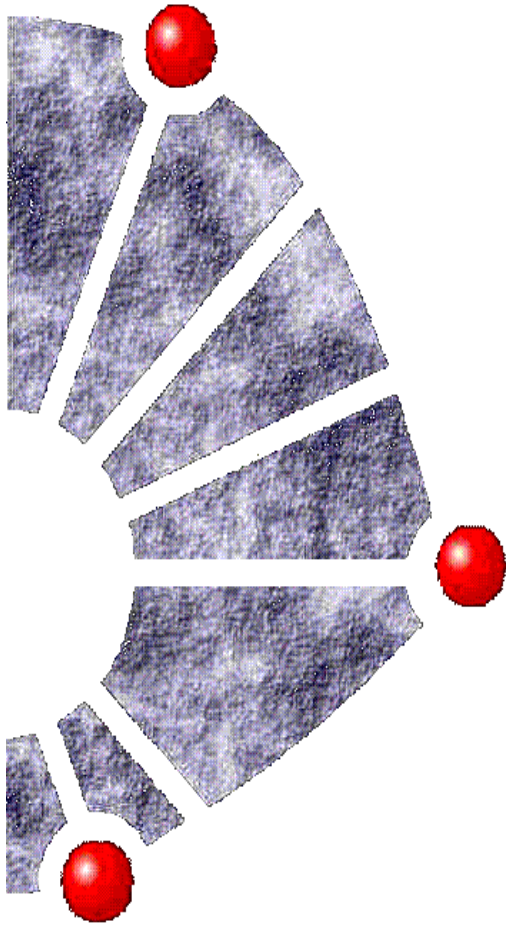


IA889 – Sistemas de Cognição Artificial



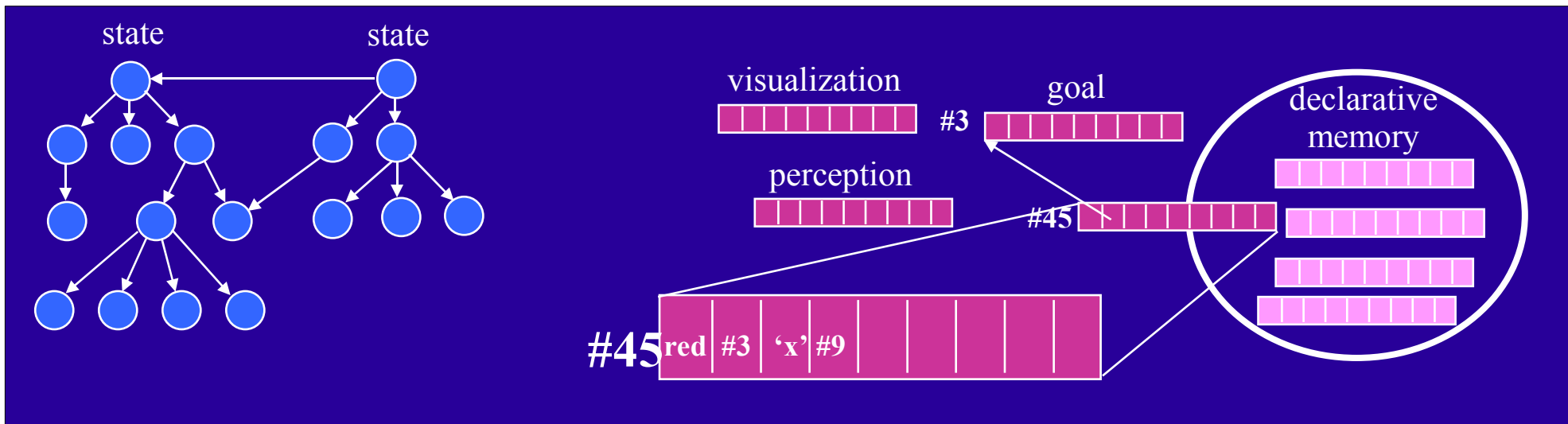
AULA 12

ACT-R e SOAR



Arquiteturas Cognitivas Genéricas

- ACT-R
 - Adaptive Control of Thought – Rational
 - Anderson and Lebiere
- SOAR
 - State Operator And Result
 - Laird, Newell and Rosembaum

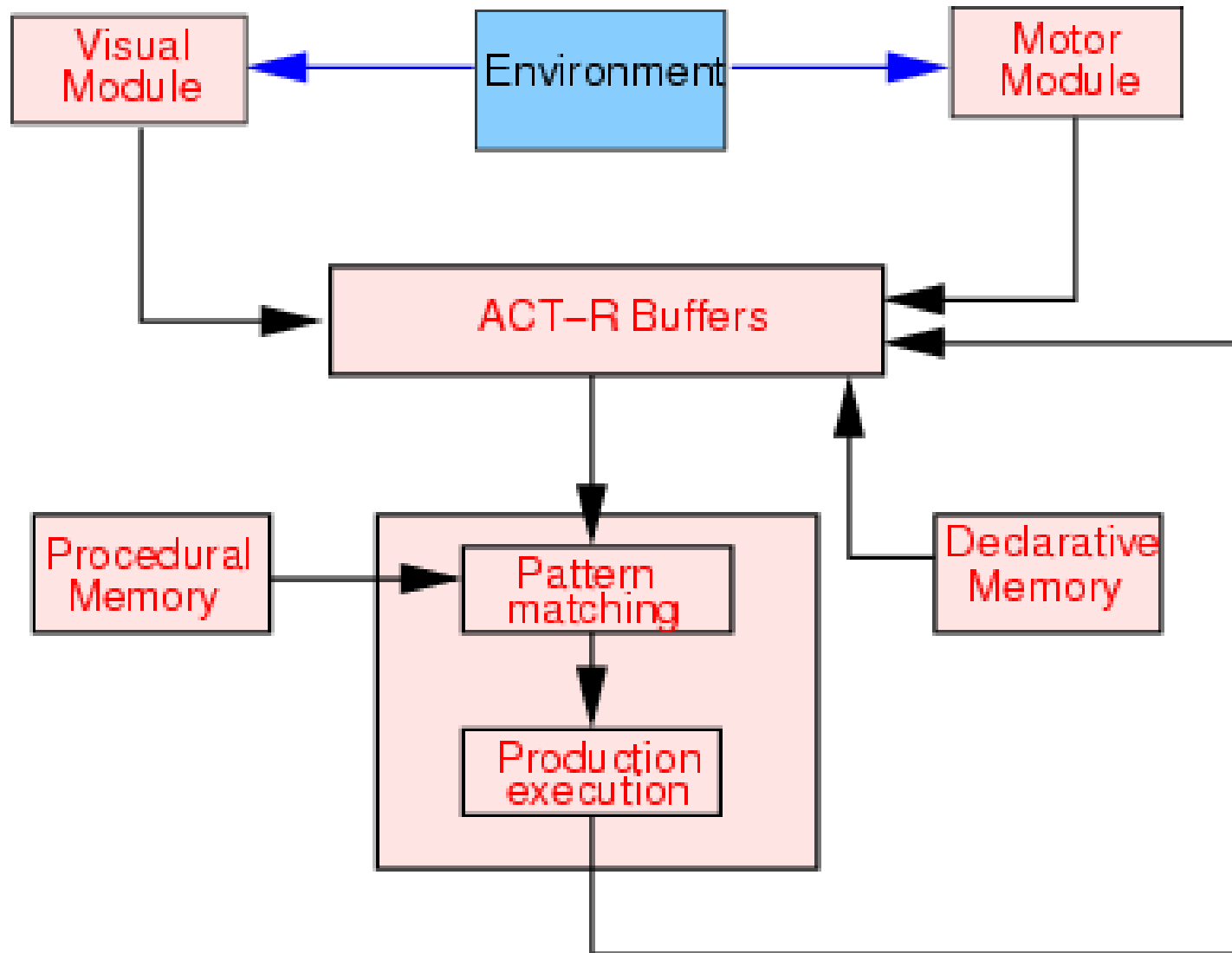




ACT-R

- ACT-R
 - Atomic Components of Thought – Rational
 - Adaptive Control of Thought - Rational
 - Carnegie Mellon University - John R. Anderson
- Arquitetura Híbrida
 - Estruturas Simbólicas
 - sistema de produção
 - Estruturas Sub-simbólicas
 - conjunto massivo de processos paralelos, sumarizados por meio de equações matemáticas
 - podem controlar processos simbólicos
 - funções de utilidade
 - processos de aprendizagem

ACT-R





Histórico do Desenvolvimento

Predecessor	HAM	(Anderson & Bower 1973)
Versões Teóricas	ACT-E	(Anderson, 1976)
	ACT*	(Anderson, 1978)
	ACT-R	(Anderson, 1993)
	ACT-R 4.0	(Anderson & Lebiere, 1998)
	ACT-R 5.0	(Anderson & Lebiere, 2001)
Implementações	GRAPES	(Sauers & Farrell, 1982)
	PUPS	(Anderson & Thompson, 1989)
	ACT-R 2.0	(Lebiere & Kushmerick, 1993)
	ACT-R 3.0	(Lebiere, 1995)
	ACT-R 4.0	(Lebiere, 1998)
	ACT-R/PM	(Byrne, 1998)
	ACT-R 5.0	(Lebiere, 2001)
	Windows Environment	(Bothell, 2001)
	Macintosh Environment	(Fincham, 2001)
	ACT-R 6.0	(Bothell 2004)



- **Conhecimento Declarativo**
 - coisas que somos consciente de que sabemos e que podemos usualmente descrever para outras pessoas
 - representado em termos de chunks
 - configurações de elementos que codificam as diversas coisas que sabemos
- **Conhecimento Procedural**
 - conhecimento que exibimos por meio de nosso comportamento, mas do qual não somos conscientes
 - regras de produção especificam como buscar e utilizar o conhecimento declarativo para resolver problemas



ACT-R

	Memória Declarativa	Memória Procedural
Simbólico	Chunks: fatos declarativos	Produções: SE (cond) ENTÃO (ação)
Sub-simbólico	Ativação de chunks (probabilidade de seleção)	Utilidade: Resolução de Conflitos (probabilidade de uso)



• Exemplos de Chunks

• Definindo tipos de chunks (categorias)

```
(chunk-type name slot-name-1 slot-name-2 ... slot-name-n)
```

```
(chunk-type bird species color size)
```

```
(chunk-type column row1 row2 row3)
```

```
(chunk-type count-order first second)
```

```
(chunk-type count-from start end)
```

• Definindo chunks

```
(add-dm
```

```
  (b ISA count-order first 1 second 2)
```

```
  (c ISA count-order first 2 second 3)
```

```
  (d ISA count-order first 3 second 4)
```

```
  (e ISA count-order first 4 second 5)
```

```
  (f ISA count-order first 5 second 6)
```

```
  (first-goal ISA count-from start 2 end 4))
```



- Buffers

- Interface entre a memória procedural e os outros módulos do sistema
 - Buffer 'goal': interface para o módulo 'goal'
 - Buffer 'retrieval': interface para o módulo 'retrieval'
- Cada buffer só pode ter um único chunk a cada instante de tempo
- Valores dos buffers podem ser alterados por produções

- Resolução de Conflitos

- Somente uma produção pode disparar a cada instante
- Escolher a produção é uma resolução de conflitos



• Exemplos de Produções

```
(P counting-example
  =goal>
    isa      count
    state    incrementing
    number   =num1
  =retrieval>
    isa      count-order
    first    =num1
    second   =num2

==>
  =goal>
    number   =num2
  +retrieval>
    isa      count-order
    first    =num2
)
```

English Description

If the goal chunk is
of the type count
the state slot is incrementing
there is a number we will call =num1
and the chunk in the retrieval buffer
is of type count-order
the first slot has the value =num1
and the second slot has a value
we will call =num2
Then
change the goal
to continue counting from =num2
and request a retrieval
of a count-order chunk to
find the number that follows =num2

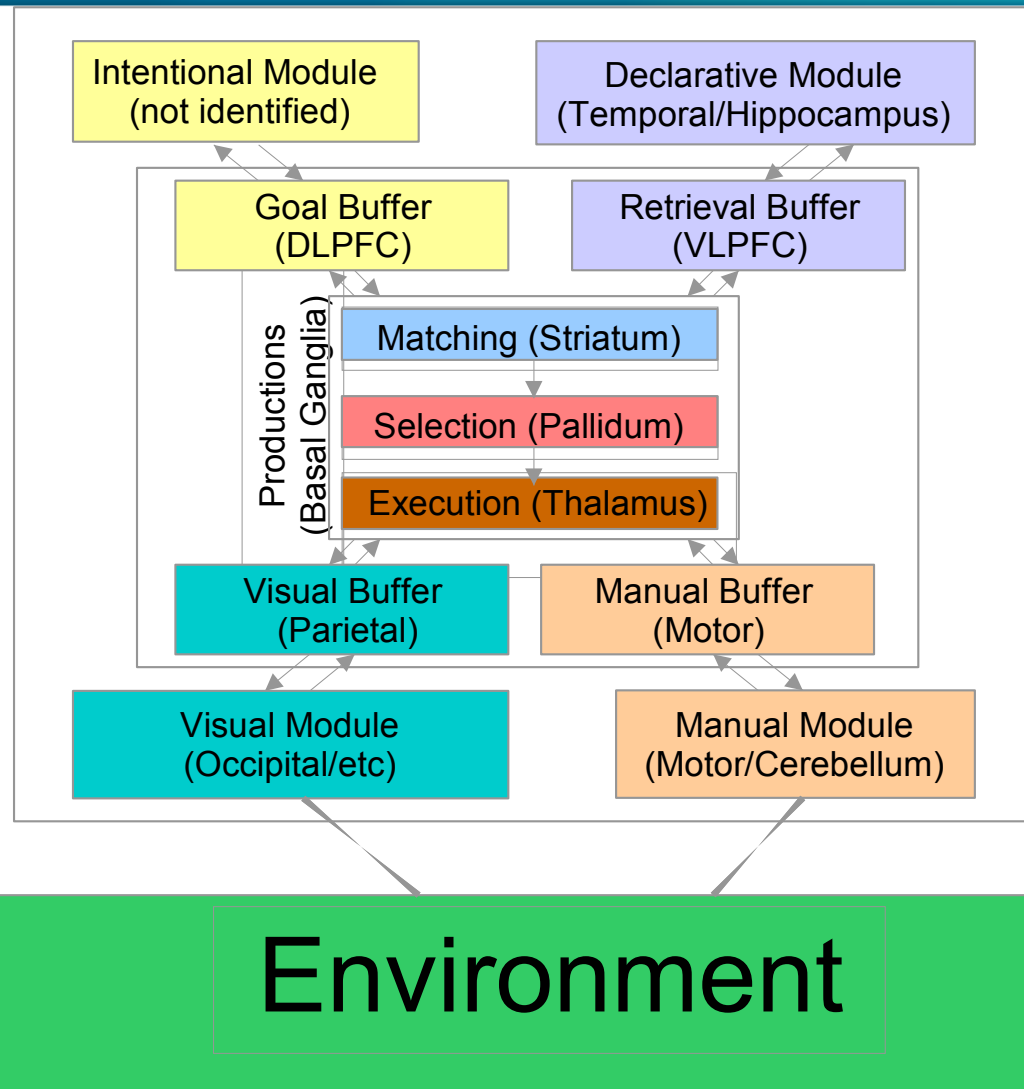


ACT-R

- Ações sobre os buffers
 - =retrieval>
 - Modifica valores dos slots do buffer
 - +retrieval>
 - Faz uma requisição ao módulo em questão
 - -retrieval>
 - Limpa o buffer em questão
 - ?retrieval>
 - Efetua 'queries' ao módulo
- Modelo ACT-R
 - Programa em LISP contendo os comandos ACT-R



ACT-R 5.0





ACT-R

- ACT-R 6.0
 - Ambiente ACT-R
 - Ferramentas de depuração e teste de modelos

```
listener
0.050 PROCEDURAL CONFLICT-RESOLUTION
0.100 DECLARATIVE RETRIEVED-CHUNK FACT67
0.100 DECLARATIVE SET-BUFFER-CHUNK RETRIEVAL FACT67
0.100 PROCEDURAL CONFLICT-RESOLUTION
0.150 PROCEDURAL PRODUCTION-FIRED ADD-ONES
0.150 PROCEDURAL CLEAR-BUFFER RETRIEVAL
0.150 DECLARATIVE START-RETRIEVAL
0.150 PROCEDURAL CONFLICT-RESOLUTION
0.200 DECLARATIVE RETRIEVED-CHUNK FACT103
0.200 DECLARATIVE SET-BUFFER-CHUNK RETRIEVAL FACT103
0.200 PROCEDURAL CONFLICT-RESOLUTION
0.250 PROCEDURAL PRODUCTION-FIRED PROCESS-CARRY
0.250 PROCEDURAL CLEAR-BUFFER RETRIEVAL
0.250 DECLARATIVE START-RETRIEVAL
0.250 PROCEDURAL CONFLICT-RESOLUTION
0.300 DECLARATIVE RETRIEVED-CHUNK FACT34
0.300 DECLARATIVE SET-BUFFER-CHUNK RETRIEVAL FACT34
0.300 PROCEDURAL CONFLICT-RESOLUTION
0.350 PROCEDURAL PRODUCTION-FIRED ADD-TENS-CARRY
0.350 PROCEDURAL CLEAR-BUFFER RETRIEVAL
0.350 DECLARATIVE START-RETRIEVAL
0.350 PROCEDURAL CONFLICT-RESOLUTION
0.400 DECLARATIVE RETRIEVED-CHUNK FACT17
0.400 DECLARATIVE SET-BUFFER-CHUNK RETRIEVAL FACT17
0.400 PROCEDURAL CONFLICT-RESOLUTION
0.450 PROCEDURAL PRODUCTION-FIRED ADD-TENS-DONE
0.450 PROCEDURAL CLEAR-BUFFER RETRIEVAL
0.450 PROCEDURAL CONFLICT-RESOLUTION
0.450 ----- Stopped because no events left to process
Command:
```

Contro

Current Model
No Model

Model
Load Model

Open Model: Existing

Save Model

Close Model

Running
Run 10.0
Reset Reload
Stepper

Inspecting
Declarative viewer
Procedural viewer
Buffer viewer
Buffer Status viewer

Visicon
Audicon

Tracing
Horiz. Buffer Trace
Vert. Buffer Trace

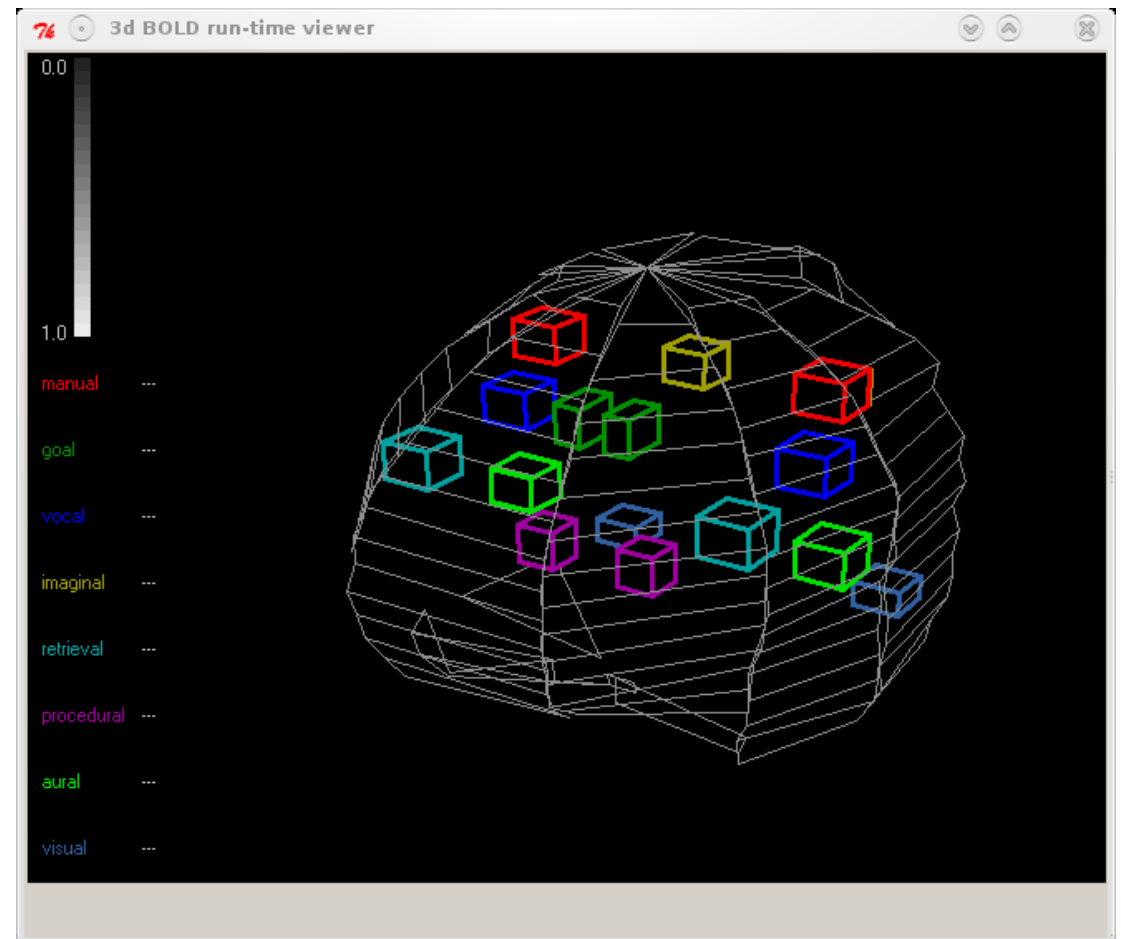
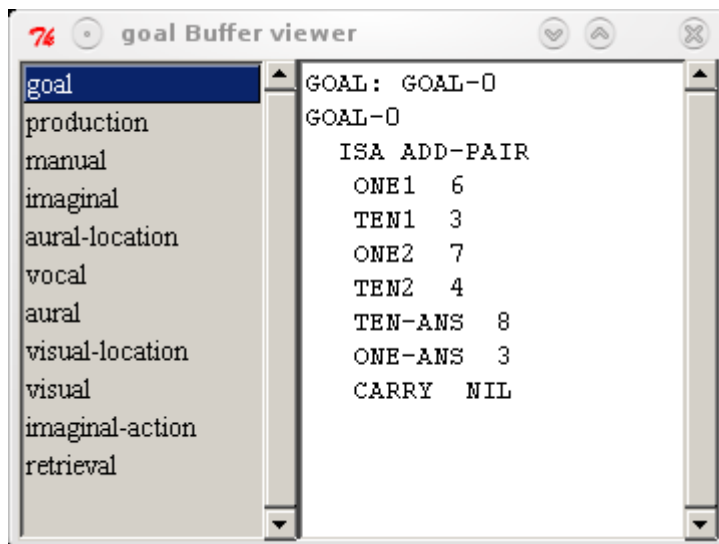
BOLD tools
Buffer graphs
2D brain
3D brain
Run-time 3D brain

Miscellaneous
Listener
Options



ACT-R

- Versão 6.0
 - Novos módulos
 - Novos buffers





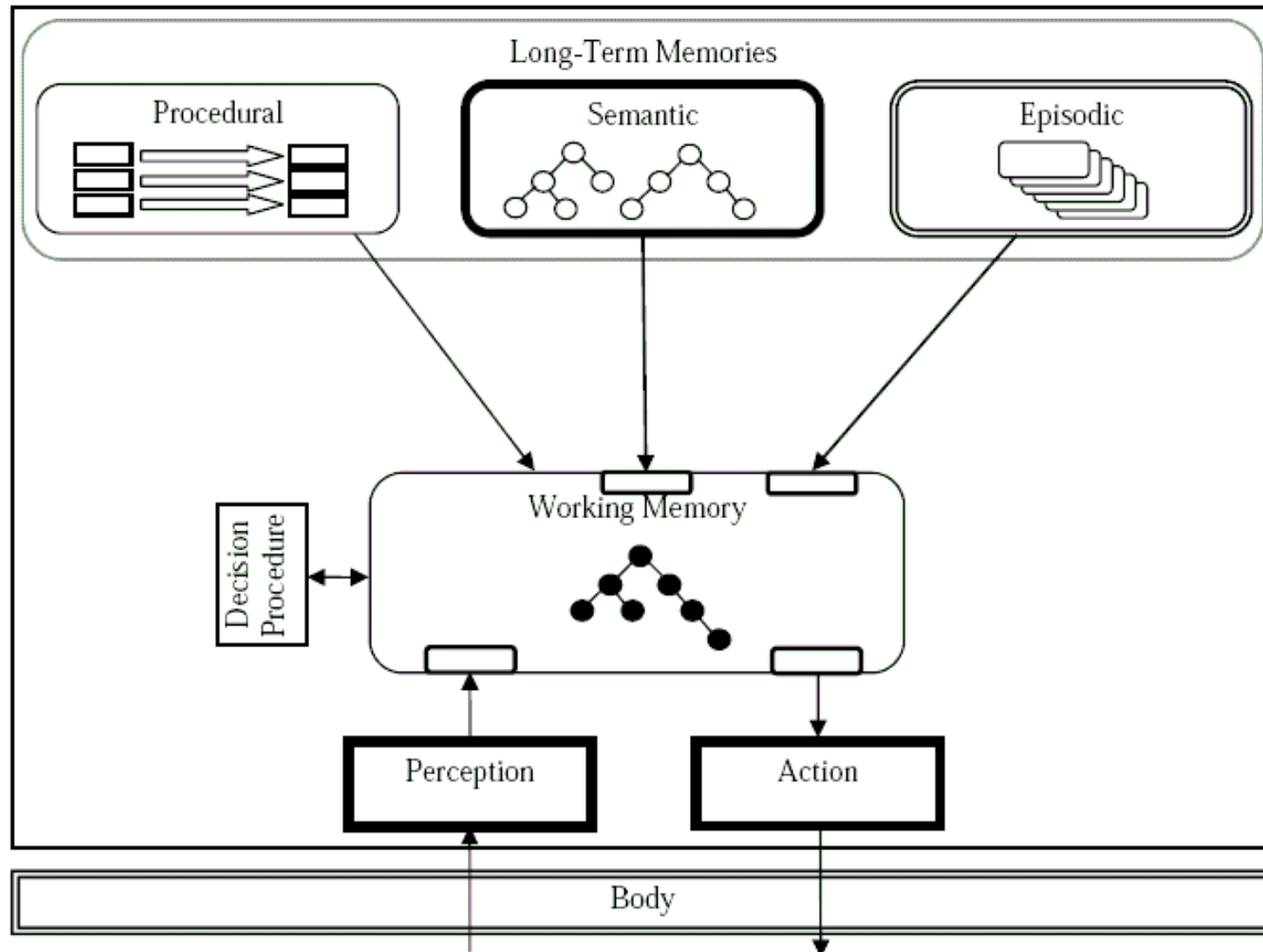
ACT-R

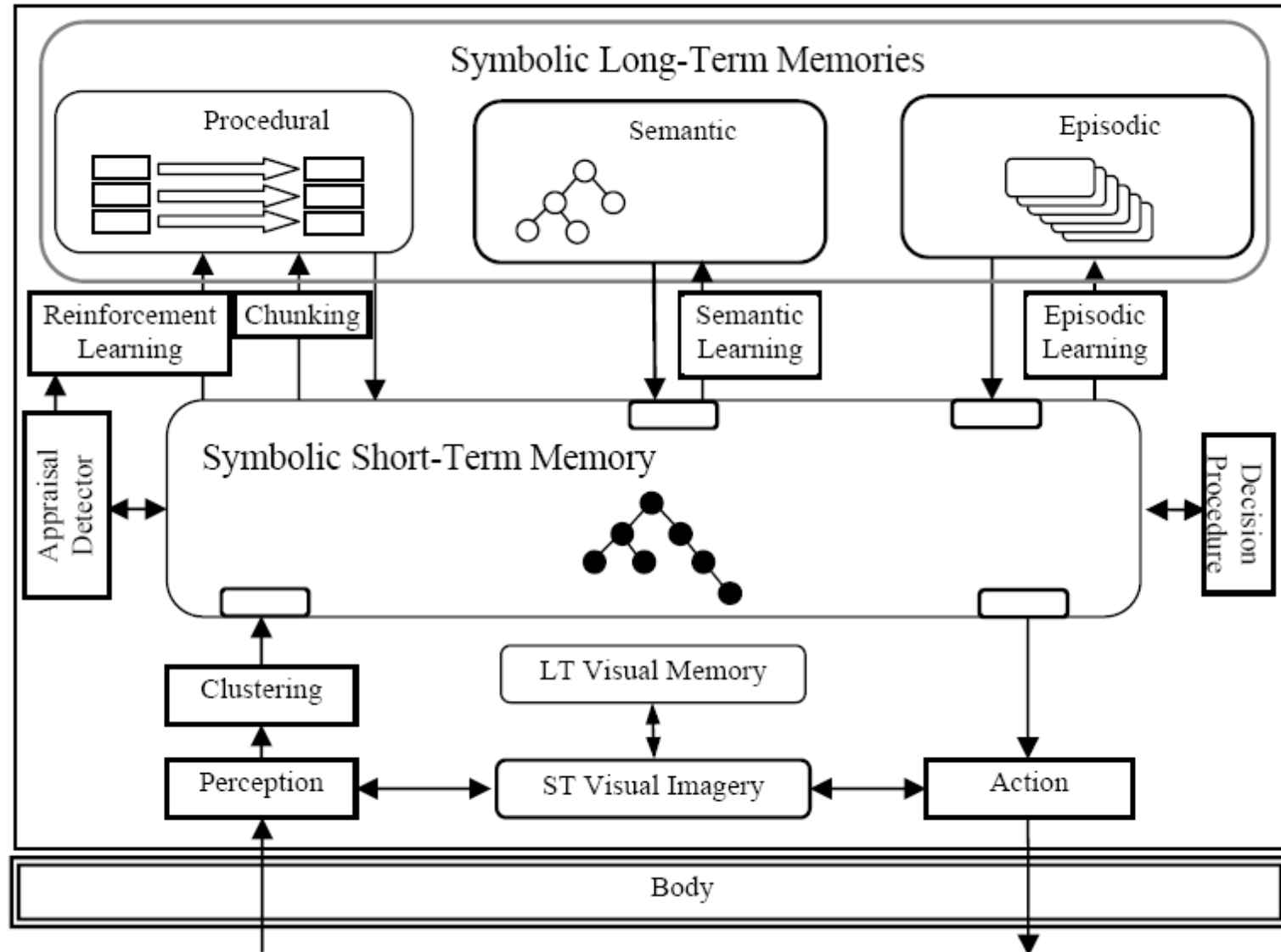
- Integração com Software
 - Versão Original
 - Lisp
 - Dificuldade de integração com outras linguagens
 - jactr.org (versão em Java)
 - Não tão atualizada
- Para saber mais ...
 - Download do software no site
 - <http://act-r.psy.cmu.edu/>
 - Tutoriais
 - Explicando o uso dos diferentes *modules*
 - Manual de Referência



SOAR

- SOAR (State, Operator And Result)
 - John Laird, Allen Newell, Paul Rosenbloom (1987)
 - Carnegie-Mellon University
 - University of Michigan
- Arquitetura cognitiva
 - Sistema de Símbolos Físicos
 - Sistemas de Produção
 - Busca em espaços de problemas
 - estrutura de controle automática/deliberativa em dois níveis
 - Aprendizagem contínua, determinada por impasses







- Estados e Operadores
 - estruturas básicas suportadas pela arquitetura
 - Estados: toda informação sobre a situação corrente, incluindo a percepção e a descrição de metas correntes e espaços de problemas
 - Operadores: ocasionam passos no espaço de problemas
- Memória de Trabalho
 - percepções e hierarquia de estados e seus operadores associados
 - conteúdo pode acionar a memória de longo prazo ou ações motoras



- **Memória de Longo Prazo**
 - Repositório do conteúdo processado pela arquitetura capaz de produzir comportamento
 - Memória Procedural: Regras
 - acessada automaticamente durante os ciclos de decisão
 - Memória Semântica: Estruturas Declarativas
 - Memória Episódica: Episódios
 - Memórias de longo prazo são impenetráveis
 - não podem ser examinadas diretamente
 - certos procedimentos recuperam informações nas memórias de longo prazo e armazenam na memória de trabalho



- Interface Perceptiva/Motora
 - Mapeamentos do mundo externo para representações internas na memória de trabalho e de representações internas para o mundo externo
 - Percepção e Ação podem acontecer em paralelo com o processo de cognição
- Ciclo de Decisão
 - Processo arquitetural básico suportando a cognição
 - Seleção e Aplicação de Operadores
 - Três fases
 - Elaboração, Decisão, Aplicação



- Fase de Elaboração
 - Acesso paralelo à Memória de Longo Prazo para elaborar o estado
 - Sugestão de novos operadores
 - Avaliação dos operadores
- Fase de Decisão
 - Procedimento de Decisão
 - Linguagem de Preferência por operadores
 - Resultado
 - Operador selecionado
 - Impasse
 - Preferências incompletas ou conflito



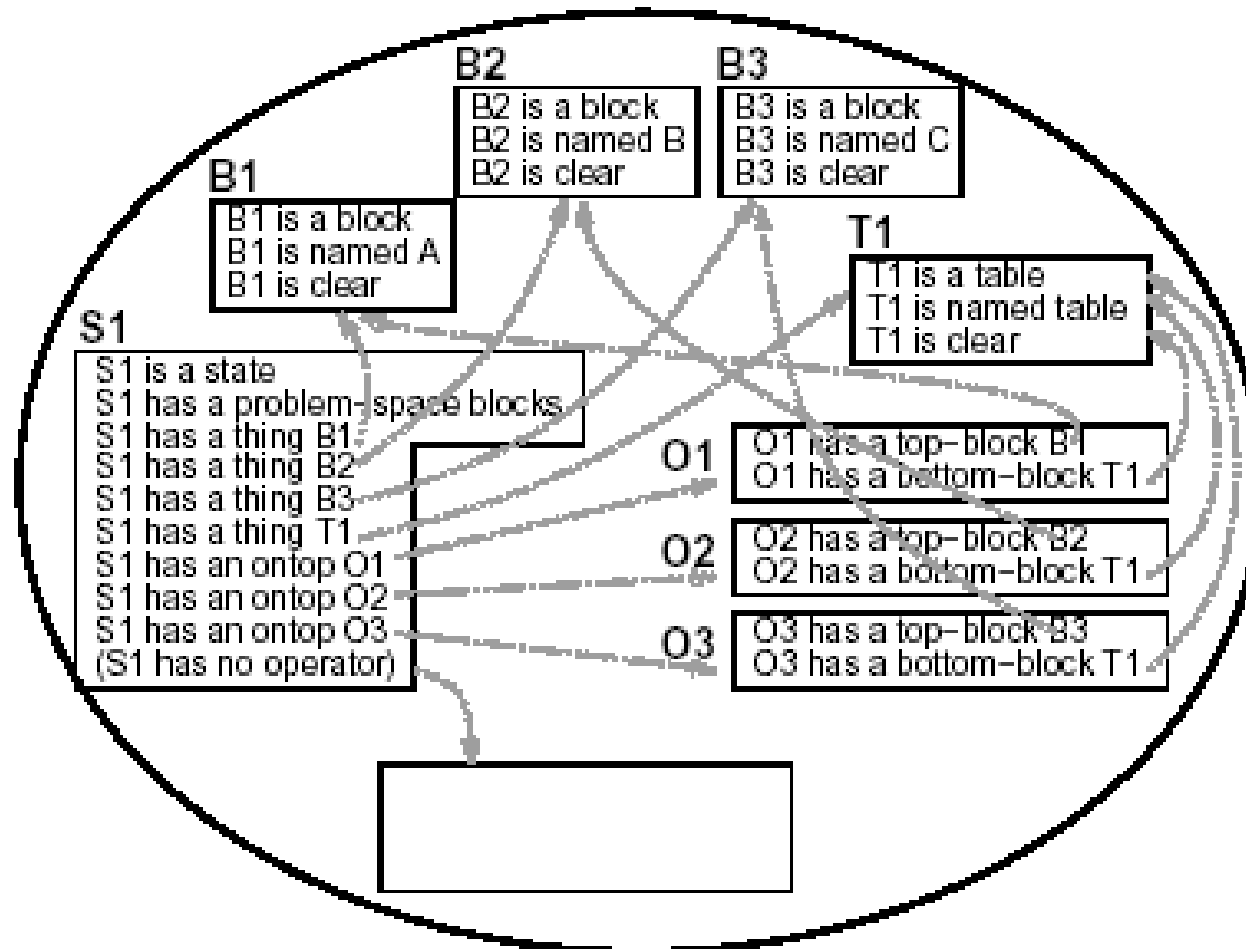
- Fase de Aplicação
 - regras são disparadas de forma a modificar os estados
 - Seleção de um único operador por ciclo de decisão impõe um gargalo cognitivo à arquitetura
 - limite no trabalho cognitivo por ciclo
- Impasses
 - sinalizam uma falta de conhecimento
 - oportunidade para aprendizagem
 - Acontecem automaticamente quando o conhecimento elicitado pelo estado corrente não é suficiente para o procedimento de decisão selecionar um operador



- Linguagem de Impasses
 - Independente de domínio
- Quando ocorre um Impasse
 - arquitetura automaticamente inicia a criação de um novo sub-estado cuja meta é resolver o impasse
 - impõe uma hierarquia de metas/sub-estados no contexto da memória de trabalho
- Quatro Mecanismos de Aprendizagem
 - Chunking, Aprendizagem por Reforço, Aprendizagem Episódica, Aprendizagem Semântica



- **Chunking**
 - cria automaticamente novas regras na memória de longo prazo utilizando-se dos resultados gerados de um impasse
 - previnem que um impasse ocorra em situações similares no futuro
- **Aprendizagem por Reforço**
 - ajusta os valores das preferências por operadores
- **Aprendizagem Episódica**
 - armazena a história das experiências
- **Aprendizagem Semântica**
 - captura asserções declarativas mais abstratas



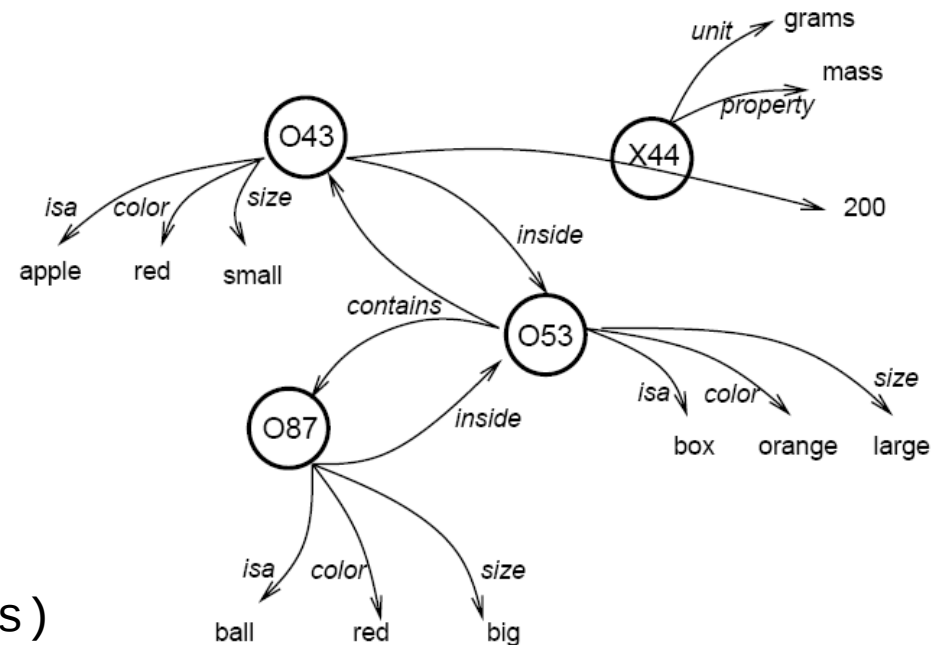
An Abstract View of Working Memory



Working Memory Elements (WME)

(identifier ^attribute value)
(identifier ^attribute-1 value-1
^attribute-2 value-2
^attribute-3 value-3...
^attribute-n value-n)

(043 ^isa apple
^color red
^inside 053
^size small
^X44 200)
(087 ^isa ball ^color red
^inside 053 ^size big)
(053 ^isa box ^size large
^color orange
^contains 043 087)
(X44 ^unit grams ^property mass)





• Preferências

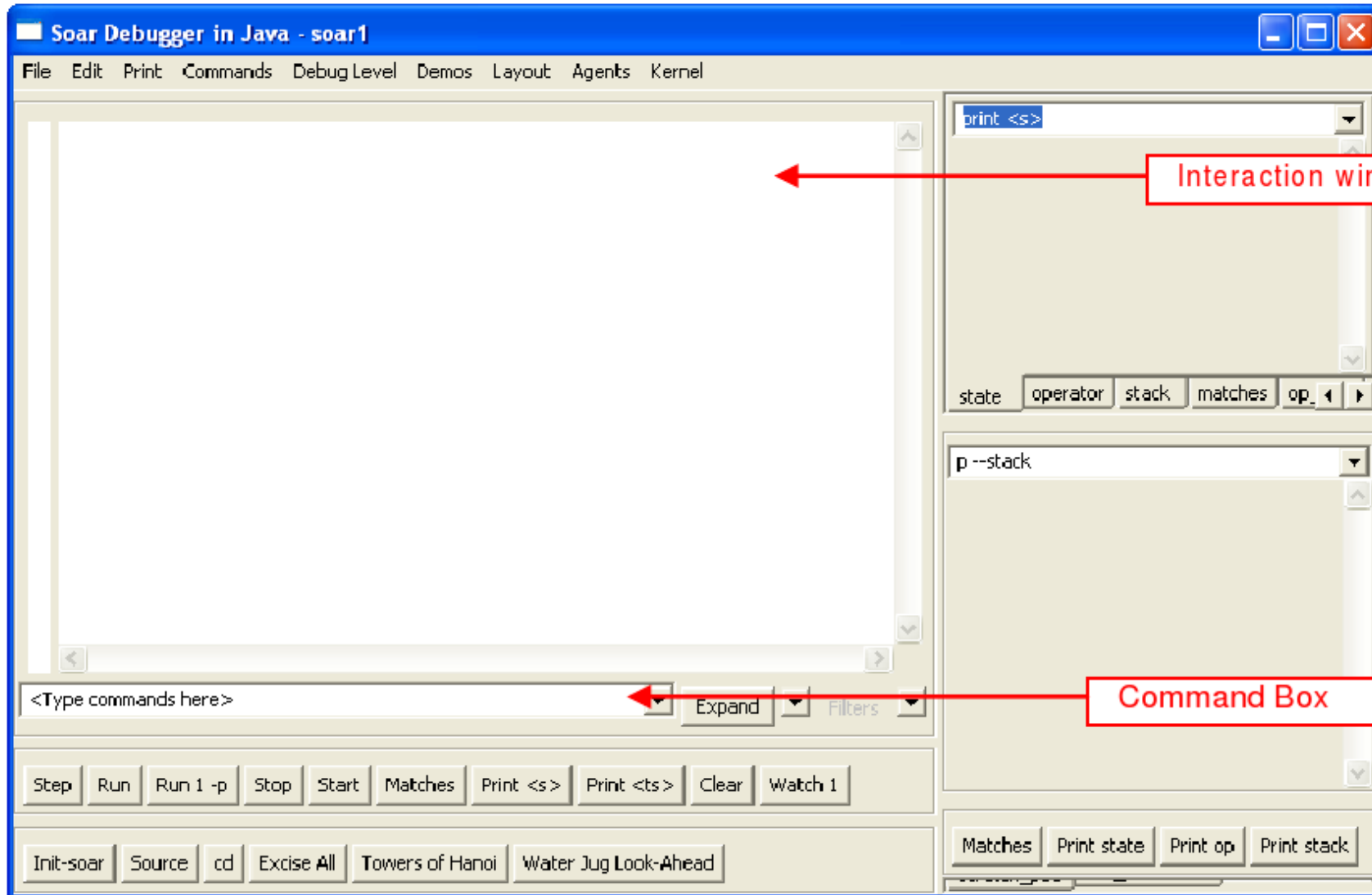
```
(S1 ^operator 03 +)  
(S1 ^operator 03 > 04)
```

• Produções

```
sp {blocks-world*propose*move-block  
  (state <s> ^problem-space blocks  
    ^thing <thing1> {<> <thing1> <thing2>}  
    ^ontop <ontop>)  
  (<thing1> ^type block ^clear yes)  
  (<thing2> ^clear yes)  
  (<ontop> ^top-block <thing1>  
    ^bottom-block <> <thing2>)  
  -->  
  (<s> ^operator <o> +)  
  (<o> ^name move-block ^moving-block <thing1>  
    ^destination <thing2>))}
```



SOAR Debugger





Visual SOAR

The image shows the Visual SOAR interface for a "water-jug" problem. The interface is divided into several panes:

- Left Pane (Tree View):** Displays a hierarchical tree of the problem state. The root node is "water-jug", which branches into "_firstload", "all", "elaborations", and "initialize-water-jug".
- Top Menu Bar:** Contains standard menu items: File, Edit, Search, Datamap, View, Soar Runtime, and Help.
- Center Pane (Datamap):** Titled "Datamap water-jug", it shows the current state of the problem. It includes a sub-menu with "Edit" and "Validation". The main content area lists the current state elements: "water-jug", "io", "name: [water-jug]", "operator: [initialize-water-jug]", "superstate: [nil]", "top-state", and "type: [state]".
- Right Pane (Feedback):** A large, empty area intended for providing feedback on the problem state.
- Bottom Pane (Feedback):** A smaller, empty area at the bottom of the interface, also labeled "Feedback".

Red arrows and text boxes provide guidance on how to interact with the interface:

- An arrow points to the "water-jug" root node in the left pane, with a text box stating: "The root of the tree and the name of your project, click here to open the Datamap".
- An arrow points to the "Datamap water-jug" title bar, with a text box stating: "This is the Datamap for water-jug".
- An arrow points to the center pane, with a text box stating: "This is the Operator window".
- An arrow points to the right pane, with a text box stating: "This is the Feedback window".
- An arrow points to the bottom pane, with a text box stating: "This is the Feedback window".



SOAR

- Integração com Software
 - Bindings para C++, Java, Tcl
 - Conexão via Sockets (outras linguagens)
 - SML: Soar Markup Language
- Para saber mais ...
 - Download do software no site
 - <http://sitemaker.umich.edu/soar/home>
 - Tutoriais
 - Explicando o uso do SOAR em diferentes situações
 - SOAR Manual
 - SOAR-RL Manual
 - Outros documentos