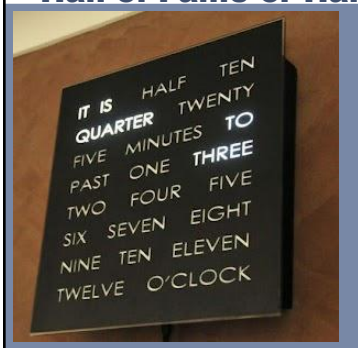
User Interface Design, Prototyping, and Evaluation

(1) Action Analysis
(2) Automated Evaluation

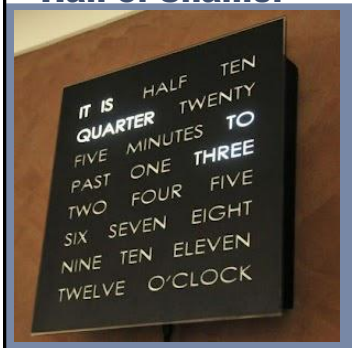
Prof. James A. Landay
University of Washington
Spring 2012

April 3, 2012

Hall of Fame or Hall of Shame?



Hall of Shame!



- Clock
- What is the purpose?
- This slows you down!

User Interface Design, Prototyping, and Evaluation

(1) Action Analysis
(2) Automated Evaluation

Prof. James A. Landay
University of Washington
Spring 2012

April 3, 2012

Outline

- Action analysis
- GOMS? What's that?
- The G, O, M, & S of GOMS
- How to do the analysis
- Automated evaluation tools

Action Analysis Predicts Performance

- Cognitive model ?
  - model some aspect of human understanding, knowledge, intentions, or processing
  - two types
    - competence
      - predict behavior sequences
    - performance
      - predict performance, but limited to routine behavior
- Action analysis uses performance model to analyze goals & tasks
  - generally done hierarchically (similar to TA)

GOMS – Most Popular AA Technique

- Family of UI modeling techniques
  - based on Model Human Processor cognitive model
- GOMS stands for (?)
  - Goals
  - Operators
  - Methods
  - Selection rules
- Input: detailed description of UI/task(s)
- Output: qualitative & quantitative measures

Quick Example

- Goal (the big picture)
  - go from hotel to the airport
- Methods (or subgoals)?
  - walk, take bus, take taxi, rent car, take train
- Operators (or specific actions)
  - locate bus stop; wait for bus; get on the bus;...
- Selection rules (choosing among methods)?
  - Example: Walking is cheaper, but tiring and slow
  - Example: Taking a bus is complicated abroad

Goals

- Something the user wants to achieve
- Examples?
  - go to airport
  - delete file
  - create directory
- Hierarchical structure
  - may require many subgoals

Methods

- Sequence of steps to accomplish a goal
  - goal decomposition
  - can include other goals
- Assumes method is *learned* & *routine*
- Examples
  - drag file to trash
  - retrieve from long-term memory command

Operators

- Specific actions (small scale or atomic)
- Lowest level of analysis
  - can associate with times
- Examples
  - Locate icon for item on screen
  - Move cursor to item
  - Hold mouse button down
  - Locate destination icon
  - User reads the dialog box

Selection Rules

- If > 1 method to accomplish a goal, Selection rules pick method to use
- Examples
  - IF <*condition*> THEN accomplish <*GOAL*>
  - IF <*car has automatic transmission*> THEN <*select drive*>
  - IF <*car has manual transmission*> THEN <*find car with automatic transmission*>

GOMS Output

- Execution time
  - add up times from operators
  - assumes ?
    - experts (mastered the tasks) & error free behavior
  - very good rank ordering
  - absolute accuracy ~10-20%
- Procedure learning time (NGOMSL only)
  - accurate for relative comparison only
  - doesn't include time for learning domain knowledge

GOMS Output Used To

- Ensure frequent goals achieved quickly
- Making hierarchy is often the value
  - functionality coverage & consistency
    - does UI contain needed functions?
    - consistency: similar tasks performed similarly?
  - operator sequence
    - in what order are individual operations done?

How to do GOMS Analysis

- Generate task description
  - pick high-level user Goal
  - write Method for accomplishing Goal
    - may invoke subgoals
  - write Methods for subgoals
    - this is recursive
    - stops when Operators are reached
- Evaluate description of task
- Apply results to UI
- Iterate!

Comparative Example – Unix shell

- Goal: Delete a File
- Method for accomplishing goal of deleting file
  - retrieve from Long term memory that command verb is "rm"
  - think of directory name & file name and make it the first listed parameter
  - accomplish goal of entering & executing command
  - return with goal accomplished

Comparative Example - Windows

- Goal: Delete a File
- Method for accomplishing goal of deleting file
  - find file icon
  - accomplish goal of dragging file to trash
  - return with goal accomplished

Comparative Example – Unix shell

- Goal: Remove a directory
- Method for accomplishing goal of removing a directory
  - ?????

Comparative Example – Unix shell

- Goal: Remove a directory
- Method for accomplishing goal of removing a directory
  - accomplish goal of making sure directory is empty
  - retrieve from long term memory that command verb is 'rmdir'
  - think of directory name and make it the first listed parameter
  - accomplish goal of entering & executing command
  - return with goal accomplished

Comparative Example - Windows

- Goal: Remove a directory
- Method for accomplishing goal of removing a directory
  - ????

Comparative Example - Windows

- Goal: Remove a directory
- Method for accomplishing goal of removing a directory
  - find folder icon
  - accomplish goal of dragging folder to trash
  - return with goal accomplished
- Note the consistency with delete file on the Windows! This makes it much easier.

What GOMS Can Model

- Task must be goal-directed
  - some activities are more goal-directed
    - creative activities may not be as goal-directed
- Task must use routine cognitive skills
  - as opposed to problem solving
  - good for things like machine operators

Applications of GOMS

- Compare different UI designs
- Profiling (time)
- Building a help system? Why?
  - modeling makes user tasks & goals explicit
  - can suggest questions users might ask & the answers

Real-world GOMS Applications

- Keystroke Level Model (KLM)
  - Mouse-based text editor
  - Mechanical CAD system
- NGOMSL
  - TV control system
  - Nuclear power plant operator's associate
- CPM-GOMS
  - Telephone operator workstation

Advantages of GOMS

- Gives qualitative & quantitative measures
- Model explains the results
- Less work than *large* user study – no users!
- Easy to modify when UI is revised
- Research: tools to aid modeling process since it can still be tedious

Disadvantages of GOMS

- Not as easy as HE, guidelines, etc.
- Takes lots of time, skill, & effort
- Only works for goal-directed tasks
- Assumes tasks performed by *experts without error*
- Does not address several UI issues,
  - readability, memorizability of icons, commands…

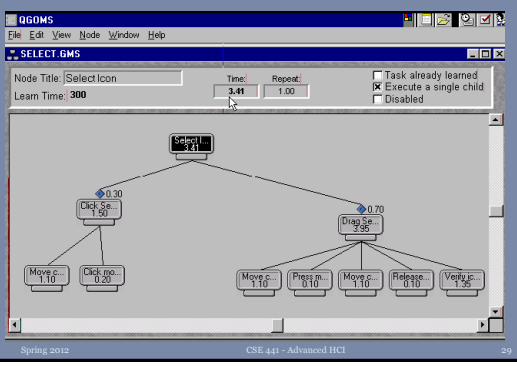
Rapid Iterative Design is the Best Practice for Creating Good UIs

We have seen how computer-based tools can improve the Design (e.g., Denim) & Prototyping (e.g., JustInTime) phases

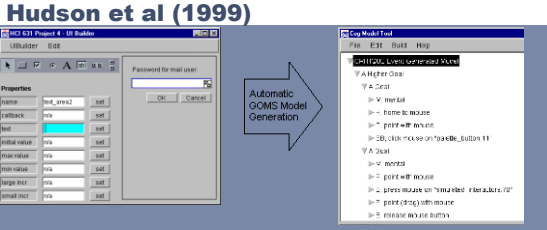
Automated GOMS Tools

- Can save, modify & re-use the model
- Automation of execution time calculation, etc.

QGOMS tool

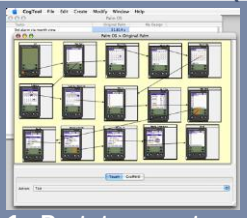


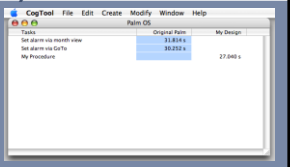
CRITIQUE
Hudson et al (1999)



1. Prototype system by programming
   - in this case with the SubArctic toolkit
2. Demonstrate a task
   - record events
   - apply rules
3. Automatically generate KLMs
4. Semi-automatically generate classic GOMS models

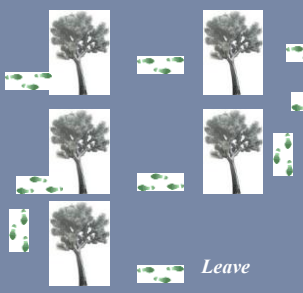
CogTool
John & Salvucci (2005)





1. Prototype system by *storyboarding*
2. Demonstrate a task
   - record events
   - apply rules
3. Automatically generate ACT-R model

The Trouble With Most Web Site Analysis Tools

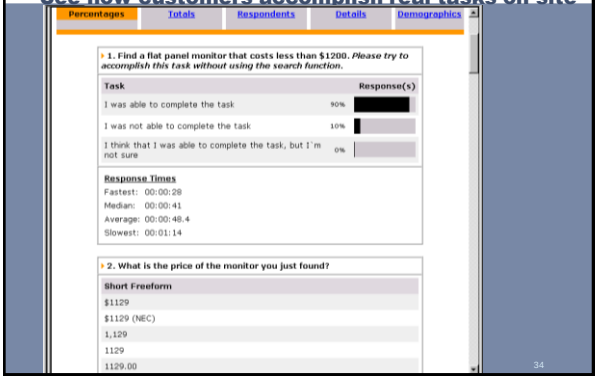


Leave

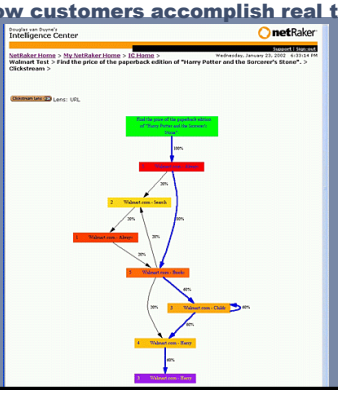
Unknowns
- Who?
- What?
- Why?
- Did they find it?
- Satisfied?

NetRaker Usability Research
See how customers accomplish real tasks on site

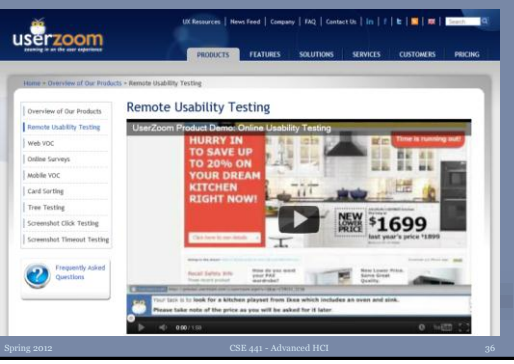
NetRaker Usability Research
See how customers accomplish real tasks on site



NetRaker Usability Research
See how customers accomplish real tasks on site



UserZoom



Advantages of Remote Usability Testing

- Fast
  - can set up research in 3-4 hours
  - get results in 36 hours
- More accurate
  - can run with large samples (50-200 users → stat. sig.)
  - uses real people (customers) performing tasks
  - natural environment (home/work/machine)
- Easy-to-use
  - templates make setting up easy
- Can compare with competitors
  - indexed to national norms

Disadvantages of Remote Usability Testing

- Miss observational feedback
  - facial expressions
  - verbal feedback (critical incidents)
- Need to involve human participants
  - costs some amount of money (typically $20-$50/person)
- People often do not like pop-ups
  - need to be careful when using them

Summary

- GOMS
  - provides info about important UI properties
  - doesn't tell you everything you want to know about UI
    - only gives performance for expert, error-free behavior
  - hard to create model, but still easier than user testing
    - changing later is much less work than initial generation
- Automated usability
  - faster than traditional techniques
  - can involve more participants → convincing data
  - easier to do comparisons across sites
  - tradeoff with losing observational data

Next Time

- Group Heuristic Evaluation assignment
- Tue Lecture on Mobile UI Design