from lecture 1:

Perspectives on Users

Evolving understanding of the user and human-computer relationship

First wave

Human Factors/Engineering

Optimal fit between man and machine

-> **The user as Operator**

Cognitive Revolution

More (cognitively) sophisticated human-computer communication to perform tasks

-> **The user as single User**

Second wave

Situated Perspective

Support for situated action in the world

-> **The user as collaborating Actor**

Third wave

Digital technologies as part of everyday environments and practices -> **The user as Consumer**

Cognitive Modelling in HCI

Motivation

Better understanding of **how people carry out computer-based tasks**.

Theories and methods from cognitive psychology were utilized to construct cognitive models.

- Cognitive psychology:
  the study of human perception and information processing.

- Cognitive models:
  specifications of the mental representations, operations and problem-solving strategies that occur during the task execution.

Cognitive Models in HCI

Cognitive Models:

Specifications of the mental representations, operations and problem-solving strategies that occur during the execution of computer-based tasks.

They are employed in (early) design and evaluation of interactive systems, e.g.

- to develop a more precise and detailed understanding of human-computer interaction (e.g., to understand how people perceive and process complex displays of information when carrying out certain tasks),
- to predict task performance and how users will behave,
- to identify and explain the nature of problems that users encounter,
- to provide information concerning the cognitive and perceptual constraints on human performance.

from: Feuerstack et al. “A Tool for Easing the Cognitive Analysis of Design Prototypes of Aircraft Cockpit Instruments”, ECCE 2015.

“Typical questions that have to be answered by HMI [Human-Machine Interaction] designers during the development process of new assistance systems are:

1. What is the ideal position of an instrument?

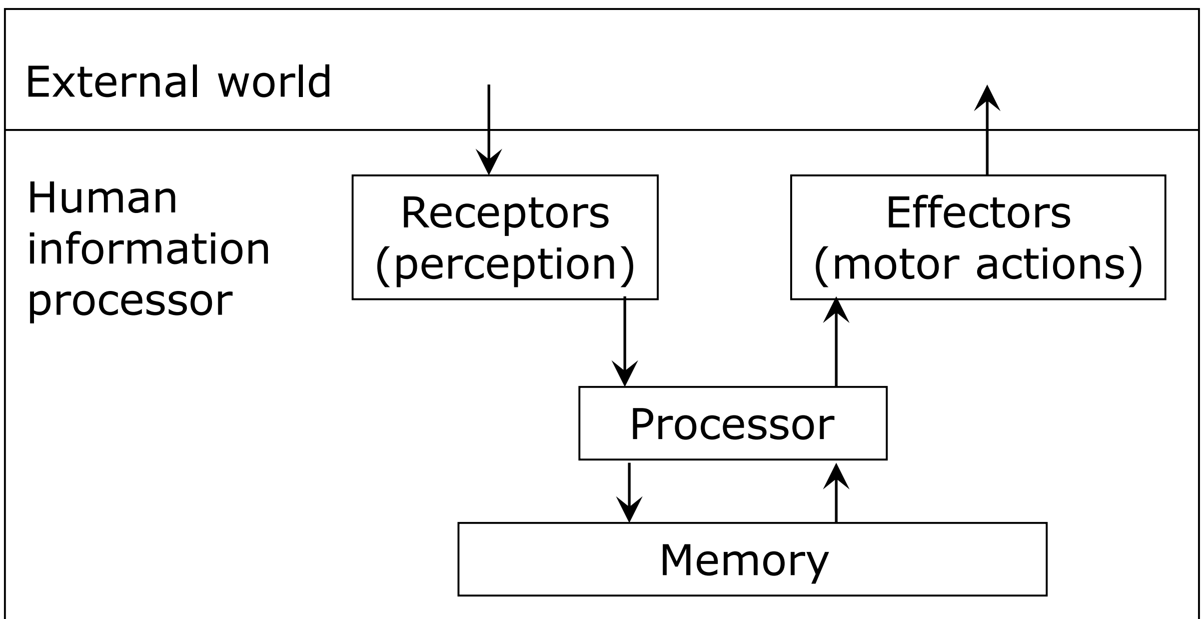
2. How does a new instrument design affect the operators’ task performance?

3. Does the design have an impact on the operators’ workload?”

Human Information Processor (HIP)

- An example of a conceptual framework that forms basis for cognitive modeling in HCI.

- Processor acts on a recognize-act cycle:
  - information is available through the receptors and from internal memory,
  - information is matched against a set of patterns (often represented as productions – if-then rules),
  - match triggers a set of actions (operators).



Assumption:

Cognition is computational in nature.

HIP as Conceptual Framework

- Informally stated assumptions about the structure of human cognition.
- Human cognition and behavior is analyzed in terms of stages (perception – processing – action).

Stages are performed **serially**

(sufficient for describing some tasks such as text editing tasks)

Stages can be performed **in parallel** to a certain extent

(necessary to get a better understanding of tasks such as reading tasks)

Model Human Processor (MHP) as an „Instantiation“ of HIP

(Card, Moran, and Newell in “The Psychology of Human-Computer Interaction”, 1983)

- MHP describes single “theoretical persons”.
- MHP consists of a set of interconnected memories and processors and a set of principles of operation.
- Depicts human cognition with 3 parallel processors:
  - perceptual processor,
  - cognitive processor,
  - motor processor.

MHP: Interacting Subsystems

Each subsystem with processor and possibly memory.

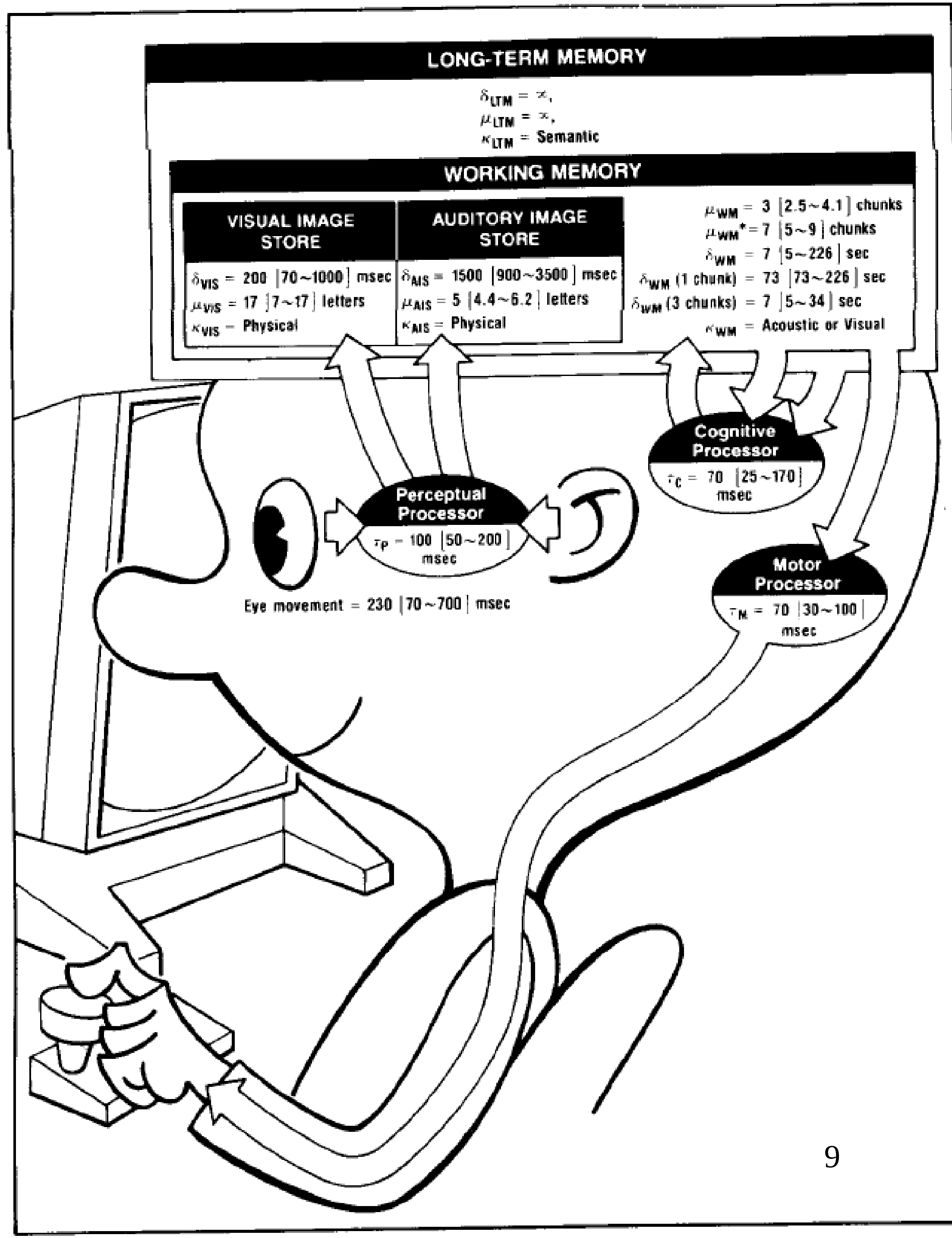
Processor:

- cycle time, includes access time (τ)

Memory parameters:

- storage capacity in items (μ)
- decay time (δ)
- main code type (κ) (physical, acoustic, visual, semantic)

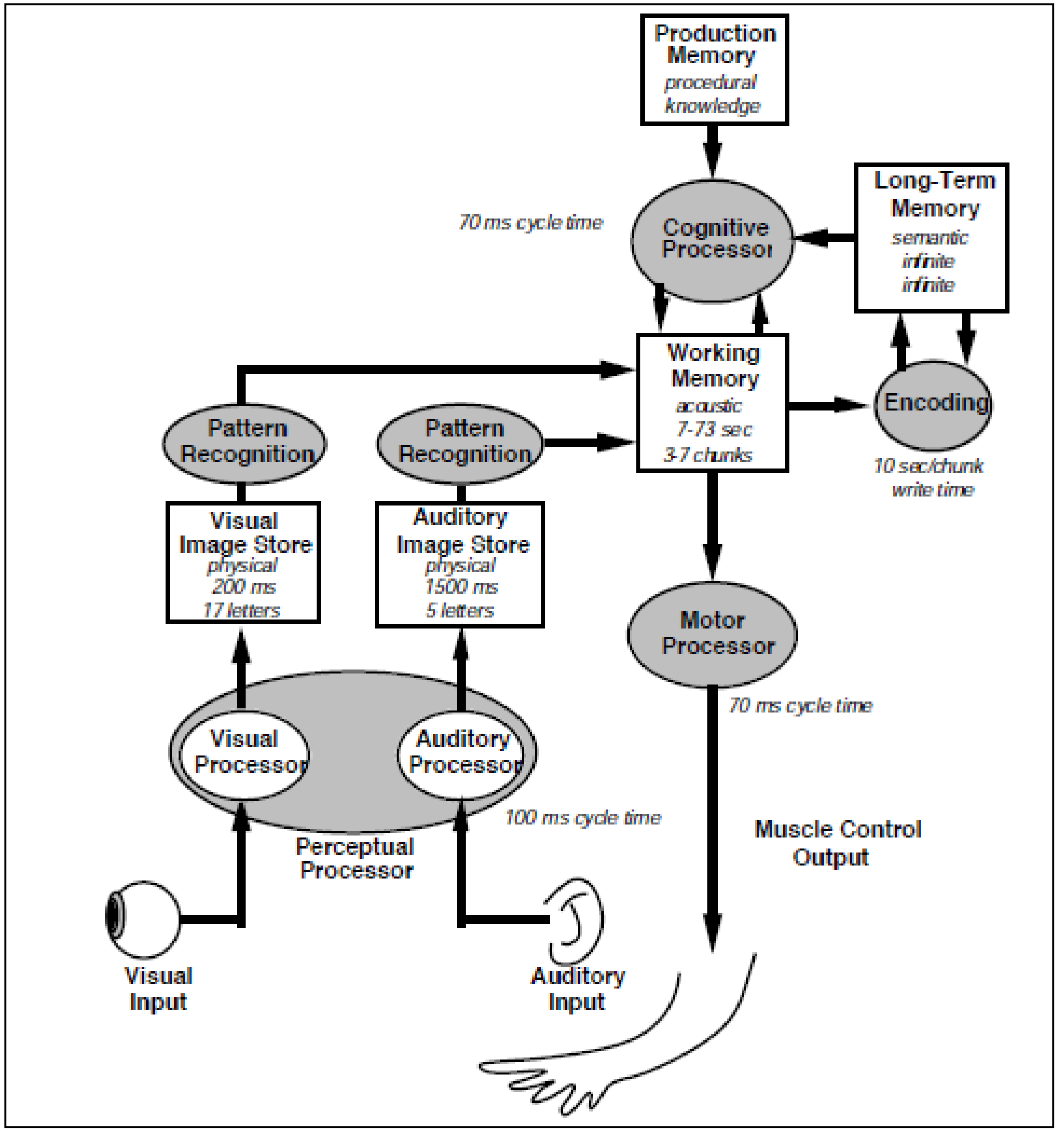
Each subsystem is guided by certain principles of operation.



Perceptual processor:
Transforms sensory input (audio and visual) into representations.

Cognitive processor:
- input from sensory buffers,
- accesses LTM to determine response,
- output response into WM.

Motor processor:
("translates thoughts into actions")
- input from WM,
- carries out response
- cyclic behavior divides output into discrete units.



Modified version by D. Kieras

MHP: Some Principles of Operation

P0 - Recognize-Act Cycle of the Cognitive Processor:
On each cycle of the Cognitive processor, the contents of the Working Memory (WM) initiate actions associatively linked to them in Long-Term Memory; the actions in turn modify the contents of the WM.

P1 – Variable Perceptual Processor Rate Principle:
The Perceptual processor cycle time τ_P varies inversely with stimulus intensity.

P8 – Rationality Principle:
A person acts so as to attain his goals through rational action, given the structure of the task and his inputs of information and bounded by limitations on his knowledge and processing ability:

Goals+Task+Operators+Inputs+Knowledge
+Process-limits → Behavior

P9 – Problem Space Principle:
The rational activity in which people engage to solve a problem can be described in terms of (1) a set of states of knowledge, (2) operators for changing one state into another, (3) constraints on applying operators, and (4) control knowledge for deciding which operator to apply next.

Model Human Processor (MHP): Summary

- People can perform actions in parallel along three dimensions — perceptual, cognitive and motor, but within each of these three dimensions, people must perform actions sequentially.

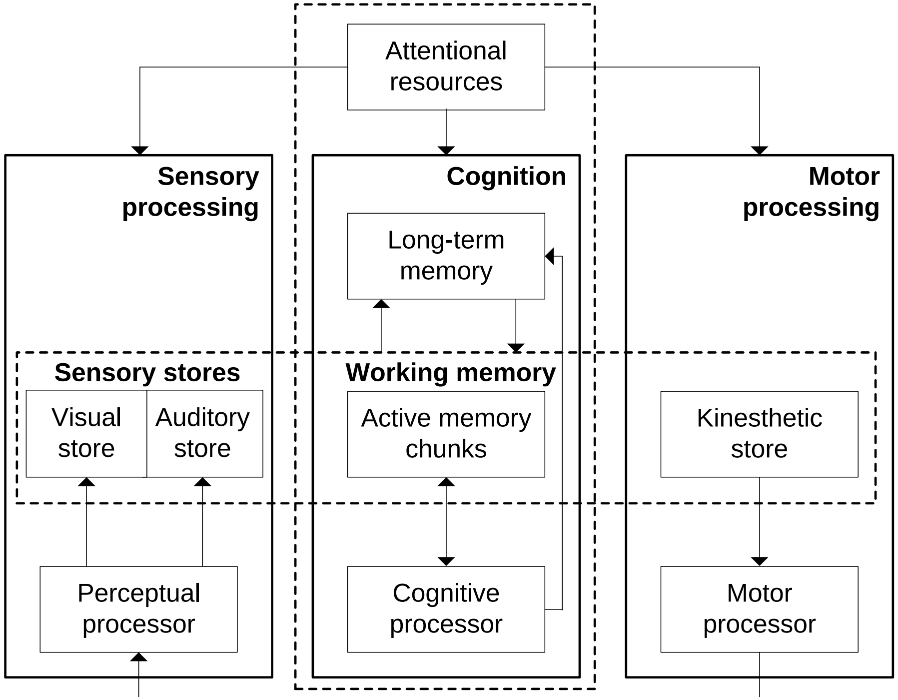
- People discretize tasks cognitively. They hierarchically decompose large tasks into smaller tasks, and they continue this decomposition until subtasks are reduced into indivisible units of work. The size of these basic units of work is related to the cycle time of human’s three cognitive processors.

- People perform large actions by executing chains of smaller discrete actions.

Not considered in MHP

- Haptic processor and memory
- Kinesthetic (or motor) memory
- Emotion
- Attention
  - directed allocation of resources (filters)
  - affects perceptual processing of stimuli

…



Main Approaches in Cognitive Modeling

Engineering models
(predictive models)

- Based on a model of human information processing, on a related task analysis method, on approximation and calculation.
- e.g. **GOMS-family**

(Computational) Cognitive architectures

- Based on theories of cognition, perception and motor control implemented as software systems.
- Simulation of human-computer interaction.
- e.g. ACT, ACT-R, CCT

GOMS as an Example for Engineering Models

GOMS is a method for describing a task and the user's knowledge of how to perform the task in terms of

- **Goals**: the aims of the user when interacting with the computer,
- **Operators**: the possible interactions that the interface allows (e.g. clicking and dragging with a mouse, pressing a key on the keyboard, opening a text editor window),
- **Methods**: sequences of sub-goals and operators that can be used to achieve a particular goal,
- **Selection rules**: by which a user chooses a particular method from a number of alternatives for achieving a goal.

GOMS is a bridge from hierarchical task structures to MHP.

Analyzes execution structure of a task:
- perception of interface (perceptual system)
- mental steps to move from perception to action (cognitive system)
- actions (motor system).

An Example GOMS Model

```
GOAL: Delete Sentence

METHOD: Menu-Method-Delete-Sentence
  1 Highlight Sentence
  2 Open Menu
  3 Select Delete-Command
  4 Accomplish_Goal Menu-Method-Delete-Sentence

METHOD: Del-Key-Method-Delete-Sentence
  1 Position-Cursor At End
  2 Press Delete For Each Letter
  3 Accomplish_Goal Del-Key-Method-Delete-Sentence

SELECTION RULE FOR: Delete Sentence
IF [Long Sentence]
THEN Accomplish_Goal BY Menu-Method-Delete-Sentence

IF [Short Sentence]
THEN Accomplish_Goal BY Del-Key-Method-Delete-Sentence
```

A GOMS model describes a hierarchy of (sub-)goals until the level of operators. Each step in a method is an operator or a sub-goal that can be further decomposed.

A Decomposition in the Example:

```
GOAL: Delete Sentence

METHOD: Menu-Method-Delete-Sentence
  1 Highlight Sentence
  2 OPEN MENU
  3 Select Delete-Command
  4 Accomplish_Goal Menu-Method-Delete-Sentence
…

METHOD: OPEN MENU
  1 Move-Cursor Menu-Bar
  2 Click Menu-Name
  3 Accomplish_Goal OPEN MENU
```

Operators in GOMS

Depend on the interaction style and application domain

- Command-line interface: command & parameters typed on a keyboard,
- GUI: menu selection, button presses, mouse clicks...

More recent work examines, e.g., operators for touch-based interfaces (see examples in tutorial part).

Example: GOMS Operators in Segmentation Tasks

(Ramkumar et al., 2017)

Segmentation of medical images is needed in healthcare to support tasks such as diagnosis, prognosis, and planning of medical interventions (e.g., radiotherapy)

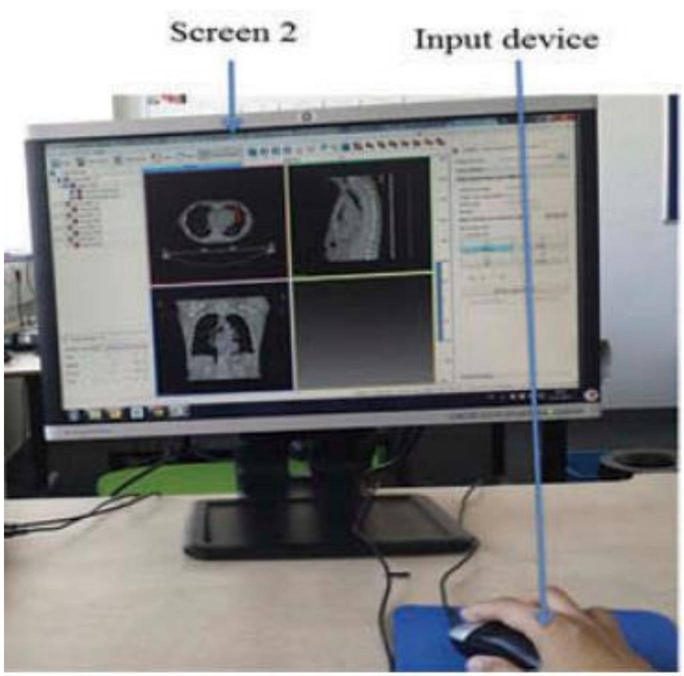


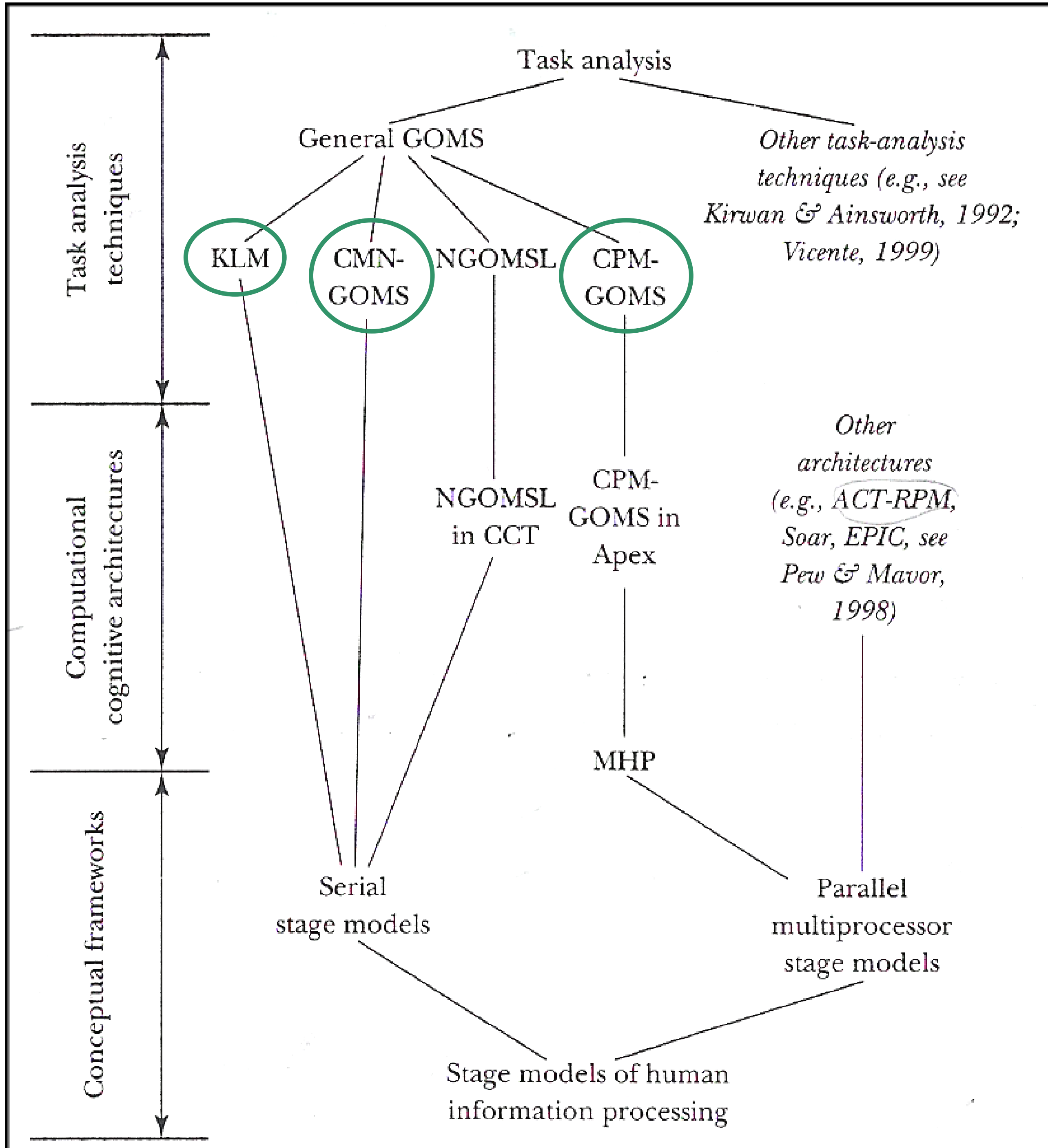
Table 1. GOMS Operators.

No.	Operators	Time (s)	Meaning
1	*Mouse cursor move*	0.9	Moving of the cursor from the drawing region to a panel to select a tool
2	*Zooming*	2	Right mouse button down and move the mouse
3	*Panning*	2	Middle mouse button down and move the mouse
4	*Mouse clicks*	0.2	Left mouse button click
	Click paint		
	Click FG tool		
	Click BG tool		
	Click Add tool		
	Click wipe		
5	*Scrolling time*		Mouse wheel scroll forth and back
	Slow scroll	0.8	Observed during decision making process
	Normal scroll		
	Fast scroll	0.3	Observed when the user wanted to reach the target region
		0.03	Mainly observed while familiarization with the anatomy of the dataset
6	*Drawing time*		Left mouse button down. Drawing time differed between the organs, interaction methods and physicians. Hence there was no fixed time for drawing
	Draw FG		
	Draw BG		
	Draw contour		
7	*Wipe*	2–6	Left mouse button down Observed mainly when the user created mistakes
8	*Adjustment of the brush size*	0.4–2	Observed with the paint tool, mainly when the users shifted between tools

GOMS refers to a whole family of (types of) models

KLM and (CMN-)GOMS only loosely related to MHP (Rationality Principle + Problem Space Principle),

CPM-GOMS based directly on MHP.



Keystroke-Level Model (KLM) as simplest GOMS Variant

Assignment 3

submission deadline: Tuesday, 5.11.2024 at 24:00 PM

Please, watch the introduction to KLM by Niels Henze at
https://www.youtube.com/watch?v=yvlMNKPum38
(approx. 16 min).

Create a Keystroke-Level Model for an own example of a similar complexity as the examples in the video with the objective

a) to predict performance time for a simple task, or

b) to compare two user interfaces with respect to performing simple task(s) efficiently.

Use the operator-time table at 8:30 (in the video) for time prediction and the heuristic rules from next slide.

Heuristic Rules for Placing the M-Operators in KLM

H0: Initial insertion of candidate **M**'s: Insert **M** before all **K**'s and **P**'s.

H1: Deletion of anticipated **M**'s: If **P** or **K** is fully anticipated by a preceding **P** or **K** then delete the middle **M** (e.g. moving the mouse to press the button: **PMK → PK**)

H2: Deletion of **M**'s in cognitive units: If a series of K represent a string then delete the middle M's (e.g. typing 1.2: **MKMKMK → MKKK**)

H3: Deletion of **M**'s before consecutive terminators: If several delimiters are typed only keep the first **M** (e.g., if "))" is the terminator: **MKMK → MKK**)

H4: Deletion of **M**'s that are terminators of commands: If the terminator is a frequently used string, delete the **M** before the terminator (e.g. <command-string><return>: **...KMK → ...KK**)

H5: Deletion of overlapped **M**'s: Do not count any portion of an **M** that overlaps with a command response.

Another Well-Known Variant: CPM-GOMS

- “Classical” GOMS is based only on the serial stage model.

- CPM-GOMS variant based on the parallel multiprocessor stage model of human information processing.

- CPM stands for both

  - **C**ognitive-**P**erceptual-**M**otor analysis of activity

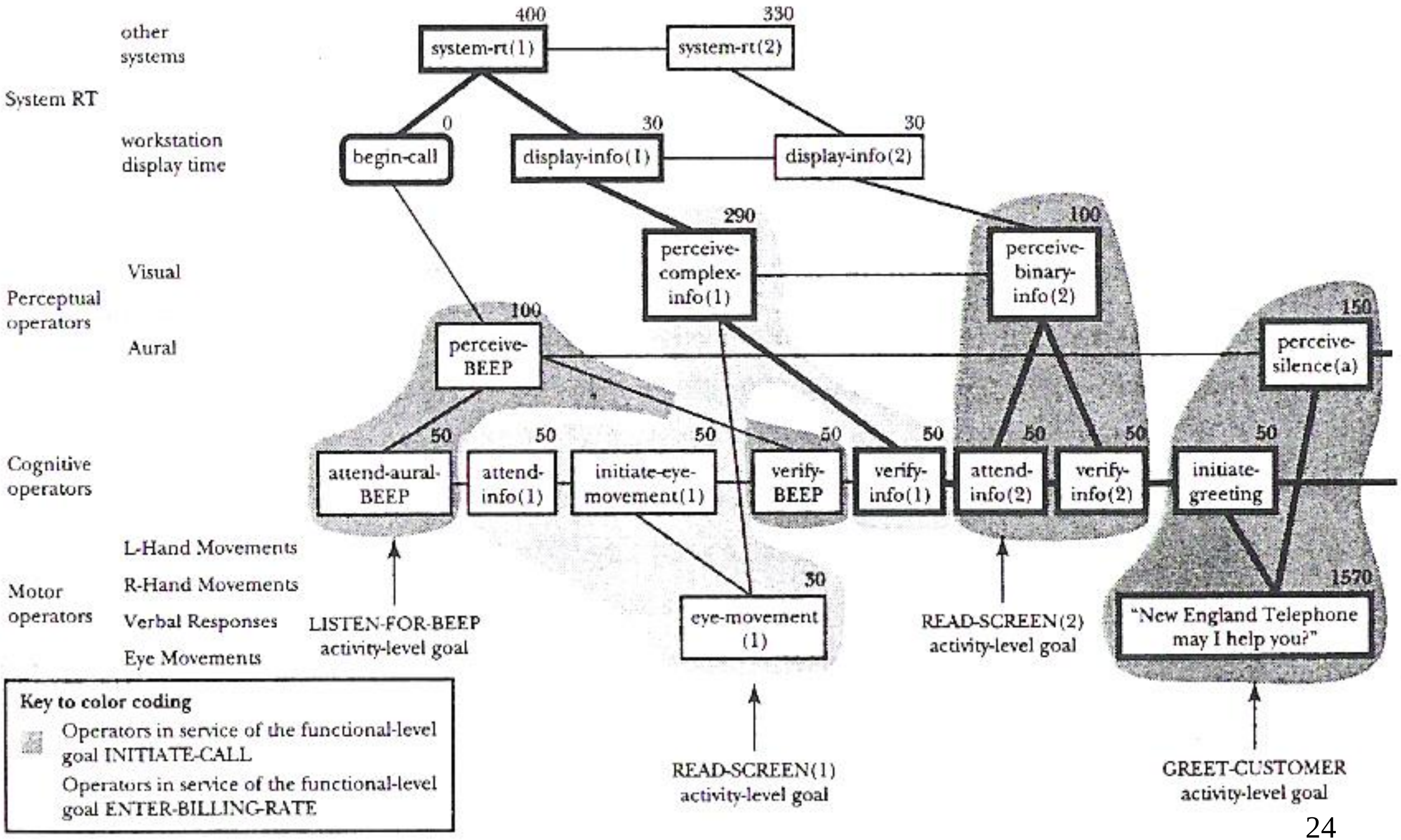
  - and for **C**ritical **P**ath **M**ethod.

- Perceptual, cognitive, and motor operations at the level of MHP cycle times can be performed in parallel as the task demands.

- CPM-GOMS models are represented by schedule charts.

- Selection rules and goal hierarchies are not explicitly represented.

An Impression of a CPM-GOMS Model



How to Develop and Use CPM-GOMS Models?

Illustrated by the case study in (Gray et al., 1993)

- In 1988, NYNEX (a telephone company) considered replacing the workstations used by toll and assistance operators (TAOs) with a new workstation.
- A major factor in making a buy/no-buy decision was how quickly the expected decrease in average worktime per call would offset the capital costs of making the purchase.
- An average decrease of 1 sec in worktime per call saved NYNEX $3 million per year.
- CPM-GOMS was used to predict experts' time
  --- the project was also used to validate GOMS.

Tasks and Environment of Toll and Assistance Operators (TAOs)

- A TAO is a person one gets when dialing 0 (e.g. in New York).
- They assist a customer in completing calls and record the correct billing.
- TAOs handle:
  - person-to-person calls,
  - collect calls,
  - calling-card calls,
  - calls billed to a third number.
- TAOs are trained to answer 3 questions in the course of a call:
  - Who should pay for the call?
  - What billing rate is to use?
  - When is the connection complete enough to terminate interaction with the customer?
- TAOs
  - converse with the customer,
  - key information into a workstation,
  - read information from the workstation screen,
  - write notes to themselves (in some cases).
- TAOs must perform many of these interactions simultaneously.

1.Step: Developing a „classical“ GOMS model for the workstation currently used

A typical dialog

Workstation: “BEEP”

TAO: “New England Telephone, may I help you?”

Customer: “Operator, bill this to 412-555-1212-1234.”

TAO: Keys information into the workstation.

TAO: “Thank you.”

TAO: Releases the workstation to accept the next incoming call by pressing the RELEASE-key.

GOMS goal hierarchy	Observed Behavior
goal: handle-calls	
. **goal: handle-call**	
. . **goal: initiate-call**	
. . . **goal: receive-information**	
. . . . listen-for-beep	Workstation: Beep
. . . . read-screen(2)	Workstation: Displays source information
. . . **goal: request-information**	
. . . . greet-customer	TAO: "New England Telephone, may I help you?"
. . **goal: enter-who-pays**	
. . . **goal: receive-information**	
. . . . listen-to-customer	Customer: Operator, bill this to 412-555-1212-1234
. . . **goal: enter-information**	
. . . . enter-command	TAO: hit F1 key
. . . . enter-calling-card-number	TAO: hit 14 numeric keys
. . **goal: enter-billing-rate**	
. . . **goal: receive-information**	
. . . . read-screen(1)	Workstation: previously displayed source information
. . . **goal: enter-information**	
. . . . enter-command	TAO: hit F2 key
. . **goal: complete-call**	
. . . **goal: request-information**	
. . . . enter-command	TAO: hit F3 key
. . . **goal: receive-information**	
. . . . read-screen(3)	Workstation: displays credit-card authorization
. . . **goal: release-workstation**	
. . . . thank-customer	TAO: "Thank you"
. . . . enter-command	TAO: hit F4 key

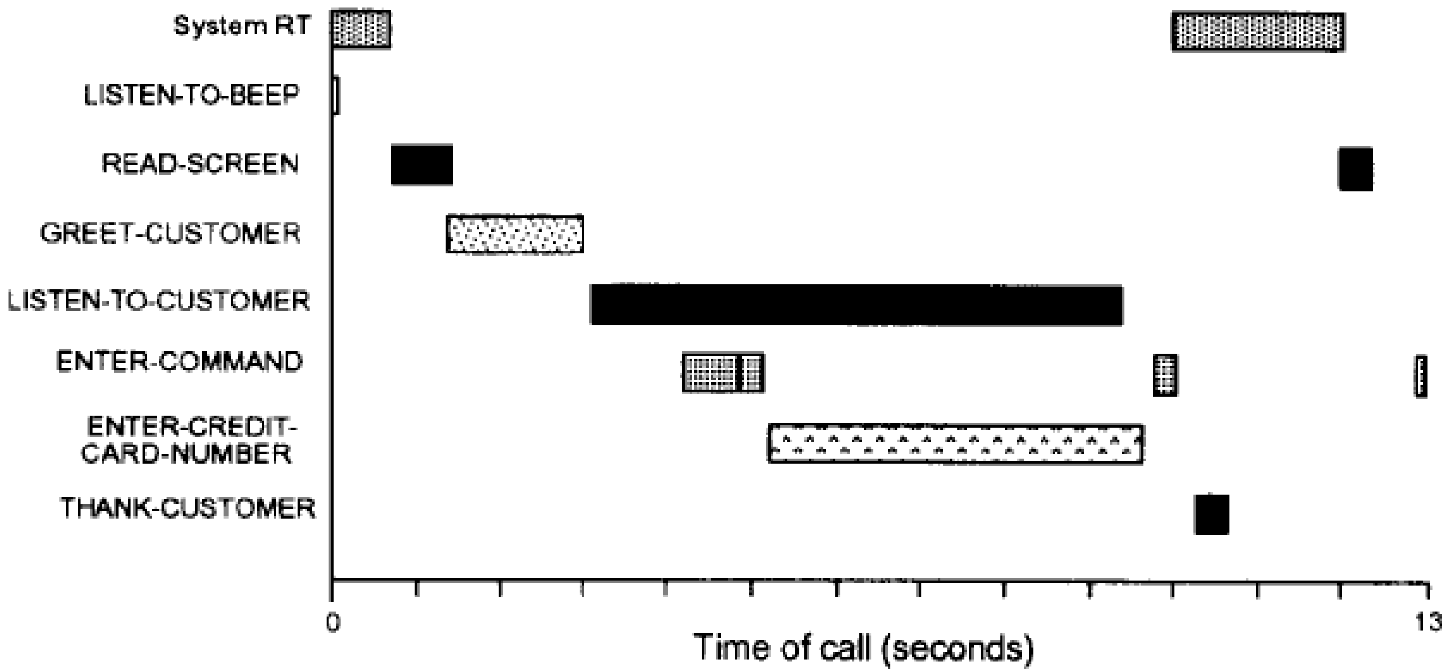
Time Prediction (GOMS Model) and Observed Time

Figure 5. Activity-level prediction of task performance time (total predicted = 17,850 msec; total observed = 13,000 msec; percent error = 37%). Notes: We include observed system response time when it does not overlap with user behavior as per Card, Moran, and Newell (1983). Operators corresponding to observed behavior with more than one occurrence (only the **ENTER-COMMAND** operator) are assigned a duration equal to the average of all observed occurrences; operators with only one occurrence are assigned a duration identical to that occurrence (all other operators set from the videotape). Unobservable operators (only the **READ-SCREEN** operator) are assigned a duration based on prior research, as noted.

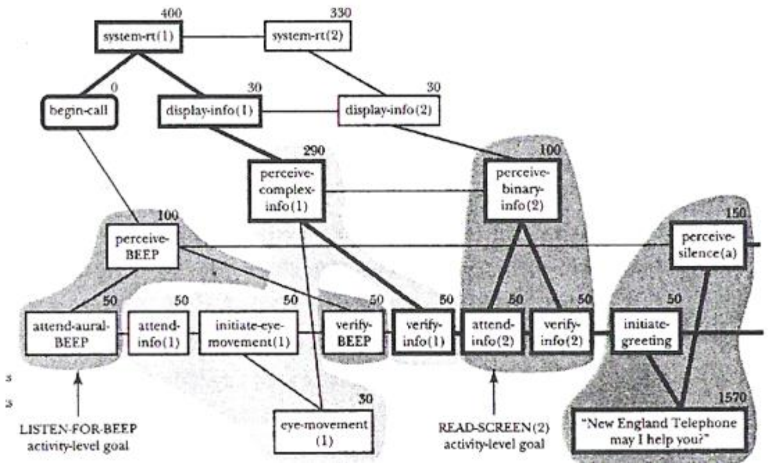
GOMS Model	Duration Estimates	Operator Source of Estimate
GOAL: HANDLE-CALLS		
. GOAL: HANDLE-CALL		
. . GOAL: INITIATE-CALL		
. . . GOAL: RECEIVE-INFORMATION		
. . . . LISTEN-T0-BEEP	100 msec	Videotaped call
(SYSTEM-RESPONSE-TIME)	730 msec	Videotaped call
. . . . READ-SCREEN(2)	340 msec	John and Newell (1987, 1989a)
. . . GOAL: REQUEST-INFORMATION		
. . . . GREET-CUSTOMER	1,570 msec	Videotaped call
. . GOAL: ENTER-WHO-PAYS		
. . . GOAL: RECEIVE-INFORMATION		
. . . . LISTEN-TO-CUSTOMER	6,280 msec	Videotaped call
. . . GOAL: ENTER-INFORMATION		
. . . . ENTER-COMMAND	320 msec	Videotaped call (average)
. . . . ENTER-CALLING-CARD-NUMBER	4,470 msec	Videotaped call
. . GOAL: ENTER-BILLING-RATE		
. . . GOAL: RECEIVE-INFORMATION		
. . . . READ-SCREEN(1)	340 msec	John and Newell (1987, 1989a)
. . . GOAL: ENTER-INFORMATION		
. . . . ENTER-COMMAND	320 msec	Videotaped call (average)

…

Reason for the Time Difference
(17,85 vs. 13 sec.)



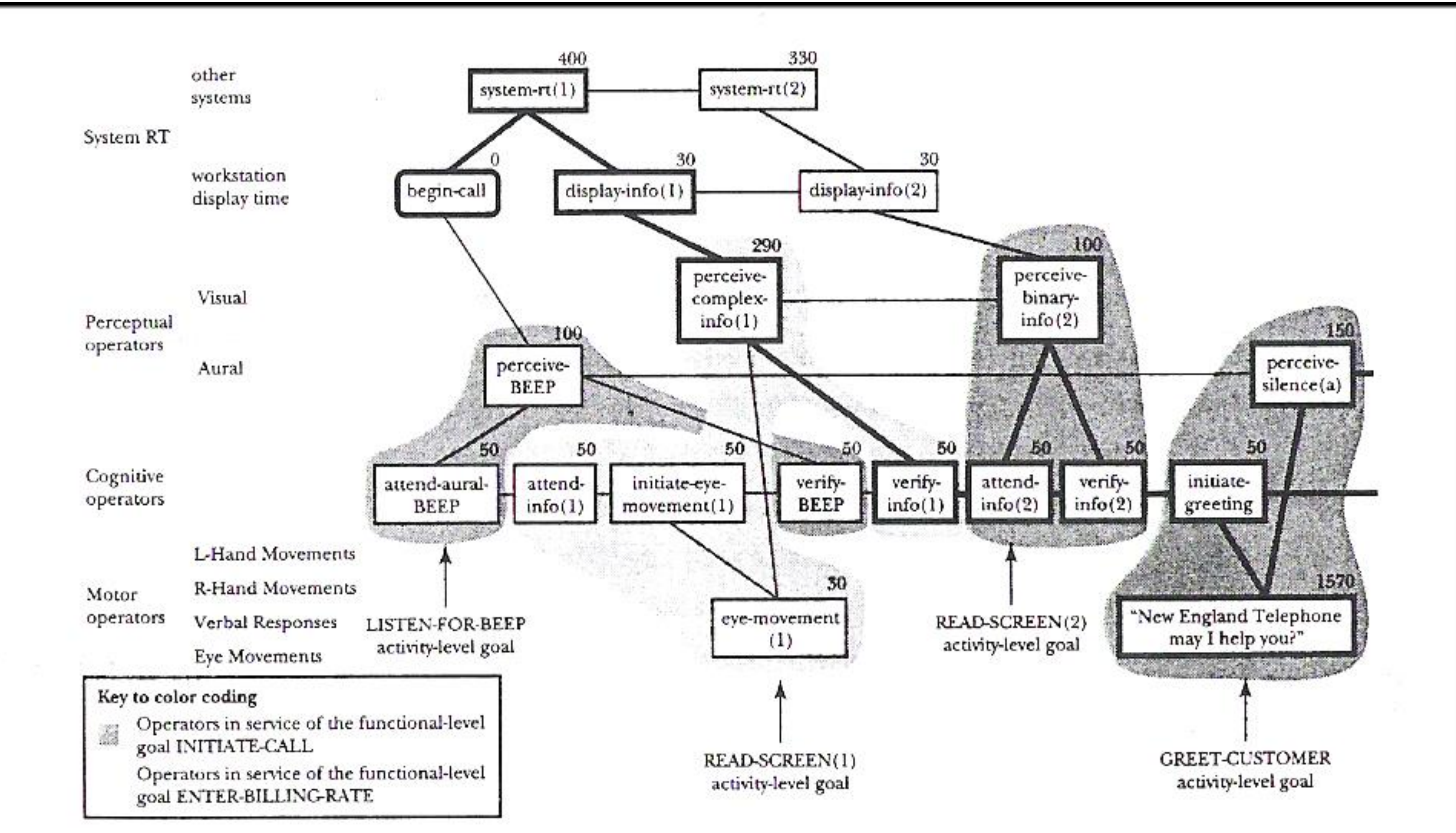
2.Step: Using the GOMS model for a CPM-GOMS Analysis



- The parallelism of the TAOs' task is represented in a schedule chart.
- Boxes: operator with associated time duration (see top corner).
- Lines: information-flow dependencies.
- Middle row: cognitive operators.
- Above: perceptual operators.
- Below: motor operators.
- **Critical path:** the sequence of processes with the longest overall duration.

→ Determines the shortest time possible for the completion of the task.

2.Step: CPM-GOMS Analysis (part of the resulting model)



In the Case Study…

The objectives were to compare the current and proposed workstation to inform a buying decision of the company, and also to validate the CPM-GOMS approach.

- Observation-based CPM-GOMS models for the current workstation,
- Field trial and data analysis,
- Comparing the CPM-GOMS model to the data,

Model validation

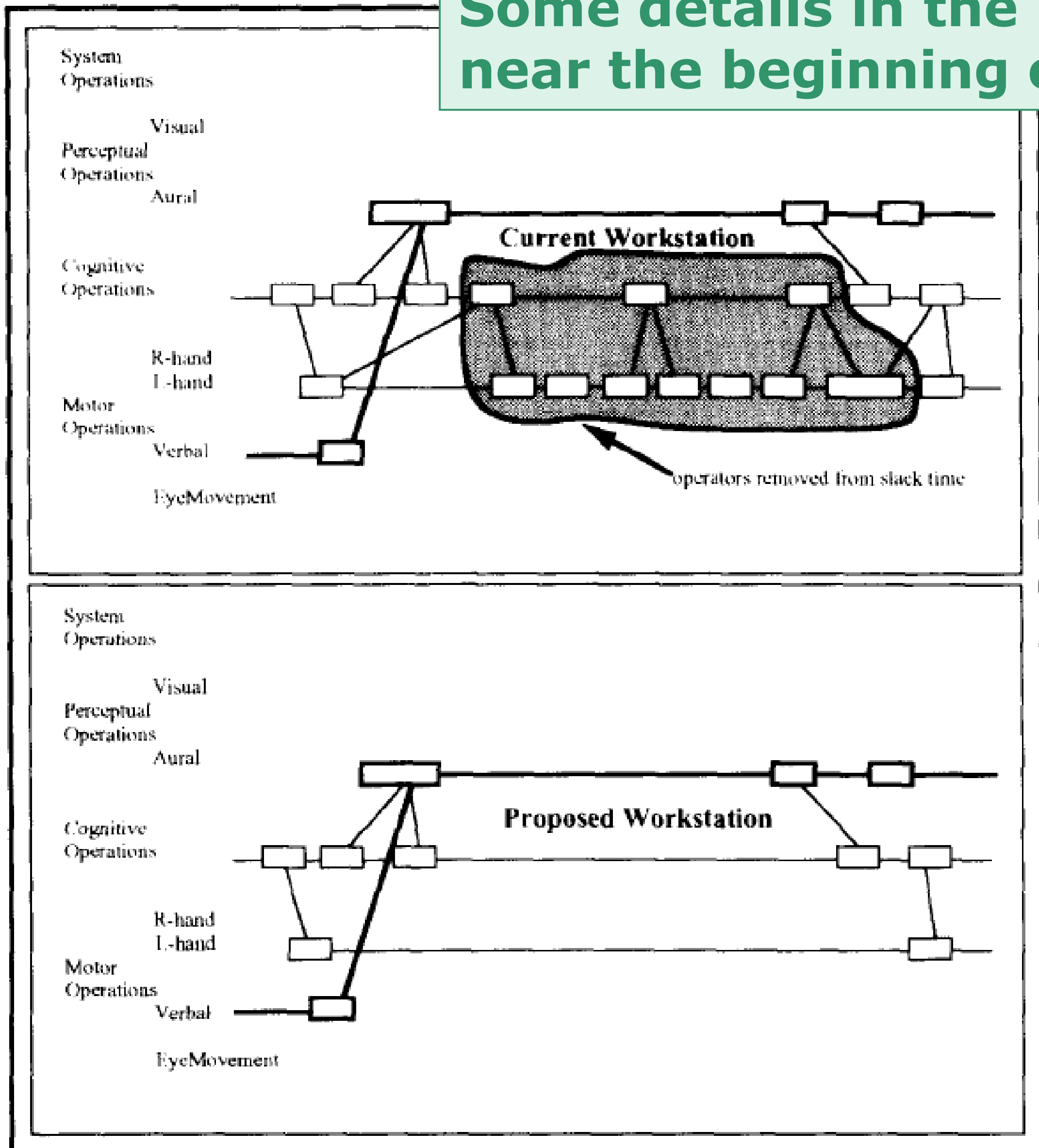
- Specification-based CPM-GOMS models of the proposed workstation,

Prediction: the new workstation would be an average of 0.63 sec slower than the old workstation!
(despite its improved technology and ergonomically superior design)

Why?
- The tasks were dominated by conversation and system response time.
- A TAO’s interaction with the workstation is seldom on a critical path.

Figure 19. Section of CPM-GOMS analysis from near the beginning of the call. Notice that the proposed workstation (bottom) has removed two keystrokes (which required seven motor and three cognitive operators) from this part of the call. However, none of the 10 operators removed was along the critical path (shown in bold).

Some details in the models from near the beginning of the call.



current workstation

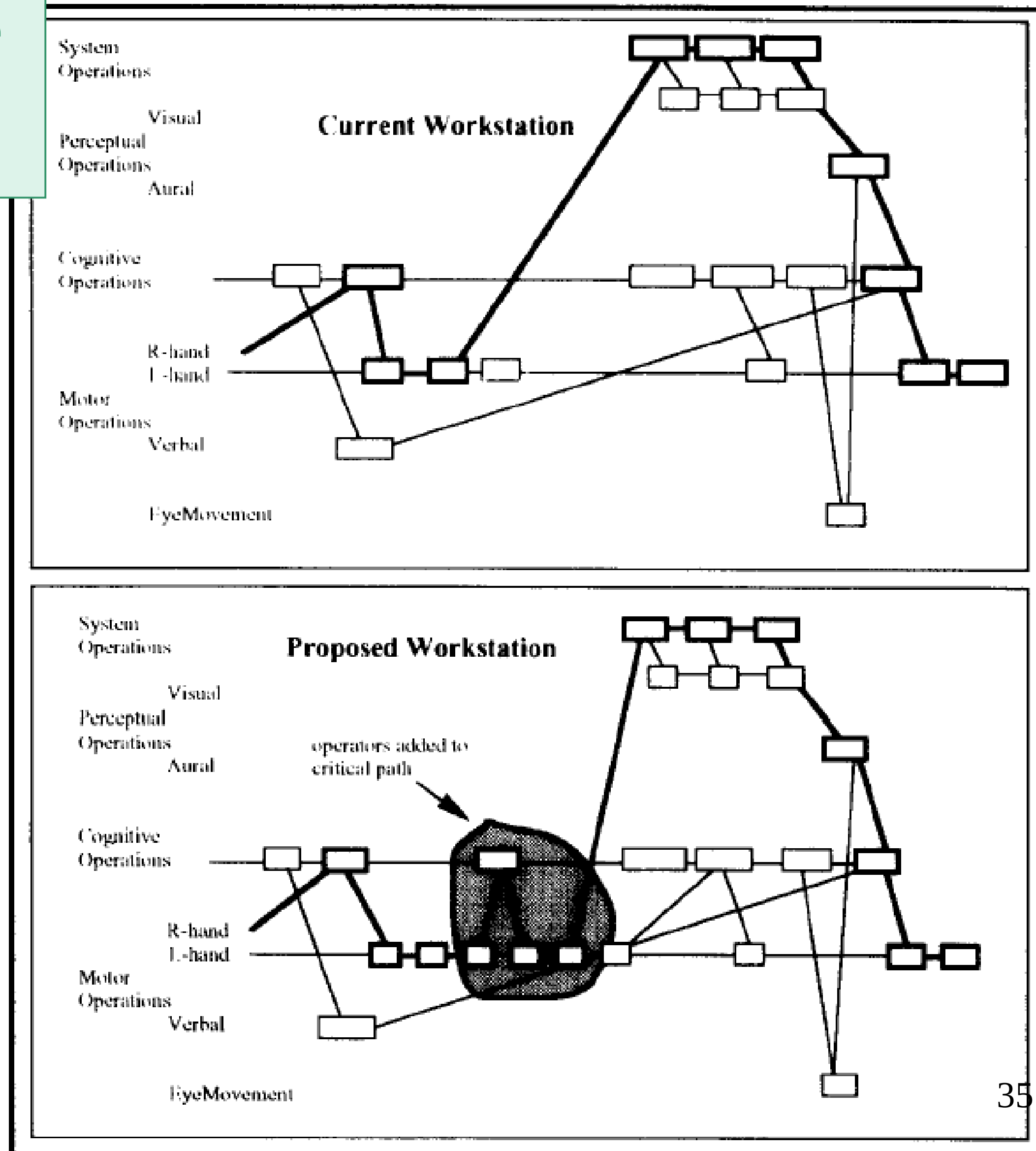
proposed workstation

Figure 20. **Section of CPM-GOMS analysis from the end of the call. Notice that the proposed workstation (bottom) has added one keystroke to this part of the call, which results in four operators (three motor and one cognitive) being added to the critical path (shown in bold).**

Some details in the models from the end of the call.

current workstation

proposed workstation



Summary GOMS

- Computational modeling of human cognition and performance.
- GOMS models describe procedural aspects of tasks. Predictions of performance time and qualitative explanations for those predictions can be provided.
- Level of formality of notation can be adjusted to the requirements of the investigated problem.
- GOMS models can be used from early design to evaluating different detailed designs.

Among the limitations of GOMS:

- Skilled, errorfree behavior is assumed.
- Modeling of single person’s behaviour.
- Problem-solving activities are hard to model.

References

Card, Moran, Newell: The Psychology of Human-Computer Interaction, Lawrence Erlbaum Associates, 1983.

Chapt. 4 in John M. Carroll, editor. HCI Models, Theories, and Frameworks: Toward a multidisciplinary science. San Francisco, Morgan Kaufman Publishers, 2003.

Chapt. 4 in Paul Cairns and Anna L. Cox, editors. Research Methods for Human-Computer Interaction. Cambridge University Press, 2008.

Gray, John, Atwood: Project Ernestine: Validating a GOMS Analysis for Predicting and Explaining Real-World Task Performance. Human-Computer Interaction, Vol.8, pp.237-309, 1993.

Ramkumar, A., Stappers, P. J., Niessen, W. J., Adebahr, S., Schimek-Jasch, T., Nestle, U., & Song, Y. (2017). Using GOMS and NASA-TLX to evaluate human–computer interaction process in interactive segmentation. International Journal of Human–Computer Interaction, 33(2), 123-134.