CS 160: Lecture 9

Professor John Canny
Spring 2004
Feb 20
Administrivia

Midterm is a week from today.
* Raise any questions you have in class today.
Knowledge representation

Three main types:

- Analogical representations
- Propositional representations
- Distributed representations
Connectionist view

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

- Sort-of example: tools from thebrain.com
Semantic networks

Associations between items defined by relations:

![Semantic network diagram]

- Animal
  - Tiger
  - Cat
  - Dog
  - Pet
  - Canary
- Bird
  - Feathers
  - Sparrow
  - Robin
  - Ostrich
- Skin
- Breathes
- Sings

- Red
- Orange
- Yellow
- Green
Schema

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

- Something like activated schema.

- Not just an image: MMs allow us to visualize what would happen if we do something
Some Piaget

- Piaget notes that there are 3 distinct stages of development.
- Stage 1 is sensori-motor or skill learning.
- But then there are two distinct “abstract” stages:
  - Concrete thought
  - Abstract thought

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

Metaphor is basic to human language for a similar reason: it allows us to talk about knew or abstract things by drawing on familiar experience:

- Time is like a line we move on
  - We can go forward and look back
  - We can push a meeting back

- Love is like a journey (also like a fall)

- Even abstract fields, like math or quantum mechanics, are rich with metaphors
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
Break
Development & Vygotsky

Beyond a fairly early age, our understanding of concepts is personal and our conceptual models diversify (Piaget)

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

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

2. 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.
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.
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.
Conceptual Model: Example

- 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.
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