

CSC 428F/2514F

HUMAN-COMPUTER INTERACTION

Lecture 8
1997 October 1

HUMAN COGNITION IN RELATION TO COMPUTERS

presented by
Ben Barkow

8.1	Purposes for learning about cognition.....	2
8.2	What is cognition?.....	3
8.3	Memory/Learning.....	4
8.4	Thinking / Decision Making.....	5
8.5	Applying principles of cognition to interface design.....	6

Ronald Baecker
Professor of Computer Science
University of Toronto

The Shepherd tones mentioned in Lecture 7 notes can be found at:
<http://asa.aip.org/sound/cd/demo27a.au>

1. Purposes for learning about cognition: without cognitive mechanisms to further process perceptions, few stimuli (reflexes excepted) would lead to performance

1. Some cognitive “illusions”

1. “Funny thing, when I first tried this software it seemed very hard to use, but now I think it is easy to use”
2. “They are really crazy drivers in [any new place at all, even Toronto]”
3. Sometimes, a specific period of time passes slowly but in retrospect it passed quickly (or the reverse)
4. Various figure/ground cognitions: don’t think of a hippopotamus; the example of SIGCHI text of last week
5. Depressive thoughts (or manic thoughts)
6. “I’m normal”
7. “I know you will find it fascinating to see my [vacation pictures, very cute new baby, or nifty interface graphics]”

2. Introduction: no sharp line dividing perception, cognition, or performance (for example, in remembering faces)

The degree of intervention of cognitive processes is always very great. But the goal of interface design is to devise stimuli which call-up and engage mechanisms of perception/cognition which are *favourable* in light of being for most users:

1. fast
2. trustworthy
3. “easy” in the sense of minimizing the mental cognitive burden and/or
4. otherwise in keeping with the designer’s purposes (for example, establish patterns or parallels with other user activities, benefit from transfer of training, motivating, etc.)

But if you are making entertainment software, then you might wish to choose if and when you want to violate these principles. Hint: let your *content* bear the fun, not your interface. Parents always think their baby is cute.

2. What is cognition?

1. It is somewhat arbitrary what distinguishes *perceptual mechanisms* from *cognitive processes*. Cognition includes:

1. *memory* which is longer than is needed to “see” a percept
2. aspects of *learning* which is memory which is longer than to recall an image
3. mental processing intervals called *reaction times* and *decision times* during which perception, cognition, and performance are connected and resulting in some behaviour or experience, and in common parlance are called “thinking”

2. Concepts:

1. channel capacity or mental load (much the same as in E.E., and suggests a finite, if flexible, capacity for handling data in the cognitive realm in respect of the number of processing steps needed, the volume or substance being processed, the state of the mental machinery, (and unlike E.E.) the meaningfulness of the material, etc.
2. detectability and discriminability are the processes of noting that a mental entity is (a) recognized and the same as or (b) different, as compared to something else in memory.

3. Memory/Learning

There is short-term memory: like a register which holds a bit of information for, say 20 milliseconds; if there is disruption during this period, the memory fails to enter long-term memory; trauma as in a car accident interrupts short-term memory.

Long-term memory appears to be phenomenally good!!! Not only do re-rehearsed items get recalled but evidence of re-learning savings of lab-grade nonsense can be demonstrated after 10 years.

In interface design, build on “the secrets” of good memory:

1. avoid depending on or invoking memory or learning — the perfect program has no manual
2. natural harmony or transfer of training (don't call the King Street station “St. Andrews” say, “Kong” or “Monarch”)
3. distinctiveness or antic touch (don't say “Enter at the 18th stainless steel door to your left” say, “the door next to the purple trombone”)
4. it is acoustic things which best pass through the door of memory (painting the whole floor blue is worse than just having a sign which says “This is the BLUE FLOOR” — except for architects)
5. engage intentionality (tell users, “Folks, you gotta remember this _____”); provide every sort of hint, reminder, and support for rehearsal
6. make memory demands as mutually distinctive as possible (What could be worse than the instruction “take highway 401 to highway 400 past highway 407 and then get off at highway 402”?)

There may be “incidental learning,” (learning which takes place in the background of the person's experience), but don't count on it.

4. Thinking/Decision Making

1. People are essentially “analog” but helping people package information into groups not exceeding say, 7 can be very helpful.
2. Choice reaction time: the length of time it takes to choose among N alternatives
3. Analogies (“life is like a blueberry tart...”) can be helpful when (a) the analogy advances the user’s understanding, (b) parallels the situation in appropriate behavioural aspects or (c) substitutes for an inconceivably complicated underlying mechanism. Would the Mac interface been better if the metaphor was a computer -- if the “trash” was just “pre-delete list” and “desktops” just looked like diskettes?
4. The more tasks underway, the poorer each one is handled. For example, it is hard to deal with “EXPRESS LANES CROWDED BEYOND NEXT TRANSFER COLLECTOR LANES CONGESTED” as compared to “EXPRESS LANES MORE CROWDED THAN COLLECTORS FOR NEXT TWO KILOMETERS”

5. Applying principles of cognition to interface design

1. Are GUIs more or less human-memory intensive than menu-driven interfaces? How would you measure the difference?
2. Keyboard shortcuts in a GUI... what are the best concepts to apply in creating keyboard shortcuts?
3. Think of a program which obliges you to learn or remember a lot of arbitrary information. Can you reconfigure this interface from a rote memory format to a “natural” or “mental map” format.
4. Transfer of training from an earlier or conventional interface approach (or from a pre-computerized situation such as the kitchen sink or Ben’s Smart Pencil) to your new program:
 1. when to build on the past and when to introduce changes
 2. can you signal the user when *the past* will work in the new program
 3. how can you test your ideas?