Human-Computer Interaction

Termin 8:

Statistical analysis

Model-based usability evaluation

Evaluation

- Why? Need *Usability* and Efficiency

- What? Usability criteria

- Where? Field study or lab experiment

- Who? Experte or novice user

- When? In *all* Studies
  (Ideas, Prototype, System)

Experimental design - factors

- Subjects
  - who – representative, sufficient sample size
- Variables
  - things to modify and measure
- Conditions
  - experimental conditions, differ only in the value(s) of some controlled variable(s)
- Hypothesis
  - what you'd like to show
  - derived from literature or some sort of `theory´ (not from data!)

Variables

- *independent* variable (IV)
  - characteristics changed to produce different conditions
    e.g. interface style, number of menu items
  - also called *controlled* variables
- *dependent* variable (DV)
  - characteristics measured in the experiment
    e.g. time taken, number of errors

Hypotheses

- formualte as if-then or the-the („je..desto“) statement
- formulate in three steps
  1. in terms of the underlying theory
  2. in terms of the variables
  3. in terms of statistical measures

$$\bar{x}(INTERAKTIONSZEIT_{MAUS}) < \bar{x}(INTERAKTIONSZEIT_{TASTATUR})$$

- Again, need to frame theoretical concepts in statistical terms

Hypotheses

- Statistical formulation calls for comparison of test series under different conditions
- Formulate and test possible explanations

- *Working hypothesis* or *alternative hypothesis* H_1
  - differences in test series are systematic and due to changes in controlled variables (IVs)
  - H_1 states expected outcome (how IVs influence DVs) $\bar{x}_A \neq \bar{x}_B$

- *Null hypothesis* H_0 :
  - there is no difference between conditions other than random variation
  - contraposition to working hypothesis
  - aim is to disprove this

    e.g. null hypothesis = "no change with font size" $\bar{x}_A = \bar{x}_B$

Principle of statistical tests

Disprove the *null hypothesis*, i.e. prove that differences between the conditions did *not* happen by chance.

Note:

Statistical conclusions are always generalizations from a sample to an overall population, where the sample will *always* be affected by random variation. There are thus no absolute decisions againts the null hypothesis, but only probabilities of their (in)validity!

Do not reject the null hypothesis before the results disprove it with a sufficient probability (significance).

Experimental design

Goal: controlled evaluation of aspects of interactive behavior

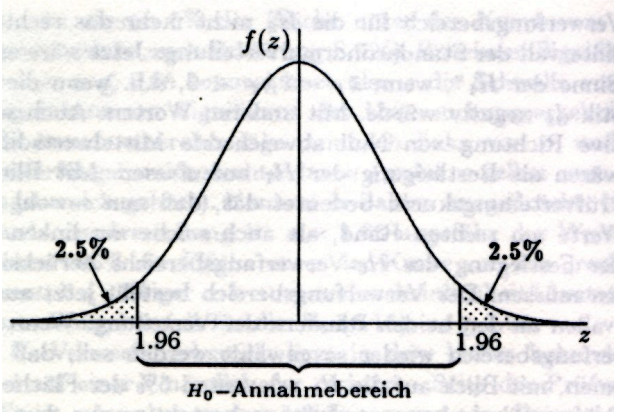
1. define appropriate task (must encourage cooperation)
2. define variables (IV, DV)
3. formulate hypothesis to be tested in terms of variables
4. choose conditions to test; changes in measure are attributed to different conditions; *control* condition without variable manipulation
5. choose how to gather data
6. choose *statistical technique* to test the hypotheses
7. Before you start to do any statistics
   - look at data, check for *outliers*
   - save original data

Choice of statistical test – depends on…

- type of data/variables
  - discrete - can take finite number of values (*levels*)
  - continuous - can take any value
  - ranking scale – interval, nominal, etc.
- type of random experimental variation
  - DVs are subject to random errors
  - do they follow a known probability distribution?
- required information
  - is there a difference between…
    - distributions?
    - frequencies?
    - means?
    - dispersions?
  - correlation of test series?
  - influential factors?
  - how accurate is the estimate?

Analysis - types of test

- *parametric*
  - powerful
  - assume normal distribution of DV
  - robust (give reasonable results also when data not exactly normal)
  - Example: completion time of complex task depends on *independent* subtasks



- *non-parametric*
  - less powerful, more reliable
  - do not assume normal distribution
  - Example: subjective usability rating

- *contingency table*
  - classify data by discrete attributes
  - count number of data items in each group

	Relevant R	Not Relevant $\tilde{R}$
Retrieved G	$G \cap R$	$G \cap \tilde{R}$
Not Retrieved $\tilde{G}$	$\tilde{G} \cap R$	$\tilde{G} \cap \tilde{R}$

Statistical test by form of IV and DV

	IV	*DV*	*Test*
Parametric	Two-valued	Normal	Student‘s t-test on difference of means
	Discrete	Normal	ANOVA (ANalysis Of VAriance)
	Continuous	Normal	(Non-)linear regression factor analysis
Non-parametric	Two-valued	Cont.	Wilcoxon/Mann-Whitney rank-sum test
	Discrete	Cont.	Rank-sum versions of ANOVA
	Continuous	Cont.s	Spearman‘s rank correlation
Contin-gency test	Two-valued	Discrete	No special test, see next entry
	Discrete	Discrete	Contingency table and Chi-squared test
	Continuous	Discrete	Group indep. Variable and then as above

Summary

- Use statistics to describe experimental data and to test hypotheses on them.
- Statistics can be (roughly) divided in: *descriptive statistics a*nd *inferential statistics*
- Methods are standardized – in science, everybody knows what you want to say
- Methods, especially of inferential statistics, are not quite easily applied; some experience needed, read text books!
- Make sure the statistical test you are using is applicable, check its requirements!
- Use software for analyzing the data

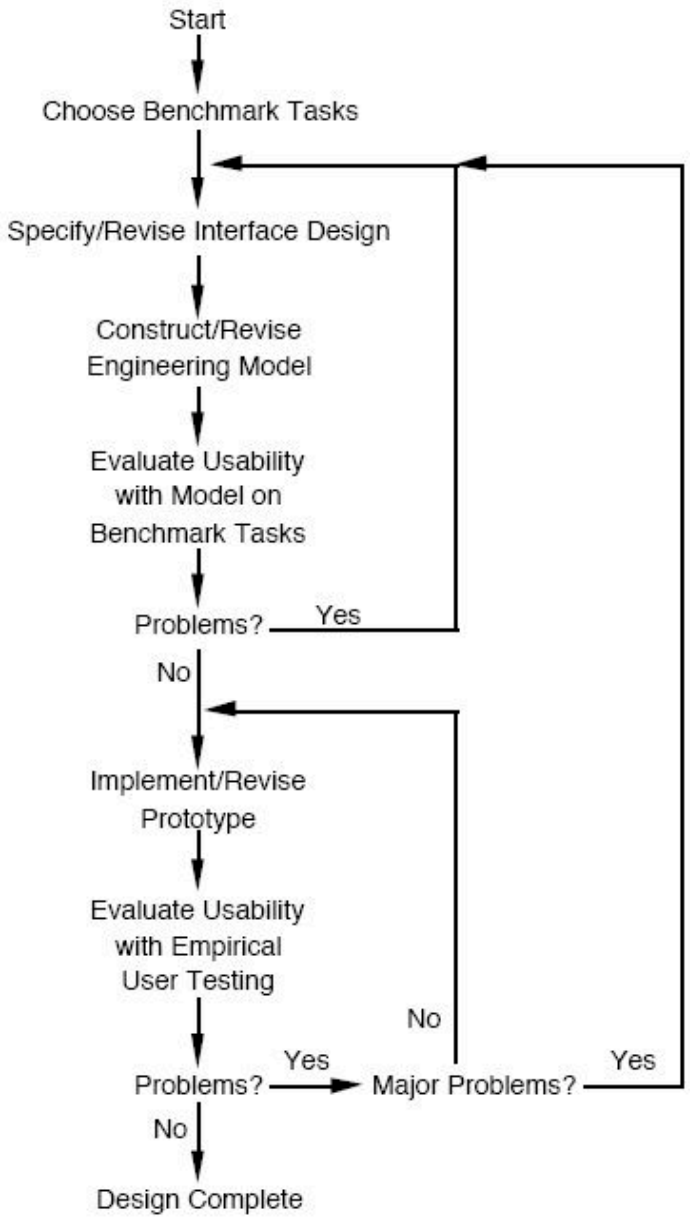
Model-based evaluation

Model-based evaluation

Four steps:

1. Describe interface design in detail
2. Build model of user doing a task
3. Use the model to predict execution or learning time
4. Revise or choose design depending on prediction

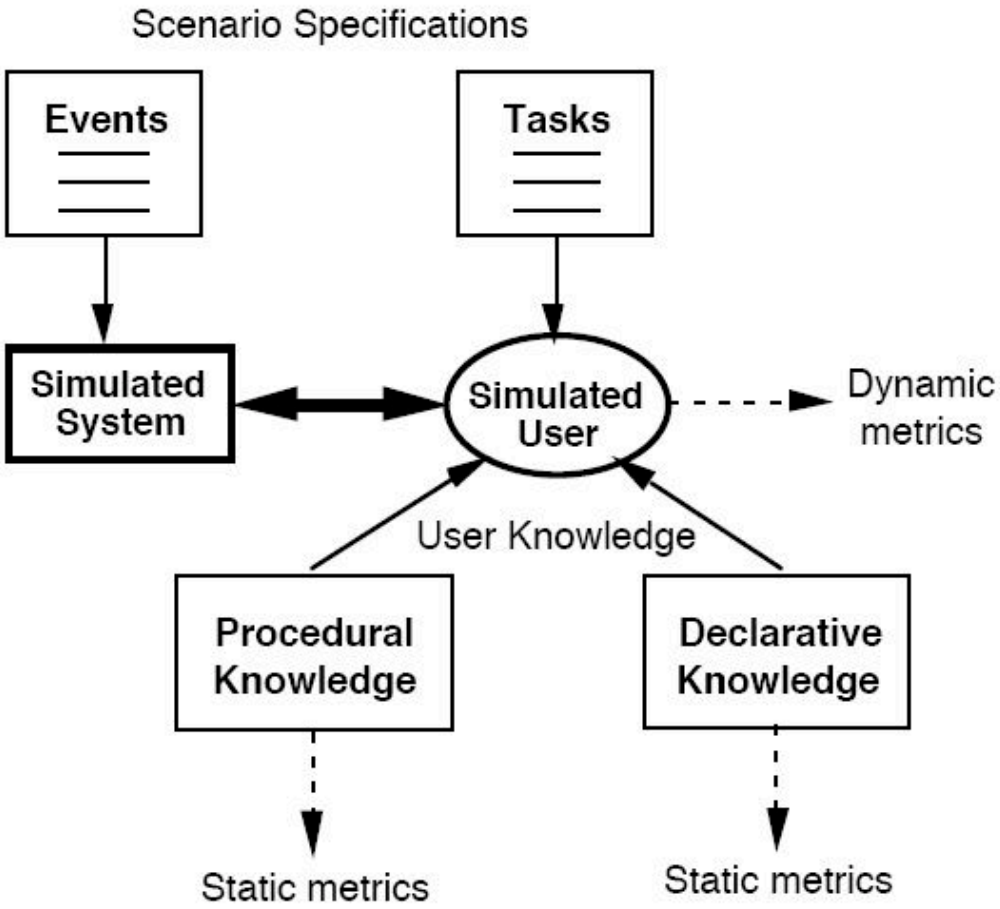
- Usability results *before* implementing prototype or user testing
- Engineering model allows more design iterations



Model-based approach

- Model summarizes the interface design from the user's point of view:
  - Represents how the user gets things done with the system.
  - Components of model can be reused to represent design of related interfaces.
- *But*, current models can only predict a few aspects:
  - Time required to execute specific tasks.
  - Ease of learning of procedures, consistency effects
- User testing still required!

Overview



Models = simulations of human-computer interaction

Procedural knowledge: how-to procedures → executable

Declarative knowledge: facts, beliefs → reportable

Psychological constraints

- Evaluation of a proposed design must be a *routine* activity, not a scientific research project.
- Need to be able to build models without inventing psychological theory.

- Modeling system must provide human psychological constraints *automatically*
  - Constrain what the model can do, so modeler can focus on design questions, not psychological basics
  - If model can be programmed to do any task at any speed or accuracy, something's wrong!

Cognitive vs perceptual-motor constraints

- What dominates a task?
  - Heavily cognitive tasks: Human "thinks" most of the time, e.g. stock trading system

- Many HCI tasks dominated by *perceptual-motor activity*
  - A steady flow of physical interaction between human and computer („doing rather than thinking")
  - Time required depends on human characteristics and computer's behavior (determined by the design)
- Implication
  - Modeling perceptual-motor aspects is often practical, useful, and relatively easy.
  - Modeling purely cognitive aspects of complex tasks is often difficult, open-ended, and requires research resources.

Modeling approaches

Three current approaches:

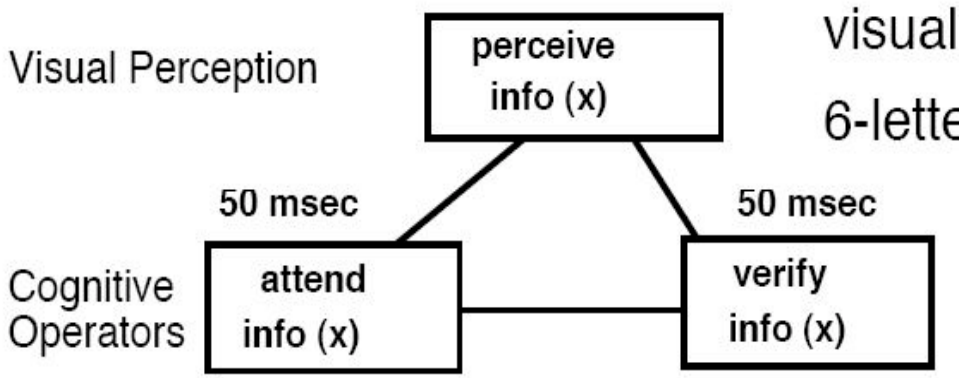
1. Task network models – before detailed design
2. Cognitive Architecture Models – packaged constraints
3. GOMS models – relatively simple & effective

Differ in constraints, detail, when to use.

Task Network Models

- Connected network of tasks:
  - Connection: one task is a prerequisite of the other
  - Both serial and parallel execution of tasks
  - Final completion time computed from chain of serial and parallel tasks
  - *Critical path* = chain with largest execution time
  - PERT charts (*Program Evaluation & Review Techn.*), (E) TAGs
- Tasks = mixture of human and machine tasks
- Each task characterized by a distribution of completion times, and arbitrary dependencies and effects

Task network - example



Cognitive architectures

Represent basic human abilities and limitations.

"Programmed" with a strategy to perform specific tasks.

- provides constraints on the form and content of the strategy.
- Architecture + specific strategy = a model of a specific task.

To model a specific task:

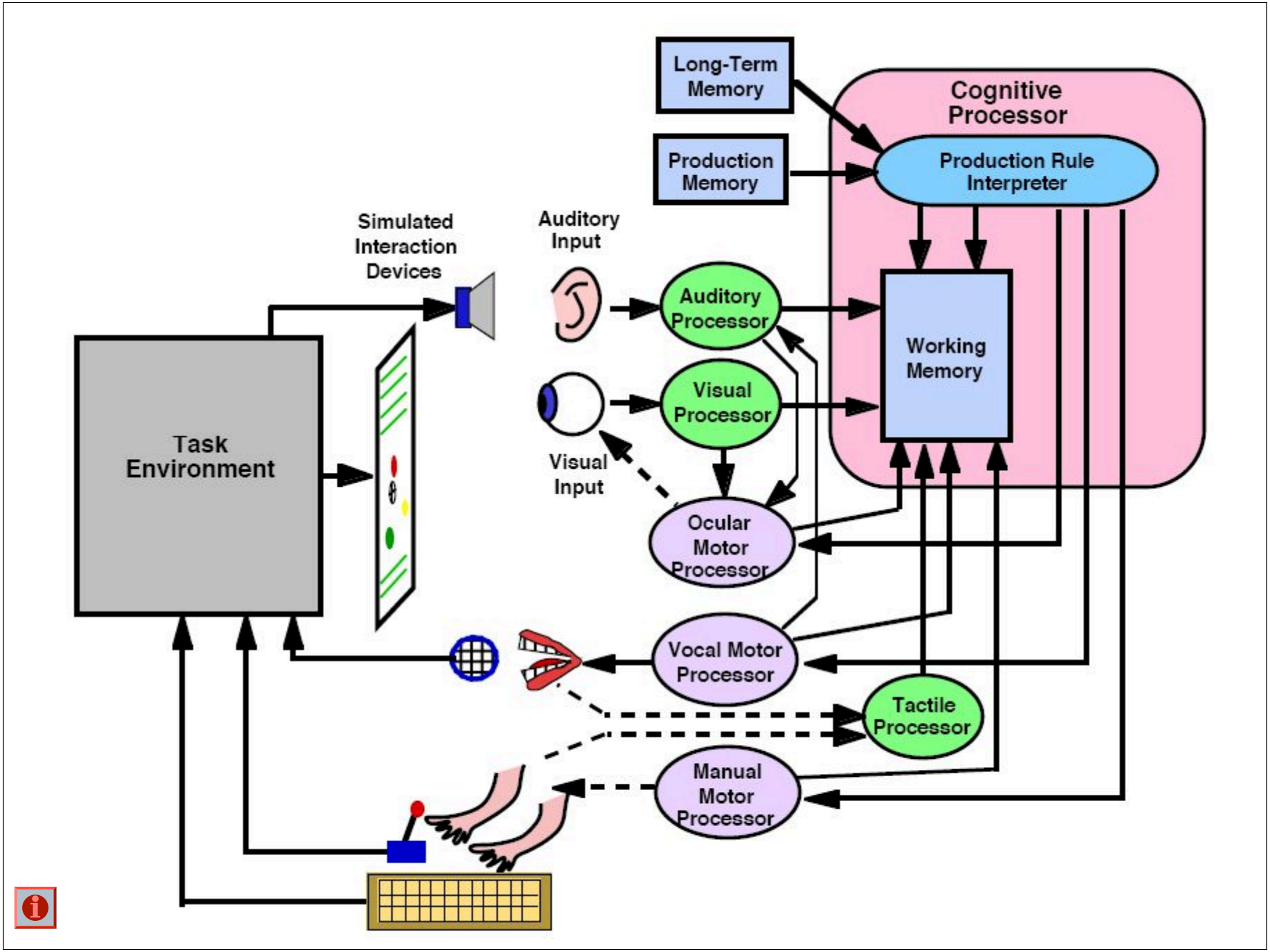
- Do a task analysis to arrive at human's strategy for the task.
- "Program" the architecture with representation of strategy.
- Run the model using task scenarios.

Result: predicted behavior and time course for that scenario and task strategy.

Needs comprehensive psychological theory, so these are quite complex; used mostly in a research settings

Example: EPIC Architecture

- Developed to represent executive processes that control other processes during *multiple task* performance.
- **E**xecutive-**P**rocess **I**nteractive ***C***ontrol (Kieras & Meyer, mid-1990s)
- Basic assumptions
  - Production-rule cognitive processor.
  - Parallel perceptual and motor processors.
- Fixed architectural properties
  - Components, pathways, and most time parameters
- Task-dependent properties
  - Cognitive processor production rules (strategy).
  - Perceptual recoding.
  - Response requirements and styles.


Long-Term Memory
Cognitive Processor
Production Memory
Production Rule Interpreter
Simulated Interaction Devices
Auditory Input
Auditory Processor
Working Memory
Visual Processor
Task Environment
Visual Input
Ocular Motor Processor
Vocal Motor Processor
Tactile Processor
Manual Motor Processor

GOMS (Card, Moran, & Newell, 1983)

- A key model-based methodology based on simplified cognitive architectures.
- An approach to describing the knowledge of *procedures* that a user must have in order to operate a system
  - ***G***oals - what goals can be accomplished with the system
  - ***O***perators - what basic actions can be performed
  - ***M***ethods - what sequences of operators can be used to accomplish each goal
  - ***S***election Rules - which method should be used to accomplish a goal
- Well worked out, quite practical, but limited due to simplifications
- Often in the "sweet spot" - lots of value for modest modeling effort

Keystroke-level model

1. Choose one or more representative task scenarios.
2. Have design specified to the point that keystroke-level actions can be listed.
3. List the keystroke-level actions (operators) involved in doing the task.
4. Insert mental operators for when user has to stop and think.
5. Look up the standard execution time to each operator.
6. Add up the execution times for the operators.
7. The total is the estimated time to complete the task (sum of times for tasks t_i multiplied by frequency n_i)

$$T_{execute} = \sum_i t_i * n_i$$

KLM – operators and times

K - Keystroke (.12 - 1.2 sec; use .28 sec for ordinary user).

- Pressing a key or button on the keyboard.
- Different experience levels have different times.
- Pressing SHIFT or CONTROL key is a separate keystroke.
- Use type operator T(n) for a series of n Ks done as a unit.

P - Point with mouse to a target on the display.

- Follows Fitts' law - use if possible: $0.1 * \log_2 (D/S + 0.5)$
- Typically ranges from .8 to 1.5 sec, average (text editing) is 1.1 sec.

B - Press/release mouse button (.1 sec; click is .2).

- Highly practiced, simple reaction.

KLM – operators and times

H - Home hands to keyboard or mouse (.4 sec).

W - Wait for system response.

- Only when user is idle because can not continue
- Have to estimate from system behavior
- Often essentially zero in modern systems

M - Mental act of thinking.

- Represents pauses for routine activity (not problem-solving).
- New users often pause to remember or verify every step.
- Experienced users pause and think only when logically necessary.
- Estimates ranges from .6 to 1.35 sec; 1.2 sec is good single value.

Example: file deletion in MacOS, original design, experienced user

General procedure

- Find the file icon to be deleted and drag it to the trash can.

Assumptions:

- user thinks of selecting+dragging icon as a single operation.
- Finding to-be-deleted icon is still required
- Moving icons to the trash can is highly practiced:
  - The trash can does not have to be located, so finding the trash can is overlapped with pointing to it.
  - Verifying that the trash can has been hit is overlapped with pointing to it.
  - Final result (bulging can) is not checked since it is redundant with verifying that the can has been hit.

Operator sequence:

initiate the deletion **M,** find the file icon **M,** point to file icon **P,** press and hold mouse button **B,** drag file icon to trash can icon **P,** release mouse button **B,** point to original window **P**

- **Total time = 3P + 2B + 2M = 5.9 sec**

Example: command key file deletion, experienced user

General procedure

- Select the file icon to be deleted and hit a command key.

Assumptions

- User operates both mouse + key with right hand.
- Right hand starts and ends on the mouse.

Operator sequence: initiate the deletion **M,** find the icon for the to-be-deleted file **M,** point to file icon **P,** click mouse button **BB,** move hand to keyboard **H,** hit command key **KK,** move hand back to mouse **H**

- **Total time = P + 2B + 2H + 2K + 2M = 5.06 sec**

Only slightly faster, due to need to move the hand!

Other models in GOMS family

- Critical-Path Method GOMS (CPM-GOMS)
  - Express activities in terms of Model Human Processor → task network → analyze for critical path
- Natural GOMS Language (NGOMSL)/ Cognitive Complexity Theory (CCL)
  - basic GOMS concept as simple production system
  - hierarchical actions as sequential/hierarchical rules, eventually keystroke level operators
- Executable GOMS Language (GOMSL)/GLEAN
  - Formalized and executable version of NGOMSL.
  - *GLEAN* **-** a simplified version of the EPIC simulation system (***G***OMS ***L***anguage ***E***valuation and ***An***alysis)

Model-based vs. inspection evaluation

	Cognitive walkthrough	*Heuristic evaluation*	*Model-based*
Stage	Throughout	Throughout	Design
Style	Lab	Lab	Lab
Objective?	No	No	Somewhat
Measure	Qualitative	Qualitative	Qual. & Quan.
Information	Low level	High level	Low level
Immediacy	N/A	N/A	N/A
Intrusive?	No	No	No
Time	Medium	Low	Medium
Equipment	Low	Low	Low
Expertise	High	Medium	High