

# ARQUITETURAS COGNITIVAS PARA CRIATURAS ARTIFICIAIS

RICARDO RIBEIRO GUDWIN\*

\*DCA-FEEC-UNICAMP

Av. Albert Einstein 400

Campinas, SP, Brasil

Email: [gudwin@dca.fee.unicamp.br](mailto:gudwin@dca.fee.unicamp.br)

**Abstract**— This work presents a tutorial introduction to cognitive architectures while applied to the control of artificial creatures. We start by presenting the notion of artificial creature, followed by the introduction of the notion of cognitive architecture. In the sequence, we show some works describing cognitive architectures applied to artificial creatures, and conclude with an overview on two of the most famous cognitive architectures: ACT-R and SOAR.

**Keywords**— Artificial Life, Artificial Creatures, Cognitive Architectures

**Resumo**— Neste trabalho, apresentamos um tutorial introdutório sobre arquiteturas cognitivas aplicadas ao controle de criaturas artificiais. Iniciamos pela apresentação do conceito de criatura artificial, seguida da introdução da noção de arquitetura cognitiva. Na sequência, mostramos alguns trabalhos colhidos na literatura, onde arquiteturas cognitivas são aplicadas a criaturas artificiais, concluindo com uma descrição geral de duas das mais famosas arquiteturas cognitivas: ACT-R e SOAR.

**Palavras-chave**— Vida Artificial, Criatura Artificial, Arquitetura Cognitiva

## 1 Introdução

A área de pesquisa em *Vida Artificial* é bastante ampla, abrangendo diferentes facetas sobre o estudo abstrato dos processos naturais que poderíamos chamar de *vida*. Se por um lado, temos abordagens do tipo *bottom-up*, que tentam criar processos emergentes que poderíamos identificar com o conceito de vida (Langton, 1986), existem abordagens do tipo *top-down* que buscam se inspirar em criaturas vivas para modelar sistemas artificiais onde esta “vida artificial” possa ser sintetizada. Seguindo essa abordagem *top-down*, toda uma área de pesquisas em Animats, foi estabelecida (Dean, 1998). O conceito de *animat* pode incluir tanto robôs físicos como simulações virtuais. Esse conceito surgiu em 1990 com a primeira conferência sobre *Simulation of Adaptive Behavior* (Meyer and Wilson, 1990). Um termo correlato a *animat*, mas um pouco mais genérico é o de *Criatura Artificial*. Podemos entender uma criatura artificial como um tipo de *agente*, onde este agente se encontra “corporificado”, ou “incorporado”, ou seja, possui um corpo. Este corpo pode ser real, como no caso de um robô, ou pode ser virtual, como no caso de um NPC (*non-player character*) de um jogo de computador. Desta forma, um animat é uma criatura artificial, e é também um agente, embora possam existir agentes que não sejam criaturas artificiais. Embora os termos animat e criatura artificial sejam as vezes utilizados de maneira intercambiável, entendemos que o termo criatura artificial é um pouco mais genérico, podendo existir criaturas artificiais que talvez não pudessem ser classificadas como animats. Sendo assim, neste trabalho nos referiremos sempre ao termo “criatura artificial”, mesmo que em alguns

artigos da literatura o termo utilizado seja animat, e em outros artigos utilize-se ainda o termo “agente”, ou “agente inteligente” ou ainda “criatura virtual”.

Diferentes estratégias podem ser utilizadas para controlar uma criatura artificial. As estratégias que nos interessam, em particular, vêm buscar inspiração na faculdade da cognição humana (as vezes cognição animal) para estabelecer arquiteturas de controle que são chamadas de maneira genérica de *arquiteturas cognitivas*. Nestas estratégias, diversos conceitos relacionados à mente humana são utilizados para fundamentar o funcionamento da arquitetura, tais como por exemplo: percepção, aprendizagem, emoções, memória, linguagem, seleção de ação, comportamento, raciocínio, tomada de decisão, consciência, etc. Embora não exista uma fronteira canônica para separar o que seria uma arquitetura cognitiva de uma arquitetura não-cognitiva, ou seja, não existe um consenso na literatura sobre o que constituiria uma arquitetura “cognitiva”, e o que esse adjetivo (cognitiva) traria de fato à arquitetura, existe algo de comum em todas as arquiteturas assim chamadas de cognitivas, que é o apelo à inspiração da cognição humana norteando o funcionamento da arquitetura.

As pesquisas em arquiteturas cognitivas estão lastreadas em alguns princípios oriundos da ciência cognitiva, tais como o conceito de cognição situada e incorporada (ou incorporada - *embodied situated cognition*) (Varela et al., 1999). De acordo com este conceito, o fenômeno da cognição só pode ocorrer caso o agente esteja situado, ou seja, localizado em algum lugar do ambiente, onde pode sensoriar e atuar. Para estar situado, o agente necessita de um corpo. Esse corpo pode

ser real ou virtual, dependendo-se do ambiente em si ser real ou virtual.

Uma terminologia que também é frequentemente encontrada na literatura diz respeito a um *avatar* localizado em um ambiente. O avatar nada mais é do que o corpo da criatura artificial em seu ambiente. O importante aqui é ressaltar que, para que o processo de cognição possa ocorrer, existe a necessidade de um corpo, mesmo que esse corpo seja virtual. Esse corpo é que permite a situabilidade do agente no ambiente, o que parece ser fundamental para o processo de cognição.

Neste trabalho, fazemos um *survey* apresentando os principais trabalhos encontrados na literatura envolvendo o uso de arquiteturas cognitivas para o controle de criaturas artificiais. Iniciamos apresentando o conceito de criatura artificial, apresentando historicamente alguns exemplos encontrados na literatura. Em seguida, apresentamos o conceito de arquitetura cognitiva, e também alguns exemplos encontrados na literatura onde diversas arquiteturas cognitivas são propostas como sistemas de controle para criaturas artificiais. Por fim, damos uma ênfase em duas arquiteturas cognitivas genéricas, o ACT-R e o SOAR, bastante populares, com aplicações em diversas situações e também utilizadas para o controle de criaturas artificiais.

## 2 Criaturas Artificiais

Como já citamos anteriormente na introdução, o conceito de *criaturas artificiais* surge dentro do escopo das pesquisas em *vida artificial*, seguindo-se uma abordagem *top-down* de investigação. Os trabalhos encontrados na literatura variam desde a simulação de criaturas muito simples (Sims, 1994b; Yaeger, 1994) até humanos virtuais (Badler, 1997; Hill, 1999).

Os experimentos com criaturas artificiais envolvem tanto casos em que o ambiente onde a criatura atua é o mundo real (criaturas robóticas) como casos onde esse ambiente é simulado computacionalmente (criaturas virtuais). Diversos autores apontam vantagens e desvantagens em se trabalhar com criaturas reais ou criaturas virtuais.

Quando as criaturas são virtuais, apontam-se algumas vantagens. Em primeiro lugar, os pesquisadores ressaltam que é mais fácil manter o foco e os objetivos nas tarefas em que se deseja estudar. Em robôs, diversos problemas tais como ruídos, problemas de quantização, problemas com sensores e atuadores acabam desviando a atenção do que seria o foco principal, ou seja, a modelagem de como o processo cognitivo é utilizado para tomar decisões inteligentes. Da mesma forma, o ambiente se encontra mais sob controle do experimentador, que pode facilmente criar novos níveis de complexidade de maneira incremental, à medida em que a arquitetura vai supe-

rando níveis de dificuldade crescentes, coisa que é mais difícil de controlar em um experimento com um robô real. Como desvantagem, observa-se que em alguns casos, torna-se difícil se portar o experimento para um ambiente real, uma vez que justamente em uma criatura virtual questões como ruídos, problemas de quantização, desvios em medições e outras características importantes do mundo real não são usualmente consideradas. Desta maneira, alguns pesquisadores que trabalham com robôs reais consideram os trabalhos simulados como “toy-problems”, ou seja, consideram os trabalhos com criaturas virtuais como trabalhos de segunda classe, menos importantes. Nossa posição, no entanto, em relação a esta questão não é tão radical. É necessário compreendermos que os objetivos envolvidos em trabalhos com robôs reais e com criaturas virtuais são normalmente diferentes. Enquanto no caso de robôs reais normalmente almeja-se construir de fato robôs que possam servir a algum propósito no mundo real, trabalhos com criaturas virtuais estão mais interessados em desenvolver aspectos mais voltados para algoritmos de tomada de decisão inteligente, que possam gerar um comportamento na criatura virtual que seja mais inteligente (Balkenius, 1995). Dessa forma, não se pode falar aqui de pesquisas mais ou menos importantes, pois ambos os tipos de pesquisas são importantes, uma vez que se prestam a objetivos diferentes.

Para desenvolver uma pesquisa com criaturas artificiais, diversos cuidados necessitam ser tomados. Em primeiro lugar, é necessário definirmos se o trabalho envolverá criaturas robóticas ou criaturas virtuais. Caso opte-se por criaturas virtuais, será necessário definir o escopo do “mundo” virtual, o que envolve normalmente definir se esse mundo será discreto ou contínuo, 2D ou 3D, se esse mundo possui uma “física” ou se é um mundo estático, com objetos simplesmente posicionados em pontos estratégicos. Necessita-se ainda definir os tipos de objetos, se esses objetos serão móveis ou imóveis, se permitirão algum tipo de interatividade ou se são simplesmente obstáculos ao movimento da criatura. Caso se opte por criaturas robóticas, é necessário definir o modelo e capacidades do robô, quais os tipos de objetos que o robô poderá encontrar ou manipular, e que tipo de manipulação ele poderá ter com esses objetos. Tanto no caso de criaturas virtuais como criaturas robóticas, será necessário definir se os experimentos serão feitos com uma única criatura artificial ou com múltiplas criaturas em um mesmo ambiente, e definir também como será a interação entre elas, caso hajam mais de uma criatura operando no ambiente. Também, de um modo geral, é necessário definir-se os sensores e atuadores das criaturas, a dinâmica desses sensores e atuadores, e as tarefas e comportamentos que se esperam das criaturas no ambiente. Tarefas típicas de pesquisas nessa área



Figura 1: Exemplo do Robô Khepera com seus vários “turrets”



Figura 2: Exemplo do Robô Khepera em operação

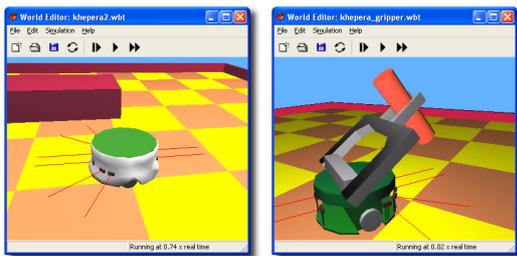


Figura 3: A Plataforma Webot de Simulação de Robôs Khepera

envolvem a navegação segura pelo ambiente, evitando colisões, buscando metas, seguindo paredes e mapeando o ambiente. Em alguns casos, pode-se estudar comportamentos coletivos, tais como movimentação em filas ou grupos, a formação de padrões de movimento. Algumas pesquisas envolvem também o desenvolvimento de um comportamento estratégico, como por exemplo os experimentos com futebol de robôs (Kitano, 1998).

### 2.1 Criaturas Robóticas: Khepera e Webot

Um exemplo de criatura artificial robótica bastante usada em diversos experimentos é o robô Khepera, ilustrado na figuras 1 e 2

O robô Khepera é um exemplo onde de plataforma onde podem ser executados tanto experimentos simulados (virtuais) como reais. A figura 3 mostra o ambiente de simulação Webot, onde pode-se simular experimentos com robôs Khepera. Os mesmos algoritmos de controle podem ser utili-

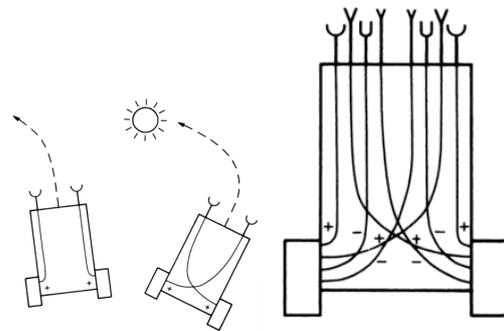


Figura 4: Exemplo de Criaturas de Braitenberg

zados para controlar o robô simulado como o robô real. Desta forma, os experimentos podem ser conduzidos inicialmente somente no simulador, e quando os primeiros resultados começarem a aparecer, os experimentos podem ser redirecionados para o robô real. Da mesma forma o uso do simulador pode permitir o desenvolvimento de pesquisas, evitando o custo da aquisição dos robôs reais.

### 2.2 Criaturas de Braitenberg

Um trabalho bastante importante, por ter sido um dos trabalhos pioneiros em criaturas artificiais foi o de (Braitenberg, 1986), que em 1986 publicou seus experimentos com criaturas artificiais estereotipadas na forma de “veículos”. Neste trabalho, ele apresenta 14 tipos de criaturas artificiais, chamadas por ele de “veículos conceituais”, descritos por comportamentos bastante simplificados, mas com um comportamento emergente relativamente sofisticado.

Um exemplo de criaturas de Braitenberg pode ser visualizado na figura 4

Das 14 criaturas descritas por Braitenberg, 12 foram implementadas em Lego por Hogg, Martin & Resnick, conforme (Hogg et al., 1991). Cinco dessas criaturas são bastante simples, possuindo apenas 1 sensor. De acordo com seu comportamento, elas foram batizadas com os seguintes nomes: Tímido, Indeciso, Paranóide, Dogged, Inseguro e Driven. Quatro das criaturas já possuem um comportamento mais complexo, recebendo os nomes de Persistente, Atrativo e Repulsivo, Consistente e Inumano. Por fim, duas criaturas foram consideradas Criaturas “Filosóficas”: Frantic e Observador.

### 2.3 Criaturas Virtuais de Karl Sims

Um trabalho que teve bastante repercussão, sendo citado muitas vezes na comunidade é o trabalho de Karl Sims (Sims, 1994b; Sims, 1994a). Em seu trabalho Sims concebeu criaturas com corpos bastante simples, formados por partes tridimensionais rígidas e acopladas, organizadas estrutural-

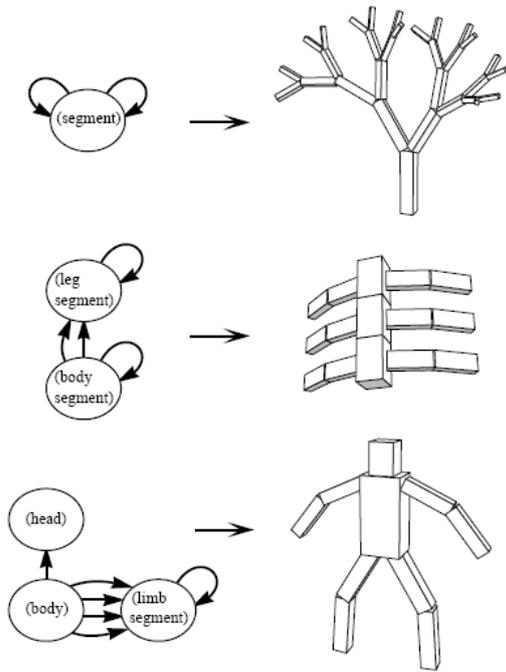


Figura 5: Modelo de Corpo para uma Criatura de Sims

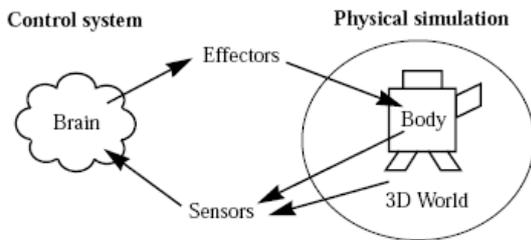


Figura 6: Interação Corpo-Mente, em uma Criatura de Sims

mente de acordo com um cromossomo que define uma máquina de estados. Um exemplo de como o corpo de uma criatura pode ser gerado a partir de uma máquina de estados pode ser visto na figura 5

Os pontos onde as diferentes partes se conectam podem ser articulados, sendo controlados por uma “mente” que define como essas articulações se movimentam, conforme a figura 6.

As criaturas de Sims possuem diversos tipos de sensores, tais como sensores de ângulos das juntas, sensores de contato e sensores de luz. O controle das criaturas é feito segundo uma rede neural definida também por um cromossomo que evolui em um algoritmo genético, onde cada neurônio pode ser de vários tipos: `sum`, `product`, `divide`, `sum-threshold`, `greater-than`, `sign-of`, `min`, `max`, `abs`, `if`, `interpolate`, `sin`, `cos`, `atan`, `log`, `expt`, `sigmoid`, `integrate`, `differentiate`, `smooth`, `memory`, `oscillate-wave` e `oscillate-saw`. Os atuadores das criaturas geram o torque nas juntas. Um exemplo de uma rede neural ge-

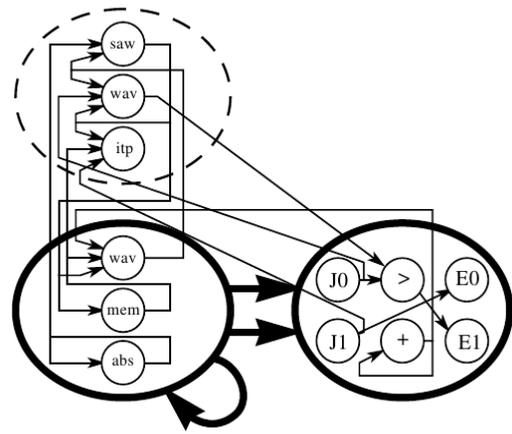


Figura 7: Exemplo de Sistema de Controle de uma Criatura de Sims

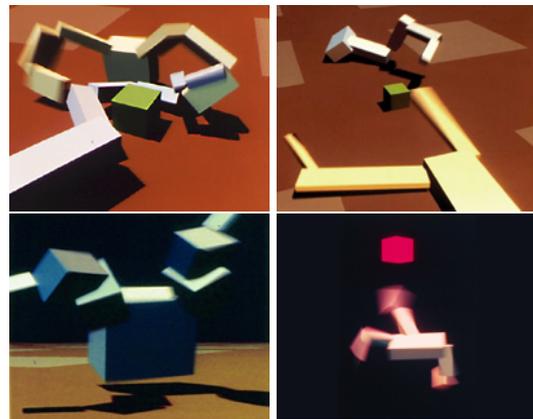


Figura 8: Exemplos de Criaturas Evoluídas nas Simulações de Sims

rada durante a evolução do sistema pode ser vista na figura 7

Dessa forma, as criaturas de Sims têm sua evolução controlada por um algoritmo genético. Esse algoritmo proporciona uma co-evolução, tanto da morfologia (corpo) das criaturas como de seu controle (mente). Sims detectou o surgimento de diversos comportamentos interessantes durante a evolução de suas criaturas. Em ambientes simulando um meio líquido, ele conseguiu observar criaturas capazes de nadar. Em simulações em meio sólido, suas criaturas aprenderam a andar, saltar e buscar a luz. Sims simulou ainda a competição entre diversos indivíduos, maximizando seu fitness em um determinado número de gerações. Ele desenvolveu simulações onde cada criatura lutava por si própria e simulações onde as criaturas desenvolviam um comportamento social, cooperando com algumas e competindo com outras.

Alguns exemplos de criaturas evoluídas pelas simulações de Sims podem ser vistas na figura 8.

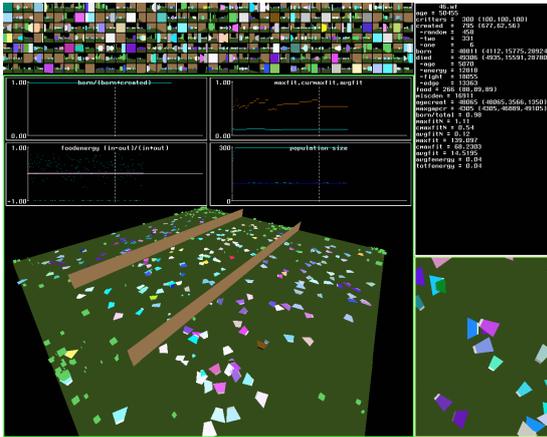


Figura 9: O Ambiente “PolyWorld” de Yaeger

#### 2.4 O PolyWorld de Yaeger

Outro exemplo de criaturas artificiais que teve grande repercussão na comunidade foi o projeto “PolyWorld” de Yaeger, conforme apresentado originalmente em (Yaeger, 1994). Neste artigo, Yaeger propõe a construção de uma ferramenta que segundo ele seria adequada para a investigação de questões relevantes em biologia evolutiva, ecologia comportamental, etologia e neurofisiologia. O objetivo de Yaeger era a construção de uma “vida artificial semelhante à vida real”. A figura 9 visualiza o mundo virtual construído por ele.

As criaturas evoluídas por Yaeger eram capazes de vários comportamentos, que Yaeger batizou de *eating*, *mating*, *fighting*, *moving*, *turning*, *focusing* (controlar o campo de visão) e *lighting* (controlar o brilho de alguns dos polígonos de seu corpo). De forma semelhante às criaturas de Sims, as criaturas de Yaeger possuem um sistema nervoso constituído por uma rede neural evolutiva com topologia variável. Ao contrário das criaturas de Sims, as criaturas de Yaeger possuem um corpo mais simples, mas em compensação possuem um sistema de controle bem mais sofisticado. As criaturas de Yaeger possuem um sistema perceptivo, com capacidade de visão colorida (retina com um array de 22x22 pontos), com um campo de visão horizontal variável de 20 a 120 graus, e um campo de visão vertical fixo, a 10 graus. Além disso, as criaturas de Yaeger possuem um sistema metabólico responsável pela manutenção da fisiologia da criatura. Esse sistema metabólico é responsável pelo cálculo energético que mantém a criatura viva. Dentre uma série de comportamentos emergentes, Yaeger descreve: busca de alimentos, comportamento de fuga, comportamento de reação quando atacado, seguir outros organismos, etc.



Figura 10: Tela do Jogo “Creatures” de Stephen Grand

#### 2.5 “Creatures” de Stephen Grand

Um outro experimento em criaturas artificiais que teve grande repercussão no meio acadêmico foi o desenvolvimento do jogo “Creatures” de Stephen Grand (Grand et al., 1997; Grand and Cliff, 1998). Ao contrário dos experimentos de Sims e de Yaeger, neste caso havia não só o interesse acadêmico, mas também o interesse comercial em gerar um jogo de computador a ser vendido no mercado. Apesar de ser um projeto comercial, o experimento apresentou um nível de sofisticação bastante impressionante. Uma visualização da tela principal do jogo pode ser encontrada na figura 10.

Neste experimento, as criaturas são chamadas de “norms”, e habitam um mundo 2D onde podem se movimentar e interagir com uma série de objetos além de outros “norms” e outras criaturas. Uma característica da simulação é que cada criatura nasce com pouco conhecimento sobre o mundo a seu redor, a sai aprendendo o funcionamento dos objetos a seu redor por meio de experimentação. Além disso, o usuário pode interagir com a criatura, atraindo sua atenção e aplicando pequenas punições e recompensas, à medida que a criatura faz coisas que o usuário considera “certas” e “erradas”. Uma das características bem interessantes desse experimento é a capacidade de aprendizado de linguagem. Dentre as possibilidades de interação com a criatura, o usuário pode emitir “palavras” em certas situações, que a criatura é capaz de “ouvir”. Dependendo do contexto em que se encontra, a criatura é capaz de associar essas palavras com objetos, ações e situações em seu entorno, e a partir daí, pode-se utilizar essa palavra, por exemplo, para solicitar que a criatura procure por um determinado tipo de objeto. A criatura não vem com nenhuma linguagem previamente conhecida, de forma que qualquer linguagem pode ser ensinada a ela por este método (até mesmo uma linguagem inventada).

O controle da criatura artificial é realizado por

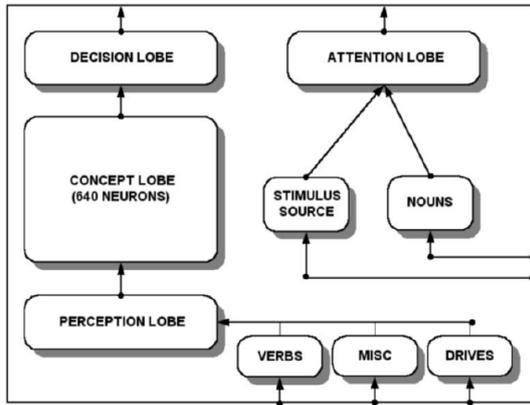


Figura 11: Rede Neural de Controle de um “norn”

meio de uma rede neural com 1000 neurônios divididos em 9 lóbulos. Um diagrama desta rede neural pode ser visualizado na figura 11.

Além desta rede neural, toda uma bioquímica é simulada no experimento. Quando a criatura se alimenta, este alimento é quebrado em compostos químicos, que sofrem reações e podem alterar o metabolismo do “norn”. Desta forma, alguns alimentos podem deixar o “norn” doente ou curá-lo.

### 3 Concebendo uma Criatura Artificial

Uma vez que tenhamos visto alguns exemplos de criaturas artificiais, algumas perguntas imediatamente nos vêm à mente:

- Quais os sub-sistemas necessários para uma criatura artificial completa ?
- Como devem interagir entre si ?

Balkenius, em (Balkenius, 1995) nos auxilia neste entendimento. Segundo Balkenius, três tipos de subsistemas, ou módulos são necessários para controlar uma criatura artificial:

- Módulos Comportamentais
- Módulos de Aprendizagem/Adaptação
- Módulos Motivacionais

Os módulos comportamentais são responsáveis pelo comportamento da criatura. Balkenius descreve uma hierarquia de módulos comportamentais que ele chama de módulos de engajamento. Cada módulo de engajamento controla uma tarefa em particular da criatura.

Os módulos de aprendizagem/adaptação ativam ou inibem diferentes módulos de engajamento, sendo responsáveis pela aprendizagem da criatura. Balkenius descreve diversos tipos de aprendizagem: condicionamento instrumental e condicionamento clássico, aprendizagem por reforço (Q-learning) e aprendizagem por diferenças temporais.

Os módulos motivacionais são responsáveis por gerar os propósitos da criatura. Esses propósitos é que guiarão a escolha de objetivos, que por sua vez interferirão nos módulos comportamentais para gerar o comportamento final da criatura. Balkenius distingue dois tipos diferentes de motivações: drives e incentivos. Drives, ou necessidades internas, segundo (Hull, 1952), definem objetivos internos da criatura, tais como a necessidade de manter o balanço energético, a necessidade de conhecer o ambiente, a necessidade de manter uma integridade estrutural, etc. Drives são geralmente inatos em uma criatura. Incentivos, que podem ser externos ou internos, inatos ou adquiridos, são uma maneira auxiliar por meio do qual uma criatura pode ter motivações. Através de incentivos, que normalmente estão associados a estímulos de prazer e dor, pode-se “treinar” o comportamento de uma criatura artificial, levando-a a incorporar novos tipos de comportamentos.

Baseando-se nestes três tipos básicos de módulos, Balkenius descreve 4 tipos diferentes de meta-comportamentos elementares, direcionados a um objeto ou situação:

- Comportamento Apetitivo
- Comportamento Aversivo
- Comportamento Exploratório
- Comportamento Neutro

No caso do comportamento apetitivo, a criatura sente uma atração por um objeto ou situação, que a faz buscar esse objeto ou a realização de determinada situação. No caso do comportamento aversivo, a criatura sente uma repulsão por um objeto ou situação, o que a faz evitar esse objeto ou a realização de uma determinada situação. No caso do comportamento exploratório, o desconhecimento de informações sobre um objeto ou local, a faz experimentar atuações sobre o objeto ou a faz navegar por locais desconhecidos, de forma a aumentar seu conhecimento. Por fim, um comportamento neutro é aquele que não está relacionado a um objeto ou situação.

Depois que a questão do comportamento de uma criatura está devidamente projetado, com suas possibilidades de aprendizagem/adaptação e motivação, uma outra questão precisa ser tratada, para a definição da arquitetura de controle da criatura. É a questão da percepção e da aprendizagem categorial. Da mesma forma que uma criatura precisa atuar no mundo (ou em si mesma, levando-a a se movimentar sobre o mundo), ela também precisa sensoriar o mundo para descobrir objetos e situações. Entretanto, a quantidade de informações oriunda dos sensores é muito grande, e necessita certamente ser filtrada ou categorizada, de tal forma que se torne útil na determinação do comportamento. A interpretação das entradas

sensoriais e sua categorização em níveis mais abstratos é a tarefa da percepção. Em outras palavras, cabe à percepção reduzir a complexidade do mundo sensoriado por meio de representações ortogonais para padrões sensoriais co-relacionados. Além disso, um sistema perceptivo pode ainda gerar o que chamamos de “expectativas perceptivas”, que podem orientar a capacidade preditiva da criatura. Nesse processo de percepção, a informação é frequentemente multi-modal, vindo de diferentes tipos de sensores, capturando diferentes tipos de grandezas físicas do ambiente.

Este nosso exercício em tentar conceber os módulos necessários para a construção de um sistema de controle para uma criatura artificial, nos leva a perceber diversos pontos. Em primeiro lugar, parece ser claro que a estrutura de um sistema de controle para uma criatura artificial é bastante diferente de um sistema de controle clássico. Estratégias clássicas de controle não nos seriam muito úteis aqui. Em segundo lugar, o que parece ficar patente é que diversos conceitos oriundos da psicologia (incluindo-se aqui a psicologia animal), parecem ser úteis para o desenvolvimento de um sistema de controle como esse. Stan Franklin, em (Franklin, 1997a) nos traz o conceito de “mentes artificiais”, ou seja, sistemas de controle para criaturas artificiais, que de certa forma teriam seu funcionamento inspirado no que se conhece sobre mentes naturais, e os modelos que temos para elas. Nos parece claro, também, que diversos conceitos das ciências cognitivas, tais como percepção, motivação, aprendizagem, etc, parecem ser importantes e relevantes para essa questão. Isso nos leva, portanto, à questão das arquiteturas cognitivas, que serão tratadas na seção a seguir.

#### 4 Arquiteturas Cognitivas

Vimos na seção anterior, que a tentativa de se delinear a concepção de um sistema de controle para uma criatura artificial, ou nos termos de Stan Franklin, uma “mente artificial”, nos leva a considerar modelos de processos cognitivos, como os estudados pelas ciências cognitivas para tentar explicar a mente humana. O exercício perpetrado na seção anterior, na realidade, foi realizado historicamente por toda a comunidade trabalhando na questão do comportamento adaptativo. Essa discussão vem se desenvolvendo desde 1990, com a primeira conferência sobre a simulação do comportamento adaptativo (Meyer and Wilson, 1990), com sucessivas contribuições da comunidade. Por volta do final da década de 90, início do novo milênio, começa a surgir o termo “arquitetura cognitiva” para designar especificamente estruturas e processos essenciais de um modelo cognitivo-computacional de domínio genérico, passível de ser utilizado em múltiplos níveis e múltiplos domínios, referente aos fenômenos

da cognição e do comportamento. Por exemplo, (Sloman, 2000) começa a tocar na questão “arquitetural” para se abordar os conceitos mentais. Sun, em (Sun, 2004) começa a catalogar o que seria necessário ou desejável para uma eventual “arquitetura cognitiva”. Em 2006, ainda (Sun, 2007) discute questões e desafios no desenvolvimento de arquiteturas cognitivas. Mais recentemente, (Langley and Laird, 2009) faz uma análise de algumas assim chamadas arquiteturas cognitivas, e avalia o avanço desta área de pesquisa.

Todos esses trabalhos apontam que a principal vantagem em se caracterizar uma arquitetura computacional como sendo “cognitiva” seria a criação de um framework concreto para uma modelagem mais detalhada do fenômeno cognitivo. Dentre outras preocupações arquiteturais, isso acabou levando à definição de um conjunto de estruturas essenciais, divisão em módulos, relações entre módulos, além de outros aspectos, permitindo uma linguagem comum para que os pesquisadores explorassem suas propostas.

Apesar desse esforço, no entanto, diversas questões permanecem ainda em aberto, com relação às arquiteturas cognitivas. Por exemplo, com relação às pressuposições cognitivas básicas, o que há de comum nas diferentes propostas? O conhecimento, nestas arquiteturas, está representado de maneira implícita ou explícita? É um conhecimento procedural ou declarativo? A arquitetura é automática ou controlada? Com relação à memória, quais os subsistemas de memória essenciais? Com relação às abordagens metodológicas para a elaboração dessas arquiteturas, deve-se utilizar o *fitting* de dados quantitativos ou somente demonstrações qualitativas? Como as arquiteturas devem responder a experimentos mentais filosóficos? Com relação ao desenvolvimento da arquitetura, como descrever os processos componenciais da cognição, de forma a facilitar a modelagem detalhada dos vários componentes e processos que envolvem a mente? Em que ordem eles devem ser elaborados? Será que existe um conjunto inicial de pressuposições que possa ser utilizado para o desenvolvimento de modelos futuros? Com relação à eventual existência de estruturas essenciais, será que existe um conjunto mínimo de estruturas essenciais que é mandatório? As arquiteturas devem ser construídas como modelos completos, ou como modelos desenvolvimentais, onde a partir de uma estrutura inicial, e por meio de algum processo evolutivo, a arquitetura vai evoluindo de maneira autônoma (como parece acontecer com o ser humano adulto)? Quais os objetivos da arquitetura? A interação com o ambiente ou a interação com outras criaturas?

Podemos ainda relacionar diversos requisitos desejáveis para uma arquitetura cognitiva. Por exemplo, podemos especificar que ela tenha um realismo ecológico, ou seja, que não funcione so-

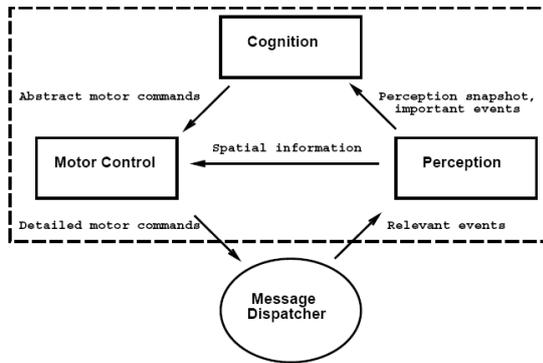


Figura 12: Arquitetura apresentada em (Rickel and Johnson, 1999) - Animated Agents for Procedural Training in Virtual Reality

mente em *toy-problems*, mas suporte as condições do mundo real. Podemos especificar um realismo bio-evolutivo, ou seja, que modele a cognição humana, a partir previamente de uma cognição animal. Podemos especificar ainda um realismo cognitivo, ou seja, que esta modele as características essenciais do processo de cognição, e não detalhes ou aberrações. Podemos especificar um ecletismo de metodologias e técnicas, evitando um compromisso prematuro com técnicas particulares. Com relação às características comportamentais, podemos especificar que a arquitetura possua reatividade, sequencialidade, que permita a modelagem de rotinas ou hábitos das criaturas, que permita a adaptação por tentativa e erro.

Com relação ao tipo de arquitetura, esta poderá ser simbólica, conexionalista ou híbridas, centralizadas ou distribuídas, poderá conter uma representação uniforme do conhecimento, representações heterogêneas, ou ainda sem representação explícita.

Ao longo dos últimos 20 anos, diversas arquiteturas cognitivas foram propostas, algumas de caráter geral, outras aplicadas somente em situações específicas. Dentre outras, podemos citar as seguintes arquiteturas: ACT-R, SOAR, EPIC, PRODIGY, DEM, COGNET, CLARION, Copycat, DUAL, Apex, Psi, Subsumption, ATLANTIS, Theo, ICARUS, AIS, MAX, Homer, Teton, RALPH-MEA, ERE.

#### 4.1 Exemplos de Arquiteturas Cognitivas Específicas

Um primeiro exemplo de arquitetura cognitiva aplicada a uma criatura virtual é a arquitetura de Rickell, apresentada na figura 12. É uma arquitetura bastante simples, somente com 3 módulos cognitivos, um para percepção, outro para cognição e outro para controle motor. Seria uma arquitetura cognitiva próxima de uma arquitetura cognitiva mínima.

A arquitetura de Albus, apresentada na figura

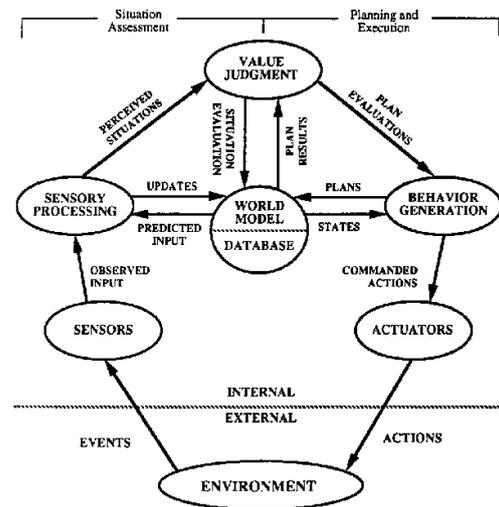


Figura 13: Arquitetura apresentada em (Albus, 1991) - Outline for a Theory of Intelligence

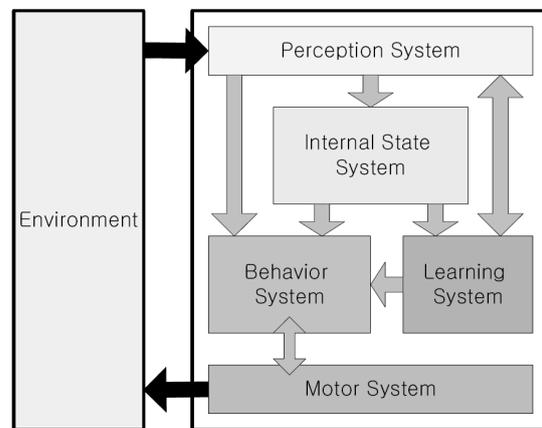


Figura 14: Arquitetura apresentada em (Kim et al., 2002) - Implementation of Artificial Creature based on Interactive Learning

13 já é um pouco mais sofisticada. Nela vemos, além dos sensores e atuadores, quatro módulos cognitivos: um para percepção sensorial, outro para geração de comportamento, outro para modelagem do mundo e outro para julgamento de valores.

A arquitetura de Kim, apresentada na figura 14. É uma variação da arquitetura de Albus. Nela não teremos um módulo de julgamento de valores, mas temos um módulo de aprendizagem.

A arquitetura de Yoon, apresentada na figura 15 é outra variação. Nesta, veremos um módulo de sistema motivacional, que poderíamos comparar com o módulo de julgamento de valores de Albus.

Trouxemos aqui a arquitetura de Arnellos, apresentada na figura 16, em função de uma curiosidade. Nesta arquitetura, vemos que a mesma identifica dois tipos de representações abstraídas do ambiente: crenças e signos. O uso de signos como veículo de representação pressupõe um mo-

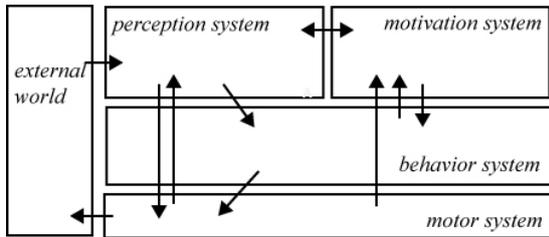


Figura 15: Arquitetura apresentada em (Yoon et al., 2000) - Interactive Training for Synthetic Characters

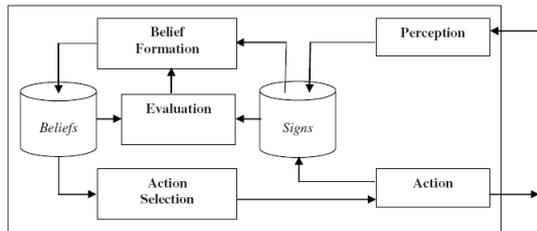


Figura 16: Arquitetura apresentada em (Arnellos et al., 2008) - Autonomy in Virtual Agents: Integrating Perception and Action on Functionally Grounded Representations

delo estendido quando comparado a arquiteturas que representam o conhecimento sobre o mundo apenas de maneira simbólica. De acordo com a semiótica, signos podem ser de diferentes naturezas, podendo ser símbolos, índices ou ícones.

A arquitetura de Isla, apresentada na figura 17, traz diversas sofisticacões. Nela já vemos uma memória de trabalho, um *Blackboard* e um sistema de navegação. Aparentemente, Isla também se preocupa em representar o mecanismo de atenção, indicando mecanismos de seleção de atenção e objetos de atenção e de desejo.

A Arquitetura “CogAff” de Aaron Sloman, apresentada na figura 18 é uma arquitetura hierárquica com 3 níveis hierárquicos. No nível inferior, vemos um conjunto de mecanismos reativos, que colhem dados dos sensores e alimentam os atuadores, por meio de mecanismos de percepção e ação. Num nível superior, os mesmos mecanismos de percepção e atenção alimentam um conjunto de raciocínios deliberativos, responsáveis por efetuar planos de ação sobre o ambiente. Esses mecanismos deliberativos, ao invés de atuar diretamente sobre o ambiente, interferem nos mecanismos reativos, modificando-lhes o comportamento. Por fim, em um nível ainda acima, os mecanismos de meta-gerenciamento, supervisionam os mecanismos deliberativos, alterando-lhes parâmetros e ajustando-os, quando necessário.

A arquitetura de Imbert, apresentada na figura 19 é uma variação da arquitetura de Sloman. Nela, o terceiro nível não é responsável pelo meta-gerenciamento, mas segundo Imbert, funci-

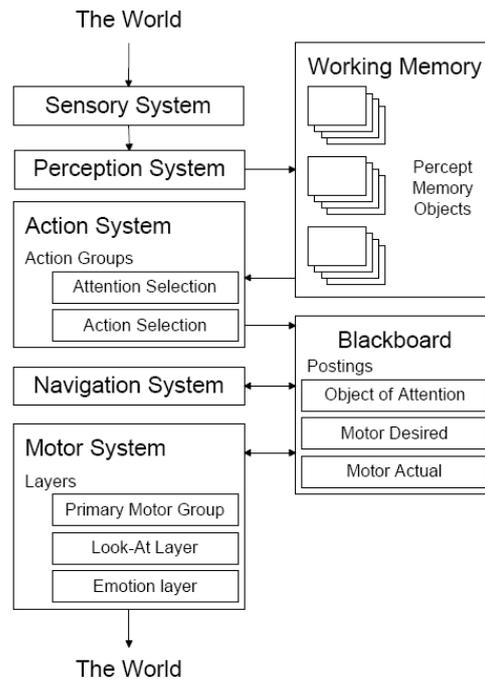


Figura 17: Arquitetura apresentada em (Isla et al., 2001) - A Layered Brain Architecture for Synthetic Creatures

Perception	Central Processing	Action
	<b>Meta-management (reflective processes) (newest)</b>	
	<b>Deliberative reasoning ("what if" mechanisms) (older)</b>	
	<b>Reactive mechanisms (oldest)</b>	

Figura 18: A Arquitetura “CogAff” de Aaron Sloman

ona como uma camada social, propiciando a interação da criatura com outras criaturas.

A arquitetura de Franklin, apresentada na figura 20 também é bastante sofisticada. O artigo que a apresenta (Franklin, 1997b), na verdade faz uma análise da evolução de diversas arquiteturas cognitivas, sendo que a arquitetura apresentada na figura 20 representa o ápice de complexidade. Nela vemos uma representação explícita de diversos módulos encontrados em outras arquiteturas, mas também uma representação explícita de mecanismos de drives, mecanismos de atenção, aprendizagem, etc.

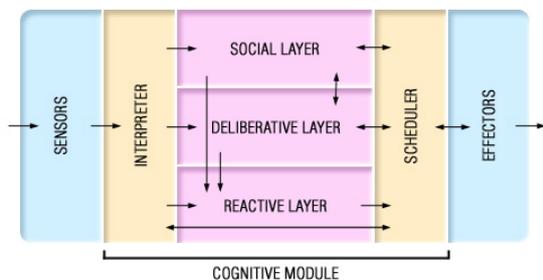


Figura 19: Arquitetura apresentada em (Imbert and De Antonio, 2005) - An Emotional Architecture for Virtual Characters

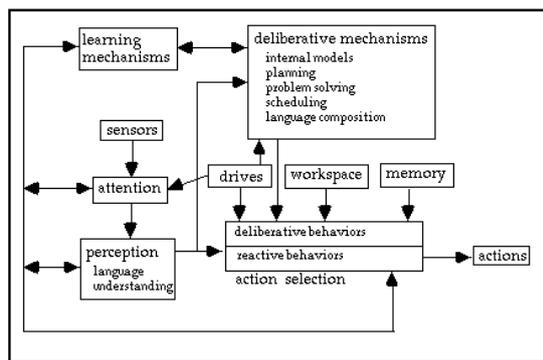


Figura 20: Arquitetura apresentada em (Franklin, 1997b) - Autonomous Agents as Embodied AI

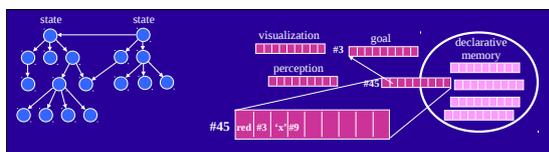


Figura 21: Exemplo comparativo entre as estruturas de conhecimento no SOAR (esquerda) e ACT-R (direita)

## 5 Exemplos de Arquiteturas Genéricas - ACT-R e SOAR

A seguir, apresentamos duas arquiteturas cognitivas de caráter genérico. Ao contrário das arquiteturas apresentadas anteriormente, que tinham um escopo bastante específico, para o controle de criaturas artificiais bastante específicas, as arquiteturas ACT-R e SOAR ao contrário são arquiteturas de propósito geral. São duas arquiteturas que vêm evoluindo já há bastante tempo, sendo que muitas vezes alguma inovação que aparece em uma delas, logo em seguida acaba tendo uma contraparte na outra.

Um exemplo figurativo e comparativo entre o ACT-R e o SOAR pode ser visto na figura 21 (SOAR à esquerda e ACT-R à direita).

Apesar de um grande número de similaridades entre as duas as estruturas de conhecimento

manipuladas por cada uma são completamente diferentes. No SOAR (à esquerda), o conhecimento é representado por meio de WMEs (Working Memory Elements) que se associam formando árvores de conhecimento. No ACT-R, o conhecimento é representado por *chunks*, estruturas lineares representadas na figura à direita.

O ACT-R (*Adaptive Control of Thought - Rational*) (Anderson et al., 2004) é uma arquitetura cognitiva que é ao mesmo tempo um modelo teórico e um sistema computacional, desenvolvido por John Anderson e sua equipe, da Universidade Carnegie-Mellon, desde os anos 70 do século XX (Anderson, 1976).

O SOAR (*State Operator And Result*) (Laird and Congdon, 2011) é também uma arquitetura cognitiva que é ao mesmo tempo um modelo teórico e um sistema computacional, desenvolvido por John Laird e sua equipe, da Universidade de Michigan, desde os anos 80 do século XX (Laird et al., 1987).

Estas duas arquiteturas cognitivas tiveram suas propostas iniciais mais ou menos ao mesmo tempo, e da mesma forma evoluíram de maneira radical, desde os seus primórdios. De uma certa maneira, podemos assumir que tanto ACT-R como SOAR não mais possuem hoje a mesma estrutura que tinham em seus primórdios. Da mesma forma, a grande maioria dos artigos e livros que visam descrever essas arquiteturas estão profundamente desatualizados, pois não incorporam as últimas inovações. Em ambos os casos, para uma compreensão mais fiel do funcionamento de cada arquitetura, o recomendável é fazer o download do software disponibilizado por seus autores em seus web-sites, e a consulta aos últimos manuais de referência.

### 5.1 ACT-R - Adaptive Control of Thought - Rational

O ACT-R é hoje uma arquitetura cognitiva híbrida (capaz de processar estruturas simbólicas e sub-simbólicas). O ACT-R é o ápice de toda uma série de predecessores, que apesar da mesma sigla, já teve diversos outros nomes, tais como *Atomic Components of Thought* ou *Adaptive Character of Thought*. Diversas versões foram desenvolvidas, as vezes com grandes diferenças funcionais entre elas. Dentre outros nomes, ACT-E, ACT\*, PUPS, ACT-R/PM e ACT-R. A presente versão do ACT-R é a 6.0.

As estruturas simbólicas funcionam como sistemas de produção. As estruturas sub-simbólicas incluem um conjunto massivo de processos paralelos, sumarizados por meio de equações matemáticas, e podem controlar os processos simbólicos. Possui ainda funções de utilidade e processos de aprendizagem.

Uma visão geral da arquitetura ACT-R pode

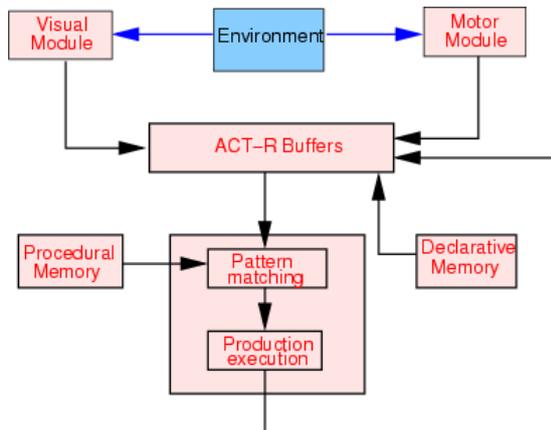


Figura 22: Arquitetura Geral do ACT-R

	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)

Figura 23: Tipos de Conhecimento Armazenados nas Memórias Declarativa e Procedural no ACT-R

ser visualizada na figura 22.

Conforme pode-se observar na figura 22, o ACT-R possui dois tipos de memórias, a memória declarativa e a memória procedural. A memória declarativa armazena conhecimento declarativo, ou seja, coisas que somos conscientes de que sabemos e que podemos usualmente descrever para outras pessoas, representado em termos de *chunks*, ou seja, configurações de elementos que codificam as diversas coisas que sabemos. A memória procedural armazena conhecimento procedural, ou seja, conhecimento que exibimos por meio de nosso comportamento, mas do qual não somos conscientes, representado por meio de regras de produção que especificam como buscar e utilizar o conhecimento declarativo para resolver problemas. Além disso, em um nível sub-simbólico, a memória declarativa armazena ainda os níveis de ativação para cada chunk, que influem na probabilidade de seleção e a memória procedural armazena funções de utilidade, que influem na probabilidade de uso, conforme mostrado na figura 23

Para compreendermos como funciona o ACT-R, é necessário que compreendamos sua terminologia básica. Um primeiro conceito que é necessário é o conceito de *chunk*. Um chunk é uma unidade elementar de cognição no ACT-R. Chunks são n-tuplas tipadas, utilizadas para armazenar qualquer tipo de informação. Um exemplo da definição de tipos de chunk pode ser visto na figura

```
(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)
```

Figura 24: Exemplos de Tipos de Chunks no ACT-R

```
(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))
```

Figura 25: Exemplos de Chunks no ACT-R

<pre>(P counting-example =goal&gt;   isa    count   state  incrementing   number =num1 =retrieval&gt;   isa    count-order   first  =num1   second =num2  ==&gt; =goal&gt;   number =num2 +retrieval&gt;   isa    count-order   first  =num2 )</pre>	<p><b>English Description</b>  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</p>
--	---

Figura 26: Exemplo de Regra de Produção no ACT-R

24. Um exemplo de chunks pode ser visto na figura 25. Um exemplo de regra de produção pode ser visto na figura 26.

Um outro conceito importante no ACT-R é o conceito de *buffers*. Buffers correspondem à interface entre a memória procedural e os outros módulos do sistema. É por meio de buffers que a memória procedural pode coletar informações de outros módulos do sistema, processar essas informações e enviar novas informações aos diferentes módulos do sistema. Por exemplo, assumindo que existam dois módulos chamados *goal* e *retrieval* o buffer *goal* é a interface para o módulo *goal*. O buffer *retrieval* é a interface para o módulo *retrieval*. Cada buffer só acomoda um único chunk a cada instante de tempo. Os valores dos buffers podem ser testados e alterados por produções. Nas regras de produção (veja figura 26), os valores dos buffers podem ser consultados ou alterados. Por exemplo, considerando-se o buffer *retrieval*, as seguintes ações podem ser executadas:

```
=retrieval> Modifica valores dos slots do buffer
+retrieval> Faz uma requisição ao módulo em questão
-retrieval> Limpa o buffer em questão
```

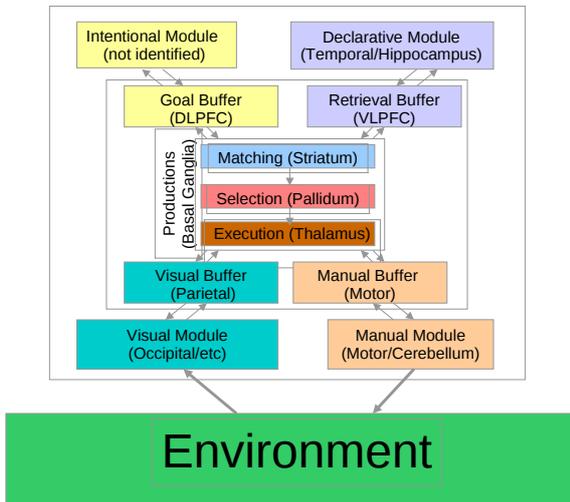


Figura 27: Módulos e Buffers do ACT-R 5.0

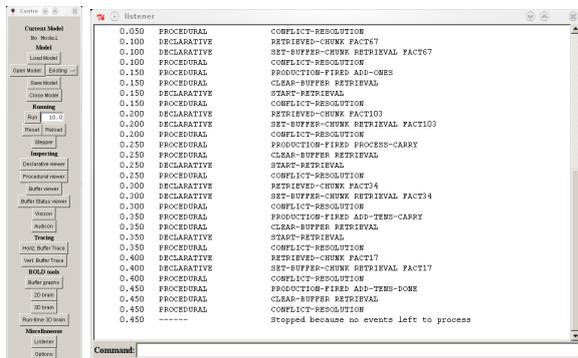


Figura 28: Ferramentas de Depuração e Teste de Modelos ACT-R 6.0

?retrieval> Efetua 'queries' ao módulo

Somente uma produção pode disparar a cada instante. Escolher a produção é chamada de uma resolução de conflitos. Um modelo ACT-R é um programa em LISP contendo os comandos ACT-R.

Um exemplo dos módulos e buffers disponíveis na versão ACT-R 5.0 pode ser visto na figura 27.

As janelas da ferramenta de depuração e teste de modelos do ACT-R 6.0 podem ser vistas na figura 28.

Uma lista dos novos módulos e buffers disponíveis na versão 6.0 do ACT-R podem ser vistos na figura 29.

A versão original da ferramenta ACT-R é programada em LISP. Isso cria algumas dificuldades de integração com outras linguagens. Existe uma versão em Java do ACT-R, chamada de JActr. Maiores informações podem ser obtidas em [jactr.org](http://jactr.org). Para saber maiores informações sobre a ferramenta computacional ACT-R ou efetuar seu download, deve-se acessar o endereço <http://act-r.psy.cmu.edu/>, onde pode-se encontrar diversos tutoriais explicando o uso dos diferentes módulos do sistema, além do manual de

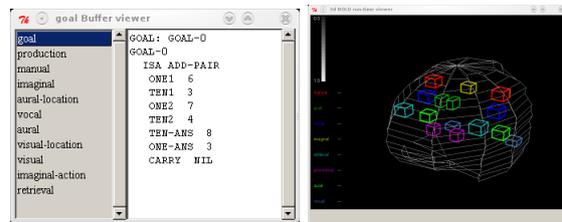


Figura 29: Novos Módulos/Buffers e sua localização estimada no cérebro humano, na versão 6.0 do ACT-R

referência.

## 5.2 SOAR (State, Operator And Result)

Da mesma maneira que o ACT-R, o SOAR também é uma arquitetura cognitiva que passou por diversas transformações ao longo de sua existência, onde diversos novos módulos foram acrescentados. A base lógica de sua constituição se apóia na teoria dos Sistemas de Símbolos Físicos de Alan Newell (Newell, 1980), bem como nos sistemas de produção dos primórdios da inteligência artificial. A proposta do SOAR é realizar uma busca em espaços de problemas, com uma estrutura de controle automática/deliberativa em dois níveis e aprendizagem contínua, determinada por impasses. Uma visão esquemática do SOAR pode ser vista na figura 30. A versão mais moderna da arquitetura pode ser vista na figura 31.

As estruturas básicas suportadas pela arquitetura são os estados e os operadores. Estados são toda informação sobre a situação corrente, incluindo a percepção e a descrição de metas correntes e espaços de problemas. Os operadores ocasionam passos no espaço de problemas. Uma memória de trabalho armazena percepções e a hierarquia de estados e seus operadores associados. Esse conteúdo pode acionar a memória de longo prazo ou ações motoras. A memória de longo prazo é um repositório do conteúdo processado pela ar-

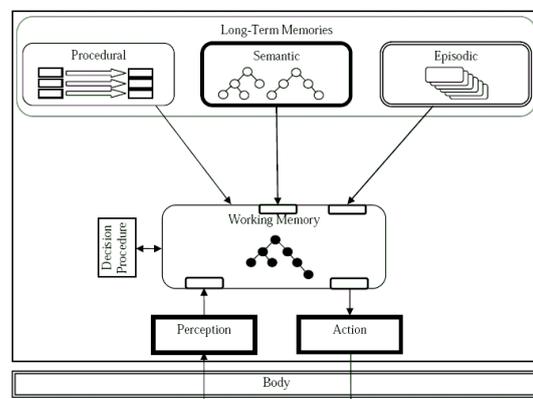


Figura 30: Visão Esquemática do SOAR em suas versões mais primitivas

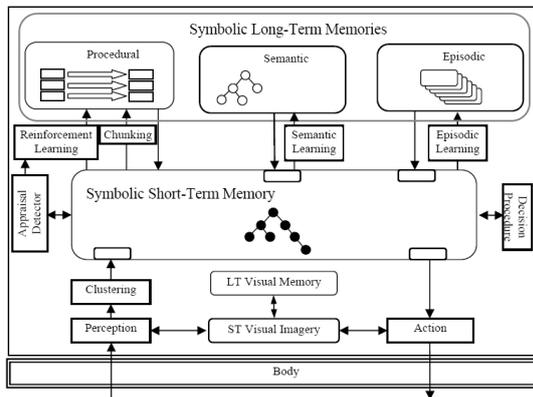


Figura 31: Visão Esquemática do SOAR em suas últimas versões

quietura capaz de produzir comportamento. A memória procedural armazena regras e é acessada automaticamente durante os ciclos de decisão. A memória semântica armazena estruturas declarativas e a memória episódica armazena episódios. As memórias de longo prazo são impenetráveis, ou seja, não podem ser examinadas diretamente. Certos procedimentos recuperam informações nas memórias de longo prazo e armazenam na memória de trabalho. Além disso, o SOAR conta com uma interface Perceptiva/Motora, que realiza 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.

O ponto fundamental do funcionamento do SOAR está em seu ciclo de decisão. Esse, é um processo arquitetural básico suportando a cognição. Esse ciclo envolve a seleção e aplicação de operadores. Ele pode ser dividido em três fases: elaboração, decisão e aplicação.

A fase de elaboração faz um acesso paralelo à Memória de Longo Prazo para elaborar o estado. É feita então uma sugestão de novos operadores e sua posterior avaliação.

A fase de decisão envolve um procedimento de decisão onde, a partir da definição de preferências por operadores (realizada por meio de uma linguagem de preferência por operadores), obtém-se como resultado que um operador é selecionado. Neste ínterim, pode ocorrer um impasse, resultado de preferências incompletas ou conflito. Nesse caso, esse impasse precisa ser resolvido.

Na fase de aplicação, regras são disparadas de forma a modificar os estados. A seleção de um único operador por ciclo de decisão impõe um gargalo cognitivo à arquitetura, limitando o trabalho cognitivo por ciclo.

Impasses sinalizam uma falta de conhecimento, e portanto a oportunidade para aprendizagem. Esta acontece automaticamente quando o conhecimento elicitado pelo estado corrente não

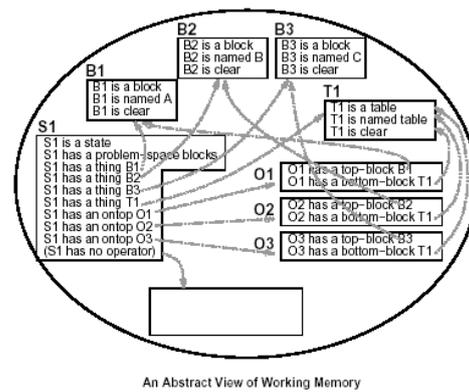


Figura 32: Exemplo de Conteúdo da Memória de Trabalho do SOAR

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

Figura 33: Definição de Working Memory Elements (WME)

é suficiente para o procedimento de decisão selecionar um operador. A linguagem de impasses é independente de domínio. Quando ocorre um impasse, a arquitetura automaticamente inicia a criação de um novo sub-estado cuja meta é resolver o impasse e impõe uma hierarquia de metas/sub-estados no contexto da memória de trabalho.

O SOAR conta com quatro mecanismos de aprendizagem: *chunking*, aprendizagem por reforço, aprendizagem episódica e aprendizagem semântica. O *chunking* cria automaticamente novas regras na memória de longo prazo utilizando-se dos resultados gerados de um impasse. De certa forma, ele previne que um impasse ocorra em situações similares no futuro. A aprendizagem por reforço ajusta os valores das preferências por operadores e a aprendizagem episódica armazena a história das experiências. A aprendizagem semântica captura asserções declarativas mais abstratas.

O centro de funcionamento do SOAR é a memória de trabalho, onde diferentes elementos, chamados de WMEs (*Working Memory Elements*) são armazenados. A figura 32 apresenta um exemplo de conteúdo da memória de trabalho. Na figura 33 temos a estrutura básica de um WME. Na figura 34 temos um possível exemplo de um conteúdo da memória de trabalho. Na figura 35 mostramos como os diversos elementos da memória de trabalho podem se inter-relacionar entre si, formando árvores ou grafos de conhecimento.

Na figura 36 vemos um exemplo de como podemos definir preferências e na figura 37 vemos um exemplo de uma produção ou regra.

O SOAR, da mesma maneira que o ACT-R

```

(O43 ^isa apple
    ^color red
    ^inside O53
    ^size small
    ^X44 200)

(O87 ^isa ball ^color red
    ^inside O53 ^size big)

(O53 ^isa box ^size large
    ^color orange
    ^contains O43 O87)

(X44 ^unit grams ^property mass)

```

Figura 34: Exemplo do Conteúdo da Memória de Trabalho

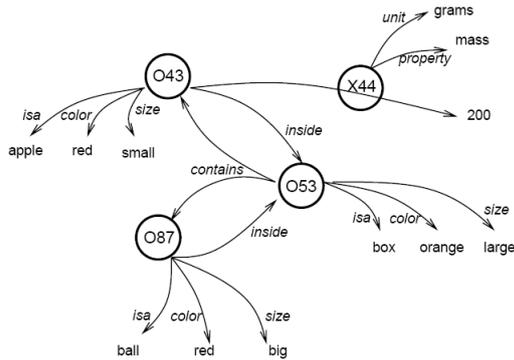


Figura 35: Associação entre WMEs

```

(S1 ^operator O3 +)
(S1 ^operator O3 > O4)

```

Figura 36: Exemplos de Definições de Preferências

é ao mesmo tempo um modelo teórico e uma ferramenta computacional. A versão mais atual do SOAR, no momento da escrita deste artigo é a versão 9.3.1. A ferramenta de desenvolvimento conta com um debugger, chamado de SOAR Debugger (figura 38) e recursos de programação visual (Visual SOAR - figura 39).

A ferramenta computacional do SOAR possui integração com diversas linguagens, tais como C++, Java, Tcl, e possui um mecanismo genérico via *Sockets*, por meio do qual pode se integrar com praticamente qualquer outra linguagem de programação. Possui ainda uma linguagem de *markup*, o SML *Soar Markup Language*. Para maiores informações sobre o SOAR, e mesmo para fazer o download da ferramenta, interessados podem consultar o web-site: <http://sitemaker.umich.edu/soar/home>, onde estão disponíveis tutoriais explicando o uso do SOAR em diferentes situações, além de manuais do usuário e manuais de referência.

## 6 Conclusões

Neste trabalho, apresentamos um *survey* sobre o uso de arquiteturas cognitivas em criaturas artifi-

```

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>)

```

Figura 37: Exemplo de uma Produção - Regra

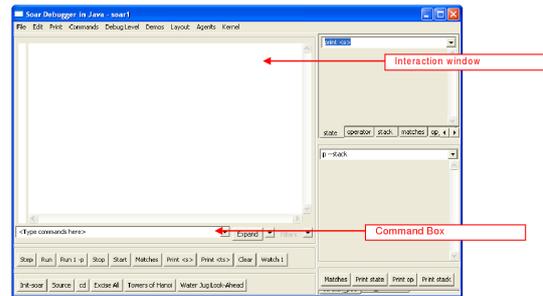


Figura 38: Ferramenta de Debugger do SOAR

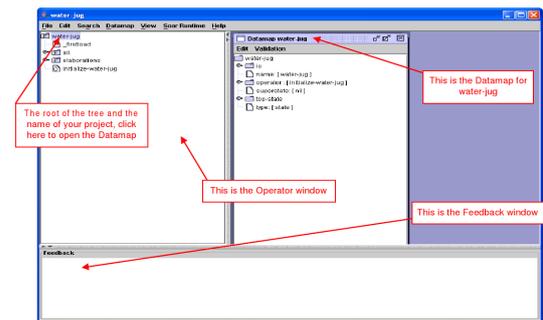


Figura 39: Tela do Visual SOAR

ciais. Esse escopo pode ser entendido como uma sub-área da *Vida Artificial* onde métodos do tipo *top-down* fomentam a criação de sistemas artificiais que possuem alguma semelhança com sistemas vivos. Podemos portanto entender essa como sendo a soleira da área de pesquisa em Vida Artificial. O presente artigo visou apresentar um retrato dessa área de pesquisa, direcionando o leitor interessado à literatura pertinente, ao mesmo tempo que propicia uma visão geral sobre essa área de pesquisa.

## Referências

- Albus, J. (1991). Outline for a theory of intelligence, *Systems, Man and Cybernetics, IEEE Transactions on* **21**(3): 473-509.
- Anderson, J. (1976). *Language, memory, and thought*, Lawrence Erlbaum.

- Anderson, J., Bothell, D., Byrne, M., Douglass, S., Lebiere, C. and Qin, Y. (2004). An integrated theory of the mind., *Psychological review* **111**(4): 1036.
- Arnellos, A., Vosinakis, S., Anastasakis, G. and Darzentas, J. (2008). Autonomy in virtual agents: Integrating perception and action on functionally grounded representations, *Artificial Intelligence: Theories, Models and Applications* pp. 51–63.
- Badler, N. I. (1997). Real-time virtual humans, *In International Conference On Digital Media Futures*.
- Balkenius, C. (1995). *Natural Intelligence in Artificial Creatures*, Lund University Cognitive Studies.
- Braitenberg, V. (1986). *Vehicles: experiments in synthetic psychology*, Bradford Books, MIT Press.
- Dean, J. (1998). Animats and what they can tell us, *Trends in Cognitive Sciences* **2**(2).
- Franklin, S. (1997a). *Artificial minds*, The MIT Press.
- Franklin, S. (1997b). Autonomous agents as embodied ai, *Cybernetics and Systems* **28**(6): 499–520.
- Grand, S. and Cliff, D. (1998). Creatures: Entertainment software agents with artificial life, *Autonomous Agents and Multi-Agent Systems* **1**(1): 39–57.
- Grand, S., Cliff, D. and Malhotra, A. (1997). Creatures: Artificial life autonomous software agents for home entertainment, *Proceedings of the first international conference on Autonomous agents*, ACM, pp. 22–29.
- Hill, R. (1999). Modeling perceptual attention in virtual humans, *Proceedings of the 8th Conference on Computer Generated Forces and Behavioral Representation*, pp. 563–573.
- Hogg, D., Martin, F. and Resnick, M. (1991). *Braitenberg creatures*, Epistemology and Learning Group, MIT Media Laboratory.
- Hull, C. (1952). A behavior system; an introduction to behavior theory concerning the individual organism.
- Imbert, R. and De Antonio, A. (2005). An emotional architecture for virtual characters, *Virtual Storytelling. Using Virtual Reality Technologies for Storytelling* pp. 63–72.
- Isla, D., Burke, R., Downie, M. and Blumberg, B. (2001). A layered brain architecture for synthetic creatures, *International Joint Conference on Artificial Intelligence*, Vol. 17, pp. 1051–1058.
- Kim, Y., Kim, J., Kim, J. and Lim, J. (2002). Implementation of artificial creature based on interactive learning, *Proceedings of the FIRA Robot World Congress*, pp. 369–374.
- Kitano, H. (1998). *RoboCup-97: robot soccer World Cup I*, Lecture notes in computer science, Springer.
- Laird, J. and Congdon, C. (2011). The soar user's manual version 9.3.1.
- Laird, J., Newell, A. and Rosenbloom, P. (1987). Soar: An architecture for general intelligence, *Artificial intelligence* **33**(1): 1–64.
- Langley, P. and Laird, J. (2009). Cognitive architectures: Research issues and challenges, *Cognitive Systems Research* **10**(2): 141–160.
- Langton, C. G. (1986). Studying artificial life with cellular automata, *Physica D* **22**: 120–149.
- Meyer, J. and Wilson, S. (1990). *From animals to animats: proceedings of the First International Conference on Simulation of Adaptive Behavior*, Complex Adaptive Systems, MIT Press.
- Newell, A. (1980). Physical symbol systems, *Cognitive science* **4**(2): 135–183.
- Rickel, J. and Johnson, W. (1999). Animated agents for procedural training in virtual reality: Perception, cognition, and motor control, *Applied Artificial Intelligence* **13**(4-5): 343–382.
- Sims, K. (1994a). Evolving 3d morphology and behavior by competition, *Artificial Life* **1**(4): 353–372.
- Sims, K. (1994b). Evolving virtual creatures, *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, SIGGRAPH '94, ACM, New York, NY, USA, pp. 15–22.
- Slooman, A. (2000). Architecture-based conceptions of mind, *Cognitive Science Research Papers-University Of Birmingham CSR*.
- Sun, R. (2004). Desiderata for cognitive architectures, *Philosophical Psychology* **17**(3): 341–373.
- Sun, R. (2007). The importance of cognitive architectures: An analysis based on clarion, *Journal of Experimental and Theoretical Artificial Intelligence* **19**(2): 159.

- Varela, F., Thompson, E. and Rosch, E. (1999). *The embodied mind: cognitive science and human experience*, MIT Press.
- Yaeger, L. (1994). Computational genetics, physiology, metabolism, neural systems, learning, vision, and behavior or poly world: Life in a new context, *Santa Fe Institute Studies In The Sciences Of Complexity-Proceedings Volume-*, Vol. 17, pp. 263–263.
- Yoon, S., Burke, R., Blumberg, B. and Schneider, G. (2000). Interactive training for synthetic characters, *Proceedings of the National Conference on Artificial Intelligence*, AAAI Press; MIT Press;, Menlo Park, CA; Cambridge, MA; London;, pp. 249–255.