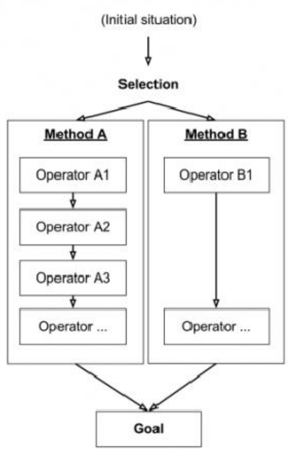
Chapter 5:

Models of Human Computer Interaction

Overview

“I work with **models**“

OTHERS ME



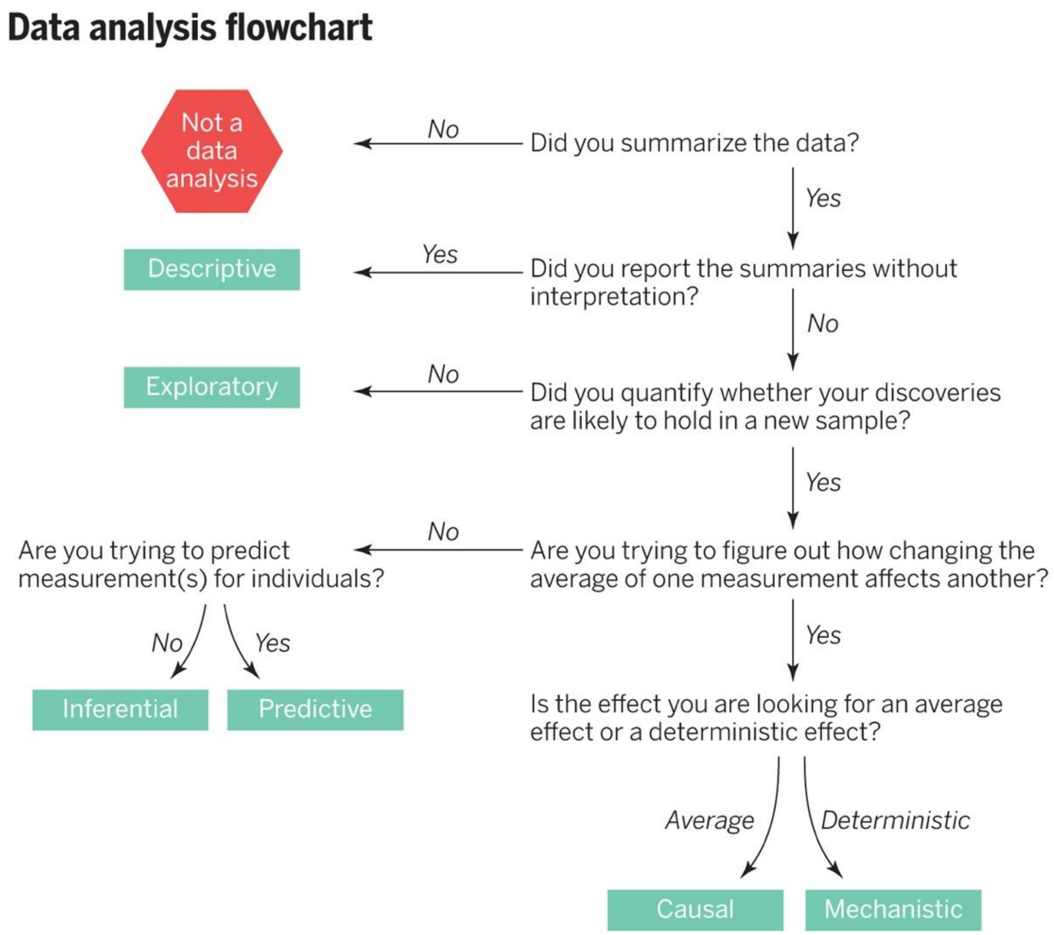
What are descriptive models?

A descriptive model describes a system or other entity and its relationship to its environment. With descriptive models, it is possible to describe real-world events and the relationships between factors responsible for them.

Descriptive models:

- Provide a basis for understanding, reflecting, and reasoning about certain facts and interactions
- Provide a conceptual framework that simplifies a, potentially real, system
- Are used to inspect an idea or a system and make statements about their probable characteristics
- Used to reflect on a certain subject
- Can reveal flaws in the design and style of interaction

To understand which data analysis you perform, you can look at a chart created by Roger D. Peng and Jeff Leek[1]:



Further Examples are:

- Descriptions, statistics, performance measurements
- Taxonomies, user categories, interaction categories

GOMS = Goals, Operators, Methods, Selection rules

GOMS is an "engineering model" of human-computer interaction from the field of "cognitive modelling". The subject of cognitive modelling is the development of concepts, methods and tools for the analysis, modelling, and design of complex human-machine systems in which human information processing at higher cognitive levels plays a special role.

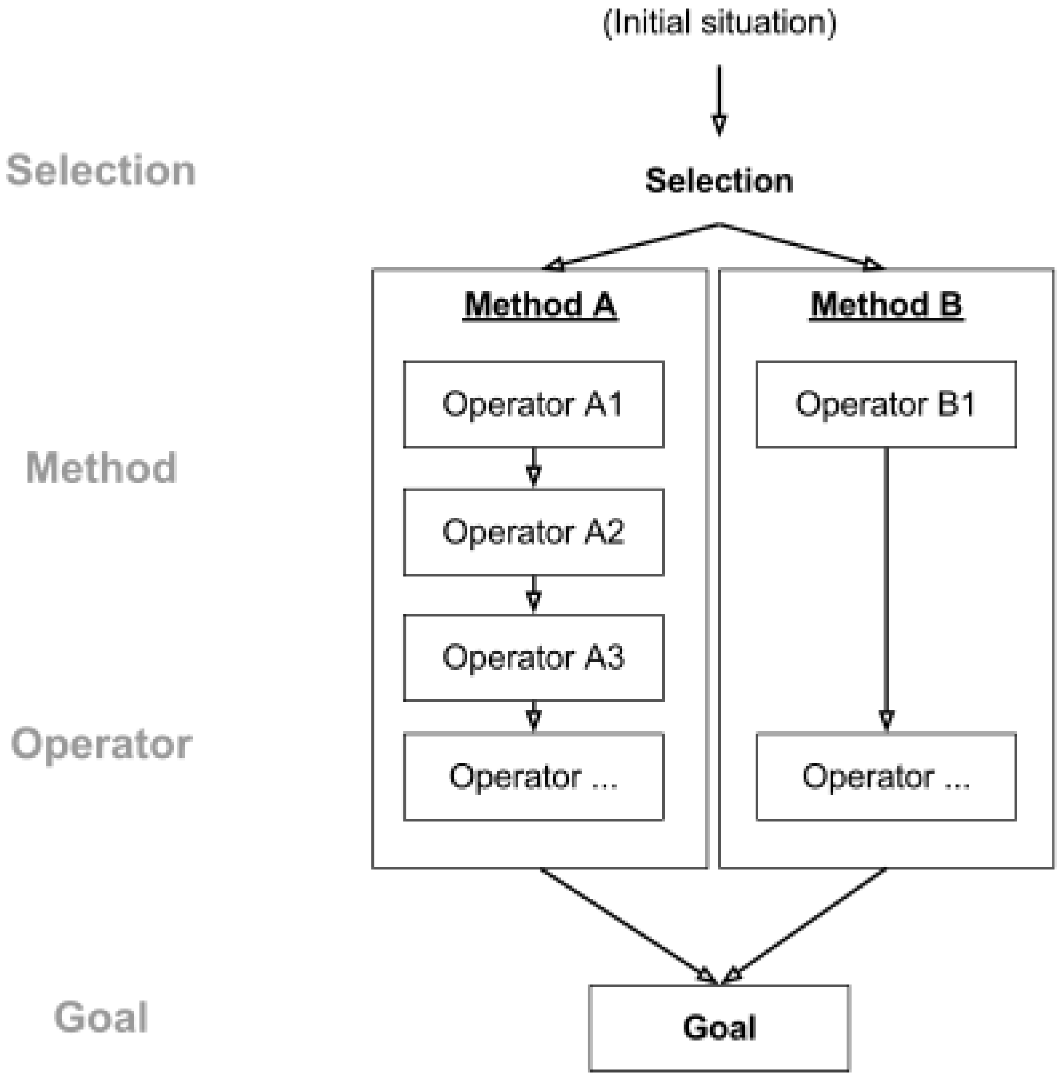
The GOMS model describes a user's cognitive structure on the following four components:

Goals	**Goals** are the goals the user wants to achieve by using a system. These should not be too abstract, creative, or problem-solving (e.g., performing "cut and paste" in a word processor).
Operators	**Operators** are the smallest, atomic actions of the user. These are definable at various levels of abstraction, but there is a tendency towards concrete actions. The operators have a context-free, parameterizable duration (e.g., a mouse click is assigned an execution time of 200msec)
Methods	**Methods** here are learned sequences of subgoals and operators to achieve a specific goal (e.g., mark a paragraph: 1. move the mouse to the beginning of the paragraph 2. press and hold the mouse button 3. move the mouse to the end of the paragraph → the method consists of 3 operators).
Selection rules	Under certain circumstances, several methods are possible for achieving a particular goal. For example, a paragraph can also be deleted character by character (cf. method above). Personal, individual selection rules of the user decide here, based on other factors, which method is used.

User tasks are split into goals which are achieved by solving sub-goals in a divide-and-conquer fashion

Many HCI methods exploit a mental processing model in that the user achieves goals by solving subgoals in a divide-and-conquer manner. This "simulated" human problem-solving process is also modelled in GOMS by decomposing the task into a hierarchy of goals.

Following the Model Human Processor, the operators represent elementary perceptual, cognitive, or motor acts, each of which can be assigned mean execution times.



The overall goal for User Experience Design of the GOMS model is to eliminate useless or unnecessary interactions and improve human-computer interaction efficiency.

Principles of GOMS

To improve the performance of a cognitive skill, eliminate unnecessary operators from the method used to do the task (or use other methods).	The operators involved in cognitive skills are highly specific to the methods used for a given task.	Task performance can be improved by providing a set of error-recovery methods.

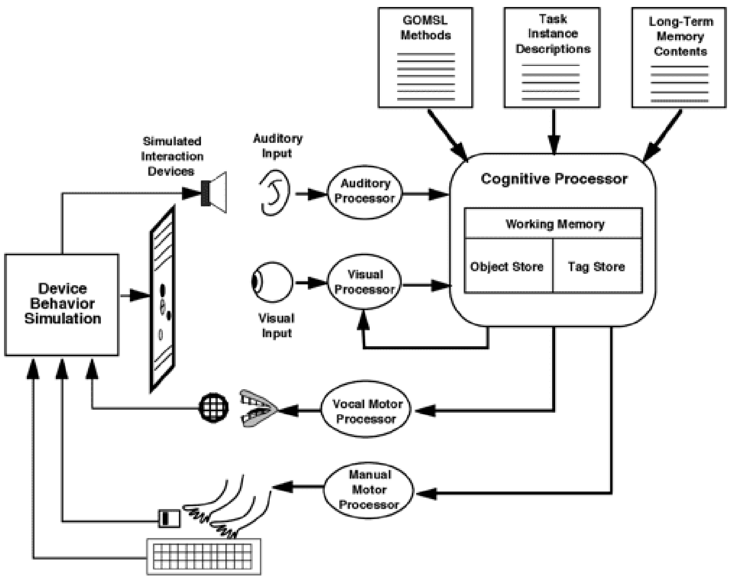
GOMS Example: Closing a Window

GOAL: CLOSE-WINDOW

[select GOAL: USE-MENU-METHOD
MOVE-MOUSE-TO-FILE-MENU
PULL-DOWN-FILE-MENU
CLICK-OVER-CLOSE-OPTION
GOAL: USE-ALT-F4-METHOD
HOLD-ALT-KEY
PRESS-F4-KEY]
VERIFY-CLOSE

For a particular user:
Rule 1: Select USE-MENU-METHOD unless another rule applies
Rule 2: If the application is GAME, select CATLRLT-F4-METHOD

Remember: The Model Human Processor models how a series of information flows in a human from the viewpoint of information processing.



Example of the connection between Model Human Processor and the GOMS model in early research [3].

There are four different types of GOMS concepts: **CMN-GOMS**, **Keystroke-Level Model** (KLM), **NGOMSL**, and **CPM-GOMS**. Although all these concepts give predictive information about how individuals use computer systems, they all highlight different parts of the task completion process. In choosing a GOMS concept to assess a design, one must know "*the type of task the users will be engaged in and the types of information gained by applying the technique*"[2].

The following chapter introduces the Keystroke-Level Model in further detail.

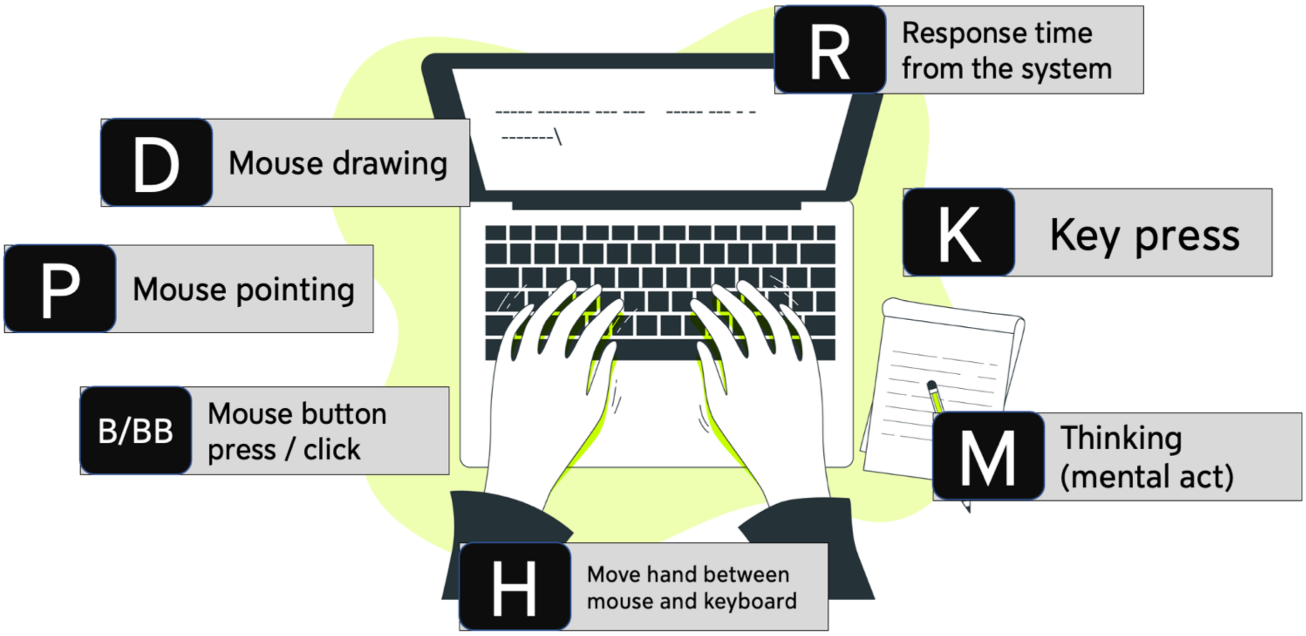
Keystroke-Level Model (KLM)

The KLM is a simplified version of GOMS.

By using the KLM, predictions can be made about the **execution time** of a task. Here, the analyst lists the operator sequence of the task and obtains the predicted execution time of the task by adding up the individual operator times. A KLM model is represented in a sequence form, i.e., like a special program run "on key level". No targets, methods and selection rules are specified.

→ Only operators on a keystroke level
→ No sub-goals
→ No methods
→ No selection rules

There are several KLM operators in this model:



Each operator is assigned a duration:

Operator	Description	Associated Time
K	Keystroke, typing one letter, number, etc. or function key like 'CTRL', 'SHIFT'	Expert typist (90 wpm): **0.12 sec** Average skilled typist (55 wpm): 0.20 sec Average non-secretarial typist (40 wpm): 0 .28 sec Worst typist (unfamiliar with keyboard): 1.2 sec
H	'Homing', moving the hand between mouse and keyboard	**0.4 sec**
B / BB	Pressing / clicking a mouse button	**0.1 sec** / **2*0.1 sec**
P	Pointing with the mouse to a target	**0.8** to **1.5 sec** with an average of **1.1 sec** Can also use be calculated using Fitts' Law
$D(n_D, l_D)$	Drawing n_D straight line segments of length l_D	$0.9*n_D + 0.16*l_D$
M	Subsumed time for mental acts; sometimes used as 'look-at'	**1.35 sec** (1.2 sec according to [Olson and Olson 1995])
R(t) or W(t)	System response (or 'work') time, time during which the user cannot act	Dependent on the system, to be determined on a system-by-system basis

Reasons to use KLM:

- KLM can help **evaluate** UI designs, interaction methods and trade-offs
- If common tasks take too long or consist of too many statements, **shortcuts** can be provided
- Predictions are mostly **remarkably accurate**: +/- 20%
- Sometimes already the comparison of the number of occurrences of the different operators for different designs reveal the difference
- Extensions for novel (mobile, automotive, touch) interfaces exist (see references)

Limitations of the KLM model:

- Only measures one aspect of performance: **time** (= execution time, not the time to acquire or learn a task)
- Only considers **expert users** (there is a broad variance of digital literacy and knowledge)
- Only considers **routine unit tasks**
- The method needs to be specified **step by step**.
- The execution of the method must be **error-free**
- The mental operator aggregates different mental operations and therefore cannot model a deeper representation of the user's mental operations. If this is crucial, a GOMS model has to be used (e.g. model K2)

Summary

Like any tool for measuring human behavior, implementation of GOMS has its advantages and disadvantages. First, it gives both **qualitative and quantitative measurements**, which can provide powerful insight on how users will approach a design. Additionally, because it is a model, it does **not require any actual users**. Often, usability testing can be both expensive and difficult to produce accurate results with physical subjects. Additionally, once GOMS data is dissected and implemented to a change in the design, it is easy to modify the model to create future iterations until the design is optimal.

(CMN-)GOMS	KLM
• Pseudo-code (no formal syntax) • Very flexible • Goals and sub-goals • Methods are informal programs • Selection rules → Tree structure: use different branches for different scenarios • Time consuming to create	• Simplified version of GOMS • Only operators on keystroke-level →Focus on very low-level tasks • No multiple goals • No methods • No selection rules →Strictly sequential • Quick and easy

Problem with GOMS / KLM in general

- Only for well-defined routine cognitive tasks
- Assumes statistical experts
- • Does not consider slips or errors, fatigue, social surroundings, …

The KLM and the GOMS models have in common that they only predict behaviour of experts without errors, but in contrast the KLM needs a specified method to predict the time because it does not predict the method like GOMS [3].

References

1. Leek JT, Peng RD. What is the question? *Science*. 2015;347(6228):1314-1315. doi:10.1126/science.aaa6146

2. John BE, Kieras DE. Using GOMS for user interface design and evaluation: which technique? *ACM Trans Comput-Hum Interact*. 1996;3(4):287-319. doi:10.1145/235833.236050

3. Kieras D. A Guide to GOMS Model Usability Evaluation using GOMSL and GLEAN3. :62.

4. Card SK. *The Psychology of Human-Computer Interaction*. CRC Press; 2018.

Further Resources:

Card S. K., Newell A., Moran T. P. (1983). The Psychology of Human-Computer Interaction. Lawrence Erlbaum Associates Inc.

Card S. K., Moran T. P., Newell A. (1980). The Keystroke-level Model for User Performance Time with Interactive Systems. Communication of the ACM 23(7). 396-410

John, B., Kieras, D. (1996). Using GOMS for user interface design and evaluation: which technique? ACM Transactions on Computer-Human Interaction, 3, 287-319.

D. A. Norman. The Design of Everyday Things. Basic Books. 2002. ISBN: 978-0465067107

B. Shneiderman. Designing the User Interface: Strategies for Effective Human-Computer Interaction , 5th Edition. 2009. ISBN: 978- 0321537355

L. Suchman, Plans and Situated Action: - The Problem of Human- Machine Communication. 1987, ISBN 978-0521337397

Alan Dix, Janet Finlay, Gregory Abowd and Russell Beale. (2003) Human Computer, Interaction (3rd edition), Prentice