Fall 2019
CS6501: Topics in Human-Computer Interaction
http://seongkookheo.com/cs6501_fall2019

Lecture 11:
Modeling Interaction 2

Seongkook Heo

Oct 3, 2019

UNIVERSITY *of* VIRGINIA | SCHOOL *of* ENGINEERING & APPLIED SCIENCE

Keystroke-Level Model (KLM)

- Developed for predicting task performance using interactive computing systems

- Predicts **expert error-free** task completion times

- Elements of a KLM prediction
  - Task (or a series of tasks)
  - Method used
  - Command language of the system
  - Motor skill parameters of the user
  - Response time parameters of the system

Why Use the KLM?

- Consider a task such as "delete a file"

- Perhaps there are two ways to do the task:

    1. Mouse + menu selection
    2. Keyboard + command entry

- The KLM can predict the time for each method

- If used at the design stage, design alternatives may be considered and compared → design choices follow

KLM Operators

- Six operators

  - Keystroke
    - Average time determined by typing tests
  - Pointing
    - Varies from 0.8 – 1.5 seconds (Fitts' Law)
  - Homing
    - 0.4 seconds based on various studies
  - Drawing
    - Roughly defined as 0.9n + 0.16l
  - Mental
    - 1.35 seconds, experimentally determined
  - Response (system response)
    - Must be input to the model, varies widely

$$t_{EXECUTE} = t_K + t_P + t_H + t_D + t_M + t_R$$

Operator	Description	Time (s)
K	PRESS A KEY OR BUTTON Pressing a modifier key (e.g., shift) counts as a separate operation, Time varies with typing skill: 　Best typist (135 wpm) 　Good typist (90 wpm) 　Average skilled typist (55 wpm) 　Average non-secretary typist (40 wpm) 　Typing random letters 　Typing complex codes 　Worst typist (unfamiliar with keyboard)	 0.08 0.12 0.20 0.28 0.50 0.75 1.20
P	POINT WITH A MOUSE Empirical value based on Fitts' law. Range from 0.8 to 1.5 seconds. Operator does *not* include the button click at the end of a pointing operation	1.10
H	HOME HAND(S) ON KEYBOARD OR OTHER DEVICE	0.40
$D(n_D, l_D)$	DRAW n_D STRAIGHT-LINE SEGMENTS OF TOTAL LENGTH l_D. Drawing with the mouse constrained to a grid.	$.9\, n_D + .16\, l_D$
M	MENTALLY PREPARE	1.35
$R(t)$	RESPONSE BY SYSTEM Different commands require different response times. Counted only if the user must wait.	t

KLM Example

Keystroke
Pointing
Homing
Drawing
Mental
Response

- Replace 5 letter word with another in a text editor

Reach for mouse	H_{mouse}
Point to word	P_{word}
Select word	K_{select}
Home on keyboard	$H_{keyboard}$
Call replace cmd	$K_{replace}$
Type new 5 letter word	$5K_{word}$

Great poem for HCI - Notepad

File   Edit   Format   View   Help

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque et ipsum varius, semper ligula sit amet, scelerisque lorem. Cras luctus at neque at consequat. Suspendisse lacinia nisl in tempor accumsan. Suspendisse facilisis fringilla scelerisque. Pellentesque id vulputate diam. Vivamus lectus libero, maximus et luctus at, mattis ac neque. Pellentesque facilisis efficitur dui. Vestibulum eu tellus vulputate, malesuada tellus nec, pulvinar ipsum. Proin ullamcorper sodales quam at accumsan. Cras iaculis, dui ut egestas malesuada, nisi enim consectetur metus, ac euismod lacus mauris a ante. Duis bibendum interdum est at venenatis. Aliquam et tellus tincidunt, congue nunc nec, luctus magna. Sed vitae accumsan magna. In hac habitasse platea dictumst.

Ln 1, Col 146   100%   Windows (CRLF)   UTF-8

$$T_{execute} = H_{mouse} + P_{word} + K_{select} + H_{keyboard} + K_{replace} + 5K_{word}$$

Original KLM Experiment

Keystroke
Pointing
Homing
Drawing
Mental
Response

- The KLM was validated in an experiment with fourteen tasks performed using various methods and systems

Command Line Editor (POET)

Jump to next line	**M K**[LINEFEED]
Call Substitute command	**M K**[S]
Specify new 5-digit word	**5K**[word]
Terminate argument	**M K**[RETURN]
Specify old 5-digit word	**5K**[word]
Terminate argument	**M K**[RETURN]
Terminate command	**K**[RETURN]

$$\textbf{T}_{\textit{execute}} = \textbf{4} \times \textbf{t}_M + \textbf{15} \times \textbf{t}_K = \textbf{8.85 sec.}$$

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Erlbaum.

Original KLM Experiment

- The KLM was validated in an experiment with fourteen tasks performed using various methods and systems

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Erlbaum.
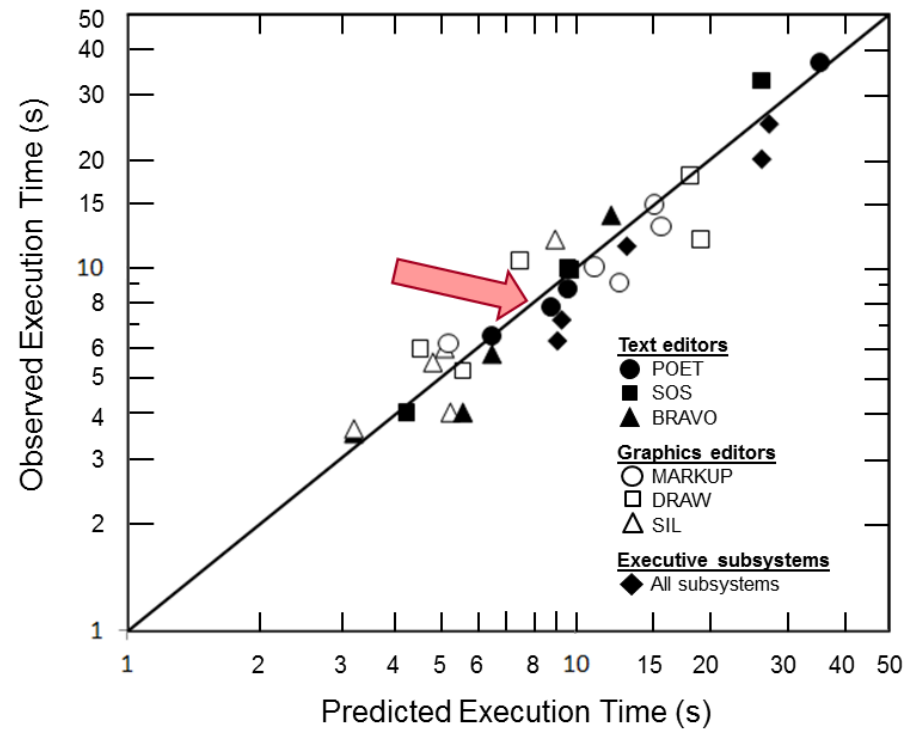
Sensitivity Analysis

- If parameters are treated as variables, the sensitivity of predictions to changes in parameters can be assessed
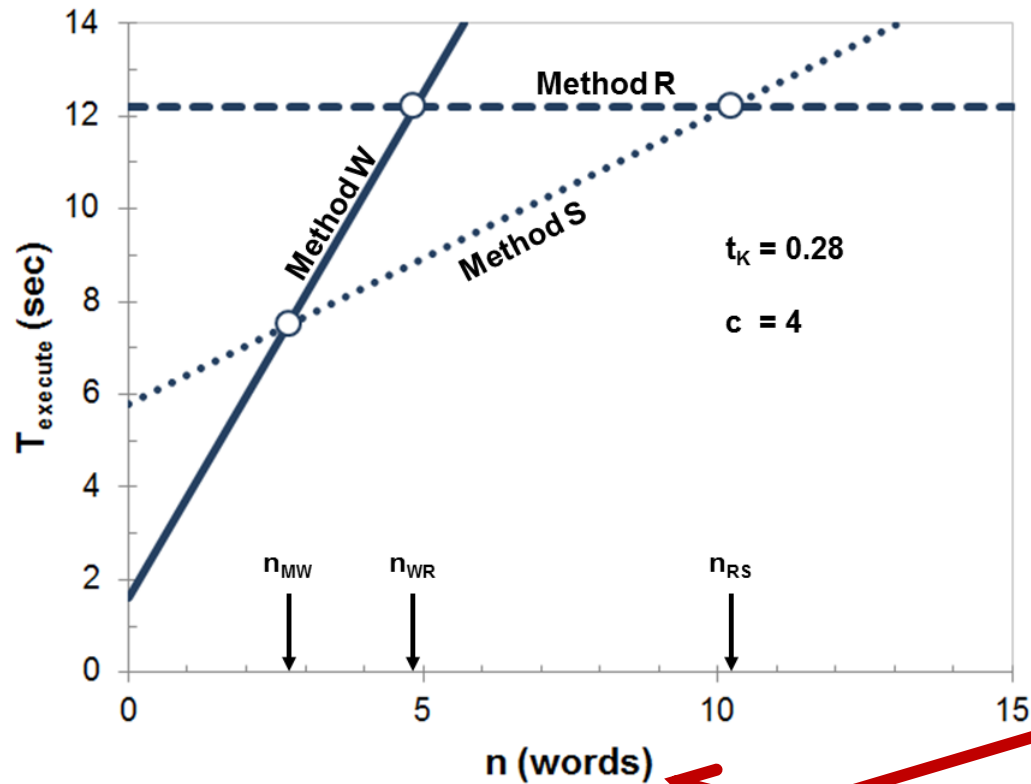
Implication: The preferred method changes with the distance to the misspelled word.

n is a parameter – the distance in words to the location of a misspelled word to correct.

Pointing Operator – Example

- Develop KLM mouse and keyboard predictions for the GUI screen below

- Task: Change the font and style for "M K" to bold, Arial
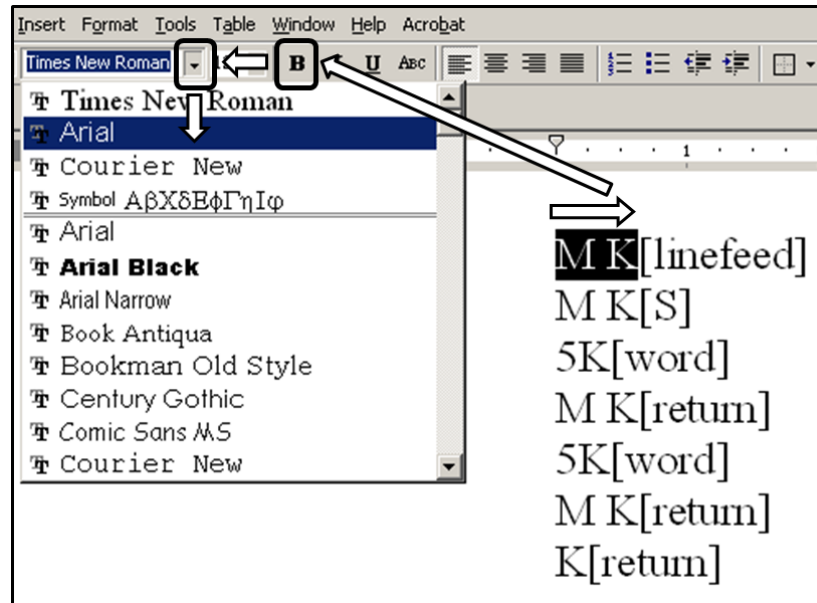
Keystroke
Pointing
Homing
Drawing
Mental
Response

Mouse Analysis

- Operations:

Mouse Subtasks	KLM Operators	t_P (s)
Drag across text to select "M K"	**M P**[2.5, 0.5]	0.686
Move pointer to Bold button and click	**M P**[13, 1]	0.936
Move pointer to Font drop-down button and click	**M P**[3.3, 1]	0.588
Move pointer down list to Arial and click	**M P**[2.2, 1]	0.501
	$\sum t_P =$	2.71

- Prediction:

$$t_{\text{EXECUTE}} = 4 \times t_{\text{M}} + \sum t_{\text{P}} = 4 \times 1.35 + 2.71 = 8.11 \text{ seconds}$$

$$t_{\text{P}} = 0.159 + 0.204 \times \log_2\left(\frac{A}{W} + 1\right)$$

Keystroke
Pointing
Homing
Drawing
Mental
Response

Keyboard Analysis

- Operations:

Keyboard Subtasks	KLM Operators
Select text	**M K**[shift] **3K**[→]
Convert to boldface	**M K**[ctrl] **K**[b]
Activate Format menu and enter Font sub-menu	**M K**[alt] **K**[o] **K**[f]
Type *a* ("Arial" appears at top of list)	**M K**[a]
Select "Arial"	**K**[↓] **K**[enter]

- Prediction:

$$t_{\text{EXECUTE}} = 4 \times t_{\text{M}} + 12 \times t_{\text{K}} = 4 \times 1.35 + 12 \times 0.75 = 14.40 \text{ seconds}$$

Use "typing complex codes" ($t_{\text{K}} = 0.75$ s)

Keystroke
Pointing
Homing
Drawing
Mental
Response

Sensitivity Analysis

- The keyboard prediction is sensitive to the parameter t_K, the keystroking time

- If t_K is allowed to vary, what is the effect on the predictions?
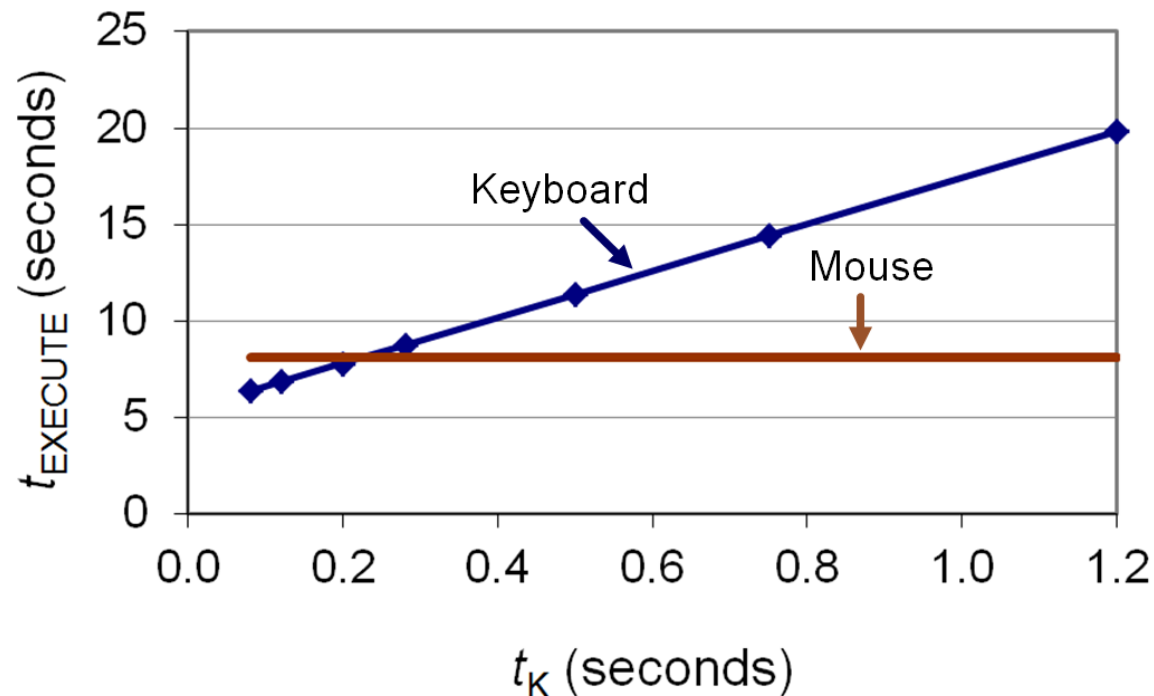
Implication: The mouse is faster than the keyboard, except for $t_K \leq 0.2$ seconds (which is unlikely, given the nature of the keyboard actions).

Updating the KLM's Mental Operator

Proposed Mnemonic	Task	Execution Time (ms)	
		Card et al.	Figure 2-28 & Figure 2-30
M_S	Simple Reaction	240 [105 – 470]	277 [±44]
M_P	Physical Matching	310 [130 – 640]	510 [±59]
M_N	Name Matching	380 [155 – 810]	485 [±52]
M_L	Class Matching	450 [180 – 980]	566 [±96]
M_C	Choice Reaction	$200 + 150 \log_2(N + 1)$	
M_V	Visual Search		$498 + 41\,N$
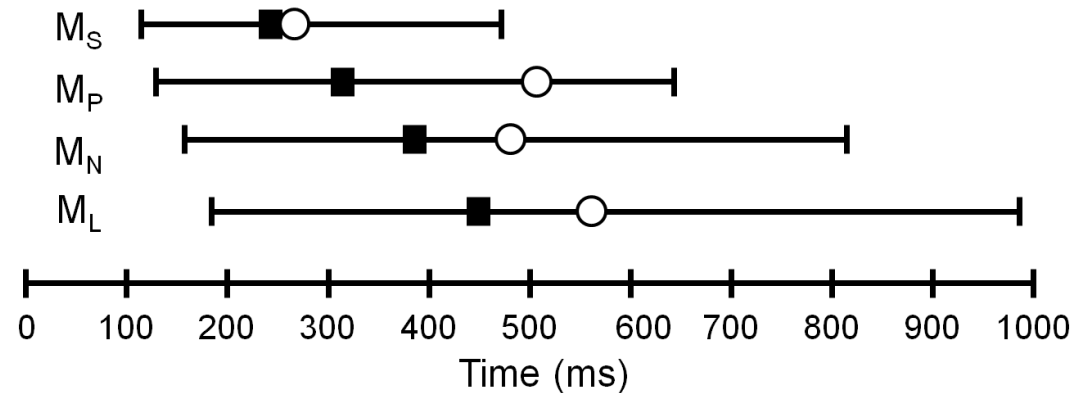
Contemporary Uses of the KLM

- The KLM continues to be widely used in HCI

- Examples:
  - Attention shifts with mobile phones
  - Stylus-based circling gestures
  - Managing folders and messages in e-mail applications
  - Predictive text entry on mobile phones
  - Task switching in multi-monitor systems
  - Mode switching on tablet PCs
  - Distractions in in-vehicle information systems (IVIS)

RIMES: Embedding Interactive Multimedia Exercises in Lecture Videos
Juho Kim, Elena L. Glassman, Andrés Monroy-Hernández, Meredith Ringel Morris, CHI 2015.

Neilson's Heuristics (updated)

- H1: Visibility of system status
- H2: Match between system and the real world
- H3: User control and freedom
- H4: Consistency and standards
- H5: Error prevention
- H6: Recognition rather than recall
- H7: Flexibility and efficiency of use
- H8: Aesthetic and minimalist design
- H9: Help users recognize, diagnose, and recover from errors
- H10: Help and documentation

https://www.nngroup.com/articles/ten-usability-heuristics/

H1: Visibility of System Status

- Keep users informed about what is going on

- Example: pay attention to response time
  - 0.1 sec: no special indicators needed
  - 1.0 sec: user tends to lose feeling of direct operation
  - 10 sec: max. duration if user to stay focused on action
  - for longer delays, use percent-done progress bars

Time Left:	00:00:19	searching database for matches

46%
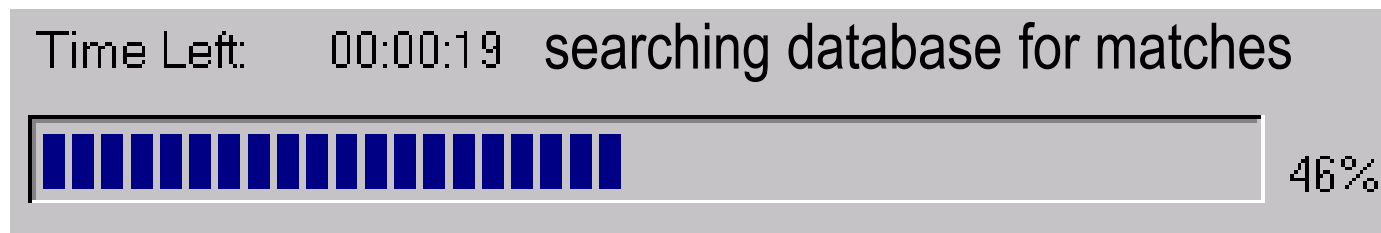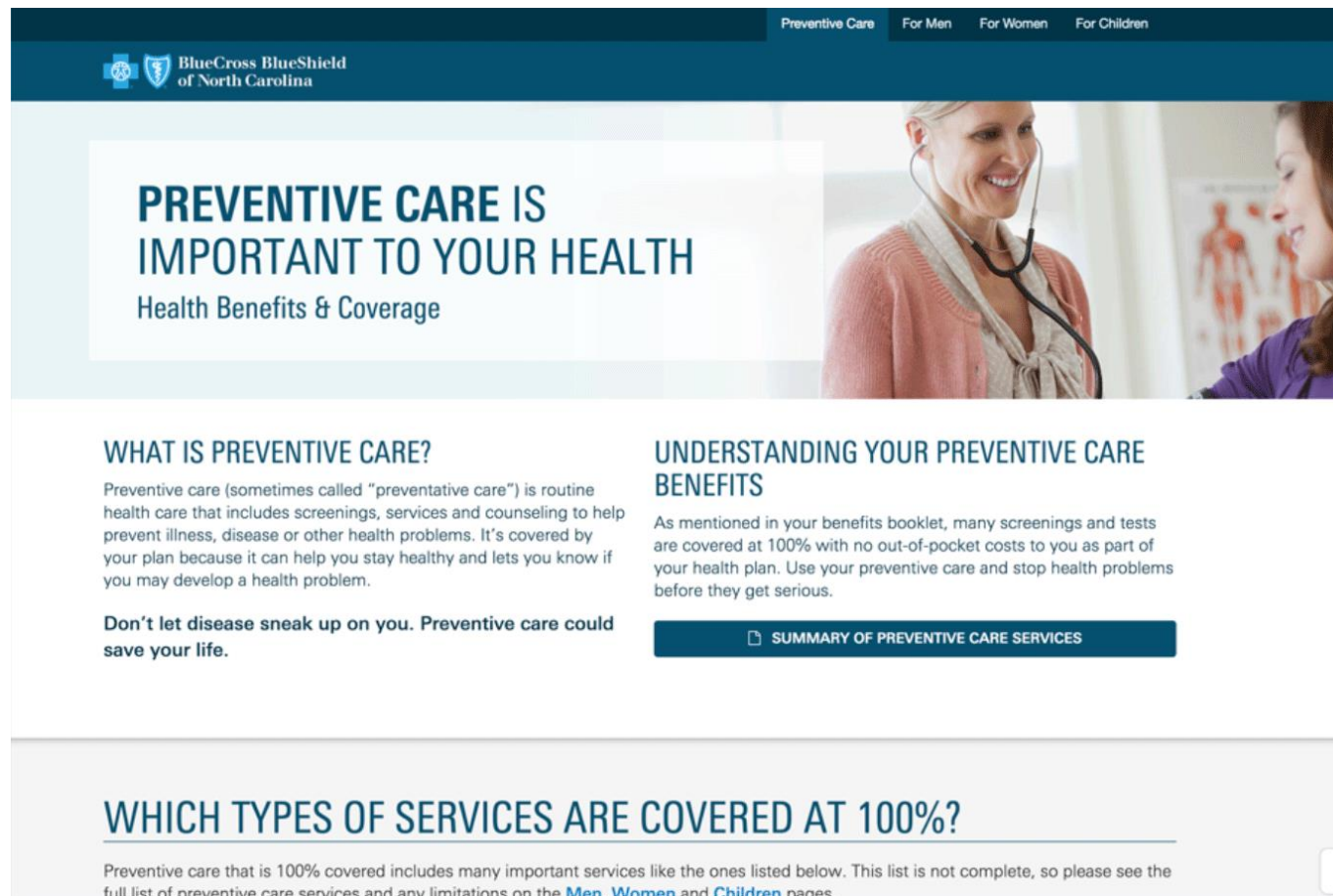
H1: Visibility of System Status

- Keep users informed about what is going on
  - Appropriate visible feedback

What mode am I in now?

What did I select?

How is the system interpreting my actions?

H2: Match Between System & Real World

- Speak the users' language

H2: Match Between System & Real World

- Speak the users' language

- Follow real world conventions
  - Old example: Mac desktop
  - Dragging disk to trash
  - Should delete it, not *eject* it!

H2: Match Between System & Real World

- Speak the users' language

- Follow real world conventions

- Metaphors can work well
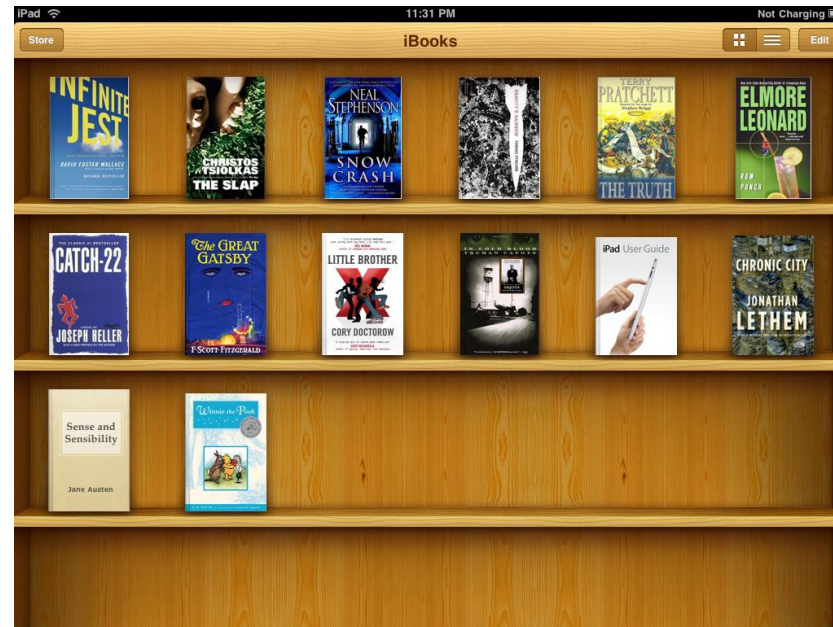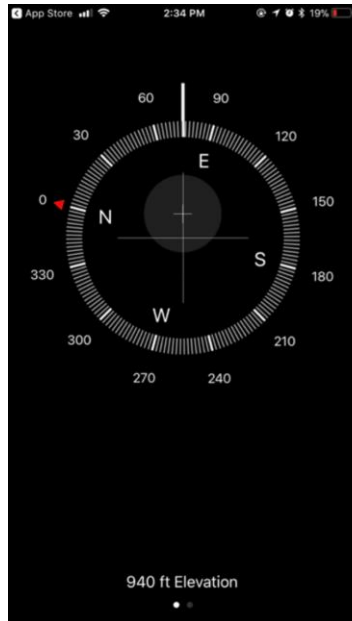
H2: Match Between System & Real World

• Speak the users' language

• Follow real world conventions

• Metaphors can work well
  • But be careful…
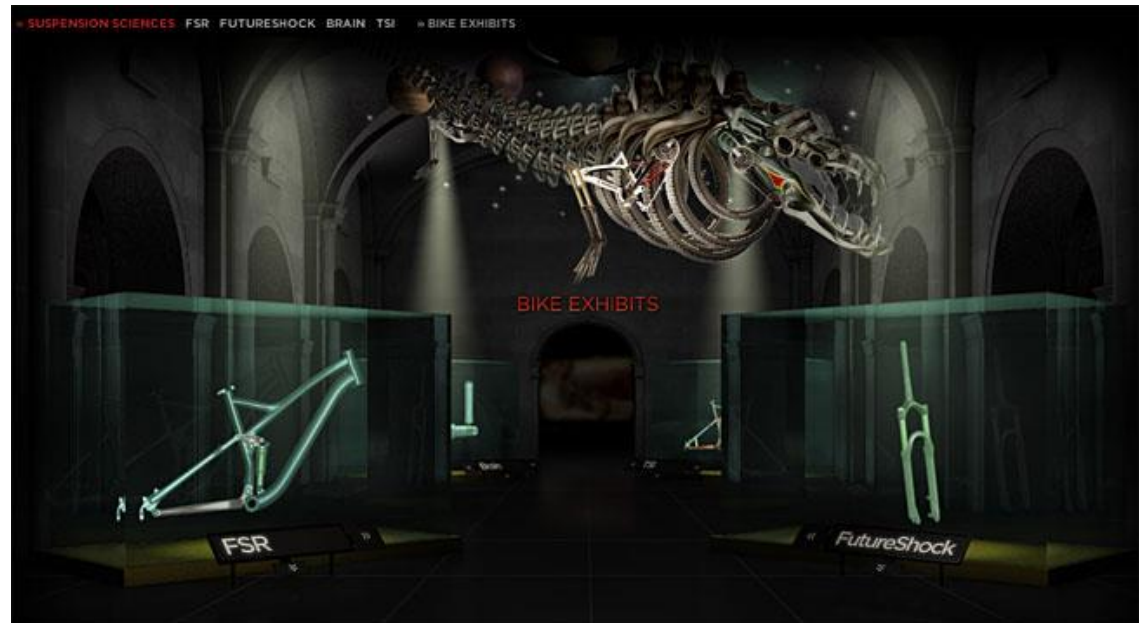
H3: User Control & Freedom

- "Exits" for mistaken choices, undo, redo

- Don't force down fixed paths

H3: User Control & Freedom

- "Exits" for mistaken choices, undo, redo

- Don't force down fixed paths

- Strategies:
  - Cancel button (for dialogs waiting for user input)
  - Universal Undo (can get back to previous state)
  - Interrupt (especially for lengthy operations)
  - Quit (for leaving the program at any time)
  - Defaults (for restoring a property sheet)

H4: Consistency & Standards

- Consistency of effects
  - Same words, commands, actions will always have the same effect

- Consistency of language and graphics
  - Same info/controls in same location on all screens/dialog boxes
  - Forms follow boiler plate
  - Same visual appearance across the system (e.g. widgets)

- Consistency of input
  - Consistent syntax across complete system
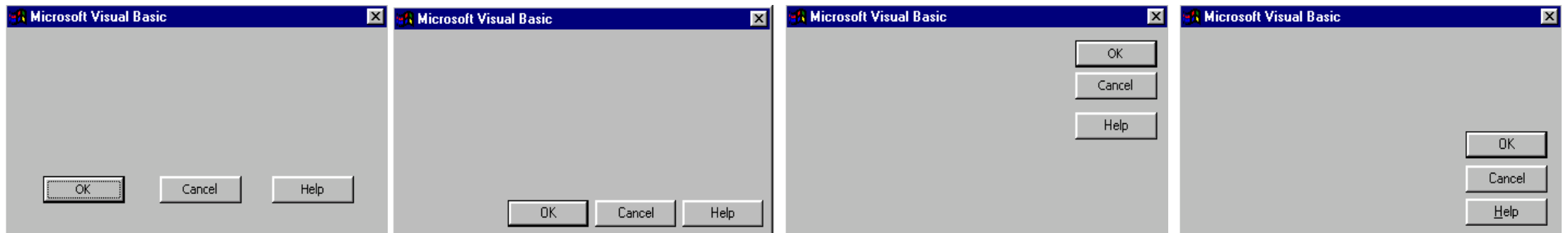
H4: Consistency & Standards

These are labels with a raised appearance.

Is it any surprise that people try and click on them?

H5: Error Prevention

- Try to make errors impossible
  - Modern widgets: only "legal commands" selected, "legal data" entered

H5: Error Prevention

- Try to make errors impossible
  - Modern widgets: only "legal commands" selected, "legal data" entered
  - Automatically correct/highlight potential errors

H5: Error Prevention

- Try to make errors impossible

  - Modern widgets: only "legal commands" selected, "legal data" entered
  - Automatically correct/highlight potential errors
  - Provide reasonableness checks on input data
    - 5000 pencils is an unusually large order. Do you really want to order that many?

H6: Recognition Rather than Recall

- Computers good at remembering things, people aren't!

- Promote recognition over recall
  - Menus, icons, choice dialog boxes vs command lines, field formats
  - Relies on visibility of objects to the user (but less is more!)
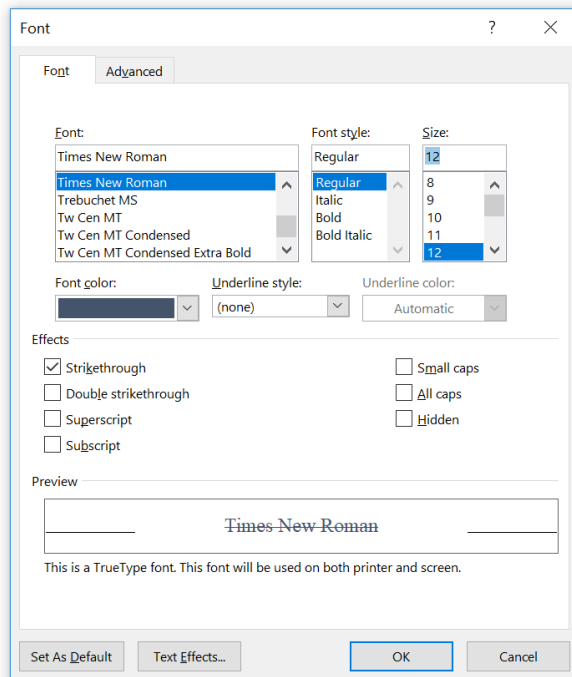
H7: Flexibility and Efficiency of Use

- Experienced users should be able to perform frequently used operations quickly

- Strategies:
    - Keyboard and Mouse Accelerators
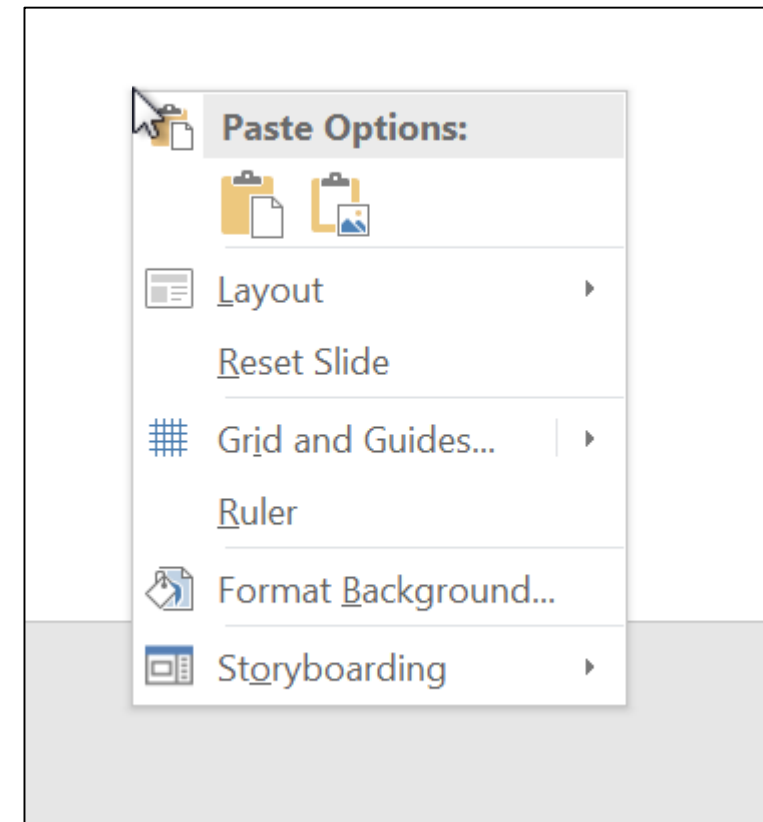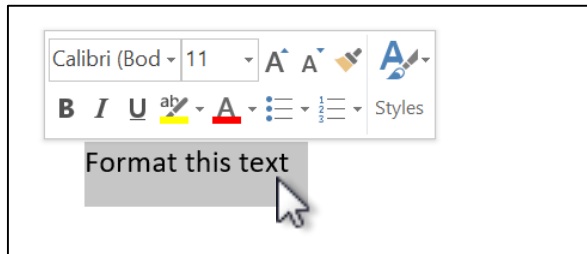
H7: Flexibility and Efficiency of Use

- Experienced users should be able to perform frequently used operations quickly

- Strategies:

  - Keyboard and Mouse Accelerators
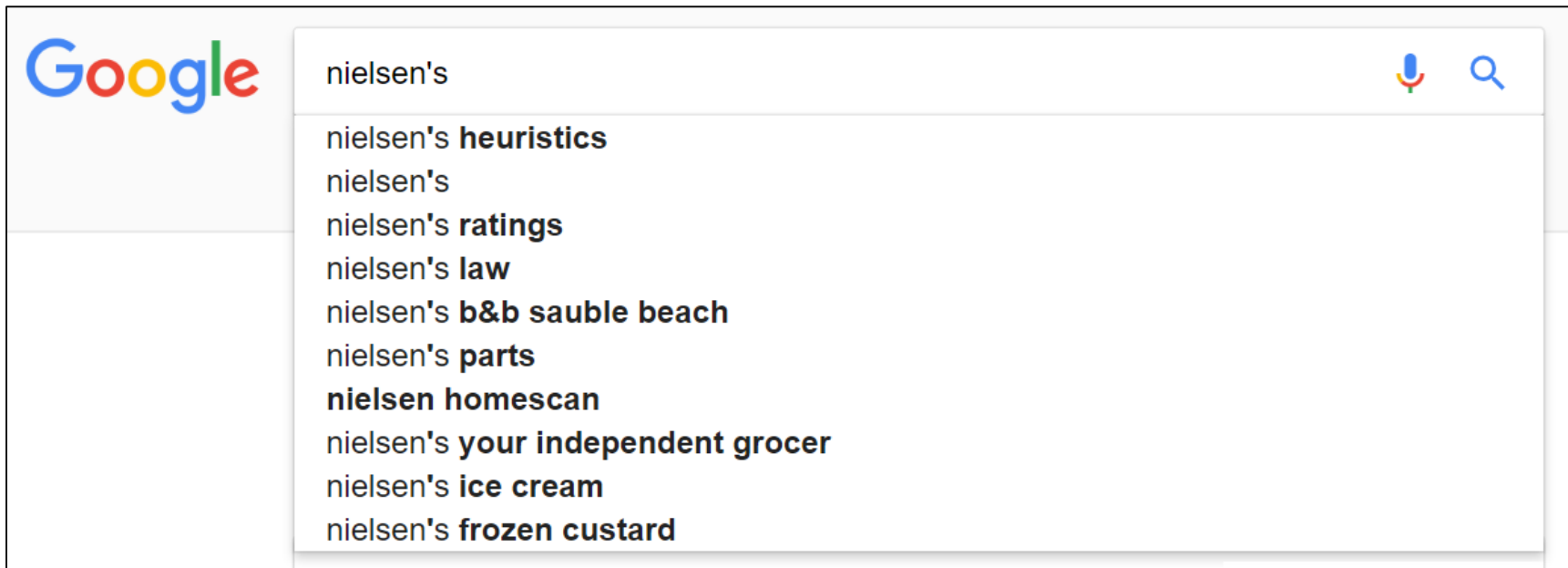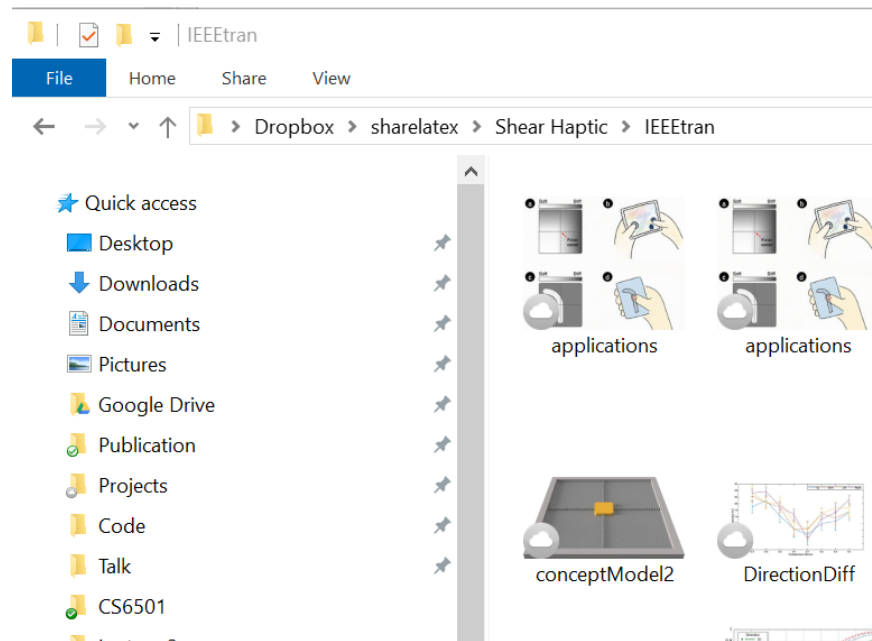 - Auto-Complete (entering input before the system is ready for it)

H7: Flexibility and Efficiency of Use

- Experienced users should be able to perform frequently used operations quickly

- Strategies:

  - Keyboard and Mouse Accelerators
  - Auto-Complete (entering input before the system is ready for it)
  - Navigation Jumps
  - History Systems

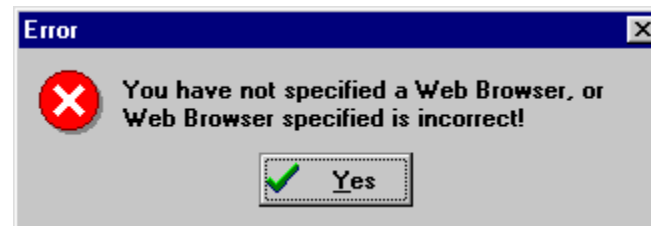# H8: Aesthetic and Minimalist Design

- No irrelevant information in dialogues

- Bad example:

H9: Help Users Recognize and Recover from Errors

- Error messages in plain language

- Precisely indicate the problem

- Constructively suggest a solution

H9: Help Users Recognize and Recover from Errors

- Error messages in plain language

- Precisely indicate the problem

- Constructively suggest a solution

H9: Help Users Recognize and Recover from Errors

- Error messages in plain language

- Precisely indicate the problem

- Constructively suggest a solution

H9: Help Users Recognize and Recover from Errors

- Error messages in plain language

- Precisely indicate the problem

- Constructively suggest a solution

H9: Help Users Recognize and Recover from Errors

- Error messages in plain language

- Precisely indicate the problem

- Constructively suggest a solution

i The page cannot be found

The page you are looking for might have been removed, had its name changed, or is temporarily unavailable.

Please try the following:

- If you typed the page address in the Address bar, make sure that it is spelled correctly.
- Open the httpd.apache.org home page, and then look for links to the information you want.
- Click the ⇐ Back button to try another link.
- Click 🔍 Search to look for information on the Internet.

HTTP 404 - File not found
Internet Explorer

H10: Help and Documentation

- Help is not a replacement for bad design!

- Simple systems:

  - Walk up and use; minimal instructions

- Most other systems:

  - Feature rich
  - Some users will want to become "experts" rather than "casual" users
  - Intermediate users need reminding, plus a learning path

H10: Help and Documentation

- Many users do not read manuals

  - Prefer to spend their time pursuing their task

- Usually used when users need immediate help

  - Indicates need for online documentation, good search/lookup tools
  - Online help can be specific to current context
  - Paper manuals unavailable in many businesses!

- Sometimes used for quick reference

  - Syntax of actions, possibilities…
  - List of shortcuts …

H10: Help and Documentation
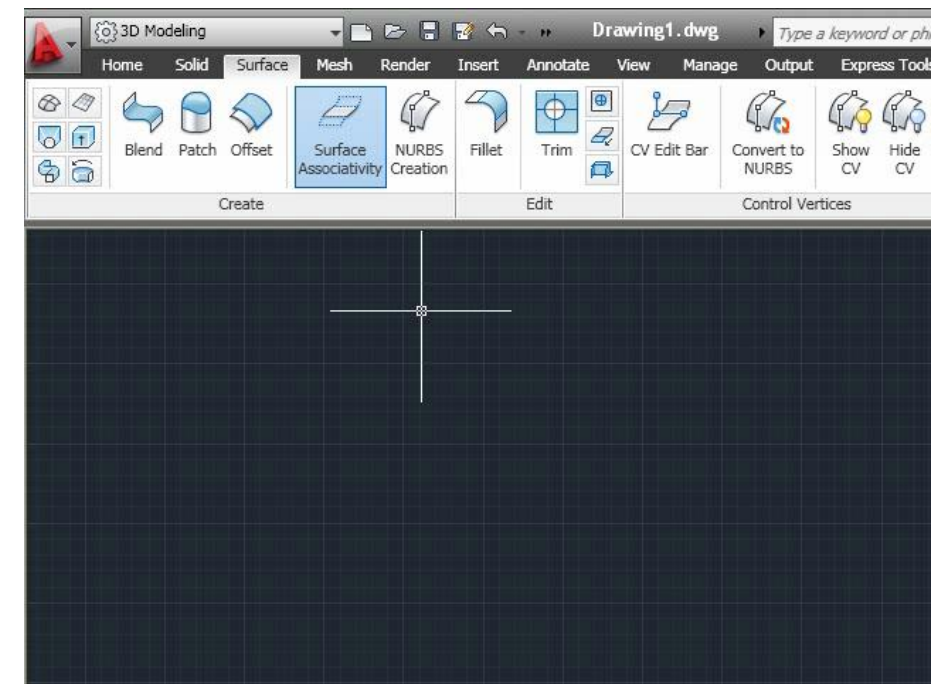
Heuristic Evaluation Process

- Multiple evaluators go through UI
  - Should do at least two passes
    - First to get feel for flow and scope of system
    - Second to focus on specific elements
  - Inspect various dialogue elements
  - Compare with list of usability principles
  - Consider other principles/results that come to mind

- Evaluator Expertise
  - Better evaluators will provide better results
  - If system is walk-up-and use, or evaluator is domain expert, no assistance needed
  - Otherwise can provide assistance or supply evaluators with scenarios

Heuristic Evaluation Process

- Each evaluator produces list of problems

  - Explain why with reference to heuristic or other information
  - Be specific and list each problem separately

- Debriefing session among all evaluators

  - Findings are aggregated
  - Use violations to redesign/fix problems
  - No formula to fix the problems

Heuristic Evaluation Process

- Where problems may be found

  - Single location in UI
  - Two or more locations that need to be compared
  - Problem with overall structure of UI
  - Something that is missing

Severity Ratings

- Used to allocate resources to fix problems

- Estimates of need for more usability efforts

- Combination of
  - Frequency
  - Persistence (one time or repeating)
  - User impact
  - Market impact

  - 0 - Don't agree that this is a usability problem
  - 1 - Cosmetic problem
  - 2 - Minor usability problem
  - 3 - Major usability problem; important to fix
  - 4 - Usability catastrophe; imperative to fix

Heuristic Evaluation Summary

- Advantages:
    - The "minimalist" approach
        - A few general guidelines can correct for the majority of usability problems
        - Easily remembered, easily applied with modest effort
    - Discount usability engineering
        - Cheap and fast way to inspect a system
        - Can be done by usability experts, double experts, and end users

- Problems:
    - The are guidelines only
    - Subtleties involved in their use
    - Doesn't provide solutions
    - Doesn't observe actual use with real users

Acknowledgements

- Some of the materials are based on materials by

  - Tovi Grossman, Univ. of Toronto
  - Juho Kim, KAIST
  - Jonathan Lazar, Jinjuan Heidi Feng, Harry Hochheiser, Research Methods in Human-Computer Interaction, 2nd Ed.

Thank you!