CogSysIII Lecture 9: User Modeling with GOMS

Human Computer Interaction

Ute Schmid

Applied Computer Science, University of Bamberg

last change June 26, 2007

Develop a Usable System

- Standard technique: empirical user testing
- Iterative process: testing and design revision
- Inherited from human factors in classical engineering
- Problem: too slow and too expensive for modern software development practice, esp. when difficult-to-get domain experts are target users (e.g. cockpit)
- Response: discount and inspection methods (Jacob Nielsen)
- Use 3 – 5 experts who inspect the system x
- Work with scenarios and use heuristics
- see e.g. Hugh Jackson, User Interface Design – How Can It Be Improved?

Inspection Methods

- Feature inspection: lists sequences of features used to accomplish typical tasks, checks for long sequences, cumbersome steps, steps that would not be natural for users to try, and steps that require extensive knowledge/experience in order to assess a proposed feature set.
- Consistency inspection: has designers who represent multiple other projects inspect an interface to see whether it does things in the same way as their own designs.
- Standards inspection: has an expert on an interface standard inspect the interface for compliance.
- Heuristic estimation: is a variant in which the inspectors are asked to estimate the relative usability of two (or more) designs in quantitative terms (typically expected user performance).
- Cognitive walkthrough: uses a more explicitly detailed procedure to simulate a user's problem-solving process at each step through the dialogue, checking if the simulated user's goals and memory content can be assumed to lead to the next correct action.
- Pluralistic walkthrough: uses group meetings where users, developers, and human factors people step through a scenario, discussing each dialogue element.

Discount Usability Engineering

- Heuristic evaluation working with a prototype (no full functionality, only some features)
- Reduce the standard usability guideline (with thousands of rules) to 10 heuristics, developed by Nielsen from examining a database of 249 usability problems

- Visibility of system status - The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
- Match between system and the real world - The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
- User control and freedom - Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Discount Usability Engineering cont.

- Consistency and standards - Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
- Error prevention - Even better than good error messages is a careful design which prevents a problem from occurring in the first place.
- Recognition rather than recall - Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
- Flexibility and efficiency of use - Accelerators – unseen by the novice user – may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
- Aesthetic and minimalist design - Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
- Help users recognize, diagnose, and recover from errors - Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
- Help and documentation - Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

User Modeling

- Alternative to empirical evaluation with users or experts: use engineering models for usability
- Seminal work: Card, Moran & Newell (1983) – GOMS
- using a program to behave like a user in order to automatically test an interface, either for usability or quality assurance purposes.
- The user model is usually rigged to send input events just like a user, e.g. mouse clicks and keypresses
- User models produce quantitative predictions of how well humans will be able to perform tasks with a proposed design
- used as surrogate for actual empirical user data (more rapid progress for design evaluations and revisions)

Remark

- If you have good explanatory theories about some areas of human information processing (e.g. perception of contrast, ease of recognition, memory) you can use these as guidelines for interface design and safe lots of time as well!

GOMS

- Goals, Operators, Methods, and Selection Rules
- Description of knowledge that a user must have to carry out tasks on a system
- representation of howto do it knowledge (cf. production system)
- GOMS models: descriptions of methods needed to accomplish specified goals
- Methods have a hierarchical structure (call for subgoals to be accomplished)
- If more than one method is applicable at a step: selection rule chooses an appropriate method

GOMS / KLM

- Simplest models: keystroke-level (KLM)
  task execution time is predicted by the total of times for the elementary keystroke-level actions to perform a task
- assumes an expert user
- see `http://www.rpi.edu/~glasse/HCI-HOWTO/ar01s03.html`

How to Model with KLM

- State assumptions
- Write out action sequences and convert them to KLM-GOMS operators
- Let the model run, which does
  - Insert Mentals
  - Add things up to a total amount of time using KLM times estimates

Times for KLM Operators

as empiricaly and analyticaly determined by Kieras (1993)

K	Keystroke	280 msec
T(n)	Type n characters	280*n ms
P	Point with mouse to something on the display	1100 mse
B	Press or release mouse button	100 msec
BB	Click a mouse button	200 msec
H	Home hands, either to the keyboard or mouse	400 msec
M	“Mental”, thinking	1200 mse
W(t)	Waiting for the system to respond	t msec

Writing a Model

```
Task_item: T1
  Name is First.
  Type is delete_file.
  Filename is "file1.txt".
  Enclosing_directory is "Work".
  Next is T2.

Task_item: T2
  Name is T2.
  Type is move_file.
  Filename is "file2.txt".
  Enclosing_directory is "Work".
  Destination_directory is "Documents".
  Next is None.
```

Heuristics for Mental Operations

- Rule 0: Insert M's in front of all K's or B's that are not part of argument strings proper (e.g., text or numbers). Place M's in front of all P's that select commands or that begin a sequence of direct-manipulation operations belonging to a cognitive unit.

  Direct-manipulation operations are when you issue commands by manipulating things on-screen. For example, rather than selecting something like Files Delete, you drag something to a garbage can.

  An argument is something that varies. For instance, if you burn CDs to files frequently, the files you burn are arguments. If you change fonts frequently, then the fonts are arguments. If, however, you always change a font to a particular font, say FreeMono, then you might find that the font is no longer an argument but part of the command – the user might not think of changing fonts, but FreeMono-izing a section of text. Since users process arguments and commands differently, this can make a difference.

Heuristics for Mental Operations cont.

- Rule 1: If an operator following an M is fully anticipated in an operator just previous to M, then delete the M (e.g., PMK becomes PK or PMBB becomes PBB). Fully anticipated means that, for instance here, the time it takes to perform the P operator (1100 msec) is almost as long as the time it takes to perform the M operator (1200 msec). It is assumed that the user is executing the P operator while preparing for the next action.
  It is also the case that the "M" drops out because the P and BB belong together in a chunk.

  The button press BB is fully anticipated as the cursor is being moved to the target.

Heuristics for Mental Operations cont.

- Rule 2: If a string of MK's or MB's belongs to a cognitive unit(e.g., the name of a command), then delete all M's but the first. For instance, an emacs command like C-x 4 C-f (find-file-other-window) would be MKKK, rather than MKMKMK, because the command name, from the view of the user, is one sequence of three keystrokes.
- Rule 3: If a K is a redundant terminator (e.g., the terminator of a command immediately following the terminator of its argument), then delete the M in front of it.
- Rule 4: If a K terminates a constant string (e.g., a command name), then delete the M in front of it; but if the K terminates a variable string (e.g., an argument string), then keep the M in front of it.

KLM Example

- Burning CDs with ECLiPt Roaster vs. GNOME Toaster
- see again `http://www.rpi.edu/~glasse/HCI-HOWTO/ar01s03.html`

Assumptions

- We are starting from the startup screen of each program.
- The user's hands are on the mouse.
- A fresh CD is in the system, so the user doesn't need to insert one.
- We will use the Kieras operators.

Action Sequence for ECLiPt Roaster

- Click Add.
- Click slider down twice. Double click writing.
- Click journal.
- Click OK (adds journal to the list of things to burn). Click Close (closes the file dialog).
- Click Burn CD. A dialog pops up, asking if, since you didn't select any files, you would like to burn all of them. Click Yes.
- Another window opens up asking some more information about the disc (write speed, CD-R or CD-RW media, whether to fixate after burning). Click Burn.

Final Operator Sequence

1. M	selecting command	1200 msec
P	point to Add	1100 msec
BB	click	200 msec
P	point to slider	1100 msec
BB	click	200 msec
BB	click	200 msec
P	move to writing	1100 msec
BBBB	double-click	400 msec
P	move to journal	1100 msec
BB	click	200 msec
P	move to OK	1100 msec
BB	click	200 msec
5. M	selecting command	1200 msec
P	move to Close	1100 msec
BB	click	200 msec

Final Operator Sequence cont.

7. M	selecting command	1200 msec
P	move to Burn CD	1100 msec
BB	click	200 msec
P	move to Yes	1100 msec
BB	click	200 msec
11. M	selecting command	1200 msec
P	move to Burn	1100 msec
BB	click	200 msec

sum: 16.9 seconds

Action Sequence for X-CD-Roast

- Click Create CD.
- Click Master Tracks.
- Click on the slider bar to make it go down twice.
- Click the plus sign next to writing to make the directory open up.
- Click down on the slider one more time.
- Drag journal to the left-hand side of the screen.
- Select “Add to root directory” (of the CDROM).
- Click OK.
- Click Create session/image. This tab has settings on how long the CD-R is, whether to eject after write, and so forth.
- Click Master and write on-the-fly.
- A dialog pops up. Click OK

Final Operator Sequence

1. M	selecting command	1200 msec
P	point to Create CD	1100 msec
BB	click	200 msec
3. M	selecting command	1200 msec
P	point to Master tracks	1100 msec
BB	click	200 msec
P	move to slider	1100 msec
BB	click	200 msec
BB	click	200 msec
P	move to "+" next to writing	1100 msec
BB	click	200 msec
P	move to slider	1100 msec
BB	click	200 msec
P	move to journal	1100 msec
B	mouse button down, to begin dragging	100 msec
6. M	beginning a sequence of direct-manipulation operations	1200 msec
P	drag to CD side of the screen	1100 msec
B	mouse button up, to end dragging	100 msec

Final Operator Sequence cont.

8. M	selecting command	1200 msec
P	point to “Add to root directory of CD”	1100 msec
BB	click	200 msec
10. M	before a B that is not part of an argument string	1200 msec
P	point to OK	1100 msec
BB	click	200 msec
12. M	selecting command	1200 msec
P	move to Create session/image	1100 msec
BB	click	200 msec
14. M	selecting command	1200 msec
P	move to Master and write on-the-fly	1100 msec
BB	click	200 msec
P	move to OK	1100 msec
BB	click	200 msec

sum: 24 seconds

KLM – Evaluation

- Esp. suitable for comparing efficiency of use of different systems, strategies
- Other than ACT-models: no generators of behavior but just descriptors; do not address problem solving
- Problem: identifying and coding all methods

Other GOMS Models

- NGOMSL (Kieras): Natural GOMS Language
  - Structured natural language to represent user's methods and selection rules
  - allows prediction of learning and execution times
- CPM-GOMS (Gray, John, Atwood): mapping of sequential dependencies between users' perceptual, cognitive and motor processes in an execution card
- Further reading: Choosing and getting started with a cognitive architecture to test and use human-machine interfaces, F. Ritter, MMI-Active, 7, June 2004