Designing Interactive Systems I

Controlled Experiments, GOMS, Interface Efficiency

Prof. Dr. Jan Borchers
Media Computing Group
RWTH Aachen University

Winter Semester ’20/’21

https://hci.rwth-aachen.de/dis

Review: 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

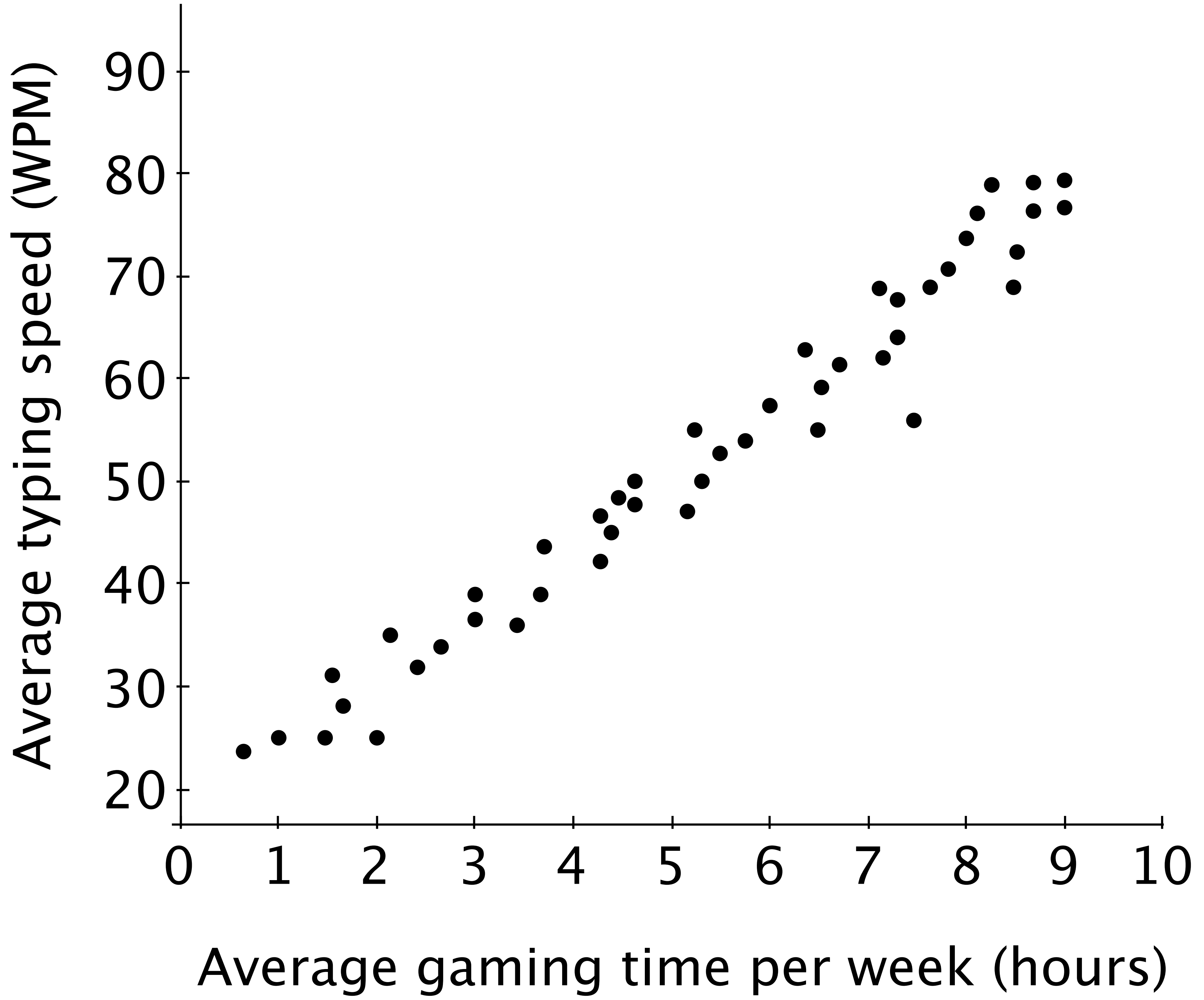
Quantitative

E10 Controlled Experiments

+ Interviews, questionnaires,...

Review: Controlled Experiments (E10)

- Quantitative, empirical method
- Used to identify the cause of a situation or set of events
    - “X is responsible for Y”
  - Directly manipulate and control variables
- Correlation does not imply causality
  - Example: relationship between typing speed and time spent playing games
- Use a controlled experiment to verify an observation, a correlation, or a “hunch”



Review: Basic Experimental Designs

- **Between-groups design**
  - Each subject only does one variant of the experiment
  - There are at least 2 groups to isolate effect of manipulation:
    - Treatment group and control group
  - \+ No learning effects across variants
    - Good for tasks that are simple and involve limited cognitive processes, e.g., tapping, dragging, or visual search
  - – But: requires more users

- **Within-groups design**
  - Each subject does all variants of the experiment
  - \+ Fewer users required, individual differences canceled out
    - Good for complex tasks, e.g., typing ,reading, composition, problem solving
  - – But: learning effects may occur

Within-Groups Design: Order Effect

- The order of presenting the treatments (IV levels) might affect the dependent variable
  - Learning effect
  - Fatigue effect
- Solutions
  - Rest period between treatments
  - Counterbalancing: all possible orders of treatments are included — but: O(n!)
  - Latin Square: A limited set of orders, O(n)

Latin Square

- Each condition appears at each ordinal position
- Each condition precedes and follows each other condition once
- Example for six treatments (A, B, C, D, E, F)

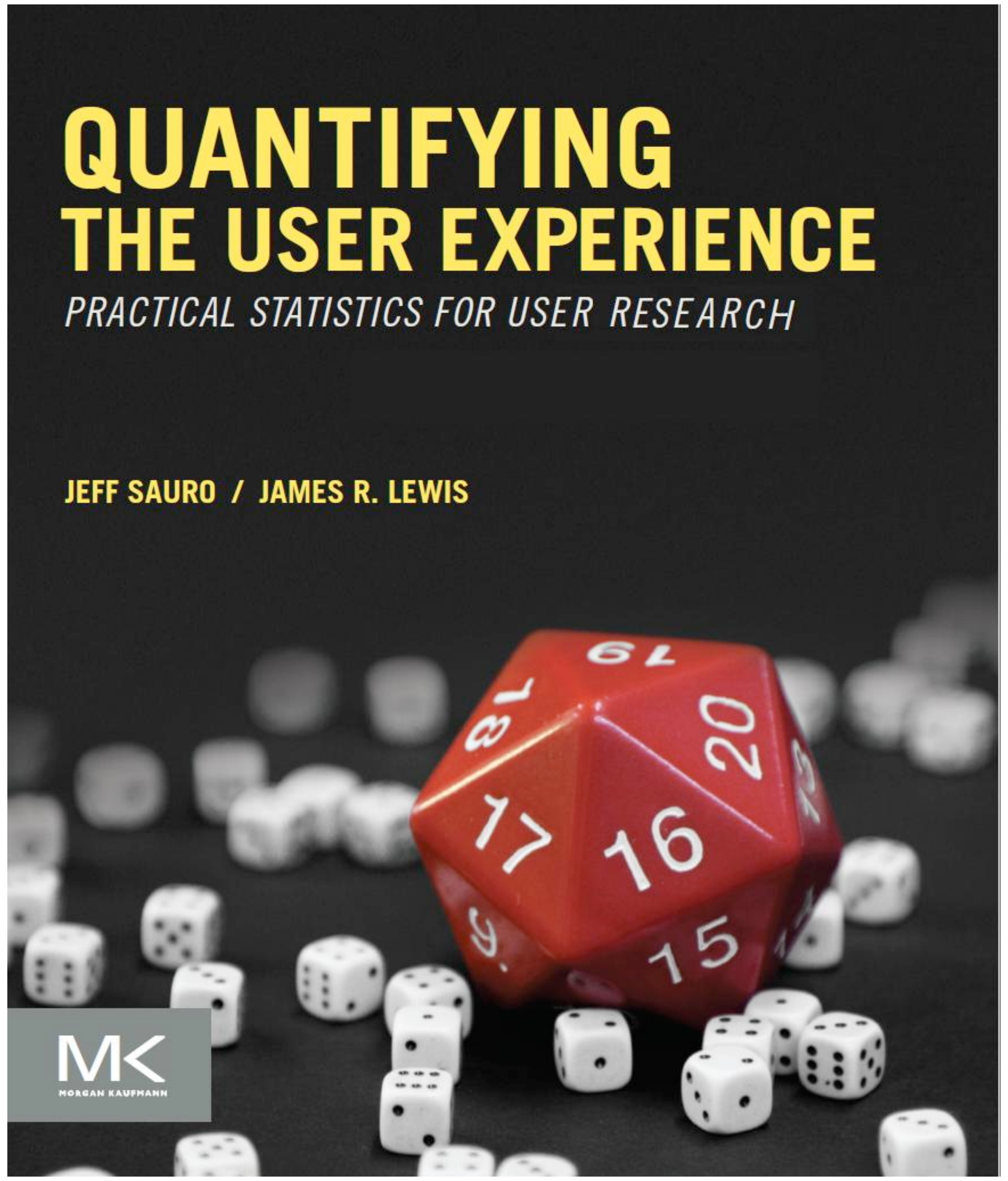
1	A	B	F	C	E	D
2	B	C	A	D	F	E
3	C	D	B	E	A	F
4	D	E	C	F	B	A
5	E	F	D	A	C	B
6	F	A	E	B	D	C

Randomization

- Randomly assign treatments to participants
- Prevents systematic bias
- But: randomization ≠ counterbalancing
  - With small numbers, randomization might not cover all combinations

Analyzing Results

- Do statistical analysis using well-defined test methods
  - E.g., Student's *t*-test, ANOVA (analysis of variance), regression analysis, Wilcoxon or Mann/Whitney test, χ^2 test
- Choice depends on number, continuity, and assumed distribution of variables, and the desired form of the result
  - Results can be simple "yes/no", size of difference, or confidence of estimate



Making Your Evaluation Valid and Reliable

- Validity: How accurate is your result?
- Reliability: How consistent or stable is your result?

- These apply to *all* evaluations — not just controlled experiments

Validity

- **Construct validity:** Were variables defined concretely enough to be manipulated or measured?

  ✗ Speed

  ✓ Time since the cursor leaves the start until it reaches the target

- **Internal validity:** Is the causal inference logical? How strong is it?
  - Usually higher in experimental methods than descriptive or correlational methods
- **External validity:** Can the result be generalized to other populations and settings?
  - Evaluations in the lab usually have lower external validity than those in the field

Reliability

- Can the experiment be replicated by other research teams in other locations and yield results that are consistent, dependable, and stable?
- Is the experimental procedure clearly described in the paper/report?
- Other causes of fluctuation
  - Random errors: cannot be eliminated — testing with more samples can help
  - Systematic errors: push the measured value into the same direction, caused by:
    - Measurement instruments
    - Experimental procedures: not randomized, not counterbalanced, instructions are biased
    - Participants: the recruitment process may filter participants unevenly
    - Experimenter behavior: bias in spoken language during experiment
    - Environmental factors: physical environment might favor one treatment over others

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)

Dealing with Users

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

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

Before the Test

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

During the Test

- Do not waste the users’ time
    - Do not let them complete unnecessary tasks
- Guarantee privacy
    - Never let users’ boss (or others) watch

- Make sure users 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 user does
    - Avoid interruptions (cell phones, …)
    - Abort the test if it becomes too uncomfortable

After the Test

- Make sure the users are comfortable
  - Stress that the user 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 users

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

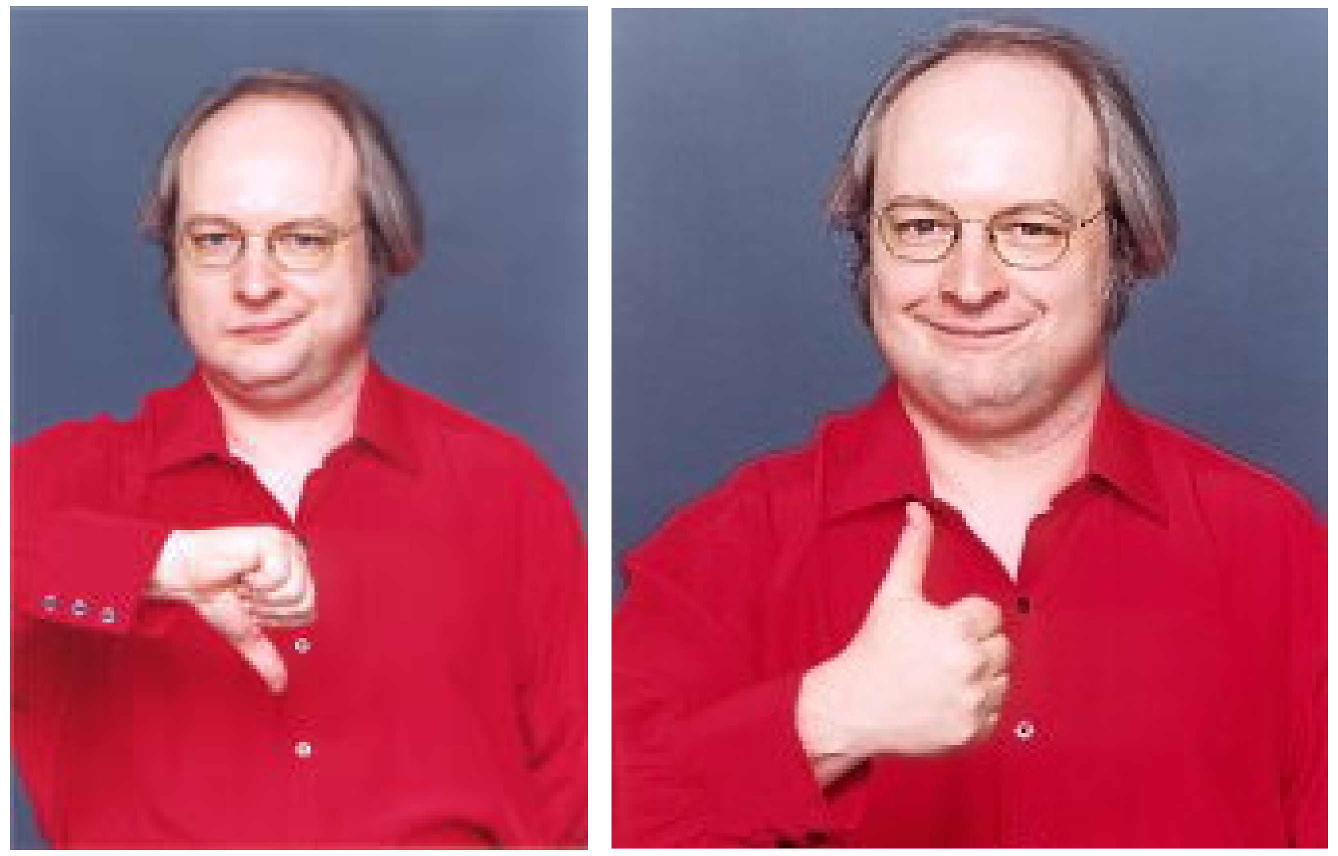
Quantitative

E10 Controlled Experiments

+ Interviews, questionnaires,...

GOMS

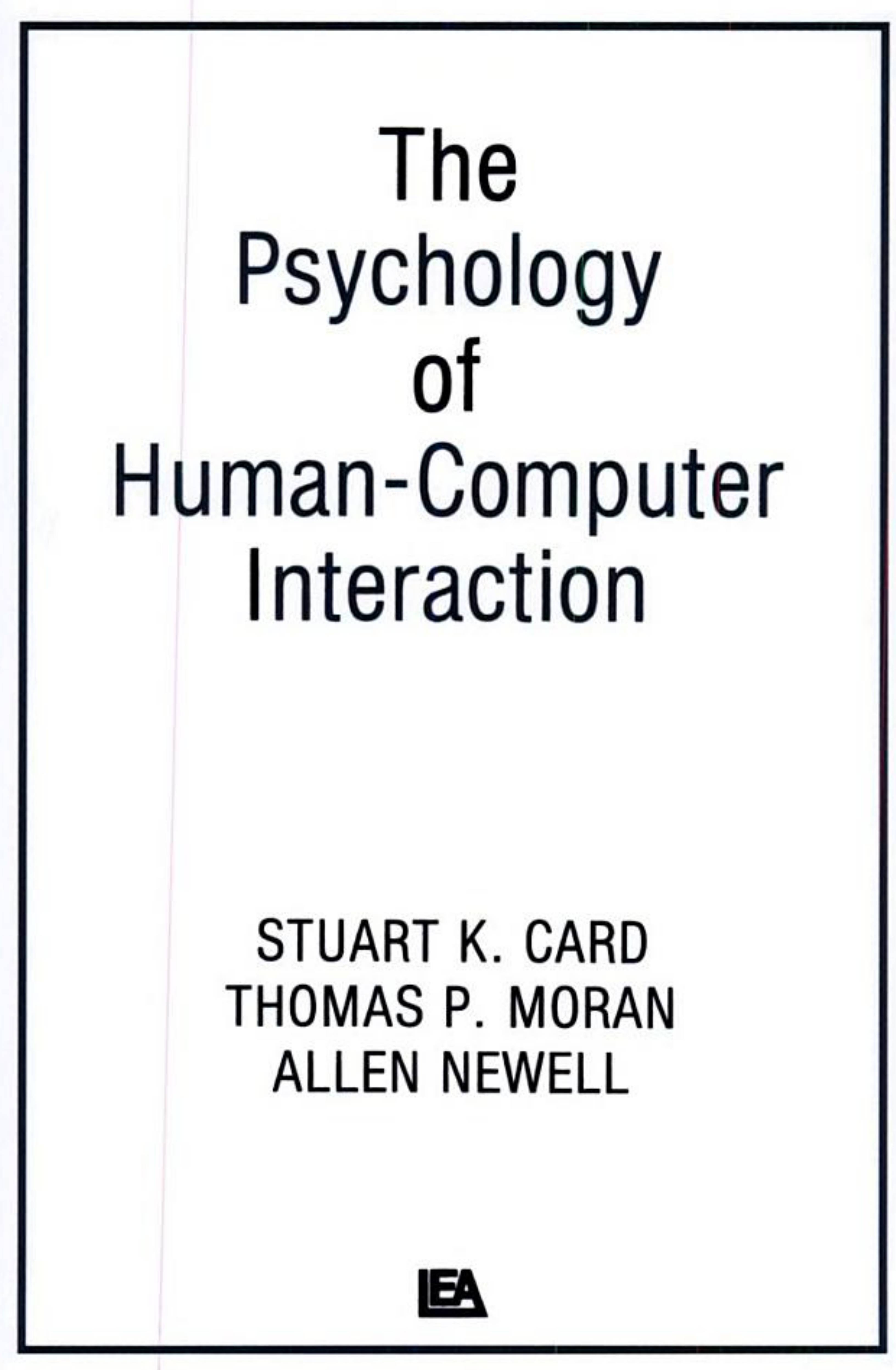
A Story



- 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



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

A.	MOVE-CURSOR-TO-BEGINNING	1.10s
B.	CLICK-MOUSE-BUTTON	0.20s
C.	MOVE-CURSOR-TO-END	1.10s
D.	SHIFT-CLICK-MOUSE-BUTTON	0.48s
E.	VERIFY-HIGHLIGHT	1.35s

GOMS Variants

- 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 model (simplified version)
  - Comparative analyses of tasks that use mouse (GID) and keyboard
  - Correct ranking of performance times using different interface designs
- CPM-GOMS (critical path method)
  - Computes accurate absolute times
  - Considers overlapping time dependencies
- NGOMSL (natural GOMS language)
  - Considers non-expert behavior (e.g., learning times)

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., “clear↵” ⇒ M K K K K K MK ⇒ M K K K K K K

      Note that the ‘clear’ command does not take any arguments, and is therefore a constant string. ‘ls,’ on the other hand, can take arguments and Rule 4 cannot be applied there.
- 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

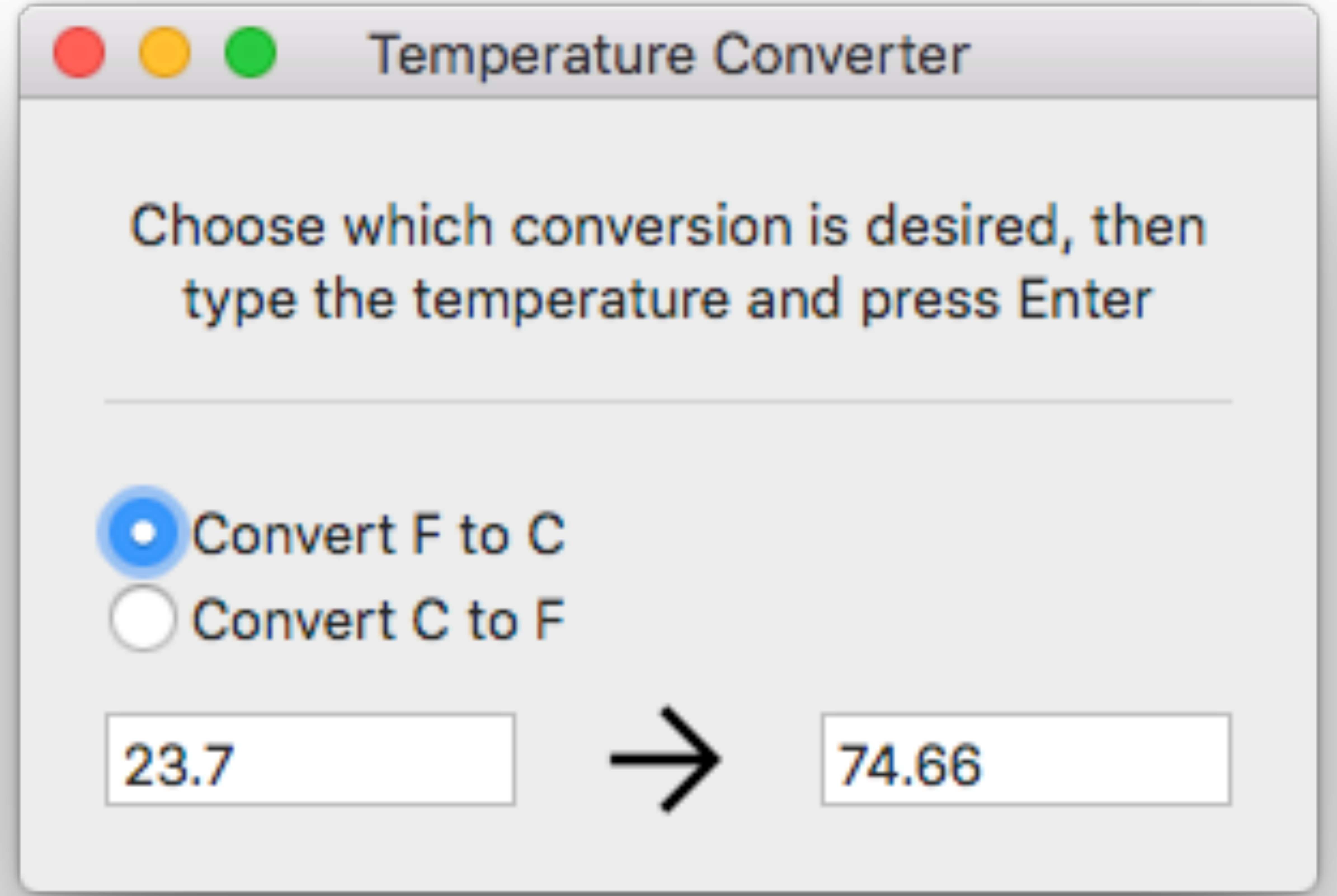
- Convert from degrees Fahrenheit (F) to Celsius (C) or vice versa, requests equally distributed

- 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 (HPK HKKKK K)
  - Rule 0 (insert M's): (HMPMK HMKMKMKMK MK)
  - Rule 1 (deletion of anticipated M's): (HMP_K HMKMKMKMK MK)
  - Rule 2 (deletion of M's within cog. units): (HMP_K HMK_K_K_K MK)
  - Result: HMPK HMKKKK MK
  - 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

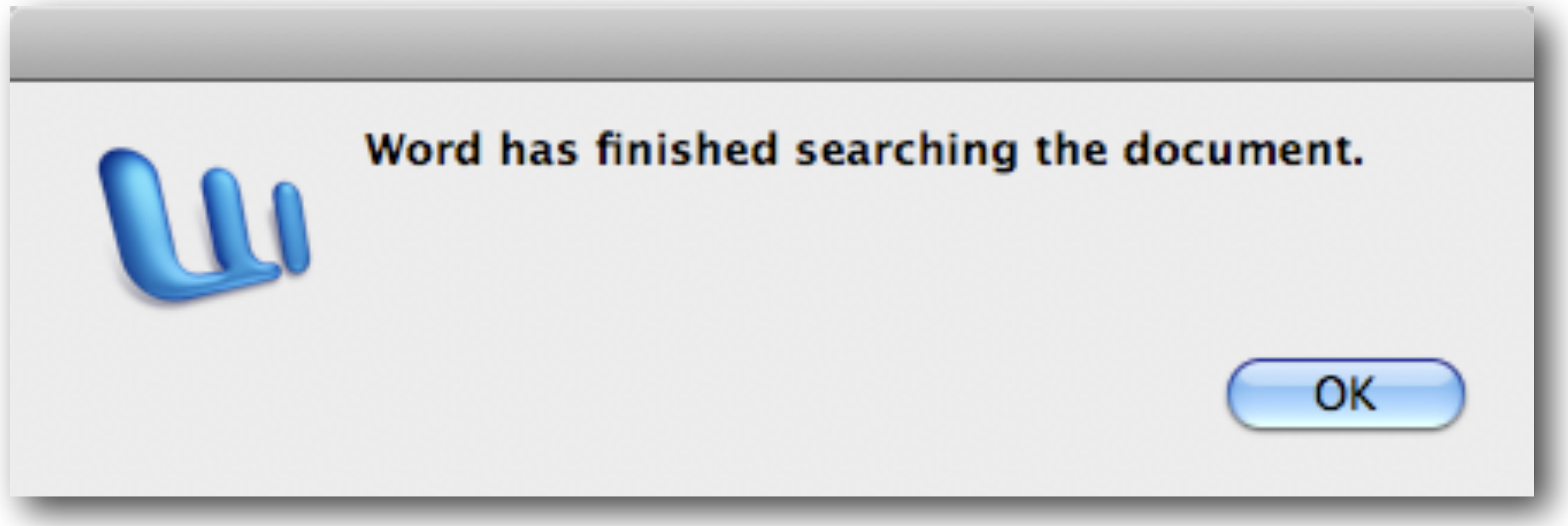
GOMS Results

- 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

Information 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 $E = \dfrac{\text{Minimal info required for the task}}{\text{Info supplied by user}}$
  - $E \in [0, 1]$ (e.g., $E = 0$ for providing unnecessary information)
- **Character efficiency** $= \dfrac{\text{Minimal number of characters required for the task}}{\text{Number of characters entered in the UI}}$

[Jef Raskin: The Humane Interface, 2000]

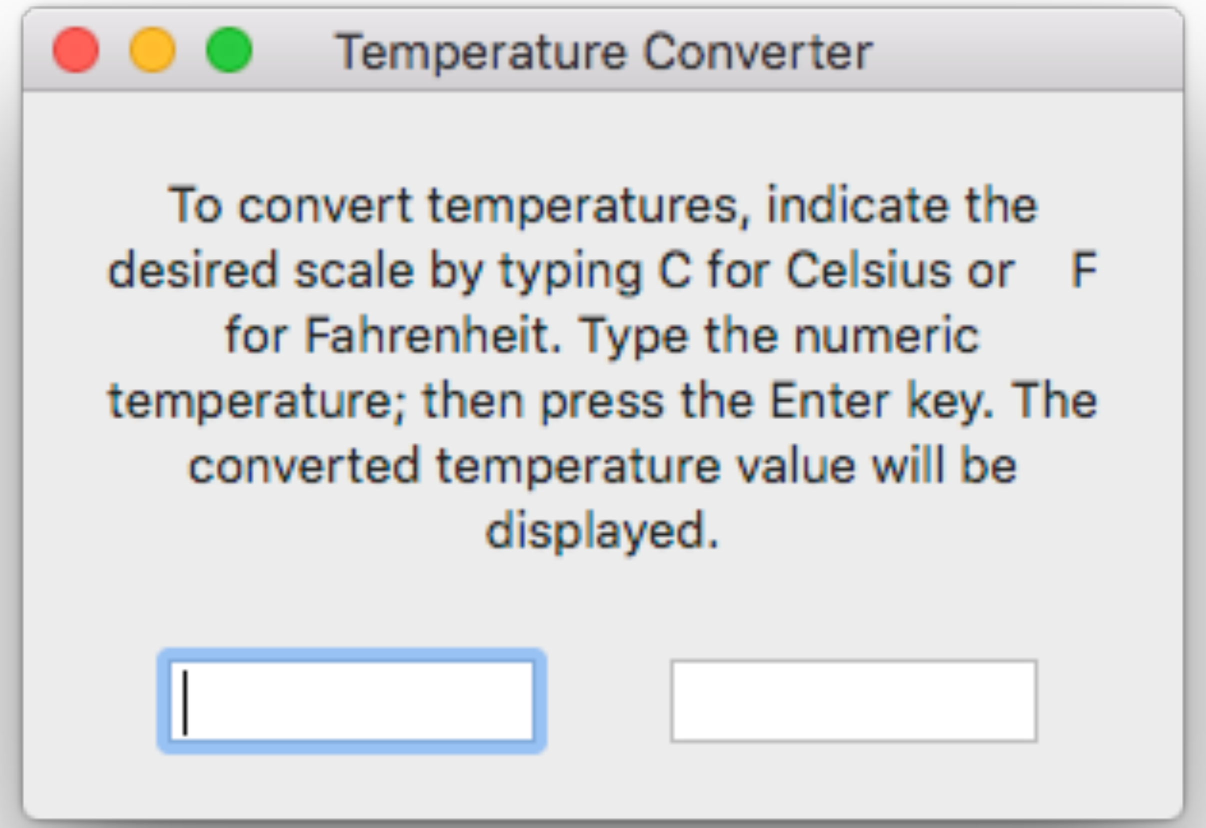
Quantify Amount of Data

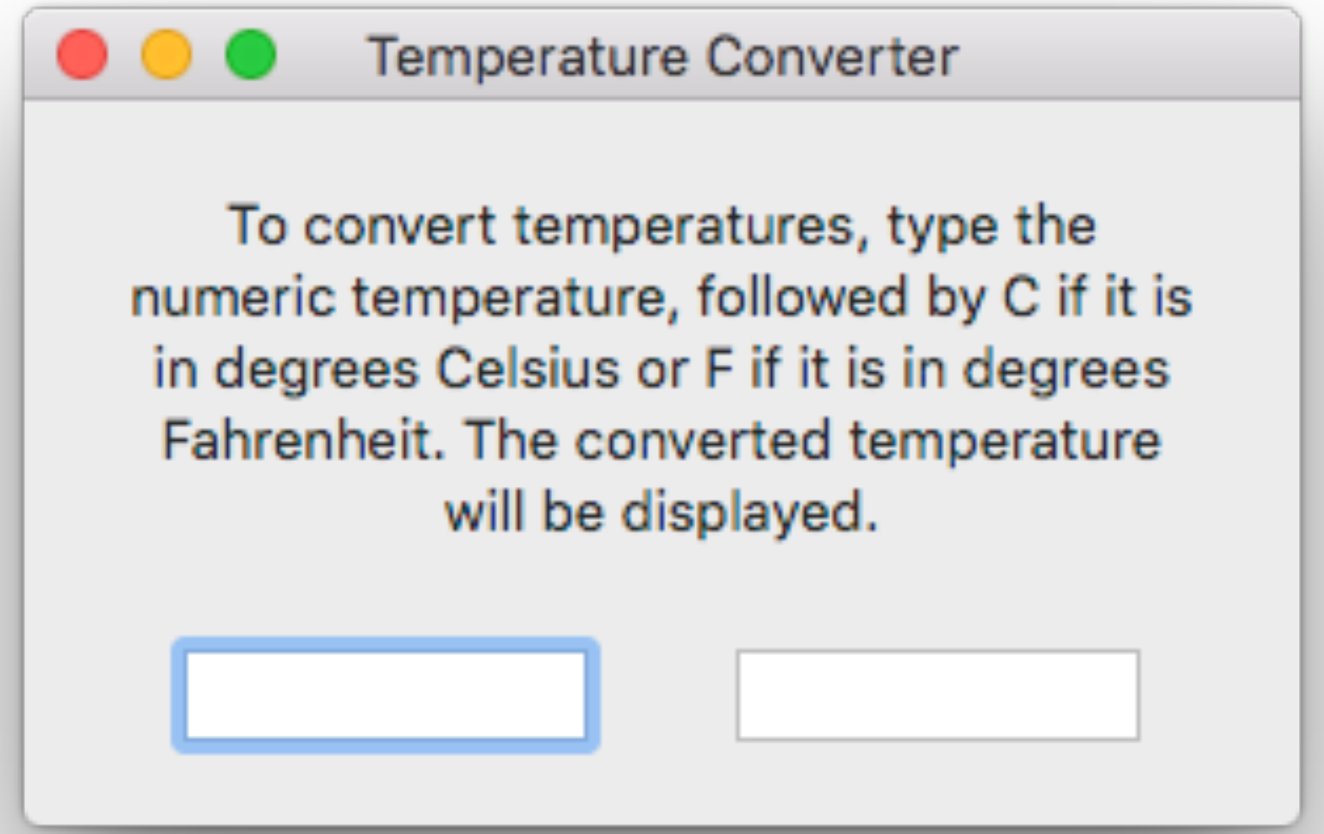
- Information is measured in bits
  - 1 bit represents choice between 2 alternatives
- n equally likely alternatives
  - Total information amount: $\log_2(n)$
  - Information per alternative: $\frac{1}{n}\log_2(n)$
- n alternatives with different probabilities $p(i)$
  - Information per alternative: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

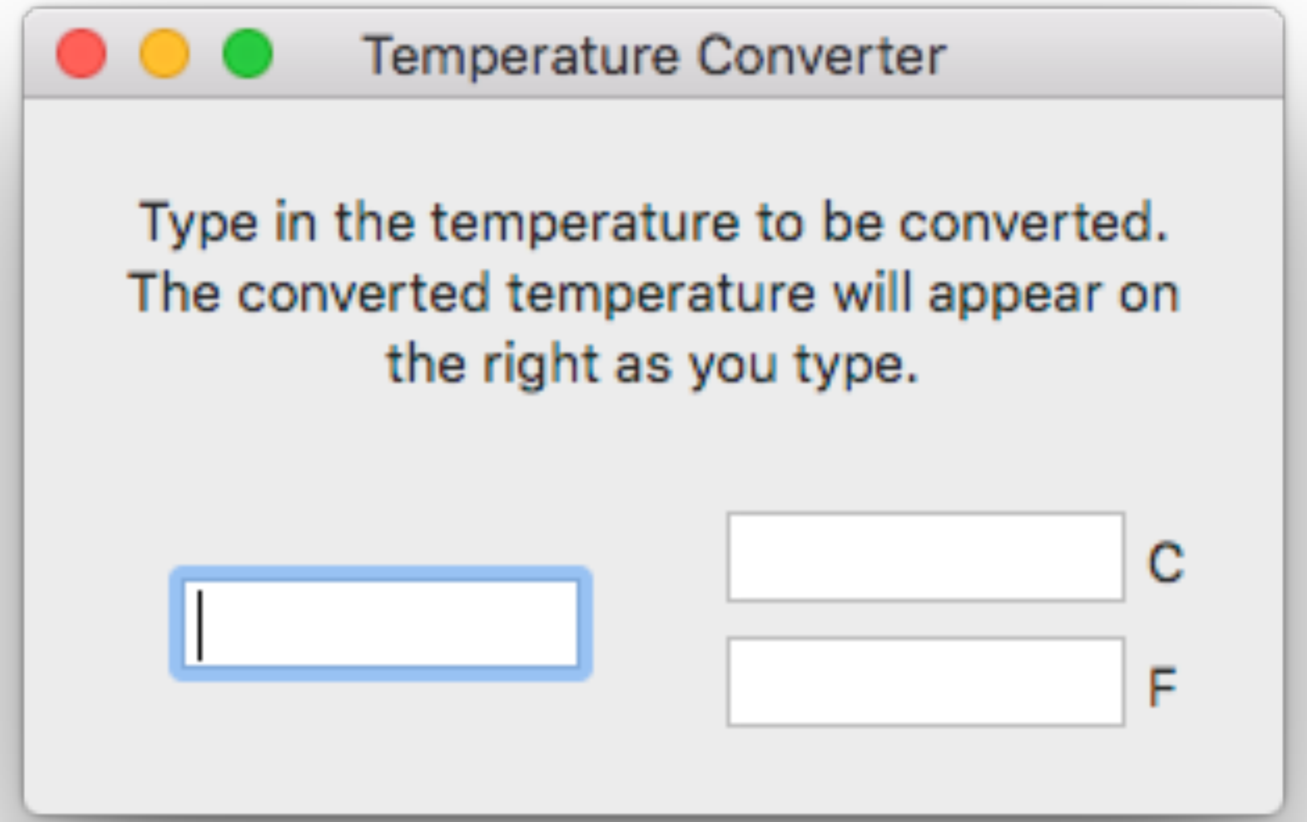
Example: Temperature Converter

- Input assumptions (given)
  - 50% Fahrenheit, 50% Degree Celsius
  - 75% positive, 25% negative
  - only decimal input (no integer numbers)
  - All digits are equally likely
  - Only four characters input

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

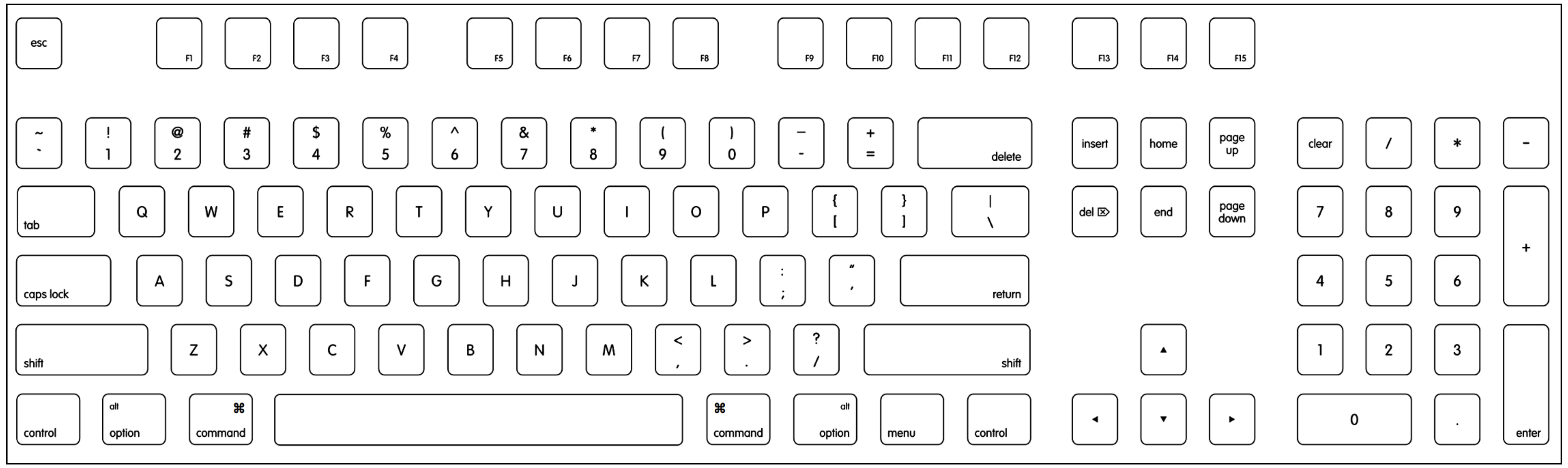
Information per alternative: $p(i) \cdot \log \frac{1}{p(i)}$

Numbers	Prob.	Values	$p(i)$	Information in bits	Overall (values × 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: $\text{E} = \dfrac{\text{11.4 bits}}{\text{Info supplied by user}}$
  - 128 keys standard keyboard (5 bits/key): $\text{E} = \dfrac{11.4}{4 \cdot 5} \approx 55\,\%$
  - 16 keys numeric keypad: $\text{E} = \dfrac{11.4}{4 \cdot 4} \approx 70\,\%$
  - 12 keys dedicated keypad: $\text{E} = \dfrac{11.4}{4 \cdot 3.6} \approx 80\,\%$

Summary: Evaluation

- When, why, where, and what?
  - To ensure that system matches the users' needs
  - In the Lab vs. In the Field
- Concrete methods to evaluate designs and implementations
  - E1 - E10, GOMS, Interface Efficiency
- How to deal with users