



**FACULDADE DE INFORMÁTICA**  
**PUCRS – Brazil**  
<http://www.inf.pucrs.br>

**Modelagem de agentes  
utilizando a arquitetura BDI**

Alexandre de Oliveira Zamberlam e Lucia Maria Martins Giraffa

**TECHNICAL REPORT SERIES**

---

Number 008  
April, 2001

Contact:

[alexz@inf.pucrs.br](mailto:alexz@inf.pucrs.br)

<http://www.inf.pucrs.br/~alexz>

[giraffa@inf.pucrs.br](mailto:giraffa@inf.pucrs.br)

<http://www.inf.pucrs.br/~giraffa>

Alexandre de Oliveira Zamberlam is a graduate student of UNIJUÍ – Universidade Regional do Noroeste do Estado do RS – Brazil. He is member of the GIE research group (Computer Science apply to Education Research Group) since 2000. He receives a federal graduate research grant from CAPES (Brazil) to support his research.

Lucia Maria Martins Giraffa works at PUCRS/Brazil since 1986. She is na titular professor and leader of the GIE Gr. She develops research in Multi-agent systems, Artificial Intelligence apply to Education, and design of educational software. She got her Ph.D. in 1999 at UFRGS (Brazil).

Copyright © Faculdade de Informática – PUCRS  
Published by the Campus Global – FACIN – PUCRS  
Av. Ipiranga, 6681  
90619-900 Porto Alegre – RS – Brazil

# Modelagem de agentes utilizando a arquitetura BDI

Relatório Técnico 008/2001

Alexandre de Oliveira Zamberlam (mestrando)\*  
Lucia Maria Martins Giraffa (orientadora)†

## 1 Introdução

A Inteligência Artificial (IA) clássica, com pressupostos de origem psicológica, tem seu estudo centrado no comportamento individual de ser humano. A Inteligência Artificial Distribuída (IAD) utiliza um modelo de inteligência baseado no comportamento social, com ênfase nas ações e interações de agentes que podem ser entidades reais ou virtuais imersas num ambiente sobre o qual são capazes de agir [ALV97]. A IAD possui pressupostos baseados em teorias sociológicas. A IAD tradicionalmente é dividida em duas áreas: resolução distribuída de problemas (RDP) e sistemas multiagentes (SMA). Esta classificação proposta por Demazeau [DEM90] é adotada por vários outros autores.

O termo agente é uma noção central e fundamental para área de IAD. Este termo vem sendo utilizado para denotar desde simples processos de hardware e/ou software até entidades sofisticadas capazes de realizar tarefas complexas. Esta diversidade reflete o estado atual da área, onde não se possui um consenso do que realmente um agente é. Cada grupo de pesquisa segue uma determinada linha, de acordo com seus próprios objetivos, apresentando sua definição personalizada do termo agente. Russell & Norvig [RUS96], definem um *agente* como um sistema capaz de perceber através de sensores e agir em um dado ambiente através de atuadores. Esta definição é enriquecida através da descrição de

---

\*alexz@inf.pucrs.br

†giraffa@inf.pucrs.br

um conjunto de propriedades usualmente aceitas e que os agentes devem exibir. Algumas das propriedades consideradas para construção de agentes são:

- Autonomia: escolhe a ação a tomar, baseado mais na própria experiência do que no conhecimento embutido pelo projetista. As ações do agente não requerem interferência humana direta;
- Temporalidade: opta por permanecer ou não no ambiente;
- Comunicabilidade: troca informações oriundas do ambiente ou de outros agentes;
- Mobilidade: capaz de se deslocar para ambientes diferentes do original;
- Flexibilidade: aceita a intervenção de outros agentes;
- Pró-atividade: é capaz de, além de responder a estímulos do ambiente, exibir um comportamento orientado a objetivos. Ou seja, é capaz de prever como atingir ou evitar um determinado estado ou objetivo.

Segundo [GIR98], o ambiente e seu respectivo projeto (*design*) irão determinar as propriedades que o agente terá, ou seja, um agente irá incorporar no seu *design* mais propriedades dependendo do seu papel dentro da sociedade que pertence.

Além destas propriedades citadas acima podemos incluir a *habilidade social* (via algum tipo de comunicação entre agentes), a *reatividade* (reage a estímulos recebidos de outros agentes ou do ambiente), a *racionalidade* (depende do grau de sucesso e percepção que o agente possui), *adaptabilidade* (capacidade de se adaptar a modificações no ambiente, alterando planos previamente concebidos ou aprendendo através da interação com o ambiente) e *representação através de personagens* (tanto em 2D, 3D como através de avatares<sup>1</sup>).

Para projetarmos um SMA, temos algumas etapas a considerar: definição do modelo (arquitetura) da sociedade, quantificação e papéis dos agentes que a compõem. A modelagem e a implementação de um agente não são tarefas triviais ou que se apoiem em modelos preestabelecidos. Cada agente possui características básicas (propriedades) que o qualificam como tal e o diferem de simples objetos.

Existe uma divisão, de cunho didático, para classificar os agentes. Esta taxonomia divide os agentes em dois grupos: reativos (não deliberativos) e cognitivos (deliberativos). Esta divisão possui variantes na literatura (vide trabalhos de [DEM90]; [ALV97]; [OLI96]; [WOO95]).

---

<sup>1</sup>Avatar é um ícone gráfico que representa uma pessoa real num sistema do espaço virtual.

O presente trabalho enfoca a sub-área de pesquisa envolvendo agentes cognitivos modelados através de estados mentais, detendo-se nos agentes que utilizam a tríade de estados mentais compostos por crença (*belief*), desejo (*desire*) e intenção (*intention*), denominado na literatura de arquitetura BDI.

O objetivo deste trabalho é aprofundar conhecimento sobre agentes modelados em arquitetura BDI, ferramentas e ambientes utilizados para modelar e/ou implementar estes agentes. O trabalho também possui a função de auxiliar a montar o referencial teórico e a base para construção do PEP (Plano de Estudos e Pesquisa) onde se pretende detalhar a proposta para construção de um “Editor de Estados Mentais” associado à ferramenta X-BDI [MÓR00] detalhada no capítulo 4.3.

Este volume está dividido em 6 seções. A seção 2 apresenta o contexto no qual se insere a abordagem de estados mentais utilizados para modelagem de agentes. A seção 3 apresenta diferentes tipos de arquiteturas baseadas no paradigma BDI. A seção 4 apresenta ferramentas para modelagem, construção e representação de sistemas cognitivos baseados na arquitetura BDI. As considerações finais sobre o trabalho realizado são apresentadas na seção 5. A bibliografia utilizada é apresentada na seção 6.

Intencionalmente é dado destaque a duas ferramentas que permitem a especificação e implementação de agentes BDI. São elas dMARS e X-BDI. Esta ênfase dá-se em função do trabalho futuro a ser desenvolvido no programa de mestrado.

## 2 Arquiteturas baseadas em Estados Mentais

O que caracteriza o agente são as interações que ele realiza com o mundo e os processos internos que possibilitam a realização destas interações. A especificação de quais e como são estes processos internos é chamada de arquitetura do agente [GIR99]. Diferentes arquiteturas têm sido propostas com o objetivo de caracterizar os agentes com um particular nível de inteligência e de autonomia, as quais podem ser classificadas de acordo com o mecanismo utilizado pelo agente para a seleção das ações. Segundo [COR94], para definir a arquitetura interna do agente é necessário saber, inicialmente, qual o tipo de tarefa que o agente irá realizar e o seu papel na sociedade. Uma vez considerado isto, o agente pode ser classificado como:

- reativo (ou não-deliberativo): a escolha da ação (resposta) está diretamente situada na ocorrência de um conjunto de eventos (estímulos) que ele percebe no e do ambiente, captados por seus sensores ou por mensagens enviadas por outros agentes.

- cognitivo (deliberativo): possui um processo explícito para escolher a ação a ser realizada. Esta ação é realizada através de uma representação simbólica do mundo, de um plano e de uma função de utilidade.

Um agente cognitivo é um agente racional que possui alguma representação explícita de seu conhecimento e objetivos. Um agente reativo não necessariamente é um agente racional: seu comportamento pode ser definido através de um padrão de estímulo-resposta. Um agente pode ser “mais cognitivo” do que outro, conforme o grau de racionalidade explícita de seu comportamento [OLI96].

Arquiteturas de agentes cognitivos, segundo [OLI96], podem ser divididas em:

- arquiteturas funcionais: onde o agente é composto por módulos que representam cada uma das funcionalidades necessárias para sua operação. O agente possui conhecimento, um conjunto de objetivos, capacidade de percepção, comunicação, decisão e raciocínio;
- arquiteturas baseadas em estados mentais: adotam uma perspectiva de inspiração psicológica para definir a estrutura dos agentes, que são entidades cujo estado é visto como consistindo de componentes mentais tais como crenças, capacidades, escolhas e compromissos. Por esta razão o estado de um agente é chamado de estado mental.

A utilização de estados mentais para modelagem de agentes cognitivos é chamada de *abordagem mentalista* [MÓR98]. A idéia principal desta abordagem mentalista, segundo [COR94], se concentra no fato de que o agente cognitivo possui estados internos que se relacionam com o estado do ambiente com o qual interage. Estes estados seriam correspondentes aos estados mentais humanos, que apresentam um vínculo com o mundo em termos da sua existência e significância. Essa característica pela qual os estados mentais humanos “são acerca de”, “referem-se a objetos ou situações do mundo”, ou “se dirigem a” é chamada de intencionalidade. Crença, desejo, expectativa e intenção são exemplos de estados intencionais [GIR99]. Estado intencional é uma propriedade de direcionamento do mundo para o agente e vice-versa. Por exemplo, “a porta está fechada”, que é uma crença sobre a porta (do mundo para o agente), e “entrar na sala”, que é o desejo (do agente para o mundo). Sendo assim, sistemas intencionais são aqueles em que se atribui direcionalidade a certos estados mentais. Considerada uma abordagem reducionista [SEA69]; [DEN87], é oriunda do entendimento popular sobre explicações para ação (“*folk psychology*” - psicologia do senso comum). É importante salientar que intencionalidade não está ligada ao

estado mental intenção e sim ligada a todos os estados mentais que exibem a propriedade de direcionalidade.

Nessa visão mentalista o que faz qualquer componente de hardware ou software ser um agente é precisamente o fato de ele poder ser analisado e controlado em termos destes componentes mentais. Assim, a questão do que é um agente fica substituída pela questão de quais entidades podem ser vistas como tendo um estado mental [OLI96].

Um bom exemplo de arquitetura baseada em estados mentais é a arquitetura proposta por [COR94] denominada de SEM - Sociedade dos Estados Mentais. Nessa arquitetura, todo agente é composto por quatro subagentes ou agentes locais, cada um deles implementando/representando um estado mental considerado necessário para definir o comportamento do agente como um todo. Estes subagentes são: crença ( $A^C$ ), desejo ( $A^D$ ), intenção ( $A^I$ ) e expectativa ( $A^E$ ). Os agentes locais trocam mensagens entre si e, possivelmente, com o meio externo. O comportamento de cada agente local é especificado formalmente com base na Teoria das Situações<sup>2</sup> Devlin apud [OLI96]. O comportamento do agente global é o resultado dos comportamentos dos agentes locais.

Maiores detalhes podem ser encontrados em [COR94] e [MOU96]. No trabalho de Mousalle e Viccari [MOU96] é apresentada uma aplicação da arquitetura SEM para construção de um Sistema Tutor Inteligente (STI) para ensino de divisão inteira. Neste trabalho podemos observar a coreografia que existe nos estados mentais associados à cada ação dos agentes envolvidos (no caso um aluno e um tutor).

O termo coreografia refere-se à dinâmica envolvendo os estados mentais que ocorre durante a interação entre os agentes. Por exemplo, a mudança na base de crenças (revisão, remoção, etc), a adoção de um desejo candidato à intenção e a confirmação ou não de uma expectativa. Estes exemplos representam alguns dos mecanismos que podem acontecer ao longo de uma interação entre agentes modelados nesta abordagem.

Dentro das arquiteturas baseadas em estados mentais, encontra-se a abordagem BDI. As idéias básicas da abordagem BDI baseiam-se na descrição do processamento interno de um agente utilizando um conjunto básico de estados mentais (crença, desejo e intenções) e na definição de uma arquitetura de controle através da qual o agente seleciona racionalmente o curso de suas ações. Algumas abordagens de arquitetura BDI agregam as noções de planos e objetivos, como por exemplo os trabalhos de Bratman et al. [BRA84]; [BRA87]; [BRA89], Rao e Georgeff [RAO92].

---

<sup>2</sup>Teoria das Situações surgiu a partir da hipótese de que os agentes cognitivos são capazes de identificar objetos, situações, sucessões de situações, fazer referências a situações e comportar-se de acordo com elas, e discriminar localizações espaciais e temporais [MOU96].

## 2.1 Estados Mentais

Os conceitos associados aos estados mentais de uma arquitetura BDI são descritos a seguir. São apresentadas diferentes visões para cada um destes estados mentais. As diferentes abordagens foram selecionadas em função do referencial teórico utilizado pelo grupo de pesquisa no qual este trabalho está inserido.

- Crenças:

Segundo Bratman [BRA89], crenças são as visões que o agente possui sobre o ambiente em que ele se encontra. É o conhecimento do mundo (ambiente) de forma explícita.

Para Corrêa [COR94], crença é um estado mental intencional fundamental para as interações dos agentes, com noção idêntica a de conhecimento.

Para Shoham [SHO93], crença é representado por um operador modal  $B$ , como:

$B_a^t \varphi$ , significando que o agente  $a$ , no tempo  $t$ , acredita na sentença  $\varphi$ .

Para [MÓR00] uma crença é representada por um conjunto  $B$  contendo sentenças que descrevem o domínio do problema e que usam ELP<sup>3</sup> (ver seção 4.3). Um agente  $A$  acredita em uma propriedade  $P$ , num tempo  $T$  se, a partir de  $B$  e  $T$ , o agente pode deduzir  $BEL(Ag, P)$  no tempo  $T$ .

Sendo assim, crenças podem ser vistas como componentes informativos do estado do sistema, necessárias para fornecer informações sobre o provável estado do ambiente. O agente pode expressar suas expectativas sobre o estado atual do mundo e sobre a chance de uma ação ou um conjunto de ações atingir um certo efeito e considerá-la como uma crença futura. Esta perspectiva de considerar o estado mental expectativa como uma crença futura que o agente tem (ou terá) foi utilizado no trabalho de [GIR99] e difere do trabalho de [MOU96], onde o estado mental expectativa foi modelado de maneira diferenciada das crenças do agente.

Para se adotar esta perspectiva é necessário que a ferramenta a ser utilizada para construção do agente incorpore mecanismos (predicados) que envolvam a questão de tempo, como no caso do X-BDI de [MÓR00] (vide capítulo 4.3).

- Desejos:

É um dos estados mentais humanos que não tem uma definição precisa, sendo, portanto, usado e explicado de diversas maneiras.

---

<sup>3</sup>*Extended Logic Programming*



Para [RAO95], os desejos representam o estado motivacional do sistema. Um estado mental é motivador se é um mecanismo ou representação que tende a produzir, modificar ou selecionar ações à luz das crenças.

Segundo Giraffa, Mora e Viccari [GIR98], é uma noção abstrata que especifica as preferências acerca dos estados futuros do mundo ou o curso das ações que o agente, possivelmente, quer que se verifiquem. Desejo é um estado mental intencional e com potencial motivador das ações do agente, apresentando as seguintes características:

- representa uma situação ou um conjunto de situações em que o agente gostaria que o mundo estivesse;
- pode estar em conflito com as crenças do agente;
- pode apresentar simultaneamente desejos conflitantes;
- não causa diretamente as ações, mas pode potencialmente gerar suas ocorrências, deixando por conta das intenções a realização de tais ações.

A idéia de desejo como motivador, no sentido de ter uma representação e um mecanismo que gere e selecione objetivos, tem sido pouco desenvolvida e aplicada na construção de agentes inteligentes. Mesmo nos trabalhos em arquiteturas BDI, os desejos são tratados no sentido de objetivos. A sua representação como operador modal do tipo  $DES(a, p)$ , onde o agente  $a$  tem como desejo a proposição  $p$ , apresenta inconvenientes, visto que se  $q$  é uma consequência lógica de  $p$  (ou seja,  $p \rightarrow q$ ), então, da semântica dos mundos possíveis decorre  $DES(a, q)$ . Isto não é, necessariamente, verdadeiro. Por exemplo: o desejo de fazer uma viagem implica em gastar muito dinheiro, não necessariamente implicam no desejo de gastar muito dinheiro.

Para [COR94] o desejo é representado por  $C \models \{ \langle Des, A, P, e, l_0, t_0, v \rangle \}$ , onde

- $Des$  é a relação para a representação de desejo;
- $A$  representa o agente que possui desejo;
- $P$  é uma proposição;
- $e$  é uma proposição particular que informa sobre a satisfação, não satisfação, urgência, intensidade e insistência do desejo  $D$ ;
- $l_0$  é uma localização espacial (local) associada à ocorrência do desejo  $D$ ;
- $t_0$  é o tempo associado à ocorrência do desejo  $D$ ;
- $v$  1, se o ocorre o desejo  $D$  ao agente  $A$ ;  
0, se não ocorre o desejo  $D$  ao agente  $A$ .

No X-BDI [MÓR00] um desejo é representado por um conjunto de sentenças  $DES(Ag, P, Atr)$ , em que  $Ag$  é a identificação de agente,  $P$  é uma propriedade e  $Atr$  é uma lista de atributos.

- Intenções:

Intenções representam os componentes deliberativos do sistema, que servem para decidir o curso de ação que deve ser tomado pelo sistema.

Intenção é o resultado da escolha com comprometimento, a qual leva o agente a ação.

Bratman [BRA84][BRA89], distingue o conceito de fazer coisas intencionalmente (ação) e possuir intenção de fazê-las (estado mental). Os exemplos, a seguir, identificam as duas formas distintas embutidas no uso cotidiano da palavra intenção: ações (físicas, verbais, etc.) e ações mentais (estados da mente):

- "escrevo esta sentença". Intencionalmente escrevo esta sentença, caracteriza ação no conceito de intenção;
- "escrevi esta sentença para ilustrar determinada distinção". A intenção de exemplificar caracteriza o estado da mente no conceito de intenção. Ter 'intenção de' não garante que a ação será realizada, mas apenas que se intenciona realizá-la.

Como ação intencional e intenção de agir são relacionados? Existem duas abordagens comuns para esse problema:

- modelo desejo-crença: a "ação intencional" está relacionada aos desejos e crenças do agente, enquanto que a "intenção para agir" é vista como redutível a certos desejos e crenças. A vinculação entre elas se reduz ao problema da relação do complexo de desejos e crenças constituinte da "intenção para agir" e dos desejos e crenças necessários para "ação intencional", Bratman apud [GIR99].
- visão simplista: para executar uma ação intencional  $A$  é necessário ter a intenção de executar  $A$ ; os estados mentais do agente, no instante da ação, necessariamente devem conter  $A$  entre as coisas que ele intenciona, Bratman apud [GIR99].

Segundo [D'AM95], na visão de Bratman, a abordagem do modelo desejo-crença está errada, porque somos criaturas que constróem planos. Esses planos ajudam a guiar nossa conduta e coordenar nossa atividade no tempo, de uma forma que nossos desejos e crenças comuns não o fazem. Nesse contexto intenções são vistas como estados da mente distintos que não podem ser reduzidos a conjuntos de desejos e crenças.

Bratman também rejeita a abordagem de visão simplista e a considera como um caso especial da concepção mais geral, denominada visão de fenômeno simplista.

Na visão de fenômeno simplista, ação intencional e o estado mental de intenção envolvem um certo estado comum e, é a relação da ação com esse estado, que torna uma ação intencional. A visão simplista requer que esse estado seja uma intenção para executar a ação. As intenções são distintas e a intencionalidade de uma ação depende das suas relações com as intenções. Embora para executar *A* intencionalmente o agente necessite ter a intenção de fazer alguma coisa, não necessita ter a intenção de fazer *A*. Existe uma distinção entre o que o agente tem a intenção de executar e o potencial motivacional de sua intenção.

Uma intenção direcionada para o presente é aquela na qual se tem a intenção de realizar a ação imediatamente, enquanto em uma intenção direcionada para o futuro se tem a intenção de realizar a ação posteriormente.

[D'AM95] salienta que Bratman apresenta a idéia de intenção como um estado mental associado a planos de coordenação e à ação intencional. Intenções direcionadas para o futuro são elementos típicos de grandes planos e auxilia a coordenação das atividades ao longo do tempo. Para coordenar as atividades do agente, o plano deve ser consistente internamente, i.é, deve ser possível realizar o plano inteiro.

Segundo [MÓR97] e [GIR98], intenções, desejos e valoração são exemplos de pró-attitudes; crenças não são pró-attitudes. Pró-attitudes determinam um papel motivacional; que, na conjunção com crenças, podem levar o agente a agir. A maioria delas são apenas influenciadores potenciais de conduta, mas a intenção é uma pró-atitude controladora de conduta. Devido às intenções serem controladoras de conduta é que o agente pode esperar para tentar executá-las quando chegar a hora certa, se elas ainda persistirem e se o agente então souber o que fazer. Como as intenções são relativamente estáveis, o agente pode contar com que elas persistirão até o momento exato de agir.

Estas são as características que permitem explicar como as intenções dirigidas para o futuro normalmente auxiliam a coordenação, dando suporte às expectativas de sucesso associadas às suas execuções. As intenções dirigidas para o futuro deverão ser consistentes internamente e com as crenças do agente, bem como, dar suporte às crenças que serão executadas com sucesso. Em casos normais, quando o agente tem a intenção de *A*, ele também acredita que pode atingir *A*.

Resumindo, a intenção de um agente pode ser definida em vários níveis de abstração. Por exemplo, um agente pode ter a intenção de comprar um livro em particular, mas

não decidir em qual livraria irá comprá-lo. Então, uma estrutura de intenção é similar a um plano hierárquico, onde os operadores nos vários níveis da abstração em um plano e as intenções na estrutura de intenção têm propriedades parecidas. Um agente BDI, gradualmente, refina suas intenções em ações primitivas que podem ser executadas.

Na arquitetura descrita por Rao e Georgeff [RAO92], um agente BDI executa as seguintes funções:

1. Gera uma lista de opções. Estas opções são os meios para satisfazer intenções atuais e novas opções geradas, baseadas em crenças e desejos do agente. As opções disponíveis para o agente, as quais são meios para satisfazer suas intenções atuais, são geradas através de refinamento; parecido com uma função de refinamento de um plano hierárquico (ref. refinamento, ver figura 1). As opções disponíveis para o agente, que são alternativas para suas intenções atuais, são geradas com base em suas crenças e desejos atuais (ref. alternativas geradas, ver figura 1).
2. Seleciona um subconjunto dessas opções para adoção. A seleção é, geralmente, feita baseada em crenças, desejos e intenções atuais do agente. Estas novas intenções, isto é, as opções que o agente decidiu adotar, são adicionadas na estrutura de intenção.
3. Se existe uma ação atômica que pode ser executada dentro da estrutura de intenção, ela é, então, executada.
4. Então, se um agente satisfaz uma intenção ou decidiu que uma intenção não pode ser satisfeita mais adiante, tal intenção é abandonada.
5. Atualiza as crenças dos agentes e repete a partir do ponto 1.

Segundo [GIR99], é importante salientar que os estados mentais utilizados nesta abordagem (crenças, desejos, intenções e expectativas) funcionam como uma metáfora dos estados mentais humanos. Por exemplo, quando se coloca a crença de um aluno a respeito de alguma coisa (“acredita que falando alto vai ser ouvido”) na realidade está se colocando a crença que temos a respeito da crença que o aluno possui a respeito disto. Na verdade, o que acontece é a explicitação de uma crença que foi transmitida oralmente e não poderá ser comprovada “fisicamente”. Não se pode abrir a mente do aluno e observar isto. Justamente pelo fato de não possuímos um “extrator” automático de estados mentais a partir da observação e/ou análise de um diálogo é que se utilizam conjuntos de estados mentais inferidos/obtidos por observação. Sendo isto, uma grande crítica para esta abordagem. O

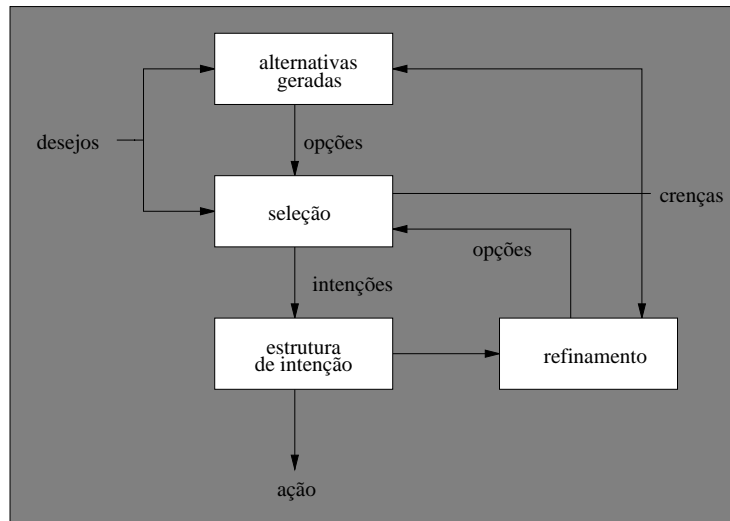


Figura 1: Arquitetura BDI

trabalho de Quaresma [QUA97] e do grupo onde está inserido (Universidade Nova de Lisboa<sup>4</sup>) apresenta uma proposta para identificação/extração de intenções a partir de diálogo. O fato de não podermos garantir que o conjunto de estados mentais é o utilizado (podendo serem outros, ex. no caso dos alunos no sistema MCOE [GIR99]) não invalida a proposta geral. O ambiente, a arquitetura e o kernel cognitivo processam os estados mentais independente destes serem fidedignos ou não. O que se busca é a aproximação mais estreita possível dos estados mentais vinculados às ações executadas pelos agentes.

As idéias subjacentes à abordagem social introduzem conceitos promissores para a compreensão de como construir agentes com habilidades especiais, tais como, aprender e ensinar. A aquisição do conhecimento pelos agentes numa sociedade multiagente emerge de suas interações, em que cada agente se comporta como um tutor e como um aluno. De fato não existe um tutor e um aluno, mas apenas atitudes de tutor e atitudes de aluno, conforme Corrêa, Coelho e Viccari apud [GIR99].

Segundo [SHO90], “o antropomorfismo<sup>5</sup> para construir os agentes artificiais é útil para a descrição da instância intencional, desde que nós entendamos suficientemente o mecanismo essencial, a fim de termos uma descrição mais simples da mecânica de seu comportamento”.

<sup>4</sup><http://www.di.fct.unl.pt>

<sup>5</sup>Aplicação a algum domínio da realidade (social, biológica, físico, etc...) de linguagem ou de conceitos próprios do homem ou do seu comportamento

### 3 Arquiteturas BDI

Ao modelarmos agentes utilizando estados mentais, não necessariamente estamos falando de agentes modelados com arquitetura BDI.

Shoham [SHO93] propõe um framework de programação orientada a agentes (POA). Ele apresenta uma classe de linguagem de agente, baseada no modelo do agente como "uma entidade na qual o estado é visto como consistindo de componentes mentais tais como crenças, capacidades, escolhas e comprometimentos". Shoham adota a noção de uma posição intencional como proposta por Dennett apud [MUL96].

Um sistema POA é definido por três componentes:

- uma linguagem formal para descrever estados mentais do agente;
- uma linguagem de programação com uma semântica correspondente a estados mentais;
- um *agentifier*<sup>6</sup>, i.é., um mecanismo para transformar um dispositivo naquilo que se pode chamar de agente, e assim construir uma ponte para a lacuna entre processos de máquina de baixo-nível e o nível intencional de programas de agentes.

A linguagem formal engloba as categorias mentais de crenças, obrigações e capacidades. Obrigação coincide com a noção de Rao e Georgeff [RAO92] de intenção e comprometimento. Capacidade não está diretamente representada como um conceito mental na arquitetura BDI; particularmente, ela é coberta por planos para executar certos objetivos.

A programação de agentes é vista como uma especificação de condições para a estruturação de comprometimentos. O controle de um agente é implementado por um interpretador genérico executado num loop de duas fases. Em cada ciclo, primeiro o agente lê as mensagens correntes e atualiza seus estados mentais; segundo, executa seus comprometimentos correntes, possivelmente resultando em modificações além de suas crenças.

A linguagem AGENTO[SHO93] é uma simples instância deste interpretador genérico. AGENTO pode modelar crenças, comprometimentos, capacidades e comunicação entre agentes. Essas atitudes são tratadas como estruturas de dados de um agente. A base da linguagem AGENTO inclui representações para fatos, ações incondicionais e condicionais (ambas podendo ser privadas ou comunicativas), e regras de comprometimento que descrevem condições que o agente assumirá como novos comprometimentos, baseados em seu

---

<sup>6</sup>identificador de agente

estado mental corrente e nas mensagens recebidas de outros agentes. Mensagens são estruturadas de acordo com o tipo da mensagem; tipos de mensagens aceitas em AGENTO são *inform*, *request* e *unrequest*. O interpretador de agente, correspondente, instancia o loop básico, fornecendo funções para atualizar crenças e compromentimentos. Crenças são atualizadas ou revisadas como sendo o resultado de ações que estão sendo executadas ou informadas. Compromentimentos são atualizados como sendo o resultado de mudança de crenças ou de mensagens do tipo *unrequest* recebidas por outros agentes.

AGENTO é uma linguagem muito simples que não constrói aplicações interessantes. AGENTO não se responsabiliza por motivação, i.é., como objetivos de agentes se envolvem na tomada de decisão (como o agente seleciona as opções alternativas).

A linguagem PLACA [THO93] e [THO95] estende AGENTO pela introdução de conhecimento sobre objetivos que o agente pode ter e refina o ciclo básico do agente por adição de um passo dependente de tempo (refinamento e construção de planos). A linguagem PLACA adota a visão de Bratman de planos como intenções, i.é., o agente tem um conjunto de planos, ou seja. suas intenções são descritas pelo subconjunto de planos do agente ao qual se comprometeu. Enquanto que PLACA, claramente, estende a expressividade da linguagem AGENTO por fornecer a noção de planos, ele não trata de outras restrições, tais como motivação, tomada de decisão e expressividade fraca da linguagem base.

Também inserida na abordagem de programação orientada a agentes proposta por Shoham, a LALO (LAngage d'Agents Logiciel Objet) é uma linguagem de programação orientada a agentes dentro de um *framework* para desenvolvimento de sistemas multiagentes inteligentes. LALO foi desenvolvido pelo CRIM<sup>7</sup> - *Centre de Recherche Informatique de Montreal*.

Mathias [MAT00], desenvolveu o SISMAT - Sistema de Matrícula Inteligente, que tem por objetivo auxiliar professores, coordenadores de curso e alunos na realização de matrículas de acadêmicos de uma universidade. O protótipo foi desenvolvido em LALO e serviu de exemplo do potencial do *framework*, onde podem ser modelados/implementados sistemas multiagentes com agentes reativos ou deliberativos.

A proposta e o trabalho de Shoham influenciaram muito a pesquisa na área de agentes cognitivos. Por esta razão, foram mencionados neste documento.

As seções seguintes apresentam exemplos de arquiteturas com a tríade (*Belief*, *Desire*, *Intention*) BDI.

---

<sup>7</sup><http://www.crim.ca>

## 3.1 Arquiteturas Deliberativas

### 3.1.1 Bratman, Israel e Pollack: IRMA

IRMA [BRA87] é uma arquitetura para agentes limitados por recursos, que selecionam seu curso de ação baseados em uma representação explícita de sua percepção, crenças, desejos e intenções. A arquitetura incorpora um número de módulos incluindo:

- uma estrutura de intenção, a qual é basicamente um conjunto ordenado por tempo;
- planos estruturados em árvore;
- um raciocinador de meios-fins;
- um analisador de oportunidade (procedimento que verifica alternativas para a ação a ser executada);
- um processo de filtragem;
- um procedimento de deliberação.

Tão logo as crenças do agente são atualizadas pela sua percepção, o analisador de oportunidade é capaz de sugerir opções para ação baseada nas crenças do agente. Mais adiante, opções são sugeridas pelo raciocinador de meios-fins, pertencente à estrutura de intenção atual do agente. Todas as opções disponíveis são executadas através do processo de filtragem, onde são testadas para verificar a consistência da estrutura de intenção atual. Por fim, as opções que passam pelo processo de filtragem com sucesso são enviadas para o processo de deliberação que modifica a estrutura da intenção adotando uma nova intenção, i.é., comprometimento com um plano parcial.

O modelo IRMA incorpora duas diferentes visões para planos:

- Os planos que são armazenados na biblioteca de planos podem ser vistos como crenças que o agente tem sobre quais ações são úteis para alcançar seus objetivos.
- O conjunto de planos adotados atualmente pelo agente definem sua estrutura intencional local.

Essa segunda visão, onde planos são associados às intenções, tornou-se o principal paradigma aceito na pesquisa em arquiteturas BDI.



IRMA adota uma posição pragmática em relação à arquitetura BDI. Em particular, ela não fornece um modelo formal explícito para crenças, objetivos e intenções, nem para seu processamento<sup>8</sup>. Assim, a principal contribuição de IRMA está na definição de um *framework* de controle para agentes BDI, os quais serviram como uma base para muitos refinamentos formais de conceitos na abordagem BDI.

### 3.1.2 Jennings: GRATE\*

Arquitetura em camadas na qual o comportamento de um agente é guiada por atitudes mentais de crenças, desejos e intenções conjunta.

O foco do trabalho de Jennings, na arquitetura GRATE\* [JEN92], foi em cooperação com sistemas inteligentes e, possivelmente, preexistentes, através de uma camada adicional de conhecimento cooperativo. A capacidade dos agentes em solucionar problemas foi estendida pelo compartilhamento de informações e tarefas entre eles. GRATE\* é uma arquitetura para o projeto (*design*) de interação dos solucionadores de problemas. Uma descrição geral de comportamento dos agentes cooperativos é representada pelo conhecimento embutido. Informação dependente de domínio sobre outros agentes é armazenada em estruturas de dados específicas (modelos de agentes).

GRATE\* consiste de duas partes distintas:

- Camadas de controle e de cooperação.
- Sistema a nível de domínio.

O sistema a nível de domínio pode preexistir ou ser construído propositalmente; o sistema fornece a funcionalidade de domínio necessária do solucionador de problema individual.

A camada de controle é um meta-controlador que opera no sistema a nível de domínio para garantir que suas atividades sejam coordenadas com outras existentes no sistema multiagente. Modelos de agentes armazenam diferentes tipos de conhecimento:

- o modelo de relação inclui conhecimento que o agente tem sobre outros agentes;
- o *self model* inclui uma perspectiva abstrata do sistema a nível de domínio local, i.é., capacidades e dons do agente.

As camadas de controle e cooperação consistem de três sub-módulos, representando a ação ou efeito recíproco entre comportamento cooperativo e local:

---

<sup>8</sup>Em [BRA87], Bratman fornece uma teoria de intenções baseada em uma abordagem filosófica.

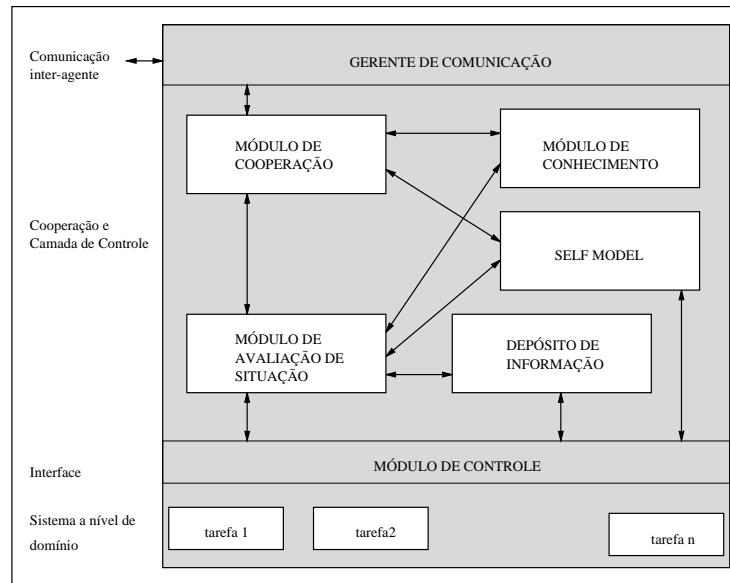


Figura 2: Arquitetura GRATE\*

- Módulo de controle: responsável pelo planejamento, execução e monitoração das tarefas locais.
- Módulo de cooperação: negocia processos de cooperação e coordenação com outros agentes.
- Módulo de avaliação de situação: forma a interface entre os mecanismos de controle social e local. Sendo assim, responsável pela decisão de escolher os métodos coordenados ou locais solucionadores de problemas.

Percebe-se, assim, que o foco do trabalho de Jennings se deu em processos de cooperação. Contudo, ele foi a procura de trabalhos que discutissem antes a definição de uma arquitetura em duas camadas do que cooperação embutida em um sistema a nível de domínio. A arquitetura não trata de questões sutis de comportamento de agentes, tais como reconciliar reatividade e deliberação. Particularmente, estes problemas são reservados para serem solucionados dentro do sistema a nível de domínio.

## 3.2 Arquiteturas Híbridas

### 3.2.1 Georgeff e Lansky: Procedural Reasoning System (PRS)

O Sistema de Raciocínio Procedural [GEO86];[GEO89] é uma arquitetura BDI genérica para representação e raciocínio de ações e procedimentos em ambientes dinâmicos, desenvolvida pela Stanford Research Institute International<sup>9</sup>.

A arquitetura PRS, como mostra a figura 3, pode ser subdividida em componentes periféricos e de raciocínio. Os componentes periféricos são:

- sensores;
- um monitor para traduzir informações dos sensores em crenças do agente;
- um gerador de comando para traduzir ações primitivas em comandos de executor;
- executores para gerar eventos no lado de fora do ambiente.

Os componentes de raciocínio são:

- um banco de dados contendo crenças correntes e fatos (expressos em lógica de primeira ordem) sobre o mundo que é automaticamente atualizado;
- uma biblioteca de planos (com representação simbólica explícita de crenças, desejos e intenções) chamada de área de conhecimento (KA - *Knowledge Area*), usada para executar/atingir objetivos ou reagir em determinadas situações; um plano pode ser ativado a partir de metas, dados ou de forma reativa, respondendo rapidamente a mudanças no ambiente;
- um conjunto de objetivos correntes (a arquitetura PRS usa o termo objetivo para se referir a desejos que são consistentes e que podem ser atingidos - viáveis);
- uma estrutura de intenção contendo um conjunto parcialmente ordenado de todos os planos (pilhas de áreas de conhecimento) selecionadas em tempo de execução;
- um interpretador (ou mecanismo de inferência) para: (i) selecionar planos aptos em resposta aos objetivos e crenças do sistema; (ii) confirmar os planos selecionados na estrutura de intenção e executá-los.

---

<sup>9</sup>SRI International - <http://www.sri.com>  
Computer Science Laboratory - <http://www.csl.sri.com>

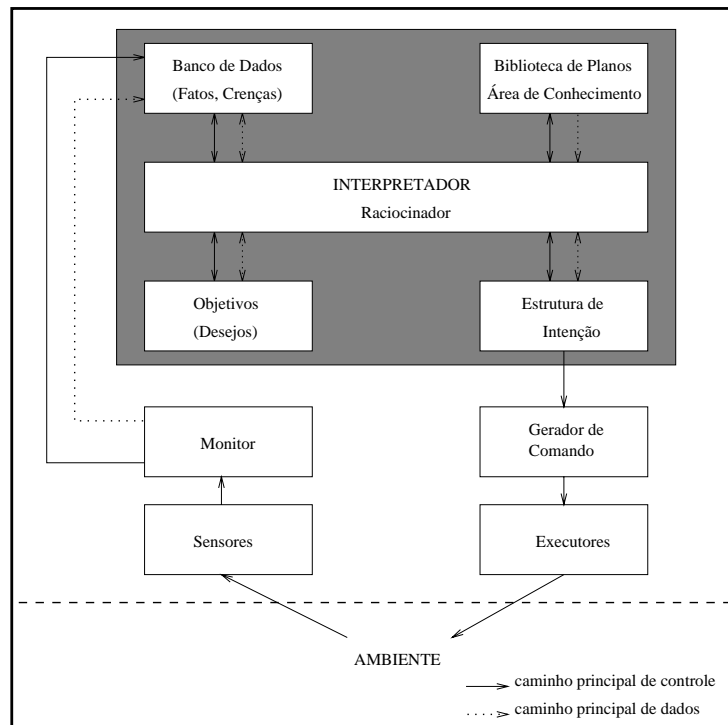


Figura 3: Arquitetura PRS

Em [GEO89], é apresentado um sistema PRS que foi utilizado no domínio de problema de manejo defeituoso<sup>10</sup> para Sistemas de Controle Reativo (RCS) de viagens espaciais<sup>11</sup> da NASA (*National Aeronautics & Space Administration*).

Uma viagem contém três tipos de sistemas: um de proa (extremidade dianteira da aeronave) e dois de popa (extremidade traseira da aeronave). São sistemas de propulsão relativamente complexos que são usados para controlar a posição/postura da viagem (ida e volta). O astronauta trata o mal funcionamento do sistema através da execução de procedimentos pré-específicos. Esses procedimentos podem ser vistos como planos não elaborados de ação e são projetados para serem executados em um ambiente complexo e de constante mudança.

O sistema decide entre diferentes cursos de ação dependendo de como as áreas de conhecimento (biblioteca de planos) foram executadas e qual tipo de prioridade tinham. Isso é uma forma fraca de planejamento e mais uma meta área de conhecimento (levando em conta tempo disponível, custos e benefícios) que pode ser aguardada para melhorar a confiabilidade/segurança do sistema.

<sup>10</sup> *handling malfunction*

<sup>11</sup> *shuttle*

As características da arquitetura PRS que contribuíram para o sucesso deste sistema foram:

- \* estratégia parcial de planejamento;
- \* reatividade;
- \* uso procedural de conhecimento;
- \* capacidades reflexivas de meta nível.

Em particular, a maneira pela qual o sistema integra raciocínio de meios-fim com o uso de conhecimento para decisão, é considerado um componente importante da atividade racional.

### 3.2.2 Burmeister e Sundermeyer: COSY

É uma arquitetura BDI híbrida que inclui elementos das arquiteturas PRS e IRMA.

A arquitetura de agente COSY [BUR92] descreve um agente através de seus comportamentos, recursos e intenções. O comportamento de um agente é classificado pela sua percepção, cognição, comunicação e execução, cada qual simulado por um específico componente na arquitetura COSY. Recursos incluem recursos cognitivos (conhecimento e crença), recursos de comunicação (protocolos de baixo nível e comunicação de hardware) e recursos físicos (o *gripper*<sup>12</sup> de um robô). Intenções são usadas num sentido diferente de Cohen e Levesque [COH90], existem:

1. intenções estratégicas modelando objetivos de longo alcance do agente, preferências, regras e responsabilidades e,
2. intenções táticas que estão, diretamente, ligadas às ações, representando um comprometimento do agente com seu curso/direção de ação escolhido.

Os módulos individuais (componentes) da COSY são:

- atuadores;
- sensores;
- comunicações;
- motivações;

---

<sup>12</sup>“agarradeira”

- cognição.

Os três primeiros, são módulos de domínio específico com suas funcionalidades intuitivas. O módulo de motivações implementa a intenção estratégica de um agente. O módulo de cognição avalia a situação atual e seleciona, executa, e monitora a ação do agente em determinada situação. Consiste de quatro subcomponentes: uma base de conhecimento, um componente de execução de *scripts*, um componente de execução de protocolo e um componente de decisão e raciocínio.

O conhecimento para solucionar o problema específico de uma aplicação é codificado em planos. Existem dois tipos de planos armazenados na biblioteca de planos:

- *scripts*: descrevem a direção esteriótipa de ação para executar determinados objetivos;
- protocolos de cooperação: descrevem padrões de comunicação.

*Scripts* são monitorados e executados pelo componente de execução de script (entregando a execução de comportamento primitivo para os atuadores) e protocolos para o componente de execução de protocolo. O componente de decisão e raciocínio é um mecanismo de controle geral, monitorando e administrando as decisões e raciocínios relativos a seleção de tarefas e seleção de planos, incluindo decisões e raciocínios relacionados a ramos de *intra-scripts* e *intra-protocolos*. Em [HAD95] Apud [MUL96], Haddadi forneceu um modelo teórico mais profundo estendendo o modelo BDI de Rao e Georgeff através da teoria de comprometimento e pelos mecanismos de definição, permitindo que agentes raciocinem sobre como explorar capacidades de cooperação por comunicação com outros agentes. Contudo, ainda há uma ampla lacuna e, parcialmente, não intuitiva entre a teoria de Haddadi e a atual implementação do modelo COSY.

COSY concentra-se em aspectos de cooperação entre agentes racionais. Oferece um mecanismo de planejamento simples. A idéia básica para usar a percepção e intenção, as quais guiam as decisões de um agente, é muito similar às idéias seguidas na arquitetura BDI. De qualquer forma, implementações recentes da COSY somente admitem seleção de estratégias muito simples e fazem suposições muito restritas. Por exemplo, uma ordem com prioridade total nas intenções estratégicas possíveis que um agente pode seguir. Além disso, não existe um conceito explícito de reatividade suportada pela arquitetura, embora a estrutura geral do módulo de cognição não seria um obstáculo do projeto (*design*) de tal mecanismo.

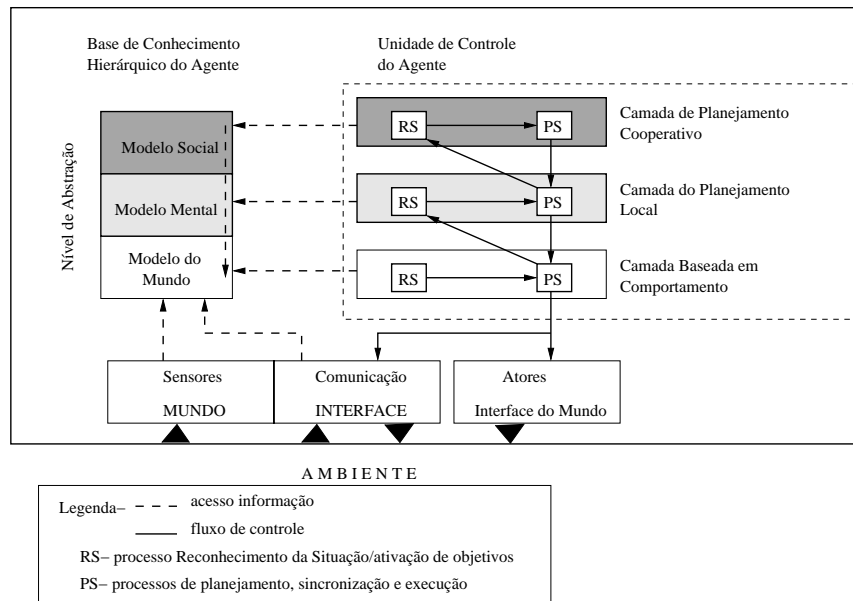


Figura 4: Arquitetura InteRRaP

### 3.2.3 Muller et al: InteRRaP

InteRRap [MUL96] é uma arquitetura em três camadas. As camadas representam um nível de abstração, da mais alta para a mais baixa. Estas camadas são subdivididas em mais duas camadas verticais: uma contendo bases de conhecimento e a outra contendo vários componentes de controle (ver figura 4).

Contudo, o foco da InteRRap foi estender o escopo das arquiteturas de controle de agentes através do suporte para interação de agentes. Para esta finalidade, a arquitetura oferece:

- uma camada de planejamento cooperativo;
- uma camada baseada em comportamento;
- uma camada de planejamento local.

A camada de planejamento cooperativo implementa um modelo de cooperação. Ela fornece protocolos de negociação e estratégias de negociação. Acionada pelas mensagens de controle a partir das camadas mais baixas, onde um agente pode decidir iniciar uma cooperação com outros agentes pela seleção de um protocolo e de uma estratégia. Em [MUL96], é exemplificado como robôs autônomos podem solucionar conflitos através de um plano em comum.

Em relação ao fluxo de controle, InteRRap abre mão de regras de controle global. Particularmente, utiliza dois mecanismos de controle hierárquico. Atividade é acionada de baixo para cima pela, assim chamada, solicitação de ativação para cima, enquanto que a execução é acionada de cima para baixo (mensagem de compromisso para baixo). Isto é, uma camada de controle na InteRRap se tornará ativa somente se a próxima camada mais inferior não puder negociar com a situação. Este controle baseado em competência permite que um agente reaja adequadamente em uma determinada situação, ou pelos padrões de comportamentos, ou pelo planejamento local, ou ainda pela cooperação. Os produtos resultantes (*output*) da execução da camada de planejamento cooperativo são planos com comandos de sincronização embutidos; estes são passados para a camada de planejamento local, na qual *os outputs* chamam procedimentos padrões de comportamento nos componentes baseados em comportamento, que enfim produzem ações.

O rígido controle na arquitetura InteRRap, consideravelmente, simplifica o projeto (*design*); uma arquitetura mais flexível, por exemplo, permite ativações concorrentes das diferentes camadas de controle. Assim, seria necessário a utilização de mecanismos sofisticados de coordenação e de antecipação de ações.

As arquiteturas descritas nesta seção possuem aplicações variadas e algumas delas em domínios no mundo real, conforme já explicitado anteriormente.

Para que estas aplicações fossem construídas, foram criadas ferramentas e/ou ambientes para viabilizar suas implementações. A seção a seguir, apresenta as ferramentas disponíveis para construção de agentes BDI.

## 4 Ferramentas para modelagem de agentes com BDI

As ferramentas apresentadas a seguir, conforme seus idealizadores, foram desenvolvidas com a intenção comum de diminuir a distância que existe entre teorias formais para especificação de agentes cognitivos e aplicações, tais como arquiteturas abstratas para construção de agentes de software e linguagens orientadas a agente.

### 4.1 AgentSpeak(L)

AgentSpeak(L)[RAO96] é uma linguagem que pode ser visualizada como uma abstração da arquitetura PRS (seção 3.2.1) e que permite que programas baseados em agentes sejam escritos e interpretados de maneira similar à programas lógicos baseados em *cláusulas de horn*. É uma versão textual e simplificada da linguagem dMARS.



Um agente modelado na linguagem AgentSpeak(L) possui um conjunto de crenças, desejos, regras de planos, intenções e eventos que constituem seu estado mental.

AgentSpeak(L) é uma linguagem baseada em regras que expressam o conhecimento do agente e estas, são usadas em planos. A leitura informal dessas regras é: "se o agente quer realizar o objetivo  $O$  e acredita que a situação  $S$  é o caso, então o plano  $P$  pode ser a coisa correta a fazer".

O comportamento do agente (i.é., suas interações com o ambiente) é comandado por programas escritos nesta linguagem. As crenças, desejos e intenções de um agente não são representados explicitamente como fórmulas modais. Em vez disso, projetistas (*designers*) podem relacionar estas noções a agentes escritos em AgentSpeak(L). O estado corrente de um agente, seu ambiente e outros agentes, podem ser vistos como seu estado de crença atual; ou os estados que o agente quer causar baseados em seus estímulos internos e externos podem ser vistos como desejos; e a adoção de programas para satisfazer tais estímulos podem ser vistas como intenções. Essa troca de perspectiva, onde uma simples linguagem de especificação é tratada como um modelo de execução de um agente e então, relacionada às atitudes mentais (crenças, desejos e intenções), a partir de um ponto de vista externo, é provavelmente a melhor chance de unificar teoria e prática.

AgentSpeak(L) como linguagem de especificação consiste de um conjunto de crenças básicas (ou fatos, no sentido de programação lógica) e um conjunto de planos. Planos são sensíveis ao contexto, são receitas invocadas por eventos que permitem decomposição hierárquica de objetivos, bem como a execução de ações. Embora que sintaticamente planos pareçam cláusulas absolutas de linguagem de programação lógica, eles são bastante diferentes em seus comportamentos. Em tempo de execução um agente pode ser visto como consistindo de um conjunto de crenças, planos, intenções, eventos, ações e funções de seleção.

## 4.2 dMARS<sup>TM</sup>: distributed Multi-Agent Reasoning System

dMARS<sup>TM</sup>[d'IN98] é uma ferramenta integrante de um ambiente para desenvolvimento de agentes de software<sup>13</sup>, baseada em C++. Foi desenvolvida no AAI ( *Australian Artificial Intelligence Institute*<sup>14</sup> ), sendo uma ferramenta comercial voltada para aplicações em domínios dinâmicos, de conhecimento incerto e complexo, tais como telecomunicações, viagens espaciais, tráfego aéreo, gerência de negócios, etc. Projetada para configuração

---

<sup>13</sup>sistemas orientados a agentes

<sup>14</sup><http://www.aai.com.au>

rápida e facilidade de integração, o ambiente facilita o projeto de sistema, manutenção e reengenharia.

Baseada na arquitetura PRS, dMARS leva vantagem nas recentes pesquisas de raciocínio para sistemas multiagentes em tempo real. O Procedural Reasoning System (PRS), apresentado na seção 3.2.1, é talvez a arquitetura de agente mais conhecida. Tanto PRS como seu sucessor dMARS são exemplos da abordagem BDI (ver seção 2). A figura 3 mostra a arquitetura BDI, tipicamente, contendo quatro estruturas-chaves de dados: crenças, objetivos, intenções e uma biblioteca de planos.

Como apresentado na seção 2, uma crença de um agente corresponde às informações que o agente tem sobre o mundo, que poderia ser incompleta ou incorreta. Crenças podem ser como simples variáveis (como por exemplo, na linguagem PASCAL), mas agentes modelados na arquitetura BDI representam crenças de forma simbólica (como por exemplo, fatos em PROLOG). Um desejo de agente (ou objetivos em um sistema) intuitivamente corresponde à tarefas estabelecidas pelo próprio agente. (Agentes BDI exigem que desejos sejam logicamente consistentes, embora desejos humanos falhem freqüentemente). A intuição em sistemas BDI é que o agente não é capaz, geralmente, de realizar todos os seus desejos, mesmo sendo consistentes. Agentes devem, portanto, estabelecer alguns subconjuntos de desejos disponíveis e comprometer recursos para realizá-los. Estes desejos escolhidos são intenções e um agente continuará a tentar realizar uma intenção até que acredite que a intenção foi satisfeita, ou acredite que a intenção não conseguirá ser realizada.

O modelo BDI é operacionalizado em agentes dMARS através de planos. Cada agente tem uma biblioteca de planos (receitas/fórmulas) especificando o curso da ação que pode ser tomado pelo agente para realizar suas intenções. Uma biblioteca de planos de um agente representa seu conhecimento procedural ou seu know-how (conhecimento sobre como fazer).

Segundo [d'IN98], cada plano contém vários componentes:

1. O *trigger*<sup>15</sup> descreve as circunstâncias sob as quais o plano poderia ser considerado. Geralmente considerado em termos de eventos. Por exemplo, o plano "tomar chá" pode ser disparado pelo evento "sede".
2. Além disso, um plano tem um contexto ou pré-condições, especificando as circunstâncias sob as quais a execução de um plano pode iniciar. Por exemplo, o plano "tomar chá" pode ter a pré-condição "ter saquinhos de chá".

---

<sup>15</sup>sinal que causa o começo de uma certa atividade ou a condição de invocação para um plano

3. Um plano também pode ter condições de manutenção que caracterizam as circunstâncias que devem permanecer verdadeiras enquanto o plano é executado.
4. Finalmente, um plano tem um corpo, definindo um curso bastante complexo de ação, que pode ser consistido de objetivos (ou sub-objetivos) e ações primitivas. Nosso plano "chá" pode ter o corpo "ferver a água"; "colocar o saquinho de chá no copo"; "adicionar água no copo". Nesse sentido, "ferver a água" é um sub-objetivo (alguma coisa que deve ser realizada/executada quando a execução do plano alcança este ponto no plano), enquanto que "colocar o saquinho de chá no copo" e "adicionar água no copo" são ações primitivas, i.é., ações que podem ser executadas diretamente pelo agente. Ações primitivas podem ser pensadas como sendo "chamadas a procedimentos".

Agentes dMARS monitoram tanto o mundo e seus próprios estados internos quanto quaisquer eventos que são percebidos e colocados em uma fila de eventos. O interpretador, na figura 3, é responsável por controlar/gerenciar a operação geral do agente. O interpretador executa o seguinte ciclo:

- observa o mundo e o estado interno do agente e atualiza a fila de eventos para repercutir nos eventos que foram observados;
- gera novos desejos possíveis (tarefas), a partir de planos encontrados com eventos compatíveis na lista de eventos;
- seleciona, deste conjunto de planos compatíveis, um evento para execução;
- coloca o evento escolhido em uma pilha nova ou já existente (*stack*) de intenção, dependendo se o evento é ou não é um sub-objetivo;
- seleciona uma pilha de intenção, pega o plano mais do topo e executa o próximo passo do plano corrente: se o passo for uma ação, executa esta ação; de outra forma, se for um sub-objetivo, envia este sub-objetivo para a fila de eventos.

Dessa maneira, quando um plano inicia a execução, seus sub-objetivos são colocados na fila de eventos, onde, um por um, acionam planos que realizam seu sub-objetivo, tornando-o ativo. Esta é a execução básica do modelo de agentes modelados em dMARS. Agentes

não planejam tudo, pois planos devem ser gerados pelo programador do agente em tempo de projeto. O planejamento feito pelos agentes consiste, inteiramente do desenvolvimento/expansão dos sub-objetivos sensíveis a contexto. Sub-objetivos são adiados até o momento em que são selecionados para a execução.

dMARS é adequada para o desenvolvimento de aplicações que exigem comportamento direcionado a objetivos pró-ativos e respostas confiáveis em um tempo-crítico para alterar. É, particularmente, adequada para aplicações onde um número enorme de procedimentos bem definidos mas complexos ou táticas que existem para executar tarefas particulares numa variedade de situações.

Conforme seus idealizadores, dMARSs foi projetada com características de robustez, eficiência e uma possível extensão para usuários. Fornece uma sofisticada ferramenta gráfica para desenvolvimento e *debugging*. Essa ferramenta não só fornece uma interface intuitiva, mas trata de questões específicas que envolvem o desenvolvimento em larga escala.

dMARS segue o paradigma de orientação a agente, o qual modela sistemas em termos de objetos de softwares conhecidos como agentes. Por agentes serem entidades que raciocinam, capazes de perseguir metas e realizar objetivos, mesmo em ambientes incertos e dinâmicos, essa abordagem facilita o desenvolvimento de sistemas complexos, distribuídos e tolerantes à falha.

Agentes dMARS modelam suas especialidades como um conjunto de planos sensíveis a contexto. Esses planos podem reagir para mudar no ambiente e seguir proativamente os objetivos do agente. Isso permite que os agentes permaneçam direcionados a objetivos sem comprometer suas habilidades de reação para novas situações. Os agentes podem modificar seus comportamentos para conservar um alinhamento das mudanças ocorridas no ambiente em tempo real.

A abordagem orientada a agente de dMARS fornece um significado natural, intuitivo, robusto e flexível de modelagem para tomada de decisão em tempo real. Concorrência e cooperação são suportadas mesmo quando os agentes envolvidos são distribuídos pela rede.

Além de raciocinar sobre seu ambiente, agentes dMARS são capazes de raciocinar sobre seu próprio estado mental (meta raciocínio). Isto é, eles podem refletir sobre suas próprias crenças, metas e intenções. Essa característica permite que usuários definam uma linha mais larga de procedimentos escalonados para tarefas.

### 4.3 X-BDI: eXecutable BDI

O X-BDI [MÓR00] é uma ferramenta que permite a descrição formal de agentes cognitivos, sendo ao mesmo tempo, uma linguagem para implementação de agentes. Segundo [MOR00], sua proposta disponibiliza um sistema formal cuja linguagem é adequada para a representação de conhecimento. Esse sistema fornece suporte para diversos tipos de raciocínios, tratados de forma computacional, além de fornecer as ferramentas necessárias para se modelar os estados mentais da tríade BDI. Maiores detalhes podem ser obtidos em [MÓR97]; [MÓR98]; [MÓR00].

O modelo possui duas características :

- serve como ambiente de especificação de agentes, em que é possível definir formalmente um agente através da descrição dos seus estados mentais (crenças, desejos e intenções) e de suas respectivas propriedades. Este ambiente pode, também, fazer a verificação destas propriedades;
- serve como um ambiente de implementação de agentes. O X-BDI é um ambiente onde se formaliza o agente e se o executa. Além disso, pode-se verificar se o comportamento desejado ocorreu ou não.

O X-BDI possui um ambiente de implementação com alto nível de abstração (os estados mentais), o que reduz a complexidade no desenvolvimento de sistemas baseados em agentes. Isto significa que o projetista da área de domínio (aplicação) apenas tem de colocar suas heurísticas a respeito do processo a ser modelado, a fim de guiar o seu comportamento. A linguagem do X-BDI possui baixa complexidade no que diz respeito à sintaxe, facilitando sobremaneira o trabalho de especificação e implementação de agentes baseados na arquitetura BDI. A arquitetura BDI passa a ser um paradigma de implementação de agentes cognitivos, devido à possibilidade de se poder excetuar o modelo formal.

Segundo [MÓR00], o X-BDI, como modelo formal de agentes, reduz a distância entre especificação e implementação de agentes. Adotando o ELP (*Extended Logic Programming*) como formalismo de apoio, preserva-se tanto as principais características dos modelos formais (habilidade para definir e verificar formalmente agentes), como fornece mecanismo para agentes raciocinarem. Além do que, o modelo X-BDI tem a vantagem de modelar aspectos estáticos e dinâmicos de estados mentais pró-ativos.

## 5 Considerações Finais

A especificação, projeto, verificação e aplicações de agentes BDI têm recebido grande atenção nos últimos anos. Agentes BDI têm sido investigados por muitos pesquisadores, tanto do lado da perspectiva de especificação teórica como da perspectiva de projeto prático. Algumas dessas pesquisas tentam diminuir a distância que existe entre teorias formais para especificação de agentes cognitivos e aplicações, tais como arquiteturas abstratas para construção de agentes de software e linguagens orientadas a agente.

No presente trabalho, apresentamos três linguagens que se propõem em reduzir a distância entre prática e teoria. Constatamos que dMARS apresenta certas vantagens em relação as linguagens de programação orientadas a agentes apresentadas, pois é uma ferramenta comercial testada, comprovada e já utilizada para aplicações de domínios, tais como telecomunicações, viagens espaciais, tráfego aéreo, gerência de negócios e inserida num grupo de pesquisa com altos investimentos financeiros.

Por outro lado, o X-BDI é uma ferramenta mais recente e em fase de finalização. O X-BDI ainda não é um produto e sua utilização necessita que uma série de refinamentos e recursos sejam incorporados.

Para usar o X-BDI não existe, neste momento, uma interface mais amigável e uma documentação organizada na forma de um manual que auxilie o projetista a utilizar a ferramenta.

Os trabalhos que utilizam o X-BDI como ferramenta para construção de agentes são o MCOE [GIR99] e a proposta de extensão da metodologia M-DRAP utilizando estados mentais [ROZ00].

O MCOE de [GIR99] utilizou o X-BDI exatamente para definir o *kernel cognitivo* do agente, i.é., a parte de deliberação (raciocínio) do agente. É, justamente, para isto que o X-BDI foi concebido.

Um agente modelado/implementado com o X-BDI requer um entorno (interface) para permitir as entradas (crenças, desejos e outras informações relevantes) e a saída (resultado do processo de deliberação) que são os planos a serem executados.

No trabalho de [GIR99] pode-se observar isto, claramente, quando é proposta uma arquitetura multiagente que trata a questão da seleção de múltiplas estratégias de ensino, levando em consideração a modelagem cognitiva de cada aluno que interage com o sistema.

Dada a importância da ferramenta X-BDI no contexto da pesquisa em agentes modelados com BDI, a realização deste trabalho permitiu organizar o referencial teórico e a base para construção do PEP (Plano de Estudos e Pesquisa) onde se pretende detalhar

a proposta para construção de um “Editor de Estados Mentais” associado a ferramenta X-BDI. Este editor visa facilitar a explicitação/declaração dos estados mentais dos agentes que serão necessários para coreografia. Além disso, o editor incorporará funções de auxílio para o usuário e fará a mediação entre o ambiente e o *kernel cognitivo* propriamente dito. [BRO99]

## Referências

- [ALV97] ALVARES, L.O.; SICHMAM, J.S. Introdução aos sistemas multiagentes. In: JORNADA DE ATUALIZACAO EM INFORMATICA, 16, 1997, Brasília, DF. **Anais...** Brasília: UnB, 1997.
- [BRA87] BRATMAN, M.; ISRAEL, D.; POLLACK, M. **Toward an architecture for resource-bounded agents**. Stanford: Stanford University, 1987.
- [BRA84] BRATMAN, M. Two faces of intention. **The Philosophical Review**, v.93, n.3, p.275–405, 1984.
- [BRA89] BRATMAN, M. What is intention? In: COHEN, P; MORGAN, J; POLLACK, M. (Eds.), 1989. **Anais...** MIT Press, 1989.
- [BRO99] BROPHY, S.; BISWAS, G.; KATZLBERGER, T.; BRANSFORD, J.; SCHWARTZ, D. Teachable agents: combining insights from learning theory and computer science. In: LAJOIE, S.P.; VIVET, M. (Eds.). **Artificial intelligence in education**. Amsterdam, Berlin, Oxford, Tokyo and Washington DC: IOS Press, 1999. v.50, p.21–27.
- [BUR92] BURMEISTER, B.; SUNDERMEYER, K. Cooperative problem-solving guided by intentions and perception. In: DEMAZEAU, Y.; WERNER, E. (Eds.). **Decentralized AI**. Kaiserslautern, Germany: North-Holland, 1992. v.3.
- [COH90] COHEN, P.R.; PERRAULT, C.P. Intention is choice with commitment. **Artificial Intelligence**, 42(3), 1990.
- [COR94] CORRÊA, M. **A arquitetura de diálogos entre agentes cognitivos distribuídos**. Rio de Janeiro: UFRJ, 1994. Tese de Doutorado.

- [D'AM95] D'AMICO, C. **Modelo de usuário para sistemas tutores inteligentes**. Porto Alegre: CPGCC/UFRGS, 1995.
- [DEM90] DEMAZEAU, Y.; MULLER, J. Decentralised artificial intelligence. In: EUROPEAN WORKSHOP ON MODELLING AUTONOMOUS AGENTES IN A MULTI-AGENT WORLD, 1990, Cambridge. **Proceedings...** North-Holland, 1990.
- [DEN87] DENNETT, D. **The intentional stance**. Cambridge, MA: MIT Press, 1987.
- [d'IN98] d'INVERNO, M.; KINNY, D.; LUCK, M.; WOOLDRIDGE, M. A formal specification of dMARS. In: INTELLIGENT AGENTS IV: 4th INTERNATIONAL WORKSHOP ON AGENT THEORIES, ARCHITECTURES AND LANGUAGES, 1998. **Proceedings...** Springer-Verlag, 1998.
- [GEO89] GEORGEFF, M.; IGRAND, F. Decision-making in an embedded reasoning system. In: 11th INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 1989, Detroit, Michigan. **Proceedings...** Ames Research Center, 1989.
- [GEO86] GEORGEFF, M.; LANSKY, A. Procedural knowledge. In: SPECIAL ISSUE ON KNOWLEDGE REPRESENTATION, 1986. **Proceedings...** IEEE, 1986. v.74, p.1383–1398.
- [GIR99] GIRAFFA, L. M. M. **Uma arquitetura de tutor utilizando estados mentais**. Porto Alegre: CPGCC/UFRGS, 1999. Tese de Doutorado.
- [GIR98] GIRAFFA, L.M.M.; VICCARI, R.M. The use of agents techniques on intelligent tutoring systems. In: 18th International Conference of the Chilean Computer Science Society (SCCC98), 18, 1998, Antofagasta, Chile. **Proceedings...** IEEE, 1998.
- [HAD95] HADDADI, A. **Reasoning about cooperation in agent systems: a pragmatic theory**. Manchester: UMIST, 1995. Tese de Doutorado.
- [JEN92] JENNINGS, N. R. Towards a cooperations knowledge level for collaborative problem solving. In: 10th EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE, 1992, Viena. **Proceedings...** 1992. p.224–228.



- [MAT00] MATHIAS, I. M. **SISMAT sistema de matrícula inteligente**. Curitiba: CP-GI - UFPR, 2000. Dissertação de Mestrado.
- [MÓR97] MÓRA, M. et al. Modelling dynamic aspects of intentions. In: PORTUGUESE CONFERENCE ON ARTIFICIAL INTELLIGENCE, 1997. **Proceedings...** Springer-Verlag, 1997.
- [MÓR98] MÓRA, M.; LOPES, J.G.; COELHO, J.G.; VICCARI, R. BDI models and systems: reducing the gap. In: AGENTS THEORY, ARCHITECTURE AND LANGUAGES WORKSHOP, 1998, Canarias. **Proceedings...** Springer-Verlag, 1998.
- [MÓR00] MÓRA, M. **Um modelo de agente executável**. Porto Alegre: CPGCC/UFRGS, 2000. Tese de Doutorado.
- [MOU96] MOUSSALE, N.M.; VICCARI, R.M.; CORRÊA, M. Intelligent tutoring systems modelled through the mental states. In: SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICIAL - SBIA, 1996, Berlin. **Proceedings...** Springer-Verlag, 1996.
- [MUL96] MULLER, J. **The design of intelligent agents: a layered approach**. New York: Springer, 1996.
- [OLI96] OLIVEIRA, F. Inteligência artificial distribuída. In: IV ESCOLA REGIONAL DE INFORMÁTICA, 3, 1996, Canoas, RS. **Anais...** Sociedade Brasileira de Computação, 1996.
- [QUA97] QUARESMA, P. **Inferência de atitudes em diálogos**. Lisboa, Portugal: Faculdade de Ciências e Tecnologia, 1997. Tese de Doutorado.
- [RAO92] RAO, A.S.; GEORGEFF, M.P. An abstract architecture for rational agents. In: INTERNATIONAL CONFERENCE ON PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING - KR, 3., 1992. **Proceedings...** Morgan Kaufman, 1992.
- [RAO95] RAO, A.S.; GEORGEFF, M.P. BDI agents: from theory to practice. In: FIRST INTERNATIONAL CONFERENCE ON MULTIAGENT SYSTEMS, 1995. **Proceedings...** 1995. p.312-319.

- [RAO96] RAO, A.S. AgentSpeak(L): BDI agents speak out in logical computable language. In: 7th EUROPEAN WORKSHOP ON MODELING AUTONOMOUS AGENTS IN A MULTI-AGENT WORLD, MAAMAW'96, 1996, Eindhoven, The Netherlands. **Proceedings...** Springer, 1996. p.42–55.
- [ROZ00] ROZA, M. P. **Uma proposta de extensão da metodologia M-DRAP utilizando estados mentais**. Porto Alegre: PPGCC/PUCRS, 2000. Relatório do Seminário de Andamento.
- [RUS96] RUSSEL, S.; NORVIG, P. **Artificial intelligence: a modern approach**. New Jersey: Prentice Hall, 1996.
- [SEA69] SEARLE, J. R. **Speech acts: an essay in the philosophy of language**. Cambridge: Cambridge University Press, 1969. Tese de Doutorado.
- [SHO90] SHOHAM, Y. **Agent-oriented programming**. Stanford: CSD-Stanford University, 1990. Tese de Doutorado.
- [SHO93] SHOHAM, Y. Agent-oriented programming. **Artificial Intelligence**, Amsterdam, v.60, 1993.
- [THO93] THOMAS, S. R. **PLACA, an agent oriented programming language**. Stanford: CSD-Stanford University, 1993. Tese de Doutorado.
- [THO95] THOMAS, S. R. The PLACA agent programming language. In: WJ95, 1995. **Proceedings...** 1995. p.355–370.
- [WOO95] WOOLDRIDGE, M.; JENNINGS, N.R. **Intelligent agents: theory and practice**. London: QM and Wc, 1995.