

A Arquitetura Cognitiva **CLARION**

Ricardo R. Gudwin

DCA-FEEC-UNICAMP

gudwin@unicamp.br

<http://www.dca.fee.unicamp.br/~gudwin>

A Arquitetura Cognitiva Clarion

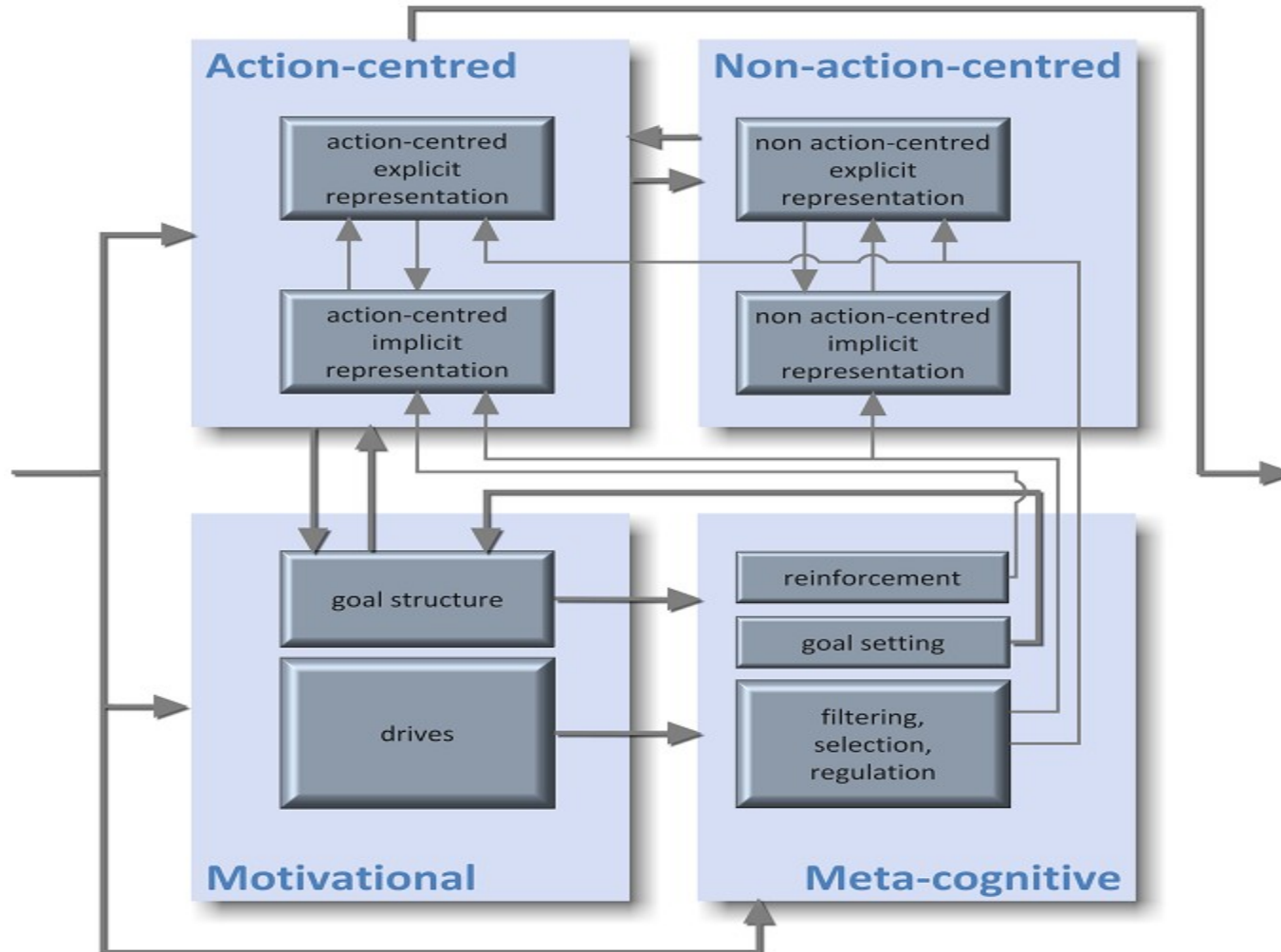
■ Ron Sun

- University of Missouri-Columbia - EUA

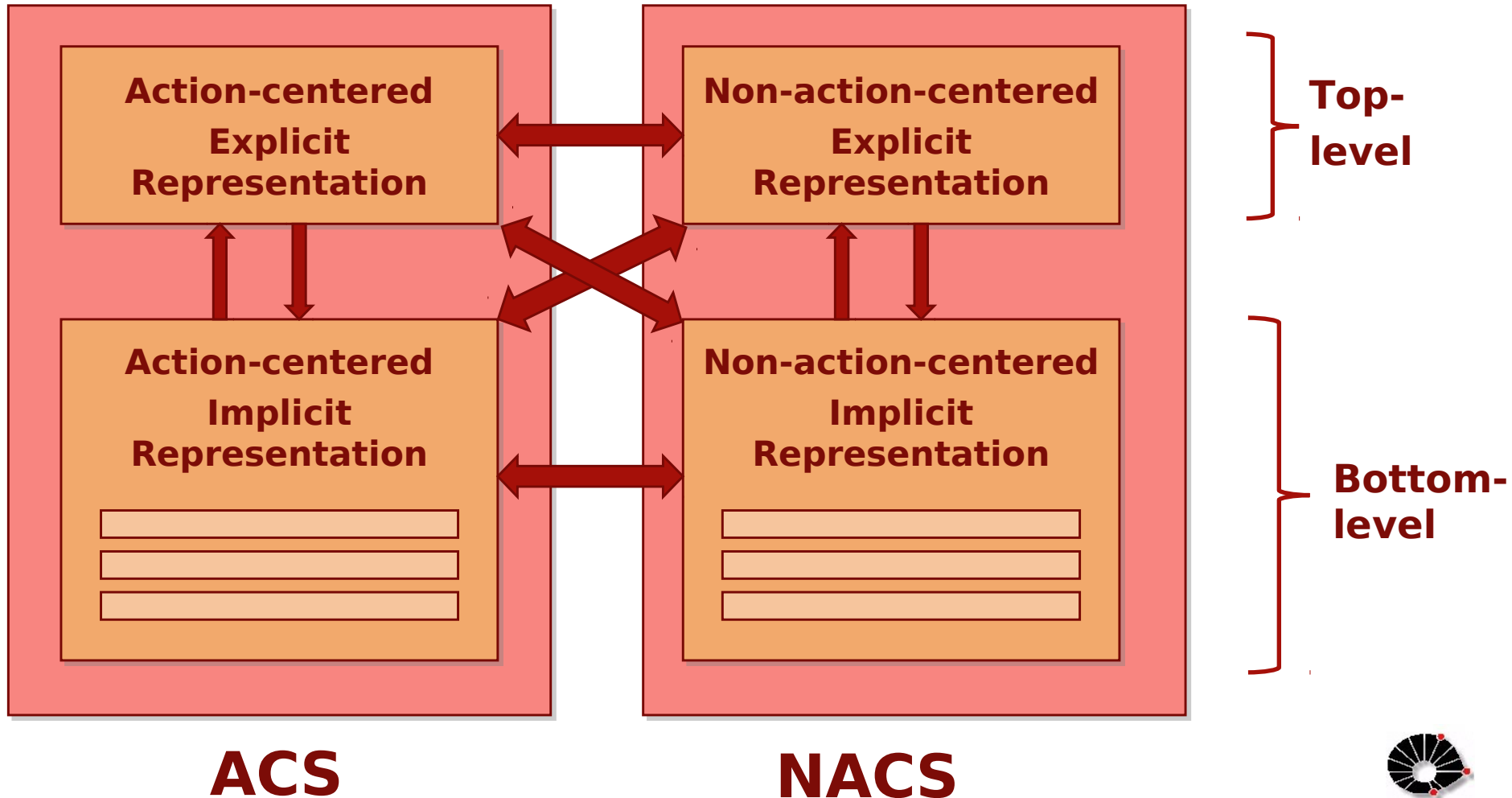
■ Clarion

- Cognição Implícita e Explícita
- Interação Cognição-Motivação-Ambiente
- Monitoramento Metacognitivo e controle do processamento cognitivo
- Módulos
 - Action Centered Sybsystem: comportamento
 - Non-Action Centered Subsystem: conhecimento
 - Supervisão
 - Motivational Subsystem
 - Metacognitive Subsystem

A Arquitetura Cognitiva Clarion



Representação: Top-Level x Bottom-Level



- Representação Explícita (“Top Level”)
 - Chunks e Rules
- Representação Implícita (“Bottom Level”)
 - Redes Neurais (Hopfield, BackPropagation)
- Ciclo Operacional do Módulo ACS
 - Observa o estado corrente
 - Computa o “bottom level” e determina uma ação
 - Computa o “top level” e determina uma ação
 - Seleciona uma ação apropriada combinando as ações do “bottom level” e do “top level”
 - Executa a ação selecionada e observa o próximo estado
 - Executa aprendizado no “bottom level” e no “top level”

■ No “bottom level”

- IDNs: Implicit Decision Networks
- Estado corrente é representado por pares Dimensão-Valor
 - $(dim_1, val_1) (dim_2, val_2) \dots (dim_n, val_n)$
 - Cada par corresponde a um nó de entrada da rede
- Três tipos de entradas
 - Sensory Input (visual, auditory,, etc.)
 - Working Memory Items
 - Goal Structure Item
- Ações são representadas como nós no layer de saída
 - Três tipos de ações:
 - WorkingMemoryActions, GoalActions, ExternalActions
- Cada ação consiste de um ou mais ActionDimensions
 - $(dim_1, val_1) (dim_2, val_2) \dots (dim_n, val_n)$

- O nível de ativação de um nó (em uma Implicit Decision Network) é calculado segundo a seguinte função de ativação

$$o = \frac{1}{1 + e^{-\sum_{i=0}^n w_i x_i}}$$

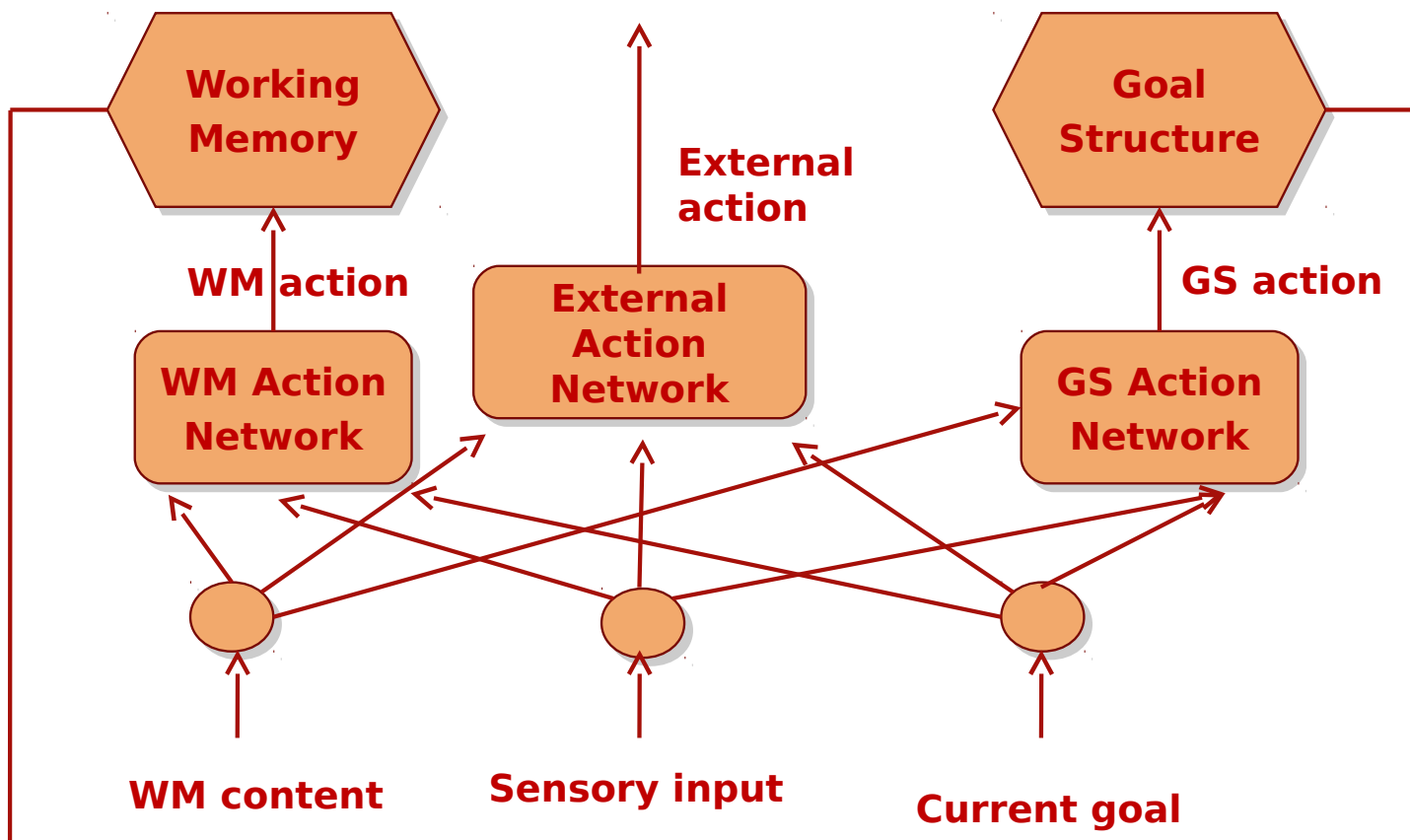
- Onde x_i é o valor da entrada i (do nó), w_i é o peso da entrada i e n é o número de entradas do nó

- Uma ação é escolhida baseada na distribuição de Boltzman de ativações dos nós do layer de saída
- A probabilidade de se selecionar uma ação particular i no “bottom level” é (de acordo com a distribuição de Boltzman):

$$p(i|x) = \frac{e^{A_i/t}}{\sum_j e^{A_j/t}}$$

- Onde A_i é a ativação da ação i e t é o parâmetro de ruído (temperatura)

Escolha da Ação de Saída no “Bottom Level”



■ No “Top Level”

- Regras explícitas
 - Pares Condição → Ação
 - Chunks
- Podem vir de diversas fontes
 - Regras Extraídas e Refinadas (RER Rules)
 - Regras Aprendidas de Maneira Independente (IRL Rules)
 - Regras Fixas (FR Rules)
- Chunks são coleções de pares Dimensão-Valor que representam ou condições ou ações no “top level”
 - Chunk-id_{*i*}: $(dim_{i1}, val_{i1}) (dim_{i2}, val_{i2}) \dots (dim_{in_i}, val_{in_i})$
e.g., table-1: $(size, large) (color, white) (number-of-legs, 4)$
- Cada chunk é representado por um nó no “top level”
- Cada par Dimensão-Valor é representado por um nó no “bottom level”

- Uma regra “top level” contém uma condição e uma ação (possivelmente com múltiplas dimensões)
- A ação está associada com os seguintes fatores:
 - Base-Level Activation (BLA)
 - Mede o quão recente uma regra foi utilizada (necessidade de uso)
 - Utilizada para determinar RTs (Response Times) e destacar regras pertinentes
 - Ativação de decaimento gradual
 - Utilidade (U)
 - Mede a utilidade de uma regra, baseada no custo/benefício da mesma (suporte de uma regra)
 - Utilizada para selecionar as regras
- Seleção da Regra no “top level”
 - Baseada em uma distribuição de Boltzman da utilidade da regra

■ Aprendizagem

- No “bottom level”
 - Utiliza Backpropagation para realizar a correção do erro na IDN
 - Três métodos de aprendizagem:
 - Standard Backpropagation
 - Q-Learning (reinforcement learning)
 - Simplified Q-Learning
- No “top level”
 - Três métodos:
 - Bottom-up rule extraction and refinement (RER)
 - Pares condição → ação são extraídos do “bottom level” e refinados
 - Independent Rule Learning (IRL)
 - Regras de várias formas são geradas de maneira independente e então refinadas ou deletadas
 - Fixed Rule (FR)
 - Regras são obtidas de experiências passadas, ou fornecidas por fontes externas

■ Integração entre níveis

- Diversos métodos de integração:
 - Seleção Estocástica
 - Combinação
 - Retificação Bottom-Up
 - Orientação Top-Down (TD-Guidance)
- Assumem uma rede no “bottom level” e um grupo de regras no “top level”
- Conclusões positivas atingidas no “top level” podem inserir recomendações de ação no “bottom level”
- Conclusões negativas atingidas no “top level” podem vetar ações no “bottom level”

■ Retificação Bottom-Up

- Saída do “bottom level” é enviada para o “top-level”
- “Top level” retifica e utiliza a saída do “bottom level” em conjunto com o conhecimento do “top level”
 - e.g. combinação por soma ponderada

■ Guidance “Top-Down”

- Saída do “top level” é enviada para “bottom level”
- O “bottom level utiliza a saída do “top level”, junto com seu próprio conhecimento, de forma a decidir a ação final

■ Working Memory

- Armazenamento de informações para uso temporário
- Prover suporte aos mecanismos de inferência e tomada de decisão

■ Uso da memória envolve

- Codificação deliberada da informação
- Esvanecimento gradual da informação no tempo
- Atualização temporal da informação
- Capacidade limitada de armazenamento
- Múltiplos locais de armazenagem
 - Seções específicas dependendo dos sensores
- Acesso à memória pode ser feito por ambos os níveis

Working Memory

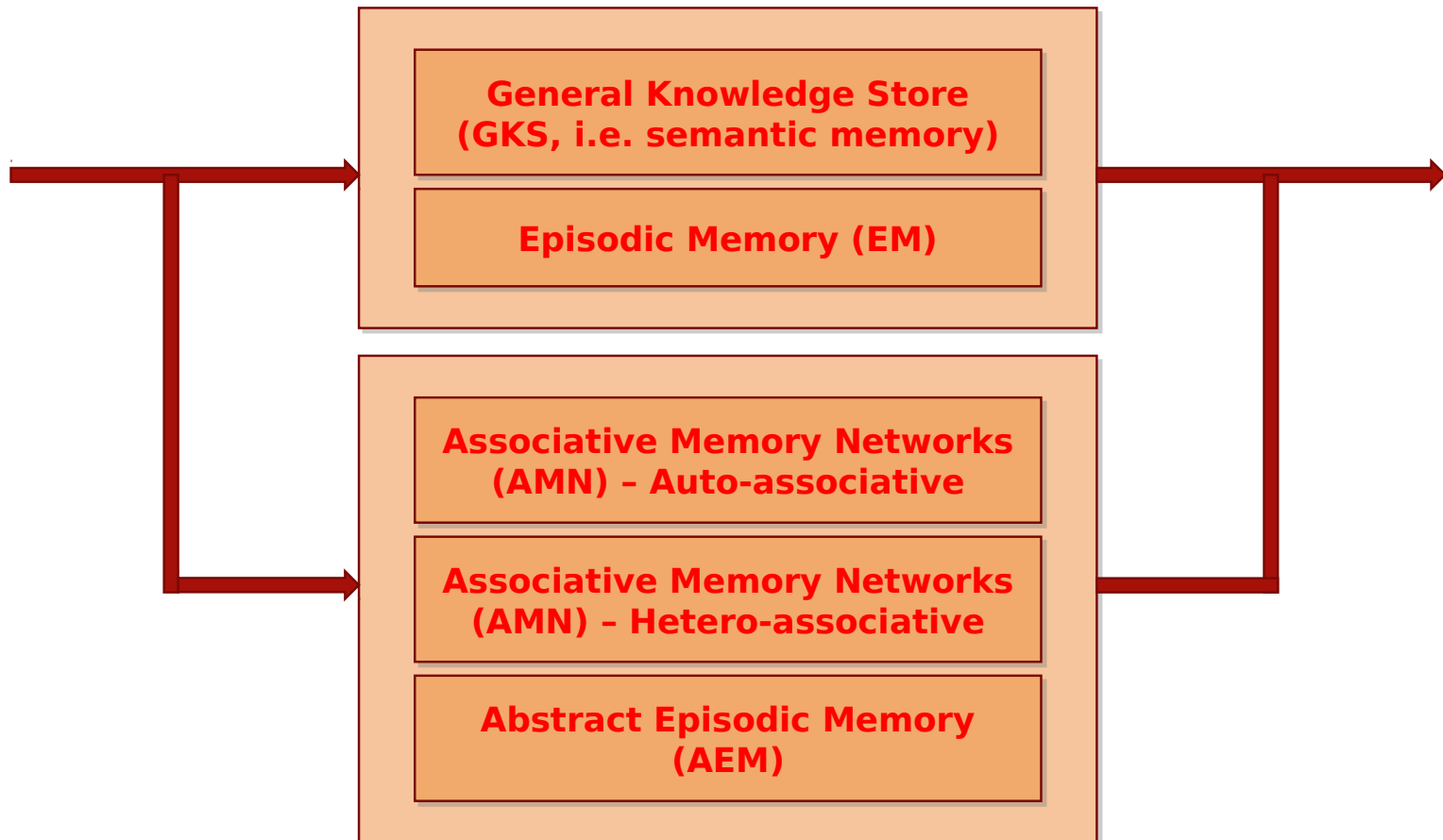
- Working Memory e o NACS
 - WM pode ser utilizada para transmitir informações entre os subsistemas ACS e o NACS
 - Uso mínimo pelo NACS
 - Informações declarativas adicionais sobre o estado corrente e ação corrente (incluindo-se episódios passados relacionados, que são armazenados no NACS)

■ Non-Action Centered Subsystem

- Representação equivalente e semelhante à ACS
- Consequentes das regras não são Actions, mas sim Chunks
- Processamento iterativo e potencialmente bi-direcional

■ Representação

- Conhecimentos gerais: Memória Semântica
- Conhecimentos sobre experiências específicas do mundo: Memória Episódica
- Vários tipos de buscas e inferências com esse conhecimento
- Controlado pelo ACS, por meio de suas ações



GKS: General Knowledge Store

■ Nível “Top”

- Codifica conhecimento explícito
- Chunks: armazenam conceitos - co-ocorrência de micro-características (do nível “bottom”)
 - `Chunk-idi (dimi1, vali1) (dimi2, vali2) ... (dimin, valin)`
 - `table-1 (type, table) (size, large) (color, white) (number-of-legs, 4)`
- Cada chunk é representado simultaneamente por um nó no nível “top” e uma micro-característica no nível “bottom”
- Links entre chunks armazenam associações explícitas entre chunks (unidirecionais ou bi-direcionais)
 - Condições: um ou mais chunks
 - Conclusão: um único chunk
- Ativação de Chunks
 - Oriunda de uma entrada, regra associativa do nível “top” ou por raciocínio baseado em similaridade da interação top-bottom

AMN: Associative Memory Networks

■ Nível “bottom”

- Conhecimento implícito (redes neurais)
- Cada chunk no nível “top” é representado por uma micro-característica no nível “bottom”
- Ativação Bottom-up
 - Por mapeamentos associativos ou processos baseados em similaridade
- Várias possibilidades em capturar associações implícitas
 - Auto-associativas (Redes de Hopfield)
 - Hetero-associativas (Redes MLP com BP)
- Ativação
 - Top-down ou
 - Bottom-up

NACS - Raciocínio

- Começa com uma ação do ACS
 - Ativações bottom-up ou top-down podem então acontecer
- No nível “bottom”
 - Começa com os nós de micro-características ativados, ativando outras micro-características
- No nível “top”
 - Inferências começam a partir de todos os chunks ativados, aplicando-se as regras simultaneamente
 - Novos chunks são inferidos
- Integração entre níveis
 - Ativação bottom-up

■ Tipos de Raciocínio

- Forward Chaining
- Forward Chaining baseado em Similaridade
 - Threshold determina se uma conclusão é aceitável ou não

■ Coordenação entre NACS e ACS

- Usualmente o NACS é controlado pelo ACS
 - Uma ação ACS comanda um passo de raciocínio no NACS
- A saída do NACS é enviada novamente para o ACS

EM e AEM: Memória Episódica

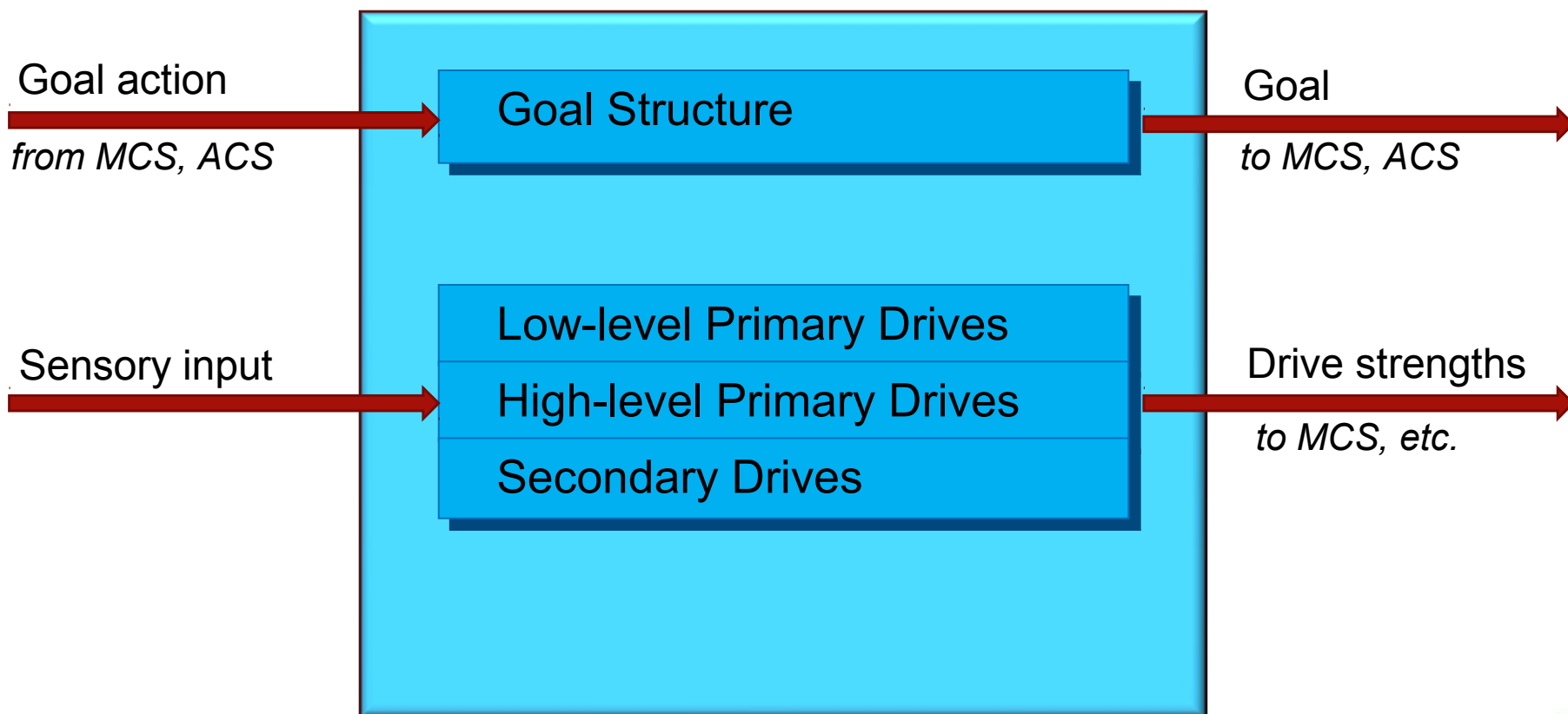
■ EM: Memória Episódica

- Parte especial da GKS
- Experiências orientadas a ações
 - Estímulos, respostas e consequências + TimeStamps
- Experiências não-orientada a ações
 - Chunks e regras associativas + TimeStamps

■ AEM: Abstract Episodic Memory

- Informação sumarizada dos episódios da EM
- AFN: Action Frequency Network
 - Mapeia os estados na distribuição de frequência das ações
- SFN: State Frequency Network
 - Mapeia os estados/ações na distribuição de frequência dos estados sucessivos, bem como de reforços

MS: 0 Subsistema Motivacional



MS: O Subsistema Motivacional

- **Motivação em Sistemas Inteligentes**
 - **Sustentabilidade:** Um agente precisa atender a suas necessidades básicas, tais como fome e sede, evitar perigos, etc.
 - **Propósito:** A ação de um agente deve ser orientada a um propósito, e não aleatória. O propósito do agente está relacionado com sua sustentabilidade
 - **Foco:** Um agente deve ser capaz de focar em diferentes propósitos em diferentes instantes de tempo, e eventualmente mudar seu foco, momentaneamente ou de maneira permanente
 - **Adaptabilidade:** Um agente deve ser capaz de adaptar seu comportamento (ou seja, aprender), para melhorar seu desempenho

MS: O Subsistema Motivacional

- Como representar
 - Drives, necessidades, desejos, propósitos ?
 - Representações explícitas ou implícitas ?
- Subsistema Motivacional
 - Permite a modelagem de drives e sua interação
 - Provê o contexto e as metas para o ACS
 - Representação Dual: Explícita x Implícita
- Representação Explícita
 - Metas específicas do agente
 - Baseadas em estados de drives internos (oriundos do MCS)

MS: O Subsistema Motivacional

■ Especificações

- **Ativação Proporcional:** a ativação de um drive deve ser proporcional a existência (ou ausência) de recursos
- **Oportunismo:** o agente deve considerar a oportunidade de se mudar o comportamento diante do surgimento de um recurso
- **Contiguidade das ações:** o agente não deve ficar mudando de objetivo o tempo todo.
- **Persistência:** após satisfazer um drive, o agente deve persistir durante um tempo, até um certo nível de satisfação
- **Interrupção em urgências:** drives mais urgentes podem interromper outros de menor prioridade
- **Combinação de Preferências:** Satisfazer diversos drives simultaneamente, com menos efetividade deve ser preferível a satisfazer um único drive com maior efetividade

MS: O Subsistema Motivacional

■ Drives Primários

- **Baixo nível:** Buscar comida, buscar bebida, evitar perigo, dormir, reproduzir, evitar saturação, curiosidade, evitar aborrecimento
- **Alto nível:** Buscar proximidade com outros iguais, auto-estima, desejo de evolução

■ Drives Secundários

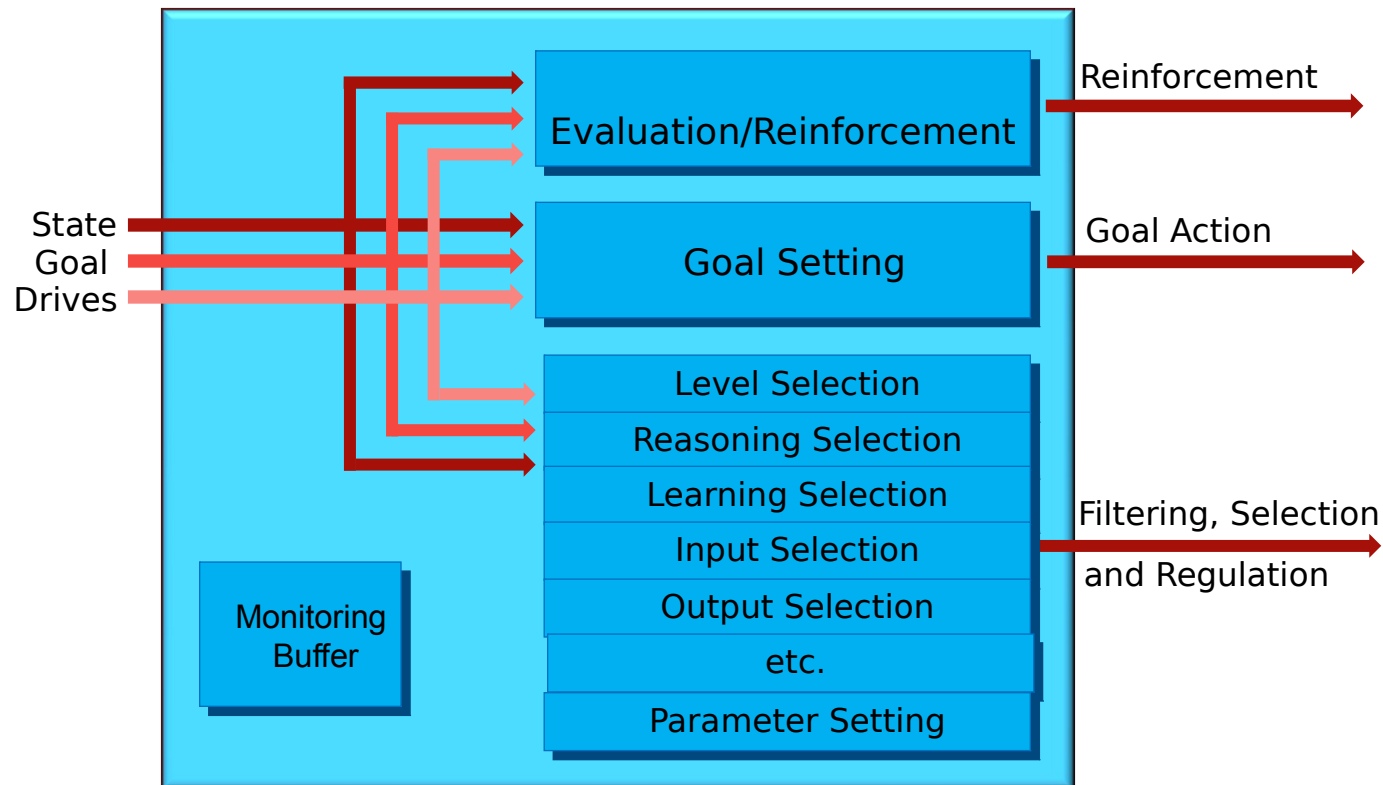
- Derivados dos drives primários
- Condicionamento, a partir de drives primários
- Drives assumidos de outros agentes, por meio de instruções ou o desejo de agradar outros agentes

MS: O Subsistema Motivacional

■ Goal Structure

- Múltiplos drives podem estar ativados ao mesmo tempo, mas apenas uma meta pode ser perseguida a cada instante
- Ações escolhidas no ACS dependem do estado corrente e da meta corrente
- Goal Chunk
 - Dimensão e parâmetros da dimensão
- Organizados em Estruturas
 - Lista de Metas (Goal List)
 - Posição do Goal não é importante
 - Pilha de Metas (Goal Stack)
 - Somente o Goal no topo do stack é acessado
 - Novos Goals podem ser inseridos ou removidos do topo da pilha

MCS: O Subsistema Metacognitivo



MCS: O Subsistema Metacognitivo

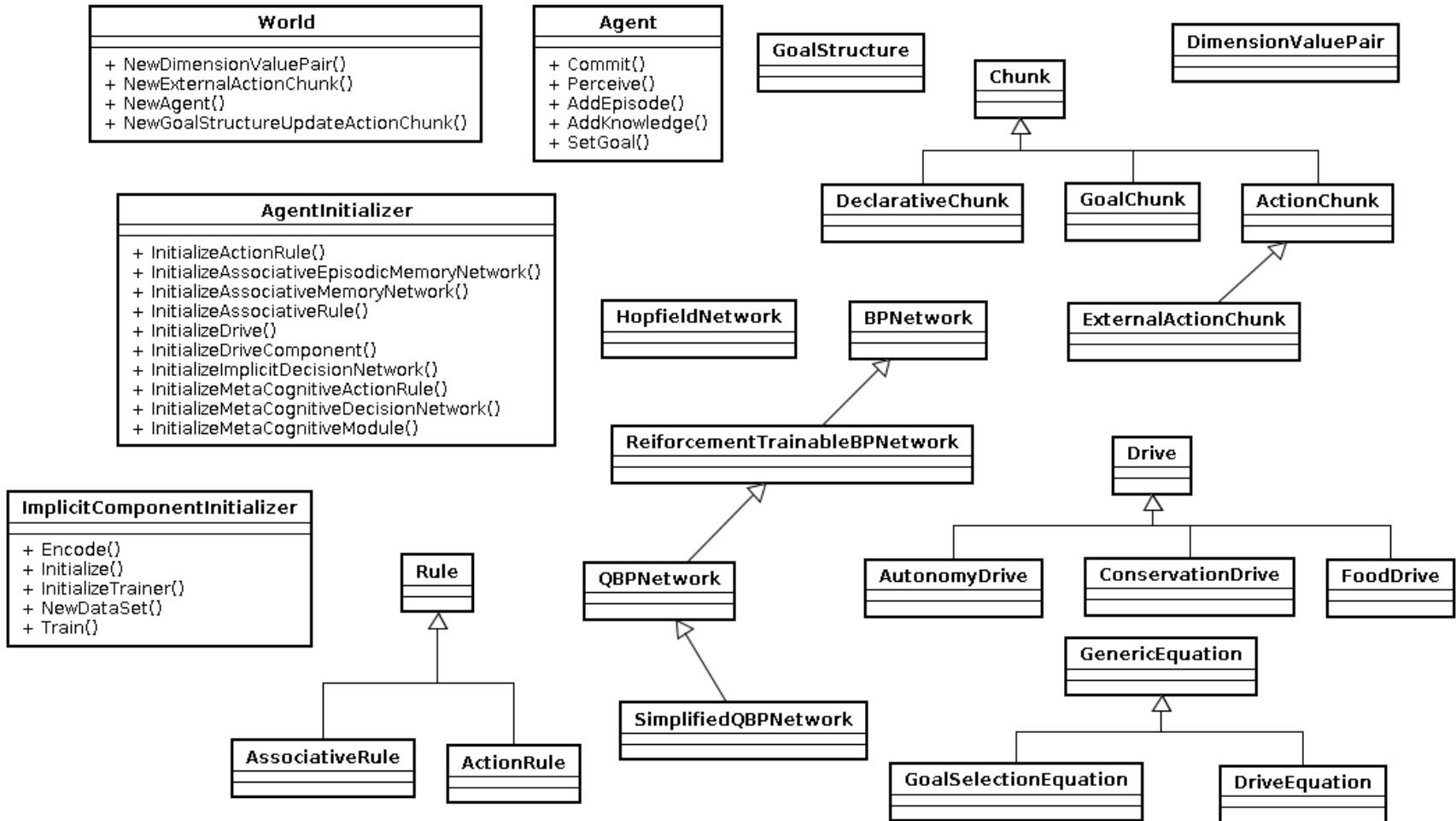
■ Meta-Cognição

- Conhecimento do próprio processamento cognitivo
- Monitoramento e regulação de parâmetros dos processos cognitivos
- Usualmente em serviço de algum objetivo concreto
- Equivalente a um ACS, mas operando sobre variáveis internas

■ Módulos

- Set-up de Goals, reforço, filtragem, aprendizagem e raciocínio, seleção de nível, set-up de parâmetros, monitoramento de desempenho, e outros

Classes Principais do Framework



O Framework de Software

- A partir do final 2011 ocorreu uma reformulação na arquitetura e atualmente encontra-se na versão 6.1.0.7 (Beta)
- Totalmente reescrita em C# utilizando .NET Framework
 - Segundo os próprios autores: “we found it much easier to build all of the new features and capabilities for version 6.1 of the CLARION Library in C#. This is owing to several language constructs in which Java is either lacking or (in our opinion) less proficient.”
- Para o desenvolvimento em Linux, existe o projeto Mono:
<http://monodevelop.com>

O Framework de Software

- A arquitetura nada mais é que uma biblioteca (arquivo .dll) que provê um conjunto de classes e métodos para a sintetização de comportamentos cognitivos em agentes artificiais.
- Para a sua utilização, basta criar um típico projeto em C# (Visual Studio ou Mono) e adicionar essa biblioteca como referência ao projeto.
- Finalmente, basta prover à arquitetura um conjunto de entradas sensorias de modo que ao final de um ciclo de cognição, esta consiga especificar uma ação para o agente.

Exemplo em C#

```
// Input Sensorial Types
```

```
DimensionValuePair hi =  
World.NewDimensionValuePair("Salutation", "Hello");  
DimensionValuePair bye =  
World.NewDimensionValuePair("Salutation", "Goodbye");
```

```
// Output Types
```

```
ExternalActionChunk sayHi =  
World.NewExternalActionChunk("Hello");  
ExternalActionChunk sayBye =  
World.NewExternalActionChunk("Goodbye");
```

```
// Initialize Agent
```

```
Agent John = World.NewAgent("John");
```

Exemplo em C#

```
// Create Decision Action Net
SimplifiedQBPNetwork net =
AgentInitializer.InitializeImplicitDecisionNetwork(John,
SimplifiedQBPNetwork.Factory);

net.Input.Add(hi);
net.Input.Add(bye);
net.Output.Add(sayHi);
net.Output.Add(sayBye);

// Associate net to the agent
John.Commit(net);
```

Exemplo em C#

```
■ // Input Sensorial Signal
  si.Add(hi, John.Parameters.MAX_ACTIVATION);

// Perceive the sensory information
John.Perceive(si);

// Choose an action
chosen = John.GetChosenExternalAction(si);

// Give positive feedback.
John.ReceiveFeedback(si, 1.0);
```

Maiores Informações

- R. Sun, A Detailed Specification of CLARION 5.0 - Technical Report (2003)
 - <http://www.sts.rpi.edu/~rsun/sun.tutorial.pdf>
- Addendum 1: The enhanced description of the motivational subsystem
 - <http://www.sts.rpi.edu/~rsun/folder-files/sun-new-MS.pdf>
- Addendum 2: The enhanced description of similarity-based reasoning.
 - <http://www.sts.rpi.edu/~rsun/folder-files/SH-SBR.pdf>
- Addendum 3: The properties of the CLARION-H implementation.
 - <http://www.sts.rpi.edu/~rsun/folder-files/SH-CLARION-H.pdf>