduction
- **Learnability**
- Metaphors
- **Protect users' work**
- **Readability**
- **Track state**
- **Visible navigation**

Some of Tog's Principles

- **Anticipation**
  - Put all needed information and tools within the user's easy reach.
  - Why a File Save dialog box needs a way to create a new folder
- **Defaults**
  - Common answers already filled into a form
  - Speeds up learning
  - Increases overall efficiency
  - Make "fragile"
  - Avoid word "default"
- **Explorable interfaces**
  - Support user's poking around (need undo/cancel)

Schneiderman's 8 Golden Rules

- Consistency
- Shortcuts
- Feedback
- **Dialog closure**
- Simple error handling
- Reversible actions
- Put user in control
- Reduce short-term memory load

Additional generic useful design advice

- Use **Organization** to create groups/regions
  - White Space
  - Alignment
  - Borders and Bounding Boxes
  - Containment
- Use **Coding** to show the types and properties of objects
  - Size
  - Shape (or picture/icon)
  - Color
  - Explicit Labeling
  - Tool tips
- Principle of hierarchy
  - overview, then zoom for details

User Participation for Evaluation

- "Computer science [students]: they are simply not representative of the intended user population"
- Sample size:
  - Nielsen and Landauer
    - One person (1/3 problems)
    - Little to be gained from 5+
  - Book: recommends at least 10
  - Recent papers ... Even more

Statistical measures

- Look at the data!
  - Intuition
  - Find outliers
  - Normal?

Think Aloud

- Observation not enough
  - Misses decision processes and attitude
- Simple
- Alternative: cooperative evaluation
  - User collaborator in evaluation

Physiological monitoring

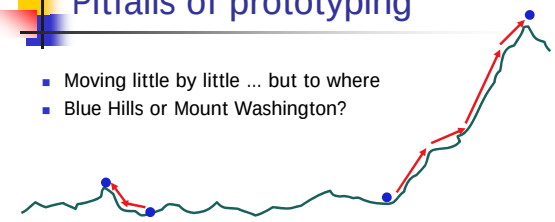
- Eye tracking
- HR
- Sweat
- Breathing rate
- Electrical activity in muscle (EMG)
- Electrical activity in brain (EEG)
- fMRI

Overview

- Tables 9.4-9.7 worth a look

Pitfalls of prototyping

- Moving little by little … but to where
- Blue Hills or Mount Washington?



1. Need a good start point
2. Need to understand what is wrong
3. Need paper napkins not cloth napkins

Tohidi article: Getting the Right Design and the Design Right: Testing Many is Better than One

- Paper prototyping: getting design right
- Most important: getting the right design
- Theory: simultaneous exploration of multiple ideas might help
  - Common in traditional design arts

Tohidi article: Getting the Right Design and the Design Right: Testing Many is Better than One

- Problem: Human reluctance to criticize
  - Don't want to be "negative people"
  - Don't want to reflect poorly on own abilities
  - Don't want to hurt the designer's feelings
- Solution? With multiple options, easier to criticize one; identify best

Tohidi article: Experiment

- 3 paper prototypes of same interface
- Investigate impact of being shown 1 vs 3 ideas on:
  - Design ratings
  - User criticism
  - Number of ideas and suggestions for designs

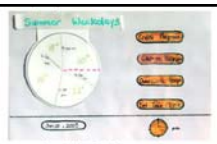


Figure 1. The "Circular" paper prototype

Figure 2. The "Tabular" paper prototype

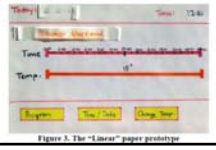


Figure 3. The "Linear" paper prototype

Tohidi article: Experiment

- Design ratings
  - Faint praise
  - "We have not made up our mind"
  - Criticize without being negative
- User criticism
  - Increased criticism

Tohidi article: Experiment

- Number of ideas and suggestions for designs
  - Did not get more suggestions for improvement with 3 designs
  - Why? Usability testing vs participatory design
    - Making suggestions involves "speculation, and stepping out on a limb for which they had no training, experience, or language"
    - Don't want to risk exposure as naïve relative to expert

Getting the right design

- "Once a design is prototyped and tested, it hardly ever gets rejected by the users"
- Of 36 participants seeing single design, nobody requested redesign
- 3 out of 12 in the multiple design condition rejected a design

User Sketches

(Tohidi et al. 06)

- Sketching: "reflective" vs "reactive" feedback
- After paper prototyping exercise, asked to sketch "ideal" design

User Sketches

(Tohidi et al. 06)

- Sketching: “reflective” vs “reactive” feedback
- After paper prototyping exercise, asked to sketch “ideal” design

3.9 min/sketch

Advantages

- Look for patterns
- Look for unique components and new ideas
- People will say they have no changes, then make changes
- Forcing people to reflect on their own ideas
- See the bias in your ideas!

Paper Prototyping

- Xueming Wu
- Chen Chu
- Yiyun Ma

Paper Prototyping

- Aida Ehyaei
- Lei Wang
- Nima Attaran Rezaei

T5: Paper Prototyping #2

- Big deal … Get going!

Research Papers – Health Interfaces #2

- Lee, Kiesler, and Forlizzi, Mining Behavioral Economics to Design Persuasive Technology for Healthy Choices, CHI 2011 (Presenter: Nima Attaran Rezaei)
- Iqbal et all, Hang on a Sec! Effects of Proactive Mediation of Phone Conversations While Driving, CHI 2011 (Presenter: Pulkit Misra)
- Lee and Dey, Reflecting on Pills and Phone Use: Supporting Awareness of Functional Abilities for Older Adults, CHI 2011
- Maitland and Chalmers, Designing for Peer Involvement in Weight Management, CHI 2011

Linehan article

How do many educational apps break these rules?

- What makes a game?
  - Short, medium, long-term goals
  - Player must take actions to reach goals
  - Immediate, appropriate, specific feedback
  - Rewards for achievement
  - Teach skills (break into components)
  - Demonstrate skills to advance
  - Where options, no one action obviously correct

Linehan article

- Presenting feedback
  - Positive reinforcement (add positive stimulus to make behavior more likely)
  - Negative reinforcement (remove aversive stimulus to make behavior more likely)
  - Positive punishment (add aversive stimulus to make behavior less likely)
  - Negative punishment (remove positive stimulus to make behavior less likely)

To do

- Read
  - Excepts from Design Basics Index (on Blackboard)
  - Universal design (Dix Ch 10)
  - 4 research papers
- Do Individual Homework I6 – Heuristics
- Do Team Homework T5 – Paper Prototyping 2 (due in 2 weeks)