Norman's Theory

Fundamentals of Human-Centered Computing

Norman's Theory

A classical theory of Human-Computer Interaction

One of the first popular accounts of HCI

Today I will cover:

- Some background
- The action cycle
- Memory and mental models
- Design using constraints, signifiers and feedback

Some background

About the early days of HCI

Some background

In the early 1980s, there was some optimism that cognitive psychology could inform HCI

> Frederic Bartlett (1932): "Cognitive research should have relevance to the real world"
>
> Donald Broadbent (1980): "Real-life problems should [...] ideally provide the starting

It turned out not to be that easy...

Problems

Research is inadequate or too general

- Or problems too specific
- Going from general to specific is difficult!

Other forces apply

- Policy and Social Science

Seat-of-the-pants solutions looked science-y but weren't

- Increased skepticism towards potential applications

Cognitive Psych

Theoretical approach

Directive tests

Theoretical issues

- No common understanding (yet)
- Will there be one?

"I'm a college professor, Jason. You need to ask someone else if you want advice about the real world."

Reprinted from Funny Times / PO Box 18530 / Cleveland Hts. OH 44118
phone: 216.371.8600 / email: ft@funnytimes.com

Theoretical issues

There are many ways to investigate the same thing

There is no 'best practice'

Results may contradict each other and allow for different interpretations

Result: most important areas are in disagreement!

Attention (early vs. late selection)

Memory (connectionism vs. classical models)

Representation (pictures vs. words)

Artificial Intelligence (real intelligence vs. fake simulation)

Bridging the gaps

Cognitive scientists and engineers:

- Do not pursue the same goals
- Do not speak the same language

Contradictions stand in the way of a decent cooperation

How to resolve these issues?

- By building a bridge between the disciplines
- By translating the findings of cognitive science into applications

Applied science

How to go from basic research...

Spatial cognition

...to applied research...

Understanding of maps

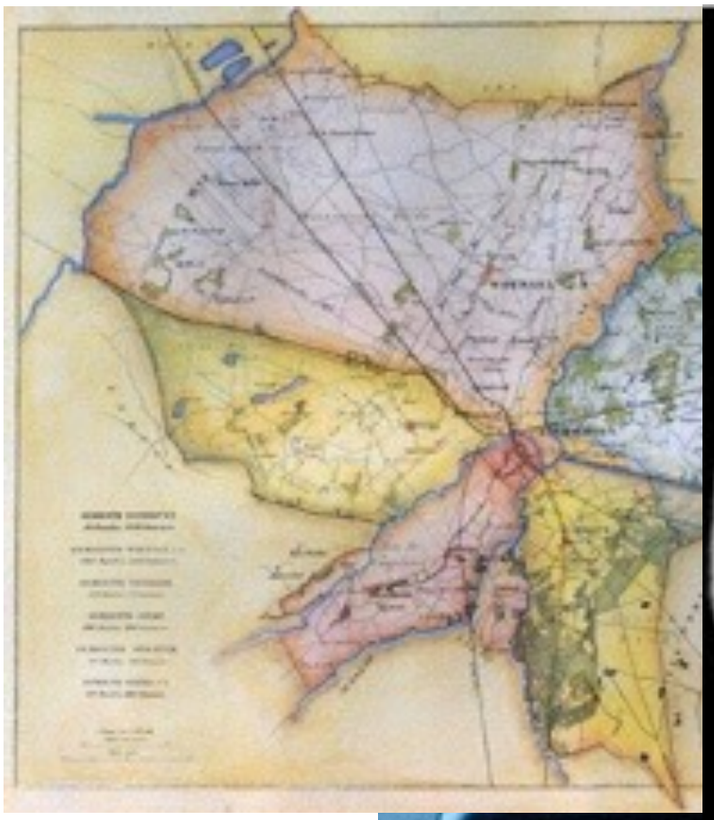
...to application?

New navigation device

Research necessary at every step

Lab studies, field studies, usability studies

Interpretation needed to move to the next level



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

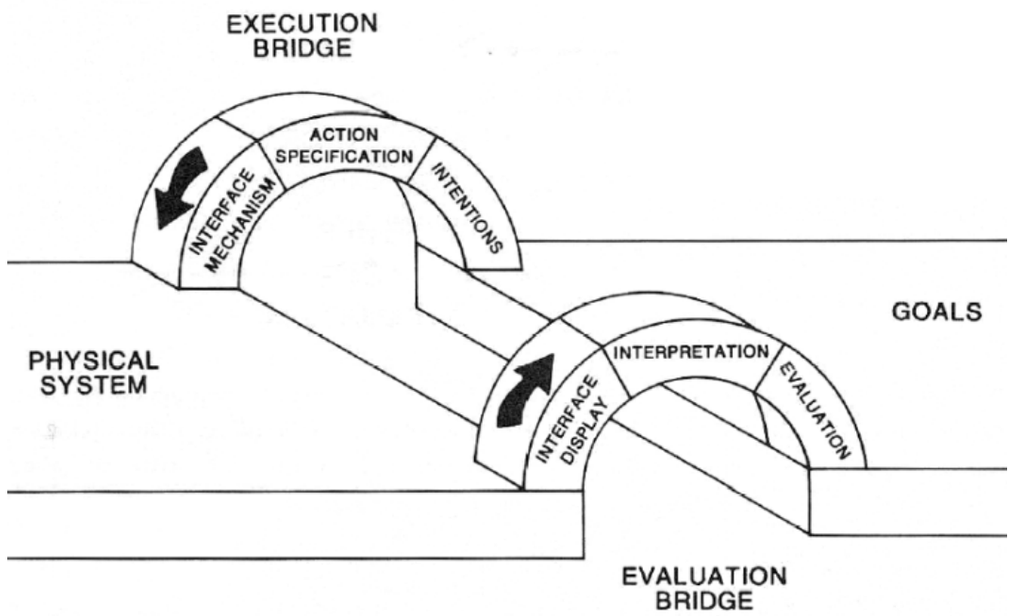
The action cycle

How people interact with computers

The action cycle

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

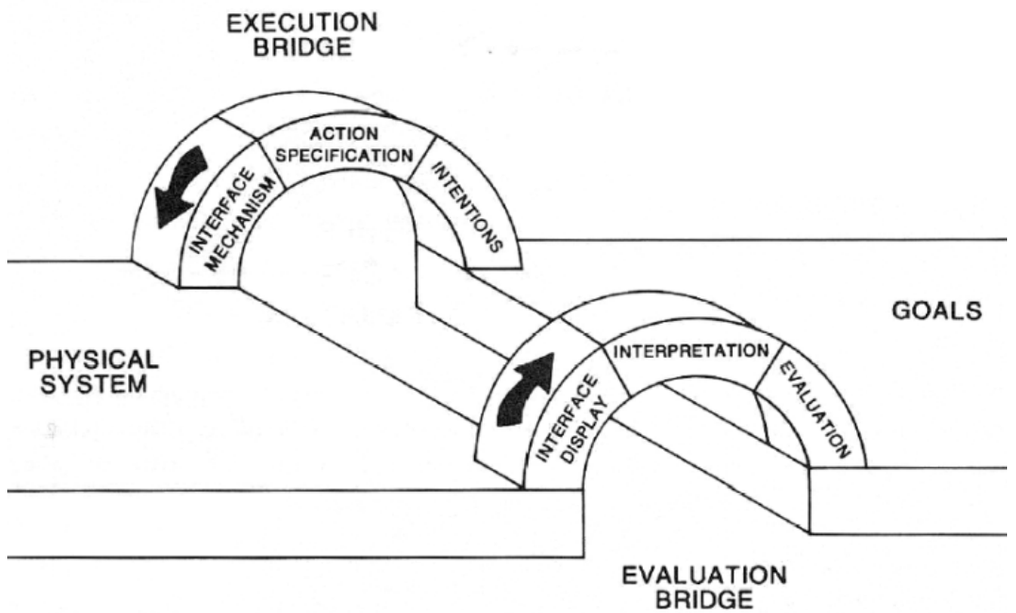


Example

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

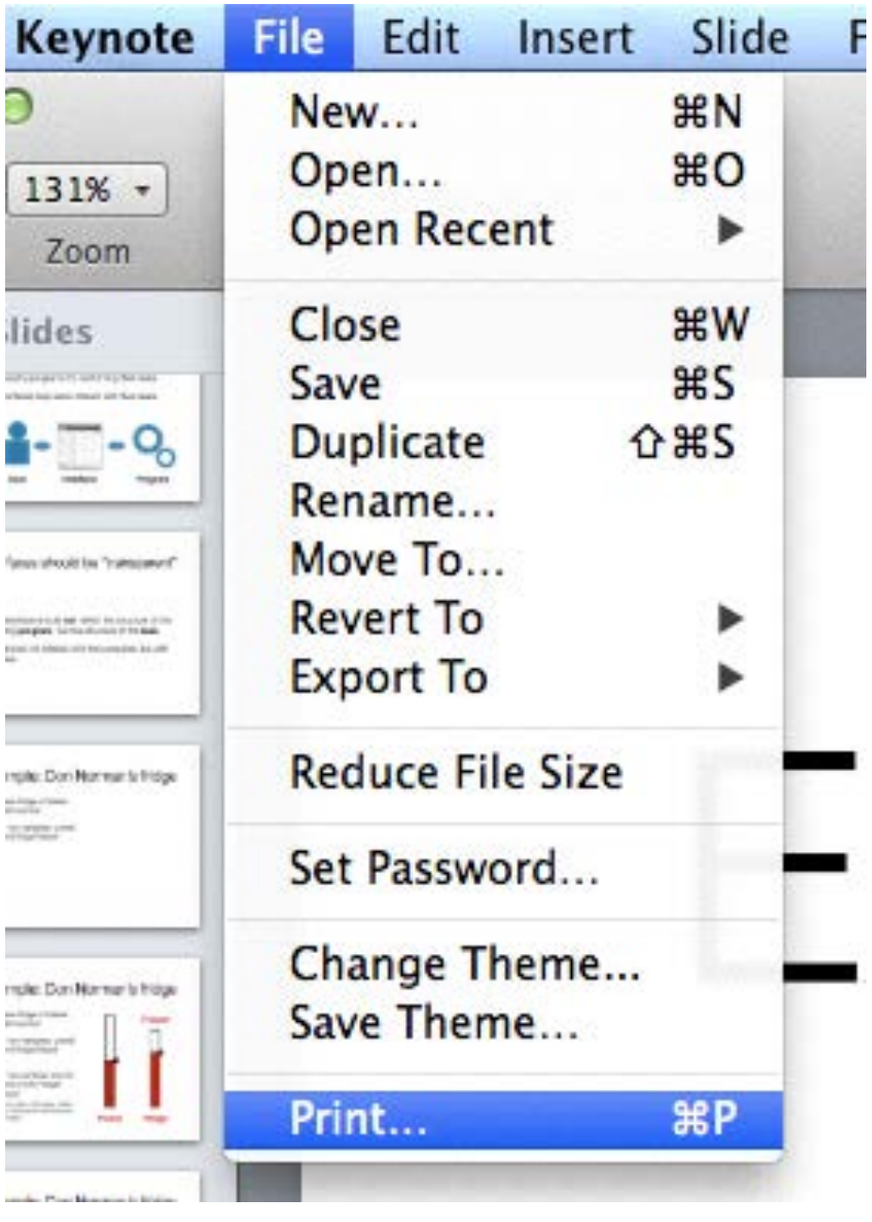
use my home printer to print the email

2. Specify an action sequence

click File > Print

3. Perform this sequence

<click>



Example

4. Perceive the change

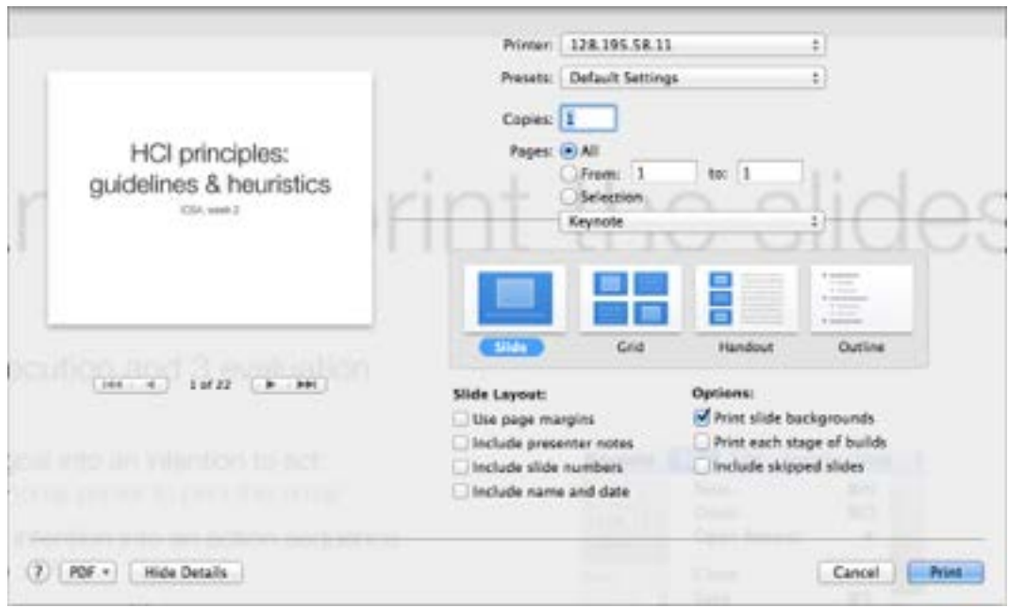
this causes a dialog to pop up...

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

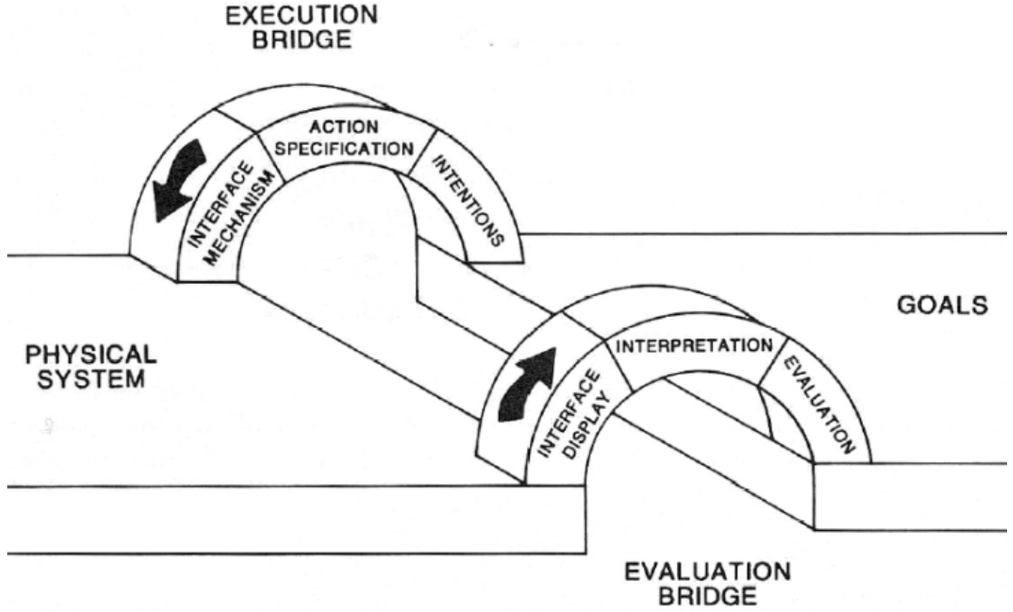
Things that can go wrong in the **evaluation**-part:

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?



Mental models

How people think about computers

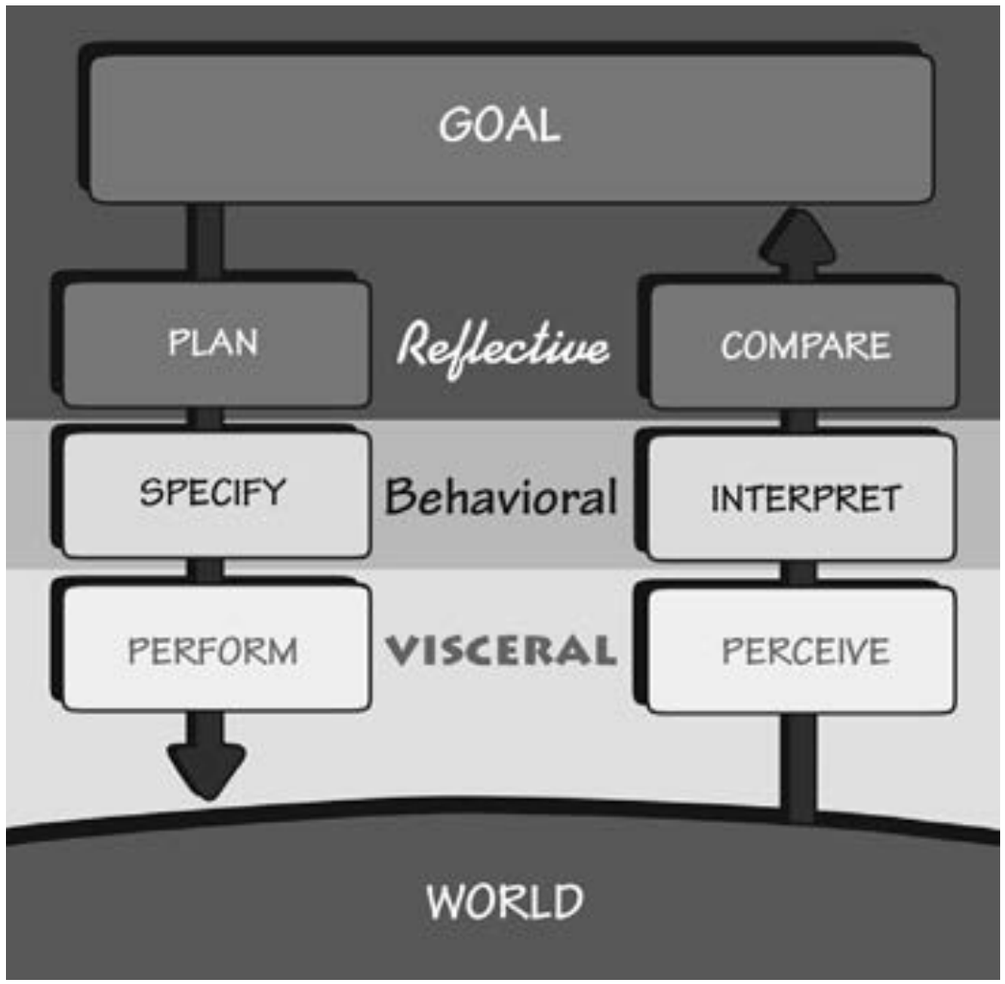
Mental models

Users' brain is involved in the action cycle at three levels

- Viceral
- Behavioral
- Reflective

Reflective processes are the most complex

Errors usually happen here

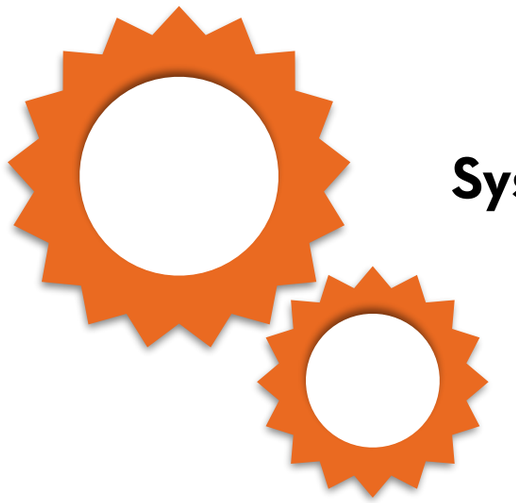


Images

Both designers and users reason about the system

- Designer image: how the designer thinks the system should work
- System image: how the system actually works
- Use image: how the user thinks the system works

Program

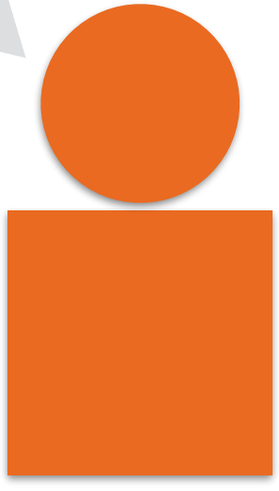


System image

Designer image

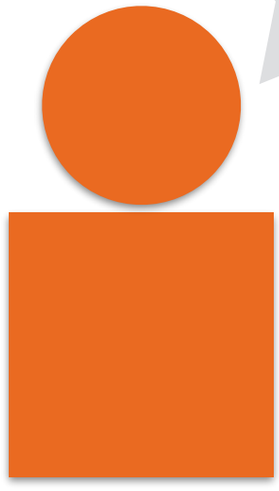
operate

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!

Discussion

How can we reason about mental models?

How can mental models be measured?

Does technology change our memory?

How can we align use image and system image?

Usable interfaces

Constraints, signifiers, and feedback

User interface

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

Good example: juicer

Bad example: iMac drives

Good solution: European ATM (forcing function)

Constraints

Physical constraints: object can only be used in one way

Good example: juicer

Bad example: iMac drives

Good solution: European ATM (forcing function)

Constraints

Physical constraints: object can only be used in one way

Good example: juicer

Bad example: iMac drives

Good solution: European ATM (forcing function)

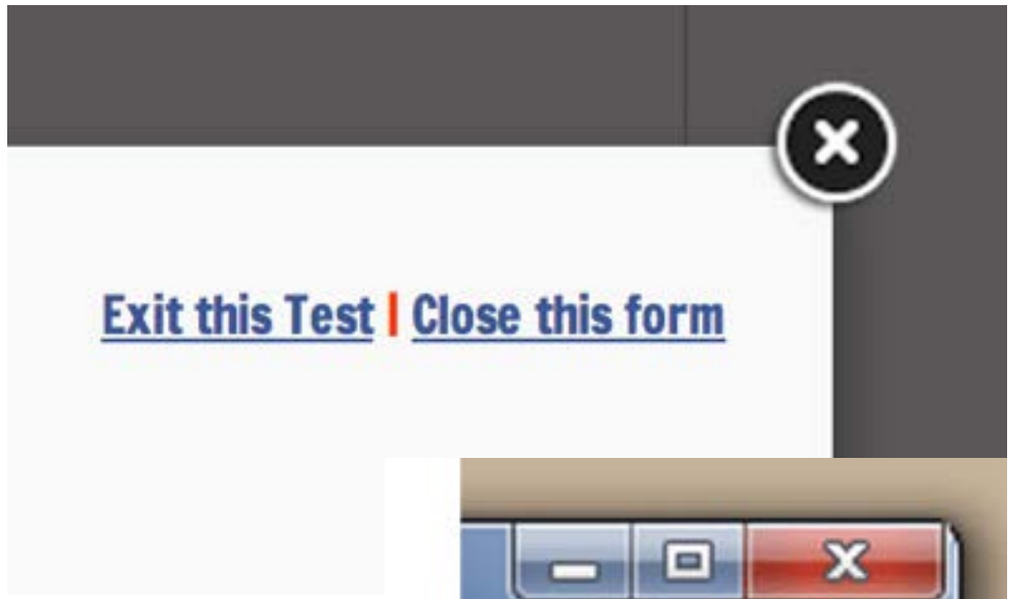
Constraints

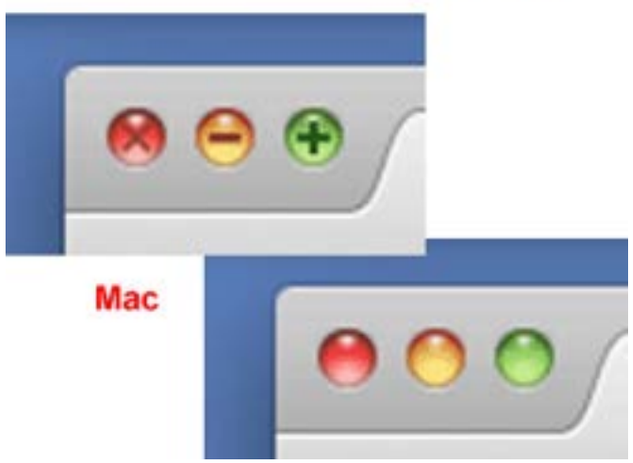
Cultural constraints: use is culturally determined

Good example: close button

Bad example: tipping

Funny problem: iPad babies





Constraints

Cultural constraints: use is culturally determined

Good example: close button

Bad example: tipping

Funny problem: iPad babies

Constraints

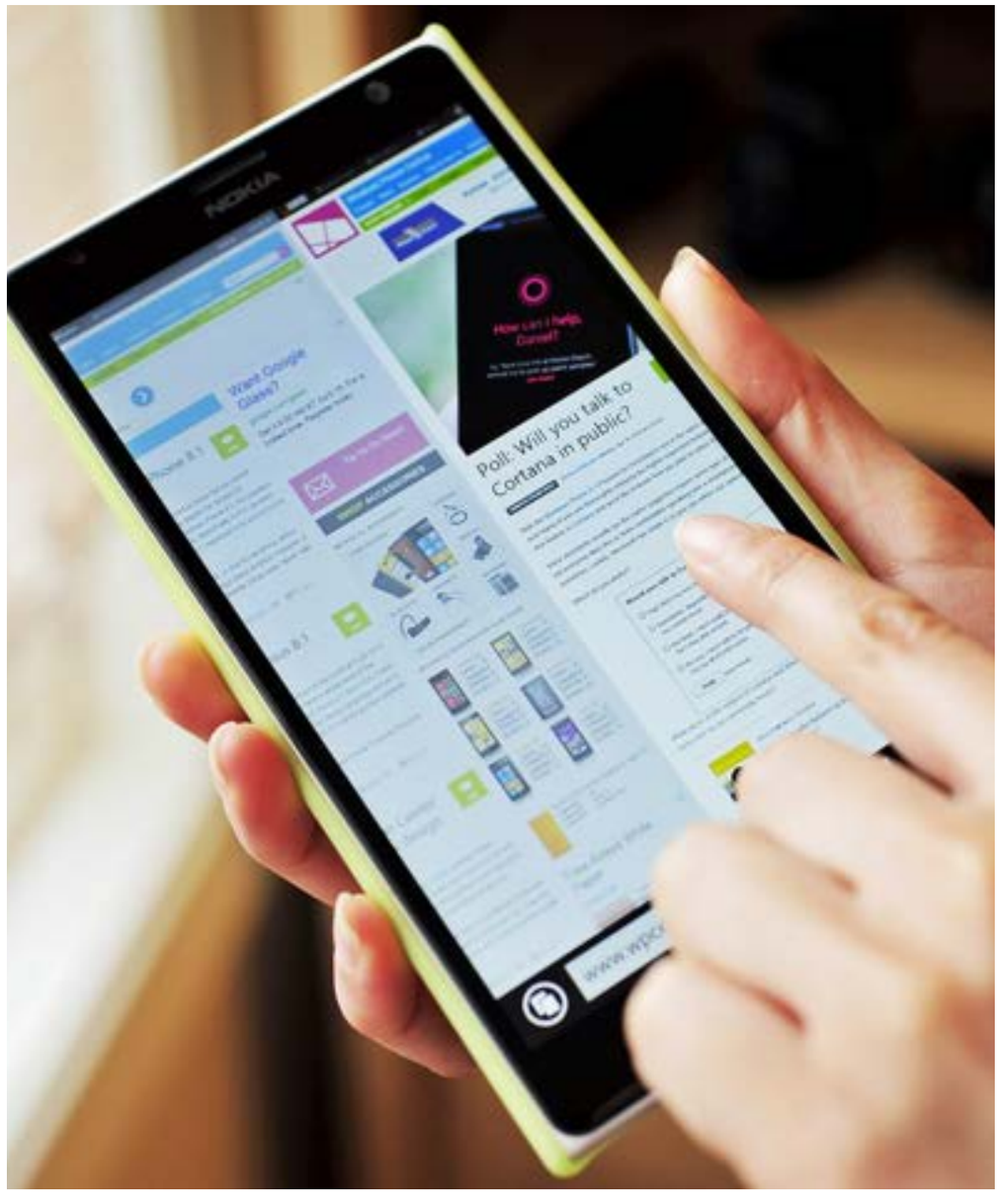
Cultural constraints: use is culturally determined

Good example: close button

Bad example: tipping

Funny problem: iPad babies

Constraints



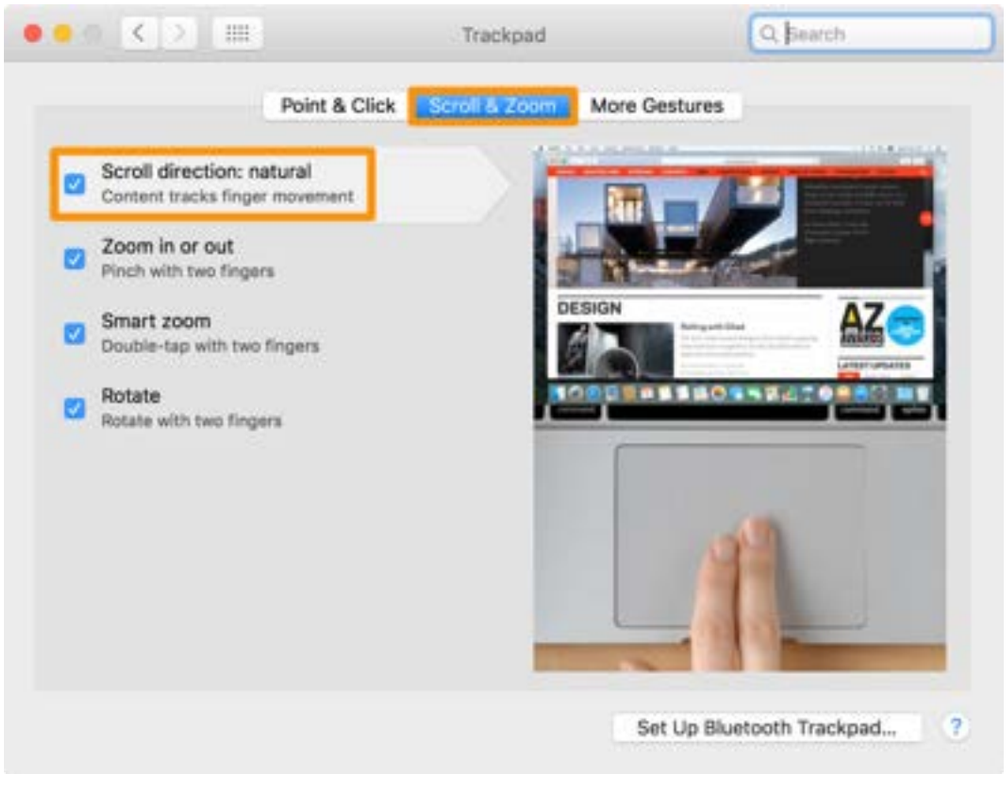
Semantic constraints: use is determined by the situation

Good example: swipes on smartphones

Bad example: scrolling

Good solution: force touch

Constraints



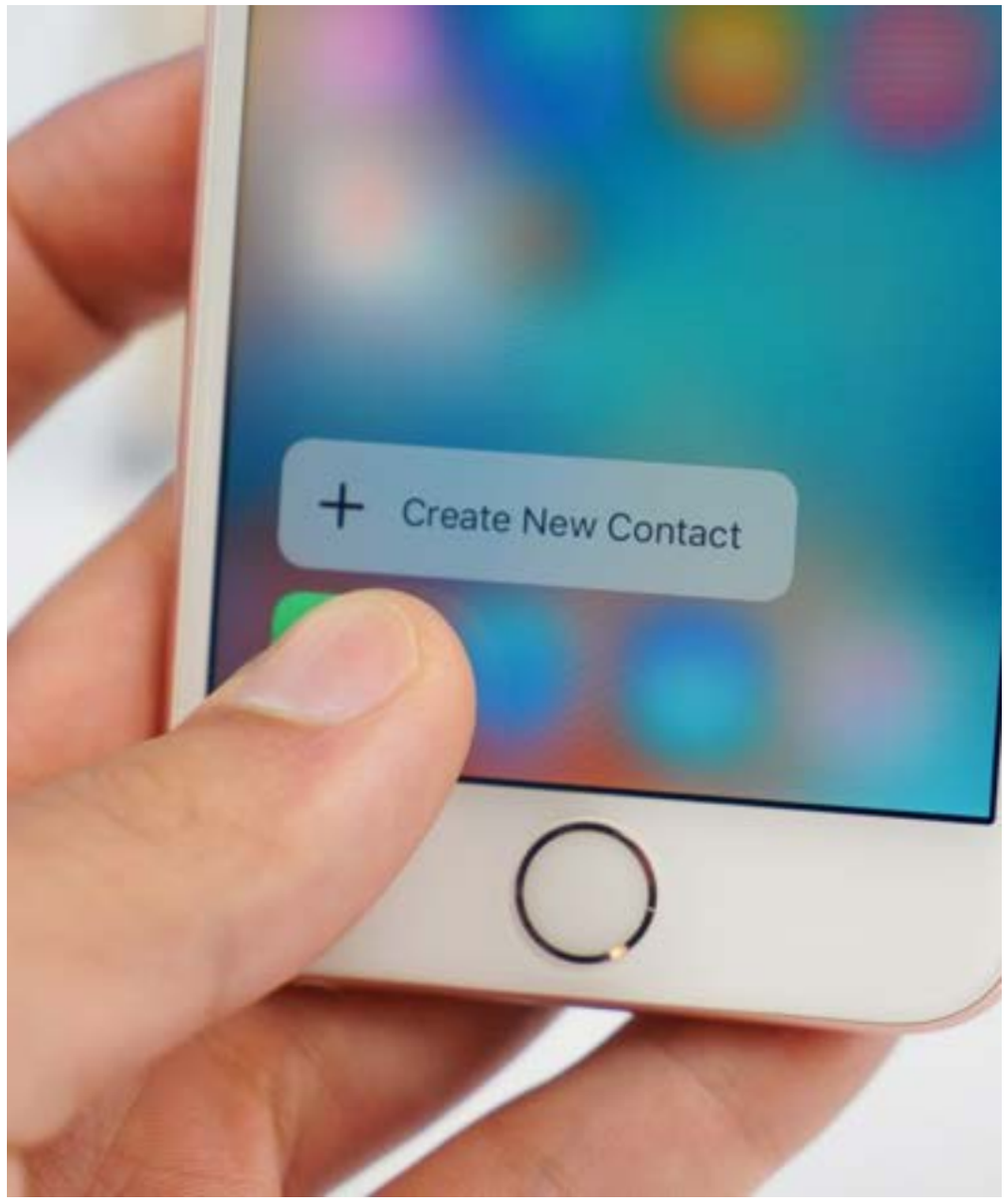
Semantic constraints: use is determined by the situation

Good example: swipes on smartphones

Bad example: scrolling

Good solution: force touch

Constraints



Semantic constraints: use is determined by the situation

Good example: swipes on smartphones

Bad example: scrolling

Good solution: force touch

Constraints

Logical (natural) constraints:

Good example: natural mapping

Bad example: many light switches

Good solution: physical answering machine

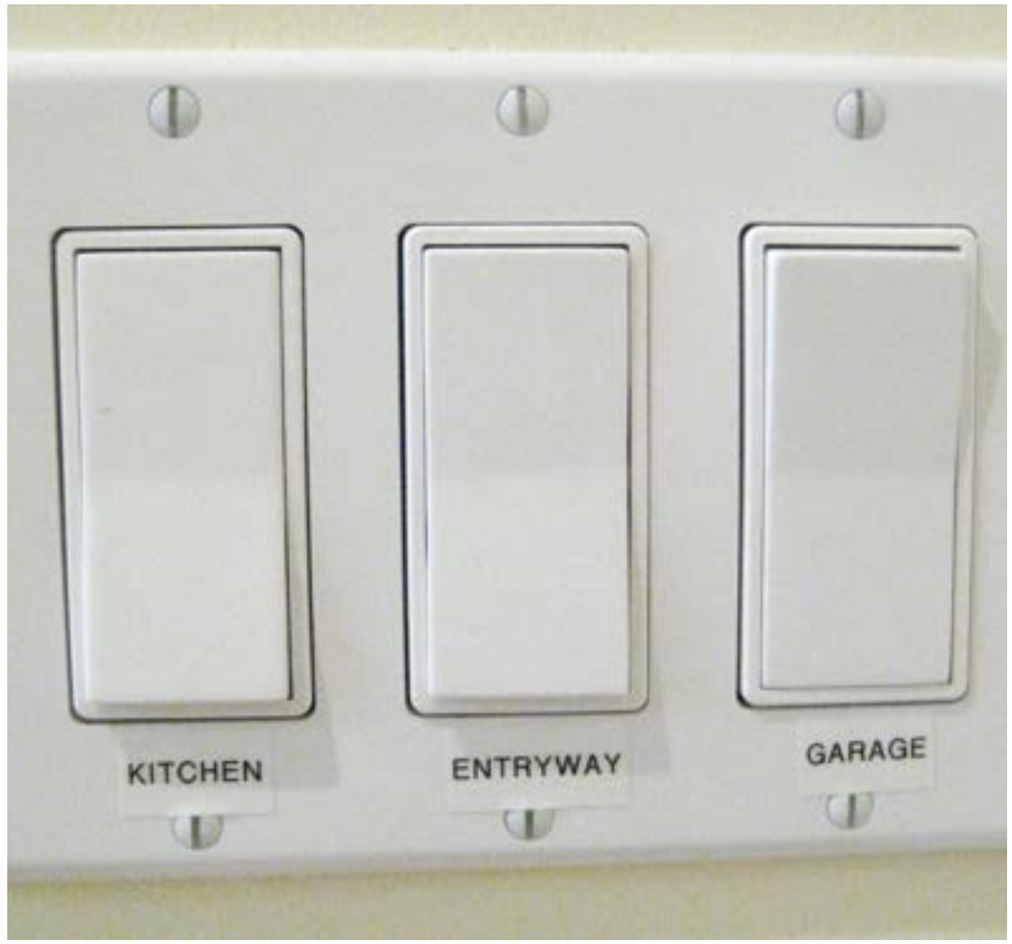
Constraints

Logical (natural) constraints:

Good example: natural mapping

Bad example: many light switches

Good solution: physical answering machine



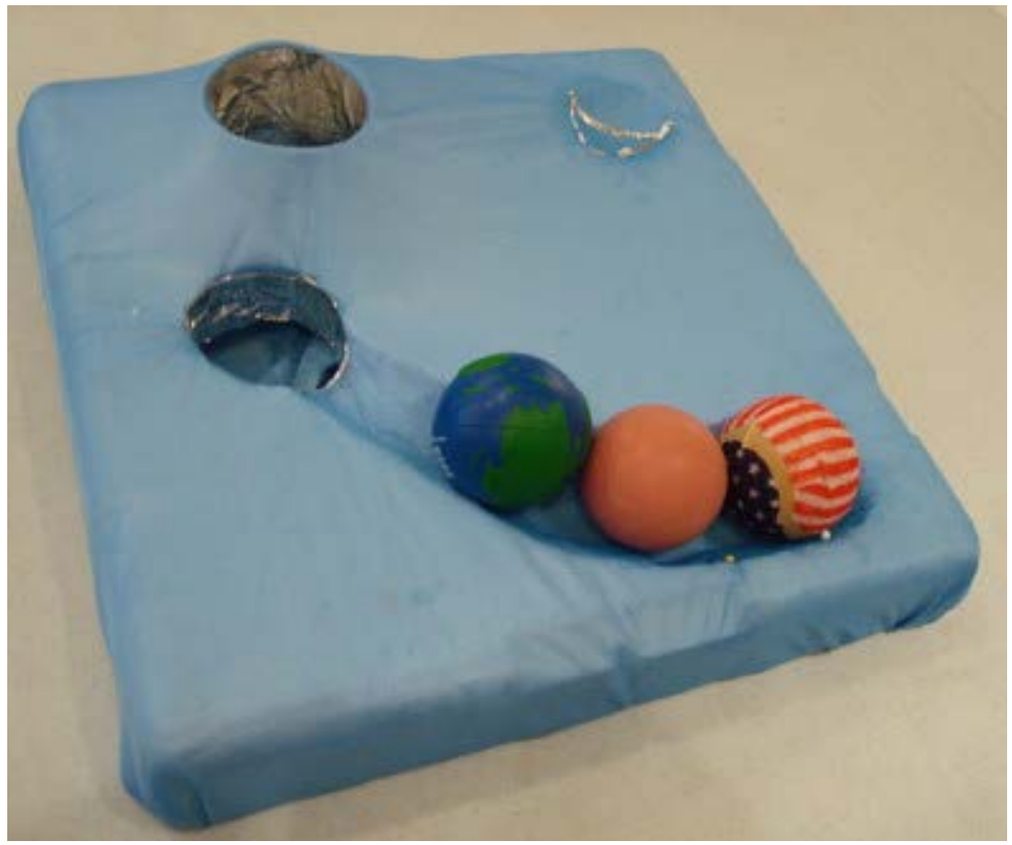
Constraints

Logical (natural) constraints:

Good example: natural mapping

Bad example: many light switches

Good solution: physical answering machine



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

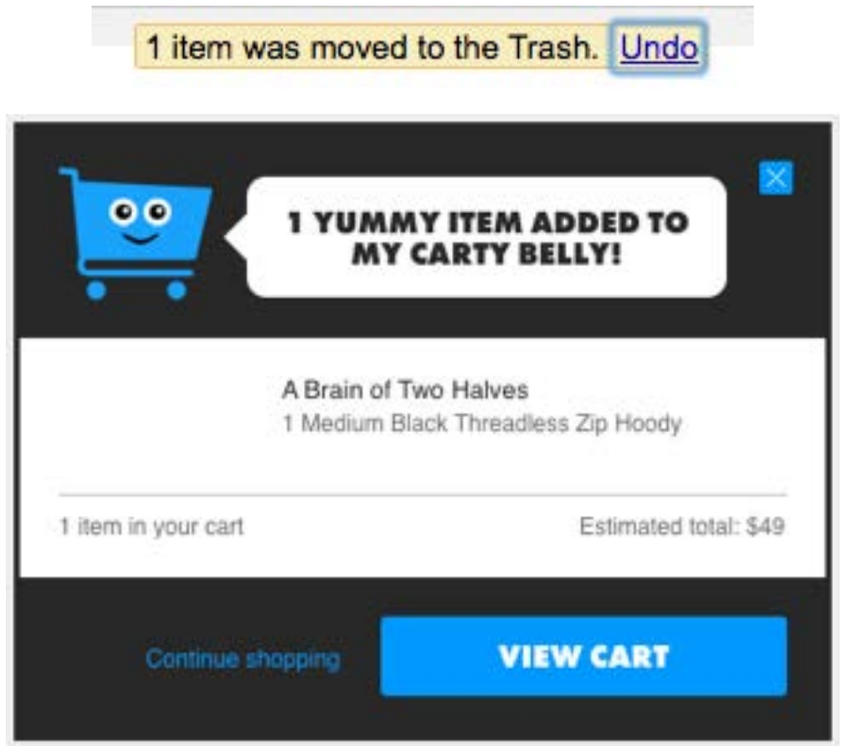
Examples:

"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 skeuomorphism?

- Do they work?

What are good examples of constructed signifiers?

- Why do they work?

How about agent-based interfaces?

- How can we give feedforward and feedback there?