CS 66-460
Introduction to Graphical Human Machine Interfaces

The Parapsychology of Everyday Things

WEDNESDAY, JANUARY 22, 1997

Instructor: G. Bowden Wise

Rensselaer Polytechnic Institute
Spring 1997

The Frustrations of Everyday Life

- We use a variety of devices, instruments, computer programs, etc., everyday.
- Some are easy to use others are difficult and frustrating to use.
- Why are they frustrating?
- Due to poor design:
  - they provide no clues or false clues to their operation
  - they trap the user
  - they thwart the normal process of interpretation and understanding
- Poor Design leads to Frustration leads to Confusion leads to Error
- On the other hand, well-designed objects, are
  - easy to understand
  - easy to interpret
  - use visible cues to their operation

Examples of Everyday Objects

- Digital watches
  - lots of features, few controls
- Telephone Systems
  - how do I HOLD, TRANSFER, PICK-UP, PARK, etc...
- Doors
  - why are some doors easier to use than others?
- VCRs
  - how many of you let your VCR clock flash 12:00?
  - are there some features you never use?
- Household Appliances
  - have many features, and controls
  - have complicated, or useless, instruction manuals
  - users memorize just a few, important functions
  - hard for casual users (e.g., microwave)
- Voice mail systems
  - seemingly endless sequence of menus to go through to get to the desired function

A Multitude of Everyday Things

- We encounter many everyday objects.
- How do we cope?
- How do we use an object?
  - how the mind works
    the psychology of human thought and cognition
  - information in the object
    the psychology of everyday things
  - the object is usable
    the object is designed well

To Become Better Designers ...

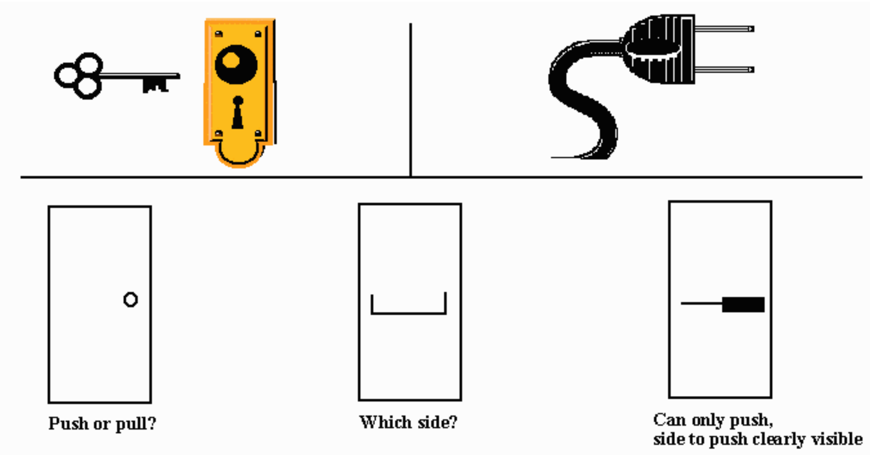
We can

- understand the psychology of everyday things
  - make use of **affordances**
  - watch out for **causality**
  - make use of **constraints**
- learn some principles of good design
  - provide a **good conceptual model**
  - make things **visible**
  - use a **good mapping**, a natural one if possible
  - provide **feedback**
  - keep the **number of features/actions and the number of controls balanced**

Make Use of Affordances

- *Affordances* refer to the perceived and actual properties of an object, which help the user determine how to use or interact with it
- When affordances are used, the user knows what to do just by looking at it
- Complex things might require some explanation, but simple things do not.
- Examples of good affordances:
  - plates (on doors): push
  - knobs: turn
  - slots: insert
  - buttons: pushing
  - chairs: for sitting
  - glass: break (!)
  - wood: write on, paint
- Others?

Some Examples of Visual Affordances



Watch Out for Causality

- When something happens immediately after an action, there is a sense of *causality*: the happening is caused by the action
- Usually just coincidence.
- Two kinds of false coincidences:
  - Incorrect effects

    This type of coincidence is common in computer systems where the user is running several applications. e.g., running a new application leads to a system crash

    Results in "superstitious" behaviors
  - Invisible effects

    An action/command with no apparent result are often repeated many times, but may actually be happening (!) e.g., repeating mouse strokes when the system is unresponsive

Make Use of Constraints

- *Constraints* restrict the allowed behavior or interaction with an object.
- **Physical constraints**

  The physical properties of an object constrain the possible operations:
  - the order in which parts can go together
  - the ways in which an object can be picked up, moved, manipulated
  - examples: scissors, doors, drawers, zippers, etc.
- **Cultural constraints**

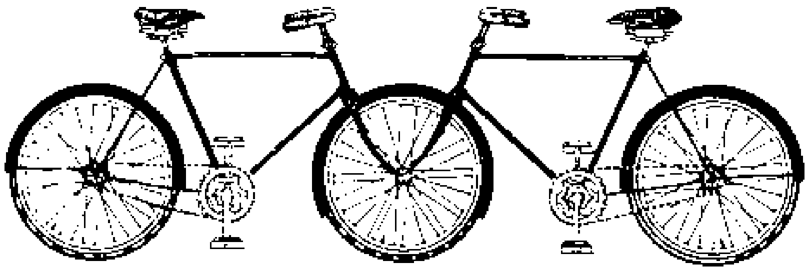
  Constraints which have evolved through artificial conventions that govern acceptable social behavior. These cultural conventions, of course, have to be learned, but once learned apply to a wide variety of circumstances.
  - tighten screws by turning clockwise
  - loosen screws by turning counter-clockwise
  - desktop metaphor used in GUIs

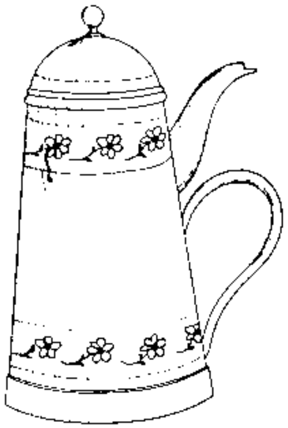
Conceptual Models

- *Conceptual models* are mental models, models that people have of themselves, others, the environment, and the things with which they interact.
- People form mental models
  - from experience
  - from training and instruction
- The mental model of a device is formed by the interpretation of its perceived actions and its visible structure.
- Upon seeing an object or a device, the user form a mental model of how the device or object operates. Users use the model to simulate the operation of a device or object.
- A good conceptual model allows users
  - to predict the effect of their actions
  - to understand the relationship between the controls of a device and the outcome
- A poor conceptual model
  - forces users to operate by rote, blindly
  - makes it difficult to determine the effects of actions
  - makes it difficult to figure out what to do in novel situations

Conceptual Models

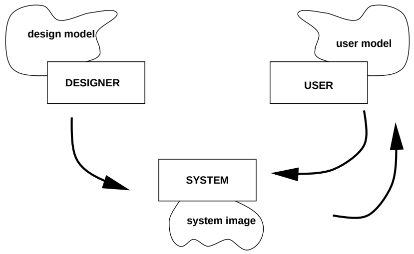
- How would you use these objects?





Three Conceptual Models

Three kinds of models:



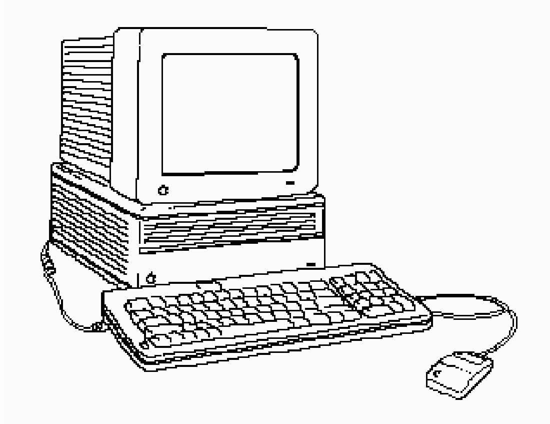
- **DESIGN MODEL**
  The designer's conceptual model
- **SYSTEM IMAGE**
  The visible structure of an object. Results from the physical structure that has been built, including, documentation, instruction, and labels
- **USER MODEL**
  The model the user develops through interaction with the system

Note that the designer would like

DESIGN MODEL = USER MODEL

But the user never sees the designer's model, only the SYSTEM IMAGE.

Conceptual Models for Computers



- Computers are much more complex than physical objects.
- As a general purpose machine, a computer does not have a conceptual model.
- Computers are used for many different purposes and run many different applications. No one conceptual model is possible.
- Designers must come up with a good underlying model that helps the user understand how to use the system.

The Use of Metaphor

- Often designers employ **metaphors** to help the user form a suitable mental model.
- Metaphors can be used to develop interfaces for applications:

APPLICATION AREA	METAPHOR	FAMILIAR KNOWLEDGE
Operating systems	The desktop	Office tasks
Spreadsheets	Ledger sheet	Columnar tables
Object-oriented environments	Physical world	Real-world behavior
Hypertext	Note cards	Flexible organization of structured text
Learning environments	Travel	Tours, guides, navigation
File storage	Piles	Categorizing objects in in terms of urgency, projects, etc
Multimedia environments	Rooms (each associated with a different medium/task)	Spatial structure of buildings
Computer supported cooperative work	Multi-agents	Travel agents, butlers, and other serving roles

- Metaphor can also be applied through analogy even if the metaphor is not concrete in the user interface. (e.g., use a word processor like a typewriter)

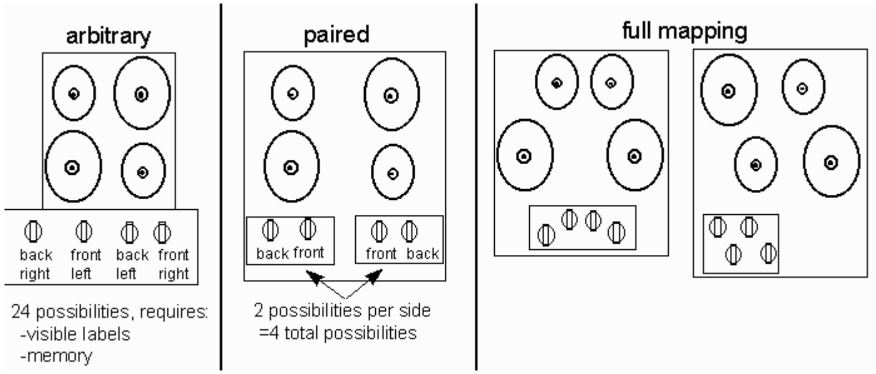
Make Things Visible

- *Visibility* is an important principle of design and is used to:
  - to make the operation of a device understandable
  - act as a reminder of what can (and cannot) be done
- Visibility is achieved by
  - making the correct parts or controls visible
  - conveying the correct message
- When the number of possible actions exceeds the number of controls, some functions become invisible, resulting in complexity.
- Good visibility leads to objects/devices
  - easier to understand
  - easier to use
  - quick to learn
  - easier to remember
- How to make things visible?
  - employ natural signals
  - use good mappings
  - good placement of controls

Make Good Use of Mappings

- A *mapping* is the relationship between two things. Many meanings.
  - e.g., mathematical definition
  - e.g., the relationship between the controls and intended function
- The easier a mapping is to learn and remember, the easier a device will be to use.
- Use natural mappings where possible.
- **Natural mappings** take advantage of physical analogies and cultural standards.
- Natural mappings come from
  - spatial analogy:
    press up button to make elevator go up
  - perception:
    louder means greater
- Some relationships do not have a natural relationship
  - e.g., pitch/hue/taste
    does higher pitch mean more or less of something?
- A device is easy to use when there is visibility to the set of possible actions actions, where the controls and displays exploit natural mappings.

Example Mappings



Provide Feedback

- *Feedback* is the act of sending information back to the user about what has actually happened as the result of his actions or about the state of the system.
- Well known concept in the science of control and information theory.
- Good example:
  - simple push-button phones
- Bad example:
  - complicated, multi-feature, modern telephones
  - Why? As number of features increase, the use of feedback diminished
- Feedback can be presented visually or aurally.
- It may cost more to develop a complex phone with good feedback, but the trade-off is increase in usability.

Manage Complexity

- Today's devices and computer systems are commonly developed with many, many, features.
- However, increase in controls/features, makes it more difficult
  - to make all the controls visible
- which makes it harder for the user to
  - understand the device
  - learn how to use it
  - memorize functions
- Keep the number of features/actions and the number of controls balanced

Multiple Views of Design Perspective

- Designing well is not easy
- There are different perspectives on what makes a good design:
  - **manufacturer:** control costs, economical to produce
  - **seller:** attractive product that customers will buy
  - **consumer:** features, price, appearance
  - **repair technician:** easy to diagnose, maintain, upgrade
- The designer must juggle all of these perspectives into a single product.
- Good design requires that the designer think about
  - how people will use the device
  - how people will abuse the device
  - the kinds of errors people will make
  - and the tasks and functions people will wish to perform

The Paradox of Technology

- Advances in technology offers the potential to make our jobs more productive, and our lives easier and more enjoyable.
- Advances can also introduce added complexity, which can increase our difficulty and frustrations.
- Example: the introduction of GUIs made computers inaccessible to blind users.
- Development of technology is U shaped
  - complex when the technology is first introduced
  - as it matures, the technology gets more comfortable, more reliable
  - as it evolves, more features are added, leading to increase in complexity
- Examples:

```
simple radios  -> tuners
analog watches -> digital watches
simple email   -> multimedia mail
```

- This paradox in technology is not an excuse for a poor design.
  - Instead, principles of good design can be used to manage complexity.

Summary

- We can become better designers by
  - understanding the psychology of everyday things
    - make use of **affordances**
    - watch out for **causality**
    - make use of **constraints**
  - learning principles of good design
    - provide a **good conceptual model**
    - make things **visible**
    - use a **good mapping**, a natural one if possible
    - provide **feedback**
    - keep the **number of features/actions and the number of controls balanced**
- Don't let the paradox of technology be an excuse for poor design!
- More easily said than done. Good design takes careful thought, planning, and hard work.