# Review

- What are the main components of the CMN Model?
  - What are the key numbers from the CMN Model?
- What is Fitts' Law?
- Why evaluate?
- Lab vs. field studies?
- Participatory Design?
- Techniques to evaluate without users?
  - Literature review
  - Cognitive walkthrough
  - Heuristic evaluation
  - Model-based evaluation



## Theory

- ✓ Models of interaction
  - ✓ Affordances, mappings, constraints, types of knowledge, errors
- ✓ Design principles

⇒ Human cognition and performance

- History and vision of HCI

## Practice

- ✓ Sketching
- ✓ User observation
- ✓ Iterative design
- ✓ Prototyping
- ✓ Ideation

⇒ User study and evaluation

# Evaluation Techniques

## Evaluating Without Users

E1 Literature Review

E2 Cognitive Walkthrough

E3 Heuristic Evaluation

E4 Model-based Evaluation (GOMS,...)

## Evaluating With Users

### Qualitative

E5 Model Extraction

E6 Silent Observation

E7 Think Aloud

E8 Constructive Interaction

E9 Retrospective Testing

### Quantitative

E10 Controlled Experiments

+ Interviews, questionnaires,...

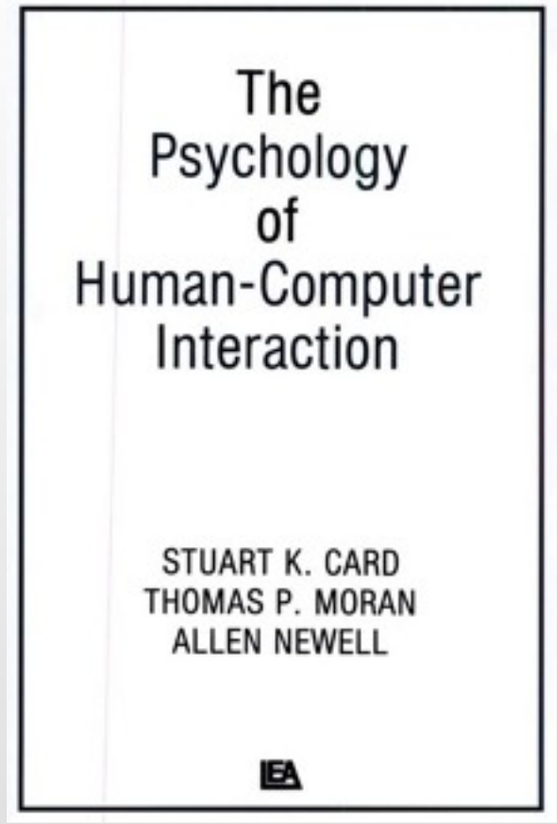
# A Story

- In 1995, now-famous web guru Jakob Nielsen had less than 24 hours to recommend if adding three new buttons to Sun's home page was a good idea.

  Check out his "Alertbox" online column for good (and often fun) web design advice

- He found that each new, but unused button costs visitors .5 million $ per year.
- 2 of the 3 new buttons were taken back out.
- The method he used for his estimate: GOMS.

# GOMS




- Goals, Operators, Methods, Selection rules

- Card, Moran, Newell: The Psychology of HCI, 1983

- To estimate execution and learning times before a system is built

# E4: Model-based Evaluation

- Some models exist that offer a framework for design and evaluation
- Examples:

  ⇒ Information efficiency

  ⇒ GOMS KLM, GOMS

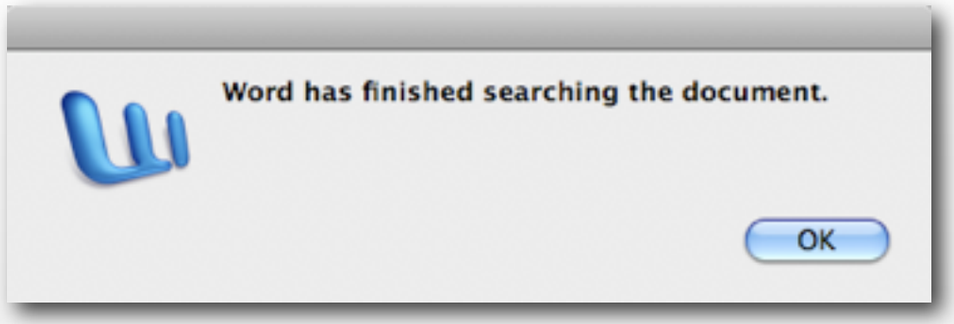
  - Design Rationale (History of design decisions with reasons and alternatives)
  - Design Patterns



Minimal information content → Information-theoretic Efficiency → Interface Efficiency

KLM → Character Efficiency → Interface Efficiency

# Measuring Interface Efficiency




- How fast can you expect an interface to be?
- Information as quantification of amount of data conveyed by a communication (Information theory)
  - E.g., speech, messages sent upon click…
- Lower bound on amount of information required for task is independent of interface design
- Information-theoretic efficiency $E = \frac{\text{Minimal info required for the task}}{\text{Info supplied by user}}$
  - $E \in [0, 1]$ (e.g., $E = 0$ for providing unnecessary information)
- Character efficiency = $\frac{\text{Minimal number of characters required for the task}}{\text{Number of characters entered in the UI}}$

# Information Content (Detailed)

- Information is measured in bits
  - 1 bit represents choice between 2 alternatives
- $n$ equally likely alternatives
  - Total information amount: $\log_2(n)$
  - Information per alternative: $(1/n)\log_2(n)$
- $n$ alternatives with different probabilities $\mathrm{p}(i)$
  - Information per alternative: $\mathrm{p}(i)\log_2(1/\mathrm{p}(i))$
  - Total amount = sum over all alternatives
- Consider situation as a whole
  - Probability of messages required
  - Information measures freedom of choice (information ≠ meaning)

# Example: NRW Area Code

- Four digits
  - First digit: 0
  - Second digit: 2 (70%), 5 (30%)
  - Third, Fourth digits: [0, 9] with equal probability
- E.g., 0241 for Aachen, 0525 for Paderborn
- What is the minimal information content of NRW landline area code?
  - Information per alternative: $p(i)\log_2(1/p(i))$

# Example: NRW Area Code

- Four digits
  - First digit: 0
  - Second digit: 2 (70%), 5 (30%)
  - Third, Fourth digits: [0, 9] with equal probability

| | Probability | Values | $p(i)$ | $p(i)\log2(1/p(i))$ (bits/alternative) | Total bits | |
|---|---|---|---|---|---|---|
| 02XX | 0.7 | 100 | $\frac{0.7}{100} = 0.007$ | $0.007 \times \log_2(1/0.007) = 0.05$ | $100 \times 0.05 = 5$ | $5 + 2 = 7$ |
| 05XX | 0.3 | 100 | $\frac{0.3}{100} = 0.003$ | $0.003 \times \log_2(1/0.003) = 0.02$ | $100 \times 0.02 = 2$ | |

# Example: NRW Area Code

- Minimal information required: 7 bits
- What is the information content of the shown numeric keyboard for 4 digits?

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |
| | 0 | |

| | | | | |
|---|---|---|---|---|
| Alternatives: | [0,9] | [0,9] | [0,9] | [0,9] |
| Counts: | 10 | 10 | 10 | 10 |

Information content = $4 \log_2(10) = 13.29$ bits

- What is the information-theoretic efficiency when you use this keyboard for NRW area code?
  - $E = \frac{\text{Minimal info required for the task}}{\text{Info supplied by user}} = \frac{7}{13.29} = 52.67\%$

# Example: NRW Area Code

- Minimal information required: 7 bits
- What is the information content of the shown numeric keyboard for 3 digits (because the first digit is always zero)?

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |
| | 0 | |

0 _____ _____ _____

| | | | | |
|---|---|---|---|---|
| Alternatives: | | [0,9] | [0,9] | [0,9] |
| Counts: | 0 | 10 | 10 | 10 |

Information content = $4 \log_2(10) = 9.97$ bits

- What is the information-theoretic efficiency when you use this keyboard for NRW area code?
  - $E = \frac{\text{Minimal info required for the task}}{\text{Info supplied by user}} = \frac{7}{9.97} = 70.21\%$

Saved 17.54%!

Minimal information content → Information-theoretic Efficiency → Interface Efficiency

KLM → Character Efficiency → Interface Efficiency



# Keystroke-Level Model

- Execution time for a task = sum of times required to perform the serial elementary gestures of the task

- Typical gesture timings
  - Keying K = 0.2 sec (tap key on keyboard, includes immediate corrections)
  - Pointing P = 1.1 sec (point to a position on display)
  - Homing H = 0.4 sec (move hand from keyboard to mouse or v.v.)
  - Mentally preparing M = 1.35 sec (prepare for next step, routine thinking)
  - Responding R (time a user waits for the system to respond to input)

- Responding time R effects user actions
  - Causality breakdown after 100 ms
  - User will try again after 250 ms ⇒ R
  - Give feedback that input received & recognized

# Keystroke-Level Calculation

- List required gestures
  - E.g., HK = move hand from mouse to keyboard and type a letter
- Compute mental preparation times Ms
  - Difficult: user stops to perform unconscious mental operations
  - Placing of Ms described by rules
- Add gesture timings
  - E.g., HMPK = H + M + P + K = 0.4 + 1.35 + 1.1 + 0.2 = 3.05 sec
- Rule terminology
  - String: sequence of characters
  - Delimiter: character marking beginning (end) of meaningful unit
  - Operators: K, P, and H
  - Argument: information supplied to a command

# Rules for Placing Ms

- Rule 0, initial insertion for candidate Ms
  - Insert Ms in front of all Ks
  - Place Ms in front of Ps that select commands, but not Ps that select arguments for the commands

- Rule 1, deletion of anticipated Ms
  - Delete M between two operators if the second operator is fully anticipated in the previous one

    E.g., PMK ⇒ PK

- Rule 2, deletion of Ms within cognitive units (contiguous sequence of typed characters that form a name)
  - In a string of MKs that form a cognitive unit, delete all Ms except the first

    E.g., "ls⏎" ⇒ MK MK MK ⇒ MK K MK

# Rules for Placing Ms

- Rule 3, deletion of Ms before consecutive terminators
  - If K is redundant delimiter at end of a cognitive unit, delete the M in front of it,

    E.g., "bla⏎⏎" ⇒ M 3K MK MK ⇒ M 3K MK K

- Rule 4, deletion of Ms that are terminators of commands
  - If K is a delimiter that follows a constant string then delete the M in front of it (not for arguments or varying strings)

    E.g., "ls⏎" ⇒ M K K MK ⇒ M K K K

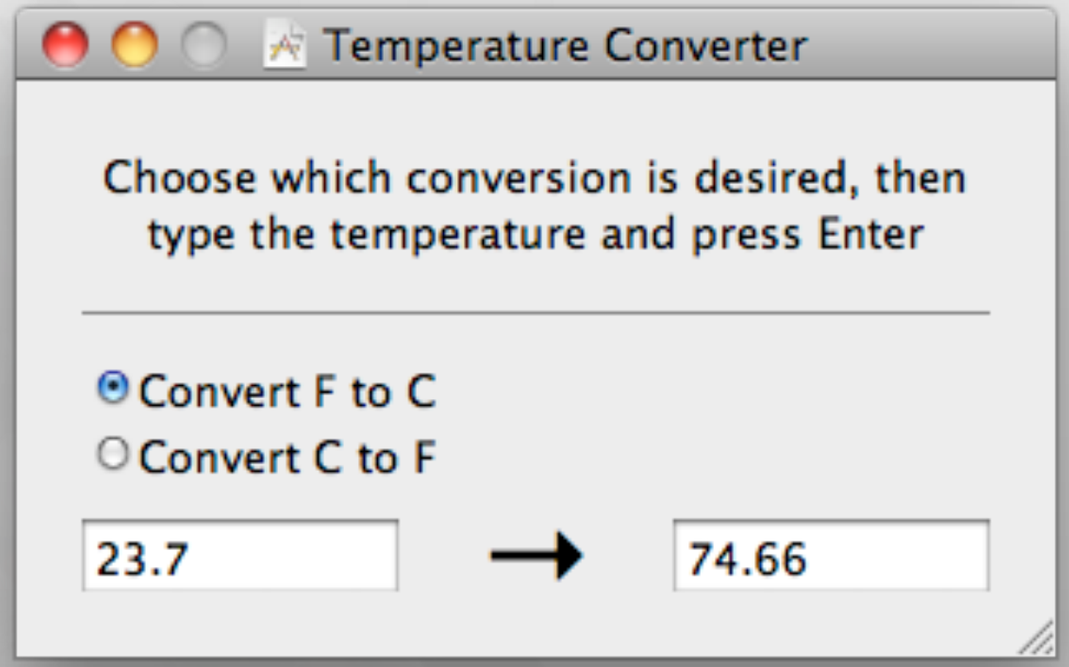
- Rule 5, deletion of overlapped Ms
  - Do not count any M that overlaps an R

    E.g., user waiting for computer response

# Exercise: Temperature Converter

- Convert from degrees Fahrenheit (F) to Celsius (C) or vice versa, requests equally distributed
- Use keyboard or mouse to enter temperature
- Assume active window awaiting input, an average of four typed characters (including point and sign), and no typing errors
- Task: create and analyze your own interface!

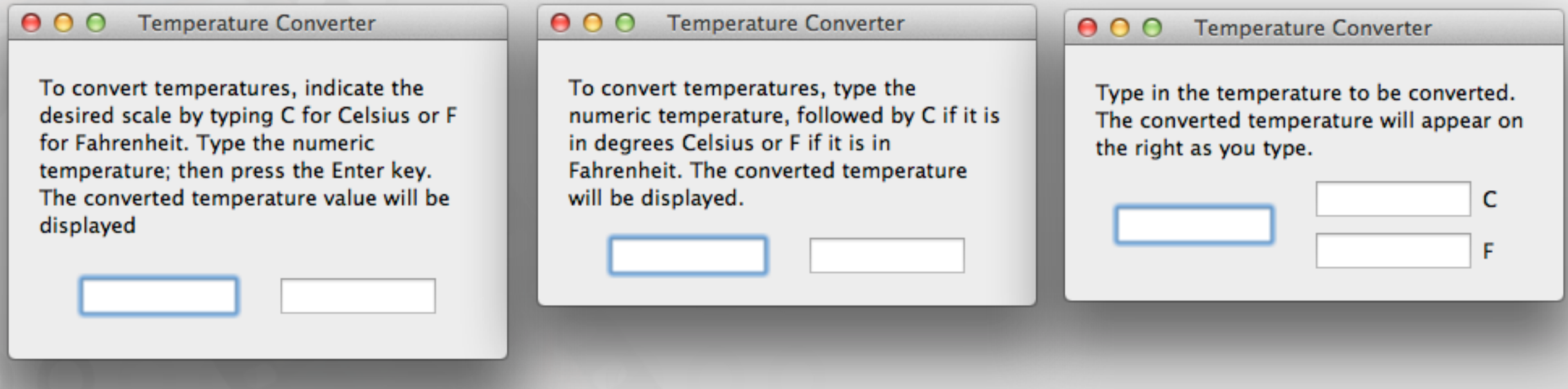
# The Dialog Box Solution with Radio Buttons…



# …and Its Keystroke-level Model

- Case 1: select conversion direction
  - Move hand to mouse, point to desired button, click on radio button (HPK)
  - Move hands back to keyboard, type four characters, tap enter (HPKHKKKKK)
  - Rule 0 (HMPMKHMKMKMKMKMK)
  - Rule 1, 2, 4 (HMPKHMKKKKMK)
  - Estimated time = 7.15 sec
- Case 2: correct conversion direction already selected
  - MKKKKMK = 3.7 sec
- Average time = (7.15 + 3.7) / 2 = 5.4 sec

# Example: Temperature Converter




- Keystroke efficiency
  - Type C or F, value, enter: M K K K K K M K ⇒ 3.9 sec (char. eff. 67 %)
  - Type value, then C or F: M K K K K M K ⇒ 3.7 sec (char. eff. 80%)
  - Bifurcated: M K K K K = 2.15 sec (char. eff. 100 %)

# Example: Temperature Converter

- Input assumptions (given)
  - 50% Fahrenheit, 50% Degree Celsius
  - 75% positive, 25% negative
  - 10% integer, 90% decimal
  - All digits are equally likely
  - Only four characters input

# Example: Temperature Converter

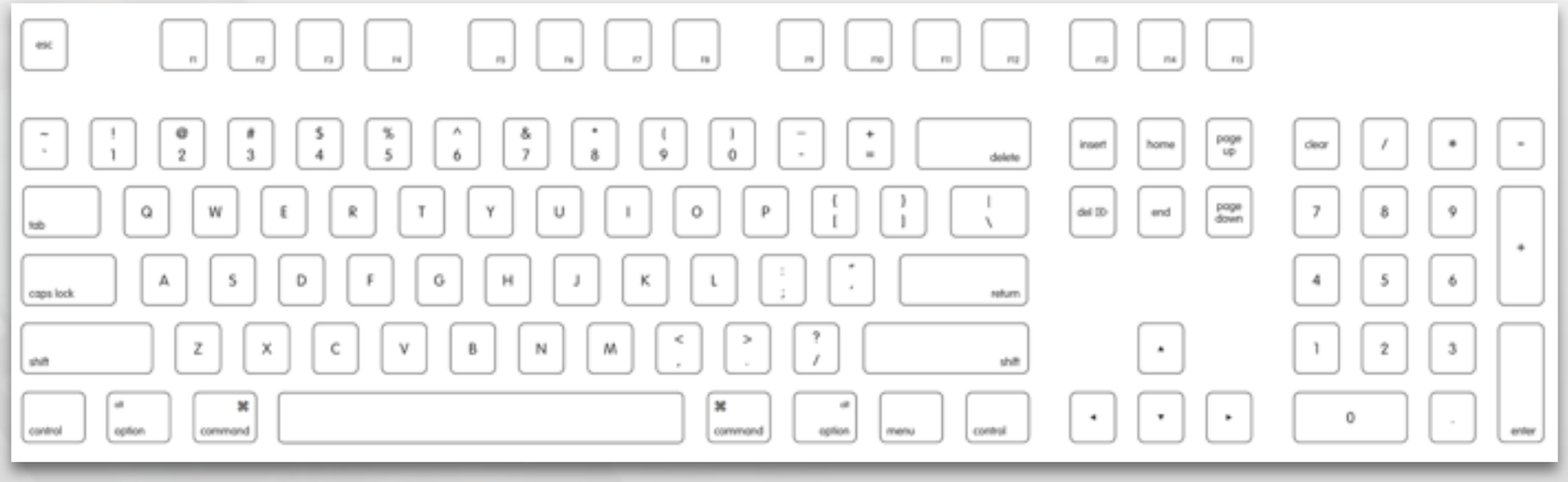
Information per alternative
$p(i)\log_2(1 / p(i))$

| Numbers | Prob. | Values | $p(i)$ | Information in bits | Overall (values x information in bits) |
|---|---|---|---|---|---|
| -.dd | 12.5% | 100 | 0.00125 | 0.012 | 1.2 |
| -d.d | 12.5% | 100 | 0.00125 | 0.012 | 1.2 |
| .ddd | 25% | 1000 | 0.00025 | 0.003 | 3 |
| d.dd | 25% | 1000 | 0.00025 | 0.003 | 3 |
| dd.d | 25% | 1000 | 0.00025 | 0.003 | 3 |

⇒ Minimal info required for the task = 11.4 bits/message

⇒ Simple approach: $4 \log_2(12) \approx 14$ bits

# Example: Temperature Converter




- Information efficiency: $E = \frac{\text{11.4 bits}}{\text{Info supplied by user}}$
  - 128 keys standard keyboard (5 bits/key): E = 11.4 / (4 x 5) ≈ 55%
  - 16 keys numeric keypad: E = 11.4 / (4 x 4) ≈ 70%
  - 12 keys dedicated keypad: E = 11.4 / (4 x 3.5) ≈ 80%

Minimal information content → Information-theoretic Efficiency → Interface Efficiency

KLM → Character Efficiency → Interface Efficiency

# GOMS: Components

- Goals describe user's end goals
  - Routine tasks, not too creative/problem-solving

    E.g., "copyedit manuscript"
  - Leads to hierarchy of subgoals
- Operators are elementary user actions
  - Key presses, menu selection, drag & drop, reading messages, gestures, speech commands, …
  - Assign context-independent duration (in ms)
- Methods are "procedures" to reach a goal
  - Consist of subgoals and/or operators
- Selection rules
  - Which method to use for a (sub)goal

    E.g., to delete some text (individual preferences apply!)

# Sample Method and Operators

GOAL: HIGHLIGHT-ARBITRARY-TEXT

1. MOVE-CURSOR-TO-BEGINNING 1.10s
2. CLICK-MOUSE-BUTTON 0.20s
3. MOVE-CURSOR-TO-END 1.10s
4. SHIFT-CLICK-MOUSE-BUTTON 0.48s
5. VERIFY-HIGHLIGHT 1.35s

# GOMS Results

- Execution (& learning) times of trained, routine users for repetitive tasks (goals), leading to cost of training, daily use, errors
  - Can be linked to other costs (purchase, change, update system), resulting in $$$ answers
  - Use to model alternative system offers

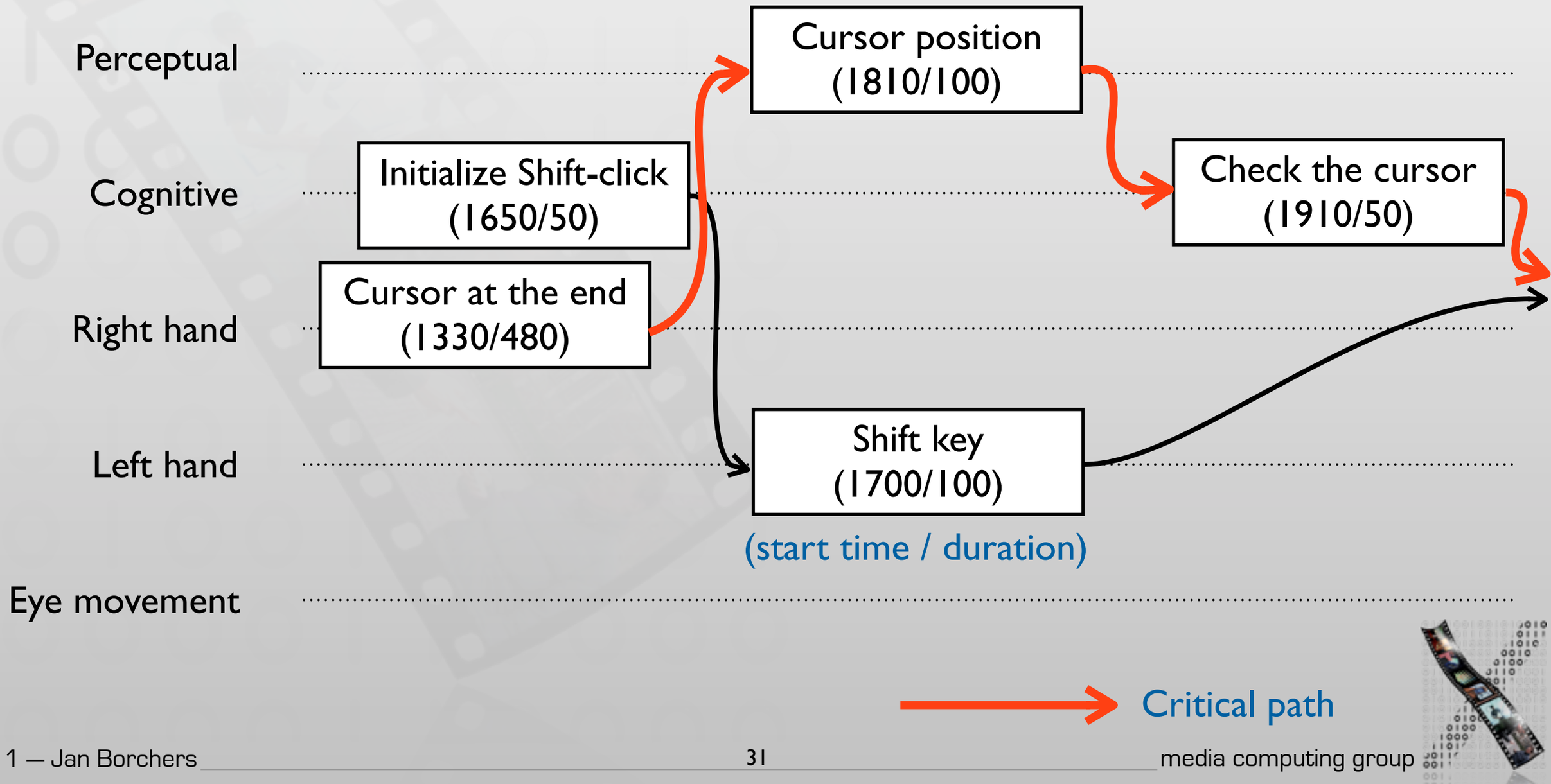
    E.g., "new NYNEX computers cost $2M/year more" [Gray93]

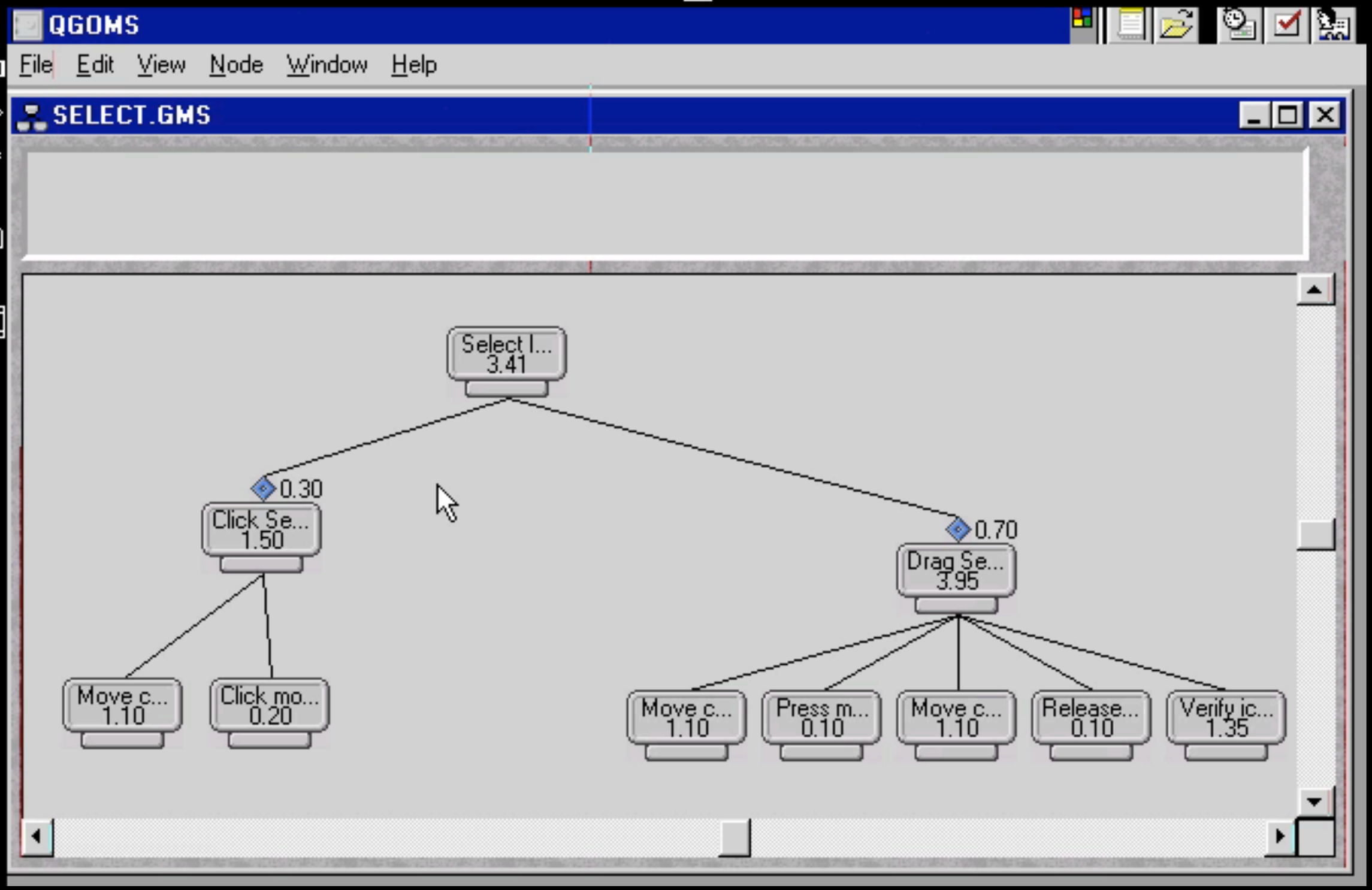
- Estimate effects of redesign
  - Training cost vs. long-term work time savings
- Starting point for task-oriented documentation
  - Online help, tutorials, …
- Don't use for casual users or new UI techniques
  - Operator times not well defined

# Variants of GOMS

- GOMS (Card, Moran, and Newell 1983)
  - Model of goals, operators, methods, selection rules
  - Predict time an experienced worker needs to perform a task in a given interface design
- Keystroke-level GOMS model (simplified version)
  - Comparative analyses of tasks that use mouse (GID) and keyboard
  - Correct ranking of performance times using different interface designs
- NGOMSL (natural GOMS language)
  - Considers non-expert behavior (e.g., learning times)
- CPM-GOMS (critical path method)
  - Computes more accurate absolute times
  - Considers overlapping time dependencies

# CPM-GOMS Example (Excerpt)

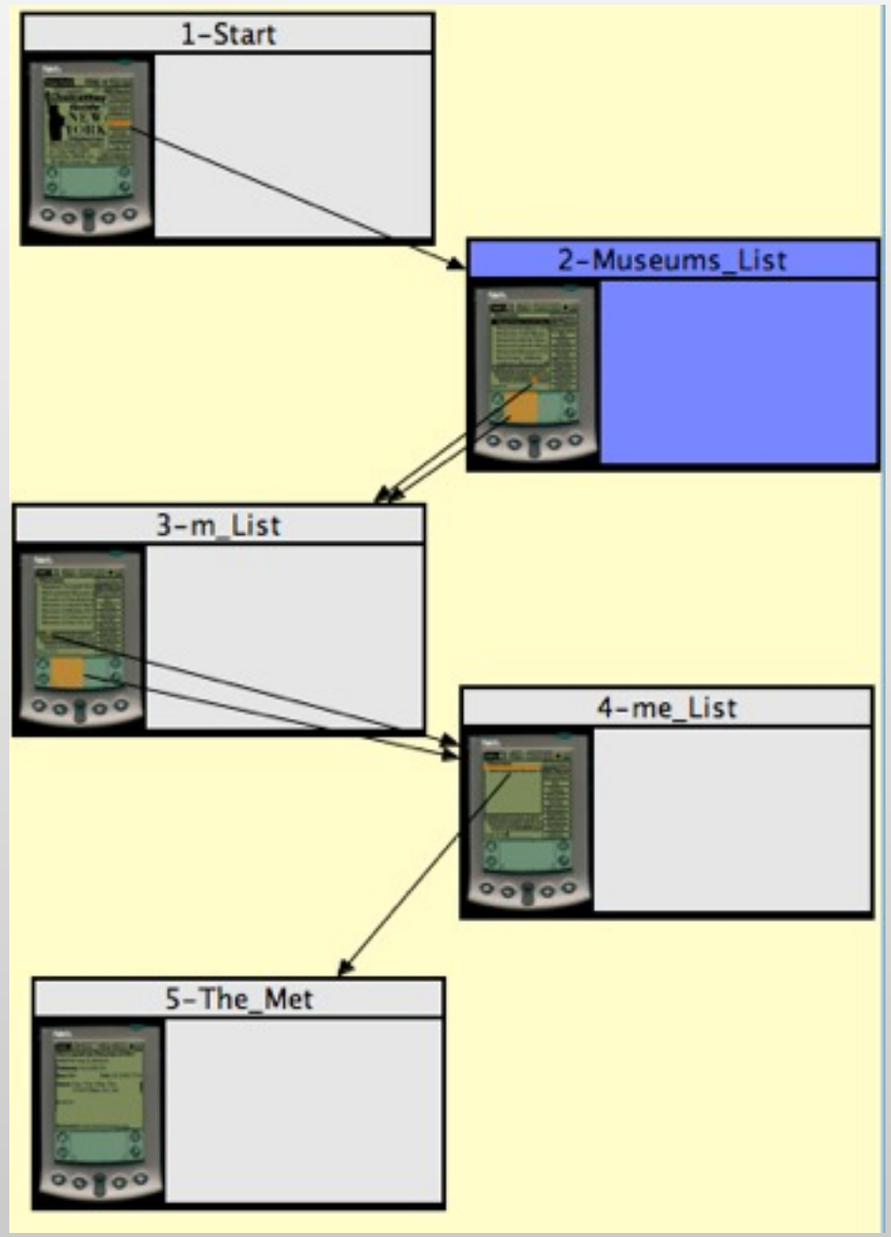


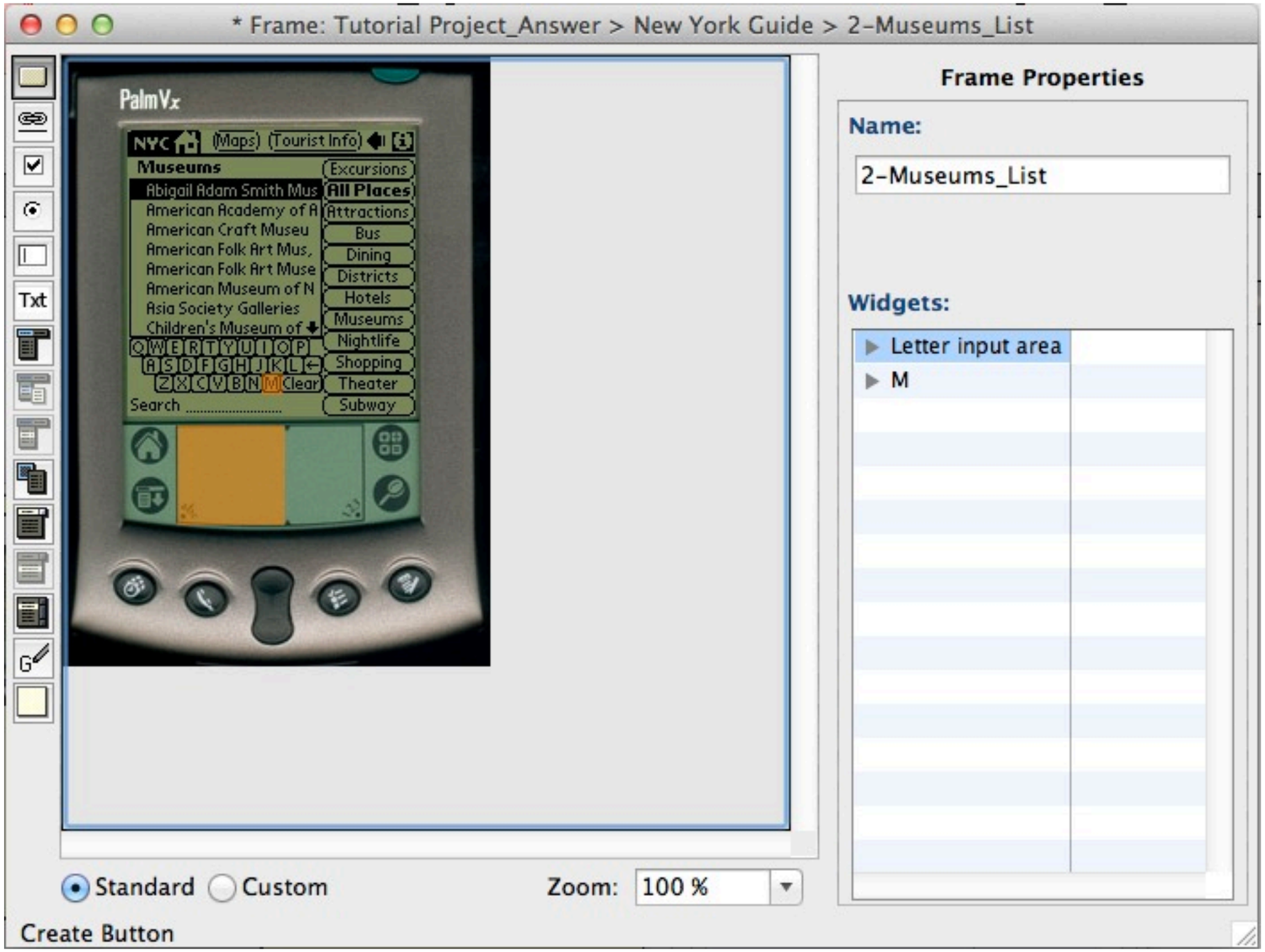



# Sample tool: QGOMS [Beard96]

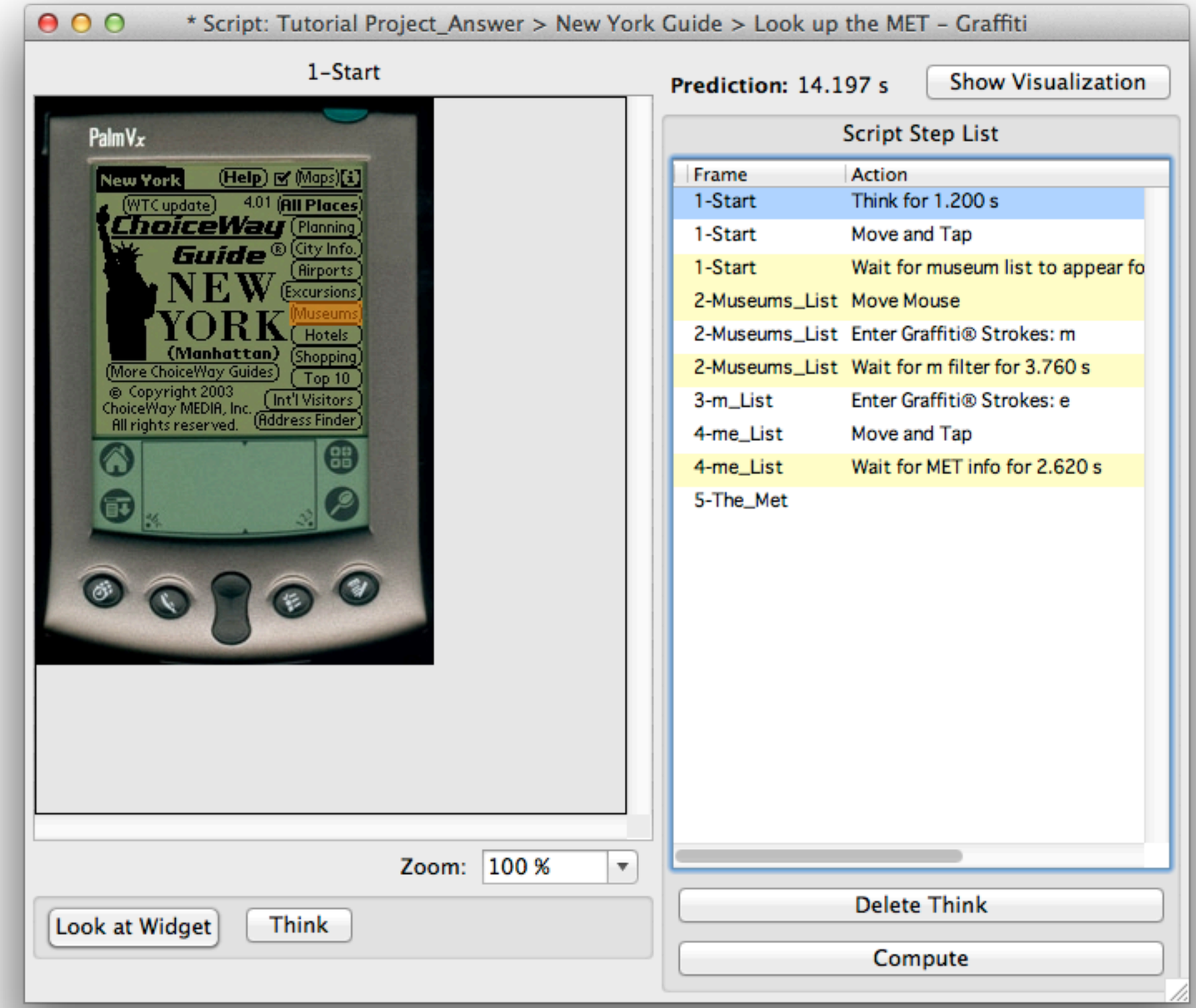
# CogTool

- UI prototyping tool with predictive human performance model
  - Create different storyboards
  - Demonstrate tasks on the storyboards
  - Produce cognitive model
- Available for free, Java
- http://cogtool.hcii.cs.cmu.edu/

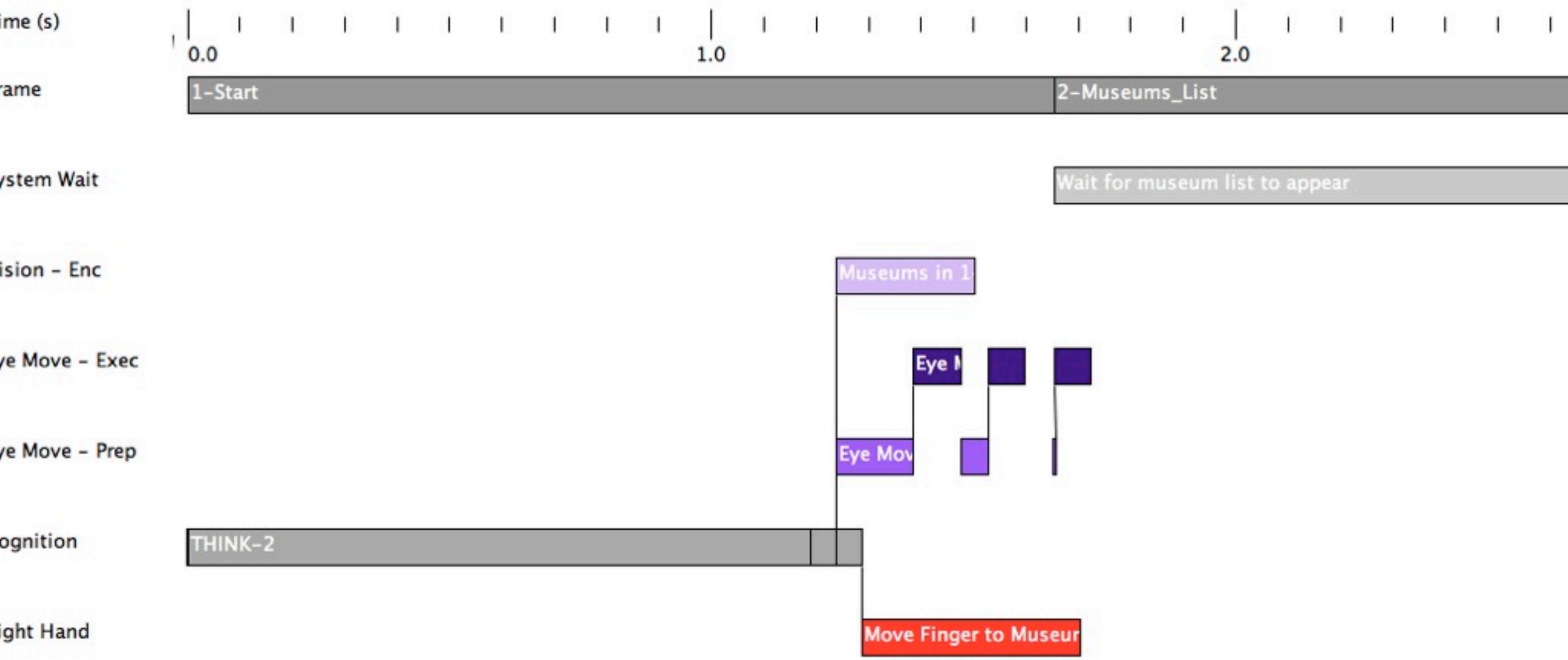





# CogTool: Defining hit zones for the UI




# CogTool: Record interactions




# CogTool: Visualize interactions in a timeline

# Evaluation Techniques

## Evaluating Without Users

E1 Literature Review

E2 Cognitive Walkthrough

E3 Heuristic Evaluation

E4 Model-based Evaluation (GOMS,...)

## Evaluating With Users

### Qualitative

E5 Model Extraction

E6 Silent Observation

E7 Think Aloud

E8 Constructive Interaction

E9 Retrospective Testing

### Quantitative

E10 Controlled Experiments

+ Interviews, questionnaires,...

# Evaluating with Users

- E1–E4 evaluate designs without the user
- As soon as implementations (prototypes) exist they should also be tested with users, using the following methods

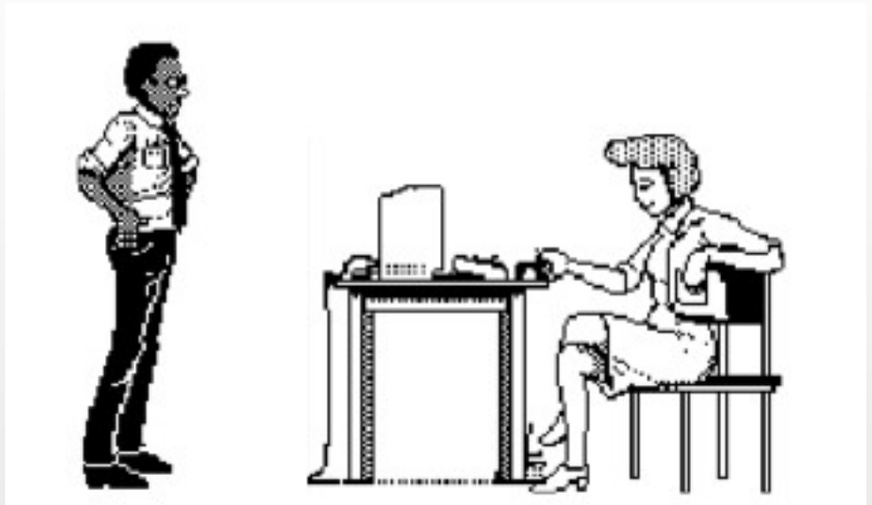
# E5: Model Extraction

- Designer shows user prototype or screen shots
- User tries to explain elements and their function

+ Good to understand naïve user's conceptual model of the system

– Bad to understand how the system is learned over time

# E6: Silent Observation



Saul Greenberg

- Designer watches user in lab or in natural environment while working on one of the tasks

- No communication during observation

\+ Helps discover big problems

– No understanding of decision process (that lead to problems) or user's mental model, opinions, or feelings

# E7: Think Aloud


- As E7, but user is asked to say aloud
  - What she thinks is happening (state)
  - What she is trying to achieve (goals)
  - Why she is doing something specific (actions)

- Most common method in industry

+ Good to get some insight into user's thinking, but:
  - Talking is hard while focusing on a task
  - Feels weird for most users to talk aloud
  - Conscious talking can change behavior

# E8: Constructive Interaction

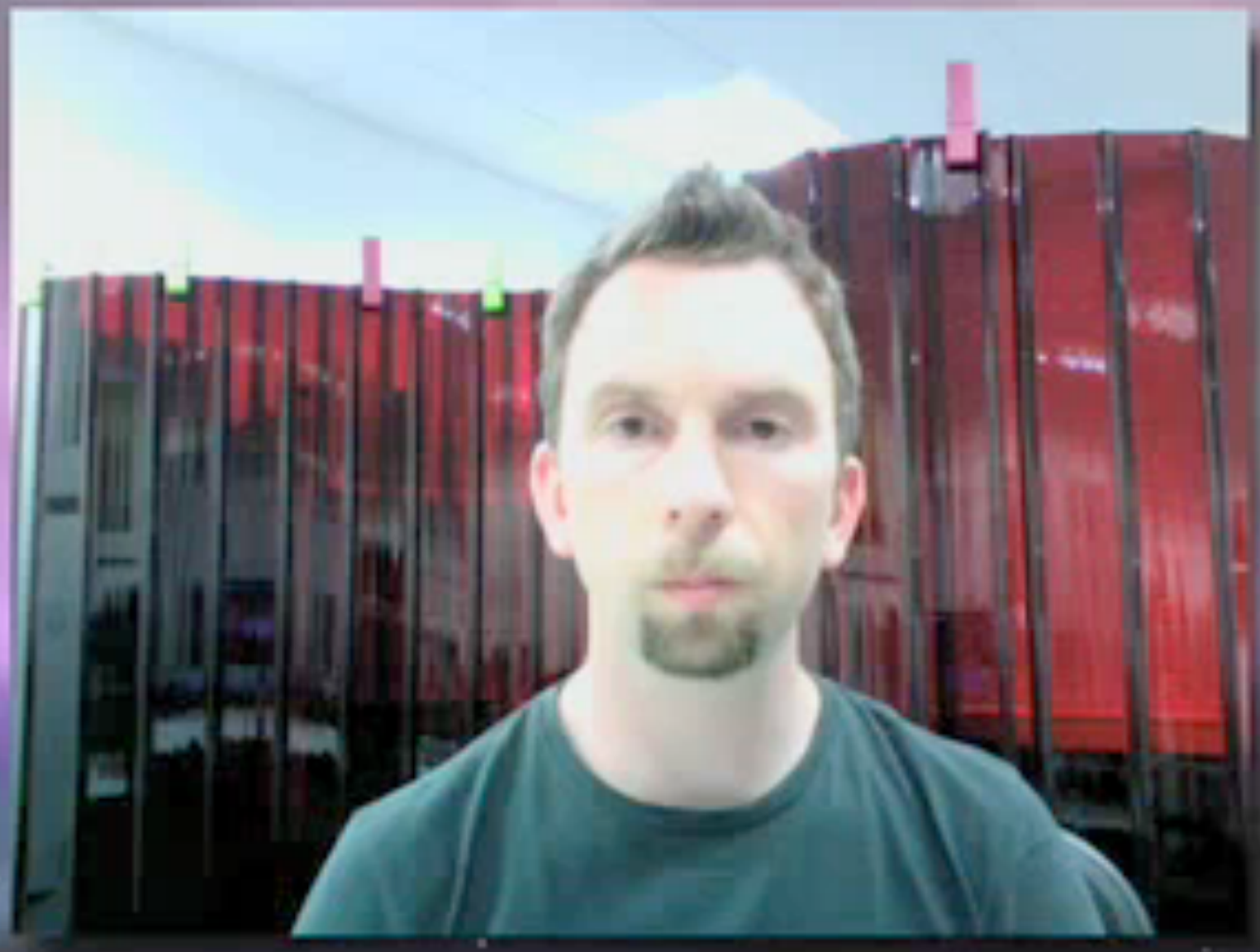

- Two people work on a task together
  - Normal conversation is observed (and recorded)
  - More comfortable than Think Aloud
- Variant of this: Different partners
  - Semi-expert as "trainer", newbie as "student"
  - Student uses UI and asks, trainer answers
  - Good: Gives insight into mental models of beginner and advanced users at the same time!

# Recording Observations

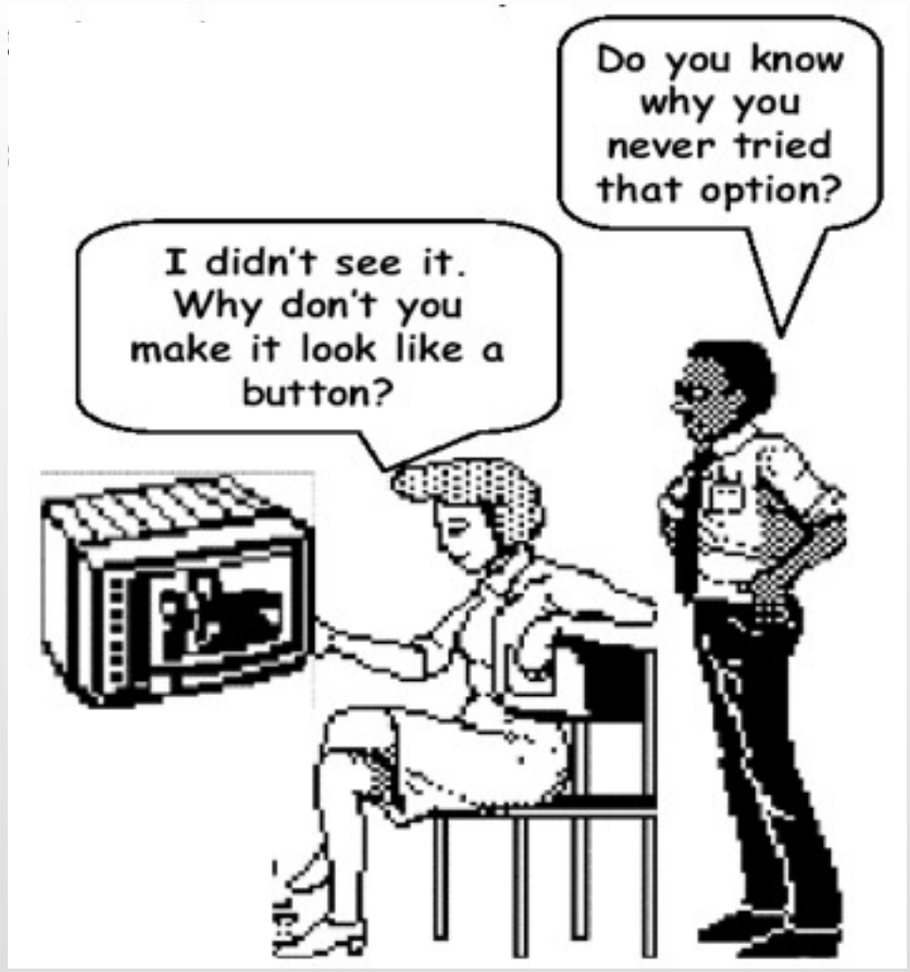
- Paper + pencil
  - Evaluator notes events, interpretations, other observations
  - Cheap but hard with many details (writing is slow). Forms can help.
- Audio recording
  - Good for speech with Think Aloud and Constructive Interaction
  - But hard to connect to interface state
- Video
  - Ideal: two cameras (user + screen) in one picture
  - Best capture, but may be too intrusive initially

# Silverback


screen capture
+
live video & audio
magic!
Silverback
UPA – Sophie Barrett.mov
Quicktime
composite



# E9: Retrospective Testing




- Additional activity after an observation
- Subject and evaluator look at video recordings together, user comments his actions retrospectively
- Good starting point for subsequent interview, avoids wrong memories
- Often results in concrete suggestions for improvement

# E10: Controlled Experiments

- Quantitative, empirical method
- Steps:
  - Formulate hypothesis
  - Design experiment, pick variable and fixed parameters
  - Choose subjects
  - Run experiment
  - Interpret results to accept or reject hypothesis

More details: next lecture

# Other Evaluation Methods

- Before and during the design, with users:
  - Questionnaires
  - Personal interviews
- After completing a project:
  - Email bug report forms
  - Hotlines
  - Retrospective interviews and questionnaires
  - Field observations (observe running system in real use)



# Evaluation Techniques

## Evaluating Without Users

E1 Literature Review

E2 Cognitive Walkthrough

E3 Heuristic Evaluation

E4 Model-based Evaluation (GOMS,...)

## Evaluating With Users

### Qualitative

E5 Model Extraction

E6 Silent Observation

E7 Think Aloud

E8 Constructive Interaction

E9 Retrospective Testing

### Quantitative

E10 Controlled Experiments

Next lecture

+ Interviews, questionnaires,...

# Dealing with Testers

- Tests are uncomfortable for the tester
  - Pressure to perform, mistakes, competitive thinking

- So treat testers with respect at all times!
  - Before, during, and after the test

# Before the Session

- Do not waste the tester's time
  - Run pilot tests before
  - Have everything ready when testers arrive
- Make sure testers feel comfortable
  - Stress that the system is being tested, not them
  - Confirm that the system may still have bugs
  - Let testers know they can stop at any time
- Guarantee privacy
  - Individual test results will be handled as private
- Inform tester
  - Explain what is being recorded
  - Answer any other questions (but do not bias)
- Only use volunteers (consent form)

# During the Session

- Do not waste the testers' time
  - Do not let them complete unnecessary tasks
- Make sure testers are comfortable
  - Early success in the task possible
  - Relaxed atmosphere
  - Breaks, coffee, …
  - Hand out test tasks one by one
  - Never show you are unsatisfied with what the tester does
  - Avoid interruptions (cell phones, …)
  - Abort the test if it becomes too uncomfortable
- Guarantee privacy
  - Never let testers' boss (or others) watch

# After the Session

- Make sure testers are comfortable
  - Stress that tester has helped finding ways to improve the system
- Inform
  - Answer any questions that could have changed the experiment if answered before the test
- Guarantee privacy
  - Never publish results that can be associated with specific individuals
  - Show recordings outside your own group only with written consent from testers

# Evaluation Techniques

## Evaluating Without Users

E1 Literature Review

E2 Cognitive Walkthrough

E3 Heuristic Evaluation

E4 Model-based Evaluation (GOMS,...)

## Evaluating With Users

### Qualitative

E5 Model Extraction

E6 Silent Observation

E7 Think Aloud

E8 Constructive Interaction

E9 Retrospective Testing

### Quantitative

E10 Controlled Experiments

Next lecture

+ Interviews, questionnaires,...