Lehrstuhl für Medieninformatik
Institut für Information und Medien, Sprache und Kultur
Universität Regensburg

Modelling Interaction

Interaction Techniques and Technologies (ITT), SS 2016

Session 9 (02.06.2016), Raphael Wimmer

Overview

These are slides/notes for the lecture, automatically generated from the slide set. Please extend this outline with your own notes.

Goals for this Week

Overall: Analyzing tasks and predicting task completion times

Know

- Power Law of Practice
- Hick's Law
- predictive models: GOMS, KLM

Learn

- implementing simple GUIs with PyQt
- predicting task completion times using predictive models

Practice

- instrumenting code in Python: `print()`, QT signals, decorators
- conducting small user studies
- overview of related work

Today

- **14:15 - 14:30** Overview of today's session
- **14:30 - 15:15** Review/overview of predictive models, GOMS and KLM
- **15:15 - 15:35** Practical experiment
- **15:35 - 15:45** New Assignment, preview of next session

Where are We?

- **Conducting and Logging Experiments** (+ intro to Python / PyQT)
- **Documenting and Visualizing Experiments** (+ intro to pylab, matplotlib)
- **Pointing** (pointing devices, Fitts' Law, Steering Law, CD gain, …)
- **Text Entry** (speed, models, keyboard layouts, input techniques)
- **Models of Interaction** (KLM, GOMS)

Predictive Models

General Question

How can we predict how much time a user will need to complete a certain task?

Fitts' Law

- Paul Fitts (1954): "The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement", J. Exp. Psy. 47
- Classic Formulation
  $ID = log_2(2A/W) bit/s$
  $I_P = 1/t \times log_2(2A/W) bit/s$
- (A = amplitude/distance of movement; W = width of target; ID = index of difficulty, I_P_ = index of performance)
- Shannon Formulation (MacKenzie, 1989)
  $MT = a + b \times log_2(A/W + 1)$
- interesting discussion: Drewes (2013): "A Lecture on Fitts' Law"[1]

Steering Law

- Acot, Zhai (1997): "Beyond Fitts' law: models for trajectory-based HCI tasks"[2], Proc. CHI '97, ACM, New York
- Idea: steering a pointer through a narrow tunnel takes longer than through a wide tunnel
- How does movement time depend on width and length of tunnel?
- Model tunnel as a series of Fitts' Law pointing tasks
- -> $T = a + b \int_C \frac{ds}{W(s)}$
- for a straight tunnel with width W, length A: $T = a + b\frac{A}{W}$

Power Law of Practice

- The logarithm of the reaction time decreases linearly with the logarithm of the amount of practice
- Newell & Rosenbloom (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Ed.), Cognitive skills and their acquisition (pp. 1-55). Hillsdale, NJ: Erlbaum.

Hick's Law

- Amount of time needed to choose one item in a set is proportional to $log_2(n)$, whereas n is the number of items in the set.
- Hick (1952), "On the rate of gain of information"[3]

Miller's Law

- Humans can keep 7 ± 2 chunks of information in their working memory.
- Miller (1956), The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information[4]
- *chunks*: numbers, letters, words, sentences, images, ...
- *recoding* of multiple chunks into fewer chunks allows for storing more information
- e.g., " T T S E" == 4 chunks, "TEST" == 1 chunk
- actual amount that can be stored depends on type of information

[1] http://www.cip.ifi.lmu.de/~drewes/science/fitts/A%20Lecture%20on%20Fitts%20Law.pdf
[2] http://www.almaden.ibm.com/u/zhai/papers/steering/chi97.pdf
[3] http://www2.psychology.uiowa.edu/faculty/mordkoff/InfoProc/pdfs/Hick%201952.pdf
[4] http://psychclassics.yorku.ca/Miller/

And now?

- aforementioned models only describe sensorimotor control
- human-computer interaction is a little bit more complex

GOMS

GOMS

- Card, Moran, Newell (1983): "The Psychology of Human-Computer Interaction", Lawrence Erlbaum Associates, Hillsdale, NJ
- descriptive / predictive model of task completion
- Four components:
  - **G**oals - what the user wants to achieve (hierarchy of goals)
  - **O**perators - individual steps towards the *goal*
  - **M**ethods - sequence of related *operators*
  - **S**election Rules - how the user selects which of several alternative *methods* they use
- Distinction of goals and operators defined by intended level of detail

GOMS: Example

- Goal: *post a comment on YouTube*
  - Goal: *log in*
    - Operator: *click on 'sign in' link*
    - Operator: *enter name*
    - Operator: *enter password*
    - **or**: form already filled in by password manager
    - Operator: *click 'sign in' button*
  - Goal: *select comment form*
  - Goal: *type something witty*
  - Goal: *submit comment*

. . .

Let's have a closer look at:

. . .

Goal: *Type and format the text 'so **COOL**' using a rich-text editor*

. . .

Which operators and methods are possible? (let's do this together)

GOMS: Use Cases

- document and discuss tasks
- identify usability problems:
  - performance bottlenecks
  - methods with high cognitive demands
  - too many alternative paths, etc.
- predict task completion times (by assigning times to each operator)

- example: Gray, John, Atwood (1992): "The precis of Project Ernestine or an overview of a validation of GOMS"[5], Proc. CHI '92, ACM, New York

GOMS: Limitations

- GOMS (in its classic form) does not account for:
  - errors
  - variations in users
    - skill
    - age
    - fatigue
    - training
  - likeability, etc.
- complex tasks result in complex GOMS trees
- How coarse/fine should operators be defined?
- What are valid selection rules? (need to find out experimentally)

KLM

KLM - Keystroke Level Model

- Card, Moran, Newell (1980): "The keystroke-level model for user performance with interactive systems"[6], Communications of the ACM 23(7), ACM, New York
- 'low-level GOMS'
- idea: deconstruct tasks down to the keystroke-level operators:
  - **K**eystroke
  - **P**ointing with mouse
  - **B**utton press or release on mouse
  - **M**ental act
  - **H**and switching between keyboard and mouse
  - **W**aiting for system response
- Practical guide: Kieras (2001): "Using the keystroke-level model to estimate execution times"[7]

KLM: Standard Values

based on (Card, Moran, Newell (1980) and Kieras (2001))

Operator	time (s)
Keystroke	0.28
Pointing with mouse	1.10
Button press or release on mouse	0.10
Mental act	1.20
Hand switching between keyboard and mouse	0.40
Waiting for system response	–

[5] http://dl.acm.org/citation.cfm?id=142821
[6] http://www.cs.cmu.edu/~cga/behavior/card1980.pdf
[7] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.7.3363&rep=rep1&type=pdf

. . .

Example: *How long will it take for you to log into GRIPS and click the first link?*

KLM: Extensions

- for mobile phones: Holleis et al. (2007)
- automatically generating KLM predictions: Schulz (2008)
- powerful tool: CogTool[8]

Exercise: QWERTY vs. Chords

- How long will it take to write "I am sitting here in a boring room."?
- typing every letter
- replacing the two most common words with a chord
- replacing the two longest words with a chord

Outlook

Course Assignment

- Read up on KLM, GOMS
- Implement a simple calculator application with PyQt (only a few basic operators)
- Determine appropriate values for the KLM operators for the calculator application
- Predict task completion times using KLM and Fitts' Law and verify your predictions

Next Session

- Knowledge test
- Discussion of previous sessions
- Introduction: Bonus Tasks
- Assigning new groups
- Distribution WiiMotes

Afterwards

- Introduction WiiMote
- Introduction to PyQtGraph

ENDE

[8] http://www.cogtool.com