Model Based Evaluation of Interactive Systems

Samit Bhattacharya

Dept. of Computer Science & Engineering

IIT Kharagpur

A Survey on UE Methods
(Ivory & Hearst, 2001)

- Total UE methods surveyed = 132
- The systems studied are of two types
  - WIMP
  - Web Based UI
- The surveyed UE methods are classified into five *method classes* each involving a number of *methods*

Survey Contd…

Testing (an evaluator observes users interacting with an interface to determine usability problems)

- Thinking-Aloud Protocol
- Question-Asking Protocol
- Shadowing Method
- Coaching Method
- Teaching Method
- Co-discovery Learning
- Performance Measurement
- Log File Analysis
- Retrospective Testing
- Remote Testing

Survey Contd…

Inspection (an evaluator uses a set of criteria or heuristics to identify potential usability problems in an interface)

- Guideline Review
- Cognitive Walkthrough
- Pluralistic Walkthrough
- Heuristic Evaluation
- Perspective-Based Inspection
- Feature Inspection
- Formal Usability Inspection
- Consistency Inspection
- Standards Inspection

Survey Contd…

Inquiry (users provide feedback on an interface via interviews, surveys, and the like)

- Contextual Inquiry
- Field Observation
- Focus Groups
- Interviews
- Surveys
- Questionnaires
- Self-Reporting Logs
- Screen Snapshots
- User Feedback

Facts and Figures

- Each of the methods consists of three stages
  - Capture: data from user trials
  - Analysis: analysis of captured data
  - Critique: suggestion of possible improvements based on the analysis
- Some statistics from the survey
  - The usage data is taken from the *real* users
  - 67% of the methods were executed manually
  - Automatic data capture in 13% cases
  - Automatic analysis in 18% of the methods
  - In 2% cases, the critique aspect had been automated

Problems

- Involving user may be very costly (e.g. pilots for designing flight simulators)
- May not be possible to involve user at all (e.g. pilots, physically disabled)
- Expert evaluators (not the designers) are needed to identify problems from usage data-a costly affair

Way Out: Model Based Evaluation

The Models Can Help in

- Reducing the role of the user in testing
- Automating the evaluation
  - Repetitive user trails can be dispensed with
  - Role of the expert evaluators can be reduced
- A combination of the two

Survey Revisited

- **Analytical Modeling -** an evaluator employs user and interface models to generate usability predictions

- **Simulation -** an evaluator employs user and interface models to mimic a user interacting with an interface and report the results of this interaction (e.g., simulated activities, errors, and other quantitative measures

Remarks

- Analytical modeling and simulation are engineering approaches to UE
- They enable evaluators to predict usability with user and interface models
- In this presentation, we will focus on such models only

Two Modeling Paradigms

- Modeling User (Computational Cognitive Models)
- Modeling System (Formal/Mathematical Models)

Computational Cognitive Models

- Generic models of the user
- Developed using theories and results from cognitive psychology

Cognitive Psychology Theories in HCI

- Informative/Prescriptive: used to develop design principles and guidelines
- Predictive: used to predict behavior of interactive system.
  - Qualitative
  - Quantitative (Predictive Engineering Models)

Predictive Engineering Models

- Used to predict cognitive phenomena of interaction (e.g. Task completion time, memory load)
  - KLM
  - GOMS and variants (NGOMSL, CPM-GOMS)
  - PUM

Programmable User Model

- Proposed by Yong, Green and Simon (1989)
- Idea was to develop a user model with embodied problem solving skills and common sense knowledge
- Could be “programmed” with task knowledge
- Essentially looks at the role that user knowledge plays in interaction
- Much in the spirit of cognitive architecture without specific architectural components
- No implementation exists

Salient Points

- All these models were developed in the 1980s
- They were operational models based on principles
  - e.g. GOMS was based on the *rationality* principle and the *problem space* principle
- Current practice is to base cognitive models on cognitive architectures

Cognitive Architecture

- “embodiment of a scientific hypothesis about those aspects of human cognition that are relatively constant over time and relatively independent of task”
- Essentially integrative (cognition, perception, motor) theory of human information processing

Cognitive Model = Architecture + Knowledge

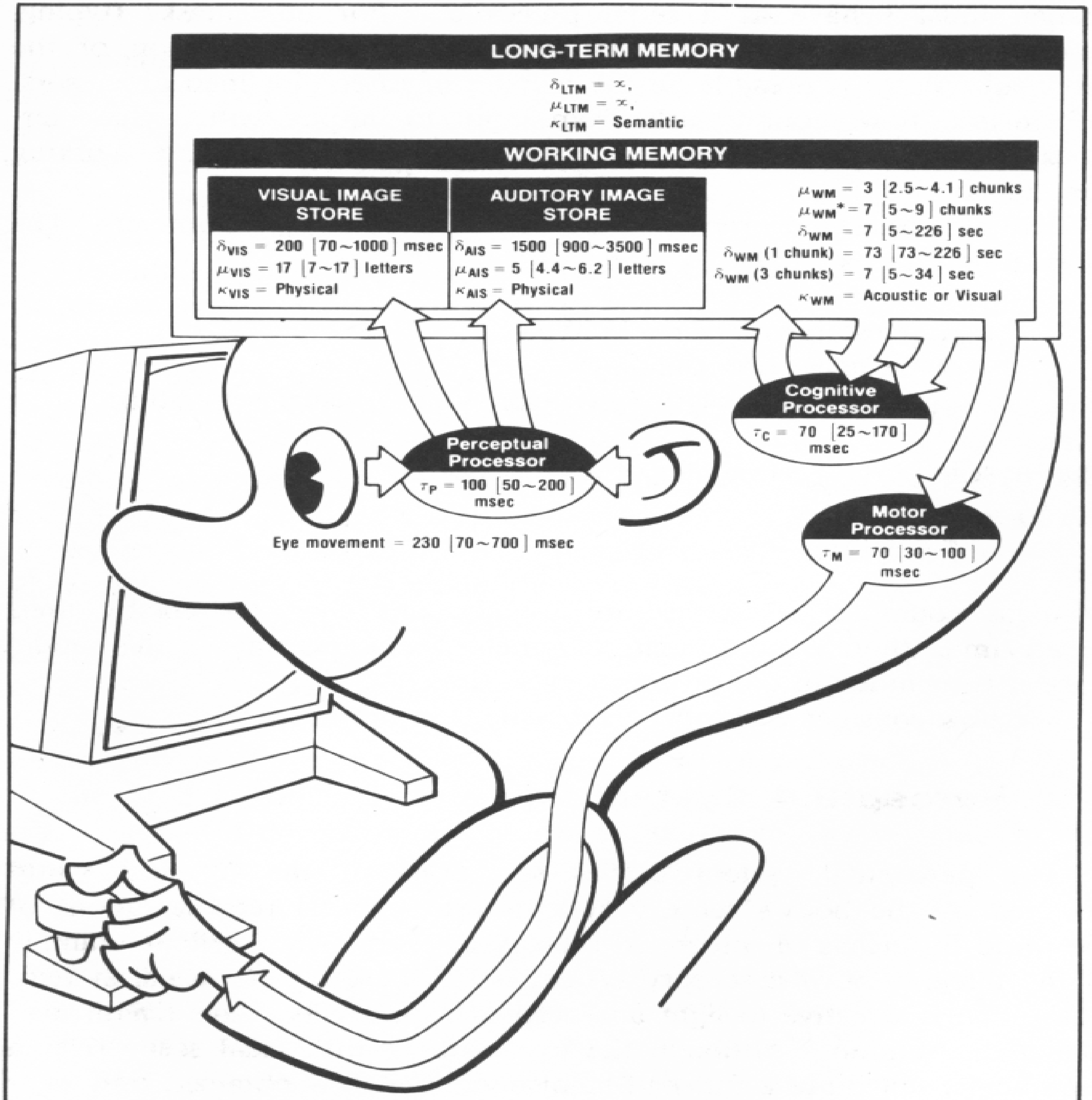
Characteristics of Architectures

- Memory Structure
- Perception (Visual, Auditory)
- Motor action (Occulo-motor, manual)
- Production system
- Interaction mechanism (among architectural components)
- Parameter values
  - Standard - system parameters fixed across all tasks (e.g. production cycle = 50 ms)
  - Typical - free to vary across task situations, but have more or less conventional values. (e.g. the time to recognize that a particular shape = 250 ms)

Some Prominent Architectures in HCI

- MHP (Model Human Processor)
- ACT-R/PM (Adaptive Control of Thought-Rational/PerceptionMotor)
- EPIC (Executive Process - Interactive Control)
- Soar (originally stood for State Operators and Result)
- ICS (Interactive Cognitive Subsystem)

MHP

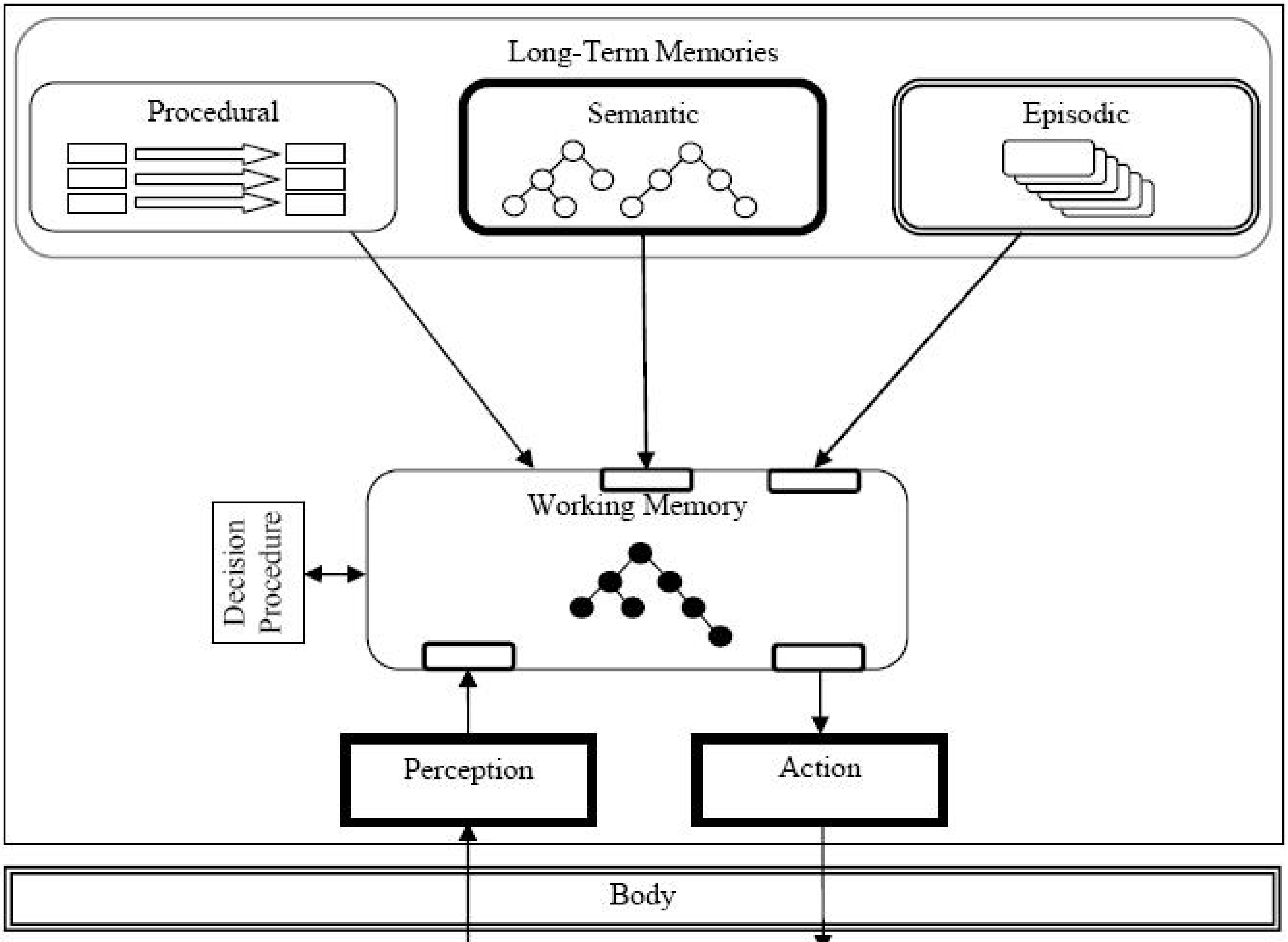


LONG-TERM MEMORY
$\delta_{LTM} = \infty$,
$\mu_{LTM} = \infty$,
κ_{LTM} = Semantic
WORKING MEMORY
VISUAL IMAGE STORE
δ_{VIS} = 200 [70 ~ 1000] msec
μ_{VIS} = 17 [7 ~ 17] letters
κ_{VIS} = Physical
AUDITORY IMAGE STORE
δ_{AIS} = 1500 [900 ~ 3500] msec
μ_{AIS} = 5 [4.4 ~ 6.2] letters
κ_{AIS} = Physical
μ_{WM} = 3 [2.5 ~ 4.1] chunks
μ_{WM}* = 7 [5 ~ 9] chunks
δ_{WM} = 7 [5 ~ 226] sec
δ_{WM} (1 chunk) = 73 [73 ~ 226] sec
δ_{WM} (3 chunks) = 7 [5 ~ 34] sec
κ_{WM} = Acoustic or Visual
Perceptual Processor
τ_P = 100 [50 ~ 200] msec
Cognitive Processor
τ_C = 70 [25 ~ 170] msec
Motor Processor
τ_M = 70 [30 ~ 100] msec
Eye movement = 230 [70 ~ 700] msec

MHP: Salient Points

- Three interacting subsystems
  - Cognitive
  - Perception
  - Motor
- Each with processor and memory
    - described by parameters
      - e.g., capacity, cycle time
- Allows both serial and parallel processing
- No implementation exists-used mainly for pedagogical purposes

Soar



Long-Term Memories
Procedural
Semantic
Episodic
Working Memory
Decision Procedure
Perception
Action
Body

Soar: Salient Points

- Soar assumes behavior is a movement through problem space
- Works based on the following principles:
  - Problem Space Principle: The rational activity in which people engage to solve a problem can be described in terms of (1) a set of states of knowledge, (2) operators for changing one state into another, (3) constraints on applying operators, and (4) control knowledge for deciding which operator to apply next
  - Principle of Rationality*:* People use knowledge to achieve goals

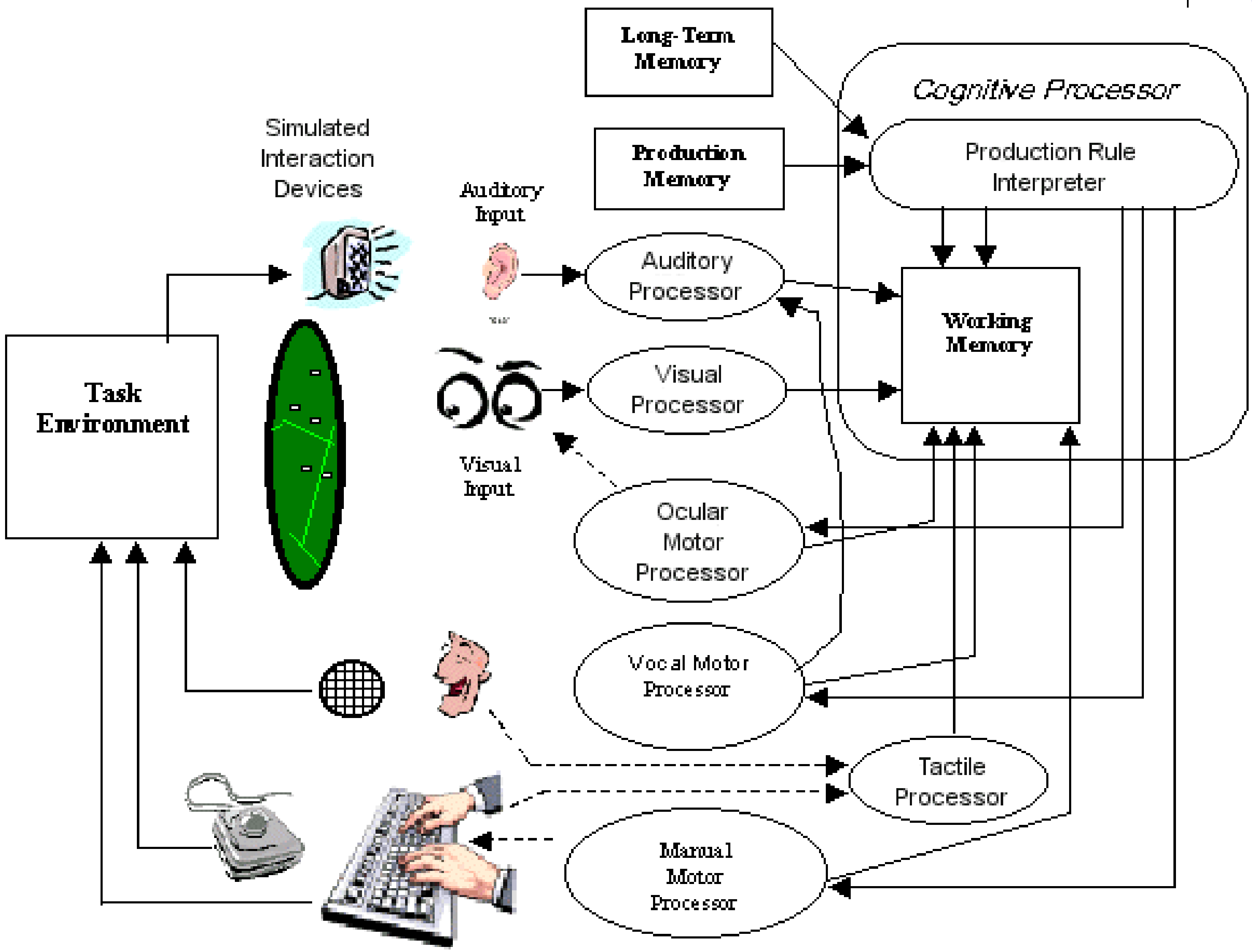
Soar: Salient Points

- Three types of long term memory
  - Procedural: procedural knowledge in the form of production rules
  - Declarative: facts about the world
  - Episodic: specific situations experienced before
- Pervasive learning mechanism
- Memory elements stored and retrieved in the form of *Chunks*
- Perception and Motor modules not well developed

Soar: Salient Points

- Production system (i.e. decision making performed with production rules)
- Decision cycle has five stages
  - Input: Working memory elements are created that reflect changes in perception
  - Elaboration: The contents of WM are matched against the “if” parts of the rules in LTM. All rules that match, fire in parallel, resulting in changes to the features and values of the state in addition to suggestions, or *preferences*, for selecting the current operator. As a result of the working memory changes, more rules may fire. Elaboration continues in parallel waves of rule firings until no more rules fire.
  - Decision: Decide which operator to choose next using preference
  - Application: Apply the operator to produce state transition
  - Output: Send commands to Perception/Motor interfaces

EPIC



Long-Term Memory
Cognitive Processor
Production Memory
Production Rule Interpreter
Simulated Interaction Devices
Auditory Input
Auditory Processor
Working Memory
Task Environment
Visual Processor
Visual Input
Ocular Motor Processor
Vocal Motor Processor
Tactile Processor
Manual Motor Processor

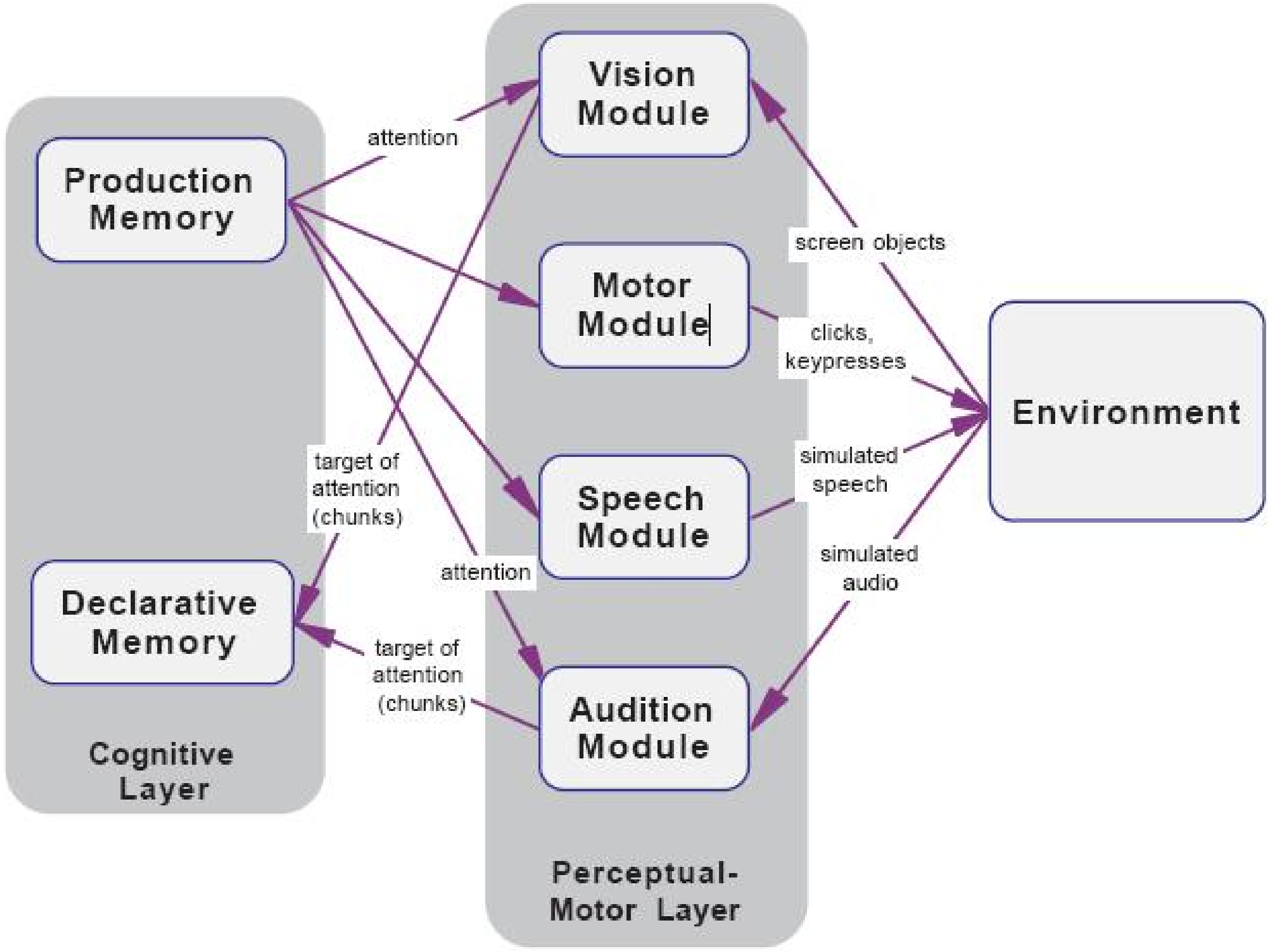
EPIC: Salient Points

- First of its kind to provide a detailed perceptual and motor mechanism
- Basic assumptions
  - Production-rule cognitive processor
  - Parallel perceptual and motor processors
  - Multiple production rules can fire in a production cycle
- No learning mechanism

EPIC: Salient Points

- Fixed architectural properties
  - Components, pathways, and most time parameters
- Task-dependent properties
  - Cognitive processor production rules
  - Perceptual recoding
  - Response requirements and styles
- Useful to model Motor-Perception intensive HCI tasks

ACT-R/PM


Production Memory
Declarative Memory
Cognitive Layer
Vision Module
Motor Module
Speech Module
Audition Module
Perceptual-Motor Layer
Environment
attention
target of attention (chunks)
attention
target of attention (chunks)
screen objects
clicks, keypresses
simulated speech
simulated audio

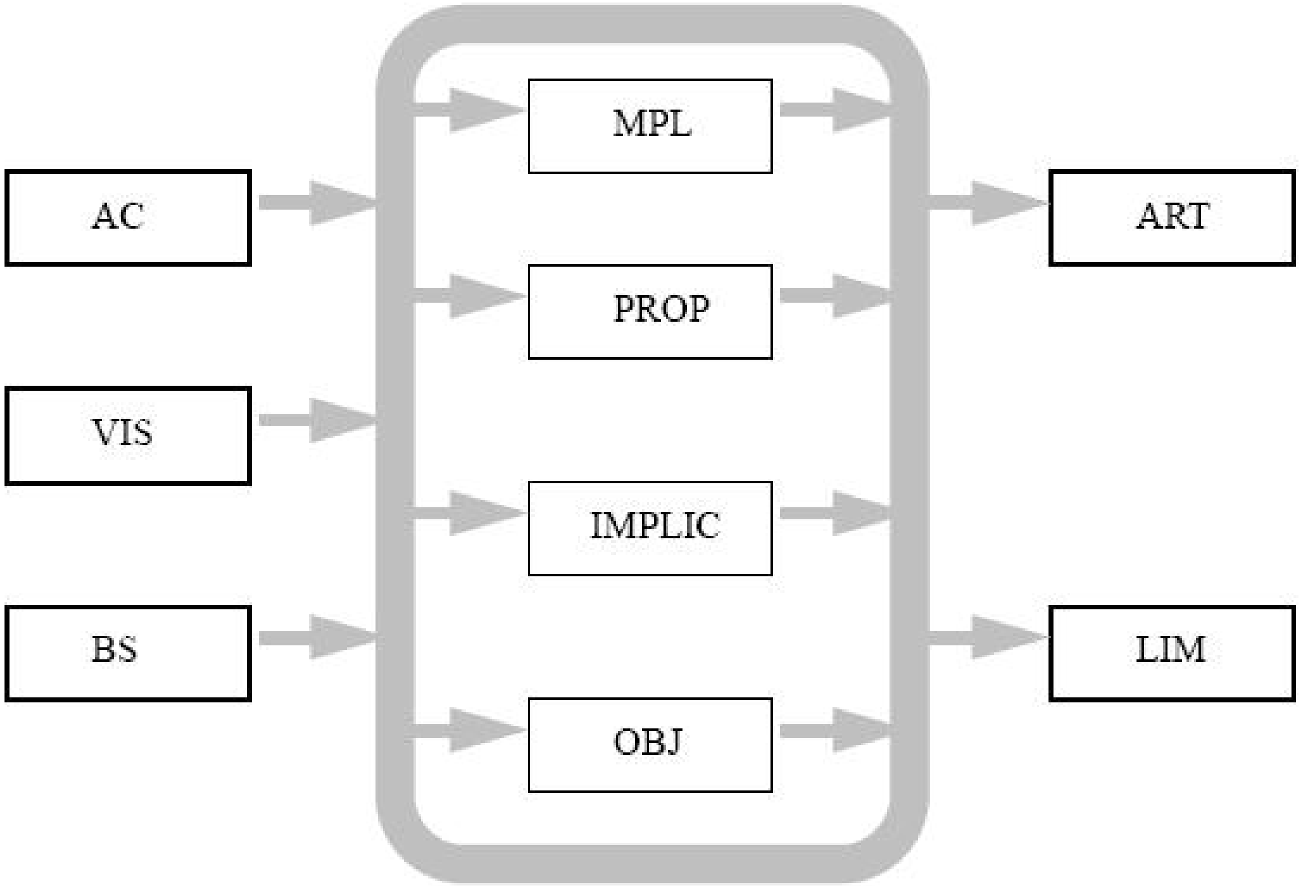
ACT-R/PM: Salient Points

- A fusion of ACT-R and EPIC
- ACT-R is the cognitive layer, production based
- Unlike Soar and EPIC, can fire only one production at a time
- If multiple production rules matches, an arbitration procedure called *conflict resolution* is used to determine the rule to fire
- LTM = procedural + declarative

ACT-R/PM: Salient Points

- Information stored to and retrieved from the memory in *Chunks*
- Each declarative memory elements has an *activation* value
- Each production rule in the procedural memory has an *utility* value
- These values are used for conflict resolution
- Learning mechanism present but not pervasive like Soar

ICS



ICS: Salient Points

- Represents cognitive activity as a configuration, or flow of information through different mental representations
- Basic operation- transformation of a mental representation from one form, in which it describes a particular class of information about the world, into another form, in which it describes a different class of information

ICS: Stages of Basic Operation

- **Storage** - Store the incoming representation in a local form of LTM called the *image record*
- **Transformation** - Access *proceduralized* knowledge acquired from previous experiences with the incoming information to try to produce the appropriate output representation
  - Procedural knowledge is embedded within the transformation processes
  - This embedded knowledge allows to produce an output for a particular input without accessing the image record
- **Revival** - If procedural knowledge is poorly developed, ICS may try to access previous experiences that are similar to the incoming representation from the image record, to elaborate and refine it

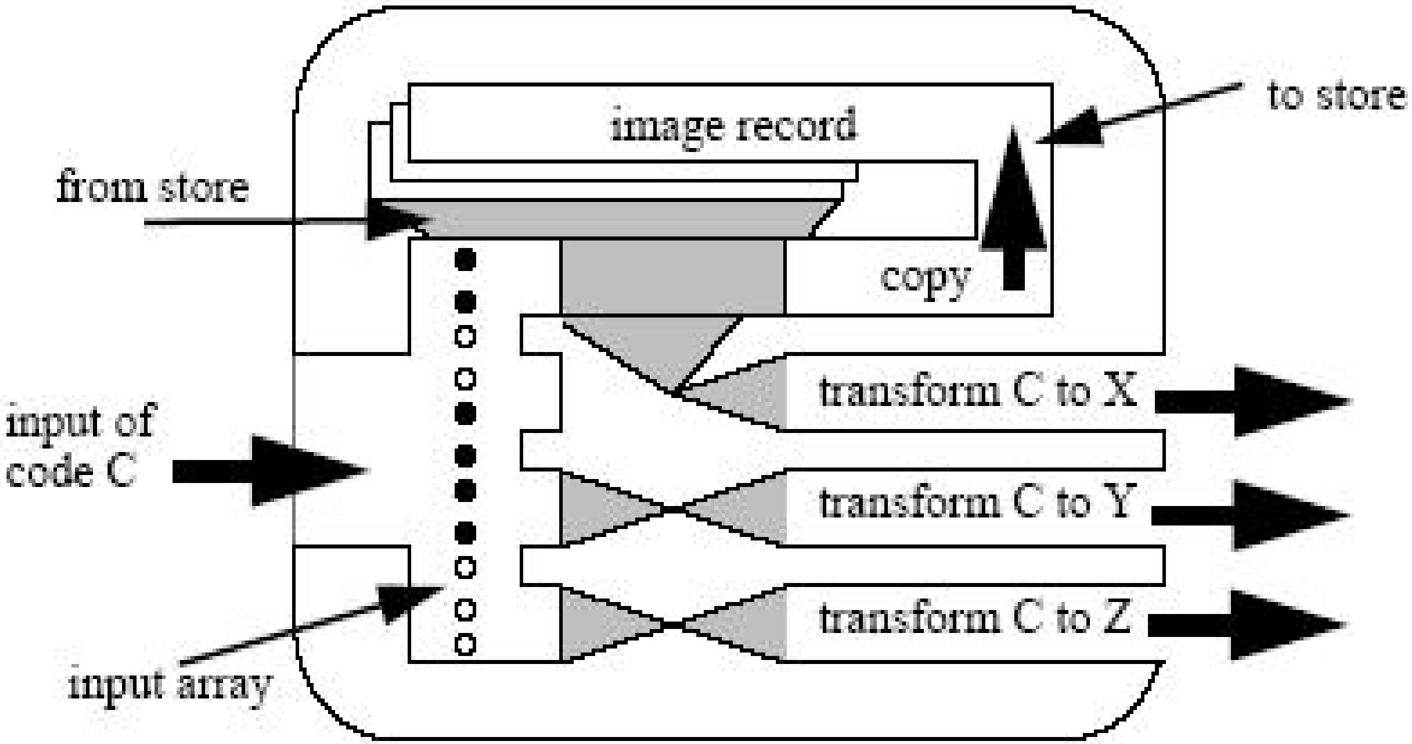
ICS: Salient Points

- Stored information is said to be “revived by memory” rather than “retrieved from memory” because memory is seen as an active entity rather than a passive store
- if the output representation that is produced is used successfully in task performance, the procedural knowledge relating the incoming and output representations is updated or strengthened, as appropriate

ICS: Salient Points

- The three basic operations—storage, transformation, and revival of information by memory—are grouped together within a "subsystem" dedicated to the processing of one particular class of information
- Nine different sub systems with same structure

ICS Sub System



ICS: Subsystems

- Sensory subsystems
  - **VIS** visual: encodes dimensions of light such as wavelength, brightness, user visual space
  - **AC** acoustic: encodes dimensions of sound such as frequency, timbre, intensity
  - **BS** body-state: encodes dimensions such as skeletal muscle tension
- Structural subsystems
  - **OBJ** object: abstract structural description of entities and relations in visual space
  - **MPL** morpholexical: abstract structural description of entities and relations in sound space

ICS: Subsystems

- Meaning subsystems
  - **PROP** propositional: abstract description of entities and relations in semantic space
  - **IMPLIC** implicational: abstract description of human existential space abstracted over sensory and propositional input
- Effector subsystems
  - **ART** articulatory: encodes dimensions such as force, target and timing of articulatory musculatures
  - **LIM** limb: encodes dimensions such as force, target, position and timing of skeletal musculatures

Use of Cognitive Models in HCI

- Explanation of (observed) interactive behavior
- Evaluation of interactive systems
  - Cognitive Task Analysis (CTA)
    - Estimate human performance metrics from a task description
    - Compare interfaces with estimated values
  - Simulated Human User (SHU)/Synthetic User in simulated interaction environment

Simulated Human User: Benefits

- Can replace actual user, thus reducing the cost of prototype development and user trials
- Can test a number of tasks with the interface
- Allows the designer to test design alternatives/changes under many different conditions with operators of varying skill levels and other cognitive differences

Formal Methods in HCI

- Grammar Based (Linguistic) Formalism in the ’80s
  - BNF
  - Task-Action Grammar (TAG)
- Linguistic Formalisms were developed mainly for Command Line Interface
- Advent of Direct Manipulation Interfaces Inspired New Approaches

State Transition Networks

- Used to describe dialog (Interaction)
- A number of variants were used
  - Basic STN
  - Generalized STN (GTN)
  - Augmented STN (ATN)
- Not good at describing concurrent interaction (common in direct manipulation)
- Alternate formalisms were used
  - Petri Nets
  - State charts

Other Methods

- Process Algebra
- Temporal Logic
- Markov Models
- Matrix Algebra

Remarks

- All the early works were aimed to formally express relevant concepts (e.g. devices, dialogs)
- Used to verify properties of interactive systems manually
- Subsequent efforts were aimed at automatic verification

Automatic Verification

Deals with automatic verification of properties of interactive systems

- Completeness: All possible transitions between all the states of interactions are taken care of.
- Determinism: Same action cuases same state change always
- Consistency: Action performed in different contexts does the same thing (difficult to verify formally)
- Reachability: A user can get to a *useful* state from the current state easily.
- Reversibility (*undo*): A special case of reachability

Automatic Verification Contd…

- Requires formal specifications of system and interaction
- The choices to the designers are
  - Design custom specification language and suitable tools for verification
  - Use existing specification language and use available verification tools
  - Design custom specification language and *mapper* to existing specification languages
- Each of the above choices have been tried

Automatic Verification Contd…

- Example of custom made specification language and tools
  - STN notation and HyperDoc
  - Matrix Algebra and MAUI
- The second and third choice led to the use of model checkers for automatic verification

Use of Model Checkers

- Example of existing specification language and model checker
  - LOTOS → CADP
- Example of custom made specification language and mapper: VEG→Promela. The model checker used in this case is SPIN

Comparison

- Cognitive Models
  - Advantage: good at evaluating interfaces w.r.t. human factors (task completion time, memory load)
  - Disadvantage: Ignore system side of interaction (completeness, determinism, reachability, reversibility)
- Formal Models (just the reverse)

Hybrid Models

- Efforts were made to combine the advantages of the two
- Such models are known as *Syndetic* Models
- These models were developed using *Interactors* - a formal architectural abstraction of interactive systems
- All the models used ICS
- Early models used Modal Action Logic as description language
- Later, LOTOS was used

Thank You

How usability is defined: Items in four different usability constructs

Shackel (1986)	Nielsen (1993)	ISO 9241-11(1998)	ISO 9126-1 (2000)
•**Effectiveness:** performance in accomplishment of tasks and the access to potential utility. •**Learnability:** degree of learning needed to accomplish tasks. •**Flexibility:** adaptation to variation in tasks. •**Attitude:** user satisfaction with system.	•**Learnability** •**Memorability** •**Efficiency** •**Subjective satisfaction** •**Low rate of errors**	•**Effectiveness** •**Efficiency** •**Satisfaction** *"The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use."*	•**Understandability** •**Learnability** •**Operability** •**Attractiveness** *"The capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions."*

WIMP Interfaces

- Stands for Windows, Icons, Menus and Pointing Devices
- Originally developed in the ’60s. Dominant Since ’80s.
- WIMP interfaces may not be Direct Manipulation interfaces
  - DM interfaces allow the user to manipulate objects of interest or perform actions on them 'directly‘. However, a WIMP interface may just offer direct access to controls (buttons/menus) but not necessarily manipulate the objects of interest directly. For example, one may click a button to put an item in a shopping cart rather than dragging the item to the cart.