HCC Theories

Fundamental theories of Human-Centered Computing

HCC Theories

Today's goal (and next lecture):

Cover a number of important HCC theories

Outline:

- Classical theories (Norman's and cognitive modeling)
- Modern theories (Distributed Cognition, Situated Action, Activity Theory)
- Contemporary theories

But first...

Project proposal: 2–3 person groups

Do you already have a project proposal group?

If not, what are you interested in?

Please talk after class and/or use the Canvas discussion board!

Send me your group composition before Monday's class

Norman's Theory

The design of everyday things

Norman's Theory

Don Norman applied cognitive psychology to the design of everyday things

This resulted in an applied but very generic theory of:

- How people interact with computers
- Why they sometimes fail
- How to make it better

Norman's Theory

The action cycle and gulfs of execution/evaluation

> Explains how people use interfaces, and why they sometimes fail

Designer image, system image, use image

> Explains what causes some systems to be less usable than others

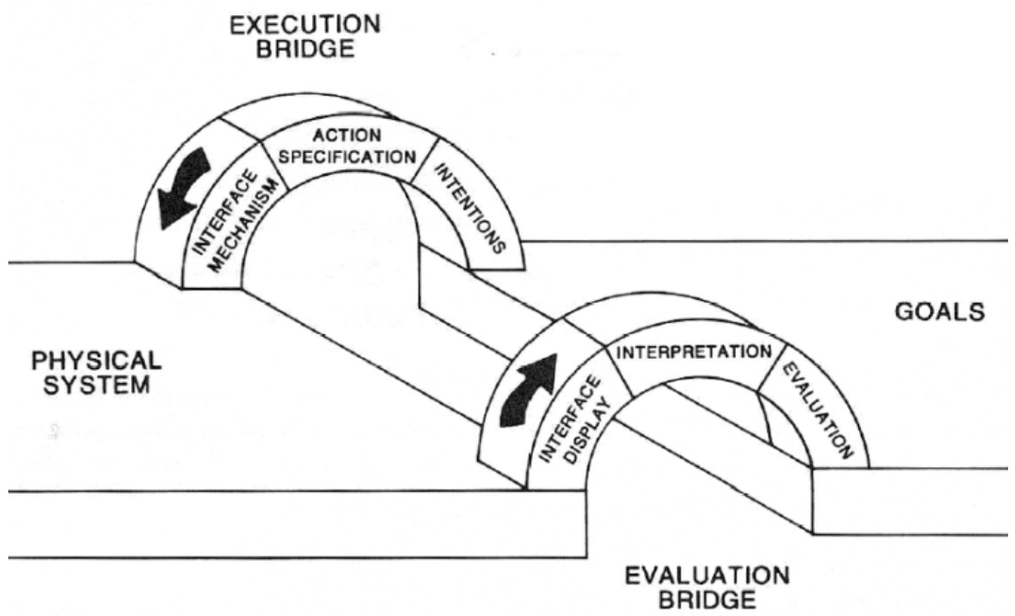
Constraints, signifiers, and feedback

> Explains how you can increase the usability of interfaces

The action cycle

Norman created an abstract representation (a model) of how users perform tasks:

- How they turn their goals into actions (system input)
- How they evaluate the resulting system output

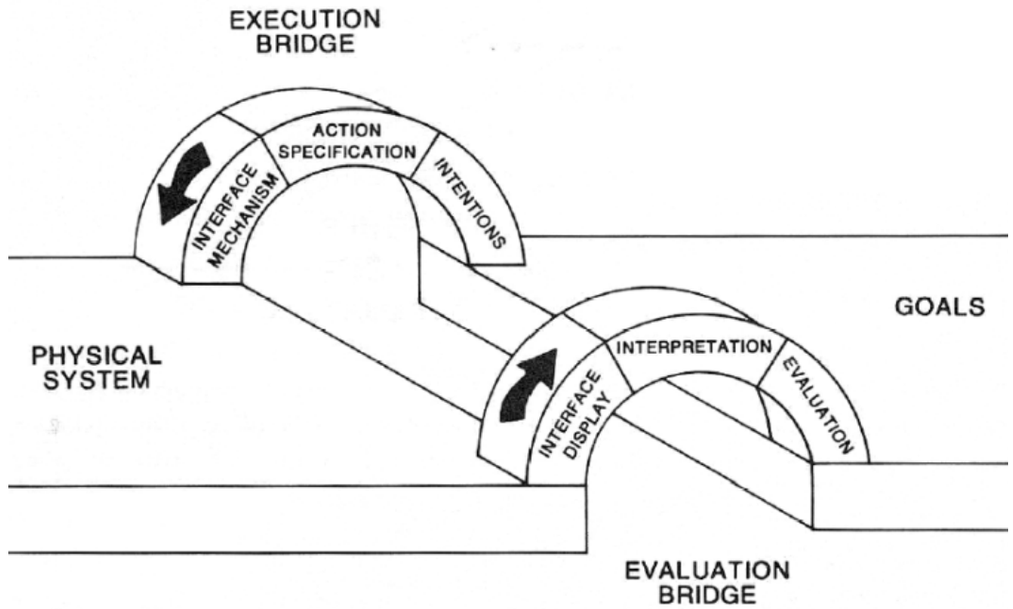


Example

My **goal** is to be able to read the slides during class

I will **execute** a series of actions to print them

After each action, I will **evaluate** whether it brought me closer to my goal



Example

1. Plan to turn my goal into an intention to act

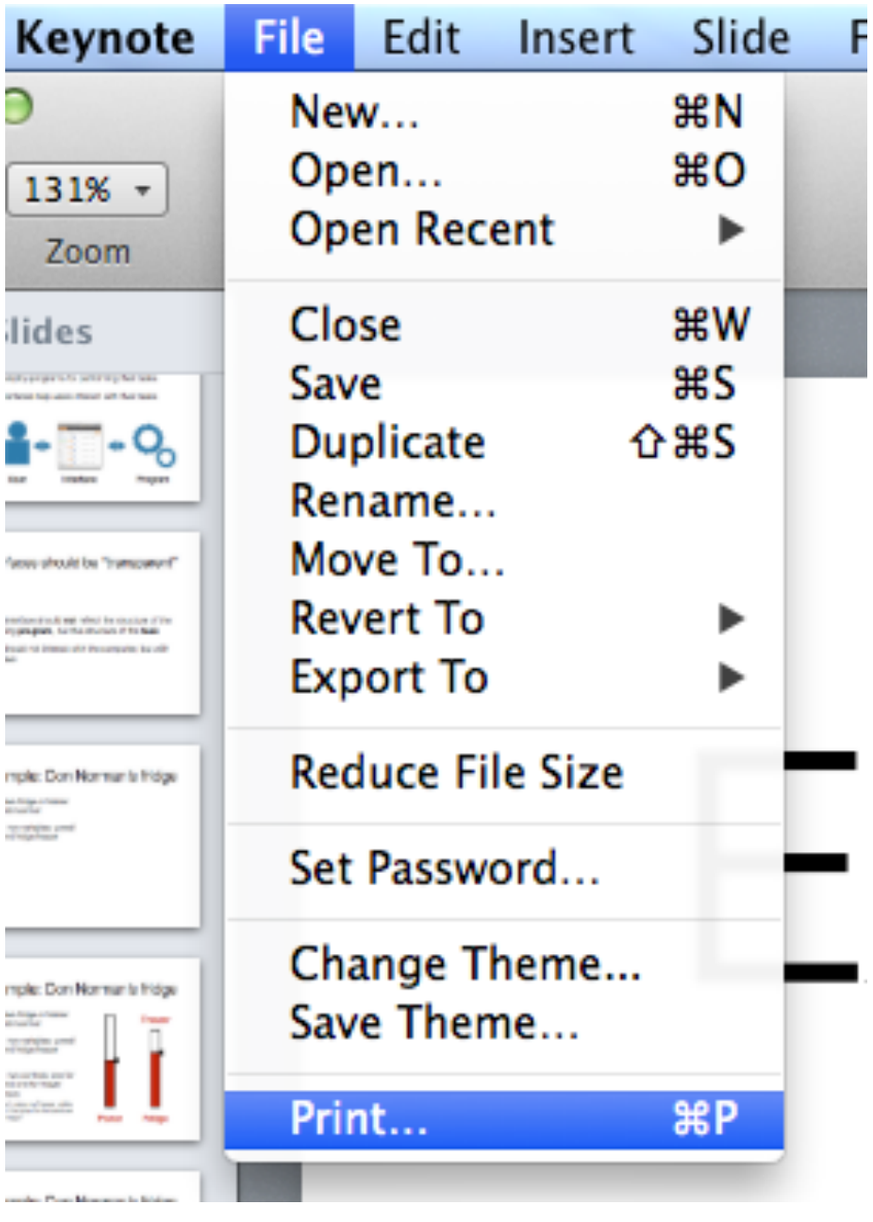
use my home printer to print the email

2. Specify an action sequence

click File > Print

3. Perform this sequence

<click>



Example

4. Perceive the change

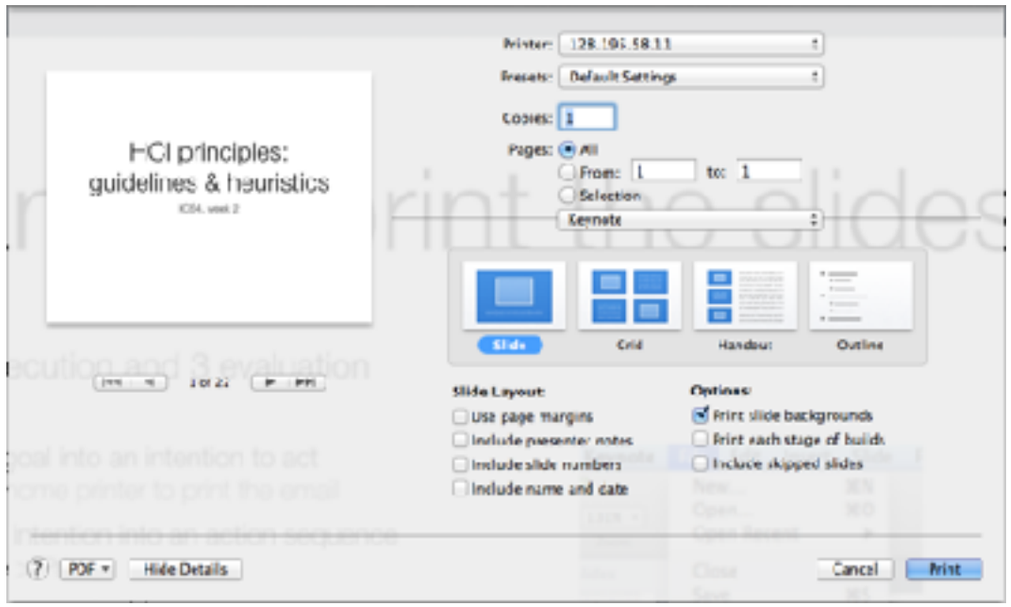
this causes a dialog to pop up...

5. Interpret the dialog

this dialog allows me to print

6. Evaluate the outcome

does this bring me closer to my goal? Yes, it does



Gulf of execution

Things that can go wrong in the **execution**-part:

- Failure to formulate an intention
  - I don't realize that I can print my document
- Failure to formulate an action sequence
  - I don't know where to find the print dialog
- Failure to execute the action
  - Some other dialog is still open, preventing me from using the menu

Gulf of evaluation

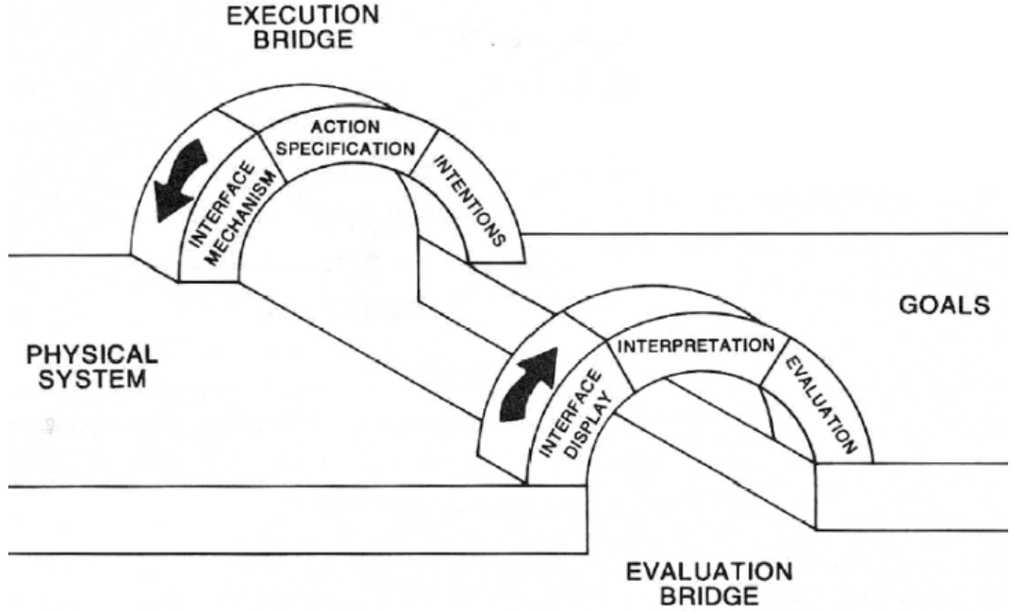
Things that can go wrong in the **evaluation**-part:

Let's say that the default printer is wrong:

- Failure to perceive the outcome
  - I don't notice the default printer in the dialog
- Failure to interpret the outcome
  - I notice it, but I think that this is the correct printer because it has almost the same name
- Failure to evaluate the outcome
  - I notice that the name is different, but I (incorrectly) assume that this is just a glitch, and I'm using the correct printer anyway

Discussion

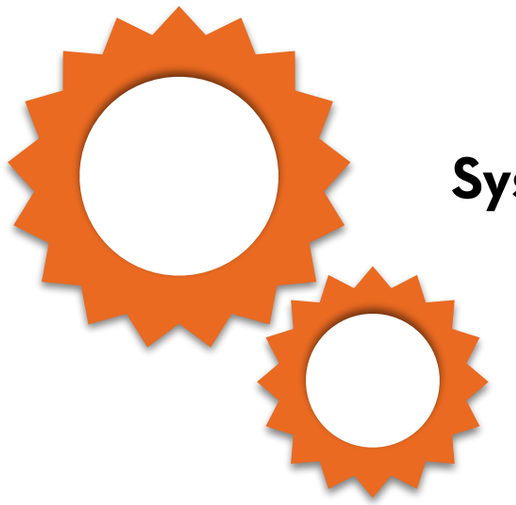
What is missing from the action cycle?



Images

Norman noticed that both designers and users reason about the system...

Program

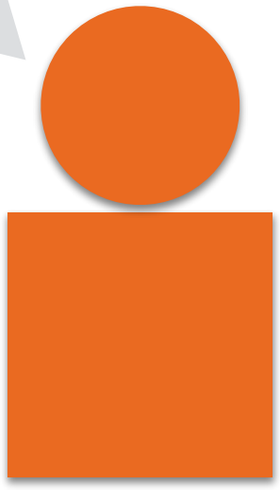


System image

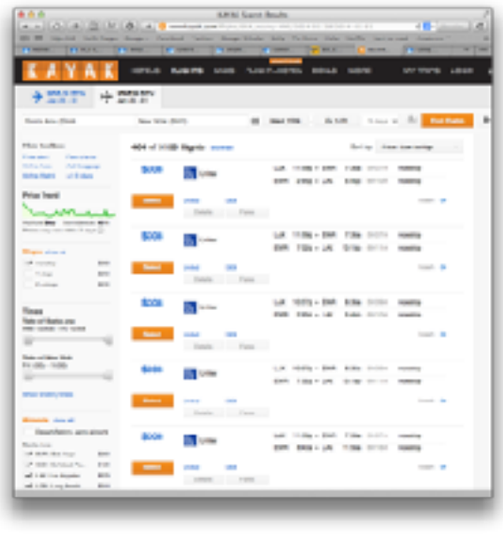
Designer image

operate

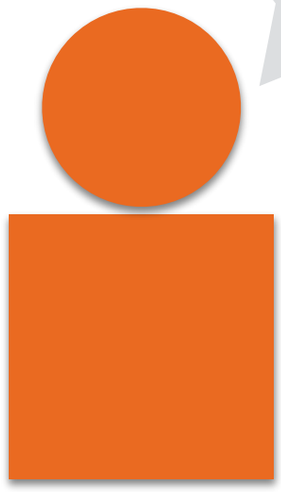
Use image



design



interpret



Designer

Interface

User

Images

Most usability problems happen due to a mismatch between system image and use image

Why does this happen? It's like a game of charades:

- The designer creates the UI based on the system image
- The user has to infer the system image from the UI

Example

User question: What does this icon mean?

- Shower?
- Spotlight?
- Kitchen vent?

Bad icon! Too many options!

Example

Designer question: Design an icon for:

- Shower
- Spotlight
- Kitchen vent

If you know the purpose, the icon looks fine!

User interfaces

Norman argued that certain aspects of a user interface can help align the use image and system image:

- Constraints
- Signifiers
- Feedback

Constraints

Physical constraints: object can only be used in one way

Cultural constraints: use is culturally determined

Semantic constraints: use is determined by the situation

Logical constraints: use follows a natural mapping

Constraints

Physical constraints: object can only be used in one way

Cultural constraints: use is culturally determined

Semantic constraints: use is determined by the situation

Logical constraints: use follows a natural mapping

Signifiers

Signifiers:

- Design that shows how it should be used
- Example: button vs. button

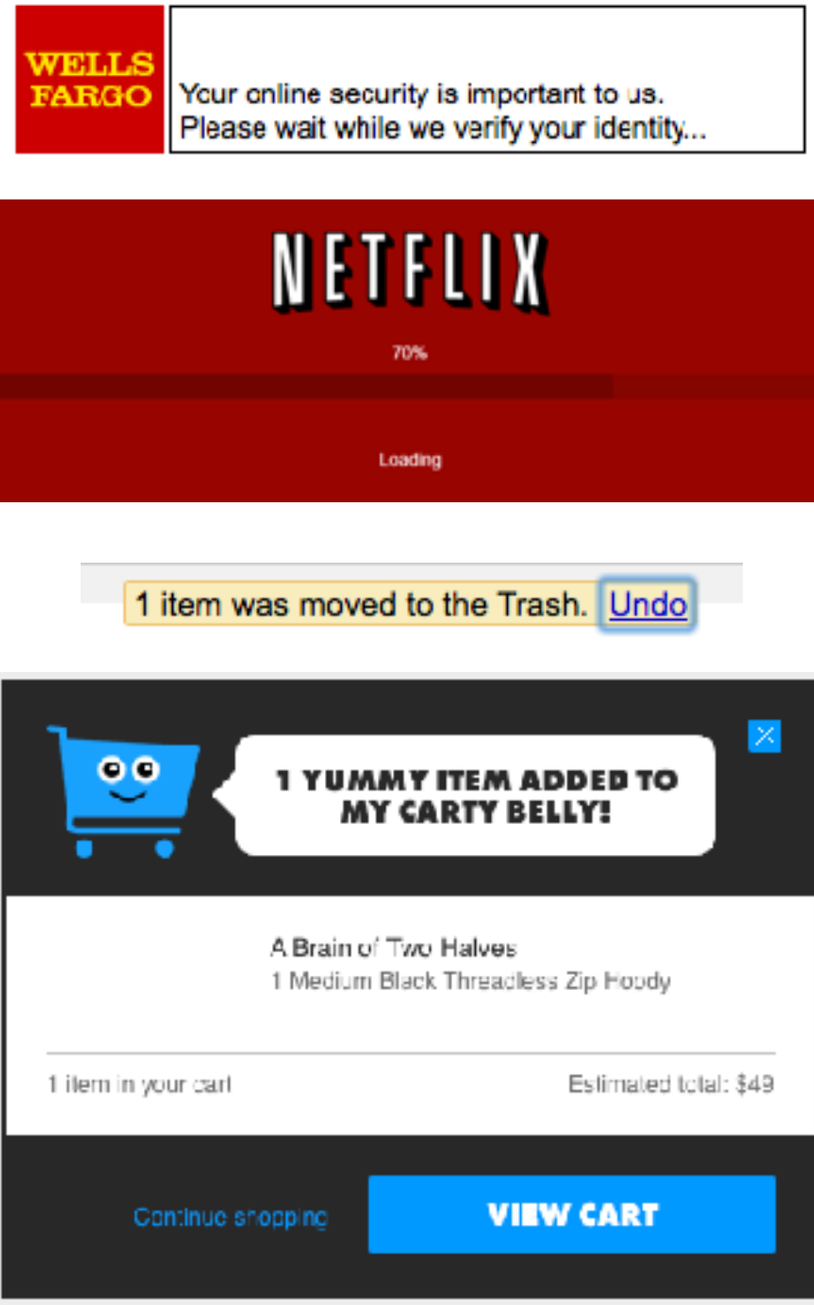
Feedback

Feedback:

- Design that shows what is happening
- ideally < 0.1 sec

Examples:

- “click!”
- highlight
- “loading”
- confirmations



Bridging the gulfs

Careful use of constraints, signifiers, and feedback help reduce the mismatch between system image and use image

Note: the system image must still match the user's task!

Discussion:

What are the limits of affordances/signifiers and feedback?

What are good examples of constructed signifiers?

Why do they work?

Cognitive Modeling

Fundamentals of Human-Centered Computing

Cognitive Modeling

Cognitive architectures

Abstractions of the mind, useful for reasoning

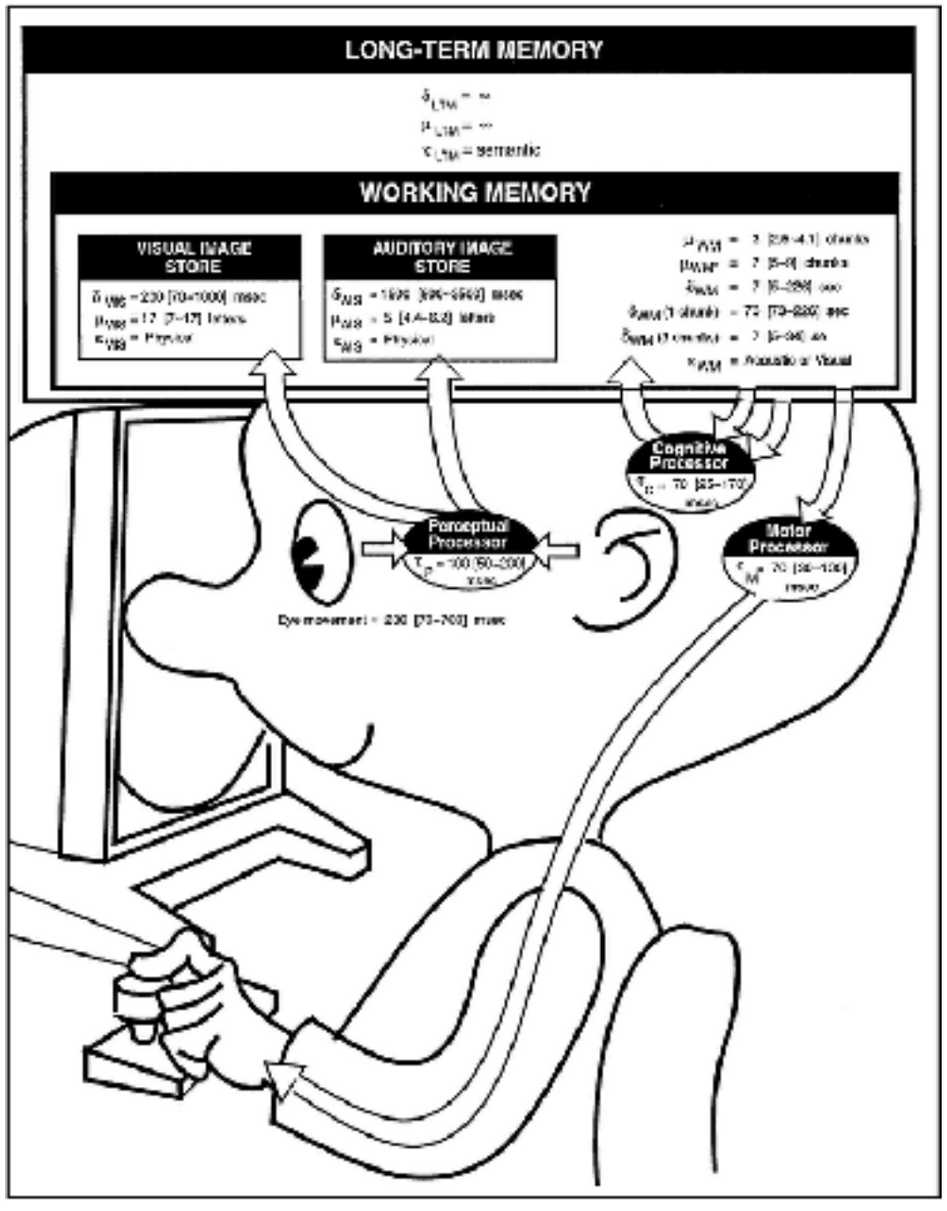
Cognitive modeling

A usability analysis based on how the brain works

Cognitive architecture

A cognitive architecture
is a specification of the structure of the brain
at a level of abstraction that explains how it
achieves the function of the mind.

Initial architecture



Model-Human Processor

Describes the brain's performance boundaries

Can be used to calculate performance in a user interface task

Problems

MHP describes the mind, but largely ignores the brain

Problem: This is like a specification of a building's architecture that ignores what the building is made out of

Some modeling parameters are impossible!

e.g. some cognitive models are intractable

ACT-R

Visual: see the problem

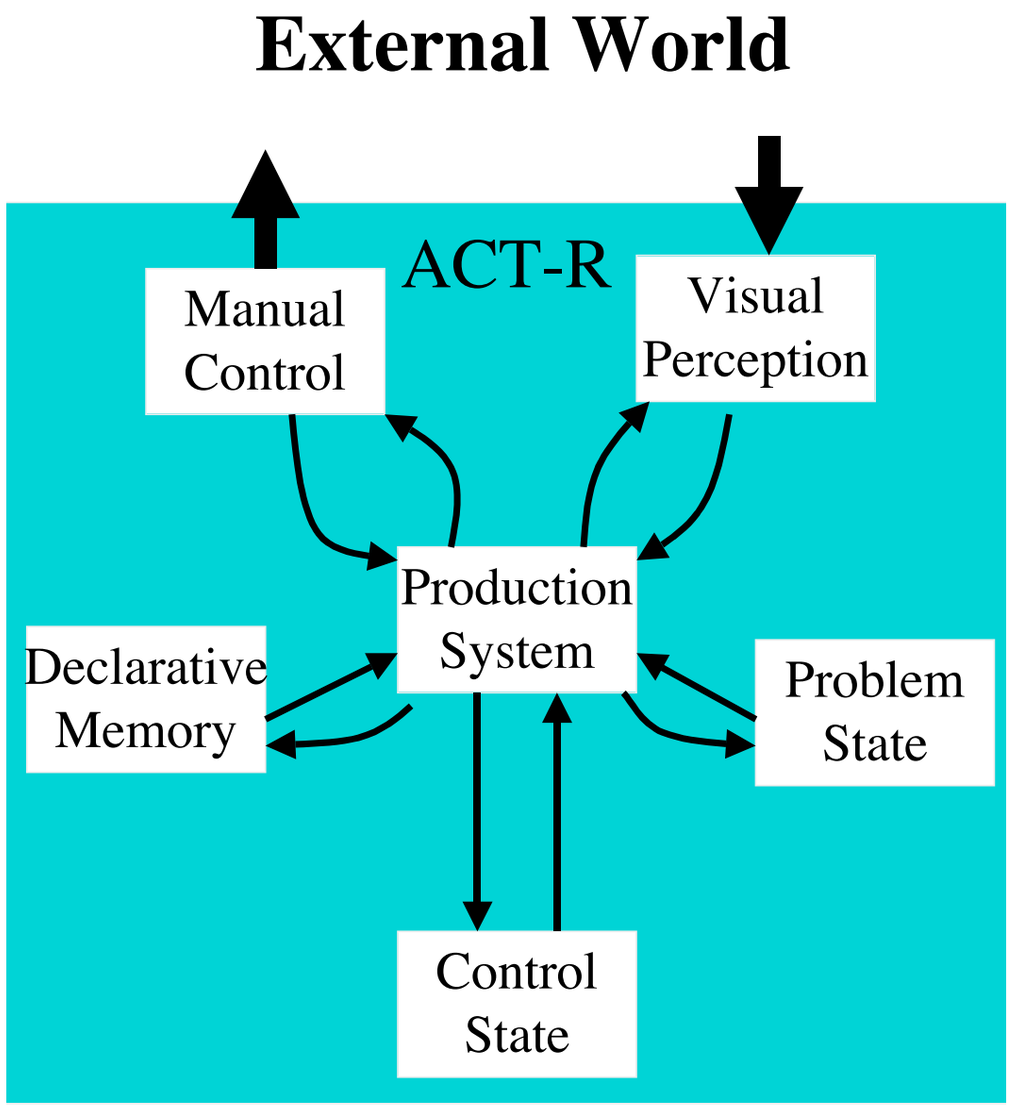
Problem state: STM-ish

Control state: objective

Declarative: LTM

Manual: create output

Production system: move things between modules



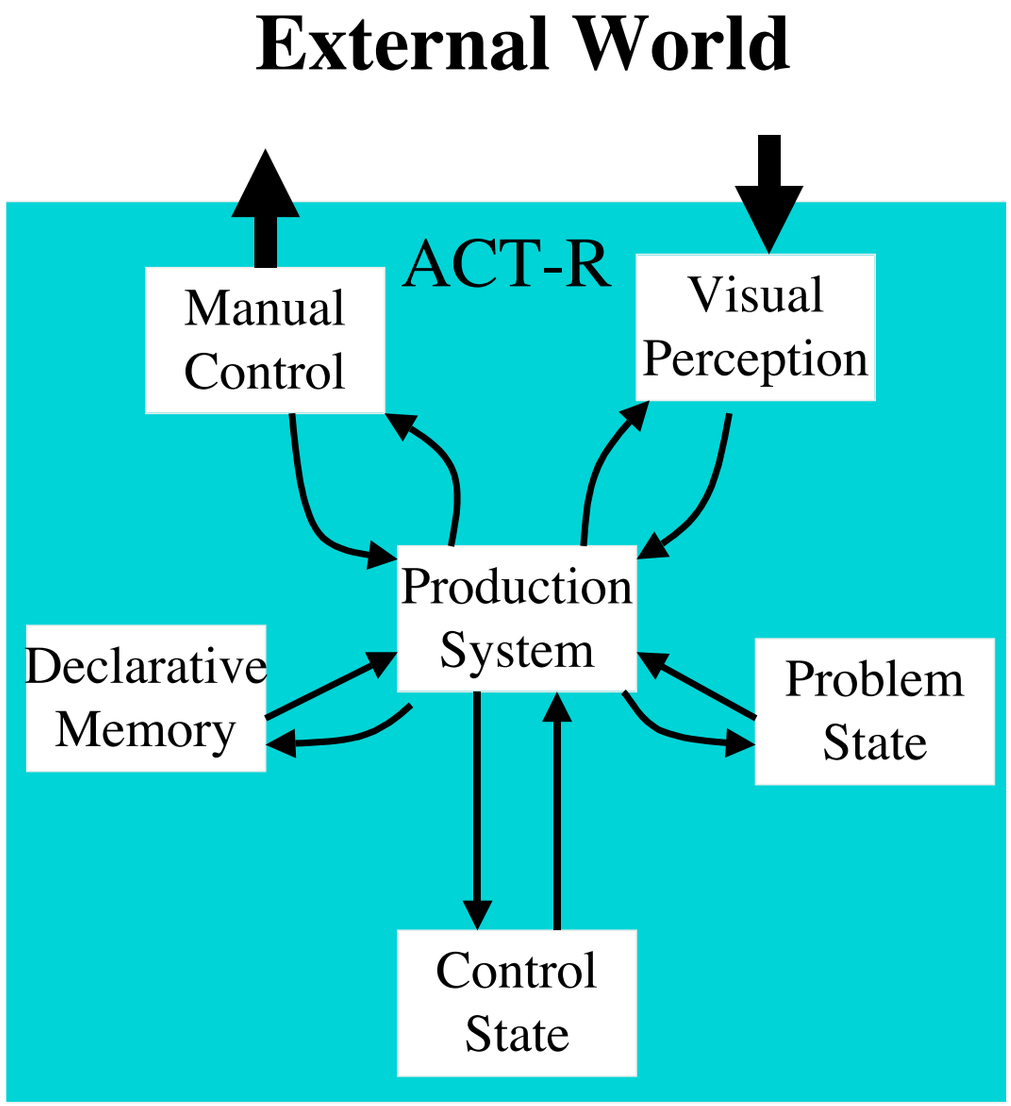
ACT-R

Within module: parallel and fast

Between modules: serial, slow, low bandwidth

> Everything flows through the production system

Production system can "learn" new rules

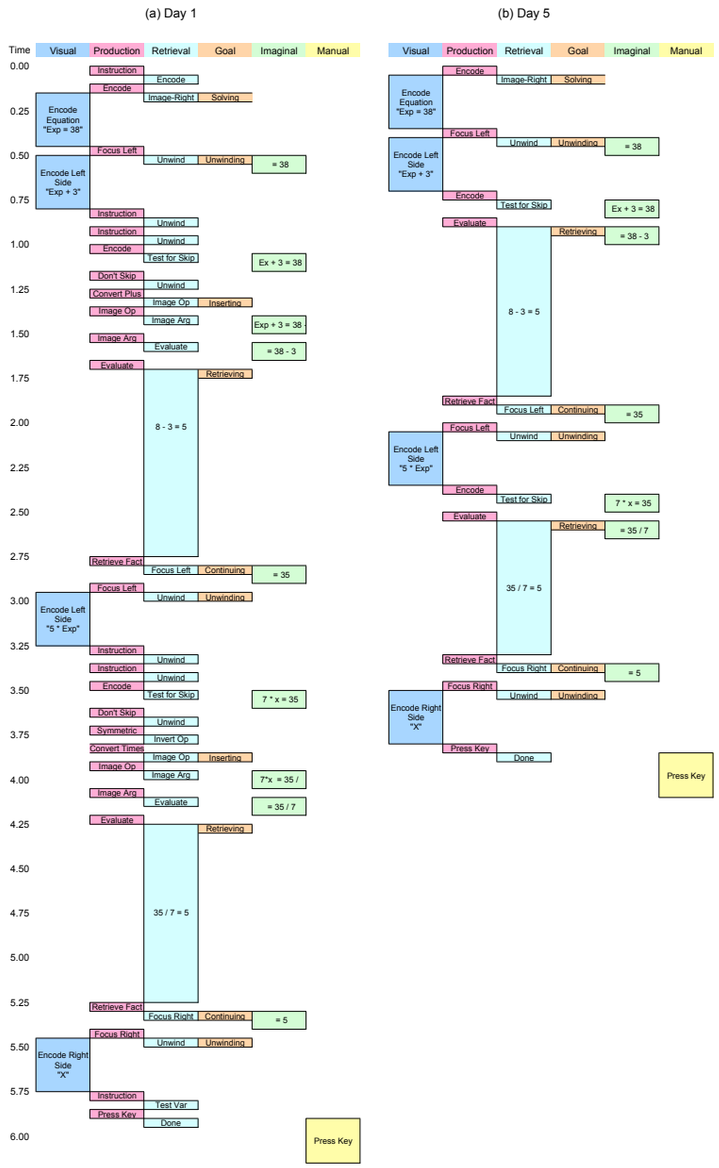


ACT-R

It models task performance pretty well

It's inspectable

It matches brain activity



Cognitive modeling

GOMS models: goals, operations, methods, and selection rules

- Quantitative prediction of **expert users'** interaction performance
- Based on measurement of human cognitive capabilities (see model-human-processor)

Advantages:

- No users needed
- Very accurate results

Keystroke modeling

Keystroke-Level models: Simplest GOMS-family member

No representation of goals, methods or selection-rules, just a sequence of operators that constitute a task

Input

A suite of benchmark tasks

A system design

Output

The time it would take a skilled user to perform the tasks

Construction

List the overt actions necessary to do the task

- Keystrokes and button actions (K), mouse movements (P), hand movements from keyboard to mouse (H)

Also system response time (if user has to wait)

- Insert mental operators (M)

Assign execution times from previous research

- K, M, H are straightforward; P requires Fitts's Law

Add up the execution times

Email login

Move mouse to input field	P	1.10
Click	K	0.20
Move hand to keyboard	H	0.40
Type username	8K	1.60
Move hand to mouse	H	0.40
Move mouse to input field	P	0.30
Click	K	0.20
Move hand to keyboard	H	0.40
Type password	12K	2.40
Move hand to mouse	H	0.40
Prepare for Log On	M	1.35
Move mouse to Log On	P	1.10
Click	K	0.20
Total		**10.05**

Keyboard shortcuts

Move mouse to input field	P	1.10
Click	K	0.20
Move hand to keyboard	H	0.40
Type username	8K	1.60
Press Tab	K	0.20
Type password	12K	2.40
Press Enter	K	0.20
Total		**6.10**

Pre-filled username

Move mouse to input field	P	1.10
Click	K	0.20
Move hand to keyboard	H	0.40
Type password	12K	2.40
Press Enter	K	0.20
Total		**4.30**

CogTool

Use an architecture to run the cognitive model!

Simulate the user using ACT-R

Benefit: more granular performance predictions

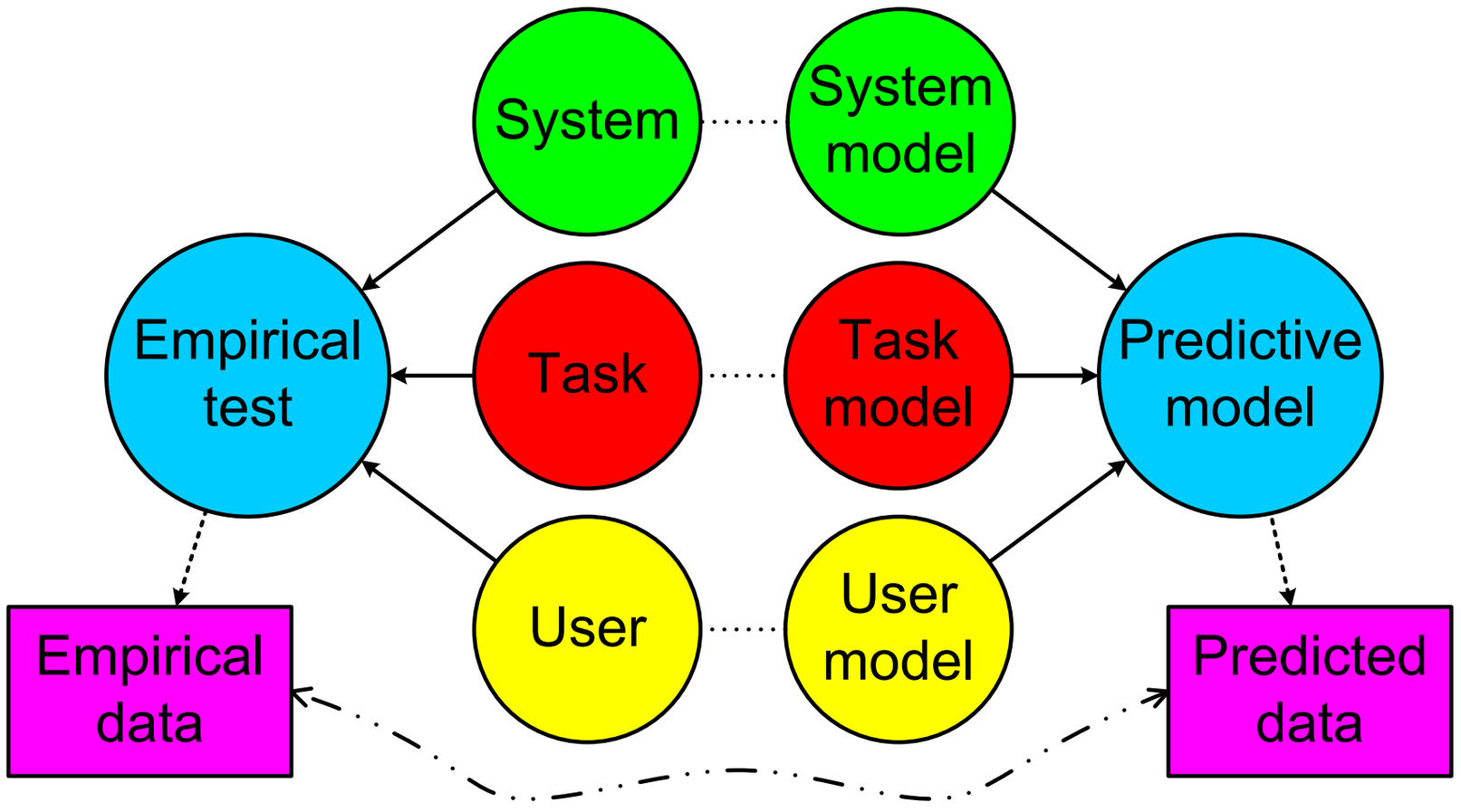
Simulate the interface

A mockup that ACT-R can “use”

Simulate the task

Construct a keystroke specification by example

Simulated test



Benefits

Let ACT-R do the specified task(s) on the provided interface(s)

- Outcome: Performance prediction
- Same characteristics as empirical user test
  - Added bonus: A breakdown for each step!
- Results reflect empirical test results (within 3%)
  - It takes about 5-10 minutes to build a model
  - Easy to tweak the interface and get immediate results

Discussion

When is this useful? When not?

How does KLM reason about the use image?

Beyond the brain

Distributed cognition

Beyond the brain

Criticism: Cognitive psychology doesn't work for HCI

Studies the mind outside the context of the real world

Solution: Create a conceptualization of cognition that works for HCI

External cognition: study the interplay between mind and interface

Distributed cognition: Study how cognition is shared among people, technology, environment

External cognition

Representations of information can be seen as external parts of one's cognition

E.g. diagrams versus text

> Diagrams are easier to process, because simultaneous information makes it easier to make inferences

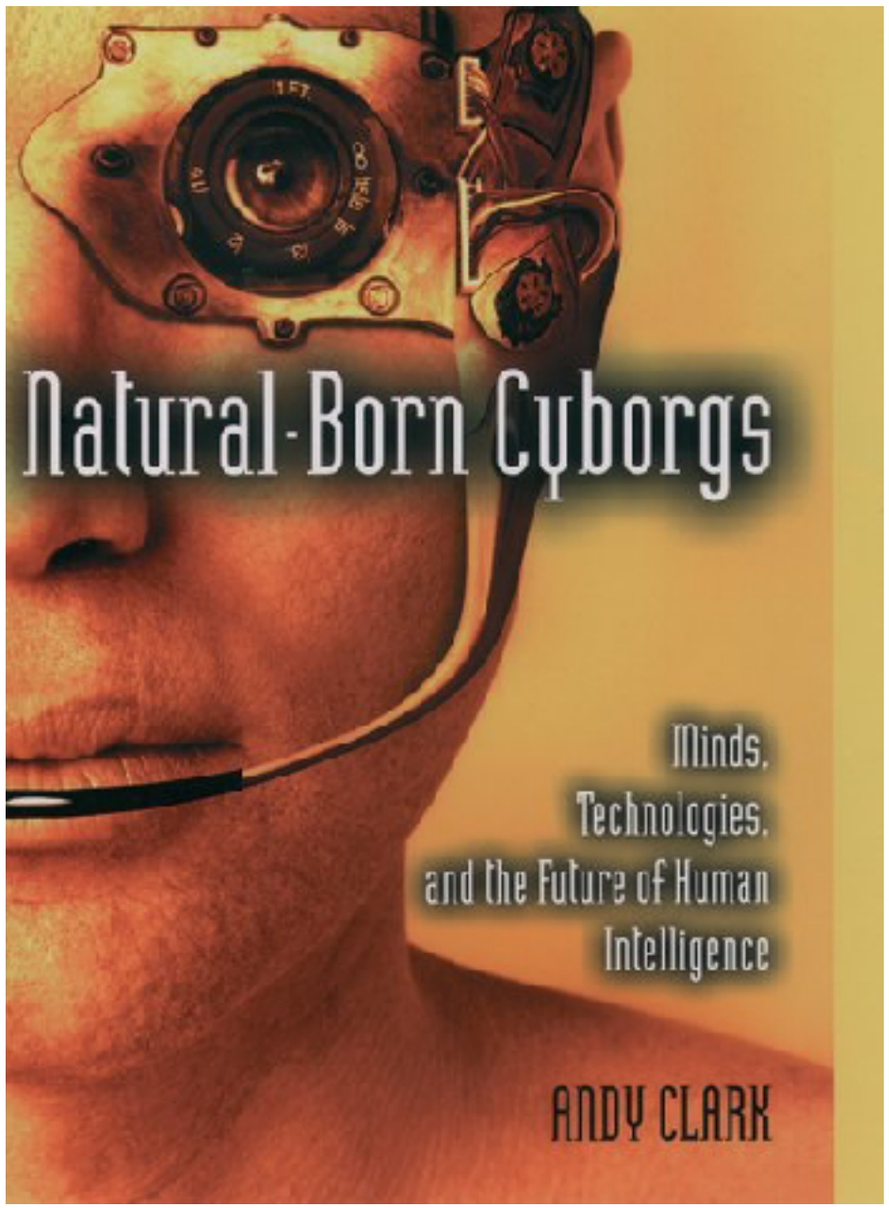
See Norman: Knowledge in the head vs. knowledge in the world

External cognition

Extended cognition:
"Scaffolding"

External manipulation as a method of "thinking"

An effective interface allows for a **structuring of external resources** that requires little reliance on internal resources in order to achieve one's goals



Distributed cognition

Combination of people, systems, and artifacts ***is*** a cognitive system

Why study cognition at this level?

- Only looking at the individual is a form of reductionism
- Studying the whole system is actually easier

How?

- Ethnography; study how information flows through a system at different levels of granularity

Distributed cognition

Assumptions of "DCog"

- An organization is a cognitive architecture
- Artifacts play an active role in cognition

Focus on:

- Planning and problem-solving
- Communication (both verbal and non-verbal)
- Coordination (rules, procedures)
- Knowledge creation and sharing (through artifacts, training, communication)

Flexible artifacts

Consider representations as both abstract forms as well as the thing that is being represented

- E.g. a form on my desk can be both a tool and a reminder

Most successful examples of DCog show how people exploit the flexibility of the digital world

- Look for secondary usage patterns

Memory as a process

Organizational memory resides in several individuals, objects and systems within an organization

- Both explicit and implicit

Memory can be viewed as both an entity and a process

- Memory processes are the transition of knowledge between humans and artifacts
- E.g. teaching a method, having a project meeting, assigning a task, writing down rules

Context

Knowledge transition happens through (mediated or direct) communication

Communication (especially when mediated by technology) results in reinterpretation and loss of context

- For efficiency reasons, the sender decontextualizes the information
- The receiver then has to recontextualize the information
- This process is not infallible, since contexts may be different for sender and receiver

Breakdowns

Result of this de- and re-contextualization? Breakdowns!

This makes it difficult to reuse knowledge

As a result, reuse is often limited to simple, familiar and frequently used pieces of information

Goal of a good information system: maintain context!

Questions

Can an organization have a goal? Or is it just the goal of its people?

How do organizations survive as a cognitive entity? How are their goal established and upheld?

How does learning occur? Does an organization have explicit and tacit knowledge?

Questions

What are good examples of breakdowns due to the de- and re-contextualization of information in knowledge transition?

How can we preserve context in these communications?

How would you build those ideas into a system?

Turn to the social

Situated Action

Turn to the social

Criticism: Cognitive psychology ignores social aspects of HCI

Solution: bring in sociologists and anthropologists

Ethnomethodology: Study HCI as social phenomena

Situated Action: examine the social context in which HCI occurs

Ethnomethodology

Ethnography: a method of studying people that involves immersing oneself in their world

Ethnomethodology: studying people with the purpose of understanding how they make sense of the world

Not a theory but an approach

Bottom-up, sometimes anti-theoretical

Careful observation exposes taken for granted work practices that turn out to be key in (re)designing the system

Situated Action

A highly detailed account of the actual interactions between people and the world they inhabit

Why study HCI like this?

Lots of HCI is informal or unstructured

Structure is an **outcome** of an orderly process, not a **condition**

Goals are retrospective reconstructions of what happened

Situated Action

How? Mostly behavioral methods:

- Record behaviors and conversations
- Following users around to study their actual movements
- Trace artifacts
- Capture interactions (e.g. screen recording)
- Study the same tasks in different contexts

Don't trust:

- What people plan to do (only use it for comparison)
- What people say they do (use real observations)

Situated Action

Assumptions of Situated Action:

Actions are constrained and supported by social and physical circumstances

People use these circumstances to achieve their goals

Humans are pulled to the artifact side

Focus on:

- Regularities and irregularities across contexts
- Deviations from and adherences to protocols, and their reason

Abstractions

Distributed coordination

How are tasks divided? Does this happen ad hoc or by plan?

Plans and procedures

Compare against real actions: do they allow procedures to take hold? If not, why not?

Awareness of work

How actions are communicated or made visible to others?
One person's action is another person's context

Outcomes

Result: An account of how technology is actually used, contrasted with how it is supposed to be used

- Plans may change due to the situation!

Practical result: Make technology fit the work practice, rather than the other way around

- Situation enables and constrains knowledge and action
- Embrace the inherent ambiguity of work, thereby creating a tool for doing the work

Questions

Can you give an example of an interaction that didn't go according to plan because of the situation?

How would you support such interactions? Context-awareness? Flexible systems?

How much are our goals dictated by the situation?

How regular is our behavior? At what level?

Is that level appropriate for HCI?

How does interactional learning occur if actions are situated?

Activity Theory

...and a comparison

Activity Theory

Treat plans as **anticipatory reflections** of recurring activity

Not fully generative, but also not mere descriptions

Because plans and activities start out as external and collective, **culture** and **society** transform all our activities, and in turn, our minds

> Hence, plans and activities are socially constructed, and may evolve in the course of action (short term) and over time (long term)

Activity Theory

How? Explain a practice based on:

- The motives behind the activity (Why do I want to be in grad school?)
- The goals the actions (Why do I take this class)
- The orienting basis of the operations (Is what I'm doing right now helpful in getting where I want to be?)

These levels are transient

Artifacts

We employ internal and external **resources** to perform our activities

> Human-computer interaction is framed as the use of external resources (artifacts) as a means of mediating an activity

We can use artifacts to:

- perform the operation
- control the task at hand
- coordinate the activity

Artifacts

In social settings, we can also use them to:

- manage our community
- adhere and implement rules
- divide labor

Artifacts

The field of HCI can study the socio-cultural practice of learning to use and using artifacts…

…to support operations, actions, and activities

Good systems support **full activities** rather than just actions or operations

How? By implementing (rather than ignoring) the plans

But it has to be done flexibly!

A comparison

Let's compare Distributed Cognition (DCog), Situated Action (SA), and Activity Theory (AT) in terms of their:

- treatment of user goals
- treatment of humans and artifacts
- opportunity for generalization
- overall merit

User goals

DCog: The system (a combination of subjects and artifacts that together perform a task) provides the goal

SA: goals are retrospective reconstructions of what happened; the situation is the driving factor

AT: Goals exist at several levels, but originate from the subject's intentionality

Humans v. artifacts

DCog: Artifacts are pulled to the human side, and assigned cognitive capabilities

SA: Humans are pulled to the artifact side; they are reactive ciphers that react to stimuli in a behaviorist manner (controlled by the situation)

AT: Humans control their activities; artifacts are just the mediators these activities

Generalizations...

DCog: ...are the result of analyzing the collective manipulation of artifacts, and the transformation of representations as they permeate through the system

SA: ...do not happen, due to the idea of moment-by-moment analysis (but less purist versions exist)

AT: ...can occur by looking at the historical development of activities and the artifacts that exist as mediators between subject and activity

Overall merit

DCog: Provides a formal analysis of artifacts and how they are used, and produces comparative data across settings

SA: Acknowledges the fluidity of goals and plans, but the exclusive focus on the situation may reduce its usefulness

AT: Like DCog, but treats consciousness at the individual level; situation influences but does not determine the actions