Human-Computer Interaction

Session 7:
Usability Evaluation

Usability (ISO 9241)

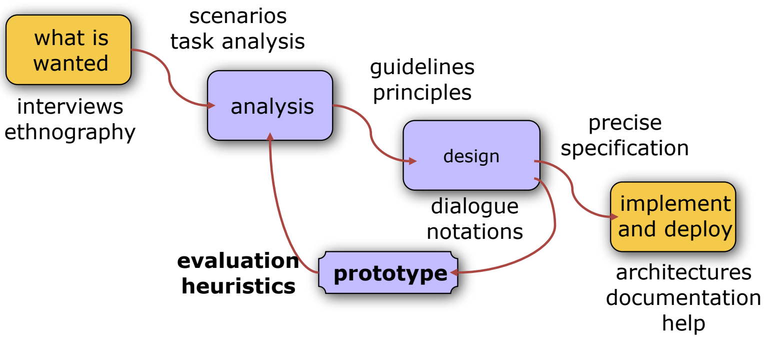
Usability = The effectiveness, efficiency, and satisfaction with which specified users achieve specified goals in particular environments.

Effectivity
- Accuracy and completeness with which the users can in principle achieve a specific goal.

Efficiency
- Effort expended in relation to the accuracy and completeness (quality) of the achieved results

Satisfaction
- Positive attitude of the user towards using the system
- Freedom of using the system without restrictions

User-centered design process



Process to develop interactive systems such that **usability** will be maximized.

Prototyping

The earlier a prototype is built and tested, the better

Horizontal vs. vertical prototypes
- horizontal: complete interface, no/little function
- vertical: functions (partially) implemented
- mixtures of both useful and common

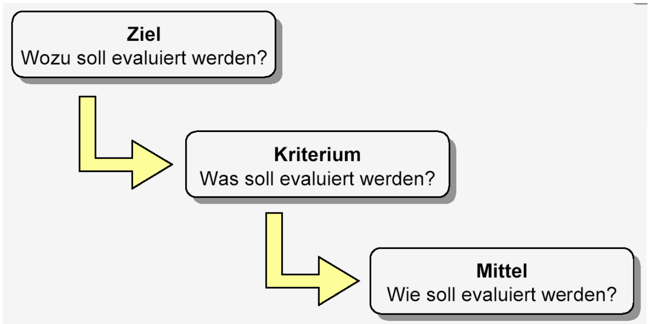
Stages of prototyping
- Conceptual prototype: User gets description/specification and imagines how the system works
- paper prototype: sketches, drafts, pictures, etc.
- static screens: single screen design snapshots
- dynamic simulation: simulates simple procedures
- Wizard-of-Oz: operated by invisible person („wizzard")

Key questions for today

How can the usability of a system be evaluated?

How can usability problems be found and improvements suggested?

Evaluation = Überprüfung eines konkreten Systems auf Übereinstimmung mit vorher festgelegten Kriterien.



Key questions for an evaluation

Why? Assess usability and user effects, find problems, make suggestions for improvement

What? lay down usability criteria

Where? lab or field

Who? expert (w/out user) or real users

When? in all design stages (concept, prototypes, final system)

- Formative evaluation: at different times, assess current system against actual requirements
- Summative evaluation: final assessment of initially defined criteria

Evaluation procedure

1. Define criteria for the system to be usable
2. Define observables and performance levels for each criterion („Operationalisation“)
3. Measurement and Analysis
   - Application of criteria and comparison with performance levels
4. Assessment (Synthesis)
   - Make judgement based on results
   - Derive suggestions for improvement on the criteria

Choosing methods and design

Validity (*Gültigkeit*): Will criteria be observed/measured?

Reliability (*Zuverlässigkeit*): Is the study reproducible?

Significance and Generalisation: Selection of participants, influence of the context of the study on observed behavior?

Pilot/Pre-Study
- If something is not fully clear, always make a pre-study
- Test feasibility and practicability, practice procedure, improve
- Can employ colleagues as test subjects
- A row of pre-studies might possibly be required

Evaluation methods

Usability inspection (*expert review*)
- Guidelines review & consistency inspection
- Cognitive walkthrough
- Heuristic evaluation
- Focus group

User studies
- Usability testing
- Thinking-Aloud
- Field studies
- Interviews & questionnaires

Model-based evaluation

Usability inspection methods

Guidelines Review
Consistency Inspection
Cognitive Walkthrough
Heuristic Evaluation

Guideline review & consistency inspection

System/interface is checked for conformance with guidelines
- Standard guidelines, e.g. Shneiderman's rules
- Organization-specific guidelines, e.g. Apple styleguide

Consistency inspection
- of terminology, colors, fonts, icons, menues, general layouts, etc.
- of interaction style

Cognitive Walkthrough

Task-oriented inspection method („Benutzbarkeits-Gedankenexperiment“)

Evaluators (usually usability experts) tests functions like an imaginary user

- selects task for the system to support
- performs task step by step (*walks through*)
- determines specific action sequences and identifies potential problems for a user

Advantage:

- Can be carried out early and spot mis-conceptions early on

Problem: Can an evaluator ever „simulate“ a user? May also employ users as evaluators

Cognitive Walkthrough

1. Prepration

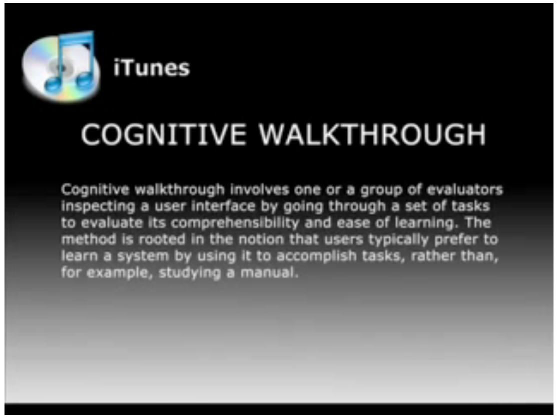
- Detailed spec of potential user
- Detailed spec of task, structured in single steps
- List of possible actions and their results
- Prototype of the system (paper, partially implemented, etc.)

2. Analysis

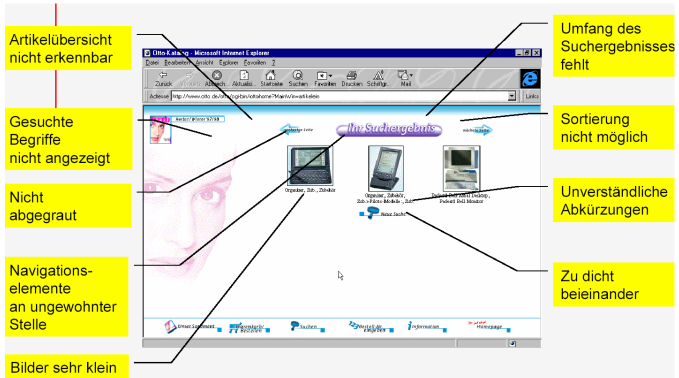
- Expert walks through all actions and system responses, each time answering the following questions:
  - Are the right actions available (effects = user goals/intentions)?
  - Will the user be able to identify the actions as such?
  - Will the user find the correct actions?
  - Will the user understand the system feedback?

3. Follow-Up

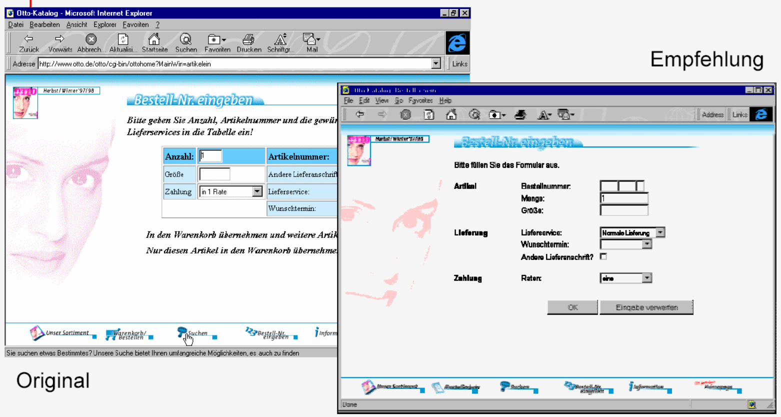
- Recordings of results and ideas about alternative design and further improvements



Example: inspection of Otto Versand webpage...



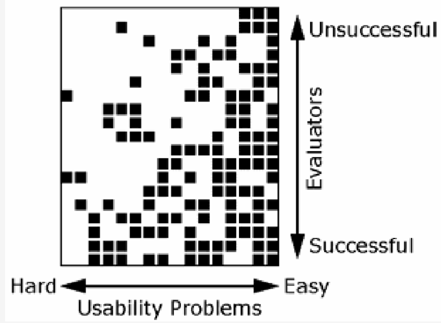
...and recommendations



Empfehlung

Original

Wieviele Reviewer ?



19 Reviewer sollten 16 Fehler in einer Bankensoftware finden:

- Unterschiedliche Reviewer fanden durchaus unterschiedliche Fehler
- Die erfolgreichsten Reviewer finden nicht immer die schwierigsten Fehler

Einsatz mehrerer Reviewer sinnvoll !

Nielsen, J.: How to conduct a heuristic evaluation, http://www.useit.com/papers/heuristic/heuristic_evaluation.html

Optimal: 4 Reviewer - Nutzen 62 mal größer als Kosten
5 Reviewer erkennen 75-80 % Fehler – gut, aber:
-> nicht im Kernkraftwerk anwenden!

Heuristic Evaluation

J. Nielsen (1993)
www.useit.com

Usability-Experten bewerten System/Prototyp anhand einfacher und allgemeiner Usability-Heuristiken

Unabhängig von mehreren Experten durchzuführen

- Daumenregel: 5 Experten finden 75% der Probleme

Testen entweder lauffähiges System oder Prototypen

Heuristiken/Kriterien:

- Nielsen‘s 10 Heuristiken (1993; siehe Vorlesung 6)
- Erweiterte Heuristiken ab 2001 (Nielsen, 2001)

Usability heuristics (1)

Visibility of system status

Match between system and the real world

- Speak the users' language, follow real-world conventions, make information appear in a natural and logical order

User control and freedom

- Provide a clearly marked "emergency exit" to leave an unwanted state (undo and redo)

Consistency and standards

- Users should not have to wonder whether different words, situations, or actions mean the same thing.

Error prevention

Usability heuristics (2)

Recognition rather than recall

Flexibility and efficiency of use

- cater both inexperienced and experienced users, allow to tailor frequent actions

Aesthetic and minimalist design

- provide no irrelevant or rarely needed info

Help users recognize, diagnose, and recover from errors

- Error messages in plain language (no codes), precisely indicate the problem, suggest a solution.

Help and documentation

- provide help and documentation, easy to search, focus on user task, list concrete steps to be carried out, not too large

Heuristic Evaluation

1. **Training session**
   - Reviewers practice detailed heuristics
2. **Evaluation**
   - Each reviewer evaluates with a list of standard heuristics the interface - normally 4 iterations
   - Tests the general flows of tasks and functions of the various interface elements (not strictly task-oriented)
   - Observer takes notes of identified problems
   - Reviewers communicate only after their iterations

Heuristic Evaluation

3. **Results and reviewer session**
   - Make list of problems (violated principles+reasons)
   - Detailed descriptions of the problems
4. **Problem assessment**
   - How serious and unavoidable is a usability problem?
   - Each reviewer assesses each identified problem with respect to its severity:
     - 0 - don't agree that this is a usability problem
     - 1 - cosmetic problem
     - 2 - minor usability problem
     - 3 - major usability problem - important to fix
     - 4 - usability catastrophe; imperative to fix
   - Final ranking of all problems

Heuristic Evaluation

Example:

- *Interface used command „Save" on 1st screen for saving the user's file, but used „write file" on 2nd screen. Users may be confused by this different terminology.*
- Violation of consistency/standards - severity rating 3

Advantage:

- fast, cheap, qualitatively good results

Problems:

- experts aren't real users
- heuristics do not cover all possible problems

User studies

Thinking-Aloud
Cooperative Evaluation
Interviews & questionnaires
Usability-Test

User studies

Interactions between actual users and a system

Measure representative users' performance on typical tasks, for which the system was designed

Use video and interaction logging to capture errors and frequencies and time of commands, or think-aloud protocols

May be performed in the lab or the field

Users may be interviewed or complete questionnaires
- gather data about users' opinions

Lab studies

- Experiment under controlled conditions
  - specialist equipment available
  - uninterrupted environment
- Disadvantages:
  - lack of context
  - difficult to observe user cooperation
- Prevalent paradigm in exp. psychology

Field studies

- Experiments dominated by group formation
- Field studies more realistic
  - *distributed cognition* ⇒ work studied in context
  - real action is *situated*
  - physical *and* social environment crucial
- sociology and anthropology – open study and rich data

Think Aloud

User is observed while performing a predefined task and asked to describe what …
- he is expecting *to* happen
- he is thinking *is* happening

- Advantages
  - simplicity - requires little expertise
  - can provide useful insight into user's mental model
  - can show how system is actually used
- Disadvantages
  - artificial test situation → cooperative evaluation
  - subjective and selective → multiple trials & users needed
  - act of describing may alter task performance

Cooperative Evaluation

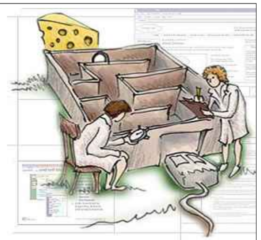
- User evalutes together with expert, sees himself as collaborator in evaluation
  - both can ask each other questions
- Additional advantages
  - less constrained and easier to use
  - user is encouraged to criticize system
  - clarification dialogues possible
- Problems with *both* techniques
  - generate a large volume of information (*protocols*)
  - 'Protocol analysis' crucial and time-consuming

Query techniques

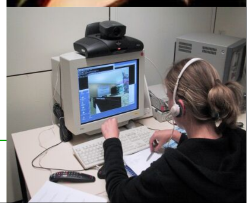
Interviews:
- analyst questions user, based on prepared questions
- informal, subjective, and relatively cheap
- can be varied to suit context, issues can be explored more fully, can reveal unanticipated problems

Questionnaires:
- fixed questions given to users, need careful design!
- Style of questions: open vs. closed, scalar vs. binary, multiple-choice, ordering, negative vs. positive, ...
- Style of answers: text, yes/no, number of options, ...
- reaches large user group, can be analyzed rigorously, less flexible, less probing

Usability Testing



- Aufnehmen typischen Benutzerverhaltens bei typischen Aufgaben in kontrolliertem Szenario
- Benutzer werden bei Aufgabenbearbeitung beobachtet und auf Video aufgenommen, Tasten/Mausbewegung "geloggt"
- Daten genutzt um Bearbeitungszeit zu berechnen, häufige Fehler zu entdecken, erkennen, warum User etwas tun
- "Satisfaction": Fragebögen und Interviews für subjektive Meinungsäußerung



Usability Testing

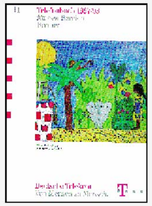
1. Suche repräsentative Benutzer
   - 5-10 Benutzer als Testpersonen
2. Kriterien der Auswertung auswählen (Beispiele):
   - Zeit für Aufgabenerfüllung
   - Zeit für Aufgabe nach Ablenkung/neuem Input
   - Anzahl und Art von Fehlern pro Aufgabe oder pro Zeiteinheit
   - Anzahl Zuhilfenahme Onlinehilfe oder Manual
   - ...
3. Entwickle Testszenarien
   - relevante Szenarien (typische vs. Extremsituationen)
   - Halte Aufgaben kürzer als 30 Minuten
   - Identische Testbedingungen für alle
4. Ethische Fragen?
   - Probanden Aufklären, Einverständniserklärung, etc.

Usability Testing

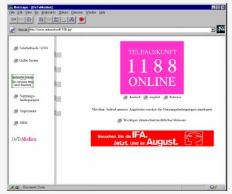
4. Vorab Pilottests
   - Schulung von Experimentatoren und Beobachtern
5. Eigentlicher Test
   - Einführung/Erläuterung des Tests für die Versuchspersonen
   - Testdurchführung und Datenaufzeichnung
6. Auswertung
   - Statistiken, z.B. Maus-Events, Menü-Auswahlen
   - Bildschirm-Layout: Blickverfolgung und Aufgabenablauf
   - Post-task Videokonfrontation und User-Interview
7. Vermittlung der Ergebnisse an Entwickler

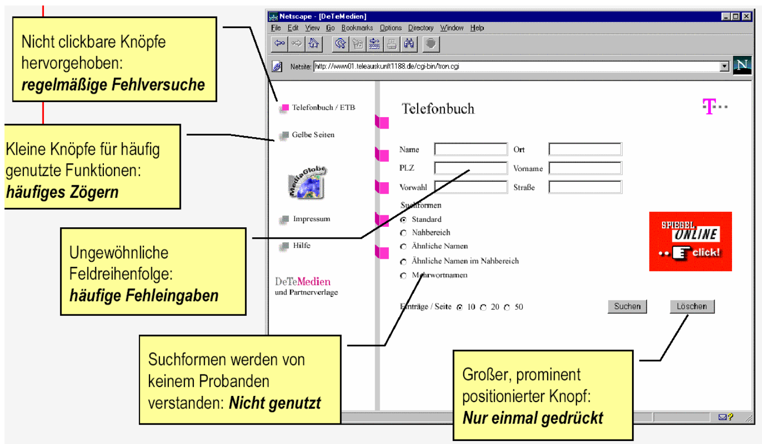
Usability Testing - Beispiel

Ziel: Vergleich unterschiedlicher Telefonauskunftsysteme
- hinsichtlich ihrer Benutzbarkeit.
- Verfahren: Vier Versuchspersonen bearbeiten jeweils 4 Prüfaufgaben.
- Die Bearbeitung wird mit Video, Audio und Logging-Programmen protokolliert.


Telefonbuch


Telefon-CD der DeTeMedien


www.teleauskunft1188.de



Zeitdauer & Korrektheit im Vergleich

Zusammengefaßte Ergebnisse

Aufgabenstellung				
1. Suche die Telefonnummer von Maria Müller. Sie wohnt Am Ziegelberg in Bremen.	Korrekte Ergebnisse	★★★	★★	★★★★★
	Bearbeitungsdauer [min]	0:45	2:30	3:00
2. Suche die private Telefonnummer von Carsten Bormann (TZI-Bereich Digitale Medien und Netze).	Korrekte Ergebnisse	★	★	
	Bearbeitungsdauer [min]	0:30	1:00	2:45
3. Marc-Oliver Schulze wohnt bei seinem Vater in Bremen. Seine Telefonnummer beginnt mit einer "40".	Korrekte Ergebnisse	★★★★★	★★★★	
	Bearbeitungsdauer [min]	1:15	1:50	4:10
4. Suche einen Sportarzt in Bremen.	Korrekte Ergebnisse	★★★	★	★½
	Bearbeitungsdauer [min]	0:30	2:30	4:20

Beobachtung Usability Test

Physiological measurements

Emotional response linked to physical changes

may help determine a user's reaction to an interface

measurements include:

- heart activity, including blood pressure and pulse
- activity of sweat glands: Galvanic Skin Response (GSR)
- electrical activity in muscle: electromyogram (EMG)
- electrical activity in brain: electroencephalogram (EEG)

often difficult to interpret physiological responses

Eye tracking

eye movement reflects amount of cognitive processing a display requires

measurements include

- fixations: eye maintains stable position. Number and duration indicate level of difficulty with display
- saccades: rapid eye movement from one point of interest to another
- scan paths: moving straight to a target with a short fixation at the target is optimal

Remember, methods in UCD

1. **Field studies**
2. User requirement analysis
3. Iterative design
4. **Usability evaluation**
5. Task analysis
6. **Focus groups**
7. **Formal heuristic evaluation**
8. **User interviews**
9. **Surveys**
10. …

Ranking based on a survey among experienced UCD practitioners (103 questionnaires) (Mao et al., 2005)

Model-based evaluation

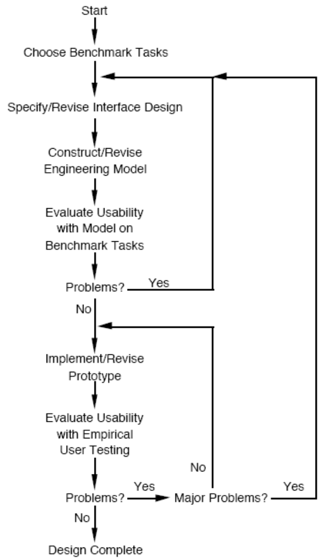
Model-based evaluation

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

Provides usability results *before* building a prototype or user testing

Engineering the model allows more design iterations



Model-based evaluation

The model summarizes interface design from the user's point of view

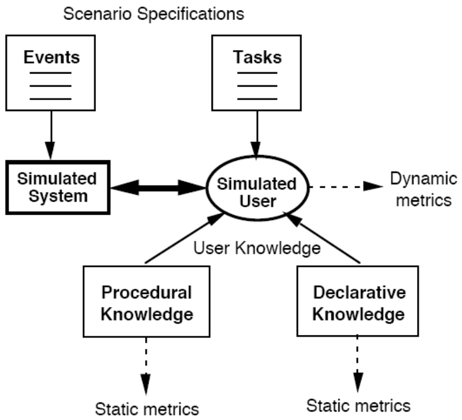
- Represents how the user gets things done with the system (user-system interaction)
- Components can be reused to represent design of related interfaces

But, current models can only predict few aspects:

- Time required to execute specific (low-level) tasks
- Ease of learning of procedures, consistency effects

Actual user testing is still indispensible!

Overview



Models = simulations of human-computer interaction

Procedural knowledge
how-to procedures
→ executable

Declarative knowledge
facts, beliefs
→ reportable

Modeling human constraints

If a model can be programmed to do any task at any speed or accuracy, something is wrong

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)

Implications:

- 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 with respect to...

- human constraints modeled (cognitive/psychological vs. perceptual vs. motoric)
- level of detail
- when to use it in the design process

Task Network Models

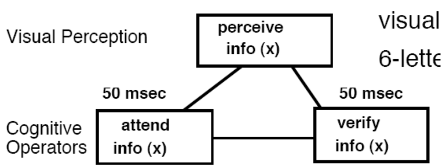
Tasks = mixture of human and machine tasks

Each task characterized by a distribution of completion times, and arbitrary dependencies and effects

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

Task network - simple example



Cognitive architectures

"Programmed" with a strategy to perform specific tasks

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

To model a specific task...

- do task analysis to arrive at human's task strategy
- "program" 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, quite complex; used mostly in a research settings

EPIC Architecture

(Kieras & Meyer, mid-1990s)

Developed to represent executive processes that control other processes during multiple task performance

Executive-**P**rocess **I**nteractive **C**ontrol

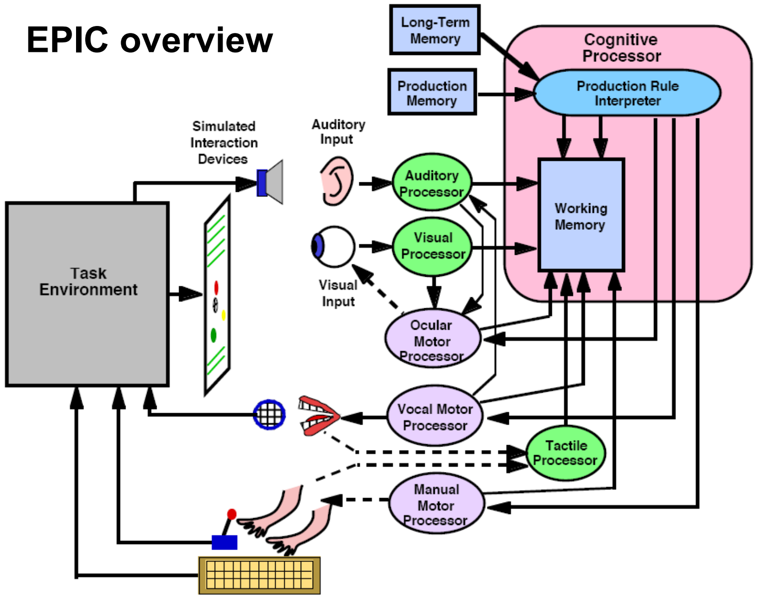
General properties

- Production-rule cognitive processor
- Parallel perceptual and motor processors
- Components, pathways, and most time parameters

Task-dependent properties

- Cognitive processor production rules (strategy)
- Perceptual recoding
- Response requirements and styles

EPIC overview



GOMS (Card, Moran, & Newell, 1983)

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
- **S**election Rules - which method should be used

Well worked out, 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 (0.12 - 1.2 sec; 0.28 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 series of *n* Ks done as a unit

P - Point with mouse to a target on the display
- Follows Fitts' law if possible: 0.1 * log2 (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
- New users often pause to remember or verify each 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

General procedure: Find file icon and drag into 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

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

General procedure: select file icon 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 the 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

Outlook - next sessions

Year	**Paradigm**	**Implementation**
1950s		*Switches, punched cards*
1970s	*Typewriter*	*Command-line interface*
1980s	*Desktop*	*Graphical user interface, direct manipulation*
1980s+	*Spoken Language*	*Speech recognition/synthesis, natural language processing, dialogue systems*
1990s+	*Natural interaction*	*Perceptual, multimodal, interactive, conversational, tangible, adaptive*
2000+	*Social interaction*	*Agent-based, anthropomorphic, social, emotional, affective, collaborative*