CS 160: Lecture 10

Professor John Canny
Fall 2004
Administrivia

Midterm is on wednesday.
* Material up to last Wednesday’s class
* Raise any questions you have in class today.
Knowledge representation

- Three types:
  - Analogical representations
  - Propositional representations
  - Distributed representations
Analogic representations

Objects represented internally as “images.” The image captures many of the properties of the thing.
Analogical representations supposedly support mental rotations:
Propositional representation

- Knowledge about objects captured in language-like propositions:

- Is-a(A, apple), Is-b(B, table), On(A, B)
Distributed representation

- Assumes all information represented as an “activation vector.”

- E.g. an apple coded as binary vector might be:
  10111001000110110111110110000011011010111111

- Similar objects should have similar codes.

- Supports parallel processing. Subclasses can be represented.
Connectionist view

- Connectionist view: Knowledge is arranged in networks with inhibition/activation

- Sort-of example: tools from thebrain.com
Connectionist network

Sample network

Links can be + or -
Semantic networks

- Associations between items defined by relations:
- Codes propositional knowledge.
Schema

- Generalized scripts for everyday actions:
- Eat at a restaurant
  * Enter
    + Walk in
    + Look for table
    + Decide where to sit
    + Go to table
    + Sit Down
  * Order...
  * Eat...
  * Leave...
Parking schema

- Look for a space
  - Look for people leaving
- Signal to park
- Enter the space
  - Use parts of the vehicle to line up
- Figure out how to pay
  - Look for meter
  - Is there a ticket machine/payment booth
  - Is it pay-on-exit?

...
Piaget again

- Piaget notes that there are 4 distinct stages of development.
- Stage 1 is sensori-motor or skill learning.
- Stage 2 is symbolic operations
- Then there are two distinct “reasoning” stages:
  - Concrete thought
  - Abstract thought

- e.g. children learn to navigate by concrete transformations before they learn to use maps.
Mental Models

“The models people have of themselves, others, the environment, and the things with which they interact. People form mental models through experience, training and instruction” - D. Norman

Built by activating various schema.

Not just an image: MMs allow us to visualize what would happen if we do something – close to Piaget’s notion of “concrete operation.”
Structural vs. Functional Models

A structural model explains what the system does independent of use (it’s a system-centered model).

A functional model explains what the system does to assist a user’s task (it’s a user-centered model).
Functional Models

- Should sound something like task analysis. In fact functional models have been called “task-action mapping models”.

- Develop from knowledge of using similar artifacts, not from how the artifact works.

- Generally strive to be much simpler than structural models. But...
Accountable Systems

- There is a problem if functional and structural models are too far out of step: exceptions to the functional model confuse the user, and may disrupt the task.
- System crashes
- Long delays
- Some researchers are therefore developing “accountable” systems that externalize a user-friendly structural model.
Break

- The midterm is on Weds - material up to last Weds.
- Sample midterms are online
- Any questions?
Metaphor

Since functional models draw on past experience and not everyone has computer experience, it's useful to draw on the real world.

Hence the “desktop metaphor”:
- Directories are like folders
- Files are like sheets of paper
- Windows are like ?:
- Menus are like menus
- Deleting is like putting in the trash
- Running an application program is like opening the doc.
- Copy to buffer and restore is like cut-and-paste...
Metaphor: Strengths

- Gives a way for people to understand a new concept quickly given what they know.
- Helps to provide good choices for terminology.
- Provides guidance in machine understanding of natural language.
Metaphor: Difficulties

- The metaphor may create expectations that are false along with the true ones:
  - Can I shred this file instead of putting in the trash can?
- Our understanding is “functional” rather than “structural”. That means understanding is relative to how we do things.
- For instance “work” has many meanings:
  - Something we enjoy, or dislike
  - Something that is primarily physical, vs. intellectual
  - Something that leads to a goal, vs. something we just do
Reverse metaphor

- Computer notions are permeating everyday life:
  * I had a disk crash thinking about it
  * This report is still buggy
  * We need to stop and reboot

- It will be increasingly easy to draw metaphors from popular computer systems
Most of the Star's metaphors are visual ones:
A Virtual Desktop

- The metaphor can help decide on what functionality is useful to the user.

- E.g. the sales rep who worked from a car, and said the car was like “a desk” rather than “an office”.

- Note that people experience the desktop metaphor by using it, rather than being told about it. Therefore they have a better chance to develop a functional model, and to act “by perceiving” rather than “by recognizing”.

Flexible Metaphors

People are usually happy stepping out of the metaphor:
* Scroll bars
* Resizing
* Iconifying

Users can use multiple metaphors at once, or other models based on familiar practice.

Over time, the original metaphor becomes redundant and the user has a new concept and set of skills.
Beyond a fairly early age, our understanding of concepts is personal and our conceptual models diversify.

Knowledge is usually deeply layered, so it understanding depends on our personal history.

Most people in the same society share similar experiences - e.g. the same language and the same conventions for using computers.
Development & Vygotsky

Our “environment” is a social one as well as a physical one:

* Knowledge work is about organization, communication, persuasion, motivation
* “Culture” is a set of norms guiding our behavior
* The artifacts (programs, documents) in our world also have history, and so there are even layers of meaning outside of people’s heads.
So the *shared experience* we use to guide design can be either physical as in a metaphor (Piaget) or

*Cultural, as in the conventions for MS windows (Vygotsky)*
Conceptual Models

Because of the difficulties with metaphors, designers try instead to come up with a clean “conceptual model”.

A conceptual model of an object allows the user to:

* **Identify** the object from others
* Know what **actions** can be performed on the object
* Know how the object **changes** in response to those actions.

The conceptual model is a **concrete** (Piaget) model of the system’s behavior.
Conceptual Models

A good conceptual model acknowledges human ability:
* Simplicity, how much to learn, how easy to apply?
* Limited short-term memory
* Expensive long-term memory
* Stimulus-action fusion

A good conceptual model is easy to learn: users understand it after a little bit of instruction and experimentation.
A spreadsheet has a clean, non-metaphoric, conceptual model. They have a familiar appearance, the actions are clear and intuitive, and with a little experience users understand how changes propagate.
Conceptual Model: Examples

Other clean conceptual models:
Summary

- Knowledge models provide guidance to how to plan an interface (and the user’s model of it)
- Functional models are usually the goal, but it's useful to expose some system behavior to help in exceptional situations
- Metaphors often provide a quick way to bootstrap use of an interface
- Conceptual models are more general, and can use knowledge of
  - Other systems
  - Social/cultural norms