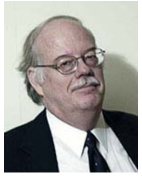
An Example Cognitive Architecture: EPIC



David E. Kieras

Collaborator on EPIC: David E. Meyer

University of Michigan

EPIC Development Sponsored by the Cognitive Science Program
Office of Naval Research

Powered by Naval Research TM/SM

Introduction to the Session

The basic idea of a cognitive architecture is to represent the fixed, task-independent constraints in a way that can be "programmed" to do a specific task.

This session (briefly) presents a specific example of a cognitive architecture that illustrates this concept.

Then walk through a series of screen shots showing the architecture at work doing a task.

Description of the EPIC Architecture

The EPIC Architecture

Diagram of the Current EPIC Architecture

Example Structural Detail - Visual System

Perceptual Processors

Perceptual Processors (continued)

Motor Processors

Motor Processors (continued)

Cognitive Processor

Cognitive Processor (continued)

Sample Rules - 1

Sample Rules - 2

Modeling Issues - Inputs and Outputs

Modeling Issues - Fixed and Free Parameters

Example of an EPIC Model at Work

The EPIC Architecture

An architecture developed to represent executive processes that control other processes during multiple task performance.

*E*xecutive-*P*rocess *I*nteractive *C*ontrol
Kieras & Meyer, mid-1990s

Basic assumptions

Production-rule cognitive processor.
Parallel perceptual and motor processors.

Fixed architectural properties

Components, pathways, and most time parameters

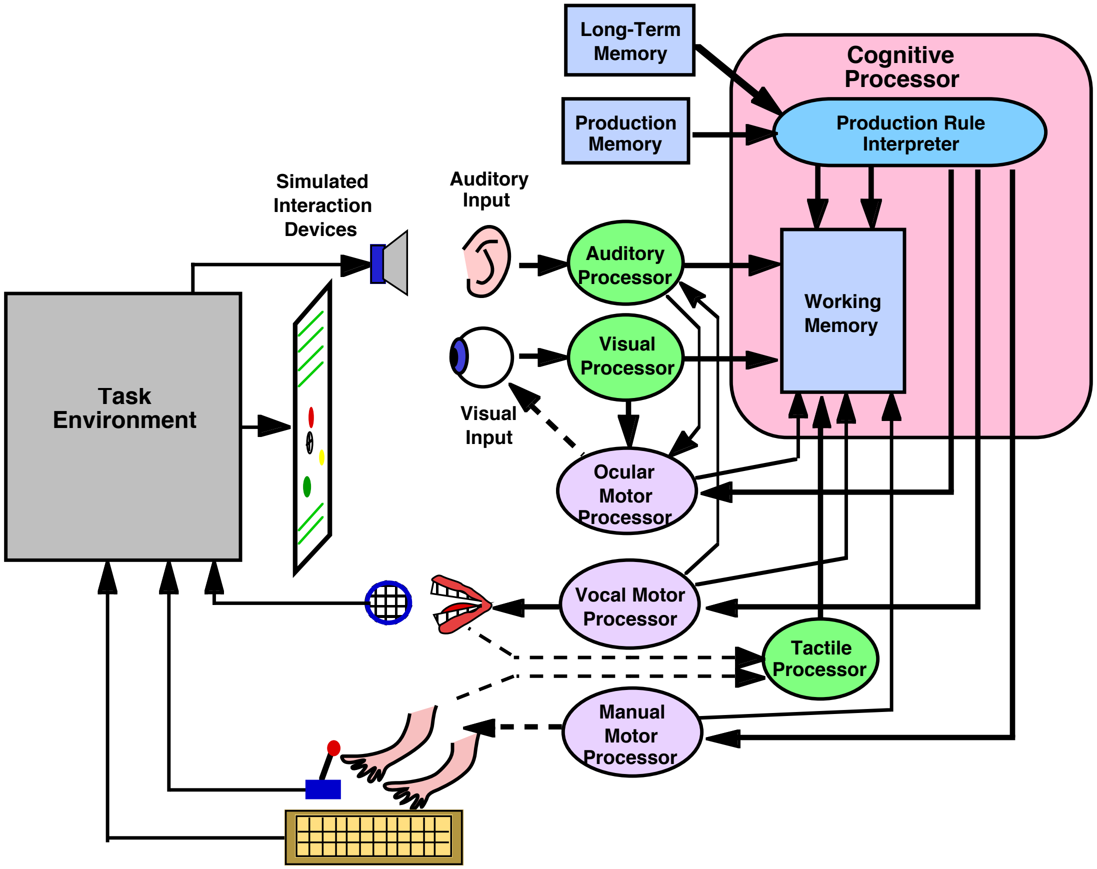
Task-dependent properties

Cognitive processor production rules.
Perceptual recoding.
Response requirements and styles.

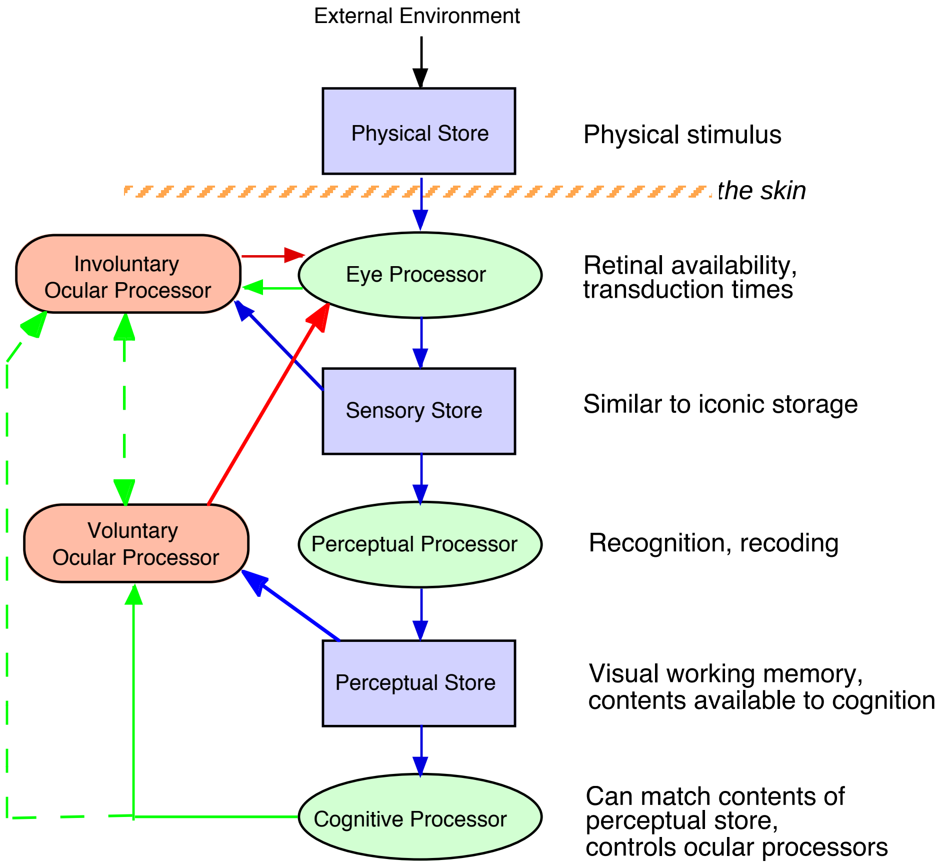
Currently, a performance modeling system.

Theory of human performance not exactly finished - plenty of work still to be done!
But learning mechanisms being planned.

Diagram of the Current EPIC Architecture



Example Structural Detail - Visual System



External Environment

Physical Store

Physical stimulus

the skin

Involuntary Ocular Processor

Eye Processor

Retinal availability, transduction times

Sensory Store

Similar to iconic storage

Voluntary Ocular Processor

Perceptual Processor

Recognition, recoding

Perceptual Store

Visual working memory, contents available to cognition

Cognitive Processor

Can match contents of perceptual store, controls ocular processors

Perceptual Processors

Inputs

Symbolically-coded changes in sensory properties.

Outputs

items in modality-specific partitions of Working Memory.

Visual

- Eye model filters input depending on visual eccentricity - distance from Fovea.
  Simple zone model: Fovea, Parafovea, Periphery.
  More realistic: Size, type of property, governs availability.
- Visual properties take different times to transduce.
  Detection: Timing: 50 ms.
  Shape information: Timing: 100 ms, typical.
- Encodes additional perceptual properties which make up Visual Working Memory.
  Timing: Additional 150 ms, typical.
- Maintains internal representation of visual objects.
  Location information directly available to motor processors.
- Certain changes reported to the Ocular Motor Processor.
  Onsets, movement.

Perceptual Processors (continued)

Auditory

- Detects onsets and offsets.
  Timing: 50 ms.
- Encodes tones, sounds.
  Timing: 285 ms, typical.
- Outputs speech input as a temporally-chained representation that decays with time.
  Timing: 400 ms, typical.

Tactile

Passes through kinesthetic feedback from motor effectors that positively identifies movement states.
Timing: 100 ms.

Motor Processors

Inputs

Symbolic instructions from the cognitive processor.

Outputs

Symbolic movement specifications and times.

Motor processing

Movement instructions expanded into motor features.

- E.g., style, effector, direction, extent.

Motor movement features prepared.

- Features can be prepared in advance or re-used.

  Later execution is faster.

Movement is physically executed.

Timing:

50 ms/feature preparation.
50 ms movement initiation delay.
Movement-specific execution time (e.g. Fitts' Law).

Cognitive processor informed of current state.

Motor Processors (continued)

Ocular Motor Processors (voluntary & involuntary)

Generates eye movements from commands or visual events.

- Long-loop cognitive control - voluntary processor.
  Saccades.
- Short-loop visual control - involuntary processor.
  Saccades and smooth movements.

Manual Motor Processor

Both hands are controlled by a single processor

- A fundamental limitation.

A variety of hand movement styles.

- Pointing, button pushing, controlling.

Vocal Motor Processor

Not very elaborated at this time.

Cognitive Processor

Perceptual-motor stores and processors operate in full parallel with cognitive processor.

Cognitive processor uses Parsimonious Production System (PPS).
- Very simple syntax and semantics.
- Rete match implementation.
- Memory items are simply lists of symbols.
- Match & fire in cycles, 50 ms period.

Production rules can fire in parallel.
- Any number can fire during a cycle.
- All rules whose conditions match will fire.
- All instantiations of a rule’s condition will fire.
- Not unlimited processing power because of peripheral limitations.

No implicit flow-of-control mechanisms.
- No “hard-wired” goal stack, data refractoriness, etc.
- Flow of control must be explicit in rules.

Cognitive Processor (continued)

Production rules triggered by items in Working Memory.

Rules can add and remove items from Working Memory.

Working Memory partitions:

Modal stores:

- Visual

  Represents current visual situation, as limited by visual system.

  Slaved to visual input.
- Auditory

  Items disappear with time - used for verbal short-term memory.
- Motor

  States of motor processors.

Control store:

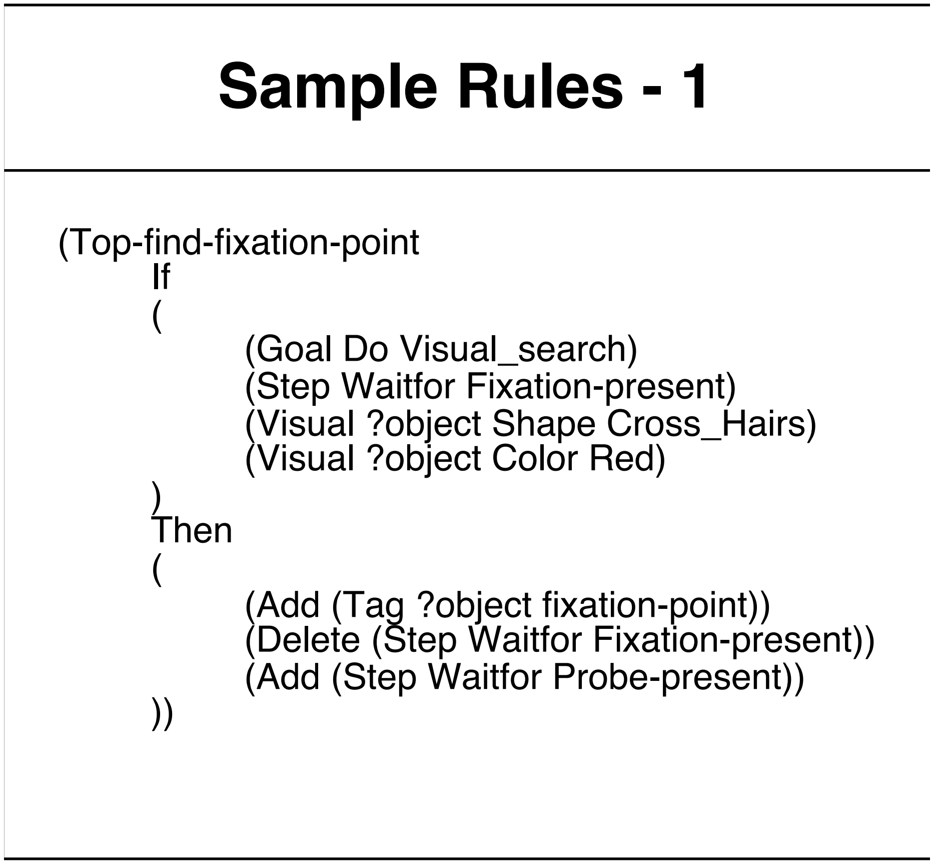
- Goal, Step, Strategy, Status items for method control and sequencing.

Tag store:

- Associates a modal working memory item with a symbol designating a role in production rules - analogous to a variable and its binding.

Amodal WM:

- Additional information whose psychological status is not yet clear.



Sample Rules - 1

```
(Top-find-fixation-point
        If
        (
                (Goal Do Visual_search)
                (Step Waitfor Fixation-present)
                (Visual ?object Shape Cross_Hairs)
                (Visual ?object Color Red)
        )
        Then
        (
                (Add (Tag ?object fixation-point))
                (Delete (Step Waitfor Fixation-present))
                (Add (Step Waitfor Probe-present))
        ))
```

Sample Rules - 2

```
(Top-make-response
        If
        (
                (Goal Do Visual_search)
                (Step Make Response)
                (Tag ?target target)
                (Tag ?cursor cursor)
                (Motor Manual Modality Free)
        )
        Then
        (
                (Send_to_motor Manual Perform Ply ?cursor ?target Right)
                (Delete (Step Make Response))
                (Add (Step Make Response2))
        ))
```

Modeling Issues - Inputs and Outputs

What you put into an EPIC model for a task:

A simulated device:

Represents system under analysis or design.
Generates display events according to supplied scenarios.
Responds to inputs from simulated human.

A simulated human specified with:

A production-rule representation of a task strategy.
Values for task-specific time parameters.
Choices of response styles not determined by the task.

EPIC supplies the cognitive architecture:

Structure of interconnected processors.
Task-independent process timing and constraints. E.g.:

- Visual resolution constraints.
- Basic perceptual processing times.
- Eye movement times.
- Hand movement times depending on style, distance.

What you get when you run the EPIC model:

- Predicted times and action sequences for all possible scenarios subsumed by the model.
- Generative property: A single rule set generates behavior for a large set of possible specific scenarios.

Modeling Issues - Fixed and Free Parameters

What is fixed

- The connection and mechanisms of processors.
- Most time parameters.
- The feature structures and time parameters of motor processors.

What is free to vary

- Task-specific production rule programming.
  Constrained by the requirement of performing the task correctly and reasonably efficiently.
- Task-specific perceptual encoding types and times.
  Must be constant over similar stimuli.
- The style of movements.
  Only a few styles, but often not determined by the task.

Example of an EPIC Model at Work

Series of screen shots.

Task is ultra-simplified version of Navy radar console task.

Simulated device shows radar-like display, responds to mouse moves, clicks.

Simulated human scans display, selects "blips" to examine, inspects displayed data, takes actions on interface.

Walk through at basic level:

See information coming through visual system into production system.

- Limited by vision model, eye movements.

Production rules are triggered.

- Rules organized into methods and submethods.

Production rules command motor actions.

- Must wait until motor processors can accept commands.

Motor actions carried out on the simulated device.

- Takes time before effects of motor commands appear on display.

Summary

The EPIC architecture represents perceptual and motor constraints in separate processors that limit what the cognitive processor can do.

The cognitive processor is programmed with production-rule procedural knowledge of how to do the task.

Clear division between task-independent and task-specific components.

Illustrate with example run of the model.