

UNIVERSIDADE FEDERAL DE PELOTAS
INSTITUTO DE FÍSICA E MATEMÁTICA
DEPARTAMENTO DE INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**Um Ambiente para Desenvolvimento
de Agentes BDI**

por
MOSER FAGUNDES

Trabalho de Conclusão de Curso

Prof. Ricardo Azambuja Silveira, Dr.
Orientador

Pelotas, setembro de 2004.

SUMÁRIO

LISTA DE FIGURAS.....	3
LISTA DE TABELAS.....	4
LISTA DE ABREVIATURAS.....	5
RESUMO.....	6
1 INTRODUÇÃO.....	7
1.1 Motivação.....	8
1.2 Objetivos.....	9
1.3 Organização do Trabalho.....	9
2 AGENTES INTELIGENTES.....	11
2.1 Arquiteturas de Agentes.....	13
2.2 Ambientes.....	15
3 MODELO BDI.....	16
3.1 Raciocínio Prático.....	16
3.2 Estados Mentais.....	17
3.3 Agentes BDI.....	20
3.4 Arquiteturas.....	22
4 MODELO PROPOSTO.....	26
4.1 Estados Mentais.....	26
4.2 Processos.....	30
5 IMPLEMENTAÇÃO DO AMBIENTE.....	35
5.1 Tecnologias Utilizadas.....	35
5.2 Implementação.....	36
6 APLICAÇÃO DESENVOLVIDA.....	39
6.1 Desejos.....	39
6.2 Projeto.....	41
6.3 Agente.....	44
6.4 Funcionamento da Aplicação.....	45
7 CONCLUSÕES.....	47
8 REFERÊNCIAS BIBLIOGRÁFICAS.....	48

LISTA DE FIGURAS

FIGURA 1 – Agente e seu ambiente.....	12
FIGURA 2 – Agente reativo proposto por Brooks.....	14
FIGURA 3 – Arquitetura IRMA.....	22
FIGURA 4 – Arquitetura PRS.....	24
FIGURA 5 – Arquitetura proposta.....	26
FIGURA 6 – Processamento da percepção.....	31
FIGURA 7 – Manutenção das Intenções.....	32
FIGURA 8 – Processo Deliberativo.....	33
FIGURA 9 – Estrutura de classes.....	37
FIGURA 10 – Classe Bebida.....	41
FIGURA 11 – Instâncias da classe Cliente.....	42
FIGURA 12 – Instâncias da classe Desire.....	42
FIGURA 13 – Instâncias das condições dos desejos.....	43
FIGURA 14 – Código resumido do agente BALBINO.....	44

LISTA DE TABELAS

TABELA 1 – <i>Slots</i> das classes criadas no Protégé.....	34
---	----

LISTA DE ABREVIATURAS

API	<i>Application Program Interface</i>
BDI	<i>Belief-Desire-Intention</i>
FIPA	<i>Foundation for Intelligent Physical Agents</i>
FIPA-OS	<i>FIPA Open Source</i>
IA	<i>Inteligência Artificial</i>
IRMA	<i>Intelligent Resource-bounded Machine Architecture</i>
JADE	<i>Java Agent Development Framework</i>
KA	<i>Knowledge Area</i>
PRS	<i>Procedural Reasoning System</i>

RESUMO

O presente trabalho está situado na área dos Agentes Inteligentes, e enfoca as arquiteturas cognitivas, mais precisamente um subconjunto das mesmas, chamado BDI (*Belief, Desire and Intention*). As arquiteturas BDI são baseadas em estados mentais, e têm sua origem na teoria de raciocínio prático humano.

O desenvolvimento de sistemas através de uma abordagem mentalística consiste basicamente em descrever os estados mentais do agente, bem como a interação entre estes. No entanto, existe um número bastante restrito de ferramentas que desempenham este papel.

Este trabalho apresenta uma proposta de uma arquitetura compatível com o modelo BDI, bem como a implementação de um ambiente para desenvolvimento de agentes baseado nesta arquitetura.

1 INTRODUÇÃO

O presente trabalho está situado na área dos Agentes Inteligentes, os quais são definidos como sistemas capazes de perceber, através de sensores, e agir, através de atuadores, em um dado ambiente [RUS95]. Segundo Michael Wooldridge [WOO00], chamamos tais sistemas de agentes porque eles são capazes de agir de maneira autônoma e independente de modo a atingir os seus objetivos, e os chamamos de inteligentes porque possuem atributos de inteligência.

O desenvolvimento de aplicações baseadas na abordagem orientada a agentes possibilita um maior nível de abstração, uma vez que o mundo no qual vivemos está repleto de agentes. Tal nível de abstração não é usual nas metodologias tradicionais de modelagem e implementação, isto é, torna explícitas, no sistema, certas funcionalidades que antes ficavam apenas subentendidas [ZAM02]. No entanto, a maior parte das aplicações de sistemas de computação é algorítmica e trabalha com informações perfeitas. Entretanto, no mundo real são necessários sistemas bastante complexos – imersos em ambientes que mudam constantemente, com acesso a informações parciais, e onde a incerteza prevalece. Além disso, a frequência com que os sistemas mudam de comportamento está cada vez maior, requisitando arquiteturas e linguagens que reduzam a complexidade e o tempo de especificação e modificação [GEO99].

Uma das principais características de um agente é a capacidade de viver em sociedade, sendo que a interação entre os mesmos é tópico de estudo dos sistemas multiagentes. Uma outra perspectiva se preocupa com a estrutura interna do agente. Wooldridge [WOO02] distingue os dois casos em macro e micro, respectivamente. Abordaremos a segunda perspectiva, que corresponde a arquitetura. Chamamos de arquitetura de um agente a descrição dos processos internos que regem a interação do mesmo com o seu ambiente. Em relação a arquitetura, a literatura relacionada costuma a classificar os agentes em dois grandes grupos: cognitivas e reativas. As arquiteturas cognitivas, também conhecidas como deliberativas, possuem uma representação simbólica do ambiente, possibilitando a execução de um raciocínio lógico sobre as informações armazenadas internamente. Em contrapartida, as

arquitecturas reativas obtêm seu comportamento inteligente a partir da interação mantida com o ambiente, não necessitando de representação simbólica do mesmo. Existem também as arquitecturas híbridas, que como o próprio nome sugere, contemplam aspectos inerentes às duas arquitecturas anteriormente citadas.

Este trabalho enfoca as arquitecturas cognitivas, mais precisamente um subconjunto das mesmas, chamado BDI (*Belief, Desire and Intention*). As idéias básicas da abordagem BDI consistem em descrever o processamento interno do estado de um agente utilizando um conjunto de categorias mentais (crenças, desejos e intenções) e definir uma arquitectura de controle através da qual o agente seleciona racionalmente o curso de suas ações [GIR99].

A teoria BDI sobre ação racional humana foi originalmente desenvolvida pelo filósofo Michael Bratman [BRA87 apud WOO00]. É uma teoria sobre raciocínio prático, o qual consiste em ponderar considerações conflitantes a favor e contra alternativas competitivas, onde as considerações relevantes são determinadas pelos desejos e crenças do agente.

Desde a sua proposição, o paradigma BDI foi incorporado em diversas arquitecturas, dentre as quais podemos citar a IRMA (*Intelligent Resource-bounded Machine Architecture*) [BRA88] e a PRS (*Procedural Reasoning System*) [GEO89]. A grande quantidade de arquitecturas e suas respectivas implementações, juntamente com aplicações significantes como controle de tráfego aéreo e diagnóstico de falhas em viagens espaciais, evidencia o sucesso do modelo.

1.1 Motivação

O crescente número de aplicações complexas, que são executadas em um ambiente dinâmico e com informações parciais ou imprecisas, têm exigido abordagens alternativas para o seu desenvolvimento. Várias técnicas para modelagem de sistemas elaboradas pela IA, como os agentes inteligentes, têm se mostrado bastante adequadas para tais fins. Os agentes inteligentes por sua vez, consistem em uma área bastante vasta, que originou diversas linhas de pesquisa, dentre as quais podemos citar o estudo de arquitecturas internas de agentes baseadas em estados mentais, que tem como mais conhecido e bem sucedido exemplo o modelo BDI.

O desenvolvimento de sistemas através de uma abordagem mentalística consiste basicamente em descrever os estados mentais do agente, bem como a interação entre estes. Porém atualmente ainda existe um número restrito de ferramentas dessa natureza, sendo que muitas delas não possibilitam que o usuário final desenvolva uma aplicação de maneira simples e amigável. Nesse caso, o usuário deverá dedicar um tempo relativamente longo no aprendizado da ferramenta, ao invés de concentrar esforços no problema propriamente dito.

No entanto, é possível criar um ambiente para o desenvolvimento de agentes BDI com um conjunto de ferramentas já existentes, que possuam compatibilidade entre si. Estas ferramentas são: um editor de ontologias, o qual será responsável pela modelagem do domínio da aplicação (representação dos estados mentais crença, desejo e intenção), uma máquina de inferência, responsável pela manipulação dos estados mentais, e um *framework* para o desenvolvimento de agentes, o qual proverá as funcionalidades básicas do mesmo.

1.2 Objetivos

Este trabalho tem como objetivo geral a modelagem e a implementação de um ambiente para desenvolvimento de agentes cognitivos compatíveis com a arquitetura BDI.

Para que este torne-se possível, foram definidos alguns objetivos específicos:

- Descrever aspectos teóricos das arquiteturas de agentes, principalmente das arquiteturas cognitivas BDI;
- Especificar a arquitetura BDI adotada pelo ambiente a ser implementado;
- Estudar as ferramentas que serão utilizadas na construção do ambiente;
- Implementar o ambiente proposto;
- Desenvolver uma aplicação com o intuito de validar o ambiente proposto e servir como exemplo para aplicações futuras;

1.3 Organização do Trabalho

A presente monografia está organizada da seguinte forma:

O capítulo 2 apresenta conceitos, características, e arquiteturas internas de agentes inteligentes. Também são apresentadas neste capítulo características dos ambientes nos quais os agentes estão inseridos e fazem parte.

O capítulo 3 aborda especificamente a arquitetura BDI, apresentando conceitos relacionados a mesma, e explicando a teoria do raciocínio prático, que a fundamenta. Também são descritos aspectos gerais inerentes a implementação de agentes segundo esta arquitetura.

O capítulo 4 apresenta o modelo de arquitetura BDI proposto pelo presente trabalho, descrevendo seus processos, bem como a estrutura dos estados mentais.

O capítulo 5 apresenta o ambiente para desenvolvimento de agentes, que implementa a arquitetura proposta no capítulo anterior.

O capítulo 6 apresenta a aplicação BALBINO, explicando de modo detalhado como a mesma foi desenvolvida com o ambiente proposto.

O capítulo 7 apresenta os resultados obtidos, bem como trabalhos futuros e os problemas enfrentados durante a elaboração deste trabalho.

2 AGENTES INTELIGENTES

Apesar de atualmente ser um termo bastante utilizado, ainda não existe um conceito unânime de agente. Pesquisadores da comunidade de IA têm dificuldade em chegar a um consenso, pois determinadas características de um agente têm a sua relevância atrelada ao contexto de sua aplicação.

Russel e Norvig [RUS95] propuseram uma definição de um agente inteligente ideal. Para cada possível seqüência de percepções, um agente inteligente ideal deve agir de modo a obter maior sucesso, com base na evidência fornecida pela seqüência de percepções e qualquer conhecimento embutido que o mesmo possua.

Franklin e Graesser [FRA96] revisam conceitos elaborados por outros pesquisadores, e logo após propõem uma definição formal bastante completa para o termo agente. Para estes autores, um agente autônomo é um sistema que faz parte do ambiente no qual está situado, e que percebe e age sobre este ambiente durante um período contínuo de tempo, em busca de sua própria agenda, sendo que seus atos podem mudar suas futuras percepções.

Para Brenner [BRE98], um agente inteligente é definido como sendo um programa de software com capacidade de executar tarefas específicas para um usuário. Deve possuir um grau de inteligência que permita que ele possa agir de maneira autônoma e interagir com o seu ambiente.

Segundo Wooldridge e Jennings [WOO95], podemos discernir dois usos para o termo agente. O primeiro é mais geral, e relativamente incontestável. O segundo é mais específico, e potencialmente controverso. De uma maneira mais geral, o termo agente é usado para denotar um sistema baseado em hardware ou software que exhibe as seguintes características:

- **Autonomia:** Um agente deve ser capaz de operar sem intervenção direta de um usuário, e ter algum tipo de controle sobre seu comportamento e seu estado interno;
- **Habilidade Social:** Um agente deve ser capaz de interagir com outros agentes, humanos ou não, em prol de seus objetivos;

- **Pró-Atividade:** Um agente inteligente não deve simplesmente responder aos estímulos vindos de outros agentes e do seu ambiente, devendo também tomar iniciativa para que seus objetivos sejam satisfeitos;
- **Reatividade:** Um agente deve ser capaz de perceber o seu ambiente, e responder em tempo a estímulos vindos do mesmo.

Para alguns pesquisadores – particularmente os que trabalham com IA – o termo agente tem um significado mais forte e mais específico. Estes pesquisadores geralmente descrevem um agente como um sistema de computação que, além de possuir as características acima citadas, são conceituados ou implementados usando conceitos que normalmente são aplicados a humanos.

Um agente pode exibir outras características, das quais podemos citar:

- **Adaptabilidade:** Um agente aprende através da experiência e conseqüentemente altera seu comportamento, permitindo que ele se adapte a um ambiente dinâmico;
- **Benevolência:** Capacidade que um agente possui de cooperar com outros agentes;
- **Mobilidade:** Habilidade que um agente possui de migrar de uma plataforma para outra através de uma rede de computadores;
- **Racionalidade:** Um agente deve agir sempre em prol de seus objetivos, sempre escolhendo a melhor alternativa disponível no momento;
- **Veracidade:** Um agente não vai propositalmente comunicar informações falsas.

A FIGURA 1 mostra o diagrama da interação de um agente com o seu ambiente, bem como o fluxo de dados interno do agente.

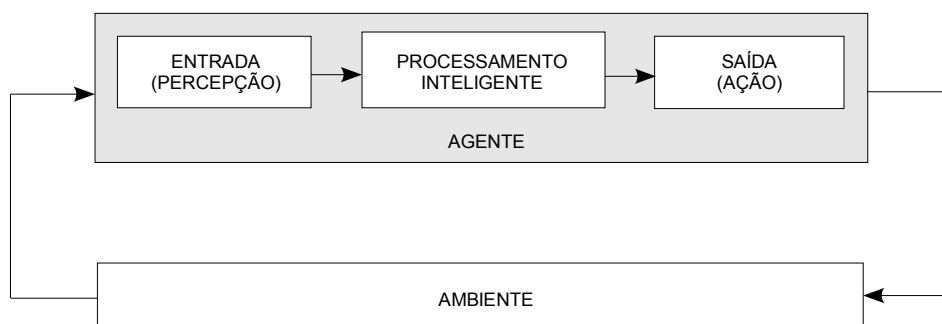


FIGURA 1 – Agente e seu ambiente.

Os agentes podem ser classificados conforme diferentes taxonomias. Uma primeira taxonomia classifica os agentes quanto a suas capacidades: grau de inteligência, mobilidade e grau de autonomia. Uma segunda taxonomia consiste em agrupar os agentes conforme a sua função: agentes de informação, agentes de interface e agentes de transação [BRE98]. No entanto, as perspectivas anteriores não são adequadas ao presente trabalho, sendo necessário classificarmos os agentes quanto a sua estratégia de processamento (arquitetura).

2.1 Arquiteturas de Agentes

A arquitetura de um agente especifica como este pode ser decomposto em um conjunto de módulos componentes, e como estes módulos devem interagir. O conjunto de módulos e suas interações descrevem como que os dados recebidos do ambiente e o estado interno do agente determinam suas ações [MAE95 apud WOO95].

Um agente realizará o mapeamento de uma percepção em uma ação conforme a sua arquitetura. Este mapeamento consiste em um processamento inteligente, que o diferencia de um simples sistema clássico de entrada e saída.

Quanto a arquitetura, os agentes costumam a ser classificados em dois grupos. O primeiro grupo corresponde aos agentes reativos (tropistas), e o segundo aos agentes cognitivos (deliberativos). No entanto, podemos ter uma arquitetura híbrida, que apresenta comportamentos inerentes tanto a arquitetura reativa como a cognitiva.

2.1.1 Arquiteturas Reativas

Os agentes reativos são estruturalmente simples, pois não possuem uma representação do seu ambiente e não possuem capacidade de realizar raciocínios lógicos complexos. Tomam suas decisões com referência no presente pois não possuem histórico. O comportamento inteligente está ligado ao ambiente, sendo produto da sua interação com o mesmo.

Segundo Brooks [BRO86 apud BRE98], um agente reativo possui módulos de competência, que permitem que o mesmo responda a eventos particulares do seu ambiente. Conforme podemos ver na FIGURA 2, os sensores percebem o ambiente e

encaminham as informações para os módulos de competência, os quais realizam o processamento que irá gerar a reação adequada.

São propriedades dos agentes reativos:

- Não possuem uma representação simbólica explícita do ambiente;
- Não têm capacidade de realizar raciocínios complexos;
- Não possuem qualquer tipo de histórico.

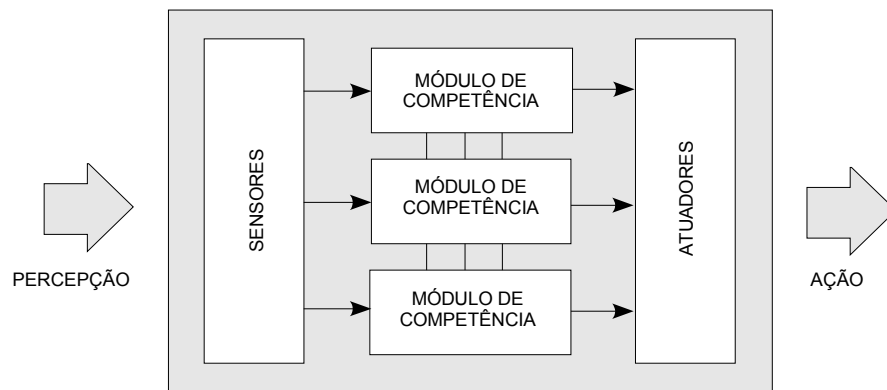


FIGURA 2 – Agente reativo proposto por Brooks [BRO86 apud BRE98].

2.1.2 Arquiteturas Cognitivas

Os agentes cognitivos, também conhecidos como agentes deliberativos, possuem uma representação simbólica explícita do seu ambiente como componente principal da sua base de conhecimentos. Mantém histórico de suas ações, possibilitando que decisões passadas interfiram em decisões futuras. Suas decisões são feitas através de um raciocínio lógico sobre a sua base de conhecimentos.

São propriedades dos agentes cognitivos:

- Representação simbólica explícita do ambiente;
- Capacidade de raciocínio lógico sobre os conhecimentos armazenados em sua base;
- Possibilidade de manter histórico de ações e interações.

Segundo [OLI96 apud ZAM02], as arquiteturas cognitivas podem ser classificadas em:

- **Arquiteturas funcionais:** O agente é composto por módulos que representam cada uma das funcionalidades necessárias para sua operação;

- **Arquiteturas baseadas em estados mentais:** Adotam uma perspectiva de inspiração psicológica para definir a estrutura dos agentes, que são entidades cujo estado é constituído de componentes mentais tais como crenças, desejos, capacidades, escolhas e compromissos.

2.2 Ambientes

Os ambientes provêm informações, as quais são captadas pelas percepções do agente. Segundo Russel e Norvig [RUS95], os ambientes podem ser classificados quanto a suas propriedades. São elas:

- **Acessível ou inacessível:** Um ambiente acessível é aquele no qual o agente pode obter informações completas, precisas e atualizadas sobre o seu estado. Caso contrário, é dito inacessível;
- **Determinístico ou não-determinístico:** Um ambiente determinístico é aquele no qual uma ação tem um único efeito possível. Quando não é possível saber o estado que o ambiente assumirá após a execução de uma ação, dizemos que o ambiente é não-determinístico;
- **Estático ou dinâmico:** Um ambiente é estático para um agente quando permanece inalterado até o momento em que ele executa alguma ação. Em um ambiente dinâmico, além das ações desempenhadas pelos agentes, existem processos que operam sobre o ambiente, alterando o estado do mesmo;
- **Discreto ou contínuo:** Um ambiente discreto é aquele que possui um número finito de estados. Um exemplo de ambiente discreto é o jogo de damas, o qual possui um número limitado de movimentos a cada jogada. Um ambiente contínuo é aquele no qual é possível assumir incontáveis estados. É o caso de um veículo em movimento, o qual possui a sua localização representada dentro de um intervalo de valores contínuos.

3 MODELO BDI

O modelo BDI representa uma arquitetura cognitiva baseada em estados mentais, e tem sua origem no modelo de raciocínio prático humano. O nome atribuído ao modelo é justificado pelos seus estados mentais: crença, desejo e intenção (*Belief, Desire and Intention*). Uma arquitetura baseada no modelo BDI representa seus processos internos através dos estados mentais acima citados, e define um mecanismo de controle que seleciona de maneira racional o curso das ações.

Segundo [WOO00], o modelo BDI é particularmente interessante porque combina três componentes distintos:

- **Componente filosófico:** O modelo BDI é fundamentado na conhecida teoria de ação racional em humanos, proposta pelo filósofo Michael Bratman;
- **Componente de arquitetura de *software*:** O modelo BDI de agência não prescreve uma implementação específica, ou seja, o modelo pode ser implementado de maneiras diferentes. O fato de que o modelo BDI tem sido implementado com sucesso, é um ponto significativo a seu favor;
- **Componente lógico:** O terceiro componente do modelo é um grupo de lógicas. Estas lógicas capturam os aspectos-chaves do modelo BDI como um conjunto de axiomas lógicos.

O conjunto de componentes acima citado torna o modelo BDI um dos mais conhecidos e estudados modelos de raciocínio prático existentes, sendo o componente de arquitetura de *software* objeto de estudo do presente trabalho.

3.1 Raciocínio Prático

O raciocínio prático consiste em ponderar considerações conflitantes a favor e contra alternativas competitivas, onde as considerações relevantes são determinadas pelo que o agente crê, importa-se, valoriza e acredita [BRA90 apud WOO00].

Não devemos confundir-lo com o raciocínio teórico, o qual é dirigido a crenças. Como exemplo de raciocínio teórico podemos citar as conhecidas premissas e sua respectiva conclusão: acredito que todos homens são mortais, e acredito que Sócrates é um homem, então concluo que Sócrates é mortal. O processo de concluir que Sócrates é mortal é um raciocínio teórico, pois afetou apenas as minhas crenças acerca do mundo. Como exemplo de raciocínio prático podemos citar o processo de decidir correr ou caminhar, uma vez que tal processo está relacionado a ações.

O raciocínio prático humano consiste em pelo menos duas atividades distintas:

- **Deliberação:** Processo que envolve a decisão de qual estado o agente quer alcançar;
- **Raciocínio meio-e-fim:** Processo que resulta em algum tipo de plano, o qual define como o agente alcançará o estado selecionado no processo anterior.

Para um melhor entendimento das atividades acima citadas, considere o exemplo a seguir. Após um intenso semestre de aulas, um estudante tem muitas alternativas de lazer durante o seu período de férias. Dentre as alternativas, podemos citar viagens, leituras, festas, entre outras. Chamamos deliberação o processo de escolher uma das alternativas. Digamos que o estudante resolveu viajar. Chamamos de raciocínio meio-e-fim o processo que resultará em um plano, possibilitando que o estudante chegue ao local destino. Neste caso um plano consiste na compra das passagens, embarque no meio de transporte, e assim por diante.

3.2 Estados Mentais

Segundo a psicologia popular, o comportamento humano pode ser previsto e explicado através da atribuição de estados mentais, como por exemplo expectativa, desejo, crença, entre outros. Entidades cujo comportamento pode ser previsto pelo método de atribuição de crenças, desejos e perspicácia de raciocínio, são chamadas sistemas intencionais [DEN87 apud WOO00].

A idéia principal dessa abordagem se concentra no fato de que o agente cognitivo possui estados internos que se relacionam com estado do ambiente com o qual interage. Estes estados seriam correspondentes aos estados mentais humanos, que apresentam um vínculo com mundo em termos de sua existência e significância [COR94].

Segundo McCarthy [MAC79 apud WOO00], atribuir crenças, livre arbítrio, intenções, consciência, habilidade, ou querer a uma máquina é legítimo quando uma atribuição expressa a mesma informação acerca da máquina daquele que é expressa sobre uma pessoa. É útil quando a relação nos ajuda a entender a estrutura da máquina, seu comportamento passado ou futuro, e como consertá-la e aprimorá-la.

Existem outros motivos, além do elevado nível de abstração, para acreditarmos que a modelagem de agentes como sistemas intencionais é útil para o entendimento de programas de computadores [HUH98 apud WOO00]. Primeiro, e talvez a mais importante, a habilidade de comunicação entre agentes heterogêneos e auto-motivados, a qual implicaria na habilidade de falar sobre seus estados mentais. Segundo, modelos mentalísticos são excelentes candidatos para representar informações sobre usuários finais.

Os estados mentais podem ser classificados em duas categorias: estados mentais de informação e estados mentais pró-ativos. A primeira categoria está relacionada à informação que o agente possui acerca do mundo que ele ocupa, como por exemplo crença e conhecimento. Os estados mentais pró-ativos são aqueles que de alguma maneira guiam as ações do agente, como por exemplo os desejos e as intenções.

A seguir, são apresentados os três estados mentais adotados pelo modelo BDI.

3.2.1 Crença

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

As crenças representam o conhecimento sobre o mundo. Em termos computacionais, crenças são apenas uma maneira de representar o estado do mundo, seja através de variáveis, uma base de dados relacional, ou expressões simbólicas em um cálculo de predicados. As crenças são essenciais porque o mundo é dinâmico (os eventos passados precisam ser lembrados), e os sistemas têm apenas uma visão local do mundo (eventos fora da sua esfera de percepção devem ser lembrados) [GEO99].

Wooldridge [WOO00] define as crenças como informações que um agente tem acerca do mundo no qual ele se encontra. Essas crenças podem ser incompletas ou incorretas.

3.2.3 Desejo

Segundo [GIR99], os desejos são relacionados eventualmente ao estado de mundos que o agente quer provocar. Os desejos não dirigem necessariamente o agente a agir, isto é, o fato de um agente possuir um desejo não significa agir para o satisfazer. Significa que antes de um determinado agente decidir o que fazer, ele passa por um processo de racionalização e confronta os seus desejos com as suas convicções. O agente escolherá os desejos que são possíveis de acordo com algum critério.

Os desejos (com frequência imprecisamente chamados de objetivos) formam um essencial componente sobre o estado do sistema. Os desejos representam um estado final que o agente quer verificar. Os *softwares* convencionais são orientados a tarefa ao invés de orientados a objetivos [GEO99].

Para [ZAM02], o desejo é um estado mental intencional 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 ser, simultaneamente, conflitante com outro desejo;
- Não causa diretamente ações, mas pode, potencialmente, gerar suas ocorrências, deixando por conta das intenções a realização de tais ações.

3.2.3 Intenção

As intenções correspondem aos estados de mundo que o agente quer efetivamente provocar, ou seja, existe um comprometimento em realizá-las. Podem ser consideradas um subconjunto dos desejos, mas ao contrário destes, devem ser consistentes. As intenções são formadas a partir de um processo de deliberação e a partir do refinamento de outras intenções. No entanto, um agente pode conter intenções iniciais inseridas pelo usuário.

Normalmente, o termo intenção é empregado tanto para caracterizar um estado mental quanto para caracterizar uma ação. O estado mental intenção está direcionado para o futuro e não desencadeará obrigatoriamente uma ação. A ação intencional está direcionada para o presente e representa o ato de agir imediatamente [BRA87 apud WOO00].

Wooldridge [WOO00] aponta como propriedades das intenções:

- **Intenções direcionam o raciocínio meio-e-fim:** Uma vez formada a intenção, um agente deve tentar realizá-la através de um plano. Caso um plano particular venha a falhar, um agente deve tentar outro;
- **Intenções persistem:** Uma intenção deve persistir até que seja realizada, e deve ser abandonada somente se for constatado que não é mais possível realizá-la, ou a razão que a formou deixou de existir;
- **Intenções restringem deliberações futuras:** O agente não selecionará novas intenções que são inconsistentes com as atuais;
- **Intenções influenciam as crenças sobre as quais os raciocínios práticos futuros são baseados:** Um agente realiza planos futuros partindo do pressuposto que as suas intenções serão realizadas.

3.3 Agentes BDI

Discutiremos a seguir alguns aspectos gerais inerentes ao desenvolvimento de agentes compatíveis com o modelo BDI, tais como processos internos, estratégias de comprometimento e reconsideração de intenções.

3.3.1 Processo Deliberativo

Consiste em formar as novas intenções do agente com base nas crenças, desejos e intenções atuais do mesmo. Normalmente, o processo é formado por duas etapas:

- **Geração de opções:** Consiste na escolha de um conjunto opções (desejos) levando em conta as crenças e as intenções atuais do agente;
- **Filtragem:** Tem como objetivo escolher a melhor alternativa gerada pela etapa anterior, a qual formará a nova intenção.

3.3.2 Raciocínio Meio-e-Fim

É um processo que está associado a idéia de planos, os quais têm como objetivo realizar as intenções do agente. A IA elaborou diversas soluções para o problema de planejamento, como métodos de busca em espaços de possíveis seqüências de ações [BRA88]. Porém, uma das abordagens mais comuns para o problema do planejamento consiste em uma biblioteca de planos prontos, que contém um conjunto de ações aplicável a uma determinada situação.

3.3.3 Estratégias de Comprometimento

Quando uma intenção é selecionada através de um processo deliberativo, o agente deve se comprometer com a realização da mesma. A estratégia de comprometimento implica em persistência temporal e especifica quando um agente deve abandonar uma intenção. Podemos classificar as estratégias de comprometimento em:

- **Cega** (*blind*): Um agente manterá uma intenção até que ela seja realizada;
- **Obsecada** (*single-minded*): Um agente manterá uma intenção até que ela seja realizada ou que ele chegue a conclusão que não é mais possível realizá-la;
- **Liberal** (*open-minded*): Uma intenção é mantida enquanto o agente crer que é possível realizá-la.

3.3.4 Reconsideração de Intenções

Normalmente, um agente reconsidera suas intenções após as seguintes condições serem satisfeitas: conclusão de um plano, realização de suas intenções atuais ou impossibilidade de realização das mesmas. Este agente é chamado de ousado. Caso um agente reconsidere suas intenções durante a execução de um plano, possivelmente após a execução de cada ação, ele é chamado de cauteloso [WOO00].

A reconsideração de intenções está relacionada a mudanças no ambiente. Um agente cauteloso em um ambiente pouco dinâmico gasta muito tempo reconsiderando os seus comprometimentos, enquanto um agente ousado está

ocupado realizando suas intenções. Um agente cauteloso ocupando um ambiente bastante dinâmico consegue detectar novas oportunidades, enquanto um agente ousado não possui essa capacidade.

3.4 Arquiteturas

A seguir serão apresentadas em detalhe algumas das mais conhecidas e importantes arquiteturas que incorporam os aspectos inerentes ao modelo BDI.

3.4.1 IRMA [BRA88]

A arquitetura abstrata IRMA – *Intelligent Resource-bounded Machine Architecture* – foi proposta por Bratman, Israel e Pollack, e tem como objetivo maior a descrição dos processos de um raciocínio prático em agentes com recursos limitados. Podemos visualizar a estrutura da arquitetura IRMA através da FIGURA 3.

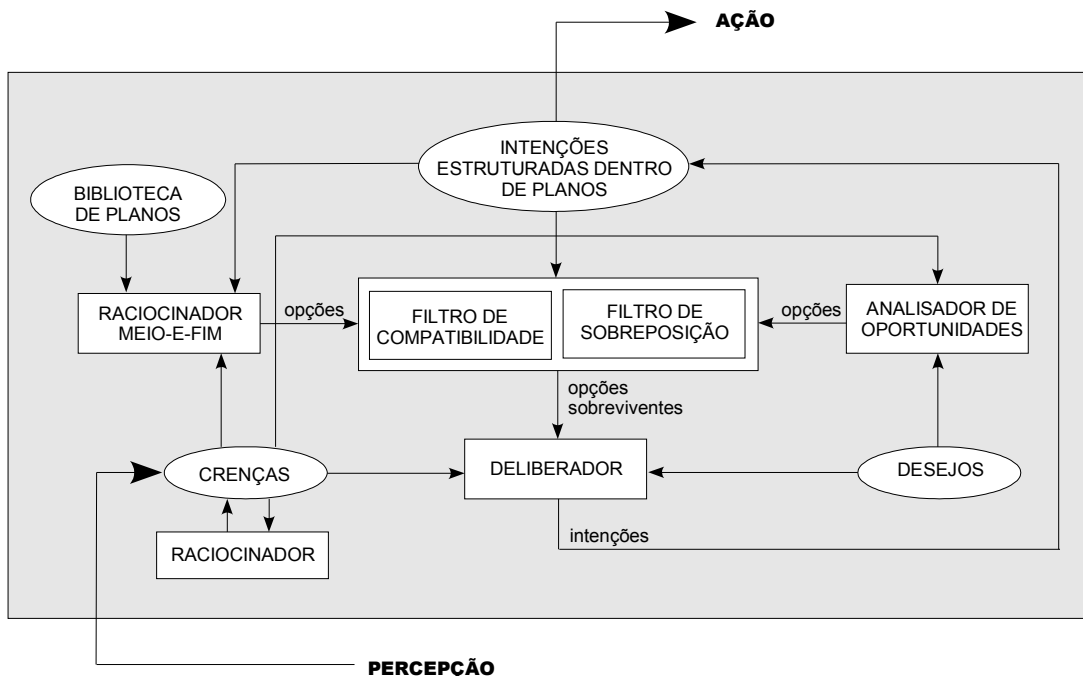


FIGURA 3 – Arquitetura IRMA.

Pode ser classificada como uma arquitetura BDI, pois inclui representação das crenças, desejos e intenções do agente. As intenções do agente são estruturadas dentro de planos, os quais podem ser agrupados em duas categorias:

- **Biblioteca de planos:** Corresponde a um subconjunto das crenças do agente sobre ações e seus efeitos sob condições específicas;
- **Intenções estruturadas dentro de planos:** Corresponde aos planos atuais do agente, os quais contêm as intenções.

Os planos da arquitetura IRMA podem ser parciais, ou seja, é possível decidir sobre um estado final, no entanto a deliberação relativa aos meios para atingir tal estado pode ser postergada. Um agente pode, por exemplo, decidir comprar um livro, mas postergar a decisão relativa a forma de pagamento.

O componente raciocinador meio-e-fim, como o próprio nome sugere, é responsável pela formação de planos. Uma vez que o sistema já possua planos parciais estruturados, os planos formados são utilizados para completá-los. O componente analisador de oportunidades tem como objetivo propor opções em resposta a mudanças percebidas do ambiente. O processo de filtragem envolve dois componentes. O componente filtro de compatibilidade verifica se as opções são compatíveis com os planos atuais. Uma opção considerada incompatível é encaminhada para o filtro de sobreposição. Este componente é sensível a problemas e oportunidades do ambiente, os quais evidenciam condições sob as quais um plano deve ser suspenso e comparado com outras opções. Caso passe por este segundo filtro, a opção será comparada com a intenção que ocasionou a incompatibilidade. Finalmente, o processo de deliberação produz as intenções que serão introduzidas nos planos do agente.

3.4.2 PRS [GEO89]

A arquitetura PRS – *Procedural Reasoning System* – foi originalmente proposta por Georgeff e Lansky. Trata-se de uma arquitetura híbrida que incorpora o modelo BDI. Na FIGURA 4 podemos visualizar a estrutura da arquitetura PRS, a qual é formada pelos seguintes componentes: base de dados, objetivos, biblioteca de áreas de conhecimento, estrutura de intenção e interpretador.

A base de dados armazena as crenças do sistema, bem como informações sobre o domínio da aplicação. O conhecimento contido na base de dados é expresso

em lógica de primeira-ordem.

O objetivos correspondem aos desejos atuais do sistema. Podemos ter um objetivo intrínseco, o qual não está relacionado a intenções existentes, ou um objetivo operacional, que é um sub-objetivo de alguma intenção existente.

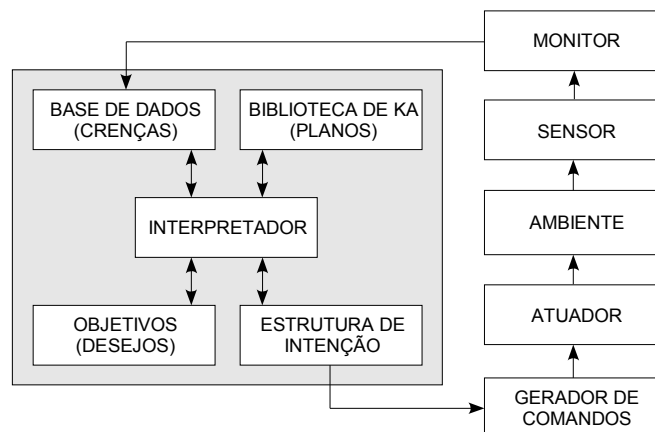


FIGURA 4 – Arquitetura PRS.

A biblioteca de KA contém o que chamamos de áreas de conhecimento (*Knowledge Areas – KA*), as quais são formadas por um corpo, o qual contém os passos do procedimento, e de uma condição de invocação, a qual especifica sob quais situações uma KA é aplicável. O corpo de uma KA pode ser visto como um plano ou como um esquema de um plano. O corpo é representado por um grafo com um único nodo inicial e possíveis múltiplos nodos finais. Os arcos do grafo correspondem a sub-objetivos. Uma KA sem corpo é dita primitiva pois está associada a uma ação primitiva. Uma meta-KA é aquela que contém informações para manipulação dos próprios estados mentais do sistema, ou seja, são literalmente KAs sobre KAs.

A estrutura de intenção contém todas as tarefas que o sistema selecionou para executar, tanto imediatamente como posteriormente. Chamamos essas tarefas adotadas de intenções. Uma única intenção consiste de uma KA inicial juntamente com todas outras [sub-]KAs que estão envolvidas na tentativa de executar com sucesso aquela KA. Uma intenção pode ser temporariamente suspensa para aguardar uma condição particular. Logo após uma intenção ter sido executada ou abandonada, ela desaparece da estrutura de intenção.

O interpretador manipula os outros componentes, selecionando os planos com base nas crenças e objetivos do sistema, e colocando os planos selecionados na

estrutura de intenção para serem executados. O interpretador compara os objetivos e as crenças com as condições de invocação dos planos, o que permite que o sistema possa reagir em tempo. Contudo, ele também é capaz de realizar inferências através da invocação de meta-KAs. Uma vez selecionada, uma KA é inserida na estrutura de intenção. Caso esta KA tenha sido selecionada devido a aquisição de um novo objetivo intrínseco, ela é inserida como uma nova intenção. Quando uma intenção é selecionada para execução, ela deverá ser preenchida com uma ação primitiva, que será executada, ou com sub-objetivos, os quais serão anunciados como novos objetivos operacionais do sistema. Nesse momento, um novo ciclo do interpretador recomeça.

As meta-KAs desempenham um importante papel no sistema, sendo normalmente invocadas mediante o surgimento de um problema particular como a possibilidade de aplicar mais de uma KA em uma situação, a ocorrência de falha ao atingir um objetivo, computar o tempo associado a execução de um plano, ou mesmo a reativação ou suspensão de intenções.

O componente interpretador possui procedimentos pré-determinados para a tomada de decisão, garantindo a reatividade do sistema quando não há conhecimento suficiente para que a melhor escolha seja feita. Entretanto, se o conhecimento sobre o processo decisório existe, o procedimento pré-determinado é substituído através da invocação de uma meta-KA apropriada.

4 MODELO PROPOSTO

Este capítulo tem como objetivo descrever a arquitetura de agentes proposta pelo presente trabalho. A FIGURA 5 ilustra a arquitetura proposta, a qual pode ser classificada como um modelo BDI, pois representa de maneira explícita os estados mentais crença, desejo e intenção.

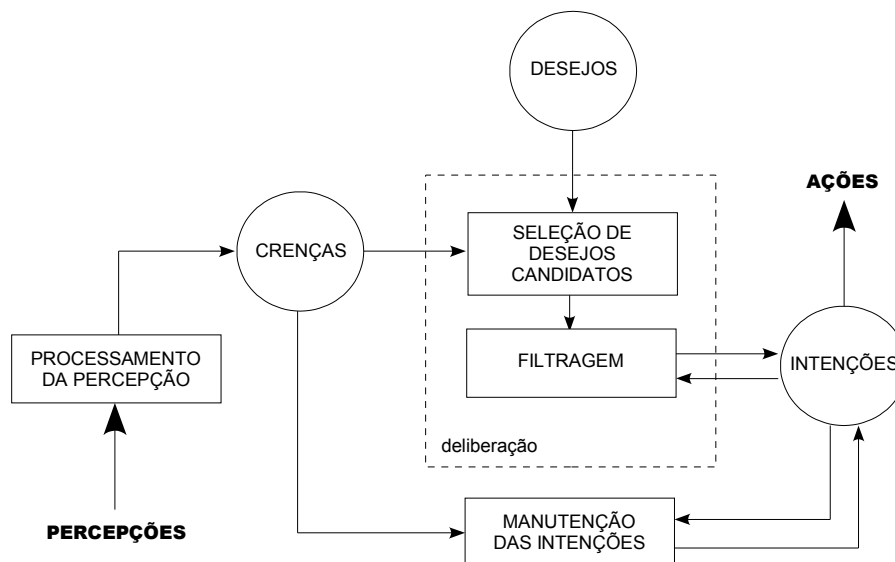


FIGURA 5 – Arquitetura proposta.

4.1 Estados Mentais

Para representar a estrutura interna dos estados mentais foi usada a teoria de quadros (*frames*). A seguir, apresentamos detalhadamente a modelagem de cada um dos estados mentais.

4.1.1 Crenças

As crenças representam os conhecimentos que o agente possui acerca do ambiente que ocupa e faz parte. Uma crença descreve as características de algo, que chamaremos de objeto.

O quadro que representa uma crença é formado pelos atributos (*slots*):

- **Agente:** nome do agente que possui a crença;
- **Código Identificador:** é preciso reconhecer o objeto ao qual a crença se refere, pois em alguns casos é possível que objetos distintos possuam as mesmas características;
- **Características:** descrição do objeto ao qual a crença se refere;
- **Localização Temporal:** corresponde ao instante de aquisição da crença, indicando de maneira implícita se uma crença está desatualizada.

Maria Silva, 35 anos, casada, é um exemplo de características de um objeto, o qual corresponde a uma pessoa. São características bastante comuns, que aumentam a possibilidade de nos depararmos com outro objeto contendo as mesmas características. Este problema é resolvido especificando um código identificador.

Resumindo, um agente acredita que um objeto particular possui um conjunto de características em um determinado momento.

4.1.2 Desejos

Os desejos representam estados do mundo que o agente quer provocar. O modelo de desejo proposto pela presente arquitetura não faz referência a um objeto específico, mas às características que o descrevem. Esta abordagem apresenta como vantagem a possibilidade de caracterizar um objeto tanto quanto necessário, podendo assim abranger um conjunto de objetos referenciando características comuns, ou apenas um objeto referenciando características particulares.

O quadro desejo é formado pelos seguintes atributos:

- **Agente:** nome do agente que possui o desejo;
- **Prioridade:** usada no processo que seleciona os desejos candidatos a gerar intenções;
- **Desejos Incompatíveis:** correspondem aos desejos que não poderão gerar intenções quando o desejo em questão já tiver gerado uma intenção,

ou seja, evita que desejos gerem intenções que não podem ser executadas paralelamente a alguma intenção já existente;

- **Condições:** conjunto de condições que denotam os estados do mundo que fazem com que um desejo seja considerado candidato a gerar intenções, e estados do mundo que indicam que uma intenção gerada pelo desejo foi realizada;
- **Ações:** conjunto de ações a serem executadas por uma intenção gerada pelo desejo, de modo a satisfazê-la.

Um desejo é selecionado segundo a sua prioridade, e tem associado a si um conjunto de condições, as quais expressam as transformações no ambiente que o agente quer provocar.

Cada condição é representada por um quadro, e possui os atributos:

- **Desejo:** uma condição pertence a um desejo;
- **Pré-condição:** denota as características que um objeto deve possuir para que a pré-condição seja satisfeita;
- **Pós-condição:** denota as características que um objeto deve assumir para que uma intenção gerada pelo desejo seja considerada realizada.

Um desejo é selecionado como candidato a gerar uma intenção quando todas pré-condições são satisfeitas. As pós-condições são usadas apenas pelas intenções geradas pelo desejo para verificar sua realização.

O desejo de ligar um computador possui uma condição, cuja pré-condição é a existência de um computador desligado, e a pós-condição é o mesmo computador ligado. Neste caso, desligado ou ligado são características de um objeto computador. Da maneira como foi modelado, este desejo se refere a um computador qualquer, e se quisermos nos referir a um determinado computador devemos adicionar uma característica particular que o torne único.

É possível criarmos uma condição sem pré-condição, sem pós-condição, ou mesmo sem ambas. Quando uma pré-condição ou pós-condição não for especificada, partimos do pressuposto que ela é sempre satisfeita. Um desejo cujas condições não possuem pré-condições é sempre selecionado como desejo candidato.

4.1.3 Intenções

As intenções, assim como os desejos, correspondem a estados do mundo que o agente quer provocar, mas ao contrário dos desejos, as intenções exigem um comprometimento por parte do agente. O modelo de intenção proposto é bastante similar ao modelo de desejo, sendo que a fundamental diferença reside nas condições da intenção.

O quadro que representa uma intenção é composto pelos seguintes atributos:

- **Agente:** nome do agente que possui a intenção;
- **Desejo:** referência ao desejo que a gerou;
- **Localização Temporal:** corresponde ao instante em que a intenção foi gerada;
- **Estado:** uma intenção pode assumir os seguintes estados:
 - Ativa: estado que a intenção assume após criada;
 - Executada: a intenção assume este estado após ter as suas ações executadas;
 - Realizada: uma intenção é dita realizada quando todas as suas pós-condições são satisfeitas;
 - Impossível: uma intenção impossível caso uma das pré-condições deixe de existir antes de o agente realizá-la;
- **Condições:** conjunto de condições que denotam os estados do mundo antes, e depois da intenção ser realizada;
- **Ações:** conjunto de ações a serem executadas de modo a satisfazer a intenção.

Uma intenção tem associada a si um conjunto de condições, as quais expressam as transformações que o agente quer provocar no ambiente através da execução do conjunto de ações. Mas ao contrário das condições adotadas pelos desejos, as condições das intenções estão associadas a objetos.

Cada condição é representada por um quadro que possui os atributos:

- **Intenção:** uma condição pertence a uma intenção;
- **Crença:** corresponde a crença associada à condição;

- **Pré-condição:** no momento em que a intenção é gerada, a pré-condição corresponde às características da crença associada, ou seja, o agente acredita que o objeto referenciado pela crença associada preenche os requisitos para satisfação desta pré-condição;
- **Pós-condição:** no momento em que a intenção é realizada, a pós-condição corresponde às características da crença associada, ou seja, o agente acredita que o objeto referenciado pela crença associada assumiu o estado que satisfaz a pós-condição.

Quando uma intenção é gerada, ela é considerada ativa e possui todas as pré-condições satisfeitas. Uma intenção é dita realizada quando possui todas as pós-condições satisfeitas.

Digamos que o desejo de ligar um computador gerou uma intenção, pois o agente passou a crer que existe um computador desligado, o que corresponde à única pré-condição do desejo. Uma vez criada a intenção, o agente executará um conjunto de ações que ligará o computador. Logo após o agente percebe que o computador está ligado. Isto faz com que a intenção seja considerada realizada, pois a sua única pós-condição foi satisfeita.

Uma vez que as condições dos desejos podem ocorrer sem pré-condição, sem pós-condição, ou sem ambas, as condições das intenções também podem. Uma intenção sem pós-condições é sempre considerada realizada.

4.2 Processos

Os processos apresentados a seguir são responsáveis pela manipulação dos estados mentais.

4.2.1 Processamento da Percepção

O processamento da percepção tem como objetivo registrar tudo aquilo que o agente percebe, mantendo a consistência de suas crenças. Este processo recebe como entrada uma percepção que o agente obteve do seu ambiente, a qual é composta por um código que identifica um objeto, e um conjunto de características do mesmo.

Antes de ser inserida, uma percepção deve ser adequada ao formato adotado pelas crenças. Caso exista uma crença que se refere ao mesmo objeto referido pela percepção, a crença é considerada obsoleta e a percepção é inserida como uma crença atualizada. Manter as crenças obsoletas permite que o agente mantenha um histórico ao longo de sua existência.

O processamento da percepção é detalhado na FIGURA 6.

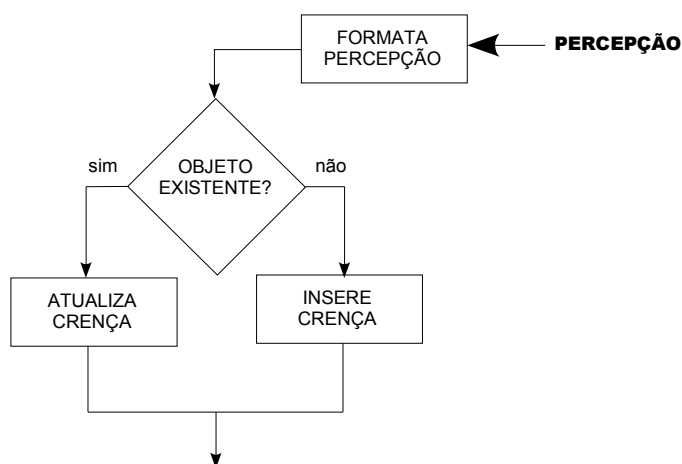


FIGURA 6 – Processamento da percepção.

4.2.2 Manutenção das Intenções

O processo de manutenção das intenções, detalhado na FIGURA 7, elimina intenções impossíveis e realizadas. Uma intenção é considerada realizada quando o agente acredita que todas suas pós-condições estão satisfeitas. Uma intenção é considerada impossível quando um agente deixa de acreditar em alguma de suas pré-condições, ou seja, o motivo que a originou deixou de existir.

Como exemplo imagine que um agente vendedor possui o desejo de vender uma mercadoria para o cliente. As pré-condições são: um cliente procurando uma mercadoria e mercadoria procurada pelo cliente disponível em estoque. As pós-condições são: o respectivo cliente satisfeito e a respectiva mercadoria vendida. Digamos que em um determinado momento o agente passa a crer nas pré-condições, e o desejo de vender uma mercadoria para o cliente acaba gerando uma intenção. Caso uma das pré-condições deixe de existir antes do agente realizar a intenção, ela será considerada impossível. Seria impossível vender uma mercadoria se não houver

alguém para comprá-la. Caso o agente passe a crer nas pós-condições, a intenção é considerada realizada.

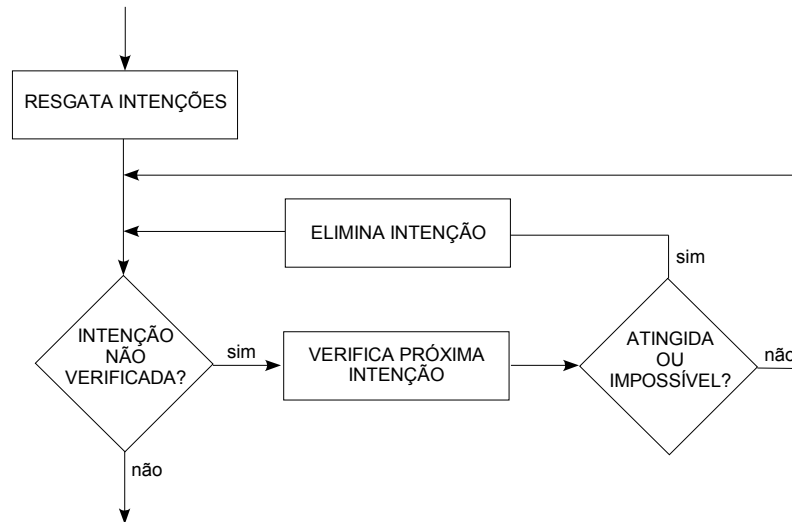


FIGURA 7 – Manutenção das Intenções.

Em determinados casos, certas condições devem ser mantidas para que as intenções sejam realizadas, ou seja, a pré-condição é igual a pós-condição.

4.2.3 Deliberação

O processo deliberativo consiste em raciocinar e decidir um estado do mundo que deve ser atingido. É formado por duas etapas, seleção de desejos candidatos e filtragem, as quais são detalhadas na FIGURA 8.

4.2.3.1 Seleção de Desejos Candidatos

Durante esta etapa, é selecionado um conjunto de desejos candidatos a gerar intenções. Os desejos são selecionados quando o agente passa a crer que todas as suas pré-condições estão satisfeitas, ou seja, quando existe pelo menos uma crença relacionada a cada condição. Esse processo seleciona primeiro os desejos de maior prioridade.

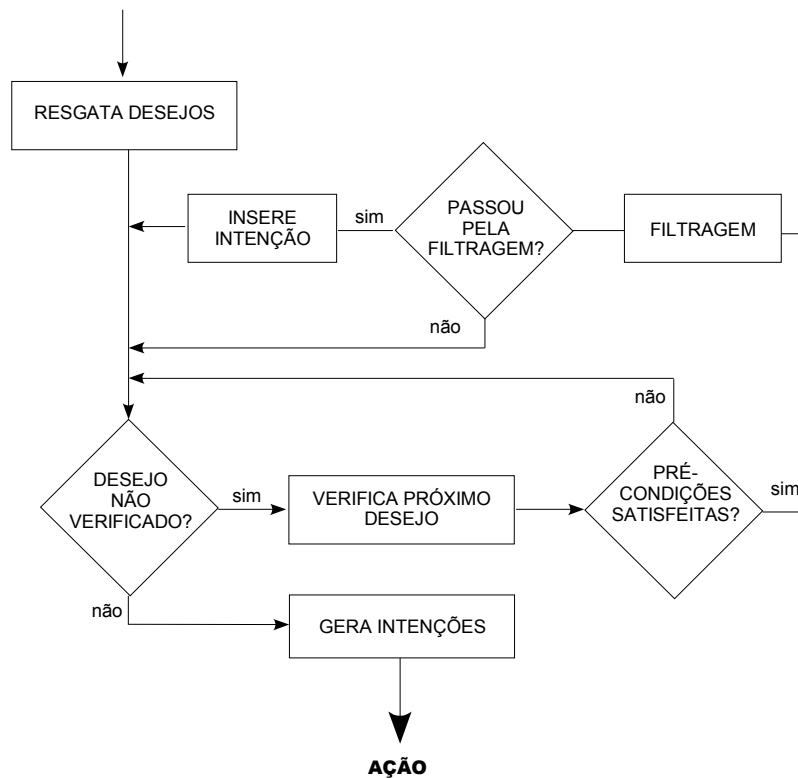


FIGURA 8 – Processo Deliberativo.

4.2.3.2 Filtragem

Os desejos candidatos selecionados na etapa anterior são encaminhadas ao processo de filtragem, que elimina os desejos que são incompatíveis com as intenções do agente.

O processo de filtragem elimina os desejos candidatos cujas ações não podem ser executadas de maneira paralela às ações pertencentes a alguma intenção existente. Tais desejos estão listados no componente de desejos incompatíveis do desejo que gerou a intenção. Como exemplo de incompatibilidade, citamos o fato de não podermos conduzir dois automóveis ao mesmo tempo, ou seja, o desejo de dirigir um automóvel é incompatível com a intenção de dirigir um automóvel.

É possível que tenhamos intenções cujas ações conduzem um determinado objeto a estados distintos, sendo que neste caso sabemos com certeza que apenas uma das intenções será realizada. As restantes serão consideradas impossíveis e removidas pelo processo de manutenção de intenções, uma vez que o objeto assumiu um estado distinto da pré-condição e da pós-condição.

Os desejos candidatos que sobrevivem a este processo geram uma intenção, a qual contém um conjunto de ações que ao serem executadas devem satisfazê-la.

5 IMPLEMENTAÇÃO DO AMBIENTE

5.1 Tecnologias Utilizadas

O modelo proposto foi desenvolvido em linguagem Java [JAV04], uma vez que as ferramentas selecionadas, Protégé e Algernon, são implementadas e possuem uma API (*Application Program Interface*) nesta linguagem. Isto possibilita a criação de instâncias a partir de um programa qualquer codificado em Java.

5.1.1 Protégé

O Protégé [PRO04] consiste em uma ferramenta que permite criar domínios de ontologias, inserir dados, e personalizar formulários para inserção de dados. Também pode ser utilizada como uma biblioteca, onde outras aplicações podem acessar as bases de conhecimento.

Trata-se de uma ferramenta utiliza uma estrutura de frames para modelar os domínio. Possui uma *interface* gráfica bastante simples, permitindo a criação de classes (*frames*), *slots*, instâncias, e formulários para preenchimento dos *slots* das instâncias.

5.1.2 Algernon

O Algernon [ALG04] consiste em um sistema de inferência baseado em regras, que foi implementado em Java, e possui compatibilidade com o editor de ontologias Protégé.

O Algernon executa processamento baseado em regras com encadeamento progressivo e regressivo em bases de conhecimento baseadas em frames, e recupera e armazena informações em ontologias e bases de conhecimento.

5.2 Implementação

A implementação do modelo consistiu em duas etapas. Em um primeiro momento foi criada uma ontologia no Protégé para representar os estados mentais, e logo após foi criado o mecanismo de raciocínio.

5.2.1 Ontologia BDI

Foi criada uma ontologia no Protégé para representar os estados mentais do agente conforme proposto pela arquitetura descrita no capítulo anterior. A ontologia criada consiste nas seguintes classes:

- **MentalState:** é uma classe abstrata, que tem como classes herdeiras as classes Belief, Desire e Intention;
- **Belief:** representa uma crença;
- **Desire:** representa um desejo;
- **Intention:** representa uma intenção;
- **Conditions:** é uma classe abstrata, que representa condições associadas a estados mentais pró-ativos, e tem como classes herdeiras as classes DesireConditions e IntentionConditions;
- **DesireConditions:** representa uma condição associada a um desejo;
- **IntentionConditions:** representa uma condição associada a uma intenção.

A TABELA 1 apresenta os slots e seus respectivos tipo de dados, criados nas classes acima descritas.

TABELA 1 – Slots das classes criadas no Protégé.

MentalState	
agent	String
Belief	
id	String
content	Instance
time	String
Desire	
action	String
priority	Integer

description	String
incompatible	Instance
Intention	
desire	Instance
state	Symbol
actions	String
time	String
Conditions	
precondition	Instance
postcondition	Instance
DesireConditions	
desire	Instance
description	String
IntentionConditions	
intention	Instance
belief	Instance

Na FIGURA 9 é possível visualizar a estrutura de classes criada no Protégé, bem como os atributos da classe Belief.

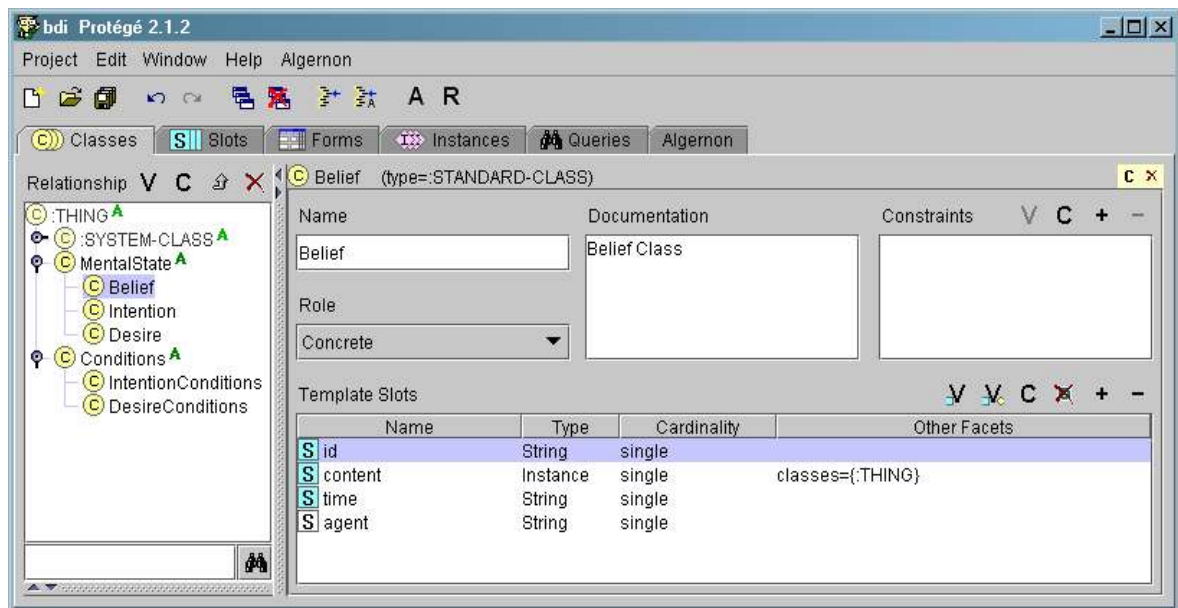


FIGURA 9 – Estrutura de classes.

5.2.2 Raciocinador

O raciocinador consiste na implementação dos processos que manipulam as instâncias das classes que representam os estados mentais. Para isso, foi criada uma classe chamada `BDIReasoner`, a qual encapsula todo mecanismo de raciocínio do agente.

A classe `BDIReasoner` possui um método para o fornecimento de percepções, as quais geram as crenças do agente. Este método é chamado `perception` e recebe um objeto do tipo `Perception` como parâmetro. A classe `Perception` encapsula as funcionalidades relacionadas a adequação de formato da percepção.

Após inserir as novas crenças, é invocado o método `maintainIntentions`, que implementa o processo de manutenção das intenções. Este método resgata todas as intenções do agente, e suas respectivas condições. Depois, verifica se as crenças associadas às condições satisfizeram as pós-condições (intenção realizada) ou se as crenças associadas correspondem a estados diferentes das pré-condições e das pós-condições (intenção impossível). Finalmente, remove as intenções realizadas e as impossíveis.

O próximo método invocado é o `deliberate`, responsável pela execução do processo deliberativo. Em um primeiro momento, são resgatados todos os desejos do agente, e conseqüentemente é verificado se existem crenças que satisfaçam suas pré-condições. Logo após, os desejos selecionados são encaminhados ao processo de filtragem onde são verificadas as incompatibilidades com as intenções. Uma vez aprovado, o desejo candidato gera uma intenção.

O último método a ser invocado é o `actions`, que simplesmente resgata as ações que devem ser executadas pelo agente.

6 APLICAÇÃO DESENVOLVIDA

Foi desenvolvida uma aplicação chamada BALBINO com o intuito de validar a implementação do modelo e servir de exemplo para aplicações futuras. A seguir é descrita a aplicação.

BALBINO é um agente com as seguintes funcionalidades:

- carregar garrafas de bebida que estão na temperatura ambiente e fora do refrigerador para dentro do mesmo, de modo a torná-las gelada;
- vender aos clientes a bebida desejada somente se estiver gelada;
- convidar os clientes que se comportarem de maneira inadequada a se retirarem do recinto.

As bebidas vendidas podem ser das marcas A ou B, podem assumir as temperaturas ambiente ou gelada, e podem estar dentro ou fora do refrigerador. Os clientes podem ter preferência por bebidas das marcas A ou B, e podem estar aguardando o pedido, podem ter sido atendidos, podem ter ido embora, ou mesmo podem estar se comportando de modo inadequado.

6.1 Desejos

Para desenvolver a aplicação não foi necessário criar intenções iniciais, nem crenças iniciais. As intenções são obtidas somente quando o agente passa a crer nas pré-condições de um desejo, e as crenças são obtidas através das percepções. Neste caso, foi necessário apenas criar os desejos do agente, que são descritos a seguir.

DES1 – Colocar garrafas de bebida da marca A no refrigerador:

- *Agente: BALBINO*
- *Pré-condições: Bebida(A ambiente fora)*
- *Pós-condições: Bebida(A ambiente dentro)*
- *Desejos incompatíveis: 1 2 3 4*
- *Prioridade: 1*
- *Ações: Pegar a garrafa
 Abrir refrigerador
 Colocar garrafa dentro do refrigerador
 Fechar refrigerador*

DES2 – Colocar garrafas de bebida da marca B no refrigerador:

- Agente: *BALBINO*
- Pré-condições: *Bebida(B ambiente fora)*
- Pós-condições: *Bebida(B ambiente dentro)*
- Desejos incompatíveis: *1 2 3 4*
- Prioridade: *1*
- Ações: *Pegar a garrafa*
Abrir refrigerador
Colocar garrafa dentro do refrigerador
Fechar refrigerador

DES3 – Atender clientes que solicitam bebidas da marca A:

- Agente: *BALBINO*
- Pré-condições: *Bebida(A gelada dentro)*
Cliente(A aguardando)
- Pós-condições: *Bebida(A gelada fora)*
Cliente(A atendido)
- Desejos incompatíveis: *1 2 3 4*
- Prioridade: *2*
- Ações: *Abrir refrigerador*
Pegar garrafa
Fechar refrigerador
Entregar garrafa ao cliente

DES4 – Atender clientes que solicitam bebidas da marca B:

- Agente: *BALBINO*
- Pré-condições: *Bebida(B gelada dentro)*
Cliente(B aguardando)
- Pós-condições: *Bebida(B gelada fora)*
Cliente(B atendido)
- Desejos incompatíveis: *1 2 3 4*
- Prioridade: *2*
- Ações: *Abrir refrigerador*
Pegar garrafa
Fechar refrigerador
Entregar garrafa ao cliente

DES5 – Solicitar a saída de clientes com preferência A e comportamento inadequado:

- Agente: *BALBINO*
- Pré-condições: *Cliente (A inadequado)*
- Pós-condições: *Cliente (A ausente)*
- Desejos incompatíveis: *5 6*
- Prioridade: *3*
- Ações: *Solicitar que o cliente se retire*

DES6 – Solicitar a saída de clientes com preferência B e comportamento inadequado:

- Agente: *BALBINO*
- Pré-condições: *Cliente (B inadequado)*
- Pós-condições: *Cliente (B ausente)*
- Desejos incompatíveis: *5 6*
- Prioridade: *3*
- Ações: *Solicitar que o cliente se retire*

6.2 Projeto

O primeiro passo consiste em criar um novo projeto no Protégé e importar o arquivo contendo as classes que representam os estados mentais. Em seguida, devemos modelar as classes que representam as características dos possíveis tipos de objetos, que neste caso são: Bebida e Cliente.

A classe Bebida é caracterizada pela sua marca, temperatura e local onde ela se encontra. Conforme mostrado na FIGURA 10, é criado na classe Bebida um *slot* do tipo *Symbol* para cada característica. O *slot* marca pode assumir os valores A ou B. O *slot* temperatura pode assumir os valores ambiente ou gelada. O *slot* refrigerador pode assumir os valores dentro ou fora.

A classe Cliente é caracterizada pela sua preferência de marca e pelo seu estado. É criado um *slot* do tipo *Symbol* para cada característica. O *slot* preferência pode assumir os valores A ou B. O *slot* estado pode assumir os valores aguardando, atendido, inadequado, ou ausente.

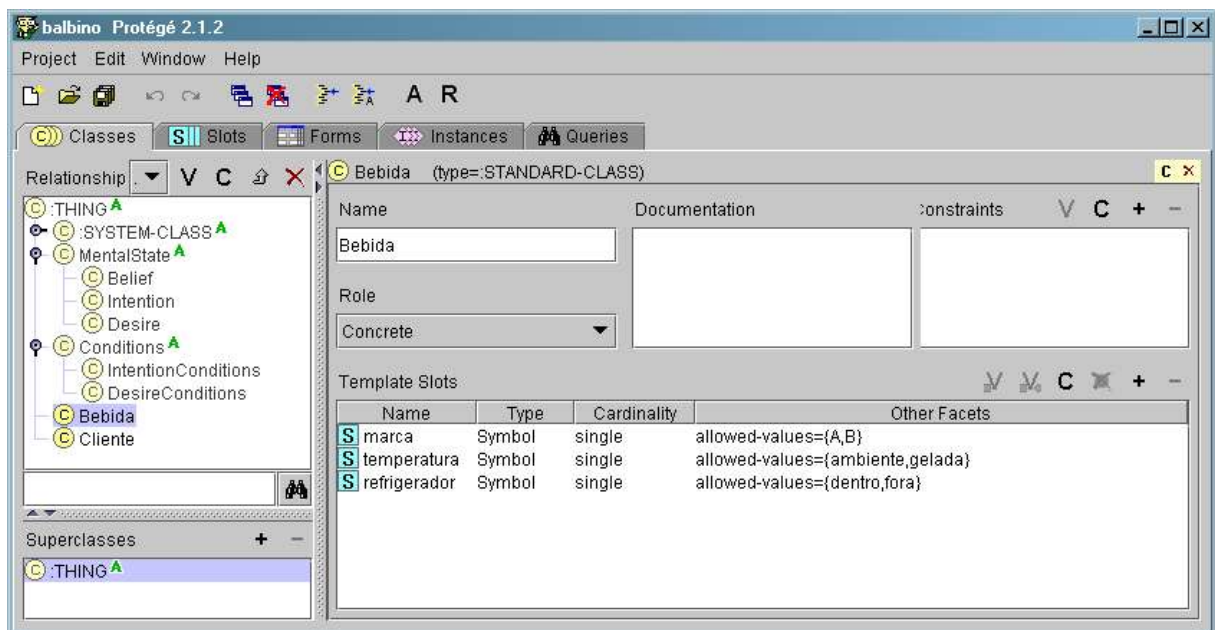


FIGURA 10 – Classe Bebida.

Uma vez criado o conjunto de classes, devemos criar instâncias das mesmas contendo todas combinações possíveis de características, formando assim o domínio das características que um objeto pode vir a assumir. A FIGURA 11 mostra todas possíveis instâncias para a classe Cliente.

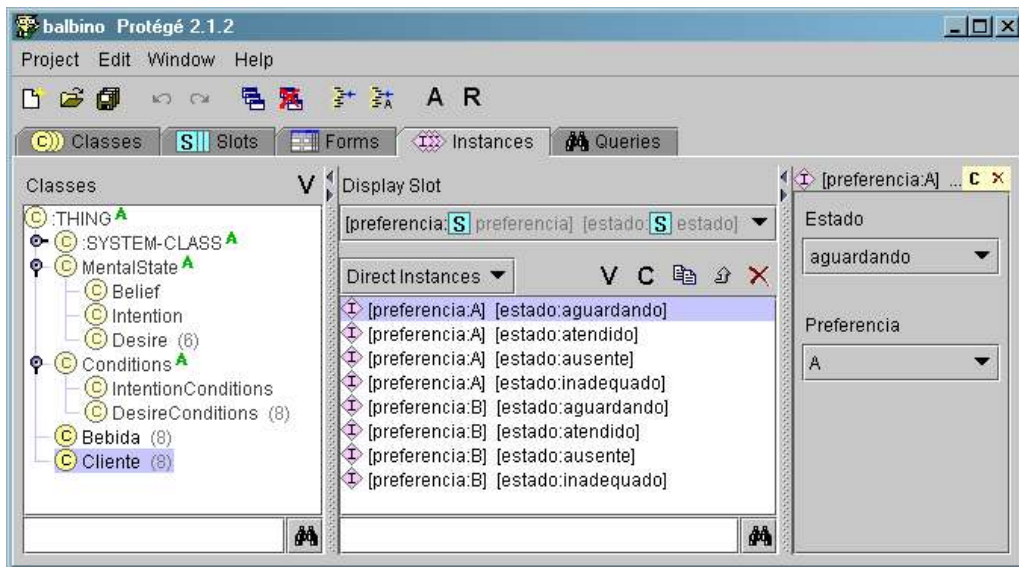


FIGURA 11 – Instâncias da classe Cliente.

O próximo passo consiste em criar os desejos do agente anteriormente descritos. Inicialmente, devemos criar as instâncias da classe Desire, preenchendo os slots conforme ilustrado na FIGURA 12.

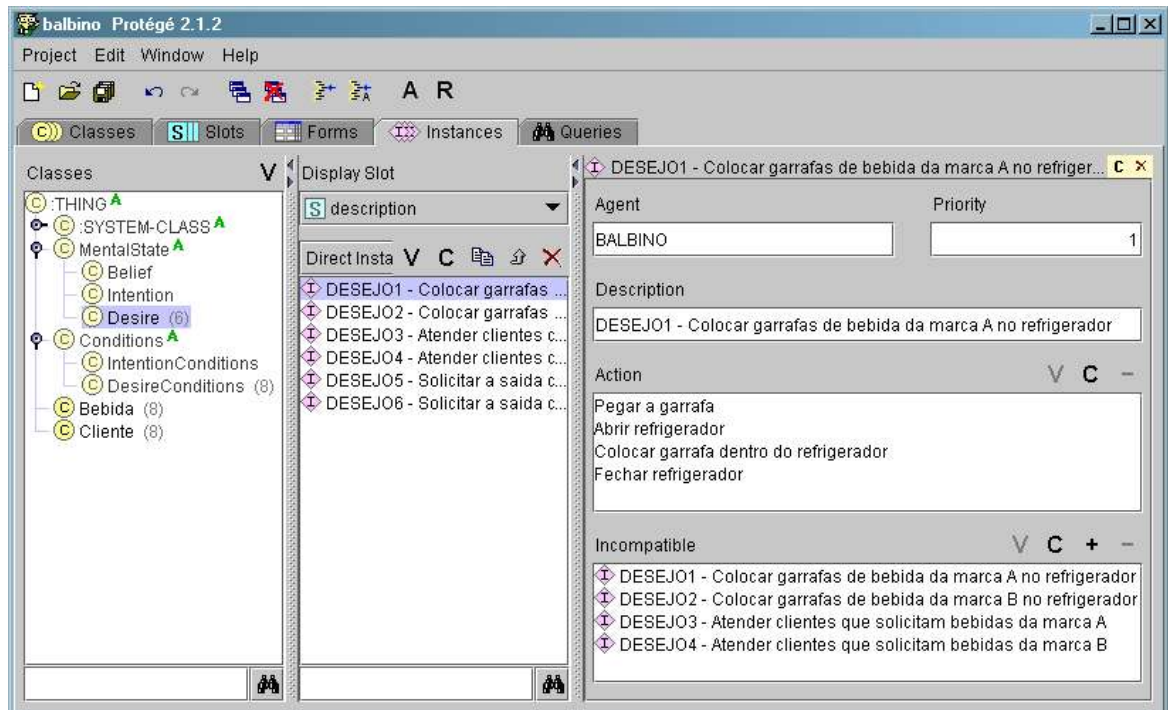


FIGURA 12 – Instâncias da classe Desire.

Com exceção do *slot* priority que é do tipo *Integer* e o *slot* incompatible que é do tipo *Instance*, os restantes são do tipo *String*. O *slot* agent recebe o nome do agente. O *slot* priority recebe a prioridade do desejo. O *slot* incompatible recebe as instâncias dos desejos que são incompatíveis com o desejo em questão. O *slot* description corresponde a uma descrição do desejo, que auxilia no seu entendimento. O *slot* actions é do tipo múltiplo, e corresponde às ações que devem ser executadas pelas intenções geradas pelo desejo.

Finalmente, devemos criar as condições relacionadas a cada desejo. Cada condição corresponde a uma instância da classe *DesireConditions*, a qual possui os *slots* desire, description, precondition e postcondition, conforme mostrado na FIGURA 13. O *slot* desire é do tipo *Instance*, e recebe a instância do desejo ao qual a condição pertence. O *slot* description corresponde a uma descrição da condição. O *slot* precondition é do tipo *Instance*, e recebe a instância de uma classe do domínio de características, neste caso *Bebida* ou *Cliente*, que representa as características que um objeto deve assumir para satisfazer a pré-condição. O *slot* postcondition é do tipo *Instance*, e recebe a instância de uma classe do domínio de características que representa as características de um objeto que satisfazem a pós-condição.

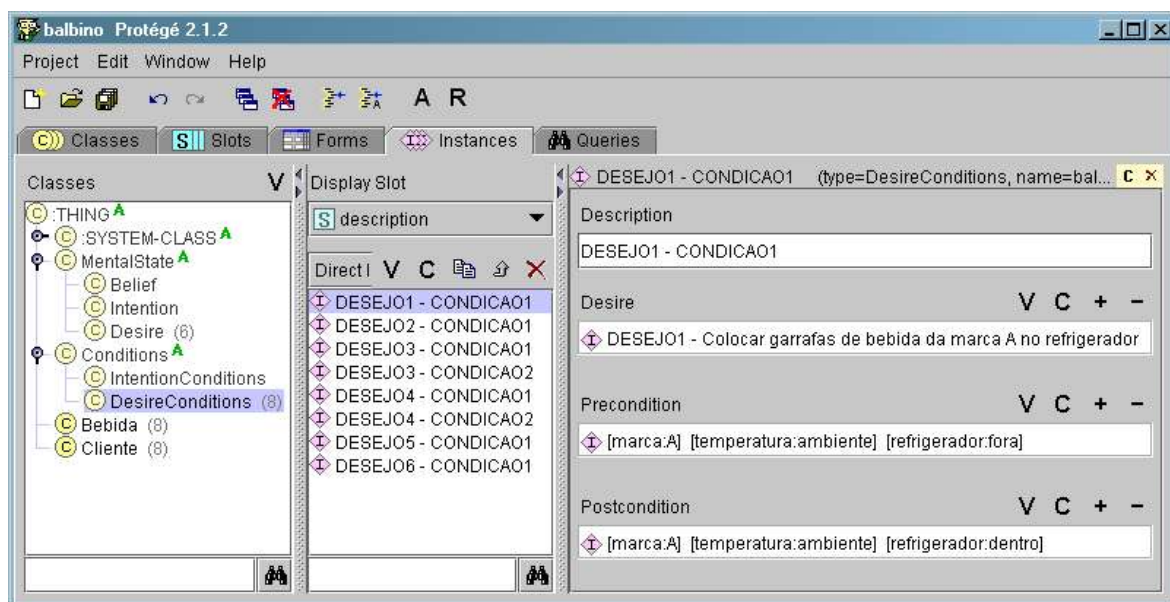


FIGURA 13 – Instâncias das condições dos desejos.

Em alguns projetos, pode ser necessário criar intenções iniciais, ou mesmo crenças iniciais. O procedimento de criação de instâncias destes estados mentais é análogo ao procedimento de criação dos desejos.

6.3 Agente

Após criar o projeto que contém o domínio da aplicação e os estados mentais iniciais, devemos construir o agente propriamente dito, e para isto devemos realizar os seguintes passos:

1. Instanciação de um objeto da classe `BDIReasoner`, que recebe como parâmetros o caminho para o arquivo do Protégé contendo o domínio da aplicação, e o nome do agente;
2. Modificação da classe `Perception`, permitindo a adequação do formato da percepção;
3. Criação de um objeto da classe `Perception`, passando como argumento a percepção obtida;
4. Invocação do método `perception` da instância de `BDIReasoner`, passando como argumento a instância de `Perception` contendo a percepção;
5. Atribuição do retorno do método `actions` da instância de `BDIReasoner` a um objeto da classe `Vector`, o qual armazenará as ações a serem executadas.

A FIGURA 14 ilustra um exemplo resumido do código do agente BALBINO.

```
public class Balbino {  
  
    private static String PROJECT = "/home/balbino/balbino.pprj";  
    private static String AGENT = "BALBINO";  
    private BDIReasoner reasoner = new BDIReasoner(PROJECT, AGENT);  
  
    public void someMethod() {  
  
        while( agent_alive ) {  
  
            /* Obtém percepção */  
            String input = getInput();  
            /* Formata percepção */  
            Perception p = new Perception( input );  
            /* Encaminha a percepção para o raciocinador */  
            reasoner.perception( p );  
            /* Recebe as ações do raciocinador*/  
            Vector actions = reasoner.actions();  
  
            /* Executa ações */  
  
        }  
    }  
}
```

FIGURA 14 – Código resumido do agente BALBINO.

6.4 Funcionamento da Aplicação

Em sua primeira percepção, BALBINO passa a crer que existe um objeto ID1, que corresponde a uma garrafa de bebida da marca A à temperatura ambiente e fora do refrigerador. O desejo DES1, que consiste em colocar bebidas da marca A no refrigerador, é selecionado como desejo candidato, pois a sua pré-condição foi satisfeita pela crença sobre ID1. Após sobreviver ao processo de filtragem, o desejo candidato DES1 gera a intenção INT1, que tem suas ações executadas. Logo após, BALBINO percebe que a garrafa representada pelo objeto ID1 está dentro do refrigerador, então atualiza sua crença sobre ID1, e considerara a intenção INT1 realizada, pois a pós-condição desta foi satisfeita.

Em sua segunda percepção, BALBINO percebe que a garrafa representada pelo objeto ID1 está gelada, e atualiza sua crença sobre ID1. Também passa a crer que existe um objeto ID2 que corresponde a uma garrafa de bebida da marca B à temperatura ambiente e fora do refrigerador, e existe um objeto ID3 que corresponde a um cliente aguardando uma bebida da marca A. Neste momento existem dois desejos candidatos: DES2 e DES3. O desejo DES2 tem sua pré-condição satisfeita pela crença sobre ID2. O desejo DES3 tem suas pré-condições satisfeitas pelas crenças sobre ID1 e ID3. Selecionamos primeiro DES3 pois este possui maior prioridade. O desejo candidato DES3 sobrevive ao filtro e gera a intenção INT3. O próximo passo consiste em selecionar o segundo desejo candidato DES2, o qual é reprovado no processo de filtragem porque é incompatível com INT3. Logo após são executadas as ações da intenção INT3. Ao perceber que a bebida representada pelo objeto ID1 se encontra fora do refrigerador, e o cliente representado pelo objeto ID3 foi atendido, BALBINO considera INT3 realizada.

Em sua terceira percepção, BALBINO passa a crer que existe um objeto ID4 que corresponde a um cliente que quer uma bebida da marca A, no entanto está se comportando de maneira inadequada. Neste momento, há dois desejos candidatos: DES2 e DES5. O desejo DES2 tem sua pré-condição satisfeita pela crença sobre ID2. O desejo DES5 tem sua pré-condição satisfeita pela crença sobre ID4. O desejo candidato DES5, que possui maior prioridade, é selecionado e sobrevive ao processo de filtragem gerando a intenção INT5. O próximo passo consite em selecionar o segundo desejo candidato DES2, que sobrevive ao processo de filtragem, gerando a intenção INT2 que é compatível com INT5. Após executar as ações de INT2 e INT5, o agente BALBINO passa a crer que o cliente representado pelo objeto ID4 está

ausente, e a bebida representada pelo objeto ID2 está dentro do refrigerador, o que torna ambas intenções realizadas.

Em sua quarta percepção, BALBINO percebe que a garrafa representada pelo objeto ID2 está gelada, e atualiza sua crença sobre ID2, que passa a corresponder a uma garrafa de bebida da marca B gelada e dentro do refrigerador. Também passa a crer que existe um objeto ID5 que corresponde a um cliente aguardando uma bebida da marca B. O desejo DES4 é selecionado como candidato, uma vez que suas pré-condições foram satisfeitas pelos objetos ID2 e ID5. O DES4 sobrevive ao processo de filtragem e gera a intenção INT4. Após as ações de INT4 serem executadas, BALBINO percebe que o cliente representado pelo objeto ID5 assumiu o estado ausente. Conseqüentemente, o processo responsável pela manutenção das intenções considera INT4 impossível e a elimina.

7 CONCLUSÕES

O presente trabalho teve como proposta a modelagem de uma arquitetura BDI, e a construção de um ambiente para desenvolvimento de agentes compatível com esta arquitetura.

Devido ao elevado nível de abstração proporcionado pelo ambiente implementado, o desenvolvedor da aplicação pode realizar seu trabalho de maneira simples e rápida, devendo apenas modelar um domínio, fornecer as percepção e obter as ações.

O ambiente proporciona também uma API Java, que permite instanciá-lo dentro agentes disponibilizados por *frameworks* codificados em Java.

Com pequenas modificações é possível resgatar as intenções do agente, ao invés das ações relacionadas às intenções, podendo assim criar externamente um mecanismo de planejamento mais elaborado.

Como trabalhos futuros, espera-se:

- adotar o padrão FIPA, e propor uma sociedade de agentes que se comuniquem através da troca de estados mentais;
- criar opções no raciocinador, possibilitando que sejam escolhidas diferentes estratégias de comprometimento, reconsideração de intenções, permitindo que o agente se adapte às mudanças do ambiente;
- criar uma interface gráfica para acompanhamento dos estados mentais;
- expandir a arquitetura, criando um elaborado mecanismo de raciocínio meio-e-fim.

8 REFERÊNCIAS BIBLIOGRÁFICAS

- [ALG04] **Algernon**. Disponível por WWW em: <http://algernon-j.sourceforge.net/> (14 de setembro de 2004)
- [BRA87] BRATMAN, M. E. **Intention, Plans and Practical Reason**. Cambridge: Harvard University Press, 1987.
- [BRA88] BRATMAN, M. E.; ISRAEL, D. J.; POLLACK, M. E. **Plans and resource-bounded practical reasoning**. *Computational Intelligence*, 4(4):349-355, 1988.
- [BRA90] BRATMAN, M. E. **What is intention?** MIT Press: Cambridge, 1990.
- [BRE98] BRENNER, W.; RÜDIGER, Z.; WITTIG, H. **Intelligent Software Agents. Foundations and applications**. Berlin: Springer-Verlag, 1998.
- [BRO86] BROOKS, R. A. **A robust layered control system for a mobile robot**. In *IEEE Journal of Robotics and Automation*, RA-2(1986)1, 14-23.
- [COR94] CORRÊA, M. **A arquitetura de Diálogos entre Agentes Cognitivos Distribuídos**. Rio de Janeiro: UFRJ, 1994.
- [DEN87] DENNETT, D. C. **The Intentional Stance**. MIT Press: Cambridge, 1987.
- [FRA96] FRANKLIN, S.; GRAESSER, A. **Is it a Agent, or just a Program? A Taxonomy for Autonomous Agents**. 1996.
- [GEO89] GEORGEFF, M. P.; INGRAND, F. F. **Decision-Making in an Embedded Reasoning System**. In: 11th International Joint Conference on Artificial Intelligence, Detroit, Ames Research Center, 1989.
- [GEO99] GEORGEFF, M. P.; POLLACK, M. E.; TAMBE, M.; WOOLDRIDGE, M.; PELL, B. **The Belief-Desire-Intention Model of Agency**. In *Proceedings of the 5th International Workshop on Intelligent Agents V: Agent Theories, architectures and languages*, Springer-Verlag, 1999.

- [GIR99] GIRAFFA, L. M. M. **Uma arquitetura de tutor utilizando estados mentais**. Porto Alegre: CPGCC/UFRGS, 1999. Tese de Doutorado.
- [MAC79] MCCARTHY, J. **Ascribing Mental Qualities to Machines**. In Philosophical Perspectives in Artificial Intelligence. Harvester Press: [s.l.] 1979.
- [MAE95] MAES, P. **Artificial Life Meets Entertainment: Life like Autonomous Agents**. Communications of the ACM, 38, 11, 108-114, 1995.
- [PRO04] **Protégé**. Disponível por WWW em: <http://protege.stanford.edu/> (14 de setembro de 2004)
- [RUS95] RUSSEL, S.; NORVIG, P. **Artificial Intelligence a Modern Approach**. New Jersey: Prentice-Hall, 1995.
- [WOO95] WOOLDRIDGE, M. & JENNINGS, N.R. **Intelligent Agents: Theory and Practice**. Knowledge Engineering Review, Vol. 10, No. 2. Cambridge: Cambridge University Press, 1995.
- [WOO00] WOOLDRIDGE, M. **Reasoning about Rational Agents**. London: The MIT Press, 2000.
- [WOO02] WOOLDRIDGE, M. **An Introduction to Multiagent Systems**. [s.l.]: John Wiley & Sons, 2002.
- [ZAM02] ZAMBERLAM, A. O. **Em direção a uma técnica para programação orientada a agentes BDI**. Porto Alegre: Campus Global/FACIN/PUCRS, 2002. Dissertação de Mestrado.