Why Cognitive Architectures for Human Factors?

Outline

- **Why CAs for Human Factors? Overview**
- **What is Cognitive Modeling and How does it differ from other approaches to modeling?**
  - **With examples**
- **How is Cognitive modeling different from cognitive task analysis?**
- **Why Architectures?**
- **Production System basics**
- **Cognitive models as Simulated Human Users (SHUs)**

Why Cognitive Architectures for Human Factors? An Overview

- Cognitive modeling is both a research tool for theory building and an engineering tool for applying theory
- To the extent that the theories are sound and powerful, cognitive modeling can aid Human Factors in the design and evaluation of interface alternatives
- To the extent that the problems posed by Human Factors are difficult to model or cannot be modeled, Human Factors has served to pinpoint gaps or inconsistencies in cognitive theory

Why Cognitive Architectures for Human Factors? An Overview

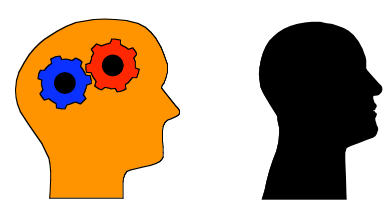
- Science is an iterative process
- The symbiotic relationship between modeling and Human Factors furthers the scientific enterprise of cognitive science and the engineering enterprise of human factors

Cognitive Modeling

- **Cognitive modeling is a form of task analysis**

Glass box versus black box modeling

What is meant by cognitive modeling?



Why Models?(1)

- **Model Frameworks**
  - Theory-based: Examples, ACT-R (Anderson, 1993) and Soar (Newell, 1990).
  - General cognitive: Examples, GOMS (Card, Moran, & Newell, 1983) and GEMS (Reason, 1990).
- **Inspectability/Objectivity**
  - The model gives the answer not the modeler

Why Models?(2)

- **Mnemonic value**
- **Permit us to focus on complex behavior**
  - for application
  - for development/testing of new microtheories
  - for development/testing of the model-specific control structures
  - to handle cognition (the ant) while we focus on understanding the task environment (the beach)

Why Models?(3)

- **A successful model becomes an artifact in its own right**
  - tools for cognitive engineering
  - Example, CPM-GOMS from models Project Ernestine
  - Argus models -- recent HFES presentation
- **Cognitive modeling as a technology for applying cognitive theory**

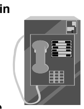
Three Examples

- **Project Ernestine**
  - Gray, John, & Atwood, 1993
- **Postcompletion Error**
  - Byrne & Bovair, 1997
- **Information Access**
  - Altmann & John, 1999

The Précis of Project Ernestine

An Overview of a Validation of Cognitive Modeling

- **Interest**
  - Modeling: parallel performance; event-driven; complexity in the world; transfer
  - Application: can modeling be used to evaluate design; operating expenses = $3 million/sec
  - Theory: can cognitive theory be applied via cognitive modeling to predict time-critical human performance in the real-world
  - Validation of both modeling & the cognitive approach
- **See Gray, John, & Atwood (1993) for more details**



Byrne & Bovair -- Postcompletion Error

- An adequate theory of error "is one that enables us to forecast both the conditions under which an error will occur, and the particular form that it will take"
- Examples of postcompletion errors
- Definition
  - From the perspective that task and device are distinct, any action that the device (the ATM) requires us to perform after we complete our task (withdrawing cash) is a postcompletion action. An omitted postcompletion action is thus a postcompletion error.

Postcompletion error

- Although they occur, they do not occur often
  - Most people, most of the time, take both the money and the card from the ATM (else, we suspect many fewer of us would use ATMs)
- What, if anything, predicts the occurrence of a postcompletion error?

Postcompletion error

- Byrne & Bovair offer an activation based account
  - Assumes that a memory element is accessible only if it has enough activation
  - Also assumes that total activation is limited
  - Activation flows from one memory element to another if the two are related and if one is the focus of attention
  - This spreading activation accounts for standard psychological effects like semantic priming, in which, for example, focusing on the notion of "doctor" might spread activation to related concepts like "nurse."

Postcompletion error

- As long as the focus is on a task goal like getting money, related device actions like take the card continue to receive activation
- However, when a task goal is accomplished, attention shifts away from it and device actions associated with the task begin to lose activation
- This is fine for actions like take the money, which are necessarily complete, but problematic for postcompletion actions like take the card
- If these postcompletion actions lose enough activation, they will simply be forgotten.

Postcompletion error

- Predictions
  - The more strongly other items in WM are activated, the less activation of the postcompletion action and the more likely it will decay before being remembered
- Data
  - Fewer postcompletion errors in condition with a prolonged tracking task
- Theory-Based Explanation
  - During tracking task the completed actions of the main task decayed
  - More activation available for the not-yet-accomplished postcompletion goal

Postcompletion error as Applied Theory

- Explanations and predictions flow from existing cognitive theory, not from ad hoc assumptions made by the analysts
  - The model functioned primarily as a means of instantiating a theory on a particular problem
- Prediction comes from the model not the modeler
  - Any analyst could run the postcompletion error model with the same outcome
  - The debate over this outcome is then limited to and focused by the representational assumptions and parameter settings reified in a running computer program

Information Access -- Altmann & John

- Information in the world is useful only if we can find it when we need it. For example, an illustration in a book is helpful only if we know it exists, if we recall its existence when it is needed, and if we can find it

Information Access

- Studied the behavior of one programmer making changes to code that had been written over a series of years by a team of which the programmer was a member in one 80-min session
- During this session, the programmer would trace the program for several steps, stop it, interrogate the current value of relevant variables, and so on
- Over the course of the session 2,482 lines of code were generated and displayed on the programmer's screen. On 26 occasions, she scrolled back to view information that had appeared earlier but had scrolled off the screen
- Can we explain these 26 scrolling events (information access)

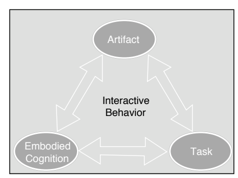
Information access

- Scrolling was not random -- pretty much went right to part of the 2,482 lines that they were interested in
- This implies that scrolling was governed by a specific memory triggered by a specific cue

Information access

- The episodic indexing model suggests that memory depends on attention, not intent
  - Episodic chunks are stored in memory as a by-product of attending to an object, with no need for any specific intent to revisit that object later
- The implication is that people store vast amounts of information about their environment that they would recall given the right cues.
- The key to unlocking this potential is to analyze the semantic structure of the knowledge being browsed and to ask how artifacts might help produce good cues later when the browsed information would be relevant

ETA Triad



Cognitive modeling versus Cognitive task analysis

- Cognitive modeling goes beyond cognitive task analysis in that each step is grounded in theory
- Also contains an explicit control structure

Dimensions of Cognitive Models

- Generative versus Descriptive
  - Project Ernestine, postcompletion error, information access?
  - Advantages of generative models
    - Sufficiency proof
    - Glass box
    - Reduced opportunity for analyst error

Dimensions of Cognitive Models

- Generality versus realism
  - Generality is the extent to which a model offers theoretical implications that extend beyond the model's domain
  - Realism is the extent to which the modeled behavior corresponds to the actual interactive behavior of a particular operator performing a given task

26 Rensselaer | Cognitive Science

Generality versus realism

- Project Ernestine
  - Behavior for an entire task; that is, one phone call of a particular call category for a particular workstation.
  - These models were not general in that it would be difficult to apply them to any task other than the one modeled

27 Rensselaer | Cognitive Science

Generality versus realism

- Postcompletion error and Information Access Models
  - Incomplete accounts of behavior (could not do entire task)
  - But, the scope of the models extend far beyond the tasks in which they are based
- For example,
  - If postcompletion actions cannot be designed out of an interface then special safeguards against postcompletion error must be designed in.
  - Likewise, episodic indexing suggests that human cognition reliably encodes a little information about whatever it attends to. With the right cue, this information can be retrieved.
  - Both the PCE and Information Access model bear on any artifact-task combination in which memory is an issue

28 Rensselaer | Cognitive Science

The Bigger Picture

Why Architectures?

29 Rensselaer | Cognitive Science

Divide-and-Conquer?

- Traditional Experimental Psychology adopts a divide-and-conquer approach
- Focus on an isolated aspect of human cognition and try to understand it through rigorously controlled experimentation
- Need to understand all the pieces before they can be put together
- Newell said "You can't play 20 questions with nature and win" (1973)

30 Rensselaer | Cognitive Science

Cognitive Science as Software Engineering

- The pieces of cognition are NOT independent!
- Numerous constraints exist among all the pieces and we need to try to understand from the beginning how they fit into the whole
- Architectures provide an approach that tries to fit in "all the pieces"

31 Rensselaer | Cognitive Science

What is a Cognitive Architecture?

- "A cognitive architecture embodies a scientific hypothesis about those aspects of human cognition which are relatively constant over time and relatively independent of task." Howes & Young (1997)
- Cognitive architectures vs traditional cognitive theorizing
  - Micro-theories of isolated phenomena or mechanisms (Newell, 1973), vs integration.
  - Emphasis on the control structure of cognition.
  - Cast in a form amenable to computer implementation, so that individual models built within the architecture take the form of computational cognitive models.

32 Rensselaer | Cognitive Science

Motivations for a Cognitive Architecture

- 1. Philosophy: Provide a unified understanding of the mind.
- 2. Psychology: Account for experimental data.
- 3. Education: Provide cognitive models for intelligent tutoring systems and other learning environments.
- 4. Human Computer Interaction: Evaluate artifacts and help in their design.
- 5. Computer Generated Forces: Provide cognitive agents to inhabit training environments and games.
- 6. Neuroscience: Provide a framework for interpreting data from brain imaging.

33 Rensselaer | Cognitive Science

These Goals for Cognitive Architectures Require

- 1. Integration, not just of different aspects of higher level cognition but of cognition, perception, and action.
- 2. Systems that run in real time.
- 3. Robust behavior in the face of error, the unexpected, and the unknown.
- 4. Parameter-free predictions of behavior.
- 5. Complete learning.

34 Rensselaer | Cognitive Science

Constructing a model within one of these cognitive architectures involves writing a program in a

- Powerful,
- Restricted, &
- Specialized

programming language

35 Rensselaer | Cognitive Science

Why are Cognitive Architectures important for Human Factors?

- CAs show how different aspects of cognition can work together to perform tasks with interactive systems
  - Enable integrated cognitive systems!!
  - More than just memory or just attention or just perception
  - Integrated cognitive systems require powerful architectures that handle the control of interactive behavior
  - Required to do a complex task such as using a computer interface

36 Rensselaer | Cognitive Science

Why are Cognitive Architectures important for Human Factors?

- CAs show how different aspects of cognition can work together to perform tasks with interactive systems
- When confronted with an interface, the cognitive modeller asks "How would C-I (or EPIC, or Soar, or ACT-R, etc.) perform the task using this design?"

37 Rensselaer | Cognitive Science

Why are Cognitive Architectures important for Human Factors?

- CAs show how different aspects of cognition can work together to perform tasks with interactive systems.
- When confronted with an interface, the cognitive modeller asks "How would C-I (or EPIC, or Soar, or ACT-R, etc.) perform the task using this design?".
- Can be applied to tasks involving both routine skill and problem solving
  - Routine interactive behavior entails more control knowledge than does problem-solving
  - Most tasks entail elements of both routine interactive behavior and problem-solving

Why are Cognitive Architectures important for Human Factors?

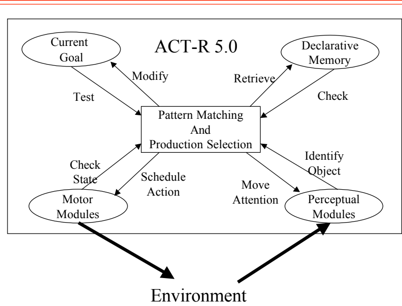
- Generative vs descriptive modeling
- Only models based on cognitive architectures can model the learning process itself
  - Can predict how a different history of learning tasks or instructions will result in a different learning outcome
  - Can predict how forgetting will affect the casual user of complex systems

Two Important Caveats

- Cognitive theory is not the only theory relevant to Human Factors. For example,
  - research on computer supported cooperative work (CSCW) brings a variety of cognitive and non-cognitive sciences to bear on HCI issues.
- Not all cognitive processes are understood well enough to be incorporated into existing cognitive architectures
  - E.g., speech understanding, visual perception, etc

Production System Basics

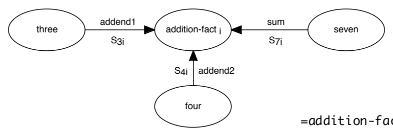
The ACT-R Architecture



ACT-R Claims

- "We think now that we can claim that productions define the units with which thought progresses and chunks define the units with which knowledge is retrieved from declarative memory" p. 13 A-C-T.
- Chunks and productions are as far down as one can go in the symbolic decomposition of thought -- 50-100 msec cycle time
- Are levels above and levels below the atomic level

Symbolic components of ACT-R




```
=addition-fact>
    isa      addition-fact
    addend1  three
    addend2  four
    sum      seven
```

LBR'99 Fig1. An ACT-R declarative memory element

"The main difference among ACT-R models is not in their way of processing information but in the initial knowledge with which they are endowed."

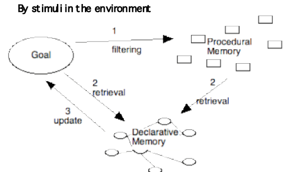
Sample Production

```
(p retrieve-arith-fact
    CONDITION
    =goal>
        isa       add-goal
        addend1   =addend1
        addend2   =addend2
        sum       nil
    =add-fact>
        isa       addition-fact
        addend1   =addend1
        addend2   =addend2
        sum       =answer
    ACTION
    =goal>
        sum       =answer)
```

How it works.

- Goal acts as a filter to select only those productions relevant to its current state
- Production with highest expected utility is selected
- Production may specify retrieval. This retrieval is influenced by both the current goal (via its connections to working memory) and the production (via the retrieval pattern specified in its condition).
- Contents of the retrieved node are used to update the current goal according to the production's action specification

- Processing of information is driven by current goal of the system (rationality principle again)
- Current goal
  - Contains the information in the focus of attention
  - Uses a declarative node structure (i.e., you can think of the goal as a specialized declarative memory element).
  - Established by previous processing or
  - By stimuli in the environment



Subsymbolic Processing

- **Base level activation**
  - Each use of a piece of knowledge or node or declarative memory element boosts the activation of that node
  - Activation decays as a function of time since the last use. Basically BLL reflects a node's prior history of use
- **Source activation (W -- dubya)**
  - Accounts for the effects of current context
  - For each goal, a fixed amount of source activation is divided among the things to which you are currently paying attention

Trace of A&L'98 Prods for 239 + 125

- **Cycle 0: Start-Problem**
  - **Start-Problem**

    IF the goal is to do an addition problem
    but no column has been identified
    THEN set a subgoal to add the digits in the ones column
    and note that the tens column is the next one to work on

Trace of A&L'98 Prods for 239 + 125

- **Cycle 0: Start-Problem**
  - **Cycle 1: Read-Number1**
    - **READ-Number1**

      IF the goal is to add the numbers in the column
      and the first number has not been encoded
      THEN encode the first number in the column

Trace of A&L'98 Prods for 239 + 125

- Cycle 0: Start-Problem
  - Cycle 1: Read-Number1
  - Cycle 2: Read-Number2
    - READ-Number2

      IF the goal is to add the numbers in the column
      and the second number has not been encoded
      THEN encode the second number in the column

Trace of A&L'98 Prods for 239 + 125

- Cycle 0: Start-Problem
  - Cycle 1: Read-Number1
  - Cycle 2: Read-Number2
  - Cycle 3: Add-Numbers
    - Add-Numbers

      IF the goal is to add the numbers in the column
      and another number is there sum
      THEN note that other number as the sum

Trace of A&L'98 Prods for 239 + 125

- Cycle 0: Start-Problem
  - Cycle 1: Read-Number1
  - Cycle 2: Read-Number2
  - Cycle 3: Add-Numbers
  - Cycle 4: Extract-Answer
    - Extract-Answer

      IF the goal is to add the numbers in the column
      and the sum has been computed
      and the sum has a ones digit and a tens digit
      THEN note the tens digit as the digit to be carried
      and set the answer to the ones digit

Trace of A&L'98 Prods for 239 + 125

- Cycle 0: Start-Problem
  - Cycle 1: Read-Number1
  - Cycle 2: Read-Number2
  - Cycle 3: Add-Numbers
  - Cycle 4: Extract-Answer
  - Cycle 5: Write-Answer
    - Write-Answer

      IF the goal is to add the numbers in the column
      and there is no carry
      and the to-be-carried digit has been determined
      THEN write the answer in the answer row of the column
      and pop the goal

Subgoal of "add the numbers in the column has been popped" top level goal is now "do an addition problem "

Trace of A&L'98 Prods for 239 + 125

- Cycle 0: Start-Problem
  - Cycle 1: Read-Number1
  - Cycle 2: Read-Number2
  - Cycle 3: Add-Numbers
  - Cycle 4: Extract-Answer
  - Cycle 5: Write-Answer
- Cycle 6: Next-Column
  - Next-Column

    IF the goal is to do an addition problem
    and the column to add has been determined
    and the carry is known
    THEN set a subgoal to add the digits in that column with the carry
    and note that the next column is the one to add

Cognitive Models as Computer Users

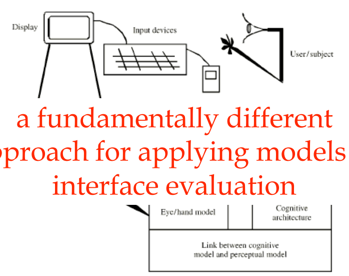
Main Ways That Models Have Been Applied to Human Factors

- To predict time
- To assist users
- As surrogate users

Why Embodied Models (aka integrated cognitive systems)?

- Models with "hands" and "eyes"
- Situate the models in an environment
- Better timing!
- More complete -- can simulate a wider array of activities
- Easier to use models in usability evaluation

Why Embodied Models?



a fundamentally different approach for applying models to interface evaluation

Embodied Models

- Although the model does not physically use the input devices of mouse and keyboard or optically see the display, so far as the running software is concerned it does use the same interface: it has access to the information on the display, and it generates events that are interpreted as key presses and mouse clicks (R&Y, p.7)

Unique contributions to interface evaluation

- Embodied models can provide automatic evaluation of aspects of the interface that depend on the user's dynamic performance, rather than being restricted to static measures
- Allows designers to appreciate the "unfolding of performance" better than less dynamic technique
- Can be carried out directly with early versions of the interface (in principle!)

Testing Users versus Testing User Models

- Testing humans is inherently expensive
- If you need novice users, you can only test people once
  - Cannot rerun humans on variants of the same interface
- Can be large individual differences between subjects
- User testing only tells what happens, testing user models tells what and why!
- User models may be tested in simulated real-time

Testing Users versus Testing User Models

- But --- need for user testing will not disappear, only its role will change
  - Initial model development
  - Validation of models
  - Final acceptance of design