UNIVERSITY OF BIRMINGHAM

School of Computer Science

EVALUATION METHODS AND STATISTICS

Prof. Chris Baber
Chair of Pervasive and Ubiquitous Computing

Human Modelling & Simulation

- Model: human behaviour can be represented by heuristics and algorithms

- Simulation: an application, in software (and hardware) that can be used to run a model

Pew and Mavor, 1998, *Modeling Human and Organizational Behavior*

UNIVERSITY OF BIRMINGHAM

Why Model?

- To research human cognitive processes
  - You can use models to predict / explain human behaviour and test them to destruction
- To evaluate Product Designs
  - You can compare 'performance' on alternative designs prior to building
- To support adaptation
  - You can anticipate the information needs of the user / learner through modelling their activity
- Computer Generated Forces
  - (Non-player) Characters in Computer Games need to have 'normal' and 'different' behaviour

Models simplify reality

- “Complexity is really just reality without the simplifying assumptions that we make in order to understand it.”

- Models simplify human performance in order to produce quantitative or qualitative descriptions of those aspects deemed relevant to the modellers aims

Allen, P.M. Et al., 2005, The implications of complexity for business process and strategy, in K.A. Richardson (ed) *Managing Organizational Complexity,* p.397

UNIVERSITY OF BIRMINGHAM

Performance vs. Competence

- Performance Models
  - Make statements and predictions about the time, effort or likelihood of error when performing specific tasks;
- Competence Models
  - Make statements about what a given user knows and how this knowledge might be organised.

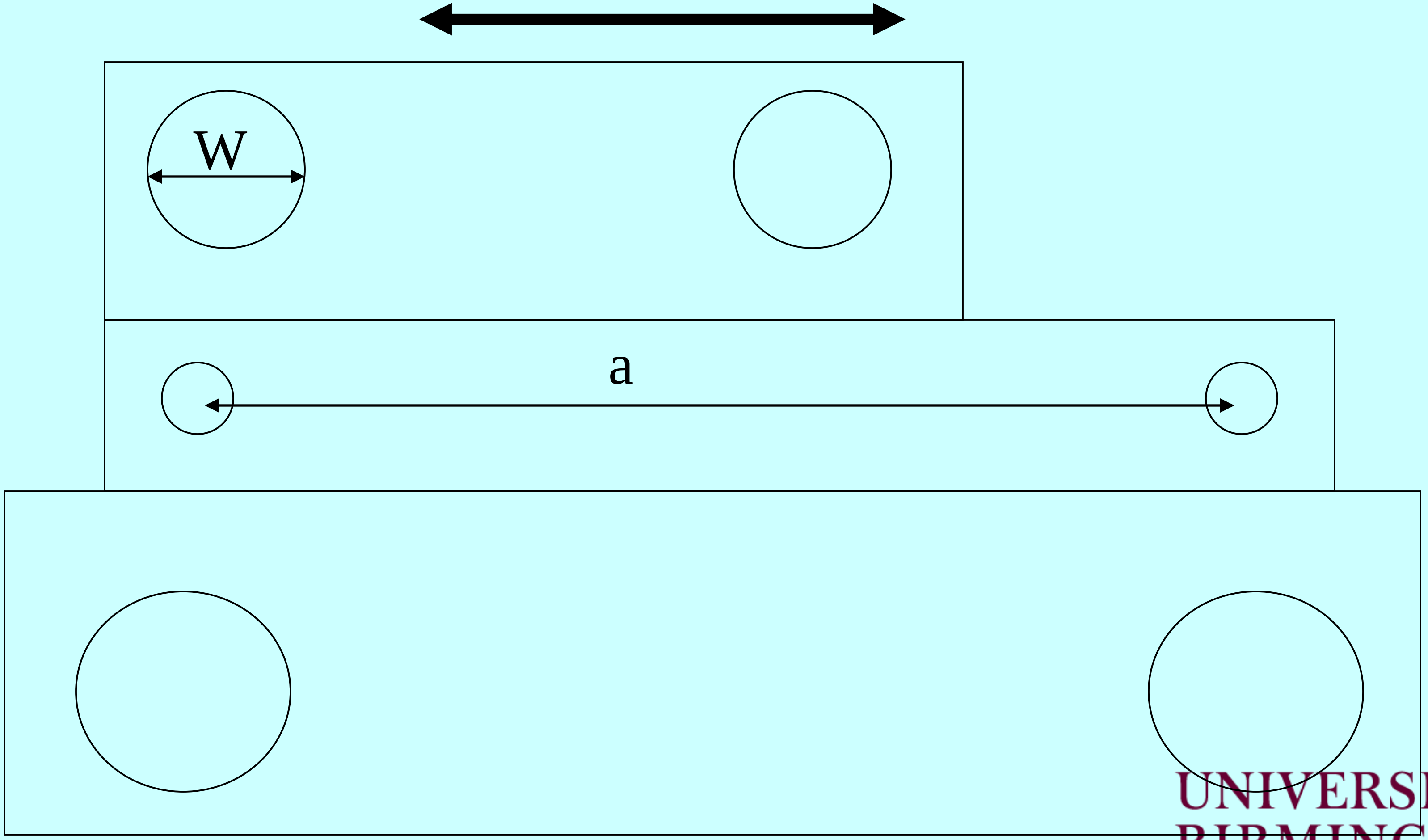
Sequence vs. Process vs. Grammar

- Sequence Models
  - Define activity simply in terms of sequences of operations that can be quantified
- Process Models
  - Simple model of mental activity but define the steps needed to perform tasks
- Grammatical Models
  - Model required knowledge in terms of ‘sentences’

Fitts' Law

- Paul Fitts 1954
- Information-theoretic account of simple movements
- Define the number of 'bits' processed in performing a given task

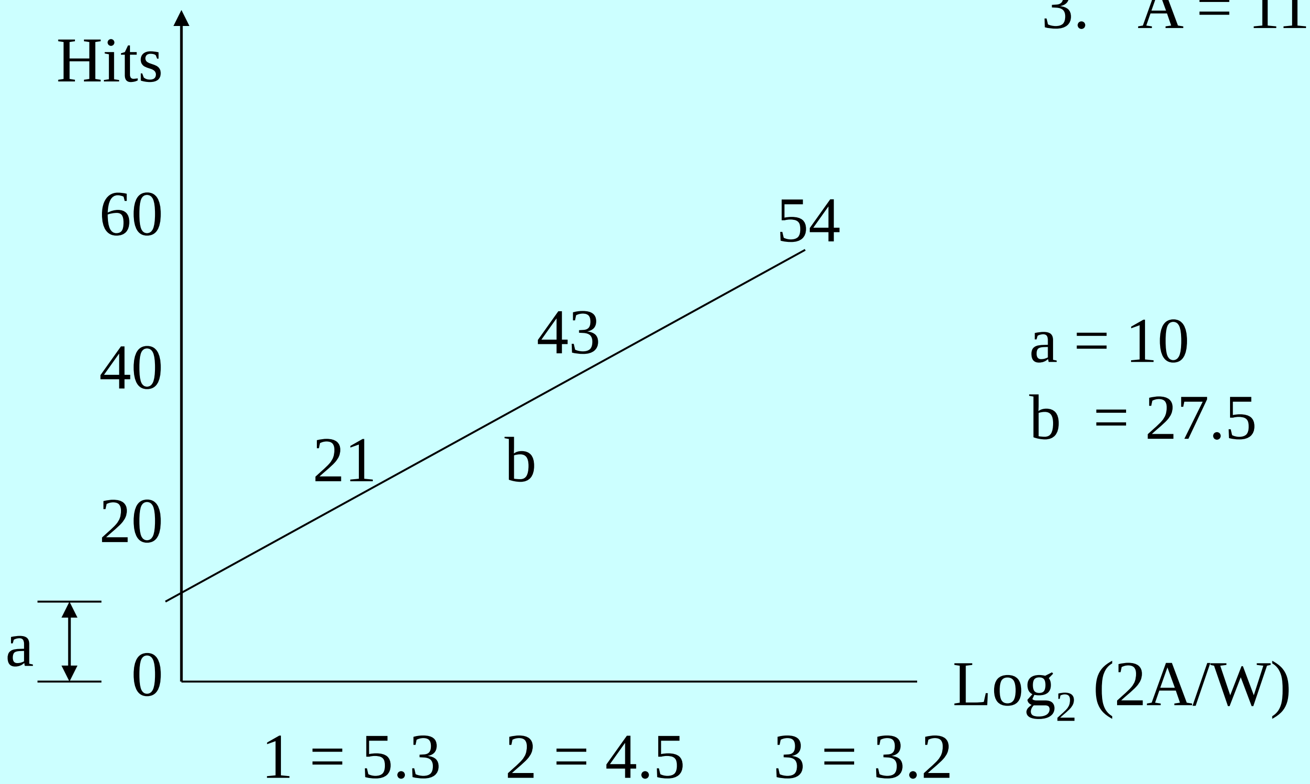
Fitts' Tapping Task



UNIVERSITY OF BIRMINGHAM

Fitts' Law

1. A = 62, W = 15
2. A = 112, W = 7
3. A = 112, W = 21



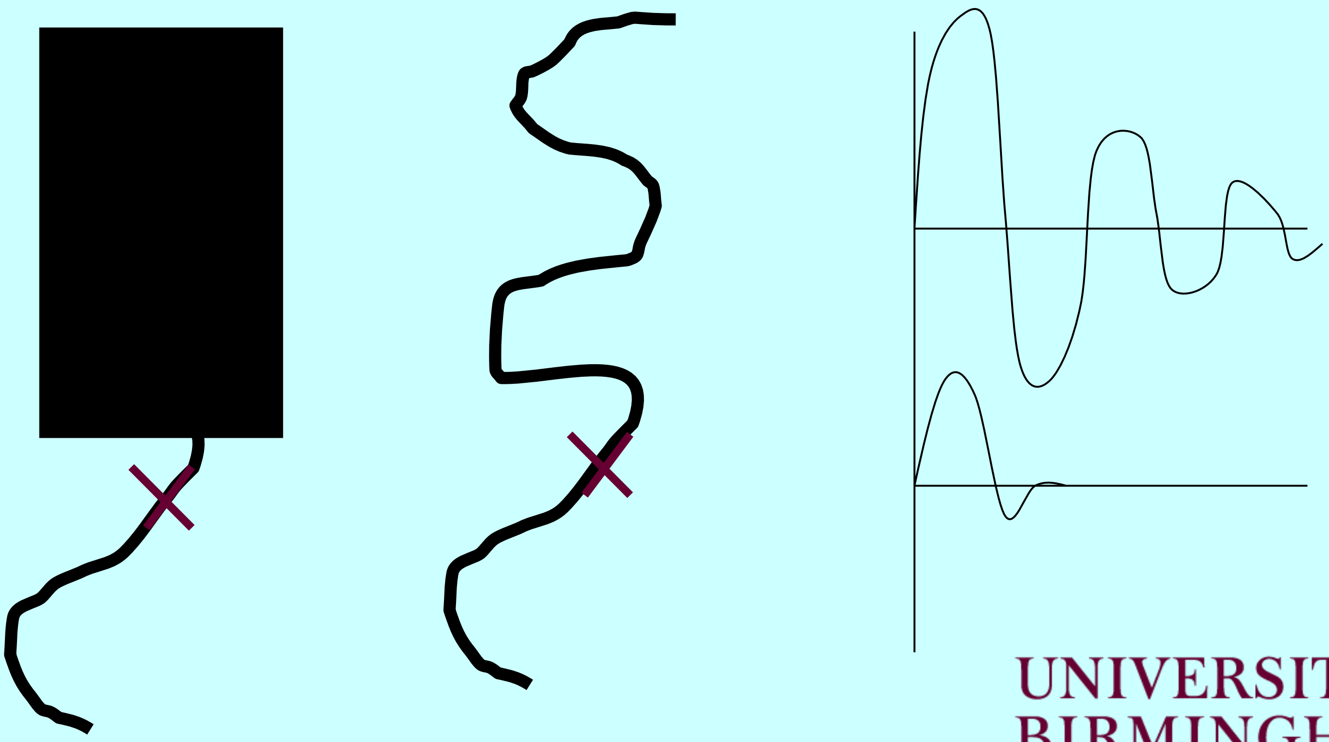
a = 10
b = 27.5

Movement Time = a + b ($\log_2$ 2A/W)

UNIVERSITY OF BIRMINGHAM

Tracking

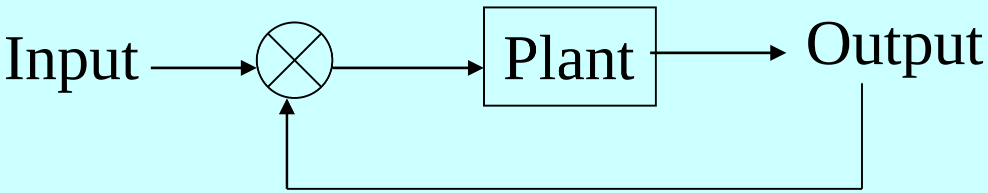
- Control movement of X to keep it on the path



Simple Tracking Activity

- Compensatory tracking
  - Closed loop
  - Sample output and compare with input
  - Correct difference (error)

- Pursuit tracking
  - Closed loop
    - Open loop
    - Focus on input
    - Assume dynamics known and anticipated



UNIVERSITY OF BIRMINGHAM

Anticipation

- Anticipation requires some ‘model’ of the system being controlled
  - Understanding system dynamics (knowledge)
  - Tuning of performance (practice)
  - Information from the world (sampling)

Keystroke Level Models

- Related to Time and Motion studies
- Human information processor as linear executor of specified tasks
- Unit-tasks have defined times
- Prediction = summing of times for sequence of unit-tasks

Example: cut and paste

Task Model: Select line – Cut – Select insertion point – paste

Task One: select line

move cursor to

start of line

press (hold) button

drag cursor to

end of line

release button

Times for Movement

- H: homing, e.g., hand from keyboard to mouse
  - Range: 214ms – 400ms
  - Average: 320ms
- P: pointing, e.g., move cursor using mouse
  - Range: defined by Fitts' Law
  - Average: 1100ms
- B: button pressing, e.g., hitting key on keyboard
  - Range: 80ms – 700ms
  - Average: 200ms

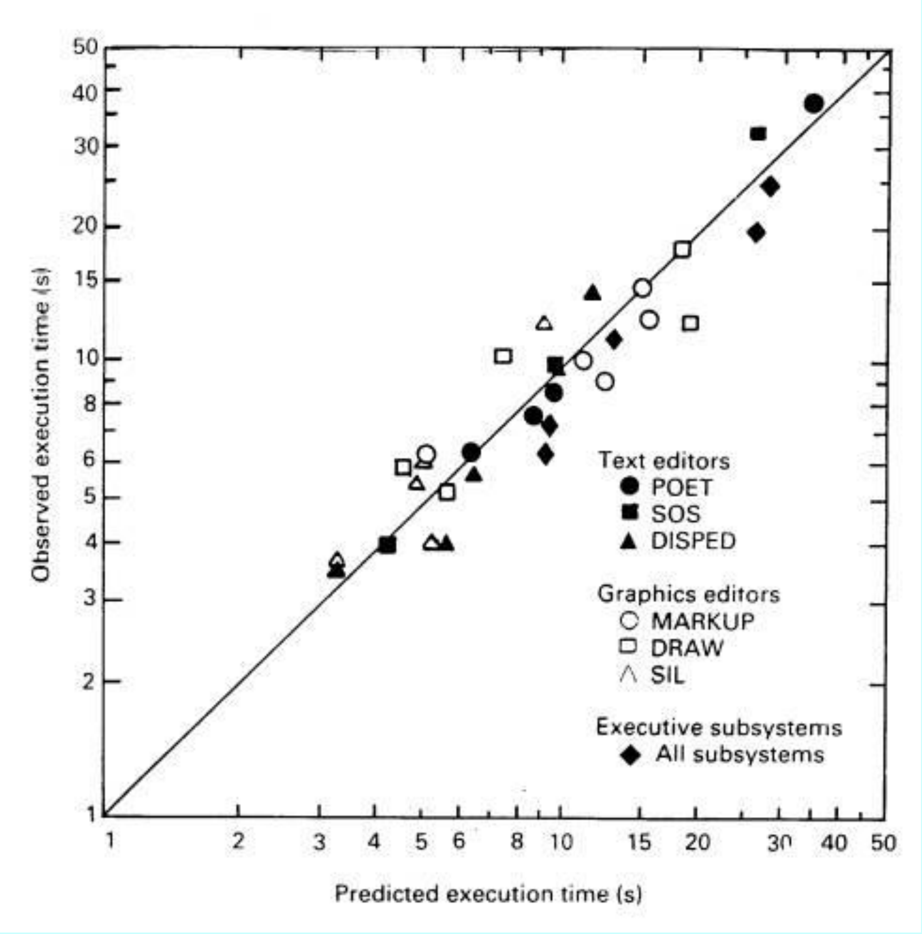
Times for Cognition / Perception

- M: mental operation
  - Range: 990ms – 1760ms
  - Average: 1350ms
- A: switch attention between parts of display
  - Average: 320ms
- R: recognition of items
  - Range: 314ms – 1800ms
  - Average: 340ms
- Perceive change:
  - Range: 50 – 300ms
  - Average: 100ms

UNIVERSITY OF BIRMINGHAM

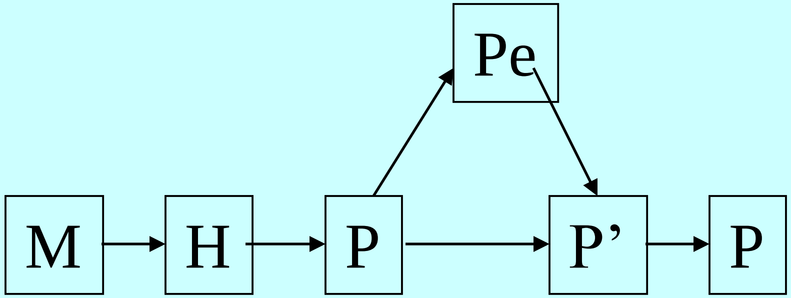
KLM Validity

Predicted values lie within 20% of observed values for Well-defined tasks



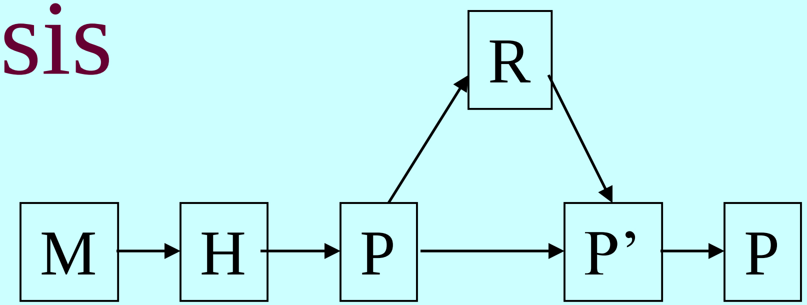
Handling Parallel Activity

- What if we use ‘accelerated scrolling’ on the cursor keys?
  - Press ↓ key and read scrolling numbers
  - Release key at or near number
  - Select correct number

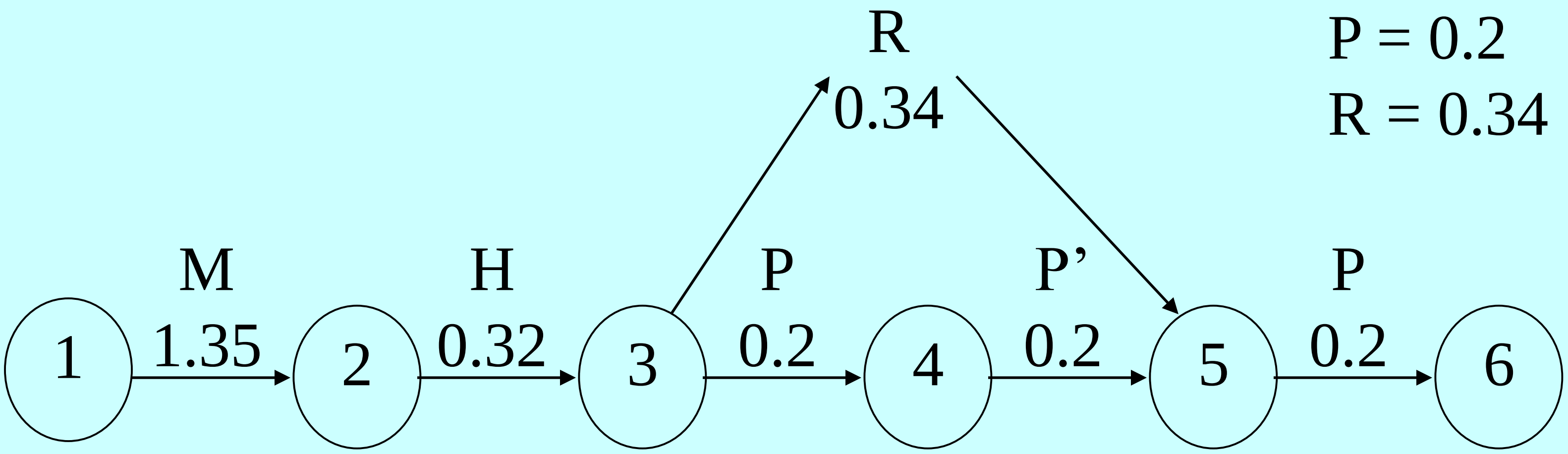


UNIVERSITY OF BIRMINGHAM

Critical Path Analysis



M = 1.35
H = 0.32
P = 0.2
R = 0.34



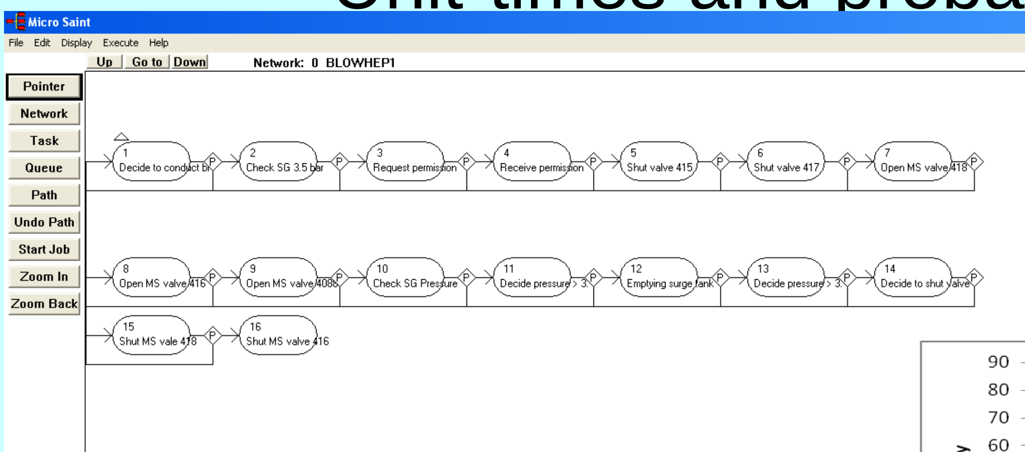
UNIVERSITY OF BIRMINGHAM

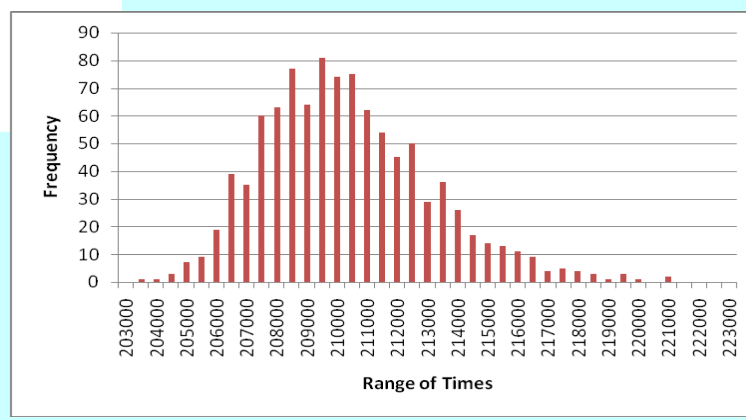
Critical Path Table

Activity	Duration	EST	LST	EFT	LFT	Float
M	1.35	0	0	1.35	1.35	0
H	0.32	1.35	1.35	1.67	1.67	0
P	0.2	1.67	1.67	1.87	1.87	0
R	0.34	1.67	1.73	2.01	2.07	0.06
P'	0.2	2.07	2.07	2.27	2.27	0
P	0.2	2.27	2.27	2.47	2.47	0

Monte Carlo Simulations

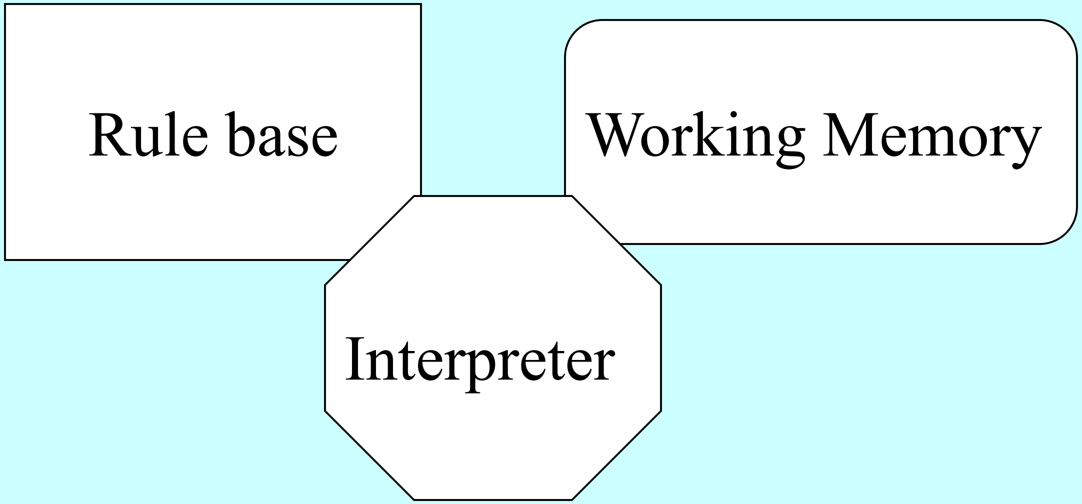
- Task-network models
  - MicroSAINT
  - Unit-times and probability of transition





Production Systems

Architecture of a production system:



Production Systems

- *If* Condition *Then* Action
- Condition
  - Event – external (to model, e.g., light on)
  - State – internal (to model, e.g., selection)
- Action – operation associated with condition

The Problem of Control

- Rules are useless without a useful way to apply them
- Need a consistent, reliable, useful way to control the way rules are applied
- Need to move task forward
- Different architectures / systems use different control strategies to produce different results

The Parsimonious Production Systems Rule Notation

- On any cycle, any rule whose conditions are currently satisfied will fire
- Rules must be written so that a single rule will not fire repeatedly
- Only one rule will fire on a cycle
- All procedural knowledge is explicit in these rules rather than being explicit in the interpreter

States, Operators, And Reasoning (SOAR)

http://www.isi.edu/soar/soar.html

UNIVERSITY OF BIRMINGHAM

States, Operators, And Reasoning (SOAR)

- Sequel of General Problem Solver (Newell and Simon, 1960)
- SOAR seeks to apply <u>operators</u> to <u>states</u> within a <u>problem space</u> to achieve a <u>goal</u>.
- SOAR assumes that actor uses all available knowledge in problem-solving

Soar as a Unified Theory of Cognition

- Intelligence = problem solving + learning
- Cognition seen as search in problem spaces
- All knowledge is encoded as productions
  - $\Rightarrow$ a single type of knowledge
- All learning is done by chunking
  - $\Rightarrow$ a single type of learning

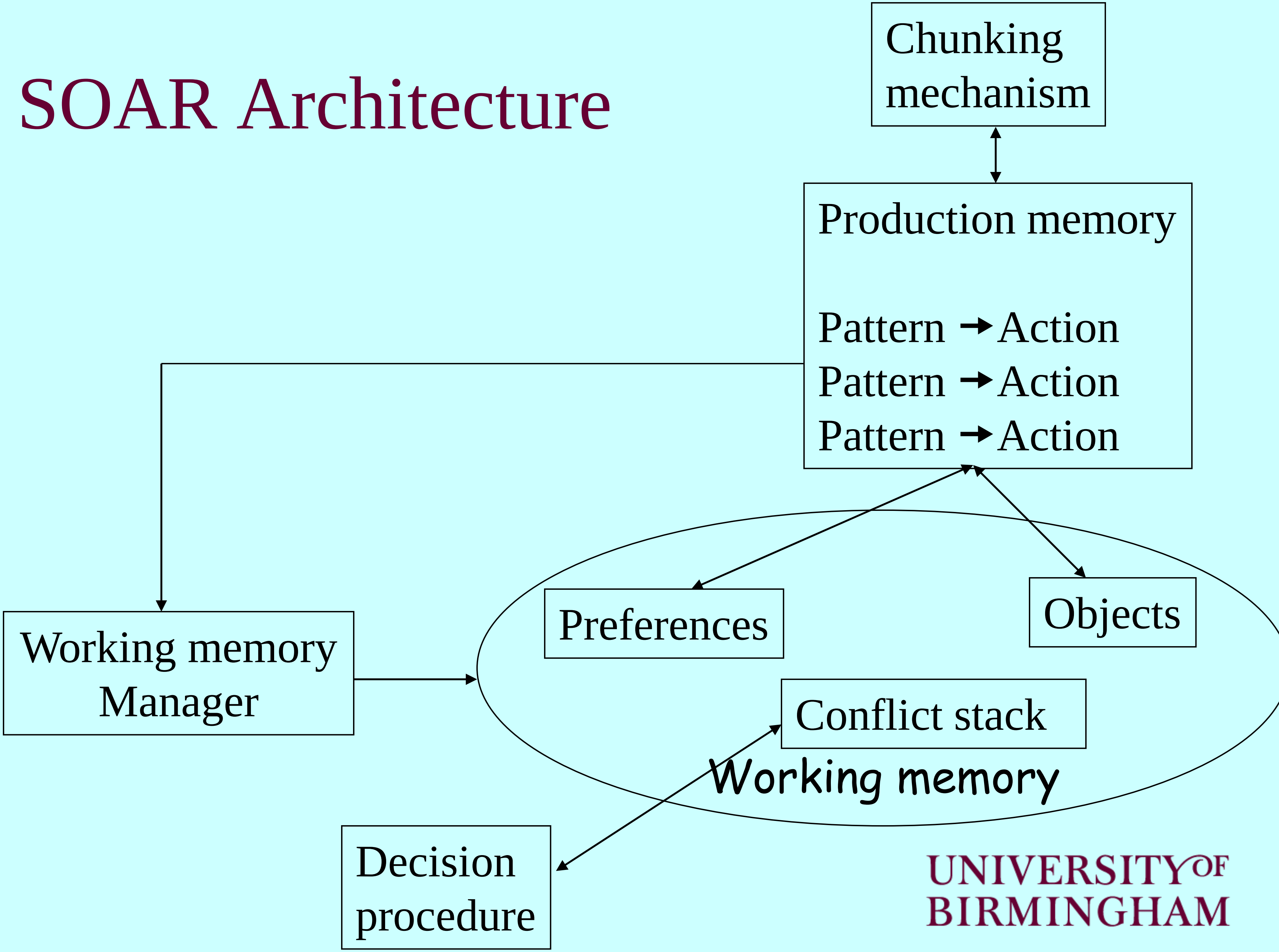
SOAR Activity

- **Operators:** Transform a state via some action
- **State:** A representation of possible stages of progress in the problem
- **Problem space:** States and operators that can be used to achieve a goal.
- **Goal:** Some desired situation.

SOAR Activity

- Problem solving = applying an Operator to a State in order to move through a Problem Space to reach a Goal.
- Impasse = Where an Operator cannot be applied to a State, and so it is not possible to move forward in the Problem Space. This becomes a new problem to be solved.
- Soar can *learn* by storing solutions to past problems as *chunks* and applying them when it encounters the same problem again

SOAR Architecture



UNIVERSITY OF BIRMINGHAM

Explanation

- Working Memory
  - Data for current activity, organized into objects
- Production Memory
  - Contains production rules
- Chunking mechanism
  - Collapses successful sequences of operators into chunks for re-use

3 levels in soar

- Symbolic – the programming level
  - Rules programmed into Soar that match circumstances and perform specific actions
- Problem space – states & goals
  - The set of goals, states, operators, and context.
- Knowledge – embodied in the rules
  - The knowledge of how to act on the problem/world, how to choose between different operators, and any learned chunks from previous problem solving

How does it work?

- A problem is encoded as a current state and a desired state (goal)
- Operators are applied to move from one state to another
- There is success if the desired state matches the current state
- Operators are proposed by productions, with preferences biasing choices in specific circumstances
- Productions fire in parallel

UNIVERSITY OF BIRMINGHAM

Impasses

- If no operator is proposed, or if there is a tie between operators, or if Soar does not know what to do with an operator, there is an impasse
- When there are impasses, Soar sets a new goal (resolve the impasse) and creates a new state
- Impasses may be stacked
- When one impasse is solved, Soar pops up to the previous goal

Learning

- Learning occurs by chunking the conditions and the actions of the impasses that have been resolved
- Chunks can immediately used in further problem-solving behaviour

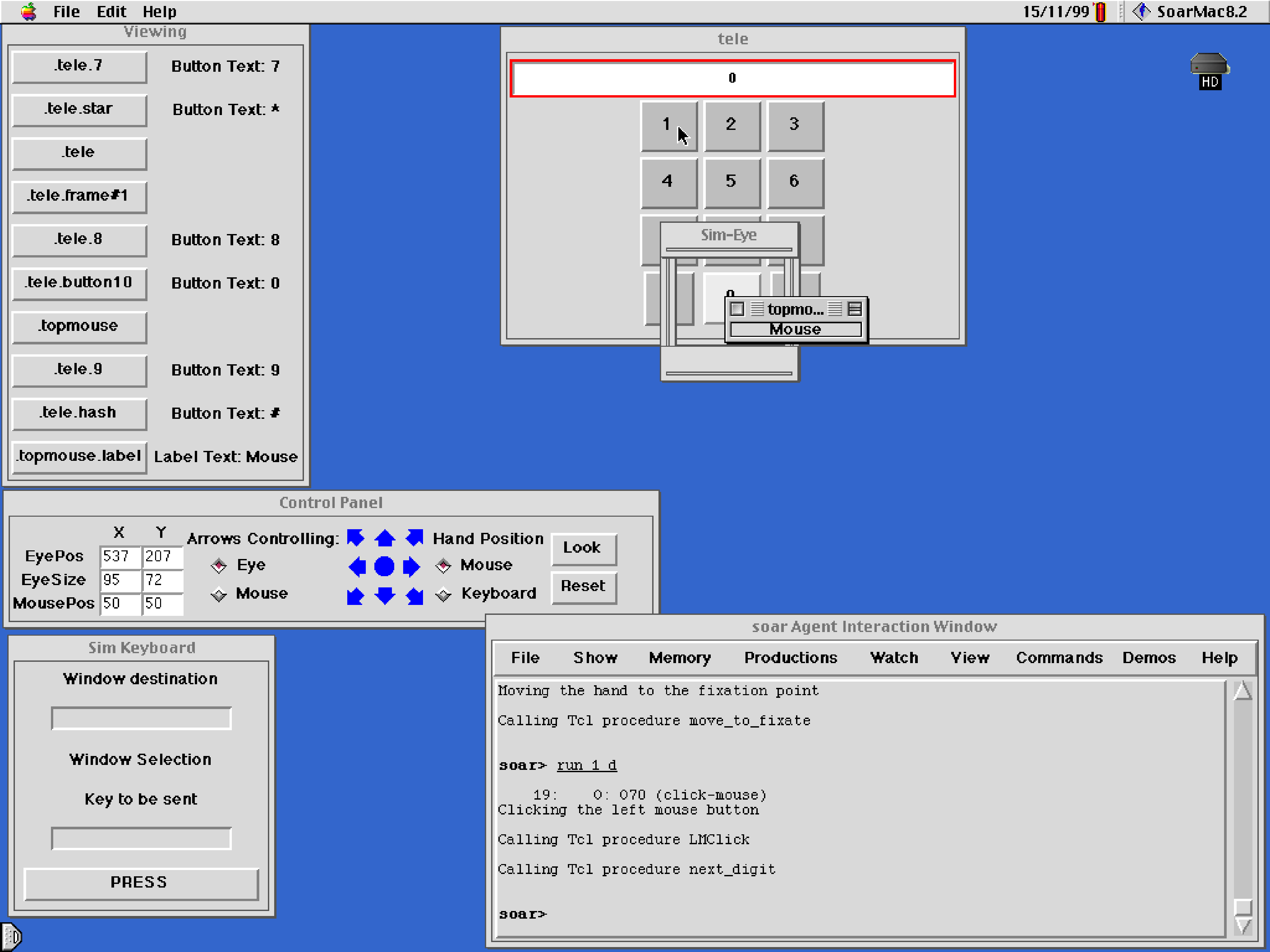
Conclusions

- It may be too "unified"
  - Single learning mechanism
  - Single knowledge representation
  - Uniform problem state
- It does not take neuropsychological evidence into account (cf. ACT-R)
- There may be non-symbolic intelligence, e.g. neural nets etc not abstractable to the symbolic level

Soar/Tcl-PM

- An interactive cognitive model written in **Soar**
- A virtual eye and hand running under **Tcl/Tk**

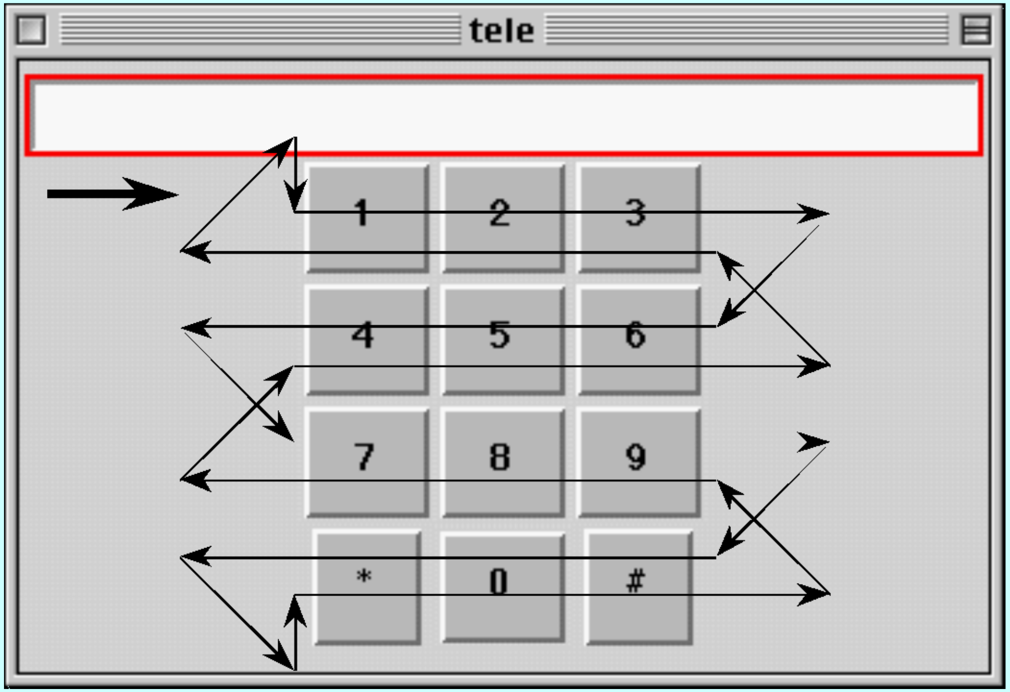
- Together: a **perceptual-motor** system allowing:
  - Perception of interfaces implemented in Tcl/Tk
  - Motor actions (pointing and clicking) on same interfaces


File Edit Help
15/11/99
SoarMac8.2
Viewing
.tele.7
Button Text: 7
.tele.star
Button Text: *
.tele
.tele.frame#1
.tele.8
Button Text: 8
.tele.button10
Button Text: 0
.topmouse
.tele.9
Button Text: 9
.tele.hash
Button Text: #
.topmouse.label
Label Text: Mouse
tele
0
1
2
3
4
5
6
Sim-Eye
topmo...
Mouse
HD
Control Panel
X
Y
EyePos
537
207
EyeSize
95
72
MousePos
50
50
Arrows Controlling:
Eye
Mouse
Hand Position
Mouse
Keyboard
Look
Reset
Sim Keyboard
Window destination
Window Selection
Key to be sent
PRESS
soar Agent Interaction Window
File
Show
Memory
Productions
Watch
View
Commands
Demos
Help
Moving the hand to the fixation point
Calling Tcl procedure move_to_fixate
soar> run 1 d
19: O: O70 (click-mouse)
Clicking the left mouse button
Calling Tcl procedure LMClick
Calling Tcl procedure next_digit
soar>

Simple model to use Soar/Tcl-PM

- Search for button
  - Include memory (or not) of found objects
- Press button
  - Move
  - Click
- Get next digit

Default Scanning Behaviour



Time to Dial

Eye Size	Time to complete dialling (s) Memory on	Memory off
20 x 20	15.4	30.2
50 x 50	15.4	31.5
100 x 100	16.2	32.3
150 x 150	16.2	22.3
200 x 200	16.2	24.3

UNIVERSITY OF BIRMINGHAM

Testing this Model

- KLM model
  - 23.1 seconds
- NGOMSL model for 11 digits
  - 28.6 seconds
- Informal data
  - About 11 seconds

Executive Process Interactive Control (EPIC)

ftp://ftp.eecs.umich.edu/people/kieras

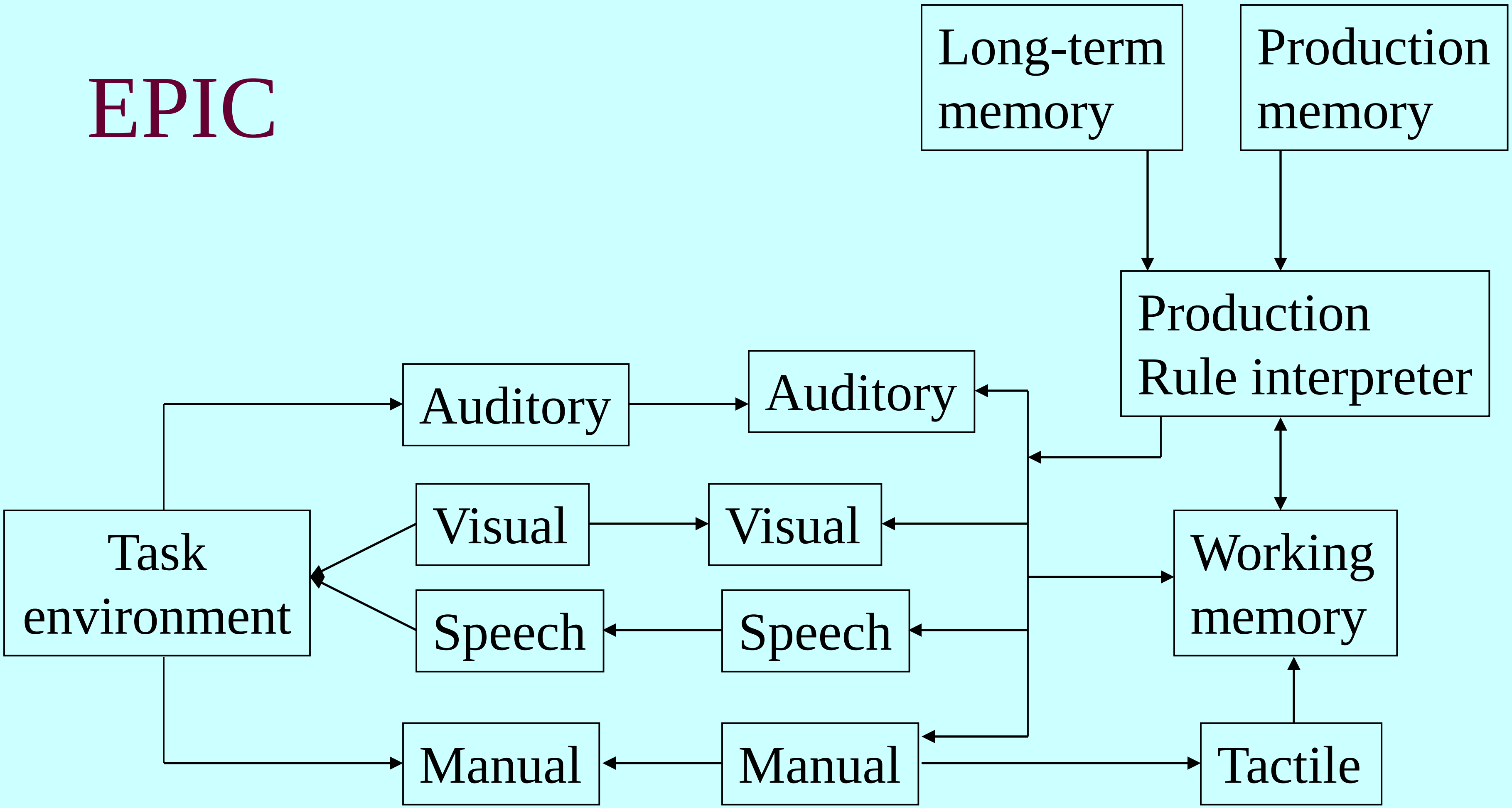
Executive Process Interactive Control (EPIC)

- Focus on multiple task performance
- Cognitive Processor runs production rules and interacts with perceptual and motor processors

EPIC parameters

- FIXED
  - Connections and mechanisms
  - Time parameters
  - Feature sets for motor processors
  - Task-specific production rules and perceptual encoding types
- FREE
  - Production rules for tasks
  - Unique perceptual and motor processors
  - Task instance set
  - Simulated task environment

EPIC



Long-term memory
Production memory
Production Rule interpreter
Auditory
Auditory
Visual
Visual
Task environment
Speech
Speech
Working memory
Manual
Manual
Tactile

Production Memory

- Perceptual processors controlled by production rules
- Production Rules held in Production Memory
- Production Rule Interpreter applies rules to perceptual processes

Working Memory

- Limited capacity (or duration of 4s) and holds current production rules
- Cognitive processor updates every 50ms
- On update, perceptual input, item from production memory, and next action held in working memory

Resolving Conflict

- Production rules applied to executive tasks to handle resource conflict and scheduling
- Conflict dealt with in production rule specification
  - Lockout
  - Interleaving
  - Strategic response deferent

Example

Task one

- Stimulus one
- Perceptual process
- Cognitive process
- Response selection
- Memory process
- Response one

Executive process

- Move eye to S2
- Enable task1 + task 2
- Wait for task1 complete
- Task1end
- Task2 permission
- Trial end

Task two

- Stimulus two
- Perceptual process
- Cognitive process
- Response selection
- Memory process
- Response two

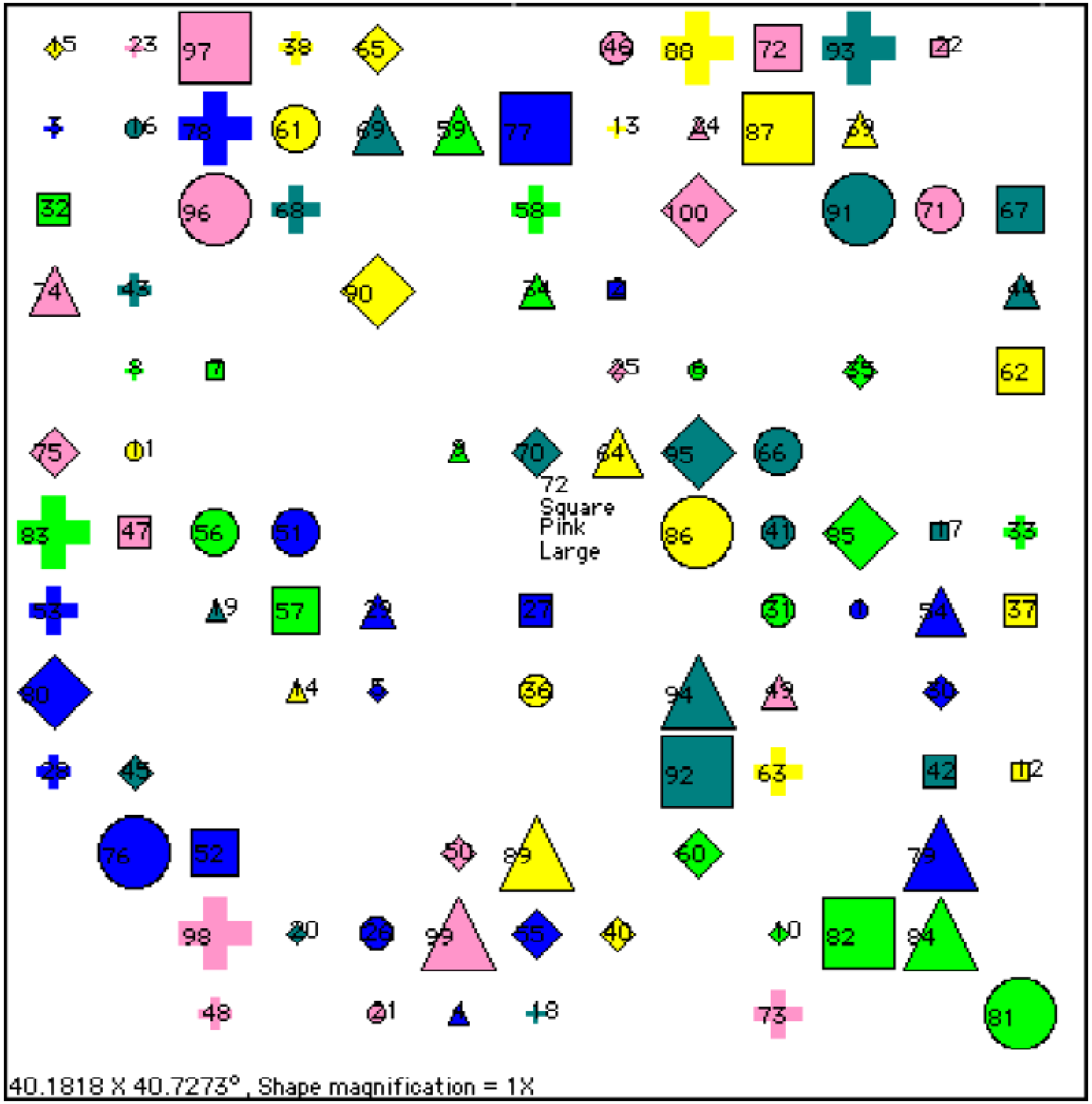
Conclusions

- Modular structure supports parallelism
- EPIC does not have a goal stack and does not assume sequential firing of goals
- Goals can be handled in parallel (provided there is no resource conflict)
- Does not support learning

UNIVERSITY OF BIRMINGHAM

Williams (1967)

- 100 objects on the display, varying in size, shape and colour (each object has unique ID number)
- Model performs a search task under different constraints, e.g., ID only; colour; size; shape



UNIVERSITY OF BIRMINGHAM

EPIC Tutorial Slides.pdf - Adobe Reader

Basic Results:

Fixation Type	Prop.	RT(s)	Est. Fixations
ID only*	.20	22.8	68
Color	.61	6.8	20
Size w/o Color	.36	16.1	48
Very Large	.54		
Oth. Size	.30		
Shape only	.25	20.7	62

Results averaged over all conditions where the stimulus property was in the probe.

E.g. all cases where Color was supplied were averaged together.

***When only the ID is supplied, fixations uniformly distributed across features; very slow.**

Proportion shown is for color and shape; size is .25

Eye movements strongly guided by color, not by shape, by largest size only.

Number of fixations estimated from RT and reported rate of 3/sec.

Adaptive Control of Thought, Rational (ACT-R)

- ACT-R symbolic aspect realised over subsymbolic mechanism
- Symbolic aspect in two parts:
  - Production memory
  - Symbolic memory (declarative memory)
- Theory of rational analysis

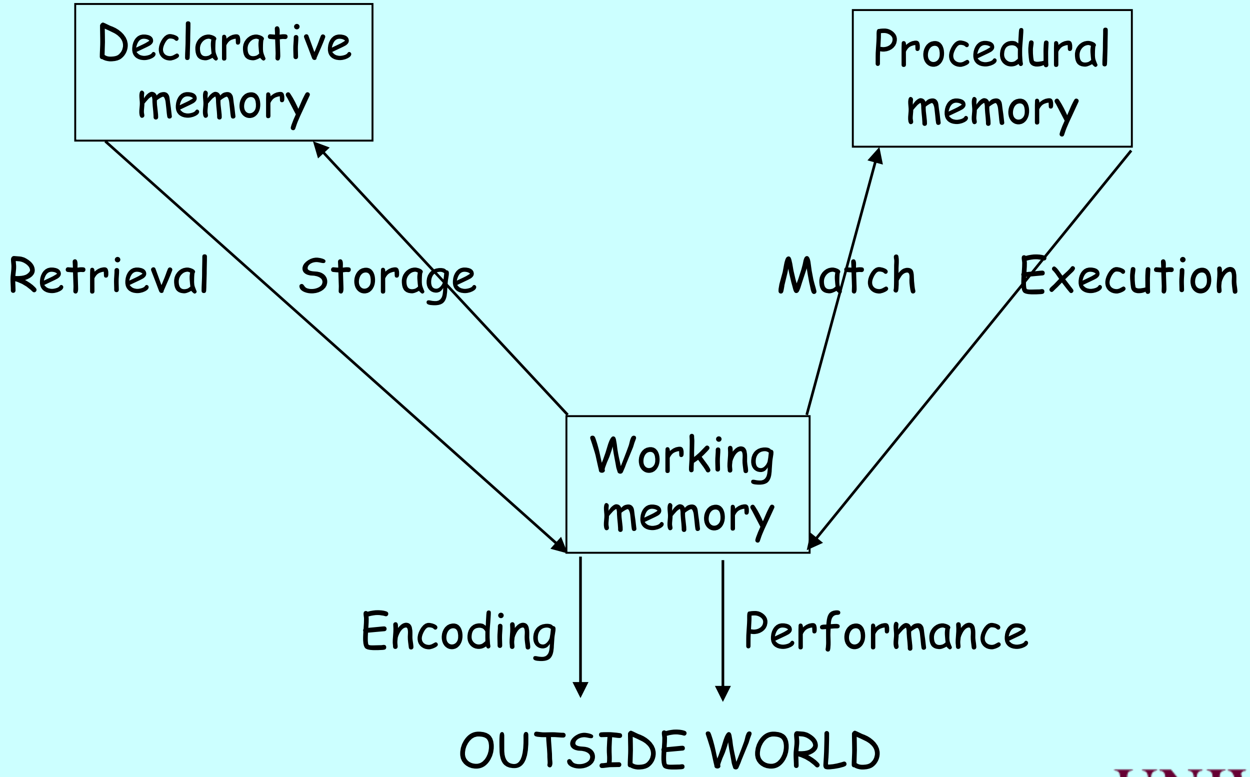
Theory of Rational Analysis

- Evidence-based assumptions about environment (probabilities)
- Deriving optimal strategies (Bayesian)
- Assuming that optimal strategies reflect human cognition (either what it actually does or what it probably ought to do)

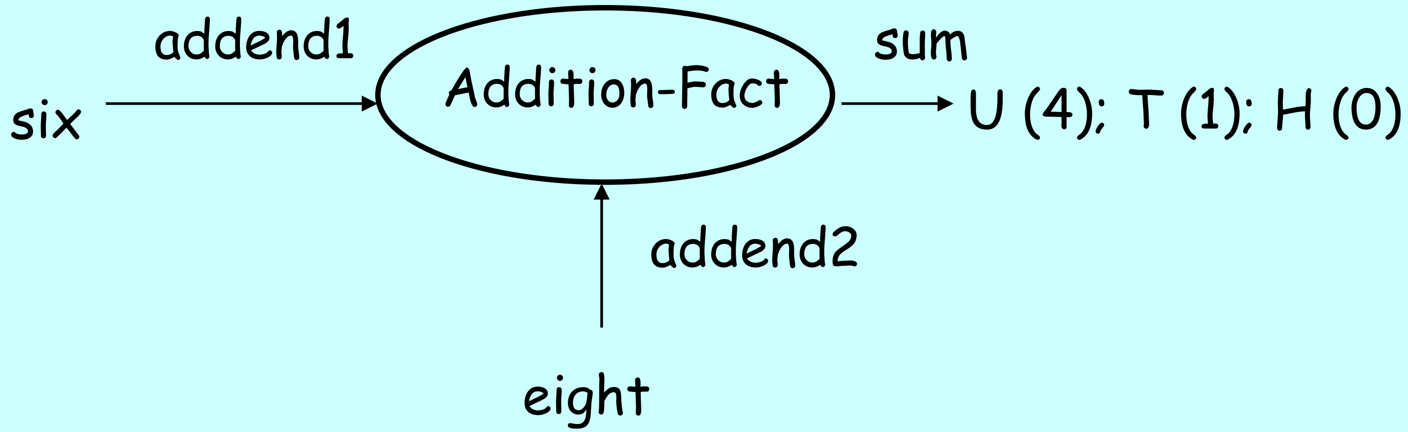
Notions of Memory

- Procedural
  - Knowing how
  - Described in ACT by Production Rules
- Declarative
  - Knowing that
  - Described in ACT by ‘chunks’
- Goal Stack
  - A sort of ‘working memory’
  - Holds chunks (goals)
  - Top goal pushed (like GOMS)
  - Writeable

ACT*


Declarative memory
Procedural memory
Retrieval
Storage
Match
Execution
Working memory
Encoding
Performance
OUTSIDE WORLD

Knowledge Representation



16
18 +
_____
34
_____
1

Goal buffer:	add numbers in right-most column
Visual buffer:	6, 8
Retrieval buffer:	14

UNIVERSITY OF BIRMINGHAM

Symbolic / Subsymbolic levels

- Symbolic level
  - Information as chunks in declarative memory, and represented as propositions
  - Rules as productions in procedural memory
- Subsymbolic level
  - Chunks given parameters which are used to determine the probability that the chunk is needed
  - Base-level activation (relevance)
  - Context activation (association strengths)

Conflict resolution

- Order production rules by preference
- Select top rule in list
- Preference defined by:
  - Probability that rule will lead to goal
  - Time associated with rule
  - Likely cost of reaching goal when using sequence involving this rule

Example

- Activity: Find target and then use mouse to select target:

Hunt_Feature

IF goal = find target with feature F

AND there is object X on screen

THEN move attention to object X

Found_target

IF goal = find target with feature F

AND target with F in location L

THEN move mouse to L and click

Example

- Model reaction time to target
  - Assume switch attention linearly increases with each new position
  - Assume probability of feature X in location y = 0.53
  - Assume switch attention = 185ms
- Therefore, reaction time = 185 X 0.53 = 98ms per position
- Empirical data has RT of 103ms per position

Example

- Assume target in field of distractors
  - P = 0.42
  - Therefore, 185 x .42 = 78ms per position

- Empirical data = 80ms per position

Python Shell

File Edit Shell Debug Options Windows Help

```
Python 2.4.2 (#67, Sep 28 2005, 12:41:11) [MSC v.1310 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.

    ****************************************************************
    Personal firewall software may warn about the connection IDLE
    makes to its subprocess using this computer's internal loopback
    interface.  This connection is not visible on any external
    interface and no data is sent to or received from the Internet.
    ****************************************************************

IDLE 1.1.2
>>> ================================ RESTART ================================
>>>
>>> ================================ RESTART ================================
>>> ================================ RESTART ================================
>>>
0.000 play_p selected
0.050 play_p fired
Choose (R)ock, (P)aper, or (S)cissors:r
     paper(  1)       rock(  0)            PLAYER 1 WINS
0.050 play_p selected
0.100 play_p fired
Choose (R)ock, (P)aper, or (S)cissors:p
     paper(  1)      paper(  0)                      TIE
0.100 play_p selected
0.150 play_p fired
Choose (R)ock, (P)aper, or (S)cissors:p
     paper(  1)      paper(  0)                      TIE
0.150 play_s selected
0.200 play_s fired
Choose (R)ock, (P)aper, or (S)cissors:r
  scissors(  1)       rock(  1)            PLAYER 2 WINS
0.200 play_r selected
0.250 play_r fired
Choose (R)ock, (P)aper, or (S)cissors:s
      rock(  2)   scissors(  1)            PLAYER 1 WINS
0.250 play_r selected
0.300 play_r fired
Choose (R)ock, (P)aper, or (S)cissors:r
      rock(  2)       rock(  1)                      TIE
0.300 play_r selected
0.350 play_r fired
Choose (R)ock, (P)aper, or (S)cissors:p
      rock(  2)      paper(  2)            PLAYER 2 WINS
0.350 play_r selected
0.400 play_r fired
Choose (R)ock, (P)aper, or (S)cissors:r
      rock(  2)       rock(  2)                      TIE
0.400 play_s selected
```

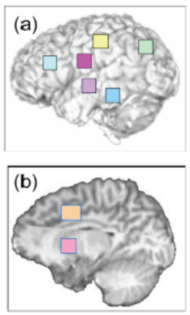
UNIVERSITY OF BIRMINGHAM

Learning

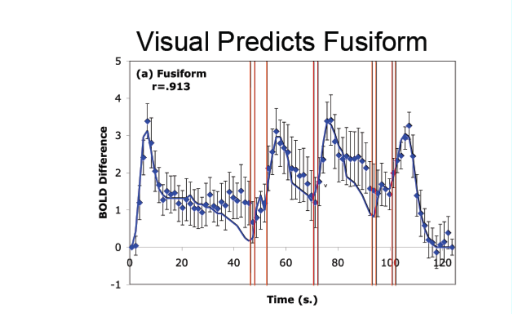
- Symbolic level
  - Learning defined by adding new chunks and productions
- Subsymbolic level
  - Adjustment of parameters based on experience

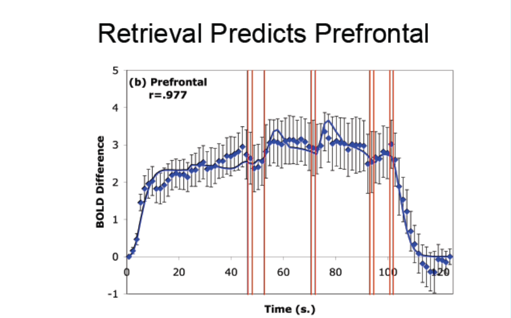
ACT-R and fMRI

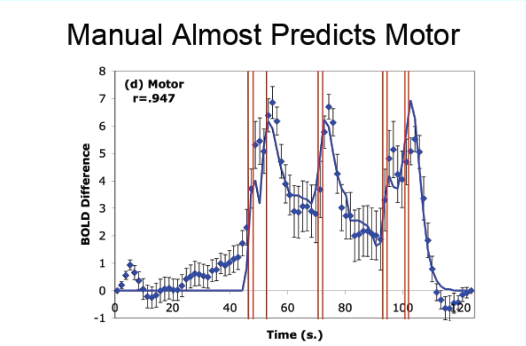
Module-Brain Mappings



	x	y	z	Region
Manual	37	-25	47	Motor
Imaginal	23	-64	34	Parietal
Vocal	44	-12	29	Motor
Declarative	40	21	21	Prefrontal
Aural	47	-16	4	Auditory
Visual	42	-60	-8	Fusiform
Goal	5	10	38	ACC
Procedural	15	9	2	Caudate







http://act-r.psy.cmu.edu/

UNIVERSITY OF BIRMINGHAM

Conclusions

- ACT use simple production system
- ACT provides some quantitative prediction of performance
- Rationality = optimal adaptation to environment