Design Theories: Conceptual Models

Chapter 12

The slides in this lecture are partially based upon this from Ms. Tiffany Tang and Drs. Vincent Ng and Saul Greenberg.

Overview

- What you will learn in this lecture:
  - What do people think when they see a design?
  - Why do people use applications/devices wrongly?
  - Why are some devices easier to use/learn than others?

Use and Context

Human Social Organization

Human-Machine Fit and Adaptation

Applications

Human

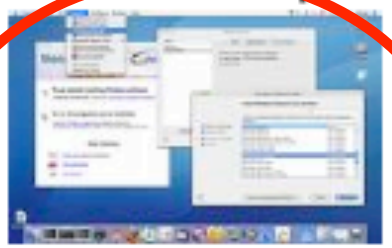
Human Information Processing

Language, Communication, Interaction

Ergonomics

I/O Devices

Computer



Interface Metaphors

Graphic Design

Dialogue Techniques

Prototypes

Evaluation Techniques

Implementation Techniques and Tools

Design Approaches

Development Process

Conceptual Models

- Many so-called “human errors” are actually errors in design.
- Human factors become important as machinery gets ever more complex and start to push performance limitations.

Conceptual Models

- When people see a new object, they form a *model* of what it is, and how it works.
- Based on past experience and the object's design.
- Extrapolates to how people view the interface controls and how to use them to affect the object.

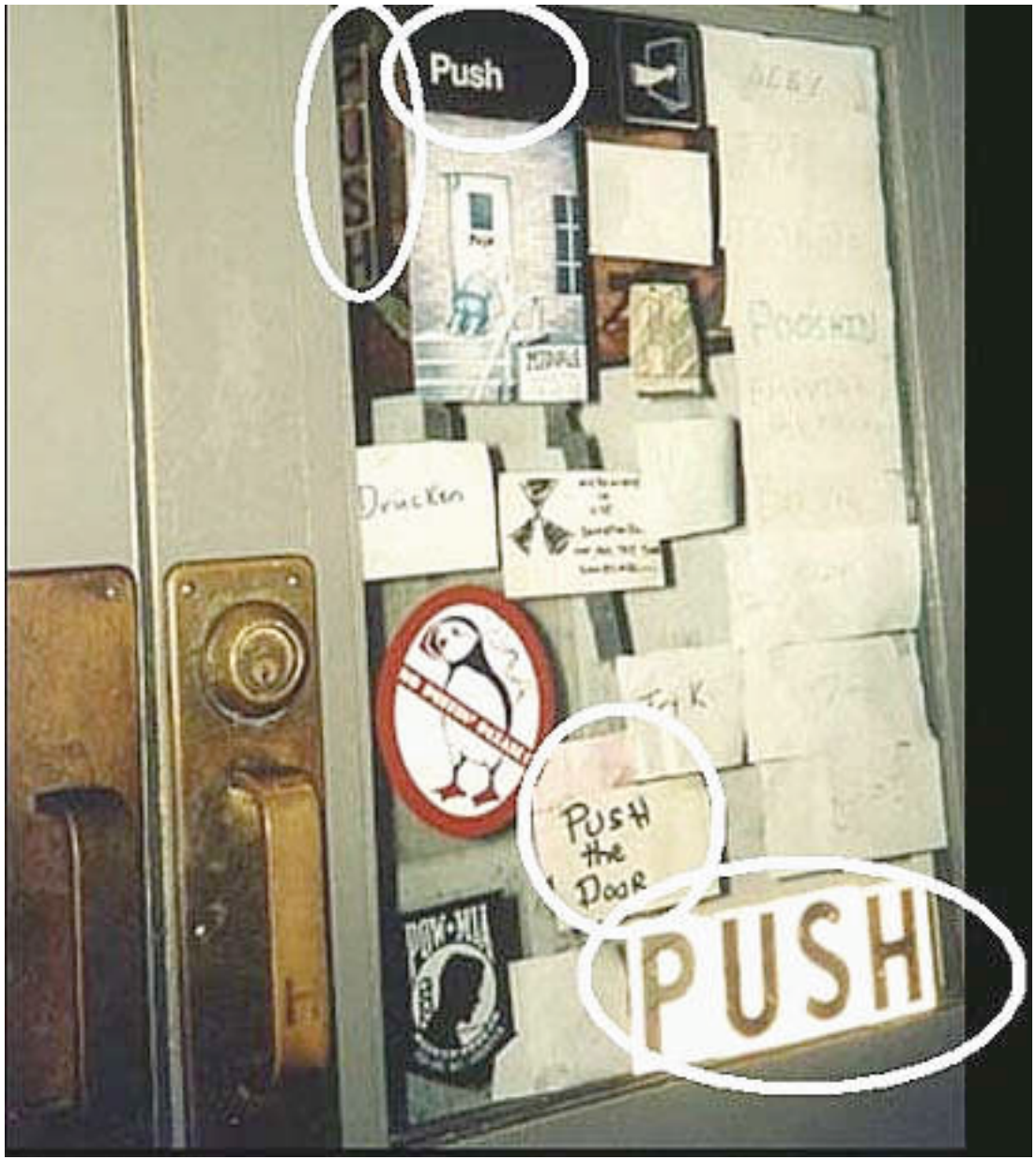
Let's take an example

- Only two things you can do to it:
  - Open it, or shut it.
- How do you work this?

How about this?
How do you
work it?

This door is obviously not communicating its controls too well…

How could it have been better designed?

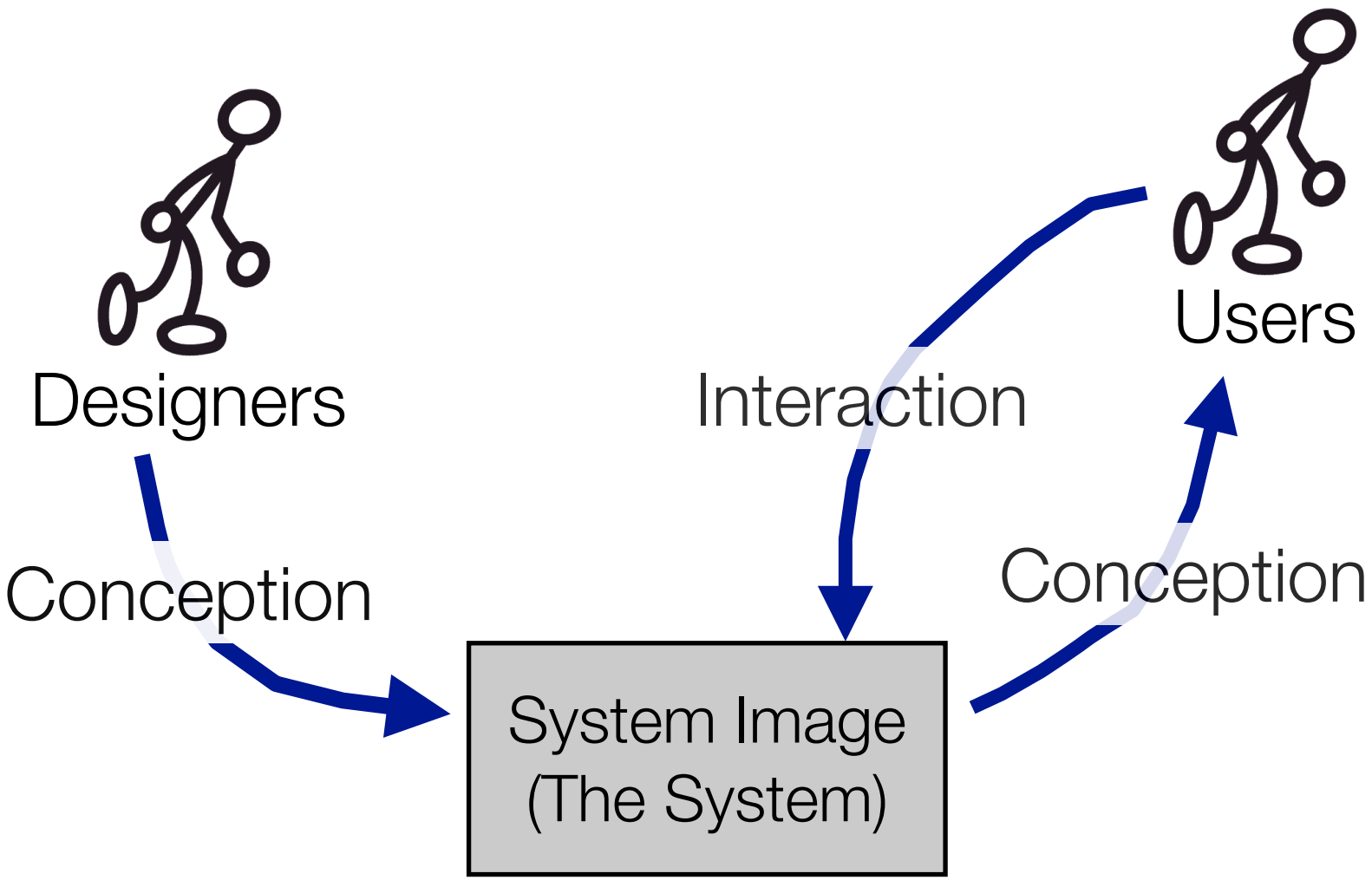


This also applies to software products!

Design Models and User Models

- Users and designers communicate through their mental models of the product.
- Designer's model = mental/conceptual model of the system.
- User's model = mental model developed through interaction with the system.
- Designer expects that user's model is the same as the designer's model.
    - E.g. I designed the system to use Ctrl-P for paste, because "P" stands for "paste", which is intuitive.
- But often it isn't intuitive to the user!

Design Model and User Model



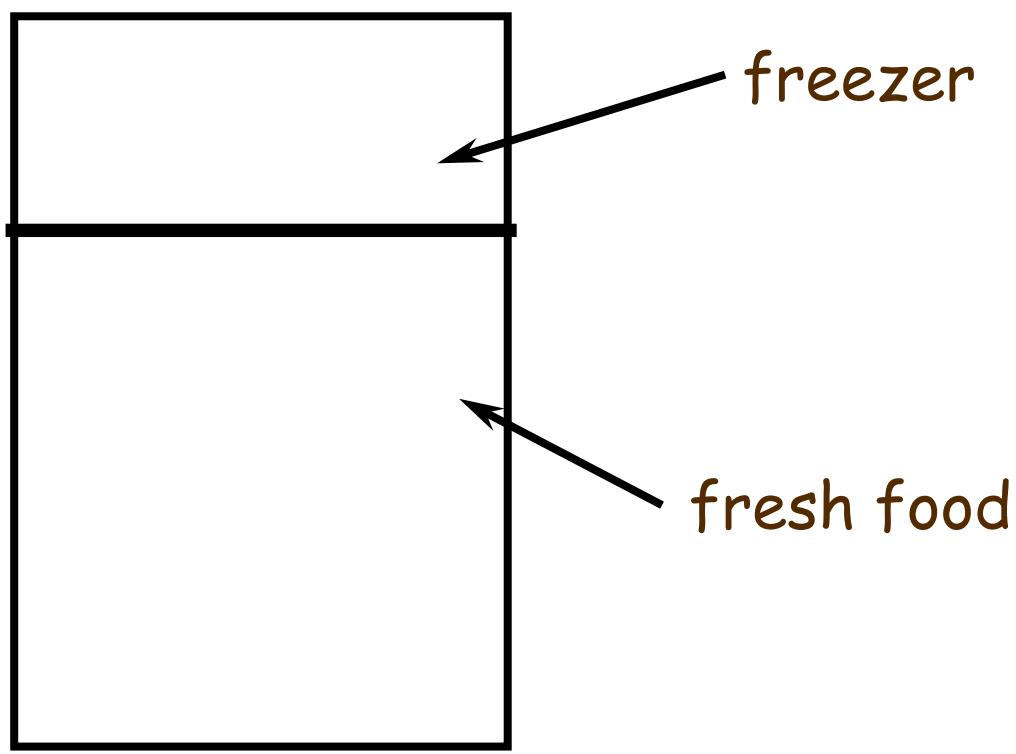
How do things go wrong?

When Conceptual Models Mismatch

- Given the same system, we have the following models:
  - Designer's Model
  - User's Model
- What happens when the models don't match?
  - Error!
  - Slowness!
  - Frustration!

Case Study: Old Refrigerators

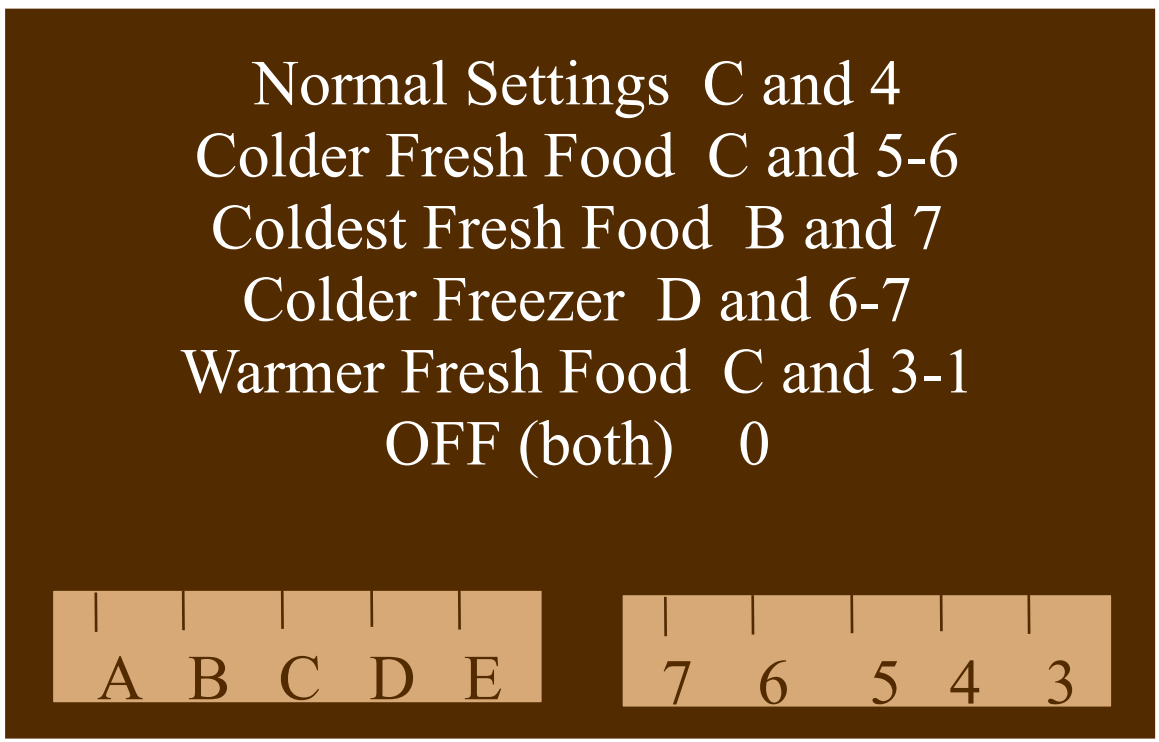
- Problem: Freezer too cold. Fresh food just right.



Refrigerator Controls

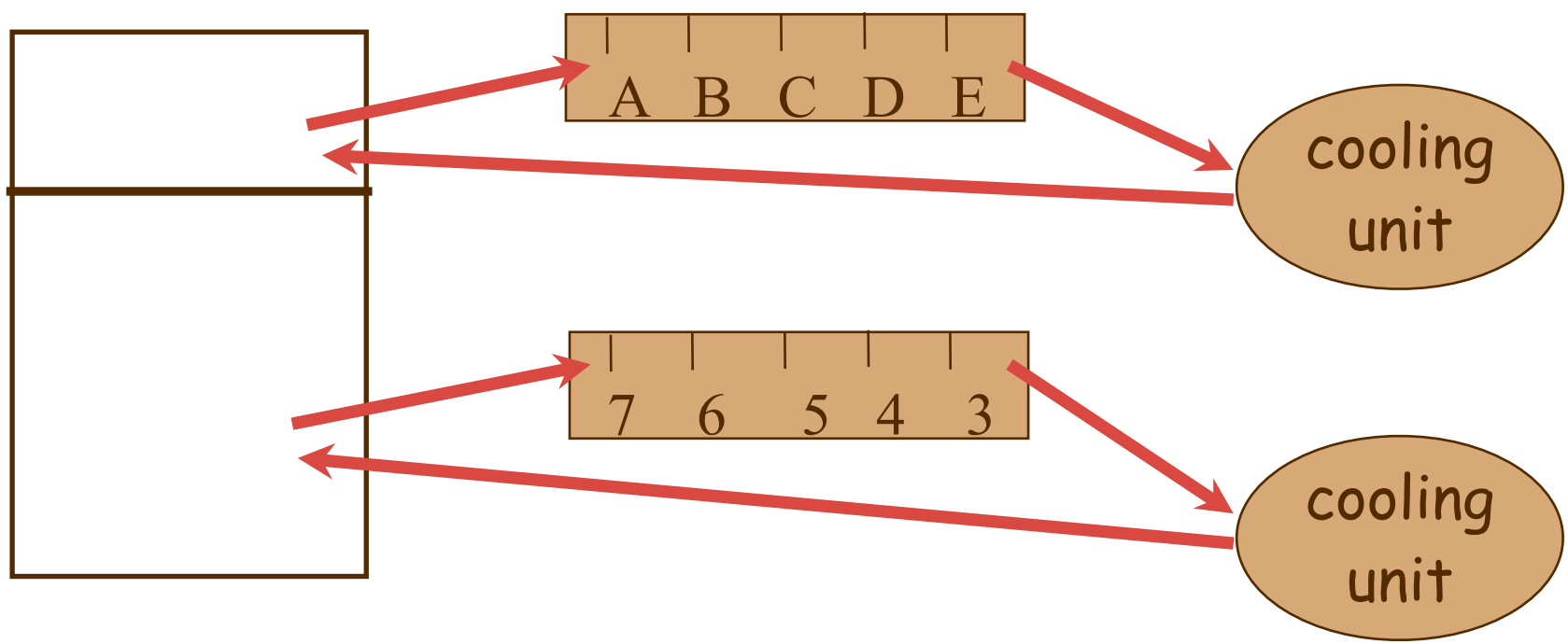
- What is your conceptual model?

- How do you think the refrigerator is controlled?

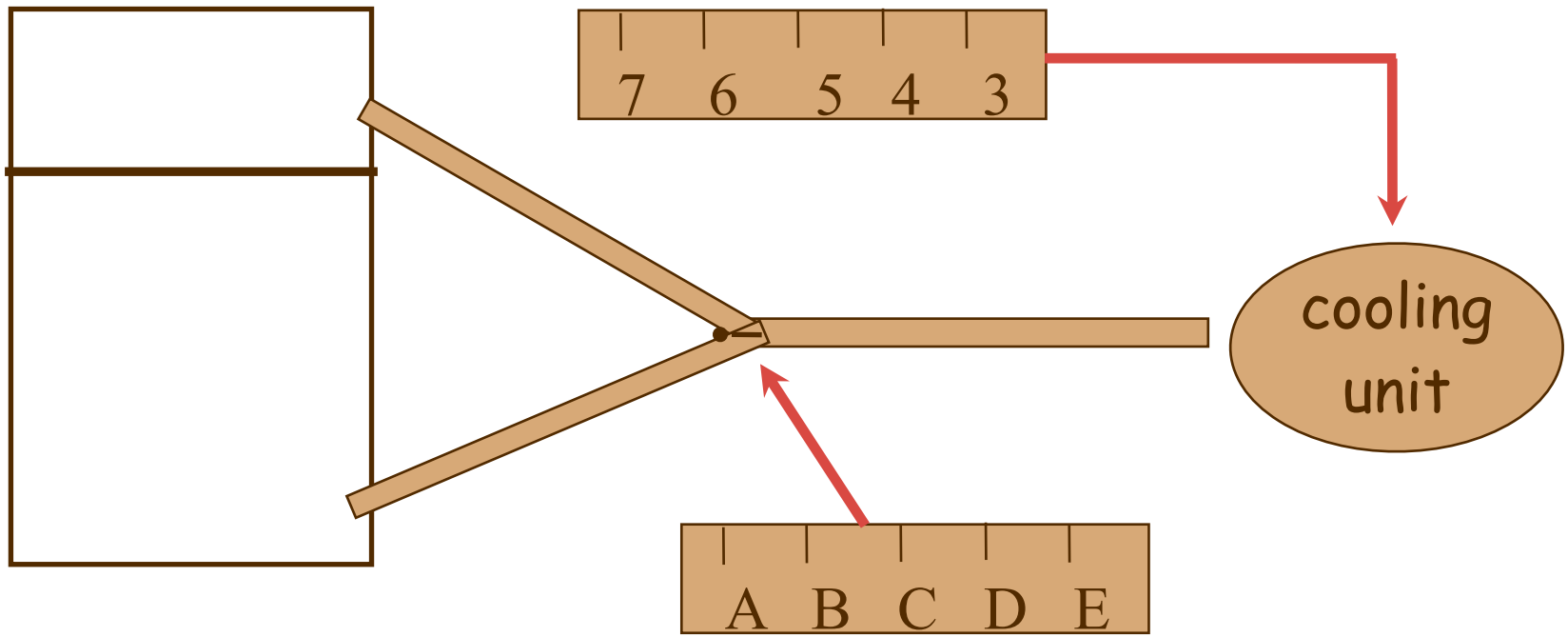


Most Likely...

- Independent Controls for Freezer and Fresh Food.

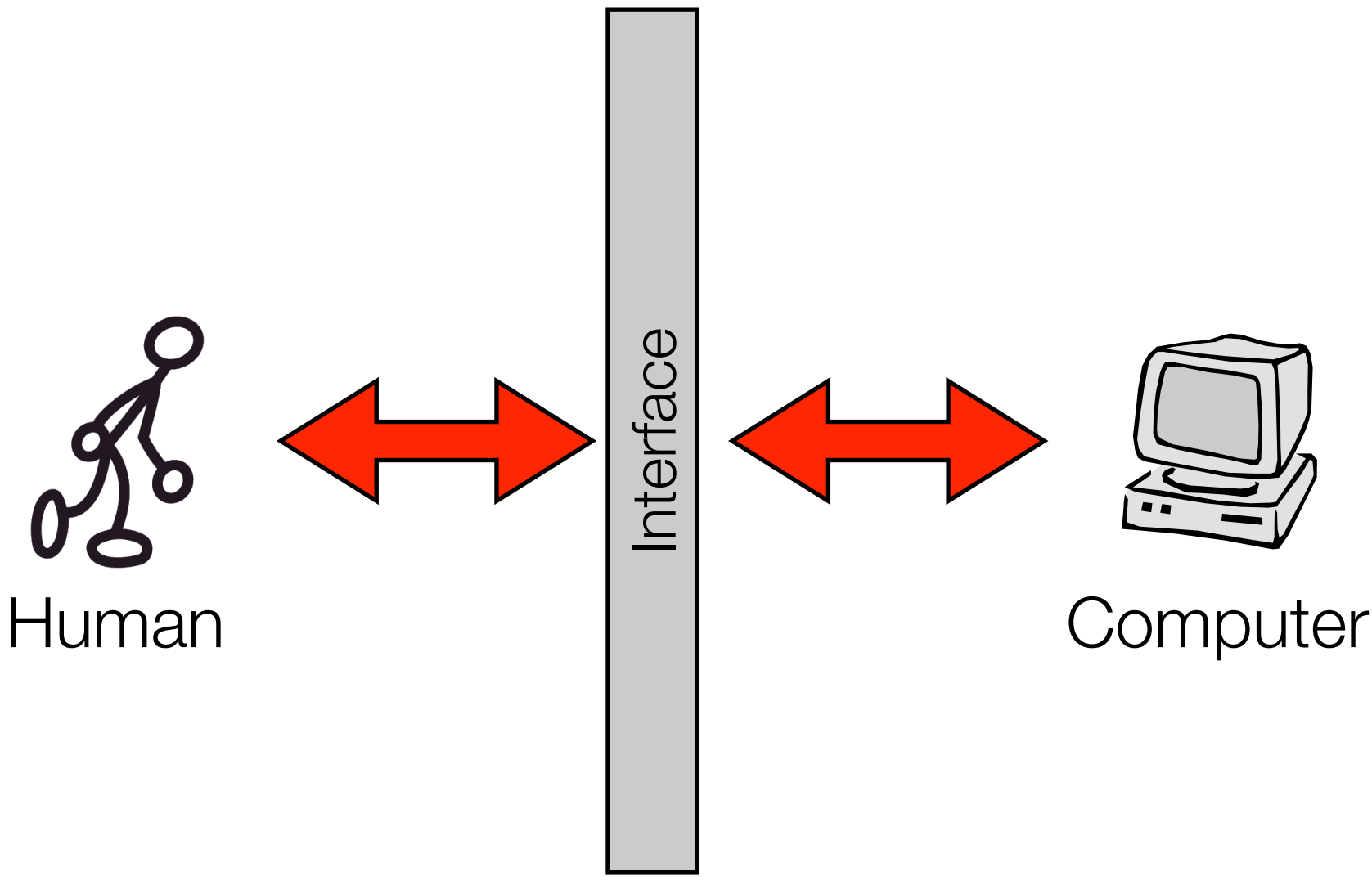


What is actually happening



- Knowing this, can you modify the controls to suggest the correct conceptual model?
  - Make controls map to user's model
  - Make controls suggest actual system.

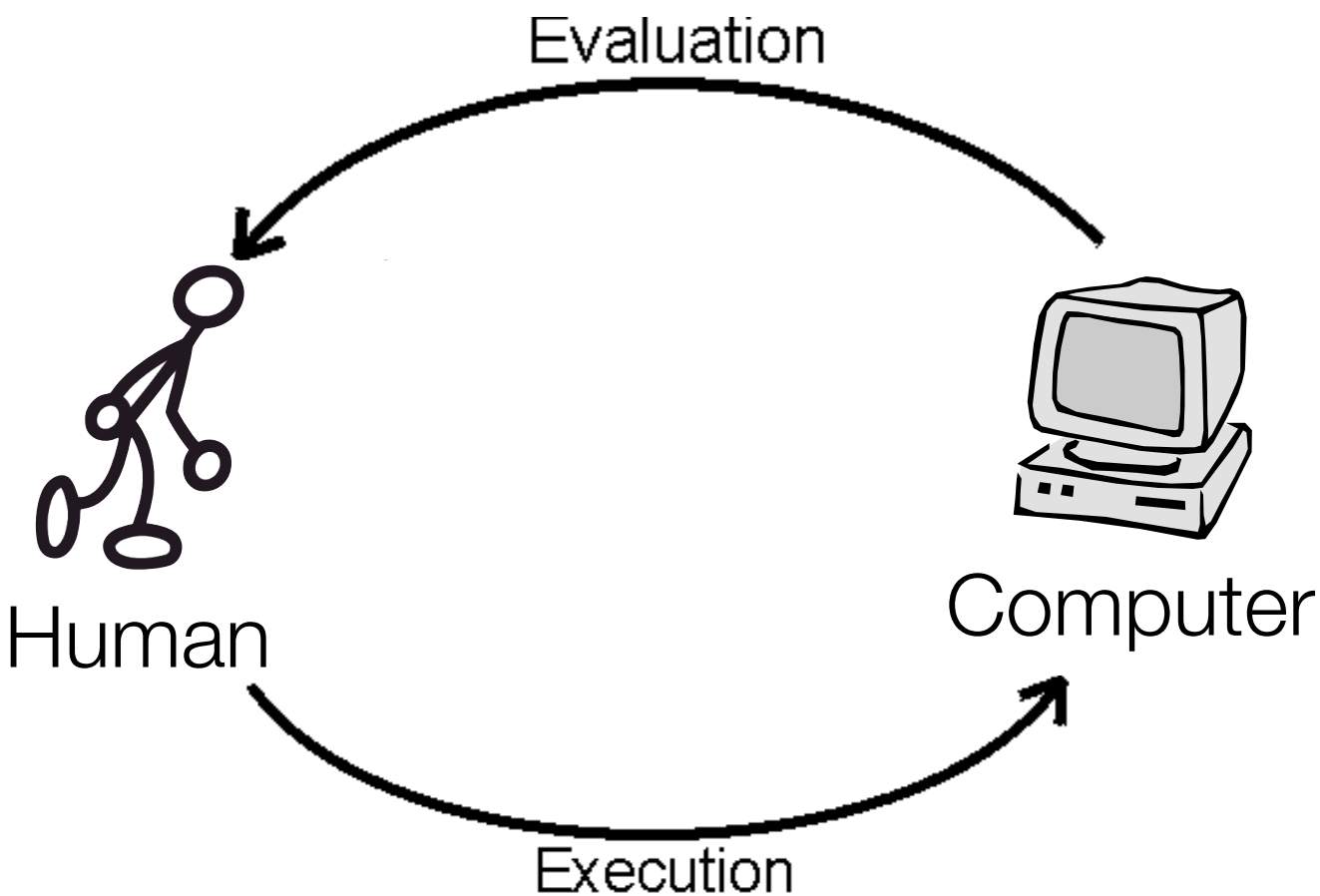
Obvious Diagram of HCI



Doesn’t really tell us how humans evaluate the computers!

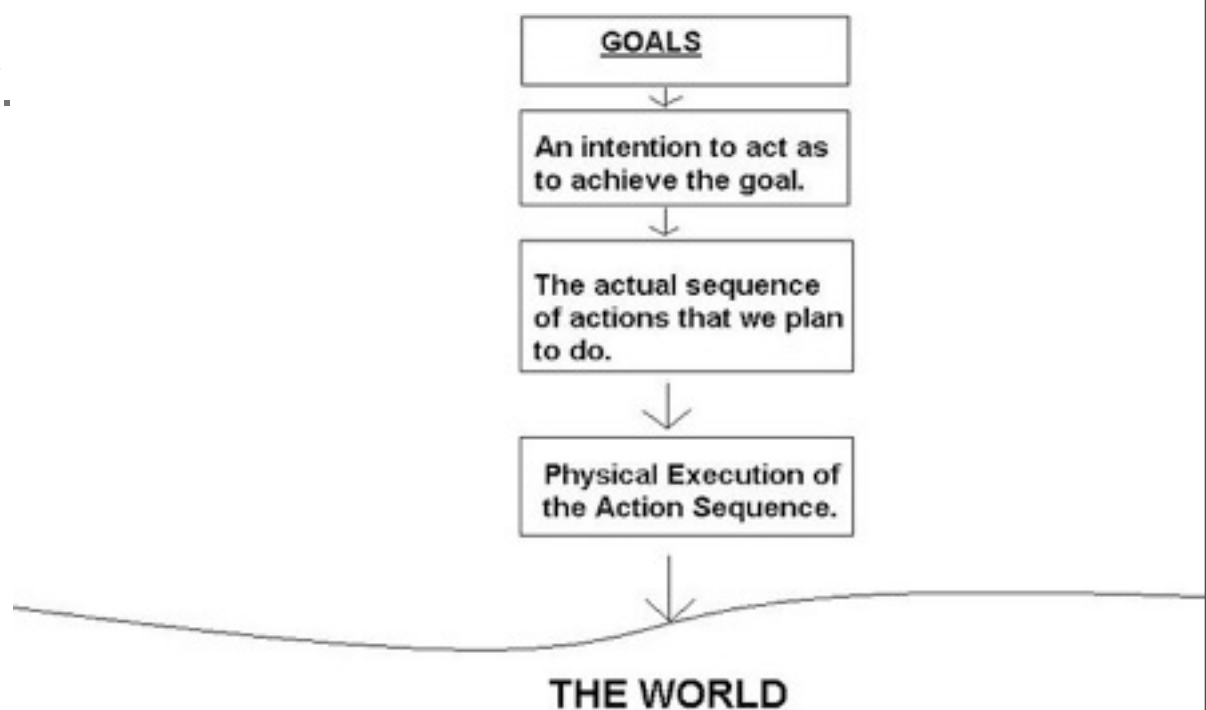
Alternate Diagram of HCI: Norman's Interaction Cycle

- Human action has two primary aspects:
  - Execution: Doing something
  - Evaluation: Comparison of what was done to what was desired/expected



Norman's Interaction Cycle: Execution Phase

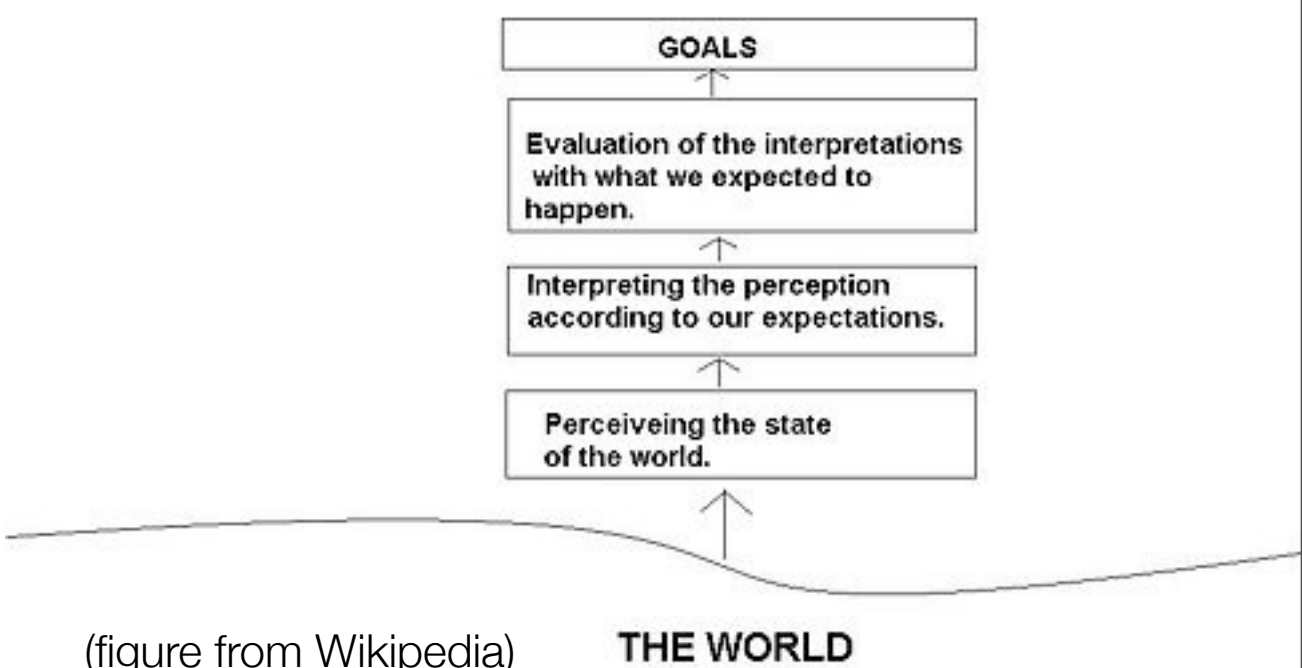
- We have a *goal* that we wish to achieve
- We translate the goal to an *intention* to do some action(s)
- The intention is translated into an *action sequence* that can be performed.
- The action sequence is *executed*.



(figure from Wikipedia)

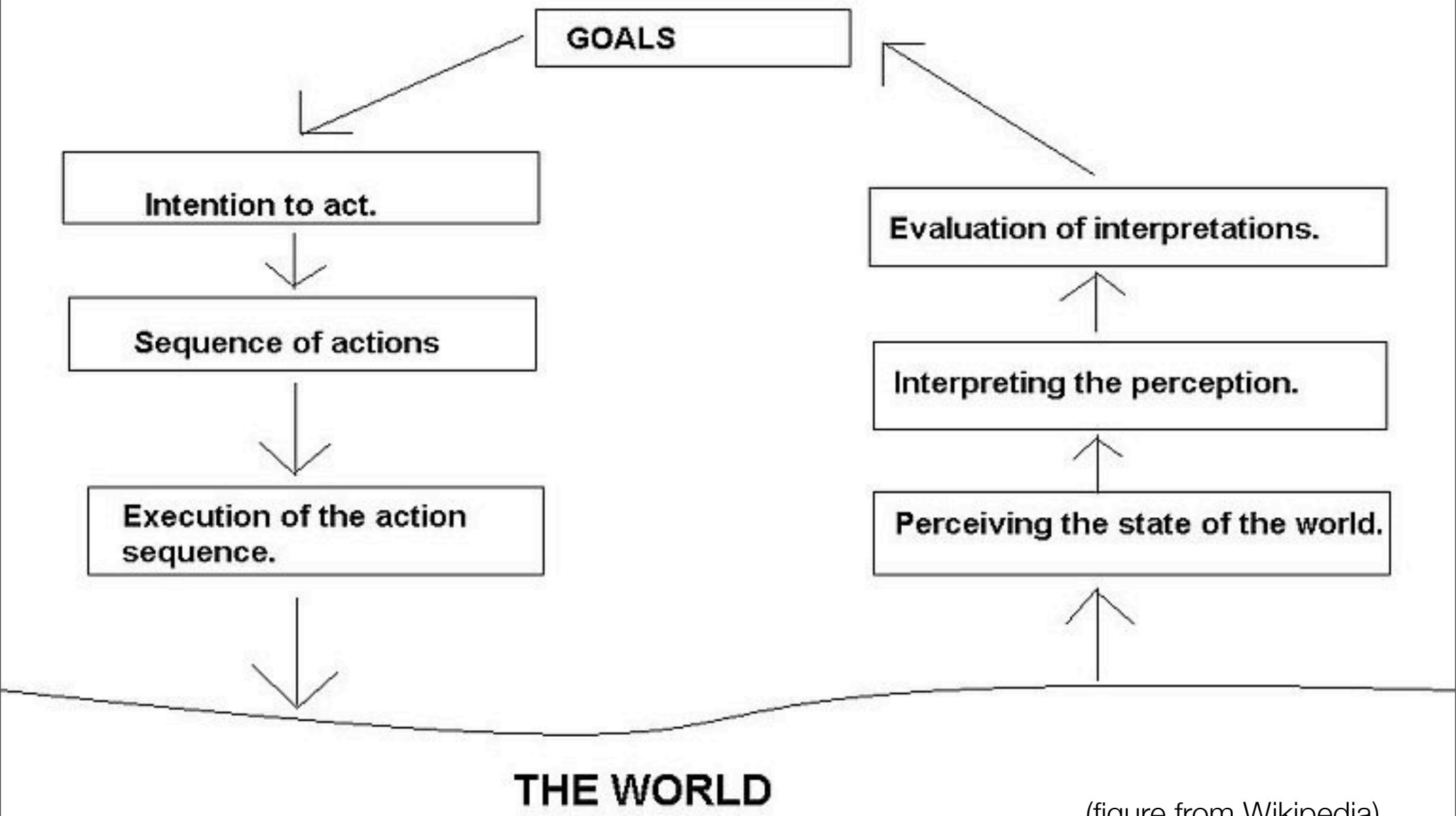
Norman's Interaction Cycle: Evaluation Phase

- We start with the *perception* of our world.
- This perception is *interpreted* according to our expectations.
- It is then *evaluated* (compared) with respect to our intentions and our goals,



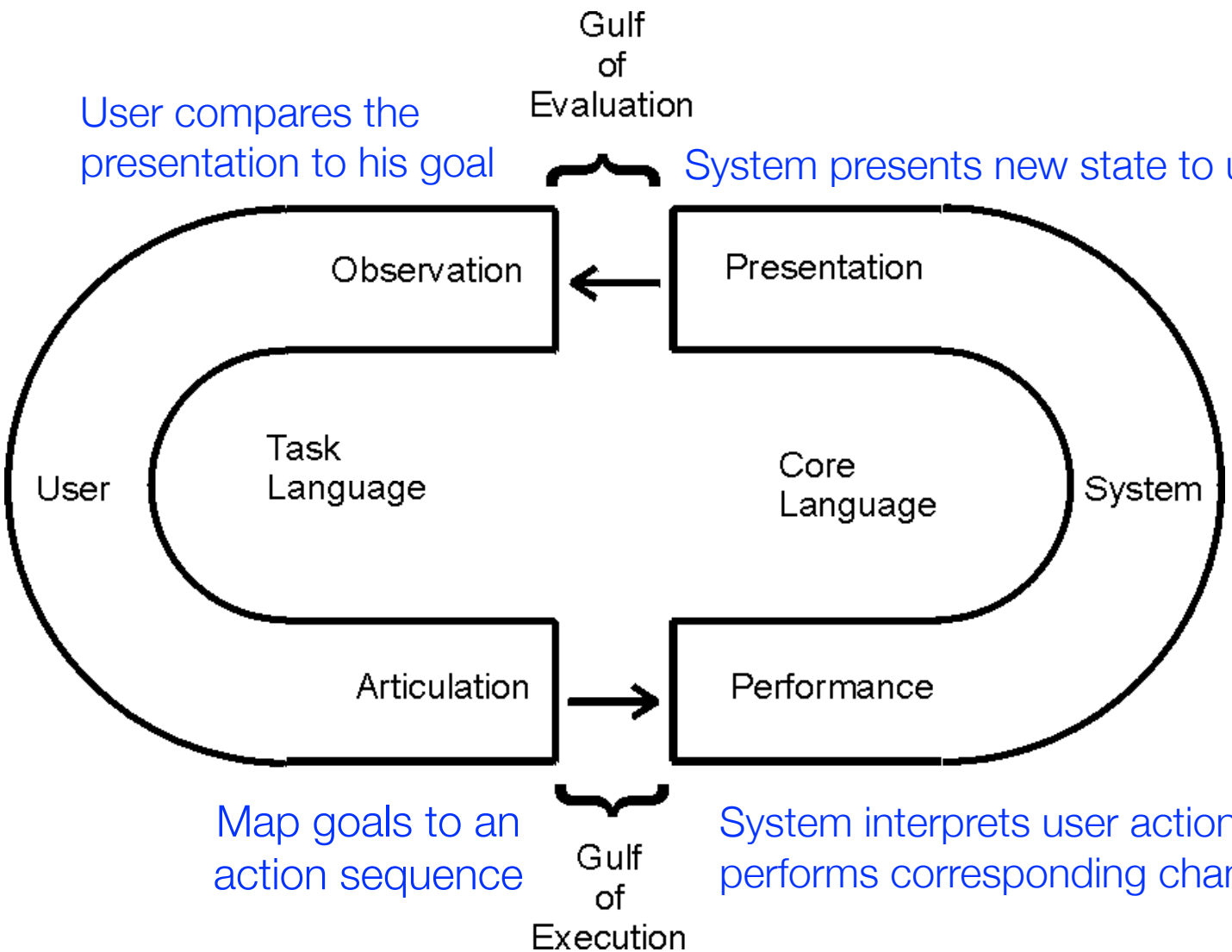
(figure from Wikipedia)

Seven Stages of Action



(figure from Wikipedia)

Put in the System: The Action Cycle



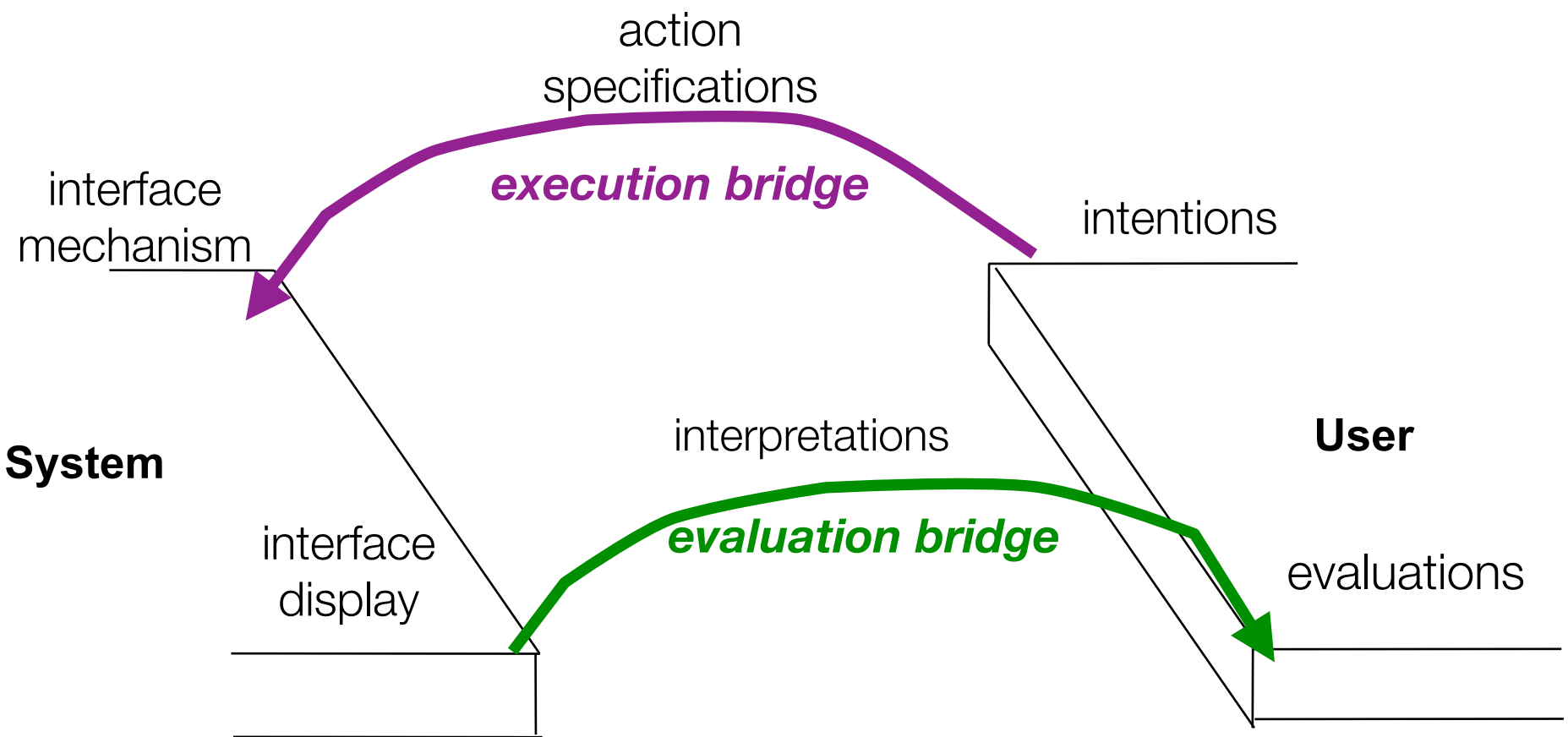
What the four stages model reveals

- The “Gulf of Execution”:
  - Amount of effort exerted to transform intentions into selected and executed actions
  - Or: Difference between intentions and allowable actions.
  - Do actions provided by system correspond to the intentions of the user?
- A good system provides direct mappings between Intention and selections
  - e.g. Printing a letter:
    - Put document on printer icon
    - vs Select print from menu
    - vs “`latex letter.tex; lpr -Palw3 latex.dvi`”
  - Drawing a line:
    - Move mouse on graphical display
    - vs “`draw (x1, y1, x2, y2)`”

What the four stages model reveals

- The “Gulf of Evaluation”:
  - Amount of effort exerted to interpret feedback
  - Can feedback be interpreted in terms of intentions and expectations?
- A good system is one where feedback is easily interpreted as task expectations.
  - e.g. graphical simulation of text page being printed
- A bad system is one where there is no feedback or gives difficult to interpret feedback.
  - e.g. Unix: “`$`”, “`bus error`”, “`command not found`”

Bridging the Gulfs of Execution and Evaluation



Good design can help bridge the gulfs...

- How easily can one:
  - Determine the function of the system?
  - Tell what actions are possible?
  - Determine mapping from intention to physical movement?
  - Perform the action?
  - Tell if system is in a desired state?
  - Determine mapping from system state presentation to interpretation?
  - Tell what state system is in?

Good design can help bridge the gulfs...

- Four main principles of good design:
- Good conceptual model
  - Consistency in presentations of operations and results
  - Coherent, consistent system image
- Good mappings: Clearly seen relations between
  - Actions and results
  - Controls and their effects
  - System state and what is visible
- Feedback
  - Full and continuous feedback about results of actions
- Visibility
  - User can easily tell state of the device and alternatives for action
  - Principle of Transparency: “the user is able to apply intellect directly to the task; the tool itself seems to disappear”

Good design can help bridge the gulfs...

- A good design should give a good idea of how each object works and how to control it.
- In other words, the interface itself discloses how it is used.
- The art in design is to translate user's cognitive capabilities and existing mental models into interfaces that work!
- So — how do users perceive how a design works??

Conceptual Models

- People have “Mental Models” of how things work, built from the following:
- Visual Cues:
  - Affordances
  - Constraints
- Logical cues:
  - Causality
  - Mapping
  - Transfer
- Population Stereotypes

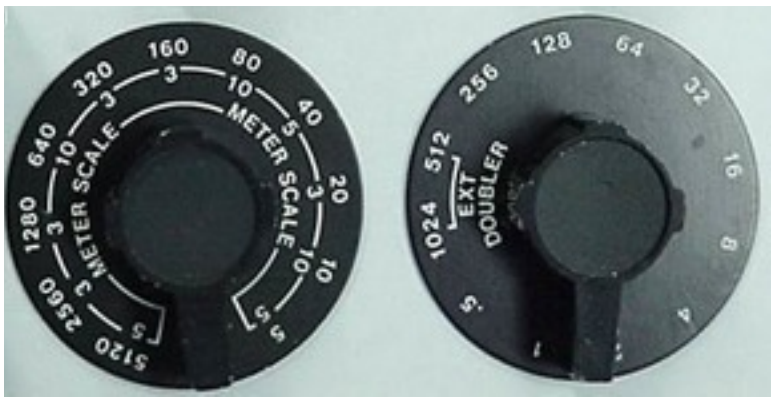
Visual Cues: Affordances

- The psychological idea:
  - “Things may suggest by their shape and other attributes what you can do to them.”
- Affordances:
  - The fundamental properties of the object that determine how it could possibly be used.
  - What it could be used for (physical)
  - What we think to use it for (cultural)

Perceived Affordance

The perceived properties of the object that suggest how one could use it

chairs are for sitting
table for placing things on



knobs are for turning

slots are for inserting
handles are for turning

buttons are for
pressing

switch for toggling

computer for…

Many concepts in this section are adapted from Don Norman's book: The Design of Everyday Things

When Affordances cause Trouble...

- Perceived Affordances: design invites people to take certain actions.
- Real Affordances: the actual actionable properties of the device/system.
- Problems occur when:
  - These are not the same.
  - People's perceptions are not what the designer expects.

Perceived Affordances

Perceived Affordances Cause Problems

Mirrors for not touching

people don't reposition image

Knobs for turning

focus or image position?

Handles for lifting

bends frame, focus distorted

Surface for placing transparencies

which way is up?

what about this?

Affordances in GUI Design

- With GUI design, perception is only through visuals
  - i.e. There aren't real knobs, buttons, etc.
  - However, the designer can create appropriate visual affordances via
    - Familiar idioms
    - Metaphors

Perceived Affordances

dials for turning

sliders for sliding

music console for controlling music

Perceived Affordances: Problems

is this equalizer control a toggle or button?

button for pressing, but action unknown

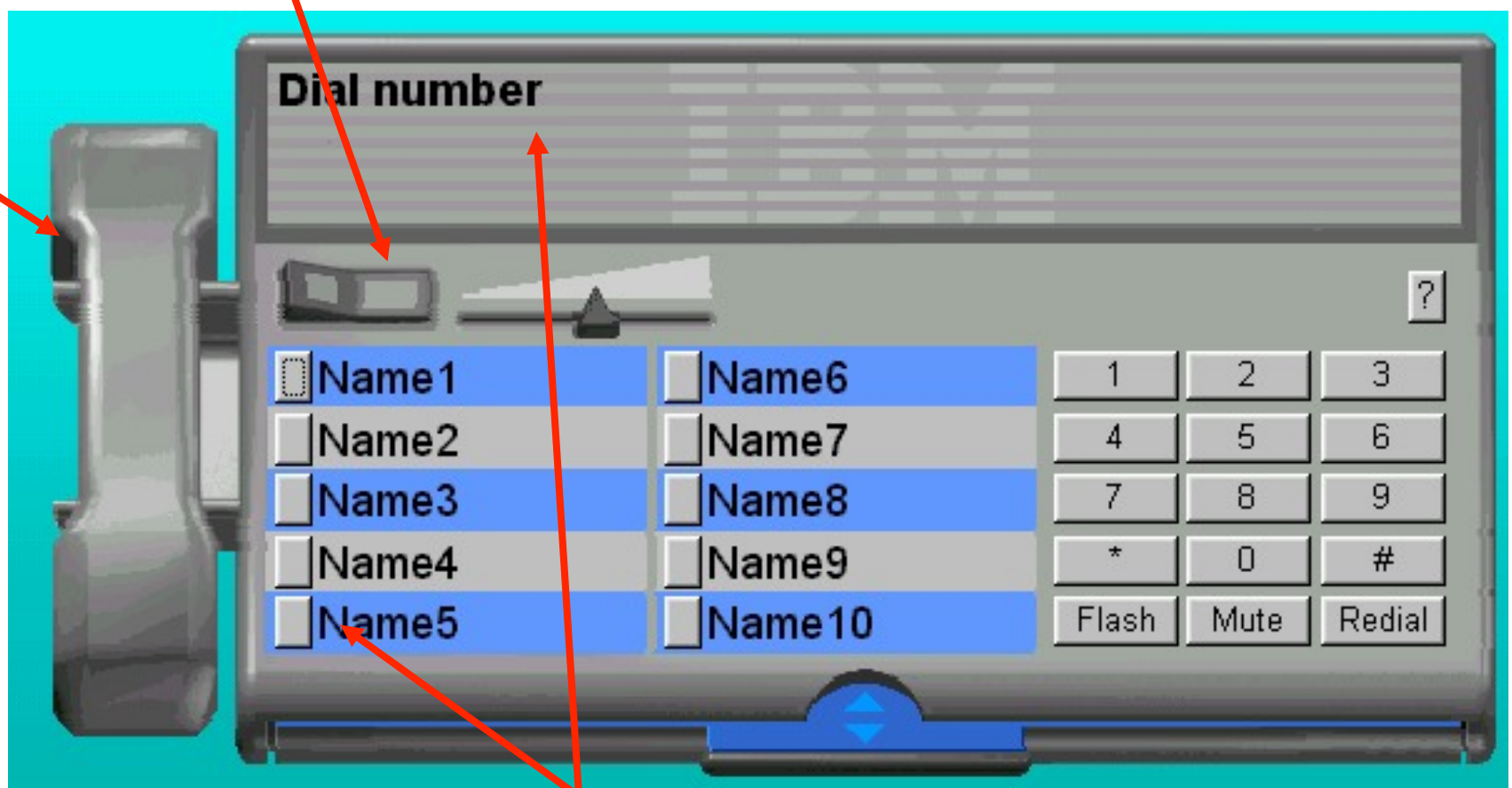
are these buttons?

Perceived Affordances: Problems

Is this a graphic or a control?

A button is for pressing, but what does it do?

Visual affordances for window controls are missing!



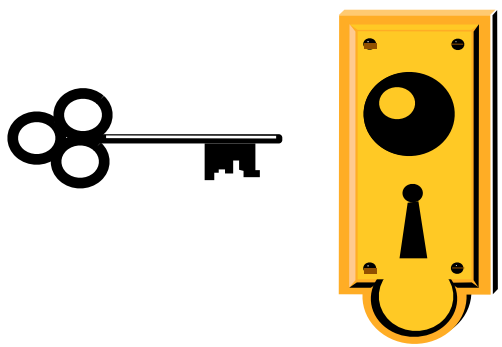
text is for editing, but you can't edit directly onto it.

IBM Real Phone

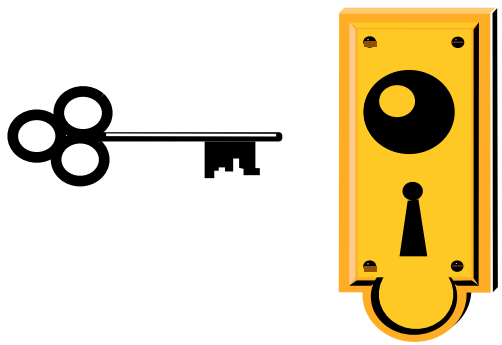
Visual Cues: Constraints

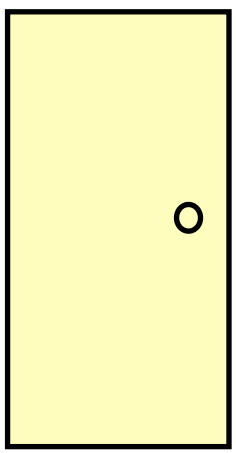
- Constraints are restrictions on affordances
    - Size, proximity, type, number
    - E.g. Scissors fit fingers but not arms
- Provides people with a wide range of possibilities
- Designed constraints can prevent incorrect use
    - But like affordances, they must be perceivable to the user!

Constraints



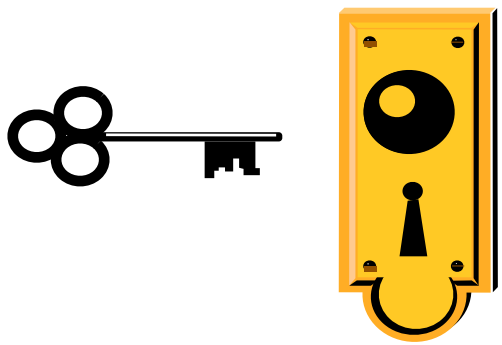
Constraints

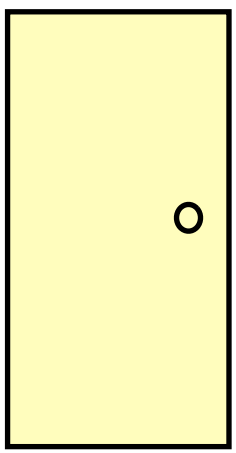




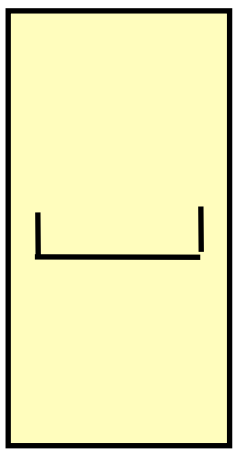
Push or pull?

Constraints



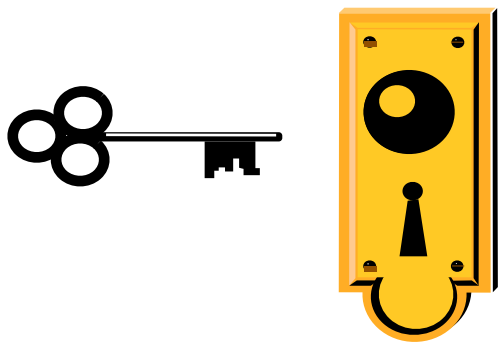


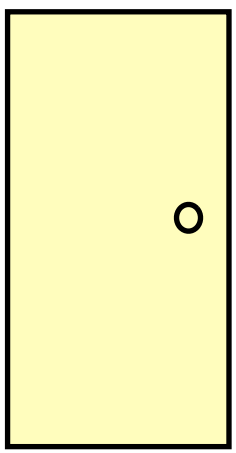
Push or pull?



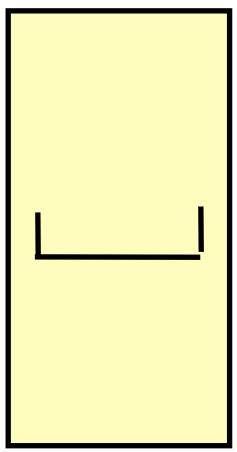
Which side?

Constraints

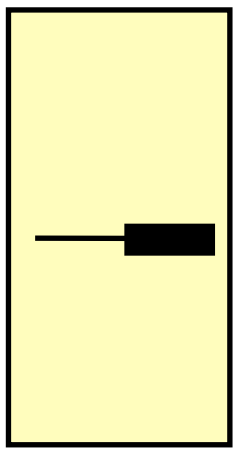




Push or pull?



Which side?



Can only push,
side to push clearly visible

Logical Structure: Mapping

- Mapping is the set of possible relationships between objects.
- Used to indicate control-display compatibility:
    - The relationship between controls and displays.

Wednesday, 28 September, 11

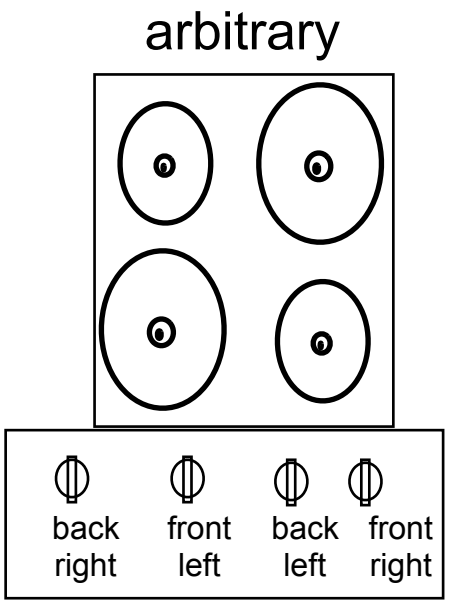

OFF
OFF
OFF
OFF
OFF
OFF
HI
LO
HI
LO
HI
LO
HI
LO
HI
LO
250
200
150
100
ON
200 230 260
400 450 500
OVEN
HOTPLATES
GRILL
Centenary

Mapping

Which control element corresponds to which heating element?

Mapping

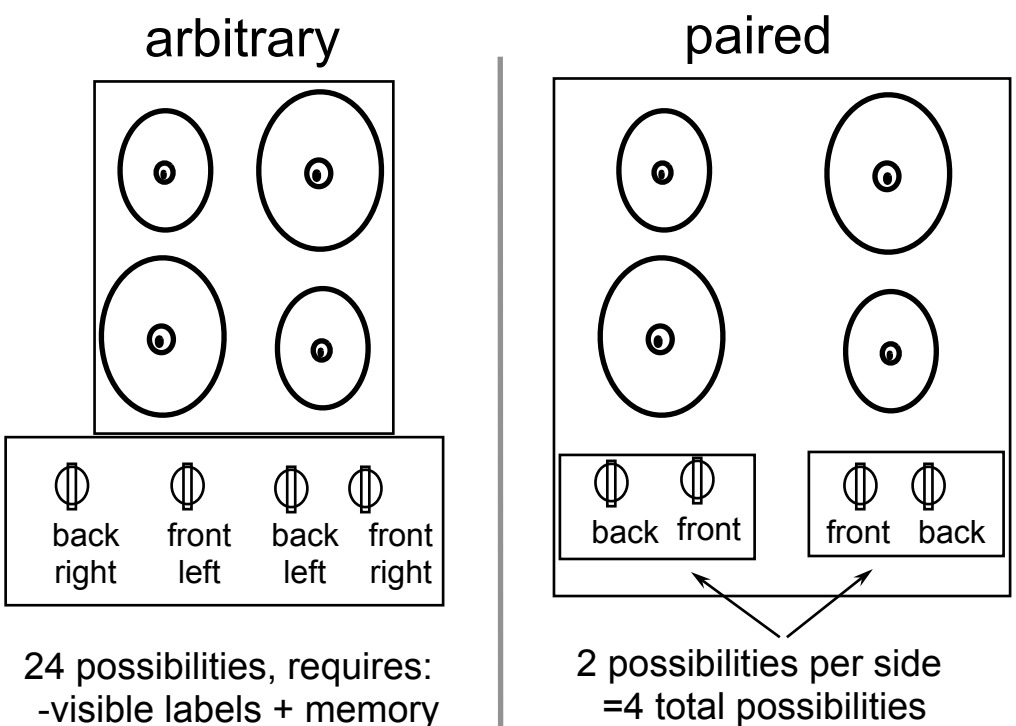
Which control element corresponds to which heating element?



24 possibilities, requires:
-visible labels + memory

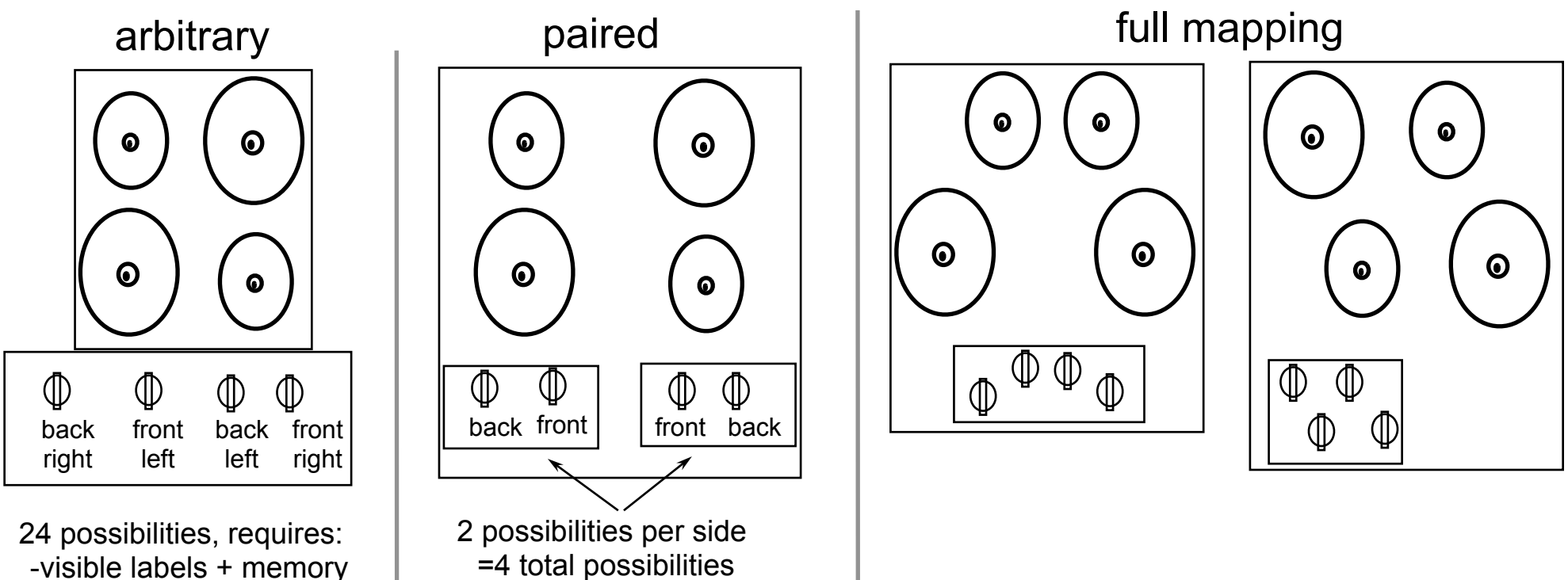
Mapping

Which control element corresponds to which heating element?

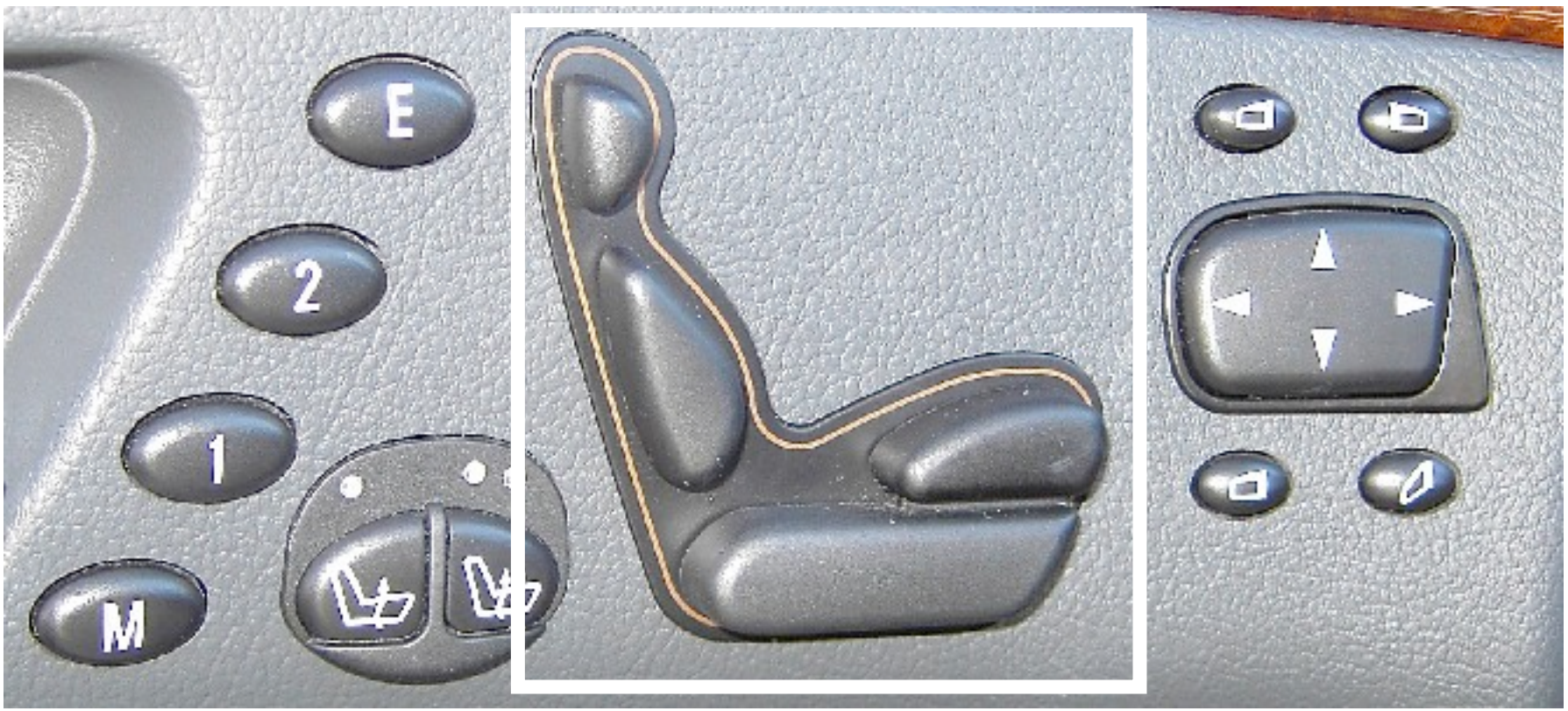


Mapping

Which control element corresponds to which heating element?



Excellent Mapping Example



Mercedes Benz s500 Car Seat Controller

http://www.lilviv.com/motoring/cars/s500/seatcont.jpg

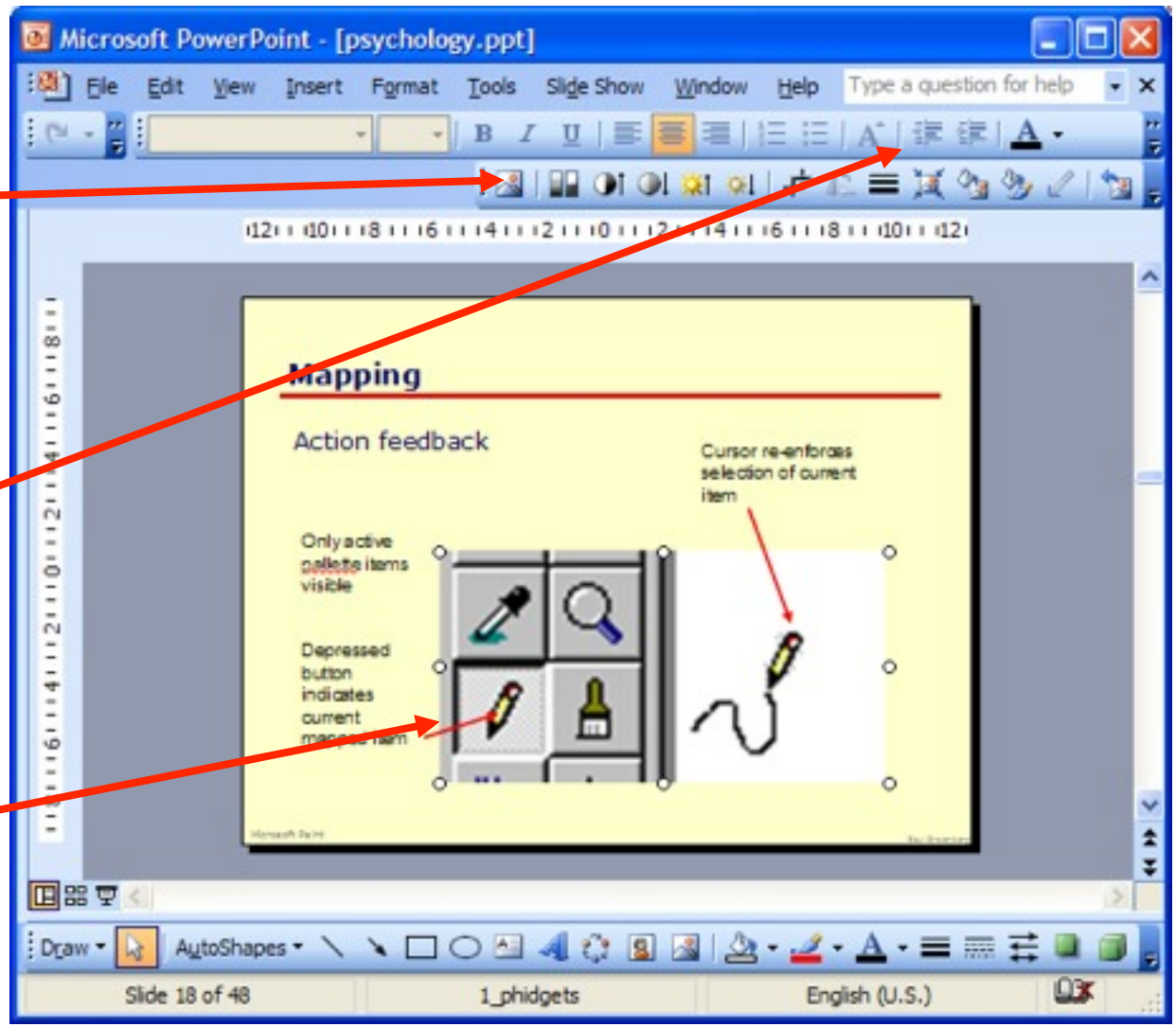
Mapping on GUIs

- Palette controls and active objects

Only controls that can operate on a picture are fully visible

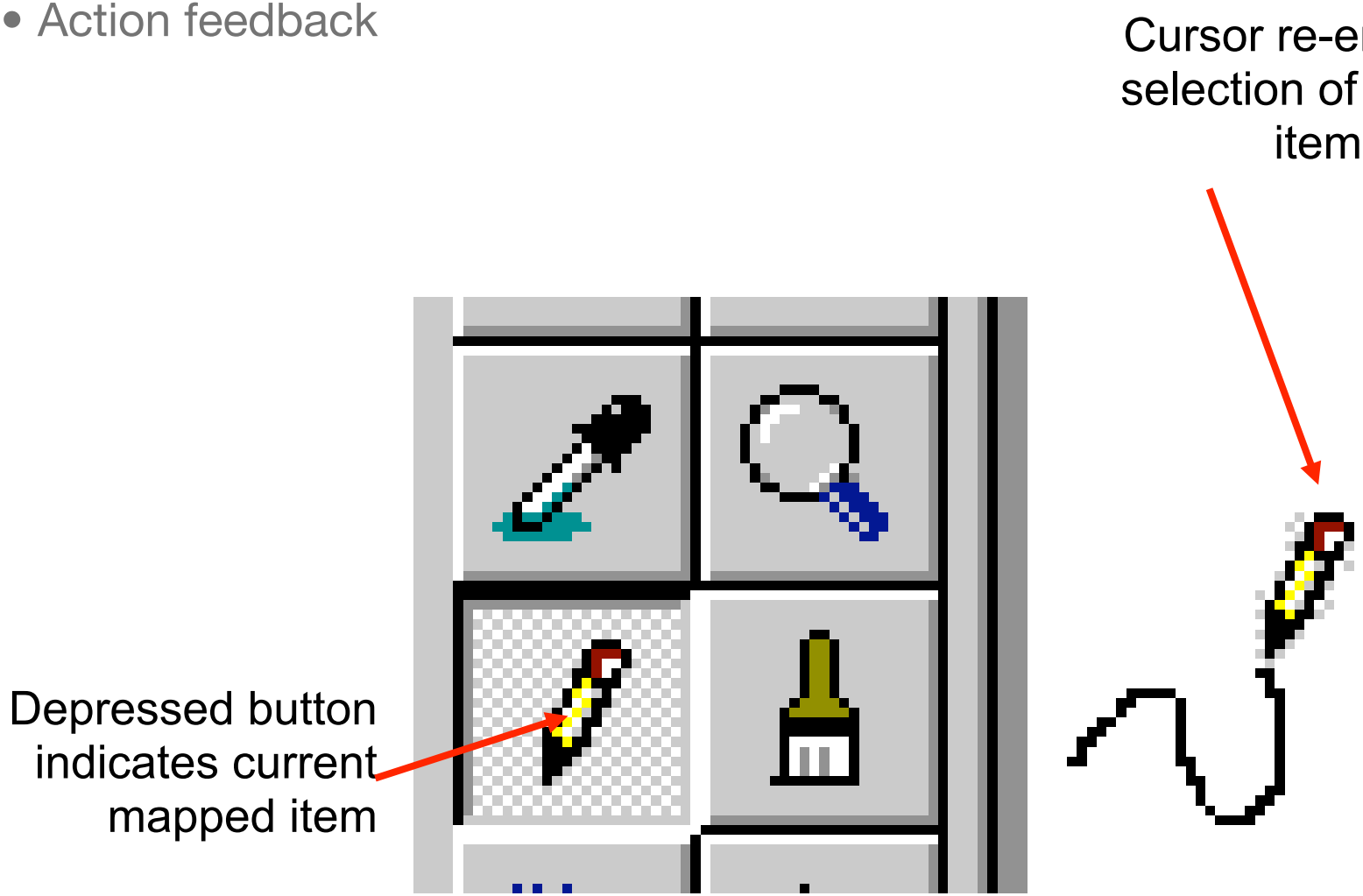
Others are grayed out

Selected picture



Mapping on GUIs

- Action feedback



Mapping Problems

Quick, open the top drawer

Photograph courtesy of www.baddesigns.com

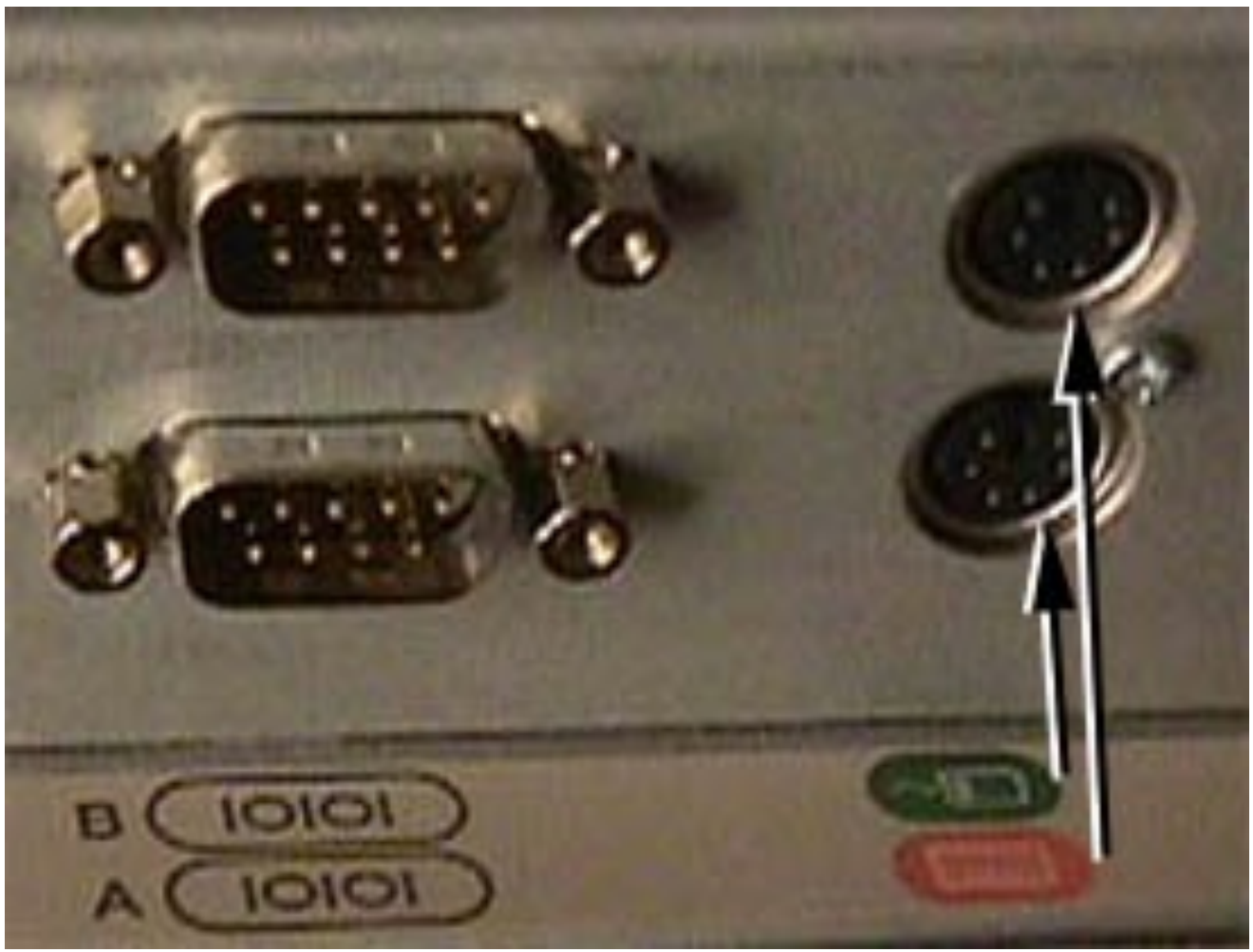
Mapping Problems

Quick, open the top drawer

Photograph courtesy of www.baddesigns.com

Mapping Problems

Where do you plug in the mouse?



Mapping ambiguous

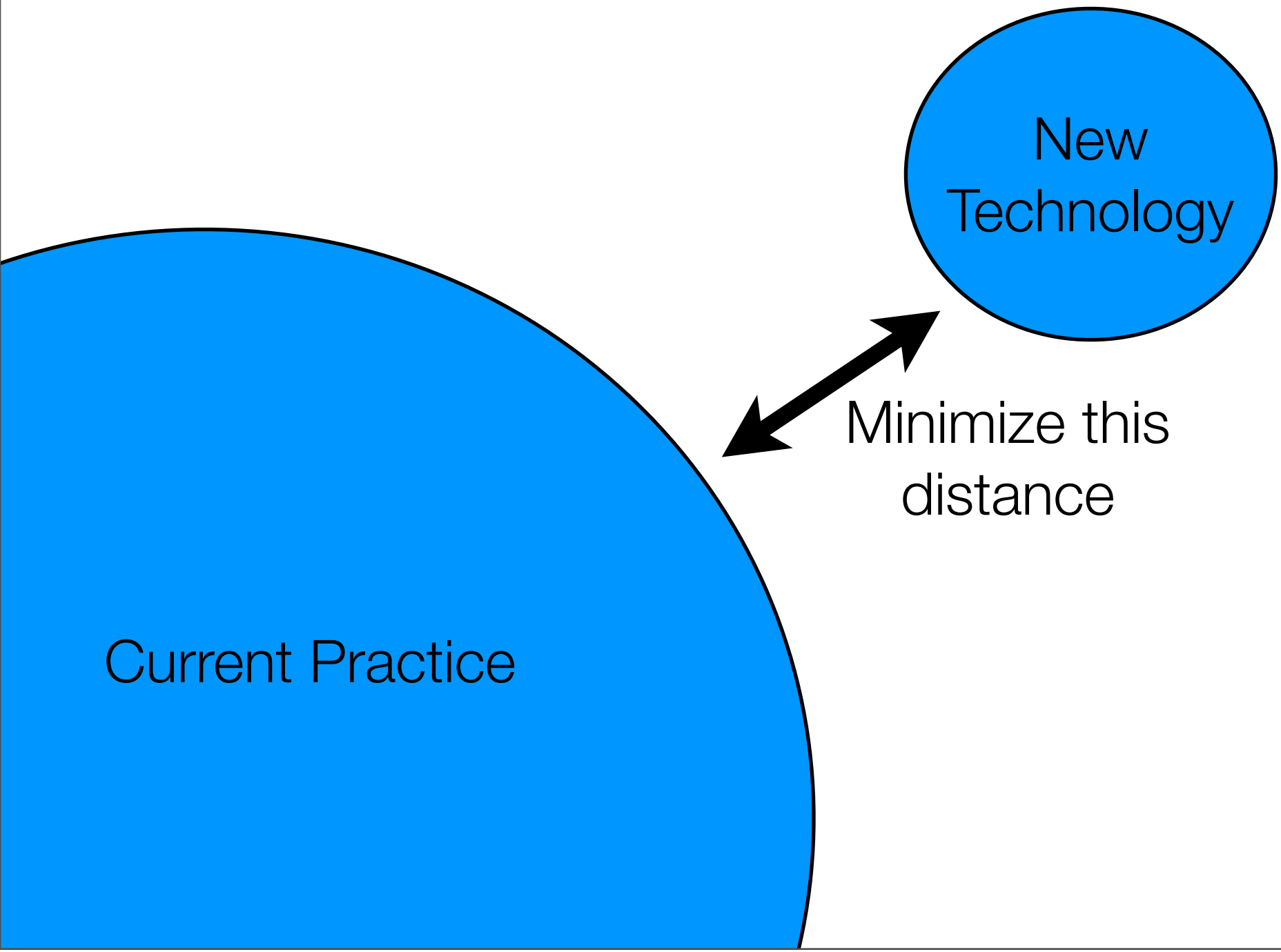
Photograph courtesy of www.baddesigns.com

Logical Structure: Causality

- The thing that happens right after an action is assumed by people to be caused by that action.
- An interpretation of “feedback”
- Potential problems:
  - Incorrect effect/assumption
    - Inserting a DVD and then the computer crashes.
    - Causes “superstitious” behaviors.
  - Invisible effects
    - Re-entering a command that produces no apparent effect.
    - E.g. Clicking a button multiple times on an unresponsive system.

Logical Structure: Transfer Effects

- When dealing with a new object, people borrow from prior experience with similar objects.
  - Positive transfer: prior knowledge works correctly in the new object.
  - Negative transfer: prior knowledge conflicts with the new object's requirements.

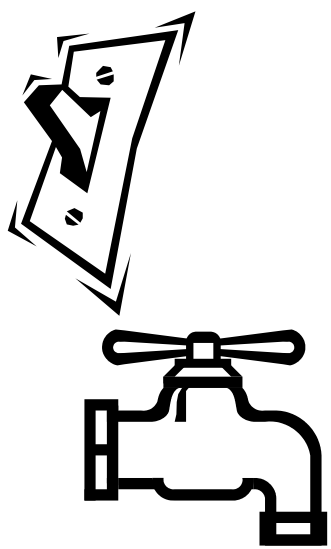

New Technology
Minimize this distance
Current Practice

Idioms and Population Stereotypes

- Interface Idioms:
  - “Standard” interface features we learned, use and remember.
- Idioms may define arbitrary behaviors
  - Red means danger
  - Green means safe

Idioms and Population Stereotypes

- Population Stereotypes: Idioms vary from culture to culture.
  - Light Switches
    - America: down is off
    - Britain: down is on
  - Taps
    - America: anti-clockwise on
    - Britain: anti-clockwise off



Idioms and Population Stereotypes

- Ignoring/changing idioms?
  - Home handyman
    - Light switches installed upside-down
  - Calculators vs. Phone number pads
- Difficulty of changing stereotypes
  - Qwerty keyboard: designed to prevent keys jamming
  - Dvorak keyboard: Proved to be faster.

Metaphors

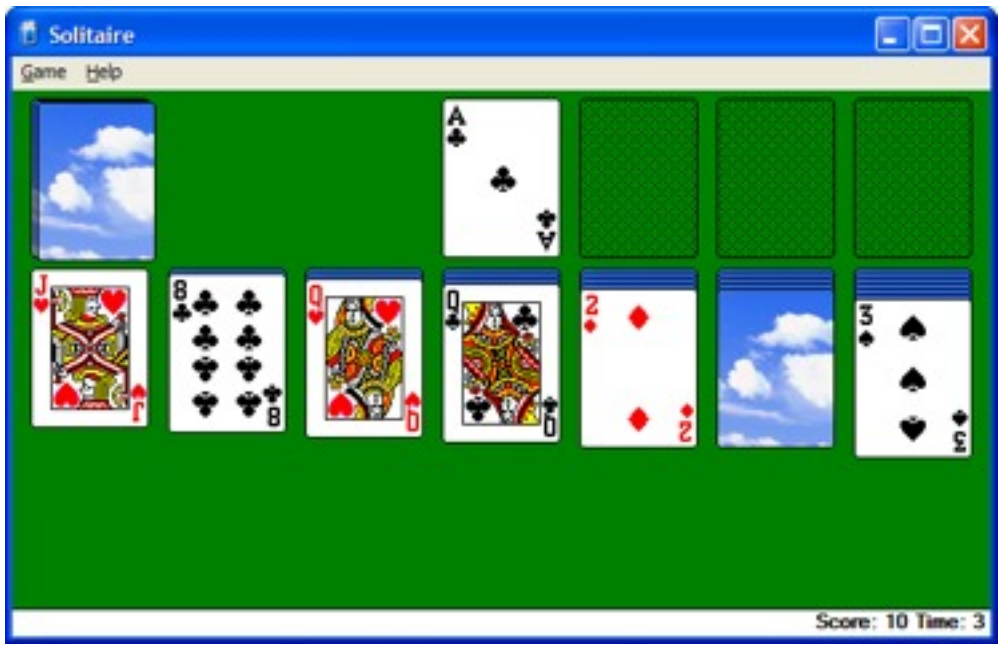
- “The transference of the relation between one set of objects to another set for the purpose of brief explanation”
- Represents a system object as if it were another type of object
    - E.g. Disc / network file structure represented as file folders
- Leverages our knowledge of familiar, concrete objects to understand abstract computer and task concepts
- Probably the most well-known:
    - Trashcan, Desktop on Windows
- We can use metaphors to highlight certain features & suppress others

Examples of Metaphors

Pervade excellent interfaces

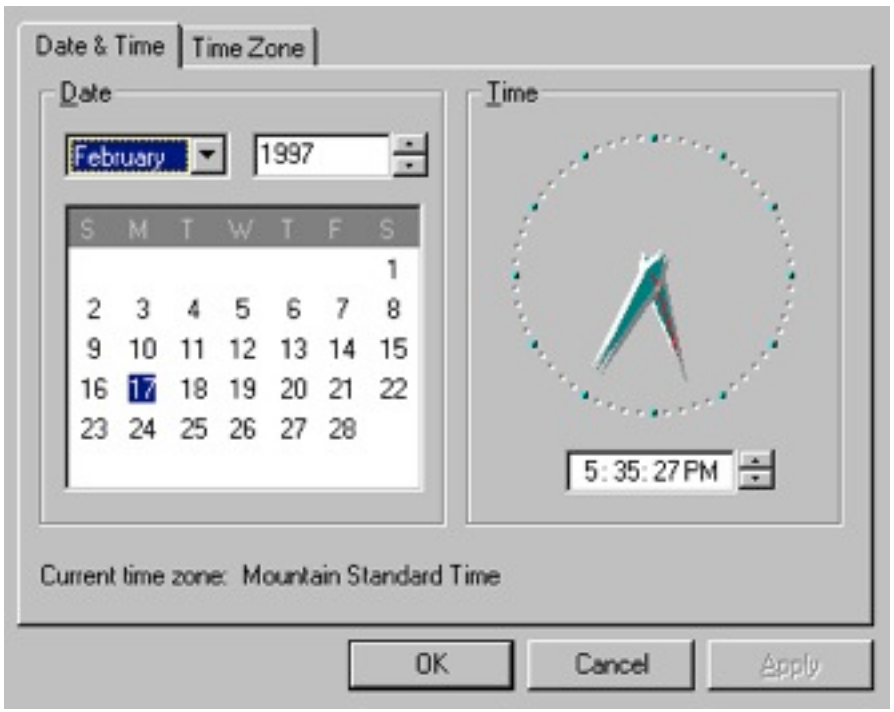
	A	B	C	D
1	Market value	Land	Improvement	Total assess
2	140.0	65,850.	73,120.	138,970.
3	147.0	77,780.	72,070.	149,850.
4	151.0	74,850.	88,740.	163,590.
5	152.0	80,110.	99,410.	179,520.
6	155.0	79,050.	109,130.	188,180.
7	170.0	94,750.	50,960.	145,710.
8	172.0	82,150.	106,250.	188,400.
9	178.0	78,560.	132,660.	211,220.
10	180.0	92,840.	105,670.	198,510.
11	180.0	80,090.	103,130.	183,220.
12	182.0	76,650.	115,210.	191,860.
13	185.0	75,590.	152,710.	228,300.
14	185.0	85,870.	105,330.	191,200.
15	185.0	80,060.	113,600.	193,660.
16	193.4	80,140.	131,340.	211,480.
17	194.5	73,400.	176,210.	249,610.
18	197.0	84,960.	129,800.	214,760.
19	203.0	91,600.	119,170.	210,770.
20	205.0	79,460.	137,250.	216,710.
21	213.0	87,060.	124,350.	211,410.
22	221.0	97,330.	167,500.	264,830.
23	225.0	87,160.	157,290.	244,450.
24	245.0	79,520.	144,840.	224,360.
25	248.0	89,470.	183,500.	272,970.
26	278.0	82,150.	168,720.	250,870.
27	302.5	118,500.	109,800.	228,300.
28	308.0	83,100.	141,730.	224,830.

spreadsheet (actuary sheet)



games (literal world)

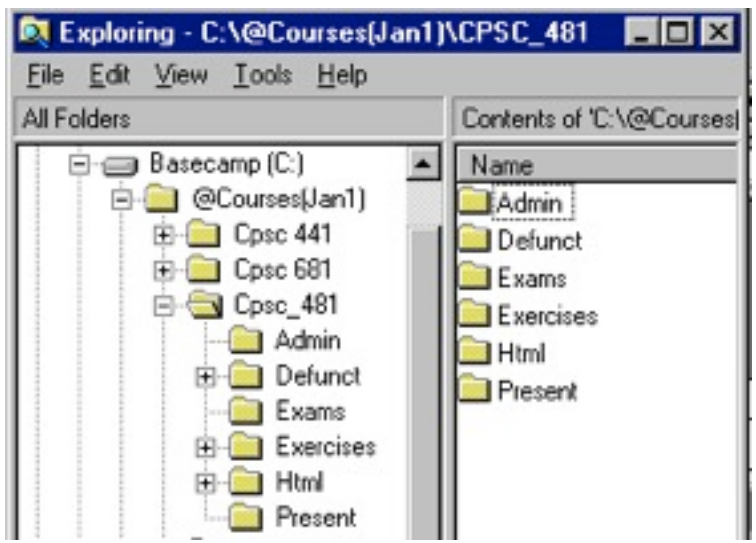
Familiar Objects



Control Panels with familiar controls

Name:_____
Address:_____
City:_____
Province:_____
Postal Code:_____

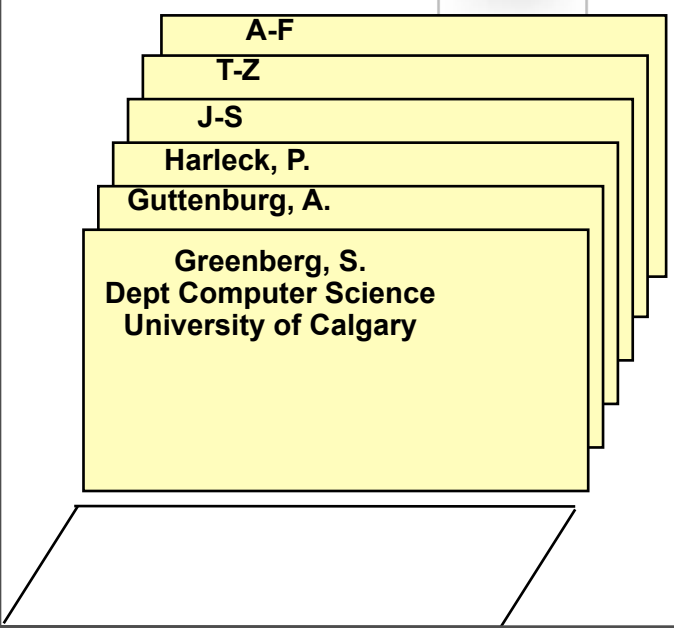
Forms

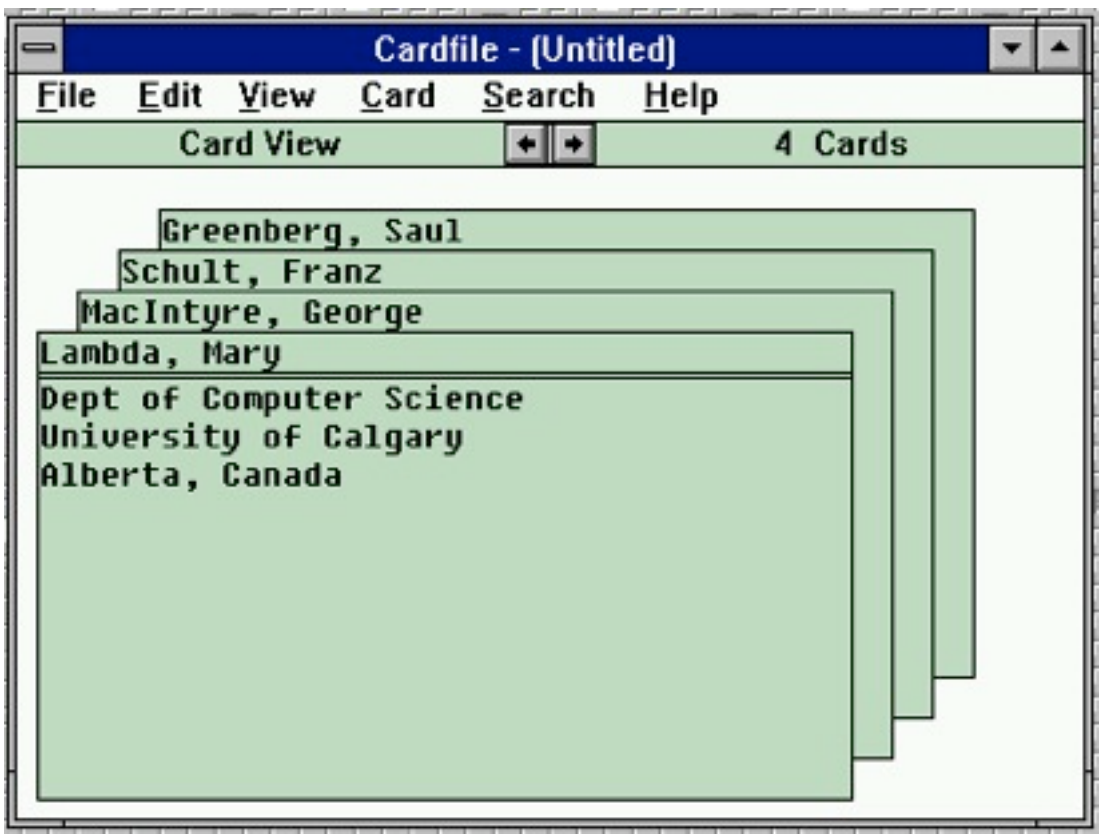


Hierarchical Folders

Contact lists

Rolodex

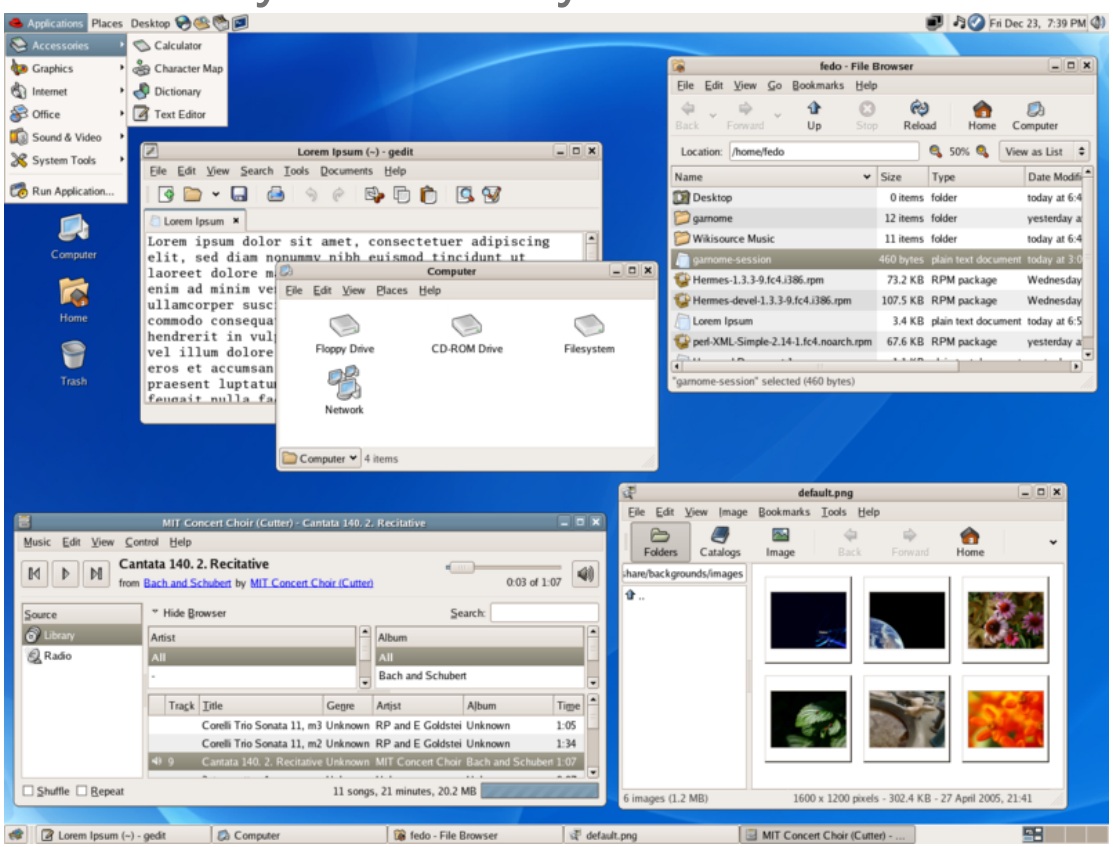




Desktop Metaphor

- Leverages existing knowledge about files, folders and trash.

- Users very quickly got used to the idea of icons representing files — to the point where they would say “this *is* my file”.



Map Metaphor

Metaphors in Interfaces

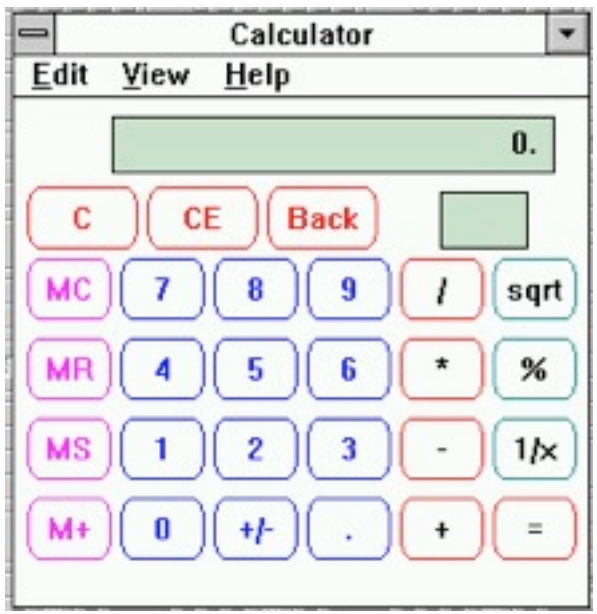
- Things to watch for
  - Use metaphors that matches user's conceptual task
    - Desktop metaphor for office workers
    - Paintbrush metaphor for artists...
  - Given a choice, choose the metaphor close to the way the system works
  - Ensure emotional tone is appropriate to users
    - eg Which is the best file deletion metaphor?
      - Trashcan/black hole/paper shredder/pit bull terrier/nuclear disposal unit...

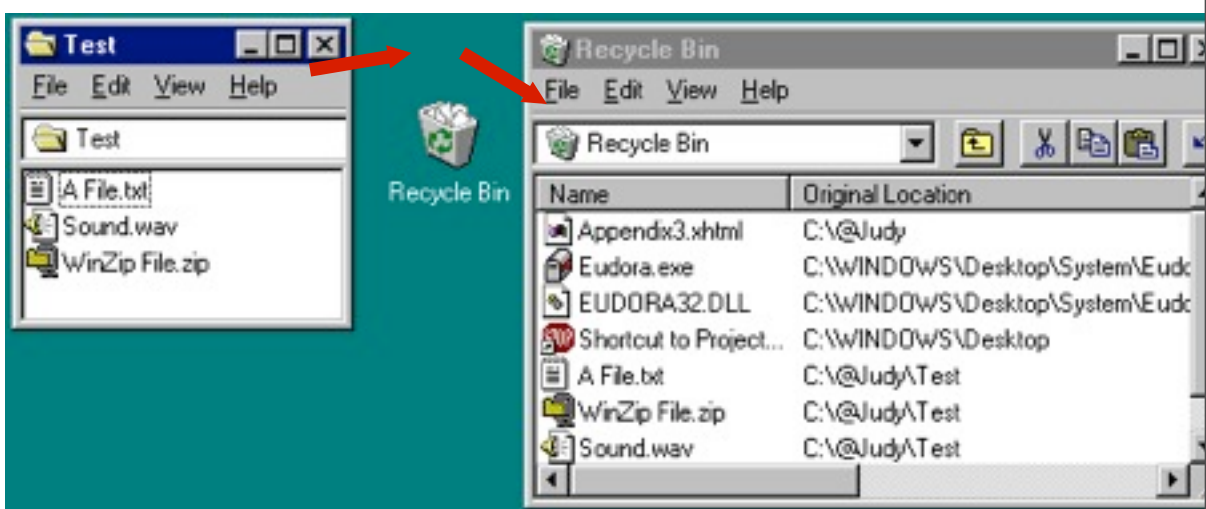
Metaphors in Interfaces

- Things to watch for
  - Will it restrict what people could actually do?
    - E.g. Strict file/folder hierarchy uses metaphors of hanging files
    - But what if the system allows links between directories?
  - Will it set unrealistic expectations?
    - Clipit

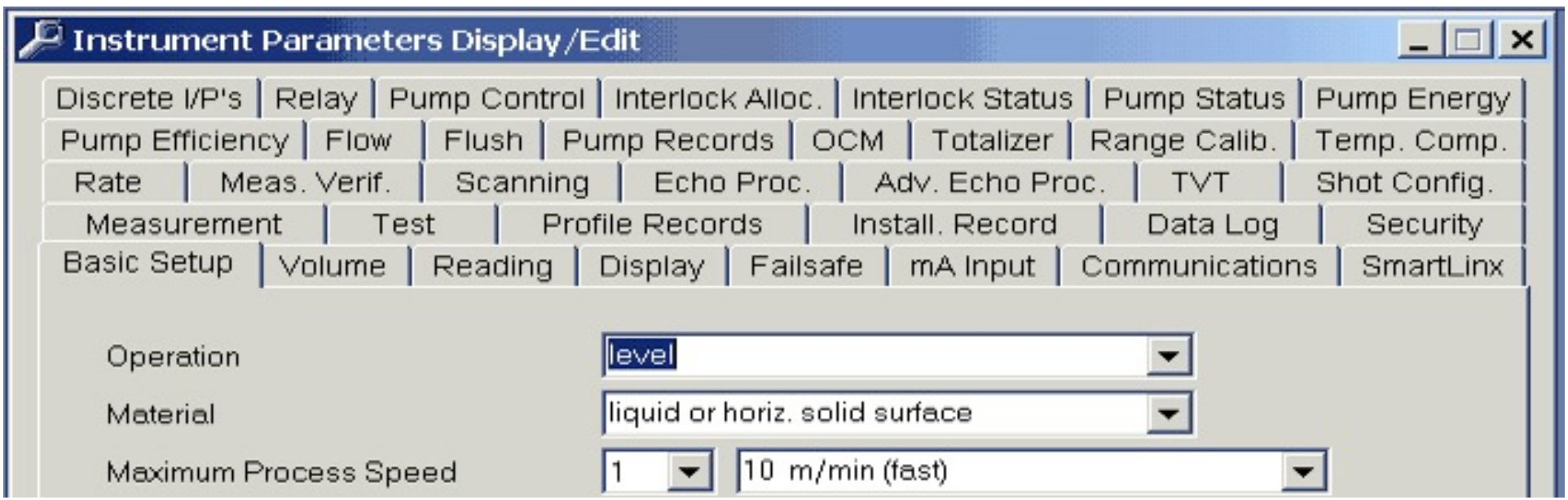
Problems with Metaphors

- Common pitfalls
  - Overly literal
    - Unnecessary fidelity
    - Excessive interactions
    - Unnecessary restrictions
  - Overly cute
    - Novelty quickly wears off
  - Mismatched
    - Does not match user's task and/or thinking





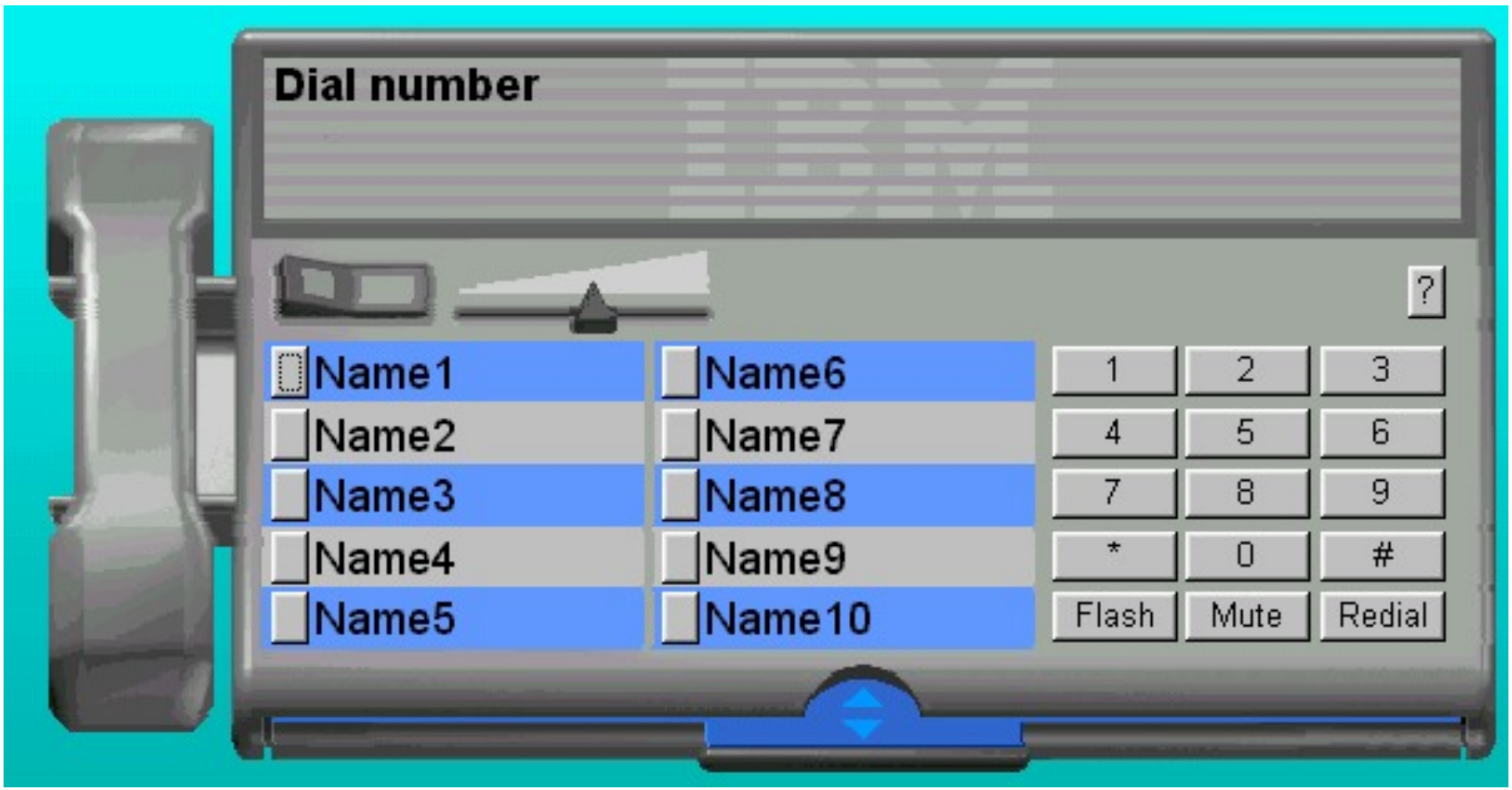
Metaphor misuses



Metaphor Misuses

Ticket counter metaphor. Okay, how do I buy a ticket?

Metaphor Misuses

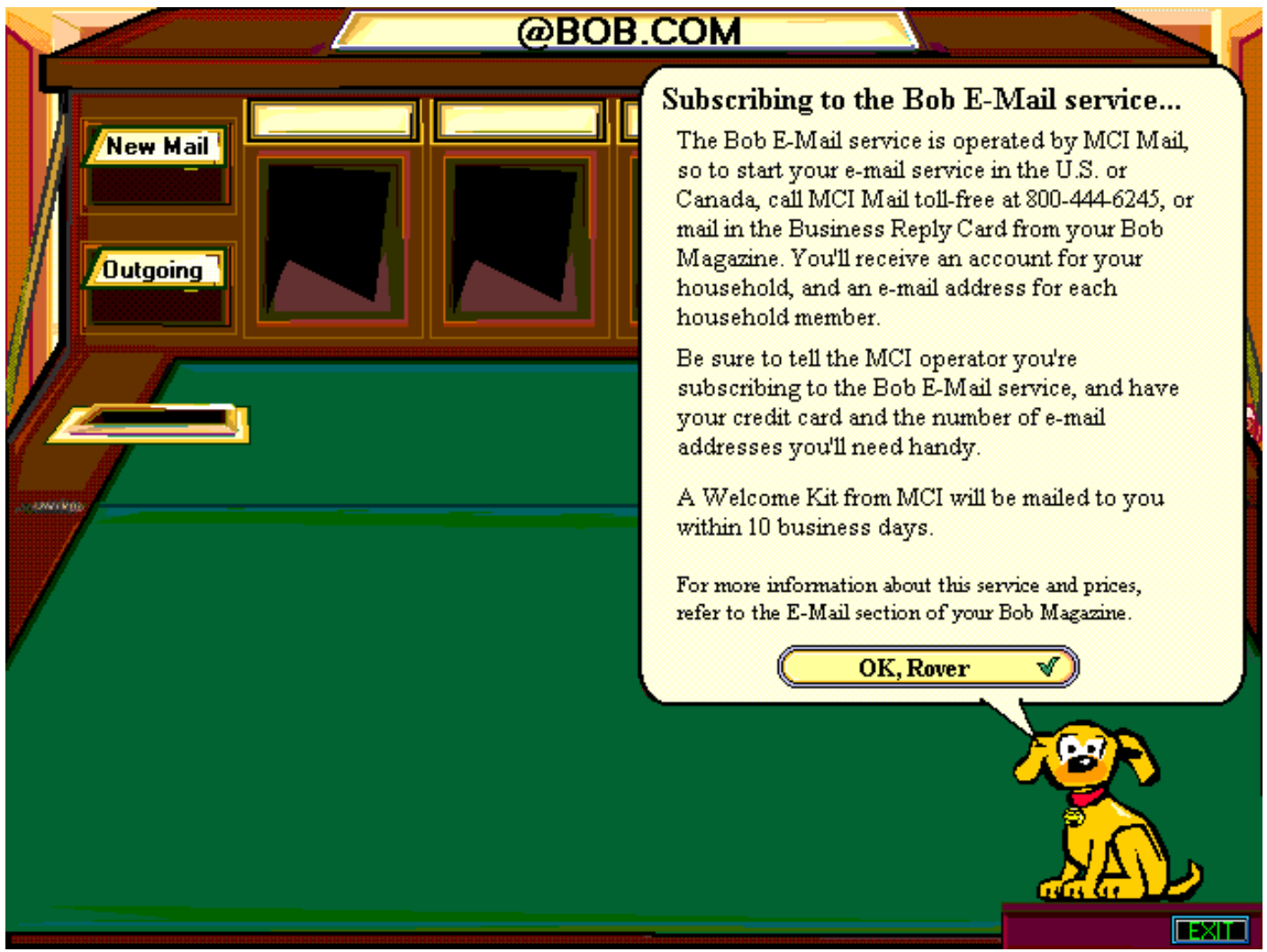


Metaphor Misuses

Microsoft Bob, 1995, designed to replace the desktop for Win 3.1 and Win 95.

Microsoft Bob

Microsoft Bob



Microsoft Bob

Conceptual Model

- Person's "mental model" of how things work
- Allows people to mentally simulate operation of a device
- Conceptual Models are built from
  - Affordances
  - Causality
  - Constraints
  - Mapping
  - Population Stereotypes/Cultural Standards
  - Instructions
  - Interaction with the device
  - Positive Transfer

Design Guidelines

- Provide good conceptual model
    - Customers want to understand how controls impact object.
- Make things visible
    - If object has function, interface should show it.
- Map interface controls to customer's model
    - Infix-vs-prefix calculator
- Provide feedback
    - What you see is what you get!

Case Study

- Scott Adams, author of the Dilbert comic strip, once spent hours moderating comments to his blog, only to have them disappear after he published them.

- What happened?

- System logs showed that he moderated the comments, and then deleted them, even though the system warned him that the files were about to be deleted.

From AskTOG, February 2006

Error 1: User Model != Design Model

- The user believed there were two databases: a “temporary working” database, and a second, publicly accessible database.
    - Metaphor from newspapers/magazines publishing — there is a copy in the office, another copy at the printers, etc etc.
- The designers believed there was only one database: simply because, there was only one database!

Error 2: Misleading Metaphor

- First move in the chain of error: The user marked the blog comments for “publishing”.
- To the public, “publish” means “to mass replicate and distribute”.
- The document is sent to the printers and multiple copies are made.
- The original draft has very little value once it has been published.
- To the blog designers (and to database people), “publish” means “to make something visible”. The original document was now marked for public reading.

Error 3: Ambiguous Confirmation Dialogs

- Second move in the chain: The user asked the system to delete what he thought were now a useless copy of the comments.

- The blog software asked the user to confirm whether he really wanted to delete the comments.

    - Message: “Delete comments?”

    - What if the blog software had said: “Remove published comments and delete this information completely from the blog?”

Error 4: No Undo Possible

- The blog software did not have an undo.
- Once the user confirmed that the deletion should go ahead, no undo is possible.
- However: the blog made it possible to undo the mistyping of a single character!
- This gave the illusion that everything could be undone.

Error 5: No Usability Evaluation

- The conceptual mistake was a serious design flaw.
- However, it would have been very easy to discover this using usability tests.