

ALAINE MARGARETE GUIMARÃES

**AGENTE-M:
Um matriculador inteligente**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Informática. Curso de Pós-Graduação em Informática, Universidade Federal do Paraná.

Orientadora: Dr.^a Aurora Trinidad Ramirez Pozo

CURITIBA
2000



Ministério da Educação
Universidade Federal do Paraná
Mestrado em Informática

PARECER

Nós, abaixo assinados, membros da Banca Examinadora da defesa de Dissertação de Mestrado em Informática da aluna *Alaine Margarete Guimarães*, avaliamos o trabalho intitulado "**Agente-M: Um Matriculador Inteligente**", cuja defesa foi realizada no dia 08 de dezembro de 2000. Após a avaliação, decidimos pela aprovação da candidata.

Curitiba, 08 de dezembro de 2000.

Prof.^a Dra. Aurora Trinidad R. Pozo
Presidente DINF/UFPR

Prof.^a Dra. Silyia Modesto Nassar
Membro Externo - UFSC

Prof. Dr. Alexandre Ibrahim Direne
DINF/UFPR

DEDICATÓRIA

A minha querida mãe, a minha melhor amiga e irmã Alana e ao meu grande amor Alexandre. A vocês dedico essa conquista que é realmente muito importante para mim.

AGRADECIMENTOS

Felizmente, durante todo o desenvolvimento desse mestrado sempre estiveram próximas de mim pessoas especiais que vibraram comigo nos bons momentos e que me ajudaram, compreenderam e incentivaram nas dificuldades.

Ao Deus que está dentro de mim, essa energia que me faz sentir viva e feliz, que me dá força e nunca me abandona, muito obrigado.

Ao meu companheiro e marido do coração Alexandre, agradeço por sempre ter-se preocupado com o meu mestrado, por ter viabilizado a estrutura necessária para sua realização e pelo amor dedicado a mim durante todo esse tempo.

Um agradecimento muito especial a minha querida irmã Alana pelo carinho e incentivo não só durante esse mestrado, mas por toda minha vida. Obrigado pela lealdade e amizade incondicional.

Agradeço a minha mãe pelo exemplo de coragem, honestidade e perseverança. Em todos os momentos difíceis dessa jornada você sempre me incentivou.

Existe uma pessoa que é fundamental nesse trabalho, a quem faço um agradecimento especial: minha orientadora Professora Aurora. Sua capacidade de, apesar do grande conhecimento e potencial, se mostrar sempre flexível e aberta a novas idéias, faz de você uma excelente orientadora. Obrigado pelas boas palavras, pela orientação técnica, pela amizade e incentivo. Eu realmente tenho um grande carinho por você.

Retribuo os agradecimentos ao colega Ivo Mathias; vencemos dificuldades juntos e daí nasceu uma grande amizade. Obrigado pelo apoio e companheirismo.

Quero agradecer aqui ao Professor Elias Duarte Júnior, da UFPR, pelo incentivo constante e interesse sincero durante a fase de qualificação dessa proposta.

Meus agradecimentos ao Professor Roberto Hexsel da UFPR, que é um excelente profissional, pela compreensão no momento em que precisei.

Não poderia deixar de agradecer à colega Tatiana Celinski da UEPG, por sua amizade e pelo incentivo a realizar o mestrado na UFPR.

Agradeço ao colega Pedro Weirich da UEPG, por não ter medido esforços para auxiliar na definição do estudo de caso a ser usado nessa tese.

Aos meus amigos da UEPG pelo apoio e amizade durante esta jornada.

À UEPG pela licença remunerada que viabilizou o meu mestrado.

Meus agradecimentos a Capes, cujos recursos me auxiliaram na realização do trabalho.

Meus agradecimentos a Allegro Common Lisp por cederem uma versão de testes para o desenvolvimento do experimento prático deste trabalho.

Sumário

Lista de Abreviaturas	vii
Lista de Figuras	viii
Lista de Tabelas	x
Resumo	1
Abstract	2
1 Introdução	3
1.1 Contexto do trabalho	3
1.2 Especificação do problema	4
1.3 Objetivos	4
1.4 Organização do trabalho	5
2 Inteligência Artificial Distribuída	7
2.1 Introdução	7
2.2 Conceitos de IAD	7
2.3 Características dos Sistemas Multiagentes	12
2.4 Discussões	15
3 Agentes	16
3.1 Introdução	16
3.2 O que são agentes?	16
3.3 Características dos agentes	19
3.4 Representação do Conhecimento: Ontologia	20
3.5 Comunicação entre agentes	21
3.6 Arquitetura para agentes inteligentes	24
3.7 Discussões	26
4 Arquitetura BDI (Crenças, Desejos e Intenções)	27
4.1 Introdução	27
4.2 Estados Mentais	27
4.2.1 Crenças	28
4.2.2 Desejos	28
4.2.3 Intenções	29
4.3 Composição do agente BDI	30
4.4 DESIRE	32

4.5	Construção de um modelo de agente BDI racional com atitudes motivadoras	38
4.6	Implementação de agentes	41
4.7	Discussões	42
5	AGENTE-M	43
5.1	Introdução	43
5.2	Visão geral do estudo de caso	44
5.3	Projeto do AGENTE-M	45
5.4	Componente Controle do Próprio Processo	46
5.4.1	Determinação das Crenças do AGENTE-M	48
5.4.2	Determinação dos Desejos do AGENTE-M	51
5.4.3	Determinação das Intenções e Compromissos do AGENTE-M	52
5.5	Componente Gerenciamento do Mundo	54
5.6	Componente Gerenciamento de Agentes	56
5.7	Componente Tarefa Específica do Agente	57
5.8	Implementação do AGENTE-M em LOOM	66
5.8.1	Implementação das Crenças	66
5.8.2	Implementação dos Desejos	68
5.8.3	Implementação das Intenções e Compromissos	69
5.9	Implementação do Gerenciamento do Mundo	71
5.10	Implementação do Gerenciamento de Agentes	72
5.11	Implementação das tarefas específicas do AGENTE-M	73
5.12	Discussões	75
6	Resultados: AGENTE-M	77
6.1	Filosofia de operação	77
6.2	Exemplo de uma sessão: matricular aluno aprovado em dependência	82
6.3	Discussões	87
7	Conclusões	88
	Referências Bibliográficas	90

Lista de abreviaturas

ACL	Agent Communication Language (Linguagem de Comunicação de Agentes)
AGENTE-M	Agente Matriculador
AGENT-M	Registrar Agent (Agente Matriculador)
AOP	Agent-Oriented Programming (Programação Orientada a Agentes)
BDI	Beliefs, Desires and Intentions (Arquitetura de Agente de Crenças, Desejos e Intenções)
BRF	Believe Revision Function (Função de Revisão de Crenças)
DESIRE	Framework for Design and Specification on Interaction Reasoning components (ferramenta para projeto e especificação de componentes racionais interagindo)
H&S	Hardware e Software
IA	Inteligência Artificial
IAD	Inteligência Artificial Distribuída
KIF	Knowledge Interchange Format
KQML	Knowledge Query and Manipulation Language
LALO	Language of Agents Logiciel Objet
LINUX	Sistema operacional compatível com o UNIX, criado por Linus Torvalds
LOOM	Linguagem de Representação do Conhecimento
RA	Registro Acadêmico
RDP	Resolução Distribuída de Problemas
SISMAT	Sistema de Matrícula Inteligente
SMA	Sistemas Multiagentes
SYSIR	System of Intelligent Registration (Sistema de Matrícula Inteligente)
UEPG	Universidade Estadual de Ponta Grossa
UFPR	Universidade Federal do Paraná

Lista de Figuras

Figura 1 – Um agente em seu ambiente	18
Figura 2 – Modelo abstrato de interoperação entre agentes de software	22
Figura 3 – Diagrama esquemático de uma arquitetura BDI	30
Figura 4 – Composição de alto nível e links de informação de um agente	36
Figura 5 – Hierarquia das tarefas do componente Controle do Próprio Processo	40
Figura 6 – Ambiente de atuação do AGENTE-M	45
Figura 7 – Refinamento do Controle do Próprio Processo do agente	47
Figura 8 – Crença interna do AGENTE-M	48
Figura 9 – Crença do AGENTE-M baseada em observação	48
Figura 10 – Crença do AGENTE-M baseada no fato comunicado pelo agente usuário	49
Figura 11 – Crenças do AGENTE-M	50
Figura 12 – Determinação de desejo do AGENTE-M	51
Figura 13 – Determinação das Intenções e Compromissos do AGENTE-M	52
Figura 14 – Especificação de Intenção baseada em desejo	53
Figura 15 – Especificação de Intenção baseada em meta atingida	53
Figura 16 – Especificação da interação do AGENTE-M com o mundo	55
Figura 17 - Meta-conhecimento do Componente Gerenciamento do Mundo	55
Figura 18 – Especificação da interação do AGENTE-M com o agente usuário	56
Figura 19 – Meta-conhecimento do Componente Gerenciamento de Agentes	57
Figura 20 – Tarefas específicas do AGENTE-M	58
Figura 21 – Meta-conhecimento para a tarefa específica Gerar_matrícula	59
Figura 23 – Definição das crenças em LOOM	67
Figura 24 – Restrição sobre fato do aluno ser calouro	67
Figura 25 – Implementação do desejo do agente de mostrar dados cadastrais do aluno	69
Figura 26 – Intenção do AGENTE-M de mostrar ao aluno os dados cadastrais	70
Figura 27 – Localização dos dados cadastrais do aluno no mundo	71
Figura 28 - Comunicação do RA a matricular	72
Figura 29 – Implementação da comunicação dos dados cadastrais ao agente usuário	73
Figura 30 – Implementação da tarefa específica do AGENTE-M para gerar matrícula	74

Figura 31 – Agente usuário informa RA e ativa AGENTE-M	77
Figura 32 – Apresentação do RA a matricular	77
Figura 33 – AGENTE-M comunica ao agente usuário que o RA informado é inválido	78
Figura 34 - AGENTE-M comunica ao agente usuário que o RA informado é válido	78
Figura 35 – Dados comunicados ao aluno resultantes da observação do mundo	79
Figura 36 – Agente-M busca dados do histórico do aluno	79
Figura 37 – Comunicação ao agente usuário de que a matrícula não pode ser realizada	80
Figura 38 – Agente matriculador informa decisão de gerar a matrícula	80
Figura 39 - Comunicação ao agente usuário sobre a matrícula gerada	81
Figura 40 - Comunicação ao agente usuário sobre o término da matrícula	82
Figura 41 - Tela 1: Comandos para informar o RA e ativar o AGENTE-M	83
Figura 42 – Tela 2: Identificação e validação do RA	85
Figura 43 – Tela 3: Dados do aluno e sua situação atual	85
Figura 44 – Tela 4: Histórico escolar do aluno e aviso sobre a decisão de gerar matrícula	86
Figura 45 - Tela 5: Matrícula gerada para a 2ª série e disciplinas reprovadas	86

Lista de Tabelas

Tabela 1 – Atributos de Sistemas Multiagentes e suas faixas de potencial	13
Tabela 2 – Classificação das atitudes do agente	33
Tabela 3 – Links para troca de informação entre as tarefas do agente	37
Tabela 4 – Links entre os componentes do AGENTE-M	62
Tabela 5 – Grade curricular do Curso de Informática da UEPG	83

Resumo

Um dos grandes desafios da inteligência artificial (IA) é a simulação do comportamento humano através da implementação de agentes deliberativos. Esses agentes se apresentam como entidades computacionais dotadas de certo grau de autonomia e que raciocinam em função da representação dos estados mentais.

O Grupo de Pesquisas da UFPR (Universidade Federal do Paraná) voltado para a área de Agentes desenvolveu um Sistema de Matrícula Inteligente – SISMAT [MAT00], onde os agentes foram definidos de acordo com os estados mentais de crença, capacidade e compromisso. Observamos que alguns processos importantes, como o de planejamento, não foram implementados e que não existe um raciocínio claro sobre a interação do agente com seu meio.

Foi então projetado um agente deliberativo que simule esses estados, incluindo o planejamento e tenha seu conhecimento representado de forma explícita, dividido em componentes específicos através dos quais possa haver controle sobre ele próprio, e sobre sua interação com o mundo e com outros agentes.

O agente desenvolvido, denominado AGENTE-M, simula o comportamento de um matriculador humano realizando o processo de matrícula de aluno em uma universidade. Suas tarefas estão distribuídas em componentes específicos, através dos quais mantém um raciocínio e controle sobre seu próprio processo, sobre o agente usuário com o qual interage e sobre o mundo. O agente usuário representa o aluno que deseja se matricular, e o mundo corresponde à base de dados referente ao curso, disciplinas e histórico escolar.

O projeto do AGENTE-M foi baseado na arquitetura BDI (Beliefs, Desires and Intentions) proposta pelo framework DESIRE (DEsign and Specification of Interactive REasoning components), e seu protótipo foi desenvolvido na linguagem de representação de conhecimento LOOM.

Os resultados obtidos mostraram que a arquitetura BDI torna o agente mais poderoso em termos de raciocínio, que seu projeto em componentes específicos facilita sua compreensão e manutenção, e que a linguagem LOOM permitiu implementar o agente de acordo como foi projetado, com algumas restrições.

ABSTRACT

One of the greatest challenges of the Artificial Intelligence (AI) is the human behavior simulation across the intelligent agents implementations. The agents are like computational entities with some degree of autonomy that reason in function of the representation of the mental states.

The Group of Researches of UFPR (Federal University of Paraná) went back to the area of Agents and developed a System of Intelligent Registration - SYSIR [MAT00], where the agents were defined in agreement with the mental states of beliefs, capabilities and commitments. We observed that some important process, as planning, were not implemented and that a clear reasoning doesn't half exist about the agent's interaction with its.

We decided then to project a deliberative agent that simulates those states and have its represented knowledge in an explicit way, divided in specific components through which it can have control on itself, and about its interaction with the world and other agents.

The developed agent, denominated AGENT-M, simulates the behavior of a human accomplishing the process of student's registration in an university. Its tasks are distributed in specific components, through which it maintains a reasoning and control on its own process, the agent user with which interage and the world. The agent user represents the student that wants to register, and the world corresponds to the base of data regarding the course, disciplines and school records.

The project of the AGENT-M is based on the BDI (Beliefs, Desires and Intentions) architecture proposed by the framework DESIRE (Design and Specification of Interactive REasoning components), and its prototype was developed at the knowledge representation LOOM language.

The obtained results showed that the BDI architecture turns the most powerful agent in terms of reasoning, its project in specific components facilitates its understanding and maintenance, and the language LOOM allowed to implement the agreement agent as it was projected with some restrictions.

1. Introdução

1.1 Contexto do trabalho

O estudo de agentes ocorre há alguns anos, mas atualmente esse paradigma está se tornando mais popular em função de sua generalização, flexibilidade, modularidade, e habilidade para modelar resolução de problemas distribuídos [NEW90].

A linha de pesquisa de IA que trata desses aspectos é a Inteligência Artificial Distribuída (IAD), onde a inteligência é distribuída utilizando vários agentes, dando origem ao conceito de Sistemas Multiagentes.

O grande desafio dos estudos sobre agentes está relacionado atualmente à representação das capacidades de autonomia, interação, cooperação e adaptação que o agente deve apresentar para poder simular o comportamento humano.

Uma das abordagens que visa essa representação, trata da incorporação de estados mentais ao conhecimento do agente, que correspondem às crenças, desejos e intenções que uma pessoa possui em relação a uma tarefa a ser executada, ou a uma meta pretendida [MOR97].

O Grupo de Pesquisas da UFPR voltado para a área de Agentes desenvolveu um Sistema de Matrícula Inteligente – SISMAT [MAT00], onde os agentes foram definidos em função de estados mentais. Observamos que alguns estados importantes, como planejamento, não foram implementados e que, embora o agente matriculador interagisse com outros agentes, não existe um raciocínio explícito sobre essa interação.

Decidimos então projetar um agente deliberativo que simule esses estados e tenha seu conhecimento representado de forma explícita, dividido em componentes específicos através dos quais possa ter controle sobre ele próprio, e sobre sua interação com o mundo e com outros agentes.

Para tanto é necessário a busca de técnicas que permitam ao agente apresentar as seguintes características comportamentais:

- Raciocinar sobre seu próprio conhecimento
- Ter autonomia para deliberar sobre a realização ou não de determinada ação
- Representar explicitamente em sua composição os seus desejos, crenças e intenções, assim como sua visão do mundo e de outros agentes.

Baseado nesse contexto estudaremos uma arquitetura de agente para solucionar o problema de simular o comportamento humano em um processo de matrícula de aluno em uma universidade. Esse problema será discutido brevemente na próxima seção.

1.2 Especificação do problema

O processo de matricular um aluno requer exaustivo conhecimento sobre o acadêmico, o curso e as regras existentes. Exige também poder de raciocínio e autonomia para tomar decisões.

O matriculador precisa:

- trocar informações, ou seja, interagir com o aluno que deseja ser matriculado,
- acessar uma base de dados para localizar informações referentes ao curso, cadastro e histórico escolar do aluno,
- decidir sobre a possibilidade ou não de matricular o aluno,
- realizar a matrícula de acordo com as condições existentes e
- registrar a matrícula gerada nessa mesma base de dados.

Assim, projetar um agente que execute essa tarefa de matrícula passa a ser um excelente caso a ser estudado, uma vez que será necessário:

- desenvolver um agente autônomo com capacidade de interação
- incorporar a esse agente estados mentais para melhor simular o comportamento do matriculador humano, e
- projetá-lo para ser um agente de fácil compreensão e manutenção tendo seu raciocínio representado da forma mais explícita possível.

Com base nesse contexto podemos estabelecer os objetivos que norteiam esse trabalho.

1.3 Objetivos

O objetivo fundamental desse trabalho é projetar um agente autônomo e cognitivo (deliberativo)¹, representando explicitamente seus estados mentais, e distribuindo suas tarefas

¹ Nesse trabalho usaremos os termos deliberativo e cognitivo indistintamente, apesar do raciocínio do agente deliberativo estar relacionado aos estados mentais e do cognitivo ser baseado em conhecimento e não envolver necessariamente os estados mentais.

em componentes específicos que possam relacionar-se entre si, com outros agentes e com o mundo.

Para tanto, apresentamos um estudo sobre IAD e Sistemas Multiagentes, concentrando nossos esforços na compreensão de agentes autônomos e na identificação de quais características se adaptam ao nosso problema.

Como pretendemos simular o comportamento humano, é necessário o estudo dos estados mentais que podem interferir na atuação do agente, verificando como relacionar tais estados com as tarefas que o agente deve executar.

Outro aspecto a ser estudado é a arquitetura a ser utilizada para projetar o agente, permitindo que o raciocínio sobre si mesmo, outros agentes e o mundo seja representado explicitamente.

Finalmente, apresentamos um protótipo do agente projetado segundo esses princípios e discutiremos os efeitos da modelagem em sua implementação e atuação.

Minsky, um dos mais eminentes pesquisadores da área de IA, em seu livro *The Society of Mind* (A Sociedade da Mente) [MIN85], já explorava a noção de agentes ao considerar que inteligência não é um produto de um mecanismo singular, mas se origina da interação entre uma grande variedade de agentes heterogêneos que simultaneamente cooperam e competem. Isso nos motiva a acreditar que o estudo aprofundado de agentes pode cada vez mais aproximar as atitudes computacionais das atitudes humanas.

1.4 Organização do trabalho

Esse trabalho está organizado em seis capítulos, onde os seguintes assuntos serão abordados :

Capítulo 1: corresponde ao presente capítulo, que contém a introdução, a apresentação do problema proposto, os objetivos desse trabalho e sua organização.

Capítulo 2: apresentação do estado da arte da inteligência artificial distribuída, introduzindo os conceitos e características de Sistemas Multiagentes.

Capítulo 3: abordagem sobre agentes enfocando suas características, aspectos de representação do conhecimento e de comunicação entre agentes, e finalmente os tipos de arquitetura pesquisados atualmente.

Capítulo 4: discute-se a arquitetura BDI baseada em estados mentais, e a proposta do DESIRE para projeto de um agente com atitudes motivadoras.

Capítulo 5: apresentação do desenvolvimento do AGENTE-M, discussão os aspectos importantes referentes ao seu projeto e implementação.

Capítulo 6: Filosofia de operação do AGENTE-M e exemplo de uma sessão de matrícula.

Capítulo 7: Conclusões , dificuldades encontradas e limitações deste trabalho.

2. Inteligência Artificial Distribuída

2.1 Introdução

A Inteligência Artificial Distribuída (IAD) é o estudo do comportamento computacionalmente inteligente, resultante da interação de múltiplas entidades dotadas de certo grau, possivelmente variável, de autonomia. Estas entidades são usualmente chamadas de agentes e o sistema como um todo, sociedade [OLI96].

Desde que foi concebida em meados da década de 70, a IAD tem evoluído e diversificado muito. Hoje ela é um campo promissor de pesquisa e aplicação que envolve conceitos e idéias de muitas disciplinas, incluindo Inteligência Artificial (IA), ciência da computação, sociologia, economia e filosofia [HUH98].

O enfoque central da IAD, que consiste em ter a inteligência distribuída em diferentes entidades, deu origem aos Sistemas Multiagentes (SMA) que apresentam características importantes para o desenvolvimento de modelos e teorias de interatividade em sociedades humanas [WEI99].

Segundo Weiss [WEI99] os SMA têm capacidade para desempenhar um papel chave na ciência da computação corrente e futura porque são desenvolvidos em ambientes distribuídos e heterogêneos, se ajustando ao fato de que os computadores não são mais sistemas isolados, mas vem sendo fortemente conectados com outros e seus usuários.

Nesse capítulo veremos os principais conceitos e características da IAD, sua evolução e suas diferenças em relação a IA tradicional. Discutiremos também as principais características dos SMA

2.2 Conceitos de IAD

A IAD está voltada para a solução colaborativa de problemas globais por um grupo de entidades distribuídas, que variam de simples elementos de processamento a entidades racionais. A atividade é colaborativa no sentido de que é necessário o compartilhamento de informações e o grupo é distribuído no senso de que o controle e os dados são lógicos e algumas vezes geograficamente distribuídos [DEM90].

Por ser um campo dinâmico, de natureza interdisciplinar, existem diferentes definições para a IAD. Encontramos em [WEI99] a definição a seguir, diretamente relacionada a SMA, a qual servirá como ponto de referência ao longo desse trabalho.

“IAD é o estudo, construção e aplicação de Sistemas Multiagentes, isto é, sistemas em que vários agentes inteligentes interagem para atingir determinado conjunto de metas ou para executar algum conjunto de tarefas.[WEI99]”

Para facilitar a compreensão da definição acima consideremos agentes inteligentes como entidades que operam flexivelmente e racionalmente em uma variedade de circunstâncias ambientais dado seu comportamento perceptivo e eficiente, sendo autônomo e dependendo, pelo menos parcialmente, de sua própria experiência [JEN98].

Os conceitos de agentes são de grande importância para a IAD, e são também muito complexos. Por isso dedicaremos todo o capítulo 3 para detalhar e discutir essas entidades que constituem o enfoque central desse trabalho.

A pesquisa em IAD concentra-se em processos como resolução de problemas, busca, planejamento, tomada de decisão e aprendizagem, que possibilitam aos agentes mostrar racionalidade e flexibilidade em sua atuação, e na realização de tais processos em cenários Multiagentes.

Existem várias vantagens em se distribuir o conhecimento em diferentes entidades (módulos). Rich em [RICH93] destaca as seguintes:

- **Modularidade do sistema:** é mais fácil criar e manter uma coleção de módulos quase independentes do que um módulo único.
- **Eficiência:** nem todos os conhecimentos são necessários para todas as tarefas. Quando o sistema é modularizado, tem-se a possibilidade de focalizar os esforços do sistema de solução de problemas do modo mais compensador possível.
- **Arquiteturas rápidas:** quando os resolvedores de problemas começam a ficar complexos, eles passam a precisar de mais e mais ciclos de processamento. Apesar das máquinas estarem cada vez mais rápidas, a real aceleração vem não de um único processador com uma enorme memória associada a ele, mas de um agrupamento de processadores menores, cada um com memória própria. Os sistemas distribuídos têm melhores condições de explorar essas arquiteturas.

- **Raciocínio heterogêneo:** as técnicas de solução de problemas e os formalismos de representação de conhecimento que mais se prestam para trabalhar em uma parte de um problema podem não ser os melhores para trabalhar em outra parte.
- **Múltiplas perspectivas:** o conhecimento exigido para solucionar um problema pode não estar na cabeça de um única pessoa. É muito difícil juntar diversas pessoas para criar uma única base de conhecimento coerente e às vezes essa reunião é até mesmo impossível, porque os modelos que cada uma delas tem sobre o domínio em questão são inconsistentes.
- **Problemas distribuídos:** alguns problemas são inerentemente distribuídos;
- **Confiabilidade:** se um problema está distribuído entre agentes de sistemas diferentes, a solução pode continuar mesmo que um dos sistemas apresente falhas.

As linhas de pesquisa em IAD podem ser divididas, segundo Bond & Gasser [BOND88], em cinco áreas diferentes:

1. **Descrição, decomposição e alocação de tarefas:** refere-se a como pode ser descrita e decomposta uma tarefa complexa em subtarefas, como essas subtarefas serão alocadas, estática ou dinamicamente e como elas serão realizadas;
2. **Interações, linguagens e comunicações:** que vocabulário de primitivas um protocolo de comunicação deve apresentar de forma a expressar os conceitos semânticos de um trabalho cooperativo;
3. **Coordenação, controle e comportamento coerente:** como assegurar um comportamento global coerente do conjunto de agentes, cada agente possuindo seus próprios objetivos e habilidades locais;
4. **Conflito e incerteza:** como nenhum agente possui a informação completa do ambiente, nascem conflitos e estes devem ser resolvidos. Os dados incompletos e incertos devem ser tratados de forma a garantir resultados coerentes;

5. **Linguagem de programação e ambiente:** devem ser definidas linguagens de programação que suportem os diferentes requisitos necessários a cada agente pertencente ao sistema.

Tradicionalmente, dois tipos de sistemas de IAD têm sido distinguidos [WEI99]:

- SMA (Sistemas Multiagentes): pode ser definido como um sistema no qual diversos agentes coordenam seu conhecimento e atividades, e raciocinam sobre o processo de coordenação [DEM90].
- RDP (Resolução Distribuída de Problemas): os agentes cooperam um com os outros, dividindo e compartilhando conhecimento sobre o problema e sobre o processo para obter uma solução. Os agentes são projetados para resolver um problema ou uma classe de problemas específicos, possuindo porém uma visão incompleta do objetivo global [DEM92].

Inicialmente, a ênfase do trabalho relacionado a SMA foi sobre coordenação de comportamento, e o enfoque central dos sistemas de RDP foi sobre decomposição de tarefa e sínteses de solução.

Atualmente, não se faz distinção explícita entre Sistemas Multiagentes e Sistemas de Resolução Distribuída de Problema, sendo esses termos usados sinonimamente [WEI99].

O papel que o conceito de um SMA desempenha em IAD é comparável ao papel que o conceito de um agente individual desempenha na IA tradicional [NIL98, POO98, RUS95].

Em geral, ambas IAD e IA tradicional trabalham com aspectos computacionais da inteligência, mas sob diferentes pontos de vista e diferentes suposições:

- A IA tradicional concentra-se em agentes como “sistemas inteligentes isolados” e em inteligência como uma propriedade de sistemas que agem isolados;
- A IAD concentra-se em agentes como “sistemas inteligentes conectados” e em inteligência como uma propriedade de sistemas que interagem;
- A IA tradicional enfoca sobre “processos cognitivos” dentro de indivíduos;
- A IAD enfoca sobre “processos sociais” em grupos de indivíduos.
- A IA tradicional considera sistemas tendo um único local de raciocínio e controle interno e requer auxílio mínimo de outros para agir com sucesso;

- A IAD considera sistemas nos quais raciocínio e controle são distribuídos e a atividade bem sucedida é um esforço conjunto.
- A IA tradicional usa psicologia e comportamento como idéias, inspiração e metáforas, a IAD usa sociologia e economia.

2.3 Características dos Sistemas Multiagentes

No paradigma SMA o enfoque principal consiste em coordenar o comportamento inteligente de um conjunto de agentes autônomos, para obter a solução de um problema apresentado. Demazeau [DEM92] expõe que os agentes são mais independentes, mas estão competindo pelos mesmos recursos limitados, tais como tempo, espaço, ferramentas (beneficiando-se de resultados de outros agentes, ou ajudando-os) e tem que evitar conflitos, e apenas em situações extremas eles resolvem problemas individualmente.

Em [JEN98] são identificadas as seguintes principais características de Sistemas Multiagentes:

- Cada agente tem informação incompleta e é restrito em suas capacidades
- O controle do sistema é distribuído
- Os dados são descentralizados
- A computação é assíncrona.

Existem três fatores determinantes da estrutura dos SMA: a composição dos seus agentes, suas interações e seus ambientes de atuação [WEI99]. A tabela 1 apresenta os principais atributos que definem as características desses fatores. Nota-se que cada atributo pode receber diferentes valores dentro de sua faixa potencial. Os agentes podem, por exemplo ser homogêneos com comportamento reativo e com habilidades simples, ou ainda heterogêneos, deliberativos e com habilidades avançadas.

Assim, cada Sistema Multiagente pode ter uma combinação diferente dos atributos e potenciais, não sendo obrigatório a especificação de todos os atributos constantes dessa tabela. Uma visão extensiva dessa tabela é oferecida em [HUH98].

FATORES DETERMINANTES DA ESTRUTURA DOS SMA		
	Atributo	Faixa
Agentes	Número	de dois acima
	Uniformidade	homogênea ... heterogênea
	Metas	Contraditórias ... complementares
	Arquitetura	Reativa ... deliberativa
	Habilidades (sensores, efetores, cognição)	Simple ... avançada
Interação	Frequência	Baixa ... alta
	Persistência	Curto-termo ... longo-termo
	Nível	Passando-sinal ...intensivo-conhecimento
	Padrão (fluxo de dados e controle)	Decentralizado ... hierárquico
	Variabilidade	Fixa ... mudável
	Propósito	Competitivo ... cooperativo
Ambiente	Predição	Previsível ...imprevisível
	Acessibilidade, conhecimento	Ilimitado ...limitado
	Dinâmica	Fixa...variável
	Diversidade	Pobre...rica
	Disponibilidade de recursos	Restrita ...ampla

Tabela 1 – Atributos de Sistemas Multiagentes e suas faixas de potencial [HUH98].

Um ponto importante, considerado como padrão de interação em SMA, é a coordenação orientada por tarefas e metas, que é vista como uma forma de interação que é particularmente importante com relação a alcançar metas e completar tarefas, com o propósito de ativar ou evitar estados de acontecimento que são considerados como desejáveis ou não por um ou vários agentes [LES99].

Existem dois padrões básicos contrastantes de coordenação que são: cooperação e competição. Vejamos alguns detalhes sobre eles:

Cooperação: vários agentes tentam combinar seus esforços para executar como um grupo o que um indivíduo não consegue, no intento de atingir uma meta comum, falhando ou tendo sucesso juntos.

Competição: vários agentes trabalham em oposição porque suas metas são conflitantes. Os agentes competitivos tentam maximizar seu próprio benefício sobre a perda dos outros, e então o sucesso de um implica no fracasso dos outros.

2.4 Discussões

A IAD é um campo de pesquisa aberto e dinâmico que trata do estudo da solução de problemas através da distribuição de conhecimento entre diversas entidades.

As metas essenciais da IAD são as de desenvolver mecanismos e métodos que permitam aos agentes interagir como humanos, e entender a interação entre entidades inteligentes, quer sejam elas computacionais, humanas ou ambos. Essas metas estão longe de serem atingidas, existindo muitos desafios a serem vencidos, os quais se constituem hoje em diferentes áreas de pesquisa. A área que estudaremos em nosso trabalho trata da descrição, decomposição e alocação de tarefas, onde buscaremos uma arquitetura para agentes que possibilite que suas tarefas sejam distribuídas em componentes específicos.

Ao analisar a forma com a IA tradicional e a IAD atuam concluímos que a IAD não é uma especialização da IA tradicional, mas sim uma generalização dela.

Podemos ver também que originalmente a IAD era decomposta em duas áreas: RDP e SMA. Hoje essas áreas são tratadas sinonimamente. Assim, consideramos que ao falarmos sobre IAD estamos nos referindo essencialmente a SMA e vice-versa.

Através das características dos Sistemas Multiagentes verificamos que seu principal enfoque é a cooperação e interação entre agentes que são considerados um dos fatores determinantes da estrutura de um SMA.

A próxima seção apresenta uma abordagem sobre agentes, visto que são os elementos essenciais da IAD.

3. Agentes

3.1 Introdução

A tecnologia de agentes visa criar sistemas mais inteligentes e autônomos para agirem corretamente mesmo que o ambiente no qual estejam inseridos sofra mudanças.

Normalmente a idéia de computadores como sendo obedientes, literais, serviçais sem imaginação, é bem aceita. Para muitas aplicações, como processamento de folha de pagamento, isso é inteiramente aceitável. Entretanto, para um grande número de aplicações, precisamos de sistemas que possam *decidir por si mesmos* o que eles precisam fazer para satisfazer seus objetivos de projeto. Tais sistemas computacionais são conhecidos como *agentes* [WEI99].

Agentes que precisam operar robustamente em mudanças rápidas, em imprevistos, ou em ambientes abertos, onde existe uma possibilidade significativa que ações falhem, são chamados *agentes inteligentes*, ou algumas vezes *agentes autônomos*.

Neste capítulo abordaremos agentes inteligentes, especificamente, suas capacidades e características, com o objetivo de entender porque são vistos como um importante novo modo de conceptualizar e implementar certos tipos de aplicação de software;

3.2 O que são agentes?

Um modo óbvio de começar a falar sobre agente é apresentando uma definição do termo *agente*. Surpreendentemente, não há uma definição aceita universalmente, sendo que o único consenso geral é que a autonomia é a sua noção central.

Parte dessa dificuldade é porque vários atributos associados a agentes são de diferente importância para diferentes domínios. Para algumas aplicações, a habilidade de agentes de aprender através de suas experiências é de grande importância; para outras aplicações, aprender além de ser desnecessário, é indesejável [WEI99].

Como vemos em Giraffa [GIR99], cada grupo de pesquisa segue uma determinada linha apresentando sua definição personalizada do termo agente de acordo com seus próprios objetivos. Entretanto, alguma ordem de definição é importante – caso contrário, é perigoso que o termo perca todos os significados. A seguir apresentam-se algumas das definições mais aceitas, sendo que em cada uma podemos ver um diferente aspecto sendo destacado:

- Um agente é um sistema capaz de perceber através de sensores e agir em um dado ambiente através de atuadores [RUS95].
- Pode-se resumir o termo agente como um processo composto por software, hardware ou por uma combinação de ambos, capaz de realizar uma determinada tarefa e disponibilizar seus serviços a uma coletividade de agentes, que em conjunto, realizam um objetivo comum [POZO95].
- Um agente é uma entidade real ou virtual que está imersa em um ambiente onde ele pode realizar ações, que é capaz de perceber e representar parcialmente este ambiente, de comunicar-se com outros agentes e que possui um comportamento autônomo, que é consequência de suas observações, seu conhecimento e interações com outros agentes [FER91].
- Agentes são entidades computacionais ou não, que desempenham o papel de executores de tarefas, particularmente ou em conjunto, para resolver problemas pertinentes a sistemas baseados em conhecimento, i.e. aplicando conhecimento, atitudes sociais e mentais sobre o propósito da ação [MAT00].
- Agente é uma entidade cujo estado é visto como consistindo de componentes mentais tais como crenças, capacidades, escolhas, e compromissos [SHO93].

Como o termo agente em si, a autonomia também é um conceito difícil de se definir precisamente. Demazeau [DEM92] expõe a autonomia como o agente tendo sua própria existência, independente da existência de qualquer problema a resolver. Neste trabalho ela é usada para indicar que agentes são capazes de agir sem a intervenção de humanos ou outros sistemas; eles tem controle sobre seu próprio estado interno e sobre seu comportamento.

A figura 1 fornece uma visão abstrata, de alto nível de um agente. Neste diagrama podemos ver que a posição do agente em relação ao ambiente é a de senti-lo e gerar ações de saída para afetá-lo [WEI99].

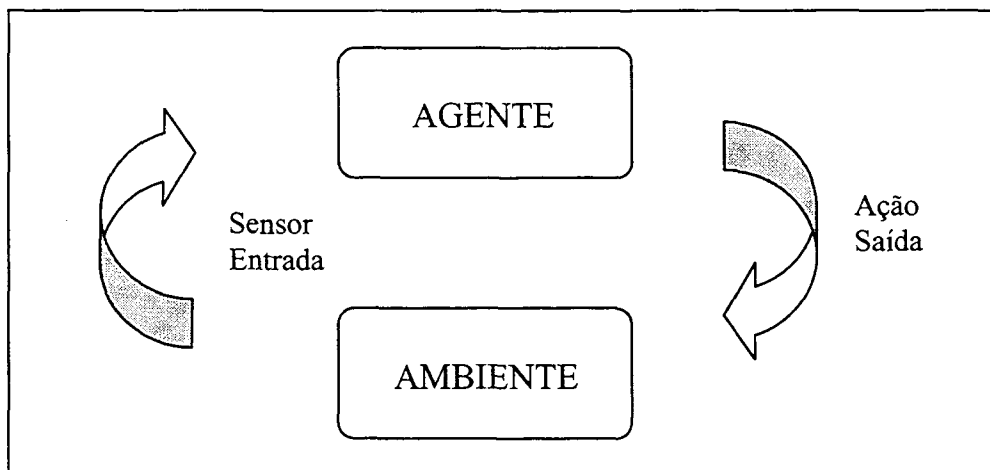


Figura 1 - Um agente em seu ambiente.

Em muitos domínios de raciocínio complexo, um agente não terá controle completo sobre seu ambiente. Ele terá um melhor controle parcial, naquilo que ele pode influenciá-lo. Do ponto de vista do agente isto significa que a mesma ação executada duas vezes em circunstâncias aparentemente idênticas podem ter aparentemente efeitos diferentes, e em particular, podem falhar para ter o efeito desejado. Então agentes precisam ser preparados para a possibilidade de falha, no mais trivial dos ambientes [FERG95].

Normalmente um agente terá um repertório de ações disponíveis para ele, que representam a *capacidade efetora* dos agentes: suas habilidade para modificar o ambiente. Nem todas as ações podem ser executadas em todas as situações; elas possuem pré-condições associadas que definem as possíveis situações nas quais elas pode ser aplicadas.

O principal problema para mapear um agente é a decisão de quais ações ele deve executar para melhor satisfazer seus objetivos projetados. A complexidade dos processos de tomada de decisão pode ser afetada por um número de diferentes propriedades do ambiente no qual o agente está presente.

Considera-se o ambiente do agente como sendo tudo que o envolve, muitas vezes modelado também como um agente. Nesse ambiente ocorre a dispersão do controle, dos dados e do conhecimento pela comunidade de agentes que o compõem.

O ambiente pode ou não incluir outros agentes, podendo ser classificado como *determinístico*, *estático* ou *dinâmico* [GIR99]. Russel e Norvig sugerem uma classificação de propriedades do ambiente que pode ser vista em [RUS95], que inclui ambientes *acessíveis*, *episódicos* e *discretos*.

A partir das definições expostas podemos analisar mais detalhadamente o agente, verificando como suas características podem influenciar na decisão sobre como projetar um agente inteligente.

3.3 Características dos agentes

Um agente deve apresentar algumas características que definem qual é a classificação do agente, ou seja, um agente inteligente, reativo, deliberativo, entre outros.

Demazeau [DEM92] considera as seguintes características pertinentes aos agentes:

- **Capacidade de percepção:** Um agente deve perceber o ambiente e as mudanças ocorridas neste.
- **Capacidade de comunicação:** Um agente deve ser capaz de comunicar-se com seu meio e com os outros agentes que interagem nele.
- **Capacidade de ação:** Um agente deve ser capaz de atuar em seu meio, como resultado de suas atividades de resolução de problemas.
- **Raciocínio Social:** Um agente deve ser capaz de raciocinar acerca de atividades de outros agentes. Isto é conseguido através de uma representação interna dos outros membros da sociedade e mecanismos de raciocínio relacionados.
- **Estrutura de controle de multiníveis:** Um agente deve decidir quando perceber, comunicar, planejar e atuar. Assim, uma estrutura de controle complexa deve ser fornecida de maneira a possibilitar a ativação destas diferentes características.

Os objetivos principais de um agente podem ser resumidos da seguinte forma: realizar as tarefas para as quais foi especificado, e cooperar com os demais agentes da rede, de forma a fornecer as informações e/ou prestar serviços para que os outros agentes possam realizar suas tarefas de forma mais eficiente.

Para atingir esses objetivos o agente deve saber como interagir, trocar conhecimento e cooperar com os demais agentes da rede. É necessário portanto, uma definição de padrões de representação do conhecimento a ser compartilhado, assunto esse que será abordado na sessão a seguir.

3.4 Representação do Conhecimento: Ontologia

Toda base de conhecimento ou agente em termos de conhecimento está comprometido, explícita ou implicitamente, com alguma conceitualização que corresponde a uma visão abstrata, simplificada do mundo que queremos representar para algum propósito [FEN98].

Uma ontologia é justamente uma especificação explícita de uma conceitualização. Isto é, uma ontologia é uma descrição dos conceitos e relacionamentos que podem existir para um agente ou uma comunidade de agentes. O termo é trazido da filosofia, onde uma ontologia é uma consideração sistemática da Existência [GRU93].

Para sistemas em IA , o que "existe" é aquilo que pode ser representado. Quando o conhecimento de um domínio é representado num formalismo declarativo, o conjunto de objetos que pode ser representado é chamado universo de discurso. Este conjunto de objetos, e os relacionamentos descritos entre eles, são refletidos no vocabulário representacional de programas baseados em conhecimento [GUA97].

As ontologias são usadas na descrição de compromissos ontológicos para um conjunto de agentes que podem comunicar sobre um domínio de discurso sem necessariamente operar numa teoria compartilhada globalmente. Considera-se compromisso ontológico um acordo para usar um vocabulário, isto é, fazer perguntas e suposições, num modo que seja consistente (mas não completo) com respeito a teoria especificada por uma ontologia [NEW90].

Pragmaticamente, uma ontologia define o vocabulário com o qual perguntas e asserções são trocadas entre agentes, enquanto que os compromissos ontológicos pactuam para usar o vocabulário compartilhado numa maneira coerente e consistente [VAN98].

Gruber [GRU92] considera que uma das características mais atraentes do uso de ontologias é justamente a possibilidade de reuso do conhecimento, mas existem vários problemas técnicos no modo de compartilhar e reusar softwares baseados em conhecimento, que são normalmente baseados em plataformas de hardware , linguagens de programação e protocolos de rede heterogêneas.

Os principais tipos de ontologia, que podem ser vistos detalhadamente em [GRU93], são: de tarefa, de domínio, técnica básica e genérica.

Uma vez definida a ontologia acerca do conhecimento a ser manipulado pelos agentes, a comunicação entre eles deve ser estabelecida para que exista efetivamente

compartilhamento e cooperação. A seguir veremos os principais aspectos relacionados a comunicação entre agentes.

3.5 Comunicação entre agentes

Na comunidade da IAD existe um consenso sobre a necessidade explícita de comunicação para que haja a cooperação entre agentes autônomos. É de comum acordo, também que comunicação em IAD é mais complexa do que aquela existente entre sistemas distribuídos tradicionais.

A maioria das propostas de comunicação entre agentes apresenta uma relação estrita com a teoria da ação da fala [COH79], porém a alta subjetividade existente na comunicação por linguagem natural, leva a ambigüidades e dificuldades em sua interpretação, tornando-se proibitivas as abordagens por estes meios [LAB97].

Como mostra a figura 2 a comunicação é composta por três partes:

- **Protocolo de Transporte:** É o mecanismo de transporte usado para a comunicação como TCP, SMTP, http, etc.
- **Linguagem de Comunicação:** É o método pelo qual são comunicadas as atitudes relativas ao conteúdo da troca.
- **Protocolo de Interação:** Se refere a estratégia de alto nível, procurada pelo agente de software que governa sua interação com outros agentes.

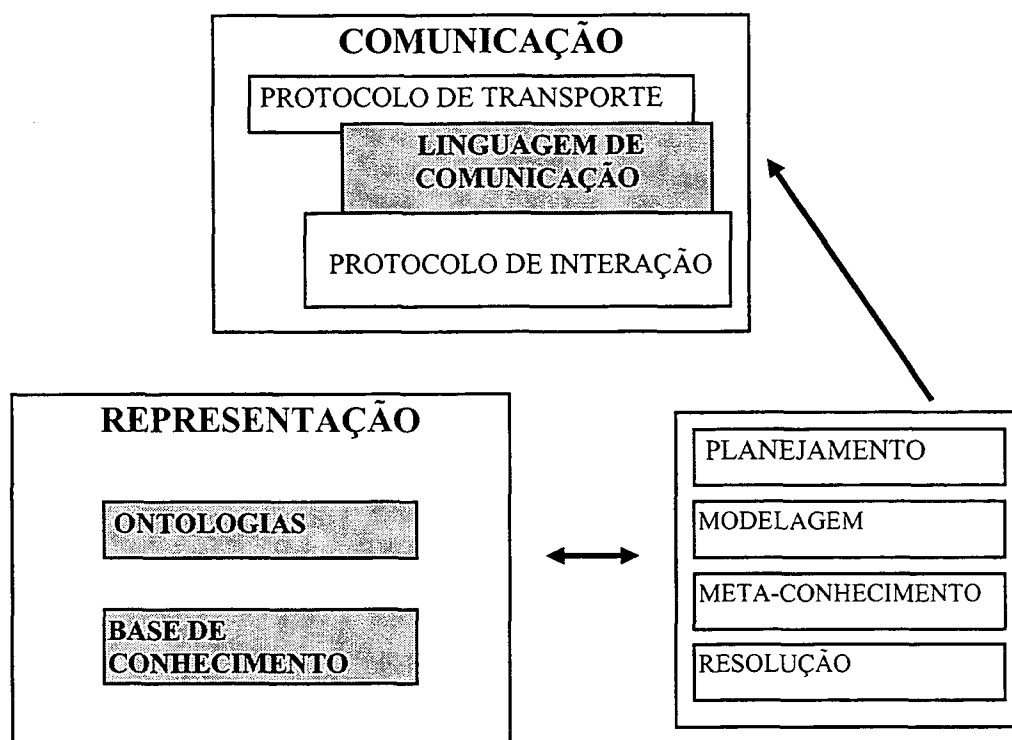


Figura 2 - Modelo abstrato de interação entre agentes de software [LAB97].

Para a comunicação a nível de conhecimento, são necessárias convenções em três níveis: o protocolo de comunicação de agentes, a especificação do conteúdo compartilhado, e o formato da linguagem.

Na literatura existente[LAB97, SIN92, SMI95], muitos autores concentram-se na descrição de protocolos para tarefas específicas. Vale destacar que protocolos de diferentes autores são incompatíveis, pois cada um deles está interrelacionando detalhes de baixo nível da implementação (por exemplo, sintaxe das mensagens) com aspectos do diálogo cooperativo de diferentes formas.

A especificação do conteúdo, que corresponde a ontologia, foi discutida na seção anterior, e com relação a linguagem de comunicação, pesquisadores tem definido uma linguagem chamada ACL (Agent Communication Language), que consiste de três partes[LAB97]:

- Vocabulário;
- Linguagem interna chamada KIF (Knowledge Interchange Format)², que descreve as ontologias, discutidas na seção 3.4;
- Linguagem externa chamada KQML (Knowledge Query and Manipulation Language)³, que descreve as performativas, ou seja, o tipo de intercâmbio (tell, ask) a ser utilizado.

Uma mensagem ACL é uma expressão KQML onde qualquer dos argumentos são termos ou sentenças no formato KIF de palavras do vocabulário.

Após termos apresentado uma visão geral sobre agentes e seus aspectos de representação de conhecimento e cooperação, estudaremos a seguir a arquitetura do agente em si.

² KIF – é um padrão de formato de estrutura do conhecimento a ser compartilhado [GEN92].

³ KQML - é uma linguagem projetada para suportar interações entre agentes inteligentes, tendo sido desenvolvida por um grupo de pesquisadores interessados em sistemas de comunicações entre agentes do Knowledge-Sharing Effort (KSE)[LAB97].

3.6 Arquitetura para agentes inteligentes

Antes de discutir a arquitetura de um agente inteligente é necessário explicar que, para os propósitos do nosso trabalho, um agente inteligente é aquele capaz de ações autônomas flexíveis para atingir seus objetivos projetados, onde flexibilidade significa três coisas [WOO95]:

- **Reatividade:** capacidade de perceber seu ambiente e responder rapidamente a mudanças que ocorram para satisfazer seus objetivos projetados;
- **Pró-atividade:** capacidade de exibir comportamento dirigido por metas tomando a iniciativa para satisfazer seus objetivos projetados;
- **Habilidade social:** capacidade de interagir com outros agentes (e possivelmente humanos) para satisfazer seus objetivos projetados.

De acordo com Brooks [BRO91] e Hayes-Roth [HAY91] um agente autônomo deve ser:

- **Adaptativo:** ele precisa se adaptar apropriadamente e rapidamente a mudanças no ambiente.
- **Robusto:** mudanças de pouca importância não devem levar o comportamento do agente a um colapso total.
- **Tático:** ele deve ser capaz de manter múltiplas metas e, dependendo das circunstâncias que ele encontrar, mudar que metas particulares ele mantém ativas
- **Versátil:** ele precisa ser capaz de executar uma grande variedade de tarefas, ao invés de ser de propósito único).

Segundo Davidsson [DAV96] é muito difícil existirem agentes com todas estas características, as quais devem ser vistas como diretrizes. Entretanto, estas características não são independentes nem exaustivas.

O agente autônomo normalmente apresenta um comportamento reflectivo: ele pensa sobre vários aspectos do seu próprio comportamento, como também sobre o comportamento

de outros agentes e sobre as interações entre os agentes. Mais especificamente um agente raciocina sobre [BRA99]:

- seu estado de informação próprio
- suas suposições
- controle de seus raciocínios próprios
- suas observações
- sua comunicação
- o estado de informação, suposições e processos de raciocínio de outros agentes.

Para a definição de como o agente deve ser projetado e como suas características e funções serão implementadas, Wooldridge propõe em [WOO97] quatro classes de arquiteturas de agentes:

- **Agentes baseados em lógica:** nessa classe a tomada de decisão é baseada em dedução lógica;
- **Agentes reativos:** a tomada de decisão é implementada em alguma forma de mapeamento direto da situação para a ação;
- **Agentes BDI (de crenças, desejos e intenções):** a tomada de decisão depende da manipulação de estrutura de dados representando as crenças, desejos e intenções do agente;
- **Arquiteturas niveladas:** a tomada de decisão é realizada através de vários níveis de software, cada qual raciocina mais ou menos explicitamente sobre o ambiente em diferentes níveis de abstração.

Decidimos adotar a arquitetura BDI ao longo de nosso trabalho por se aproximar dos nossos objetivos de modelar estados mentais. O próximo capítulo aborda essa arquitetura, e maiores detalhes sobre as outras classes de agentes podem ser obtidas em [WEI99].

3.7 Discussões

Neste capítulo verificamos que embora o termo “agente” seja usado freqüentemente, ainda não existe uma definição precisa para ele. Existem na verdade várias definições que normalmente refletem o domínio da aplicação.

Mas, apesar das diferentes definições, observa-se que existem características que são comuns entre elas, isto é, consideradas importantes para o agente como: autonomia, percepção, comunicação e cooperação.

Consideramos que essas características são indispensáveis, mas não são suficientes para representar o comportamento humano de forma eficiente, sendo necessário acrescentar ao agente estados mentais, que serão discutidos no próximo capítulo.

Concluimos também que as ontologias são de fundamental importância no desenvolvimento de agentes, e que seu uso em nosso trabalho contribuirá na representação do conhecimento a ser manipulado.

Com relação à comunicação entre agentes consideramos a linguagem ACL como possível padrão a ser adotado, mas não nos aprofundamos nesse assunto uma vez nosso trabalho não enfocará os aspectos de padrão de comunicação, e sim a composição do agente em si e seu raciocínio sobre o mundo e sobre outros agentes.

Finalmente, ao analisar as classes de arquitetura de agentes, optamos pelo avanço dos estudos sobre a arquitetura BDI por tratar do uso de crenças, desejos e intenções, que correspondem aos estados mentais que, conforme comentamos acima, acreditamos que contribuirão para aproximar o agente de uma representação mais eficiente do comportamento humano.

O próximo capítulos apresenta em detalhes a arquitetura BDI assim como a implementação de estados mentais no agente.

4. Arquitetura BDI (Crenças, Desejos e Intenções)

4.1 Introdução

Um agente BDI é visto como um agente racional que tem certas atitudes mentais de crenças, desejos e intenções, representando respectivamente, os estados internos de informação, motivação e deliberação do agente [RAO95].

Segundo Giraffa [GIR97] esses estados internos se relacionam com o estado do ambiente com o qual o agente interage, e devem corresponder aos estados mentais humanos, mas essa correspondência não precisa ser exata.

Essa arquitetura tem suas raízes nas teorias filosóficas de Bratman [BRT87] que argumenta que o agente BDI deve ter um raciocínio prático que envolve dois importantes processos: decidir *quais* metas queremos ativar, e *como* ativaremos tais metas.

Para que o agente tenha um raciocínio prático é necessário que utilize de forma organizada a representação dos estados mentais que interferem nesse raciocínio. Vejamos a seguir o que são e como se relacionam esses estados mentais.

4.2 Estados Mentais

Para Shoham [SHO93] o termo agente é uma entidade cujo estado é visto como consistindo de componentes mentais tais como crenças, capacidades, escolhas e compromissos. Estes componentes referem-se aos estados mentais internos dos agentes, através dos quais, os agentes se relacionam com o ambiente e com outros agentes com os quais interagem.

Os estados mentais humanos têm vínculo com o mundo em termos do qual estabelecem a sua existência e a sua 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 intencionalidade [GIR97]. Crença, desejo e intenções são os estados intencionais utilizados na arquitetura BDI, mas existem outros como expectativa, por exemplo.

Para compreender o modelo BDI é importante considerar um exemplo simples de raciocínio prático extraído de [WEI99] :

Quando uma pessoa está na universidade, ao nível de graduação, é deparada com uma decisão a tomar – sobre o que fazer com sua vida. O processo de decisão começa tentando entender quais opções disponíveis a pessoa tem. Por exemplo, se tiver um bom desempenho na graduação uma opção é tentar a vida acadêmica (se a pessoa falha para obter uma boa graduação, esta opção já não está disponível para ela). Outra opção é entrar para a indústria. Após gerar um conjunto de alternativas, é preciso escolher entre elas, e comprometer-se com alguma. Essas opções de escolha tornam-se intenções, que então determinam as ações do agente. As intenções levam ao raciocínio prático futuro do agente. Por exemplo, se a pessoa decide ser acadêmica, então ela deve comprometer-se com esse objetivo, e dedicar tempo e esforços para realizá-lo.

4.2.1 Crenças

A definição de crença varia entre autores, mas a idéia essencial é que as crenças estão relacionadas à visão que o agente tem, em um dado momento, do ambiente em que está inserido.

Shoham [SHO93] associa crença a tempo, quando diz que um agente acredita em certas coisas em certos tempos e sobre certos tempos.

Giraffa [GIR99] aborda a idéia de que as crenças expressam as expectativas do agente sobre o estado atual do mundo e sobre a possibilidade de determinada ação atingir um certo efeito.

Correa [COR99] considera crença como conhecimento e um estado mental intencional fundamental para as interações dos agentes.

4.2.2 Desejos

O desejo é um estado mental relacionado aquilo que o agente espera que aconteça em relação ao mundo.

Giraffa [GIR97] coloca uma questão importante relacionada ao desejo que é o fato de ser permitido ao agente ter desejos inconsistentes, que ele não precisa acreditar que possam ser alcançados. Ter desejos não significa que o agente agirá para atingi-los. É um dos estados

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

4.2.3 Intenções

As intenções desempenham um papel crucial no processo de raciocínio prático, apresentando as seguintes propriedades [WEI99]:

- intenções tendem a levar a ação. Se uma pessoa tem determinada intenção então ela deve desejar agir sobre aquela intenção para tentar ativá-la, tomando algum curso de ação que acredite que melhor satisfaça tal intenção. Porém se um curso de ação falha na ativação da intenção, então a pessoa deve ter a expectativa de tentar novamente. Deve ser mencionado também que a escolha de uma intenção restringirá o raciocínio prático futuro.
- As intenções persistem. Se a pessoa tem uma intenção deve persistir até atingi-la, mesmo que falhas ocorram. A persistência é inviável quando a intenção é muito longa, sendo claro que nunca será atingida. Neste caso o mais racional é interromper tal intenção. Similarmente se a razão para se ter uma intenção já é ausente, então é prudente acabar com a intenção.
- As intenções são fortemente relacionadas às crenças sobre o futuro. Se uma pessoa tem determinada intenção ela deve acreditar que conseguirá torná-la realidade.

O principal problema no projeto de agentes de raciocínio prático é o de conseguir um equilíbrio entre os diferentes aspectos das intenções, mencionados acima, sendo visível que o agente deve periodicamente interromper algumas intenções e reconsiderar outras.

4.3 Composição do agente BDI

Wooldridge em [WEI99] resume o processo de raciocínio prático em um agente BDI em um esquema que apresentamos abaixo:

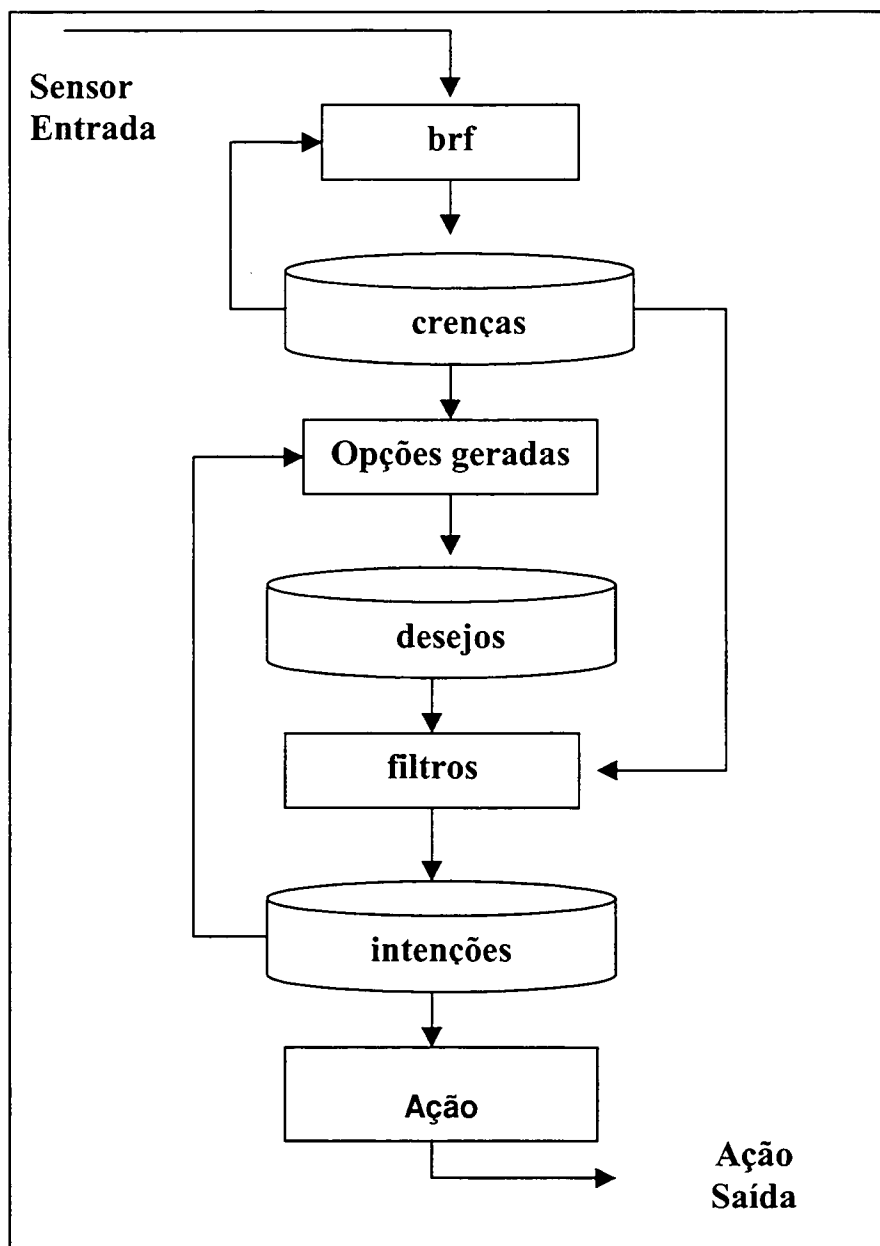


Figura 3 - Diagrama esquemático de uma arquitetura BDI [WEI99].

O diagrama da figura 3 ilustra a existência de sete principais componentes de um agente BDI :

- Um conjunto de crenças correntes, representando a informação que o agente tem sobre seu ambiente;
- Uma função de revisão de crença (brf), que obtém uma entrada perceptiva e as crenças correntes do agente, e com base nisso, determina um novo conjunto de crenças;
- Uma função de geração de opções que determina as opções disponíveis para o agente (seus desejos), com bases nas suas crenças atuais sobre seu ambiente e nas suas intenções atuais;
- Uma função de filtro, que representa o processo de deliberação do agente, e que determina as intenções do agente com base nas suas crenças, desejos e intenções atuais;
- Um conjunto de intenções atuais, representando o enfoque atual do agente – aqueles estados de obrigação que ele tem comprometido para tentar realizá-los.
- Uma função de seleção de ação, que determina uma ação a executar com base nas intenções atuais.

Segundo Rao & Georgeff [RAO95], deve-se adaptar um sistema BDI a aplicação desejada, baseado no comportamento racional necessário para tal aplicação, caracterizando diferentes interações entre as crenças, desejos e intenções, o que determina que não existe uma arquitetura BDI rígida ou completamente definida.

A arquitetura de agentes proposta pelo grupo de pesquisa DESIRE⁴, corresponde a uma extensão do modelo BDI, permitindo modelar explicitamente, além do conhecimento e

⁴ DESIRE – (DESign and Specification of Interactive REasoning components). Um framework de modelagem que permite projetar, especificar e operacionalizar tarefas [BRA95].

raciocínio do agente sobre ele mesmo, o raciocínio sobre o mundo e outros agentes. A seguir apresentaremos as principais características dessa arquitetura.

4.4 DESIRE [BRA97]

A metodologia DESIRE parte do princípio de que para modelar um agente capaz de raciocinar sobre suas próprias tarefas, processos e planos, sobre seu conhecimento a respeito de outros agentes bem como sua comunicação com eles, sobre seu conhecimento do mundo e sobre sua interação com ele, é necessário idealizar uma arquitetura onde tais tipos de raciocínio sejam alocados de forma transparente em componentes específicos de um agente [BRA97a].

O DESIRE propõe um modelo genérico para o comportamento dinâmico interno de agentes BDI, que representa explicitamente crenças e atitudes motivadoras como desejos, metas, intenções, compromissos, planos e seus relacionamentos.

O modelo proposto incorpora crenças, desejos e intenções, nos quais intenções com respeito a metas são distinguidas de intenções com respeito a planos. O resultado é um agente BDI mais específico em que as dependências entre crenças, desejos e intenções são feitas explicitamente [BRA98b].

A principal ênfase está nas relações dinâmicas e estáticas entre as atitudes mentais, sendo os processos pertinentes ao agente, projetados em componentes internos, que interagem baseados no conhecimento próprio que possuem.

Os componentes podem ser primitivos, que raciocinam usando uma base de conhecimento, mas também podem ser subsistemas capazes de executar tarefas usando diversos métodos como teoria da decisão, redes neurais e algoritmos genéticos [BRA99a].

O DESIRE modela várias atitudes para os agentes que são assumidos para ter intenções e compromissos ambos com respeito a metas e planos.

Brazier [BRA96b] propõe uma classificação de atitudes mentais, apresentada na tabela 2.

Atitudes Mentais		
Atitudes Informativas	Conhecimento	
	Crenças	
Atitudes Motivadoras	Desejos	
	Intenções	Metas Pretendidas
		Planos Pretendidos
	Compromissos	Metas Comprometidas
		Planos Comprometidos

Tabela 2: Classificação das atitudes mentais do agente [BRA96b] .

Nesta classificação a atitude motivadora mais fraca é o desejo, que pode ser ordenado de acordo com preferências, e corresponde a única atitude motivadora sujeita a inconsistência. Um número limitado de metas pretendidas são escolhidas por um agente, com base em seus desejos (e crenças). Com respeito as intenções, suas condições são elaboradas com base nos conceitos de Bratman [BRT87], Cohen e Levesque [COH90].

Conforme as intenções, um agente se compromete para ativar metas e executar planos, podendo também se comprometer com outros agentes. Tais compromissos sociais são explicitamente modelados e, como proposto em [CAS95], são mais fortes que as intenções, porque envolvem os aspectos de obrigação e interesse nos compromissos de outros agentes.

Após comprometer-se com uma meta e um plano associado, um agente inicia a realização do plano, sendo necessário o conhecimento das estratégias e dependências para determinar em quais situações um agente abandona uma intenção ou compromisso, e como. O tipo de comportamento que o agente manifesta depende do ambiente e das características comportamentais iminentes, incluindo suas estratégias pretendidas e comprometidas.

As estratégias de intenção foram introduzidas por Rao e Georgeff [RAO95], e inspiraram a definição de estratégias de compromisso social apresentada por Dunin-Keplicz e

Verbrugge [DUN96]. Estas estratégias de compromisso incluem aspectos de comunicação e coordenação, uma vez que um agente pode abandonar ou manter um compromisso em função da crença que possui sobre a possibilidade de atingi-lo e de acordo com suas metas, comunicando então aos agentes com os quais interage, a respeito de sua decisão.

Para descrever a arquitetura do agente BDI proposta pelo DESIRE, são utilizados quatro tipos de conhecimento: composição de tarefa, troca de informação, sequenciação de tarefas e estruturas de conhecimento [BRA96]. Vejamos a seguir um detalhamento sobre esses conhecimentos.

▪ **Composição de Tarefas**

Considera-se que o agente deve ser capaz de seis tarefas [BRA97]:

1. controlar seus próprios processos,
2. executar suas tarefas específicas próprias,
3. gerenciar sua interação com o mundo (observação, execução de ações),
4. gerenciar sua comunicação com outros agentes,
5. manter informação sobre o mundo, e
6. manter informação sobre outros agentes.

Através dessas tarefas o agente tem especificado que deverá exercer um controle sobre si mesmo, mas também sobre os outros agentes e sobre o mundo, o que o tornará muito mais poderoso.

Para modelar a composição das tarefas é necessário conhecimento sobre a hierarquia das tarefas, sobre as informações que a tarefa requer como entrada e que gera como um resultado de sua execução, e também conhecimento sobre as relações entre as tarefas [BRA97b].

A idéia central aqui é criar um componente para cada tarefa necessária , sendo que de acordo com a complexidade do agente, algumas tarefas poderão ou não ser descartadas.

▪ **Troca de Informação entre as tarefas**

As informações trocadas entre as tarefas do agente são especificadas através de links de informação entre os componentes.

A figura 4 mostra uma representação gráfica da arquitetura de um agente BDI onde as tarefas são tratadas como componentes dentro do agente, que interagem através dos links de informação [BRA98b].

Os links de informação estabelecidos na figura 4, bem como seus componentes de origem e destino são apresentados na tabela 3 [BRA98b].

A utilização dos links dentro do agente ocorre de acordo com a tarefas que ele realiza, sendo que nem sempre todos os links serão estabelecidos.

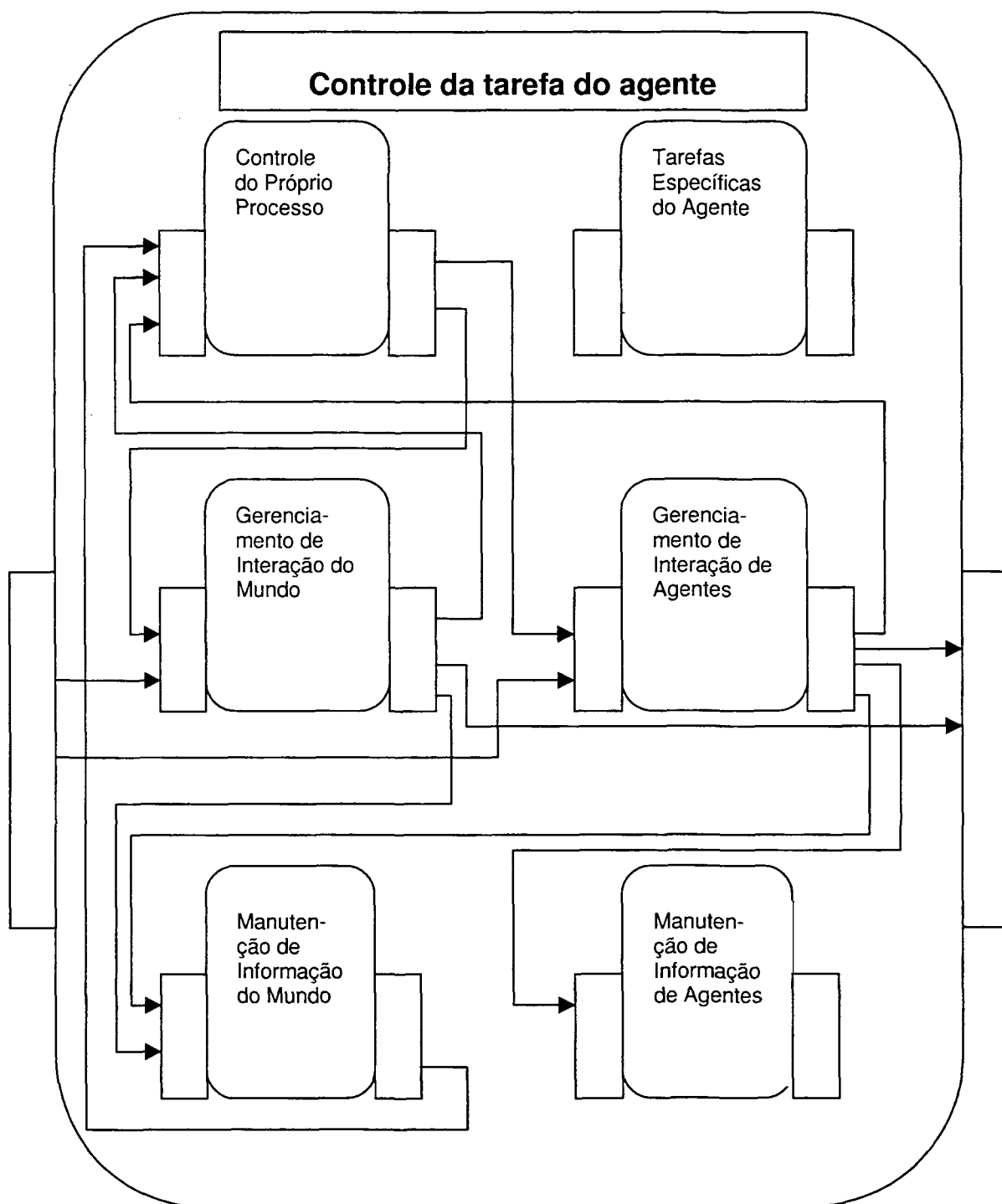


Figura 4 - Composição de alto nível e links de informação de um agente.

Nome do Link	Componente Origem	Componente Destino
Importa informação do mundo	Agente(interface de entrada)	Gerenciamento de Interação do Mundo
Exporta informação do mundo	Gerenciamento de Interação do Mundo	Agente (interface de saída)
Transfere informação comunicada do mundo	Gerenciamento de Interação do Agente	Manutenção de Informação do Mundo
Transfere informação de manutenção do agente	Manutenção de Informação do Agente	Gerenciamento de Interação do Mundo
Provê informação do estado do mundo	Manutenção de Informação do Mundo	Controle do Próprio Processo
Importa informação do agente	Agente (interface de entrada)	Gerenciamento de Interação do Agente
Exporta planejamento comum	Gerenciamento de Interação do Agente	Agente (interface de saída)
Provê informação do agente	Manutenção de Informação do Agente	Controle do Próprio Processo
Transfere ações e observações comprometidas	Controle do Próprio Processo	Gerenciamento de Interação do Mundo
Transfere compromissos do agente	Controle do Próprio Processo	Manutenção de Informação do Agente
Transfere planejamento comum	Controle do Próprio Processo	Gerenciamento de Interação do Agente
Transfere crenças do mundo	Controle do Próprio Processo	Manutenção de Informação do Mundo
Transfere informações de comunicação	Tarefas Específicas do Agente	Gerenciamento de Interação do Agente
Transfere informação de ação e observação	Tarefas Específicas do Agente	Gerenciamento de Interação do Mundo
Transfere informação comunicada	Gerenciamento de Interação do Agente	Tarefas Específicas do Agente
Transfere informação observada	Gerenciamento de Interação do Mundo	Tarefas Específicas do Agente
Transfere informação do agente comunicada	Gerenciamento de Interação do Agente	Manutenção de Informação do Agente
Transfere informação do mundo observada	Gerenciamento de Interação do Mundo	Manutenção de Informação do Mundo
Transfere informação do agente observada	Gerenciamento de Interação do Mundo	Manutenção de Informação do Agente
Transfere informação do agente	Manutenção de Informação do Agente	Gerenciamento de Interação do Agente
Transfere informação de manutenção do mundo	Manutenção de Informação do Mundo	Gerenciamento de Interação do Agente
Transfere informação do mundo	Manutenção de Informação do Mundo	Gerenciamento de Interação do Mundo

Tabela 3 – Links para troca de informação entre as tarefas do agente.

- **Sequenciação de tarefas**

A sequenciação de tarefas é modelada explicitamente dentro dos componentes como um conhecimento de controle da tarefa, que inclui não somente conhecimento de que tarefas devem ser executadas, quando e como, mas também conhecimento das metas associadas com a ativação da tarefa [BRA96a].

O controle de tarefa é baseado puramente na informação sobre o sucesso e/ou falha do raciocínio do componente.

- **Estruturas de conhecimento**

Durante a aquisição de conhecimento deve-se idealizar uma estrutura apropriada para o conhecimento do domínio, determinando o significado dos conceitos usados para tal domínio e as relações entre conceitos e grupos de conceitos. Os conceitos são necessários para a identificação dos diferentes objetos em um domínio (ontologia orientada a domínio), mas também para expressar métodos e estratégias envolvidas na execução de uma tarefa (ontologia orientada a tarefa) [BRA98a].

4.5 Construção de um modelo de agente racional com atitudes motivadoras

Segundo Jonker [JON98] os agentes são considerados racionais no sentido em que precisam ser capazes de gerar metas e agir racionalmente para ativá-las, planejando, replanejando e executando o plano.

Além disso, é interessante que a atividade do agente seja descrita usando as noções mentalísticas usualmente aplicadas a humanos. Isto não implica que um sistema de computador seja visto atualmente como “tendo” crenças e intenções, mas que estas noções são consideradas úteis na modelagem e especificação do comportamento requerido para construir Sistemas Multiagentes efetivos.

Na construção do modelo do agente racional com atitudes motivadoras existem 4 suposições a serem consideradas [BRA98b]:

1. As atitudes motivadoras como crenças, desejos, intenções e compromissos são definidas como declarações reflectivas sobre o agente em si e sobre o agente em relação a outros agentes e ao mundo externo. As relações lógicas ou funcionais entre as atitudes motivadoras, e entre atitudes motivadoras e atitudes informativas são expressadas como um meta-conhecimento, que pode ser usado para executar meta-raciocínio resultando em futuras conclusões sobre atitudes motivadoras. Por exemplo, em uma instanciação simples do modelo do agente, crenças podem ser inferidas do meta-conhecimento de que qualquer fato observado é um fato acreditado e que qualquer fato comunicado por um agente confiável é um fato acreditado.
2. Uma informação é classificada de acordo com sua fonte: informação interna, observação, comunicação, dedução, suposição. Ambas atitudes informativas como crenças e atitudes motivadoras como desejos, dependem dessas fontes de informação.
3. A dinâmica dos processos é explicitamente modelada, ou seja, pode ser especificado explicitamente se determinados processos devem ser executados em paralelo ou sequencialmente.
4. As intenções e compromissos são definidos com respeito a metas e planos. Assim um agente determina quais metas ele pretende atingir, e compromete-se com um subconjunto selecionado dessas metas. Da mesma forma um agente determina que planos ele pretende executar e compromete-se com um subconjunto selecionado desses planos.

Tanto as atitudes informativas, como as atitudes motivadoras fazem parte da tarefa de controle do próprio processo do agente. Assim esse componente é refinado obedecendo a hierarquia de tarefas apresentada na figura 5.

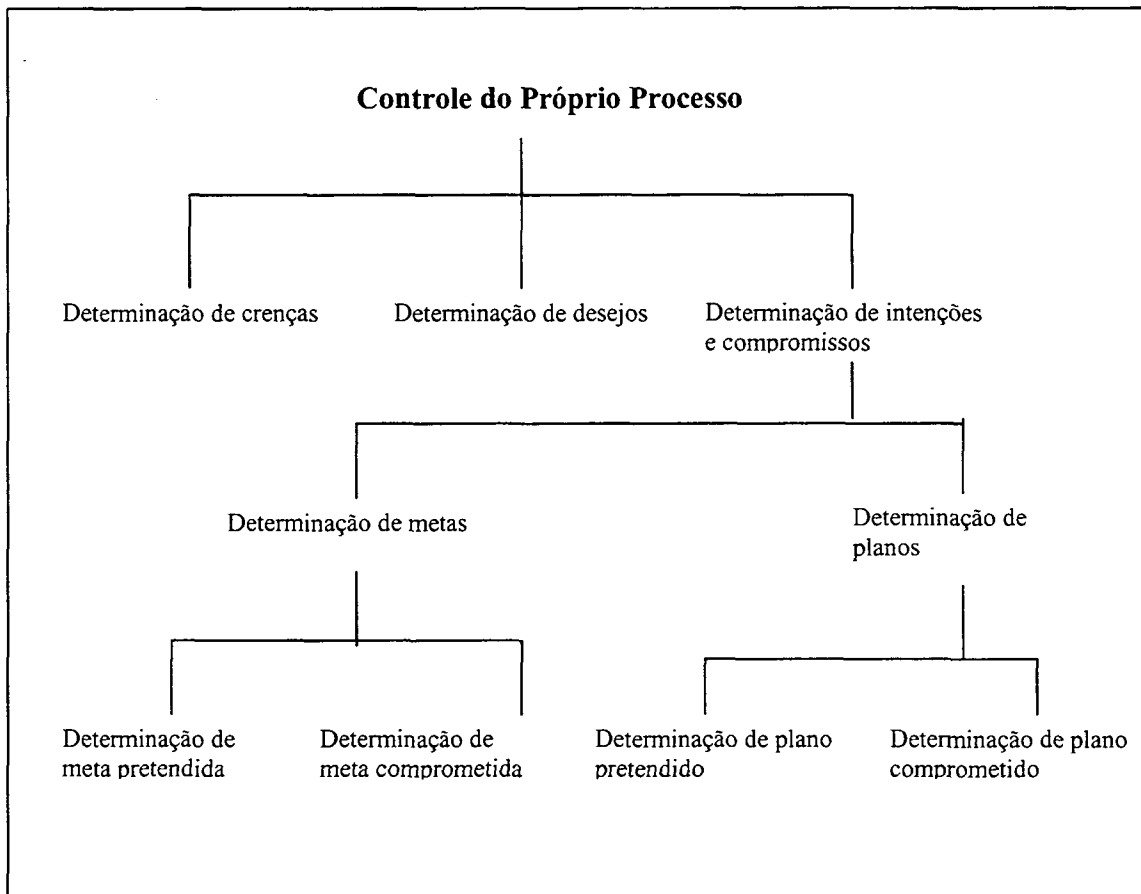


Figura 5 - Hierarquia de tarefas do componente Controle do Próprio Processo [BRA98b].

Através do diagrama podemos verificar que o Componente Determinação de Crença executa raciocínio sobre crenças relevantes em uma dada situação. No Componente Determinação de Desejos um agente determina quais desejos ele tem, relacionados a suas crenças. Metas e planos pretendidos e comprometidos derivam do Componente Determinação de Intenções e Compromissos.

Neste modelo, as crenças e desejos se influenciam reciprocamente, assim como influenciam as intenções e compromissos.

4.6 Implementação de agentes

No trabalho de Mathias [MAT00] podemos verificar que as atuais linguagens de programação orientada a agentes não apresentam uma forma de raciocínio explícito do agente em relação ao mundo e a outros agentes.

Como vimos na capítulo 3 agentes raciocinam em função do conhecimento que possuem. Assim, entendemos que a linguagem para implementar agentes, deve ter expressividade suficiente para que possamos escrever o que queremos dizer, tornando clara a representação do conhecimento e dos estados mentais do agente.

Realizamos uma busca exaustiva de uma linguagem que pudesse atender esses objetivos. Verificamos que muitas linguagens estão mais concentradas no aspectos de comunicação entre agentes do que na representação do agente em si. Podemos citar como exemplo as linguagens Magenta [GEN97], Prologic [PRO99] e Infomaster [OLI97], sendo que ambas utilizam a linguagem KIF como formato padrão para troca de conhecimento. Tentamos utilizar essas linguagens mas não encontramos material suficiente liberado para fins de pesquisa.

Identificamos então o sistema de representação de conhecimento LOOM que corresponde a uma linguagem de programação de alto nível usando lógica descritiva⁵, sendo também um ambiente para a construção de sistemas baseados em conhecimentos [LOOM91].

Hayes em [HAYE99], cita LOOM como uma linguagem para modelagem de agentes. Embora em nossas consultas a bibliografia específica da linguagem não tenhamos encontrado nenhuma referência a isso, acreditamos que pelo seu forte poder de expressividade e inferência, ela possa realmente se apresentar como uma boa opção para implementação de agentes inteligentes.

⁵ Lógica Descritiva - é a representação do conhecimento e do raciocínio a respeito de indivíduos que são agrupados em classes e interligados por relações binárias [SPI00].

4.7 Discussões

A arquitetura BDI utiliza os estados mentais: desejos, crenças e intenções, através dos quais pretende-se representar o comportamento humano em um agente de software.

Com relação a cada um desses estados mentais observa-se que apesar das intenções corresponderem ao estado mental fundamental do agente, por levá-lo a agir, as crenças são a base para o início do raciocínio do agente, devendo por isso receber especial atenção.

Quanto aos desejos, eles correspondem ao estado mental mais difícil de se definir e até mesmo aplicar, justamente por ser inconsistente e por não necessitar obrigatoriamente ser alcançado.

Ao estudarmos a composição do agente BDI vemos que a forma adequada de organizar o raciocínio do agente é: com base em crenças o agente tem desejos que se transformam em intenções. Essas intenções levam a ação do agente.

Analisando a arquitetura BDI proposta por Wooldridge observa-se que trata somente do raciocínio de um componente, o agente em si, enquanto que a proposta do DESIRE acrescenta ao agente componentes que raciocinarão sobre outros agentes e sobre o mundo.

Concluimos que a arquitetura proposta pelo DESIRE torna o agente mais completo em sua representação do comportamento humano, porque sabemos que o raciocínio sobre o ambiente é também uma característica humana, que influencia diretamente nos processos de tomadas de decisão.

Uma característica interessante do DESIRE é a alocação das tarefas do agente em componentes específicos que interagem. Isso torna o projeto do agente mais claro, além de facilitar sua manutenção.

Outro ponto importante é que os estados mentais do agente estão concentrados em um só componente: o Controle do Próprio Processo, sendo que os demais componentes agem em função de planos estabelecidos por esse componente.

Através dessas conclusões decidimos utilizar a arquitetura BDI proposta pelo DESIRE como base para o projeto do agente que desenvolveremos na seqüência desse trabalho, e que é descrito no próximo capítulo.

5. AGENTE-M

5.1 Introdução

O grupo de pesquisa da UFPR voltado para a área de Agentes desenvolveu o SISMAT, um Sistema de Matrícula Inteligente composto inclusive por um agente matriculador [MAT00]. Sua implementação foi realizada através da linguagem LALO (Langage d'Agents Logiciel Objet) que é abordada em [MAT00a].

Esse trabalho utilizou o paradigma de programação orientada a agentes AOP (Agent-Oriented Programming) [SHO93], onde um agente é determinado por suas crenças, capacidades, escolhas e compromissos. No entanto, o agente matriculador desenvolvido no SISMAT não tem capacidade de planejamento, e não possui um raciocínio definido sobre o mundo e sobre outros agentes.

Acreditamos que o fato do ambiente de desenvolvimento utilizado possuir uma estrutura pré-definida para o agente, limita sua composição e com isso aspectos importantes de seu comportamento ficam ocultos.

Buscamos então desenvolver um novo agente matriculador, denominado AGENTE-M, onde o principal enfoque foi o agente em si, tornando explícitos seus aspectos de comportamento e raciocínio sobre o mundo, e sobre outros agentes. Esperamos com isso, facilitar sua compreensão e contribuir para o avanço dos estudos relacionados a aproximação do agente do comportamento humano.

O protótipo do AGENTE-M segue os critérios de matrícula que regem a Universidade Estadual de Ponta Grossa - UEPG [UEPG99].

Seu projeto está fundamentado na arquitetura BDI proposta pelo DESIRE, que foi adaptada ao tipo de aplicação em questão. Implementamos esse agente através da linguagem de representação de conhecimento LOOM.

A função do AGENTE-M é raciocinar sobre a possibilidade de matricular um aluno, e efetivar tal matrícula, baseado no conhecimento que possui, atuando de acordo com seus estados mentais.

O agente está especificado em componentes internos que interagem quando necessário, de acordo com a situação em que se encontra.

Dessa forma, podemos através do AGENTE-M avaliar o desempenho da arquitetura BDI adotada, identificar as facilidades e dificuldades de implementação, as vantagens e

desvantagens de sua utilização, e as diferenças em relação ao agente matriculador do SISMAT.

5.2 Visão geral do estudo de caso

A UEPG adota a seguinte definição para o processo de matrícula [UEPG99]:

“Uma matrícula constitui-se da formulação periódica de um plano de estudo a ser cumprido pelo acadêmico, nos períodos letivos em que estiver realizando o curso, conforme o regime interno da instituição e os calendários estabelecidos. Ao ingressar na UEPG, o acadêmico recebe um código numérico que será a sua identificação. Este código é chamado Registro Acadêmico (RA) ”.

O aluno pode apresentar várias situações em relação ao curso:

- Calouro : está iniciando o curso
- Aprovado sem dependência: aprovado em todas as disciplinas cursadas até o momento.
- Aprovado com dependência: reprovado em até duas disciplinas, mas pode avançar de série
- Retido: retido na série atual devendo cursar somente as disciplinas em que reprovou
- Reprovado: retido na série, devendo cursar toda a série novamente.
- Trancado: está com o curso trancado
- Jubilado: perdeu os direitos de se formar
- Transferido: aluno transferido de outra universidade
- Formado: aluno que já concluiu toda a grade curricular

O processo de matrícula envolve conhecimento sobre o aluno, o curso, a grade curricular e as regras da instituição. Assim, o agente matriculador deve ter esses conhecimentos, saber manipulá-los e simular as capacidades que um agente matriculador humano apresenta como a crença das disciplinas ofertadas, o desejo de matricular e a intenção de validar o registro acadêmico, entre outras.

Para realizar a matrícula é proposto um ambiente, como mostrado na figura 6, composto por três elementos que são:

- Uma base de dados com informações sobre os alunos, as disciplinas e o histórico escolar.
- Um agente usuário que pretende matricular-se, e informa seu RA ao agente matriculador.
- Um agente matriculador, que deseja matricular o aluno cujo RA foi informado. Para isso, ele analisa a situação atual do aluno e compromete-se a, se possível, efetivar a matrícula.

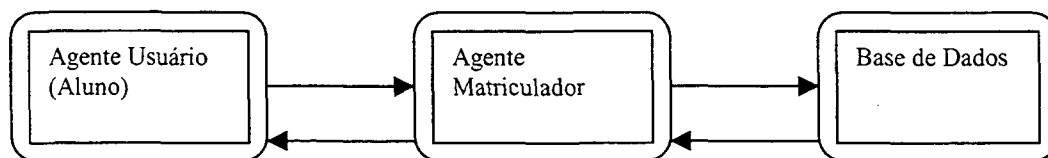


Figura 6 – Ambiente onde está inserido o agente matriculador.

O agente usuário informa o RA do aluno ao matriculador que, raciocinando sobre a base de dados e seu próprio conhecimento, valida o RA e efetiva a matrícula mostrando os resultados ao agente usuário.

Nosso objetivo é projetar o agente matriculador que denominamos AGENTE-M, através de uma arquitetura BDI, representando explicitamente o seu relacionamento com o mundo e com o agente usuário.

A seção a seguir apresenta o projeto do AGENTE-M.

5.3 Projeto do AGENTE-M

Como vimos no capítulo 3 a principal característica de um agente inteligente é sua capacidade de controle próprio, e de gerenciamento de interação com outros agentes e com o mundo. Assim, estruturamos o AGENTE-M em 4 componentes internos: Controle do Próprio Processo, Gerenciamento do Mundo, Gerenciamento de Agentes e Tarefas Específicas do Agente.

Cada componente possui em sua estrutura:

- Átomos de entrada: correspondem às informações que transferidas para o componentes.

- Átomos de saída: são as informações exportadas pelo componente, podendo ser transferidas para o agente usuário, para o mundo ou para outros componentes dentro do próprio agente.
- Base de conhecimento: contém o meta-conhecimento que define o raciocínio do componente, determinando qual o próximo componente que será ativado por ele (próximo_estado_componente) e qual informação (próximo_estado_link) a ser transferida para ele, que normalmente corresponde a tarefa que deverá executar.

Vamos agora discutir a estrutura e função de cada componente do AGENTE-M.

5.4 Componente Controle do Próprio Processo:

Este é o principal componente dentro do agente, uma vez que define suas características e determina o enfoque para o seu comportamento, com base nas especificações dos seus estados mentais.

Considerando que o AGENTE-M tem uma arquitetura BDI foram estabelecidos os seguintes subcomponentes: determinação de crenças, de desejos e de intenções. Assim cada estado mental é definido separadamente e o relacionamento entre eles ocorre através da interação desses subcomponentes.

Dentro do Componente Determinação de Intenções e Compromissos definimos subcomponentes responsáveis pelos planos que o agente deve estabelecer para realizar seus desejos. O agente pode ter planos que pretende realizar e planos que se compromete a realizar. Dessa forma os subcomponentes criados foram: Determinação de Planos Pretendidos e Determinação de Planos Comprometidos.

O diagrama a seguir apresenta o refinamento do componente Controle do Próprio Processo do AGENTE-M baseado em [BRA96b]:

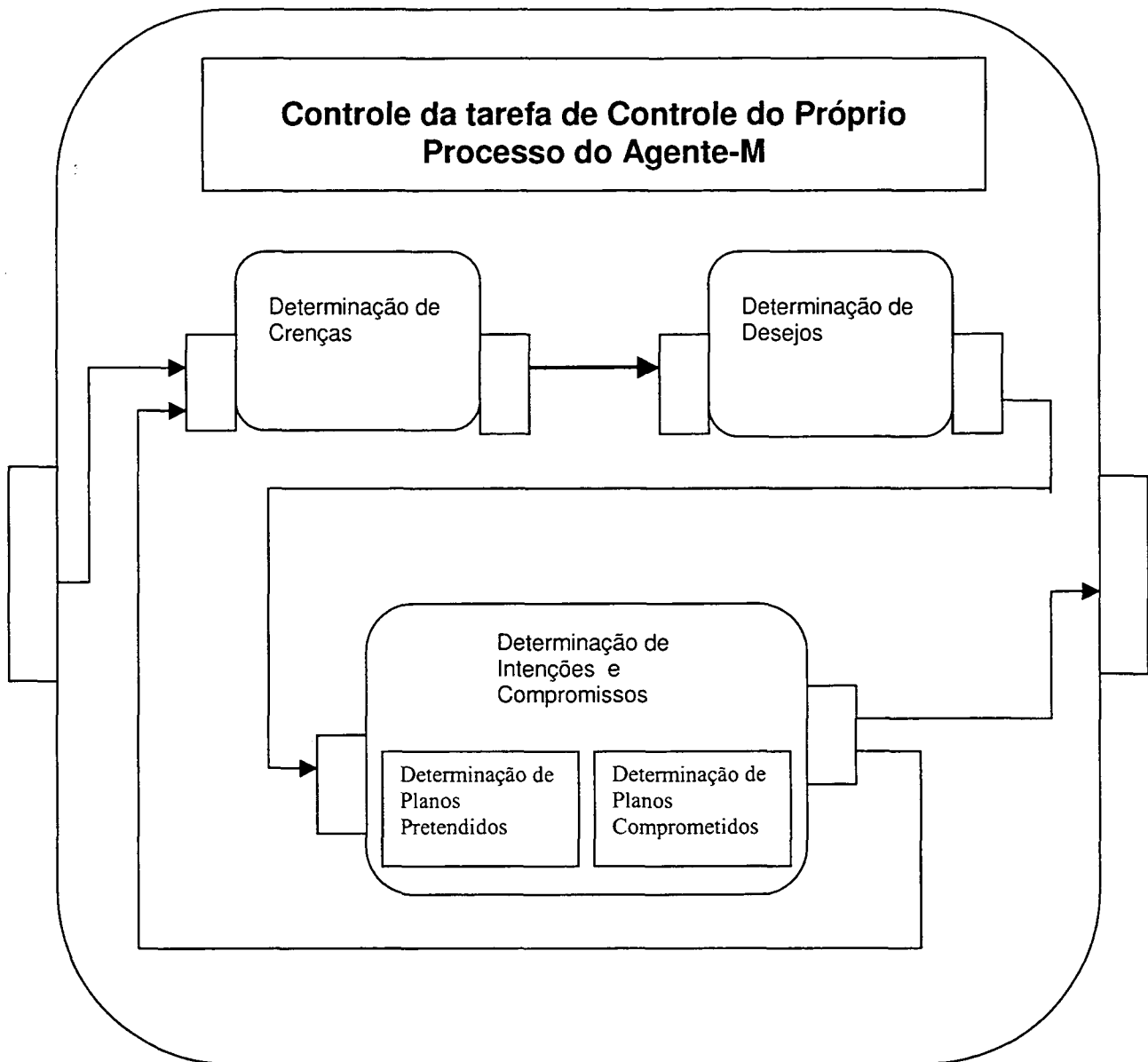


Figura 7 - Refinamento do Componente Controle do Próprio Processo dentro do AGENTE-M.

5.4.1 Determinação das Crenças do AGENTE-M

A tarefa de determinação das crenças requer um meta-raciocínio explícito para gerar crenças. O conhecimento do AGENTE-M para esse propósito está baseado nas seguintes fontes: crenças internas, observações e comunicações.

- **crenças internas:** são as crenças que o AGENTE-M possui inerentemente, são os fatos iniciais.

Na figura 8 apresentamos o meta-conhecimento que estabelece a crença de que existe dados do aluno na base de dados, derivando da crença interna que indica tal fato.

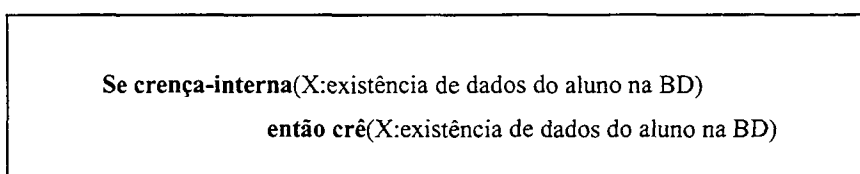


Figura 8 – Crença interna do AGENTE-M.

- **crenças baseadas em observações:** Para que o agente matriculador acredite na situação do acadêmico, ele observará este fato no mundo e acreditará no resultado da observação realizada. Essa crença será transferida para o Componente Determinação de Intenções e Compromissos, servindo de base para a decisão sobre qual será a próxima ação do agente.

O meta-conhecimento especificado na figura 9 deriva uma crença baseada em observação, indicando que o agente crê no fato observado na base de dados (no mundo) e decide que a próxima ação do agente será a transferência da informação de crença para o componente determinação de intenções e compromissos.

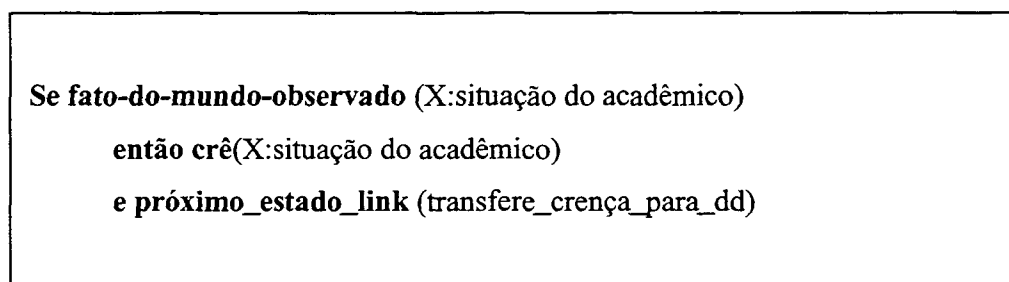


Figura 9 – Crença do AGENTE-M baseada em observação.

- **crenças baseadas em comunicação:** são as crenças que o AGENTE-M possui através da comunicação com o agente usuário. Considerando que o agente que comunica o fato é confiável, isso pode levar o agente a derivar uma nova crença.

O meta-conhecimento da figura 10 estabelece que o fato comunicado pelo agente usuário (RA do aluno) leva o AGENTE-M a acreditar que existe um aluno que deseja ser matriculado.

<p>Se fato_comunicado_por (X:RA do aluno, A: usuário) então crê (X:aluno_deseja_matricula)</p>
--

Figura 10 – Crença do AGENTE-M baseada no fato comunicado pelo agente usuário.

Foram estabelecidas várias crenças para o AGENTE-M utilizando essas diferentes fontes de informação. Todas as crenças estão definidas dentro da base de conhecimento do Componente Controle do Próprio Processo.

Os átomos de entrada são as informações transferidas comunicadas pelo agente usuário (**Info_de_agente_transferida**), e observadas no mundo (**Indo_do_mundo_transferida**).

Os átomos gerados de saída correspondem a transferência de crenças para o Componente Determinação de Desejos (**transfere_crença_para_dd**). A figura 11 apresenta a estrutura interna do Componente Determinação de Crenças destacando duas crenças do AGENTE-M.

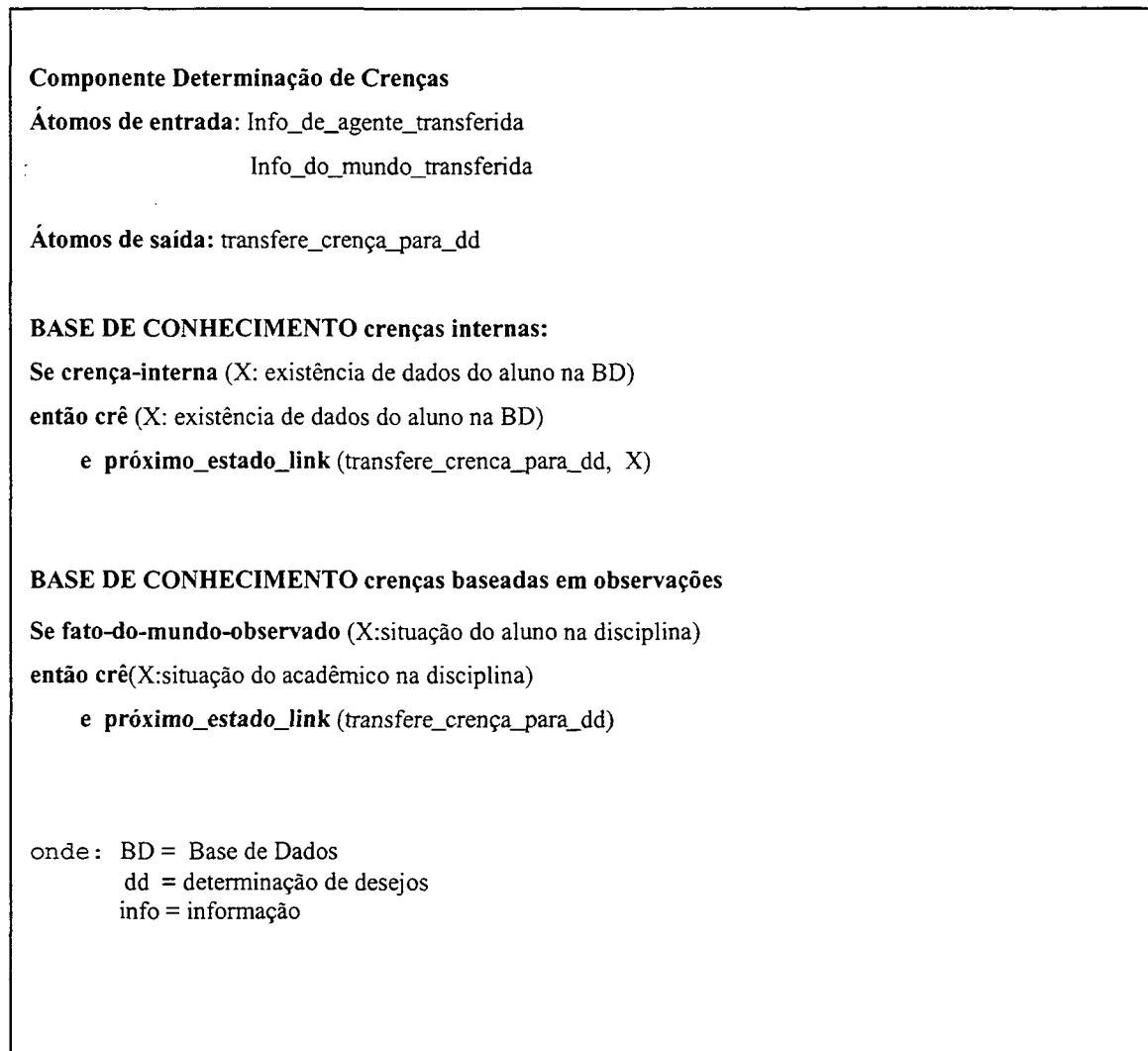


Figura 11 – Crenças do AGENTE-M.

Observamos na figura 11 duas crenças: a primeira é interna e diz respeito ao fato do agente acreditar que existem dados cadastrais do aluno na base de dados que corresponde ao seu mundo.

A segunda crença deriva do resultado da observação no mundo da situação do aluno em determinada disciplina. Alocamos essa crença dentro da base de conhecimento específica denominada “crenças baseadas em observações” porque para cada fonte de informação criamos uma base de conhecimento própria.

5.4.2 Determinação dos Desejos do AGENTE-M

Os desejos do AGENTE-M referem-se aos estados de acontecimento desejados no mundo e as ações que se deseja que sejam em executadas. Os desejos do agente são influenciados pelas suas crenças e pela avaliação do estado atual. Assim, se determinada ação foi realizada com o sucesso isso desperta no agente o desejo de atingir um novo objetivo. Esse desejo é então transferido para o Componente Determinação de Intenções e Compromissos que traçara o plano para realizá-lo.

O meta-conhecimento especificado na figura 12 expressa que se a avaliação da crença de que o RA é válido foi bem sucedida, ou seja a crença é verdadeira, o agente passa a desejar mostrar dados cadastrais para o agente usuário, sendo então estabelecido que o próximo link de informação será transferir a informação desse desejo para o componente determinação de intenções que estabelecerá o planos para atingir essa meta.

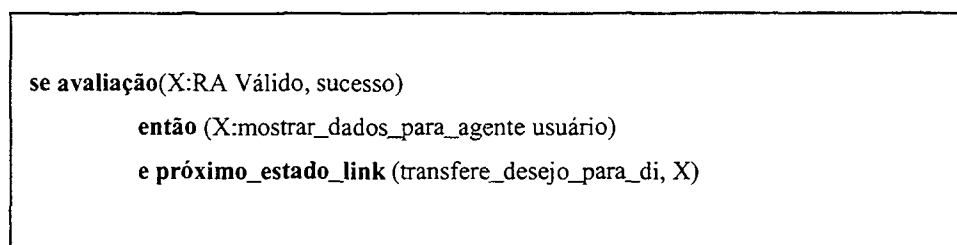


Figura 12 - Determinação de desejo do AGENTE-M.

Os átomos de entrada para esse componente são as informações de crenças (Info_de_crenças_transferidas) transferidas pelo componente Determinação de Crenças.

Quanto aos átomos de saída destacamos que, ao partirmos da idéia de que os desejos levam a intenção de realizá-los, sempre teremos como saída a transferência de desejos para o Componente Determinação de Intenções e Compromissos (transfere_desejo_para_di).

A figura 13 apresenta a estrutura interna do Componente Determinação de Desejos e a especificação do desejo do AGENTE-M de gerar matrícula, caso a avaliação da situação regular do aluno seja bem sucedida. A informação a respeito desse desejo será então transferida para o Componente Determinação de Intenções e Compromissos.

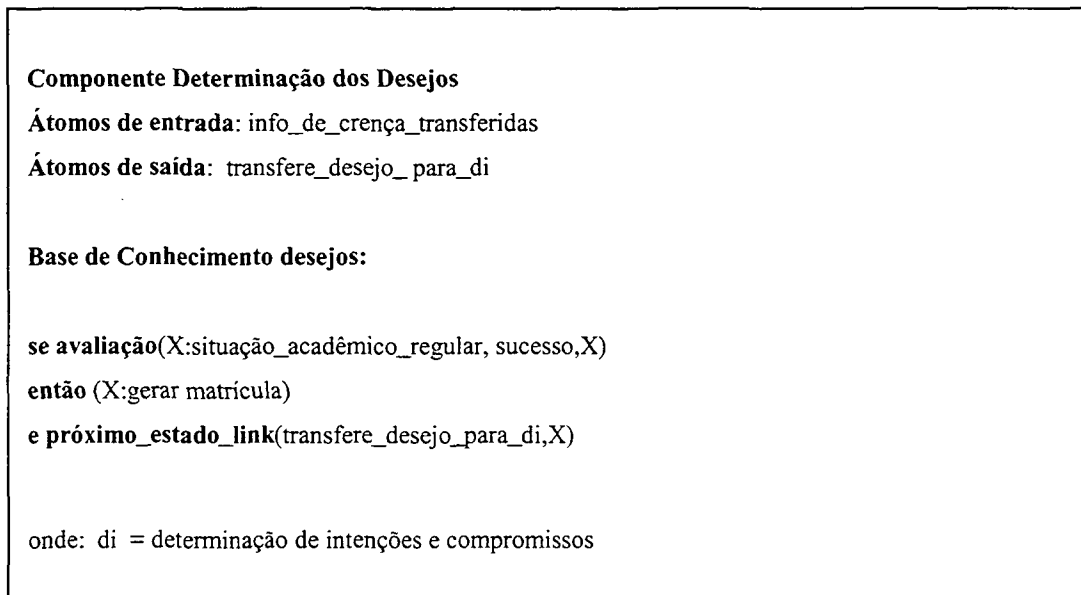


Figura 13 - Determinação de desejos do AGENTE-M.

5.4.3 Determinação das Intenções e Compromissos do AGENTE-M

As intenções e compromissos do AGENTE-M existem para definir uma estratégia para a realização de um determinado desejo do agente, e estão baseadas em planos e metas. Dessa forma, quando o agente tem a intenção de atingir uma meta estabelece um plano, em função de uma crença ou meta já atingida.

Na ativação do plano é definido qual o próximo conjunto alvo, que pode ser um subcomponente ou outro componente, e também que link de informações será ativado.

O meta-conhecimento mostrado na figura 14 estabelece uma intenção baseada em desejo. Se o agente deseja mostrar dados cadastrais sua meta será mostrar tais dados. Para tanto estabelece um plano pretendido que requer a ativação do Componente Gerenciamento do Mundo para localizar tais dados e, uma vez encontrados, assume o compromisso de mostrá-los, sendo necessário então ativar o componente Gerenciamento de Agentes.

```

se desejado(X: mostrar dados cadastrais)
  então meta_pretendida(X:mostrar dados cadastrais)
    e plano pretendido(X: localizar dados cadastrais)
    e próximo_estado_componente (gerenciamento_do_mundo, ativar)
    e próximo_estado_link(transfere_plano_pretendido,X)
    e plano comprometido(X: mostrar dados cadastrais)
    e próximo_estado_componente (gerenciamento_de agentes, ativar)
    e próximo_estado_link(transfere_plano_comprometido,X)

```

Figura 14 - Especificação de Intenção baseada em Desejo.

Na figura 15 temos a expressão que determina um compromisso baseado em uma meta atingida. Esse meta-conhecimento indica que se a meta que havia sido estabelecida, de gerar a matrícula, foi atingida, então o agente se compromete a mostrar o resultado da composição para o agente usuário. Assim, o próximo componente a ser ativado é o Gerenciamento de Agentes para que mostre as informações da matrícula gerada.

```

se meta_atingida (X:gerar matrícula)
  então meta_comprometida (X:mostrar_composição)
    e próximo_estado_componente (gerenciamento_de_agentes, X)
    e próximo_estado_link (mostrar_matricula,X)

```

Figura 15 – Especificação de Intenção baseada em meta atingida.

5.5 Componente Gerenciamento do Mundo

No Componente Gerenciamento do Mundo modelamos a interação do AGENTE-M com o mundo (com o mundo Base de Dados), realizando observações e atualizando o estado do mundo em função dos fatos informados. A base de dados que se encontra no mundo e que interessa ao AGENTE-M é composta por: Cadastro de Alunos, Histórico Escolar e Grade Curricular.

Esse componente define que informação será exportada para o mundo, bem como qual observação deverá ser feita, interagindo sempre com a base de dados. Assim seus átomos de entrada são as informações observadas do mundo (Info_observada_do_mundo) e seus átomos de saída correspondem às informações geradas para o mundo (Info_gerada_para_mundo).

O meta-conhecimento especificado na figura 16 estabelece que se o fato informado ao componente indicar que deve localizar dados cadastrais então ele deverá importar do mundo tais dados , procurando-os na base de dados “Cadastro_de_Alunos”.

A figura 17 mostra que quando o Componente Gerenciamento do Mundo é ativado recebendo a informação de que deve atualizar base então deverá alterar o estado do mundo, ajustando a base de dados “Histórico_Escolar” à nova situação do aluno.

Se fato-informado (X:localizar_dados_cadastrais)
então importar_mundo(X:dados_cadastrais, BD:Cadastro_de_Alunos)
e próximo_estado_link (importa_informação_do_mundo, X)

Figura 16 - Especificação da interação do AGENTE-M com o mundo.

Componente Manutenção de Informação do Mundo

Átomos de entrada: info_observada_do_mundo

Átomos de saída: info_gerada_para_mundo

Se fato-informado (X:atualizar_base)

então alterar_mundo(X: série, disciplinas, turma, ano de matrícula, BD:Histórico_Escolar)

e próximo_estado_link (exporta_informação_para_o_mundo, X)

onde : BD = base de dados a ser observada/alterada

Figura 17 – Meta-conhecimento do Componente Gerenciamento do Mundo.

5.6 Componente Gerenciamento de Agentes

O Componente Gerenciamento de Agentes mantém informações sobre a interação do AGENTE-M com o agente usuário. As ações que esse componente deve realizar dependem do fato que lhe é comunicado, que pode ser originado pelo próprio agente ou pelo agente usuário.

O fato comunicado pelo agente usuário é somente o RA a matricular (fato_comunicado) e corresponde ao átomo de entrada do componente.

Já o AGENTE-M transfere dados para o agente usuário durante toda a matrícula mostrando sempre os resultados de suas observações e de seu raciocínio, mantendo-o ao par de como esta se sucedendo sua matrícula. Esses resultados informados pelo agente matriculador, que correspondem aos átomos de saída, são a validade do RA informado, os dados cadastrais do aluno, sua situação no curso, seu histórico escolar, e a matrícula gerada (fato_comunicado_agente).

O meta-conhecimento mostrado na figura 18 especifica que se o fato comunicado for proveniente do próprio AGENTE-M e tiver como conteúdo mostrar_dados_cadastrais, o agente matriculador deverá exportar esses dados para o agente usuário.

Se fato-comunicado_por (X:mostrar_dados_cadastrais, Agente: M)
então exportar (X:dados_cadastrais, Agente: U)
e próximo_estado_link (exporta_informação_para_agente_usuario, X)

Figura 18 - Especificação de interação do AGENTE-M com o agente usuário.

A figura 19 apresenta a estrutura do Componente Gerenciamento de Agentes e destaca o meta-conhecimento que determina que, se o fato comunicado pelo Agente:M (AGENTE-M) for início de execução, sua ação será a de importar do Agente:U (agente usuário) o RA do aluno a matricular.

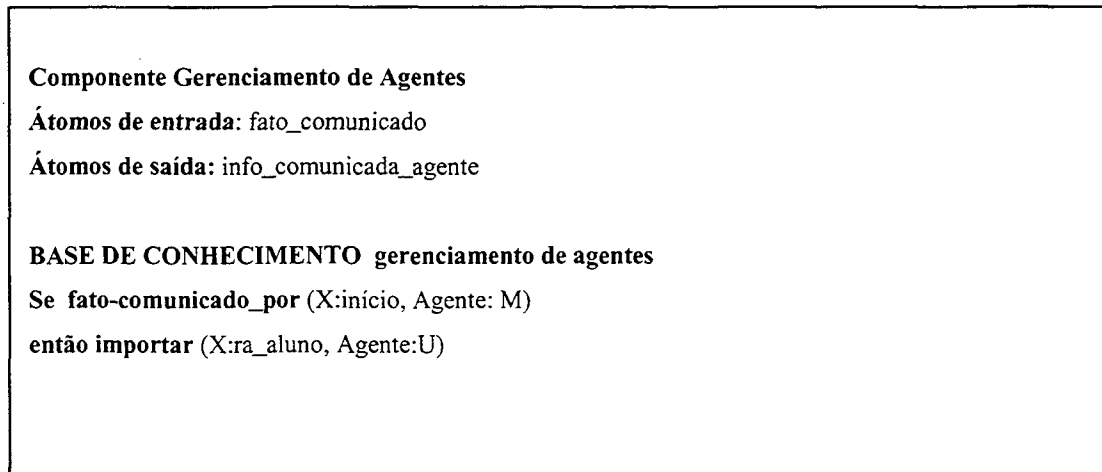


Figura 19 – Meta-conhecimento do Componente Gerenciamento de Agentes.

5.7 Componente Tarefas Específicas do Agente

Neste componente foram modeladas as tarefas próprias do domínio do agente, que são definidas em subcomponentes específicos., existindo também um subcomponente que avalia o estado do processo dentro do próprio componente.

O diagrama da figura 20 apresenta a (de)composição do Componente Tarefas Específicas do Agente. O conhecimento do controle da tarefa desse componente determina que inicialmente o Componente Avalia Estado do Processo é ativado, para identificar qual tarefa específica deverá ser realizada no momento, de acordo com a informação recebida durante sua ativação.

A figura 21 apresenta a especificação da tarefa através da qual o AGENTE-M gera uma matrícula.

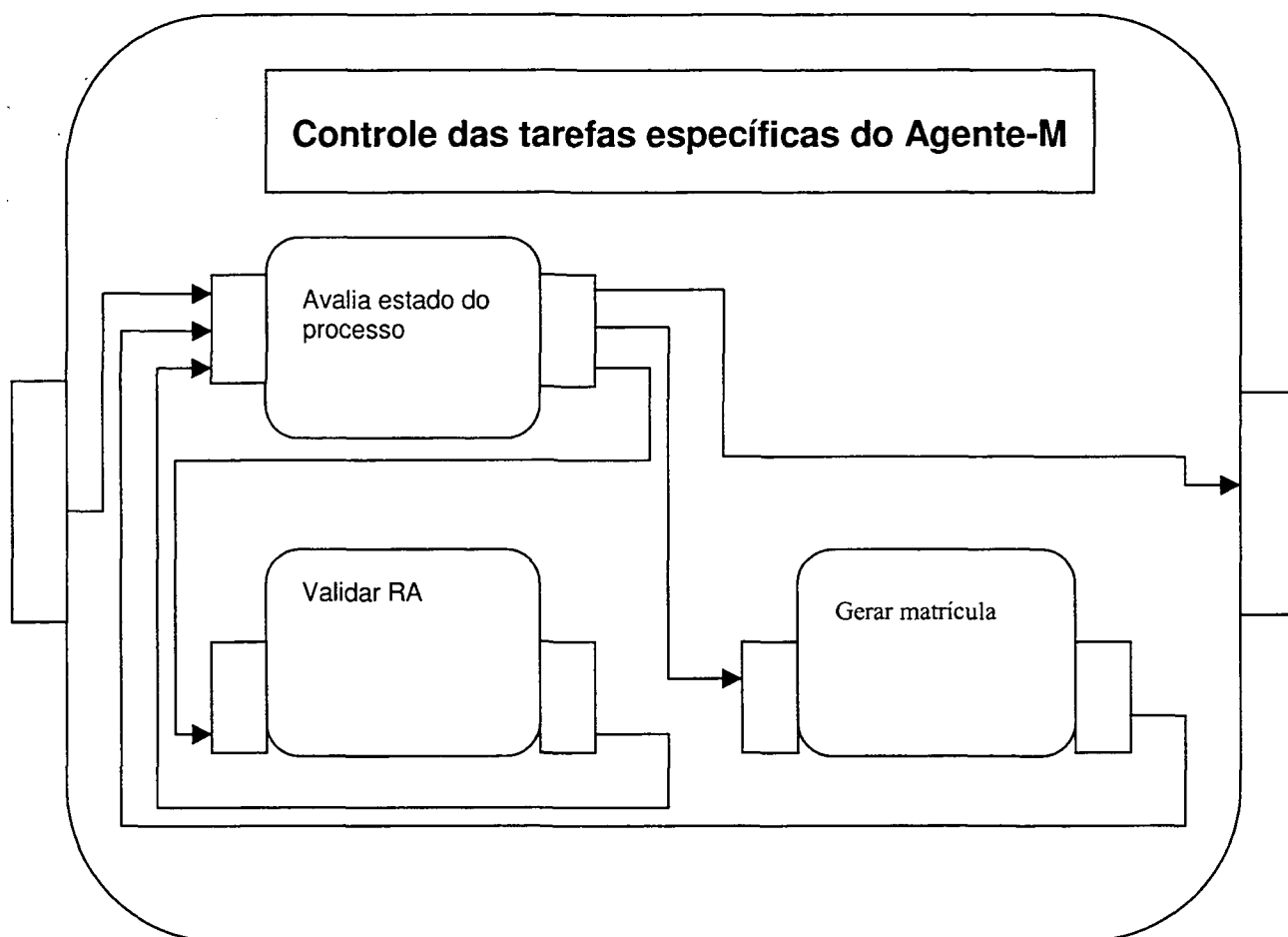


Figura 20 - Tarefas específicas do Agente-M.

Componente GERAR_MATRÍCULA**ÁTOMOS DE ENTRADA:** ra_validado**ÁTOMOS DE SAÍDA:** resultado_da_matricula**BASE DE CONHECIMENTO gerar_matricula:**

se avaliação (X:situação_academico, trancado, formado, jubilado, transferido)

então próximo_estado_link (transfere_resultado_matricula, matrícula não gerada)

se avaliação(X:situação_academico, calouro)

então (matricula gerada:Situação , Aprovado s/dependência)

e (matricula gerada:série , primeira série)

e (matricula gerada:disciplinas, disciplinas da primeira série)

e **próximo_estado_link** (transfere_resultado_matricula, matrícula gerada)

se avaliação(X:situação acadêmico, aprovado s/dependência)

então (matricula gerada:serie, serie atual + 1)

e (matricula gerada: disciplinas, disciplinas da série)

e **próximo_estado_link** (transfere_resultado_matricula, matrícula gerada)

se avaliação(X:situação acadêmico, aprovado c/dependência)

então (matricula gerada: serie, serie atual + 1)

e (matricula gerada: disciplinas, disciplinas da série + disciplinas reprovadas)

e **próximo_estado_link** (transfere_resultado_matricula, matrícula gerada)

se avaliação(X:situação acadêmico, retido)

então (matricula gerada:serie, serie atual)

e (matricula gerada: disciplinas, disciplinas reprovadas)

e **próximo_estado_link** (transfere_resultado_matricula, matrícula gerada)

se avaliação(X:situação acadêmico, reprovado)

então (matricula gerada: serie, serie atual)

e (matricula gerada: disciplinas, disciplinas reprovadas)

e **próximo_estado_link** (transfere_resultado_matricula, matrícula gerada)**Figura 21 – Meta-conhecimento para a tarefa específica Gerar_matricula.**

A tarefa de gerar matrícula é realizada em função da avaliação da situação do aluno, que determinará em que série a matrícula será gerada, em quais disciplinas e quais os dados que serão exportados para o mundo.

Se a avaliação da situação resultar em que o aluno não está em situação regular, então será exportado para o agente a informação de que a matrícula não será gerada.

Vale destacar que se, para cooperar com outros agentes em determinado ambiente, for atribuída uma nova tarefa específica para o AGENTE-M, basta criar um novo subcomponente dentro do Componente Tarefa Específica do Agente, com as especificações da nova tarefa, e ajustar o Componente de Avaliação do Estado do Processo. Os demais componentes do AGENTE-M não precisam ser alterados em função da nova tarefa específica.

Após a descrição dos componentes apresentaremos na figura 22 o diagrama de fluxo de raciocínio estabelecido para o AGENTE-M. A tabela 4 contém a descrição dos links estabelecidos entre os componentes, numerados no diagrama.

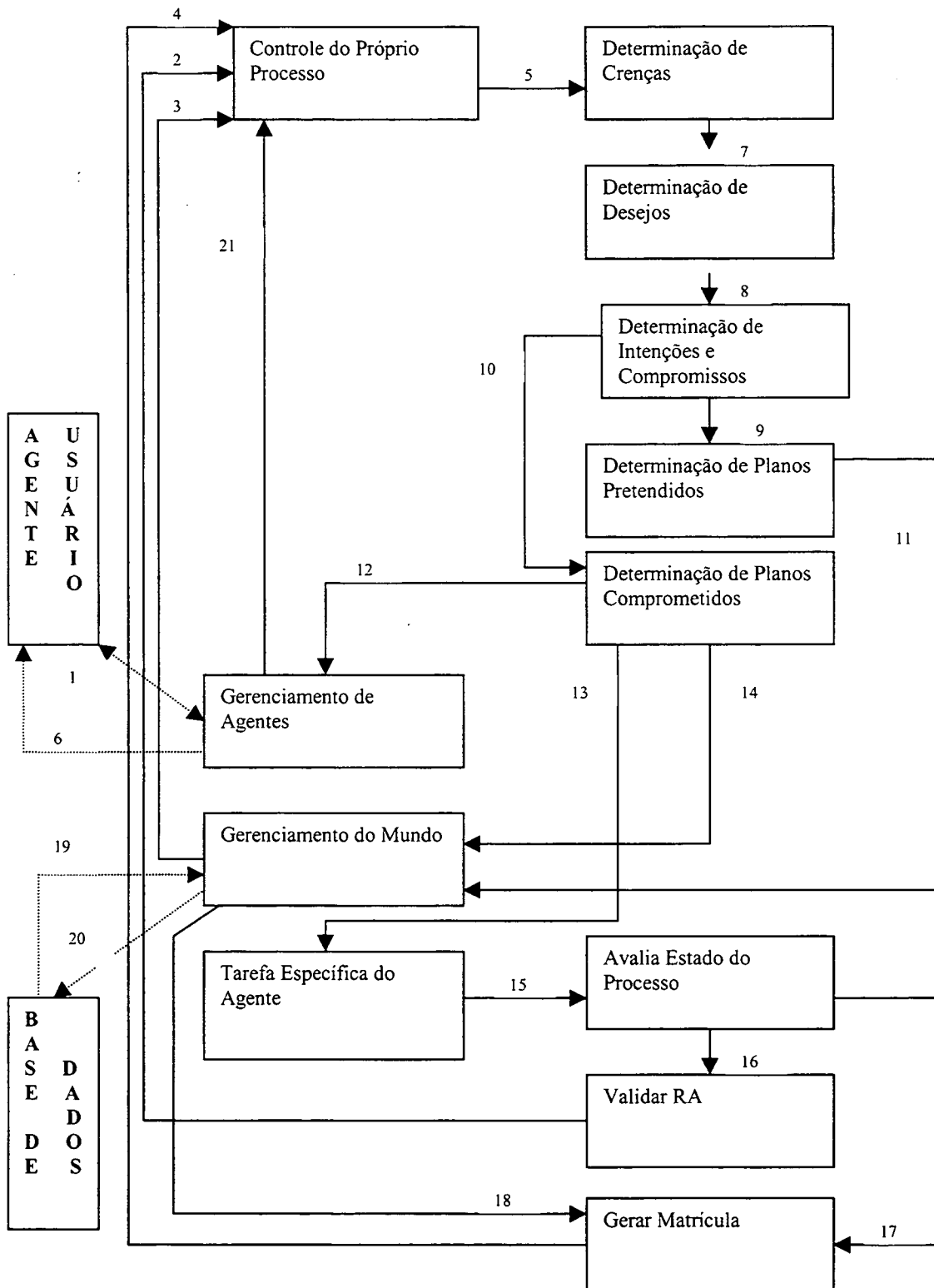


Figura 22 – Diagrama de Fluxo de Raciocínio do AGENTE-M.

Nome do Link	Componente Origem	Componente Destino
1 - Importa RA do agente usuário	Agente usuário	Gerenciamento de Agentes
2 - Transfere resultado da validação do RA	Validar RA	Controle do Próprio Processo
3 - Prove informação do estado do mundo	Gerenciamento do mundo	Controle do próprio processo
4 - Transfere resultado da geração de matrícula	Gerar Matrícula	Controle do Próprio Processo
5 - Transfere estado atual do agente	Controle do Próprio Processo	Determinação de Crenças
6 - Exporta informação para o agente usuário	Gerenciamento do Agente	Agente usuário
7 - Transfere crença	Determinação de Crenças	Determinação de Desejos
8 - Transfere Desejo	Determinação de Desejos	Determinação de Intenções e Compromissos
9 - Transfere Intenção	Determinação de Intenções e Compromissos	Determinação de Planos Pretendidos
10 - Transfere Compromisso	Determinação de Intenções e Compromissos	Determinação de Planos Comprometidos
11 - Exporta plano de observar dados	Determinação de Planos Pretendidos	Gerenciamento do Mundo
12 - Exporta plano de mostrar informações	Determinação de Planos Comprometidos	Gerenciamento de Agentes
13 - Transfere ações comprometidas	Determinação de Planos Comprometidos	Tarefa Especifica do Agente
14 - Exporta plano de alterar dados	Determinação de Planos Comprometidos	Gerenciamento do Mundo
15 - Transfere tarefa a executar	Tarefa Especifica do Agente	Avalia Estado do Processo
16 - Ativa tarefa de validar RA	Avalia Estado do Processo	Validar RA
17 - Ativa tarefa de gerar matrícula	Avalia Estado do Processo	Gerar matrícula
18 - Transfere dados observados	Gerenciamento do Mundo	Gerar Matrícula
19 - Importa informação do Mundo	Base de Dados	Gerenciamento do Mundo
20 - Exporta informação para o mundo	Gerenciamento do Mundo	Base de Dados
21 - Transfere informação comunicada	Gerenciamento de Agentes	Controle do Próprio Processo

Tabela 4 – Links entre os componentes do AGENTE-M.

Através do diagrama da figura 22 podemos destacar alguns aspectos importantes dessa estrutura:

- Componente Controle do Próprio Processo e seus subcomponentes são os responsáveis pela determinação do fluxo de raciocínio do agente.

- A interação entre os componentes acontece de diferentes formas dependendo do estado mental que o agente apresentar., ficando evidente que toda interação com agente (agente usuário) ocorre através do componente Gerenciamento de Agentes, e com o mundo (base de dados) através do Componente Gerenciamento do Mundo.
- Componente Determinação de Planos Pretendidos interagem somente com o Gerenciamento do Mundo porque a pretensão do agente depende das informações contidas na base de dados.
- O Componente Determinação de Planos Comprometidos interage com os componentes que viabilizam a execução de um plano para atingir determinada meta. São eles: Gerenciamento do Mundo, Gerenciamento do Agente e Tarefa Específica do Agente.
- Toda vez que uma tarefa é executada os componentes retornam o controle para o Componente Controle do Próprio Processo para que o estado mental do agente seja ajustado em função da tarefa executada.

Podemos também ,através do diagrama da figura 22, identificar o fluxo básico de tarefas e interação entre os componentes do AGENTE-M. Usemos como exemplo a situação ideal que é aquela em que o agente usuário informa um RA válido, cuja situação acadêmica é regular, podendo o agente matriculador gerar uma matrícula sem restrições.

Para essa situação o fluxo básico de tarefas dentro do AGENTE-M é:

1. O Componente Gerenciamento de Agentes é ativado recebendo a informação do RA a matricular comunicada pela agente usuário, e transfere essa informação para o componente Controle do Próprio Processo.
2. Identifica-se a crença de um usuário que deseja efetuar sua matrícula despertando, no Componente Determinação de Desejos, o desejo de validar o RA.
3. Esse desejo é transferido para o Componente Determinação de Intenções e Compromissos que ativa seu componente Determinação de Planos Comprometidos para definir a estratégia para realizar a validação.

4. A informação sobre o planejamento é transferida para o Componente Tarefa Específica do Agente, onde o Componente Avalia Estado do Processo determina a ativação da tarefa “Validar RA”.
5. O resultado da validação do RA é transferido para o Componente Controle do Próprio Processo. Sendo a validação bem sucedida o AGENTE-M passa a acreditar na possibilidade da efetivação da matrícula.
6. Seu próximo desejo é mostrar os dados cadastrais do aluno, necessitando encontrá-los na base de dados. Para tanto o Componente Determinação de Desejos transfere tal desejo para o Componente Determinação de Intenções e Compromissos que ativa a intenção(Planos Pretendidos) de localizar os dados. Essa intenção é transferida para o Componente Gerenciamento do Mundo que fará a interação com o mundo para observar esses dados.
7. A informação dos dados encontrados do aluno é transferida para o Componente Controle do Próprio Processo e o AGENTE-M assume o compromisso de mostrá-los ao agente usuário. Esse estado mental é controlado dentro do agente através da ativação do plano comprometido de mostrar dados cadastrais ao aluno, que ativará o Componente Gerenciamento de Agentes para interagir com o usuário comunicando tais dados.
8. A mesma seqüência de raciocínio dos itens 6 e 7 ocorre em relação ao desejo de mostrar os dados referentes ao histórico escolar ao aluno. O agente observa no mundo a existência desses dados e comunica-os ao agente usuário.
9. Como o RA é válido e o dados cadastrais e de histórico já foram mostrados ao aluno o agente deseja (Determinação de Desejos) e ,consequentemente, tem a intenção (Determinação de Intenções e Compromissos) de matriculá-lo. Para tanto traça um plano pretendido (Determinação de Planos Pretendidos), onde deverá primeiramente verificar a situação do acadêmico (reprovado, aprovado, etc.). O Componente Determinação de Planos Pretendidos transfere ao Componente Gerenciamento do Mundo a necessidade de verificar no mundo (base de dados) a situação do aluno.

10. Tendo o aluno uma situação normal, que permite ser matriculado, o agente passa a acreditar nessa situação, o que desperta nele o desejo de gerar a matrícula. O agente assume o compromisso de realizar tal desejo e traça um plano comprometido para atingir essa meta. Para a execução desse plano o Componente Determinação de Planos Comprometidos informa ao Componente Tarefa Específica do Agente a necessidade de gerar matrícula.
11. O Componente Avalia Estado do Processo determina a ativação da tarefa “gerar matrícula”. Surge a necessidade de buscar dados no mundo, o que é feito pela ativação do Componente Gerenciamento do Mundo. Os dados observados são transferidos para o Componente Tarefa Específica do Agente que continua gerando a matrícula.
12. Uma vez gerada a matrícula, o Componente Controle do Próprio Processo percebe esse fato e deseja mostrar a composição feita para o aluno. Esse desejo leva a intenção de realizá-lo e para isso é estabelecido um plano comprometido. Ativa-se então o Componente Gerenciamento de Agentes que comunicará o resultado da matrícula ao aluno.
13. O próximo passo do agente é atualizar o estado do mundo em função da matrícula gerada. O Componente Gerenciamento do Mundo determina a atualização da base de dados com as novas informações referentes ao aluno.
14. Tendo concluído o processo de matrícula o agente comunica esse fato ao agente usuário, via Componente Gerenciamento de Agentes. Agora pode esquecer essas tarefas realizadas e ficar disponível para realizar outra matrícula. Caso o aluno que foi recém matriculado tente repetir a matrícula, o AGENTE-M não realizará o processo porque possui o conhecimento, acessando a base de dados, de que tal matrícula já foi realizada.

Uma vez definidos todos os componentes do AGENTE-M, bem como seu fluxo de raciocínio e execução de tarefas, veremos a seguir como foi realizada a implementação do AGENTE-M na linguagem LOOM.

5.8 Implementação do AGENTE-M em LOOM

Utilizamos para implementar nosso agente a linguagem LOOM que corresponde a uma biblioteca de subrotinas LISP [LOOM91a]. Por questões de limitação financeira utilizamos a versão “trial” do Allegro Common Lisp [FRAN99]. O ambiente de base foi um equipamento PENTIUM III com 64MB de memória, e Sistema Operacional LINUX.

Para a implementação do AGENTE-M estabelecemos que o comportamento do agente, definido na base de conhecimento de cada componentes, fosse representado através de métodos, que correspondem a conjuntos de operações associadas a determinada ação. A representação do conhecimento próprio do agente, ou seja, sua ontologia, foi feita através de definição de conceitos e instâncias.

A chamada entre os métodos corresponde a interação do agente com seus próprios componentes, com o mundo e com o agente usuário.

Vejamos agora como foram implementados os estados mentais do AGENTE-M em LOOM.

5.8.1 Implementação das Crenças

Para expressar em LOOM as crenças do agente partiu-se do princípio de que tudo aquilo em que o agente acredita é um conceito que ele compreende.

A definição de conceitos que existem no mundo são feitas através do comando `defconcept`. Essas definições são adicionadas a base de conhecimento e o Loom executa então uma classificação, onde as definições dos conceitos e relações são organizadas para formar uma taxonomia de termos específica do domínio.

A linguagem Loom utiliza a noção de conceito primitivo, como sendo o conceito que não é ou não pode ser definido completamente por referência a outros conceitos. Declarando um conceito como primitivo estamos em efeito dizendo ao Loom que existem atributos ocultos sobre objetos daquele tipo que nós não estamos representando [LOOM91a].

A escolha de quais conceitos devem ser declarados primitivos depende muito do uso pretendido do conceito representado.

Em nossa aplicação é importante saber que um **RA válido** é “um Registro acadêmico de um aluno e que seu curso é 29 (Informática)”. Assim **aluno** e **curso** são conceitos primitivos.

Os conceitos compostos de combinações e especialização de outros conceitos, tal como o RA válido em nosso caso, são chamados conceitos definidos. A figura 23 mostra a implementação desses conceitos:

```
//conceitos primitivos
(defconcept Aluno
  :is primitive)
(defconcept Curso-Aluno
  : is primitive )

//conceitos definidos

(defconcept Ra-Valido
  :is (and Aluno (:the Curso-Aluno 29))
  :identifier Ra-Valido-Concept)
```

Figura 23 – definição das crenças em LOOM

A especialização de um conceito formando um novo, foi feita através da restrição de valor de determinada função. O mostrado na figura 24 define a ontologia para o conceito **Aluno-Calouro** como sendo uma especialização do conceito **Aluno** que restringe a situação do mesmo em ser **Calouro**.

```
(defconcept Aluno-Calouro
  :is (and Aluno (:some Situacao "Calouro")))
```

Figura 24 – Restrição sobre o fato do aluno ser calouro.

Vale mencionar que ao definirmos as crença, ou seja os conceitos, estamos definindo a ontologia para o agente, a qual permitirá que ele seja inserido em vários ambientes, desde que adotem tal ontologia.

5.8.2 Implementação dos Desejos

Considerando a arquitetura projetada para o AGENTE-M, vemos que os desejos fazem parte do subcomponente Determinação de Desejos. Sendo assim sua implementação está em um submétodo que recebe o mesmo nome do subcomponente, e a lista de chamadas dos possíveis desejos.

A figura 25 demonstra como foi implementado esse componente. Todos os desejos do agente estão no método Determinação-dos-Desejos. A ativação de cada um deles depende da condição estabelecida no comando :SITUATION ser verdadeira.

Para representarmos os desejos definimos uma ontologia para “desejo”, através da definição de um conceito correspondente a esse termo. Como o AGENTE-M tem vários desejos possíveis, estabelecemos também o conceito de “desejo atual” cuja definição determina aquilo que espera-se que aconteça no momento vigente.

Usemos como exemplo o desejo do agente de mostrar os dados cadastrais, cujo meta-conhecimento está especificado na seção 5.4.2.

Para que esse desejo seja ativado é necessário que o estado atual do agente seja ter o RA válido. Assim, o comando “perform desejo-mostrar-dados-cadastrais” significa apenas que será verificada a possibilidade de sua ativação. Caso a condição dentro do comando :SITUATION seja satisfeita, utilizando o comando <TELLM> dizemos que o desejo atual do agente tem como definição “mostrar-dados-cadastrais”.

A partir do desejo estabelecido ativaremos o subcomponente Determinação-de-Intenções através do comando PERFORM.

```

(DEFMETHOD DETERMINACAO-DOS-DESEJOS (?R)
:TITLE "Componente Determinação dos Desejos"
:RESPONSE
((PERFORM (DESEJO_VALIDAR_RA))
 (PERFORM (DESEJO_MOSTRAR_DADOS_CADASTRAIS))
 (PERFORM (DESEJO_MOSTRAR_HISTÓRICO))
 (PERFORM (DESEJO_MOSTRAR_HISTÓRICO))
 (PERFORM (DESEJO_GERAR_MATRÍCULA))
)

(DEFMETHOD DESEJO_MOSTRAR_DADOS_CADASTRAIS (?R)
:TITLE "Desejo de mostrais os dados cadastrais para o agente
      usuário"
:SITUATION (RA-VALIDO ?RA)
:RESPONSE
( (TELLM (:ABOUT DESEJO_ATUAL DESEJO
            (DEFINICAO_MOSTRAR_DADOS_CADASTRAIS))
  (PERFORM (DETERMINACÃO-DE-INTENCOES ?R ))
)

```

Figura 25 – Implementação do desejo do agente de mostrar os dados cadastrais do aluno.

5.8.3 Implementação das Intenções e Compromissos

A implementação das intenções e compromissos está em um submétodo, seguindo a mesma linha do subcomponente Determinação dos Desejos. O submétodo Determinação-de-Intenções contém a lista das intenções, as quais são ativadas de acordo com um desejo ou meta atingida. Para os planos definimos ontologia onde são definidos os conceitos de “plano pretendido” e “plano comprometido”, que são planos com características especiais.

Dando continuidade ao exemplo da seção 5.8.2 o desejo de mostrar os dados cadastrais leva a intenção de realizar tal desejo. Uma vez ativado o método que corresponde a

intenção teremos a programação do plano de execução, conforme o meta-conhecimento especificado na seção 5.4.3.

A figura 26 apresenta a implementação da intenção do AGENTE-M de mostrar os dados cadastrais. Vemos que o comando SITUATION expressa que essa intenção deriva do desejo “mostrar-dados-cadastrais”.

O comando <TELLM> diz que o plano pretendido envolve a estratégia de localizar dados. Para executar esse plano é determinada a ativação do Componente Gerenciamento do Mundo. Em seguida o agente compromete-se com um novo plano cuja estratégia determina que deve-se comunicar os dados localizados. Finalmente, ativa-se o Componente Gerenciamento de Agentes, que deverá executar esse plano..

```
(DEFMETHOD INTENÇÃO-MOSTRAR-DADOS (?R ?INTENCAO)
:TITLE "Intenção de mostrar ao aluno os dados cadastrais"
:SITUATION (MOSTRAR-DADOS-CADASTRAIS DESEJO_ATUAL)
:RESPONSE
((TELLM (:ABOUT PLANO_PRETENDIDO PLANO
          (ESTRATÉGIA LOCALIZAR-DADOS-CADASTRAIS))
((PERFORM (GERENCIAMENTO-DO-MUNDO ?R ))
((TELLM (:ABOUT PLANO_COMPROMETIDO PLANO
          (ESTRATÉGIA COMUNICAR-DADOS-CADASTRAIS))
(PERFORM (GERENCIAMENTO-DE-AGENTES ?R))
)
```

Figura 26 – Intenção do AGENTE-M em mostrar ao aluno os dados cadastrais.

5.9 Implementação do Gerenciamento do Mundo

Criamos uma base de dados no ambiente LOOM para representar o mundo do AGENTE-M. As observações e alterações do estado do mundo são feitas nessa base.

O método Gerenciamento-do-Mundo que implementa tal componente contém vários submétodos que acessam essa base de acordo com sua tarefa.

Criamos o conceito “dados localizados” definido a partir do conceito “dados cadastrais”, isto é, os dados localizados são os dados cadastrais que serão utilizados pelo agente.

Na figura 27 apresentamos o submétodo Localizar-Dados-Cadastrais que foi ativado na execução do plano referente a intenção de localizar os dados cadastrais, citado na seção 5.8.3.

Nota-se que as informações observadas na base de dados são armazenadas como dados localizados. O nome do aluno, por exemplo, é transferido do conceito “NOME-ALUNO” para “NOME-LOCALIZADO”.

```
(DEFMETHOD LOCALIZAR-DADOS-CADASTRAIS (?R ?PLANO_C)
 :TITLE "Localizar no mundo os dados cadastrais do aluno"
 :SITUATION (LOCALIZAR-DADOS-CADASTRAIS PLANO_COMPROMETIDO)
 :RESPONSE

 ((TELLM (:ABOUT DADOS-LOCALIZADOS DADOS CADASTRAIS
 (NOME-LOCALIZADO NOME-ALUNO)
 (ENDERECO-LOCALIZADO ENDERECO-ALUNO)
 (FONE-LOCALIZADO FONE-ALUNO)
 (SERIE-LOCALIZADA SERIE-ATUAL))
```

Figura 27 – Localização dos dados cadastrais do aluno no mundo (base de dados).

5.10 Implementação do Gerenciamento de Agentes

A interação do AGENTE-M com o agente usuário ocorre de forma restrita em LOOM, por ser um sistema de representação do conhecimento que trabalha em forma de perguntas e asserções. Assim a comunicação do RA a matricular feita pelo agente usuário ao agente matriculador é feita através de uma asserção, como mostra a figura 28.

```
(setq ?r (get-instance 'RA11029))  
(perform (Agente_M ?r))
```

Figura 28 – Comunicação do RA a matricular.

A comunicação de dados observados e resultados obtidos pelo AGENTE-M ao agente usuário ocorre da seguinte forma: o método Gerenciamento de Agentes tem a lista de chamada de todas as comunicações possíveis de serem feitas. De acordo com a informação que recebe quando é ativado realiza uma ou outra comunicação.

A figura 29 mostra a implementação da tarefa do AGENTE-M de comunicar os dados cadastrais ao agente usuário. Para realizar essa tarefa é utilizado o comando <KBCLASSES:SAY> que mostrará para o agente usuário os dados localizados.

```

(DEFMETHOD MOSTRAR-DADOS-CADATRAIS (?R)
:TITLE "Comunicar dados cadastrais ao agente usuário"
:SITUATION (COMUNICAR-DADOS-CADASTRAIS PLANO-COMPROMETIDO)
:RESPONSE
((KBCLASSES:SAY "DADOS CADASTRAIS" )
 (KBCLASSES:SAY "Nome: " NOME-LOCALIZADO)
  (KBCLASSES:SAY "Endereço: " ENDERECO-LOCALIZADO)
  (KBCLASSES:SAY "Telefone: " FONE-LOCALIZADO)
  (KBCLASSES:SAY "Série Atual: " SERIE-LOCALIZADA)
  (KBCLASSES:SAY " ATENÇÃO: Caso seus dados estejam desatualizados,
informe na secretaria" )
)

```

Figura 29 – Implementação da comunicação dos dados cadastrais ao agente usuário.

5.11 Implementação das tarefas específicas do AGENTE-M

O AGENTE-M possui duas tarefas específicas já definidas: validar o RA e gerar matrícula. Cada tarefa foi definida em um método.

Com relação à tarefa de validação do RA, tivemos um problema relacionado à versão que utilizamos da linguagem LOOM, que só permite a inferência sobre instâncias já criadas na base de conhecimento. Isso requer que todo RA a ser matriculado exista na base de dados. Assim a validação do RA só pode ser feita no sentido de o aluno pertencer ao curso de informática ou não, não podendo ser informado um RA que não exista porque resulta em erro e interrupção de execução do programa LOOM.

A implementação da tarefa para geração de matrícula está descrita na figura 30. Através do comando <RETRIEVE> são observados no mundo todas as disciplinas na qual o aluno deve ser matriculado. Cada disciplina é acrescentada ao histórico pelo comando <TELLM (CREATE)> , que na verdade cria uma nova instância para o conceito HISTÓRICO.

As informações relacionadas as disciplinas que são incorporadas ao histórico do aluno, correspondem a relações que foram definidas para o domínio do conceito HISTÓRICO como, por exemplo, a situação do aluno em relação a disciplina (SITUAÇÃO-DISCIPLINA). Essa

relação é estabelecida aqui tendo como conteúdo "C", indicando que o aluno passa a estar cursando a disciplina.

```
(DEFMETHOD GERAR-MATRICULA (?R) :TITLE "Matricular Aluno Aprovado"
  :SITUATION (ALUNO-APROVADO ?R) :RESPONSE
  ((KBCLASSES:SAY "Matriculando Aluno Aprovado...")
   (SETQ ?SA (GET-VALUE ?R 'SERIE-ATUAL))
   (TELLM (:ABOUT ?R ALUNO (SERIE-ATUAL (+ ?SA 1))))
   (SETQ ?SN (GET-VALUE ?R 'SERIE-ATUAL))
   (SETQ ?T (GET-VALUE ?R 'TURMA))
   (SETQ ?RA (GET-VALUE ?R 'RA))
   (DO-RETRIEVE (?D)
     (:ABOUT ?D DISCIPLINA
      (:FILLED-BY SERIE-DISCIPLINA ?SN))
     (SETQ ?CD (GET-VALUE ?D 'CODIGO-DISCIPLINA))
     (KBCLASSES:SAY "Disciplina: " ?CD)
     (SETQ ?P (GET-VALUE 'SEQ1 'PROX-HISTORICO))
     (TELLM (:ABOUT
              SEQ1
              SEQ-HISTORICO
              (PROX-HISTORICO (+ ?P 1))))
     (SETQ ?H
      (RETRIEVE
       ?H
       (:ABOUT
        ?H
        HISTORICO
        (:FILLED-BY SEQ-H ?P))))
     (TELLM (CREATE ?H HISTORICO)
      (ABOUT
       ?H
       (RA-HISTORICO ?RA)
       (COD-DISCIPLINA ?CD)
       (ANO-MATRIC-DISCIPLINA 2000)
       (SITUACAO-DISCIPLINA "C")
       (TURMA-MATRIC-DISCIPLINA ?T)
       (SEQ-H ?P))))))
```

Figura 30 – Implementação da tarefa específica do AGENTE-M para gerar matrícula.

5.12 Discussões

Esse capítulo descreveu o AGENTE-M que foi projetado com base na arquitetura BDI proposta pelo DESIRE e implementado na linguagem de representação de conhecimento LOOM.

O grupo de pesquisas do DESIRE possui um ambiente próprio de implementação mas que é restrito a pessoas que fazem o curso específico na Universidade de Amsterdã, o que inviabilizou nosso acesso a esse ambiente.

Projetamos então o AGENTE-M utilizando as características do DESIRE que consideramos mais interessantes e adequadas para serem aplicadas. Esse projeto resultou numa arquitetura composta por quatro componentes principais e oito subcomponentes, com raciocínio explícito sobre o próprio agente, sobre o mundo e sobre o agente usuário.

O AGENTE-M realiza com autonomia o processo de matrícula, interagindo com o aluno que deseja ser matriculado (agente usuário) e com a base de dados que contém as informações necessárias (mundo).

Na arquitetura do AGENTE-M optamos por concentrar o conhecimento referente a interação dos agentes em um único componente, assim como para o conhecimento sobre a interação com o mundo.

Concluimos que o projeto do agente em componentes específicos facilita a compreensão e manutenção do seu conhecimento, e o uso de componentes específicos de interação com o mundo e agentes contribuiu para nosso objetivo de tornar explícito seu raciocínio sobre ele mesmo, sobre o mundo e sobre outros agentes. Esse objetivo foi alcançado através do uso de uma base de conhecimento específica para cada componente, e pela especificação dos átomos de entrada e saída que deixam claro quais informações são recebidas e enviadas pelo componente.

Sentimos certa dificuldade em estabelecer o relacionamento entre os estados mentais, uma vez que já no ser humano isso não é fácil. Contornamos essa situação ao adotar uma seqüência lógica de raciocínio considerando que desejamos algo com base em crenças. Esses desejos geram intenções que levam a ações que devem ser executadas. Se considerarmos exatamente o raciocínio humano sabemos que existem momentos em que os estados mentais entram em conflito. Essa questão não foi considerada em nosso trabalho.

A principal diferença da estruturação dos estados mentais do AGENTE-M em relação ao DESIRE corresponde ao tratamento de metas. Enquanto o DESIRE possui componentes específicos para a determinação de metas, em nosso projeto as consideramos como parte de

um plano que deve ser executado para realizar um desejo. Assim elas foram especificadas como parte dos planos nos componentes Determinação de Planos Pretendidos e Determinação de Planos Comprometidos.

Com respeito à implementação, a linguagem LOOM apresentou forte poder de representação do conhecimento e inferência. Embora Hayes [HAYE99] comente que essa linguagem é voltada para agentes, não identificamos um ambiente previamente estruturado para a implementação dos mesmos, mas não encontramos grandes dificuldades em compor o AGENTE-M.

Utilizando LOOM conseguimos criar uma estrutura distribuída em componentes, como propusemos no projeto do agente e , através dos comandos de definição de conceitos, implementamos as ontologias para conceitualizar os estados mentais do agente e as demais informações pertencentes a sua base de conhecimento. Assim, o AGENTE-M teve tanto sua estrutura como seu conhecimento implementados de forma explícita.

Em comparação ao agente desenvolvido anteriormente dentro do SISMAT, consideramos que o AGENTE-M possui um poder maior de representação de seu conhecimento, porque representa explicitamente o raciocínio sobre o agente usuário e sobre a base de dados, enquanto no SISMAT essa representação era implícita.

Concluimos então que para construirmos um agente BDI o ambiente ideal é aquele que permite a flexibilidade de criar o agente em componentes e que ofereça facilidades de comunicação.

6. Resultados: AGENTE-M

6.1 Filosofia de operação

Por motivos de restrições da versão da linguagem Common Lisp utilizada para a linguagem LOOM, não pudemos gerar uma produção do AGENTE-M, o que significa que ele é sempre usado dentro do ambiente LOOM, num esquema interativo de perguntas e asserções.

A operação do AGENTE-M é bastante simples, bastando setar uma variável com o número do RA do aluno que se deseja matricular, e dar o comando para ativar o agente matriculador AGENTE-M, como mostrado na figura 31.

```
(SETQ ?RA (GET-INSTANCE 'RA971021129'))  
(PERFORM (AGENTE-M ?R))
```

Figura 31 – Agente usuário informa o RA e ativa o AGENTE-M.

A partir desse ponto o agente é completamente autônomo raciocinando sobre o RA informado, sendo apresentado na tela os resultados de seu raciocínio.

No início de execução o AGENTE-M mostra para confirmação o RA que será matriculado (figura 32).

=====

SISTEMA DE MATRICULA DO CURSO DE INFORMÁTICA

=====

RA DO ALUNO A MATRICULAR: RA971020829

Figura 32 - Apresentação em tela do RA a matricular.

O agente matriculador tem o compromisso de validar o RA. Uma vez atingida a meta de validação teremos duas situações possíveis de serem informadas ao agente usuário: RA inválido ocasionando o fim de operação do AGENTE-M (figura 33), ou RA válido (figura 34) permitindo a continuidade do processo de matrícula.

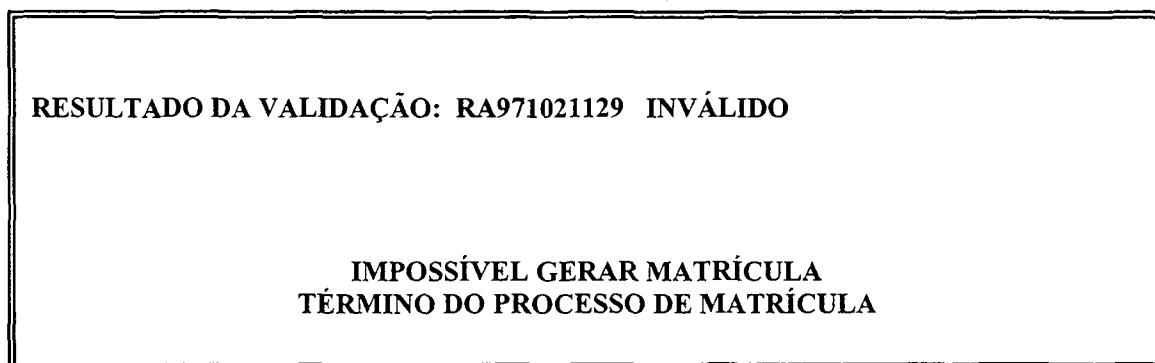


Figura 33 – AGENTE-M comunica ao agente usuário que o RA informado é inválido.

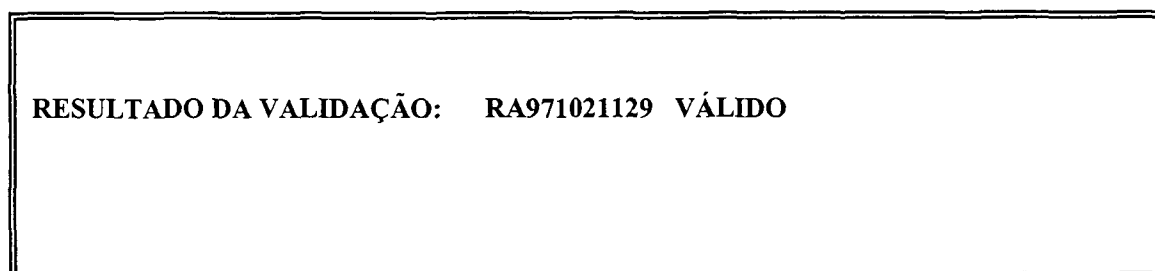


Figura 34 – AGENTE-M comunica ao agente usuário que o RA informado é válido.

Sendo válido o RA a próxima meta do AGENTE-M é localizar dados pertinentes ao aluno e mostrá-los ao agente usuário, como mostra a figura 35. Para tanto, o Componente Gerenciamento do Mundo é ativado e a interação com o mundo é estabelecida sendo observada a base de dados Cadastro do Aluno.

DADOS DO ALUNO	
Nome:	LUCIANA MIRELLI
Data de Nascimento:	05/11/77
Série Atual:	2
Data da última matrícula:	10/02/1999
Situação Atual:	APROVADA

Figura 35 – Dados comunicados ao aluno resultantes da observação do mundo.

Como o agente acredita que existe um histórico do aluno na base de dados, a meta seguinte é mostrar o histórico do aluno, exceto se for calouro. Nova interação com o mundo é estabelecida, mas a base de dados observada agora é referente a históricos escolares. O agente usuário poderá ver então a tela apresentada na figura 36.

HISTÓRICO DO ALUNO				
DISCIPLINA	CÓDIGO	SITUAÇÃO	ANO DE CURSO	TURMA
LÓGICA COMPUTACIONAL	501070	A	1998	B
CDI	101051	A	1998	B
INGLÊS TÉCNICO	505027	A	1998	B
ESTRUTURA DE PROGRAMAÇÃO	203015	A	1998	B
PLANEJAMENTO EM INFORMÁTICA	203121	A	1999	B
SISTEMAS OPERACIONAIS	203141	A	1999	A
LABORATÓRIO DE HARWARE	203020	A	1999	A
CÁLCULO NUMÉRICO	101053	A	1999	B

Figura 36 – O agente busca dados sobre o histórico do aluno.

O AGENTE-M deseja então matricular tal aluno, mas tem autonomia para decidir se essa tarefa será realizada ou não. Tomará essa decisão com base em suas crenças a respeito

das condições para matrícula que são essencialmente a situação regular e a matrícula para o ano vigente ainda não ter sido efetuada.

Caso sua situação não permita matrícula essa decisão será comunicada ao agente usuário, como mostrado na figura 37.

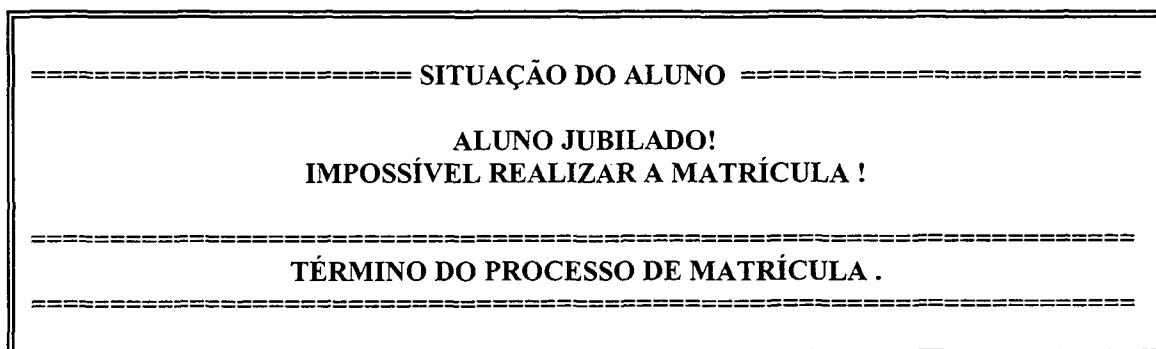


Figura 37 – Comunicação ao agente usuário de que a matrícula não pode ser realizada.

Se o aluno apresentar uma situação possível de matrícula, como o exemplo da aluna Luciana cuja situação e histórico são mostrados nas figuras 35 e 36, o AGENTE-M executará a tarefa específica de gerar matrícula, comunicando também ao agente usuário sua decisão, como mostrado na figura 38.

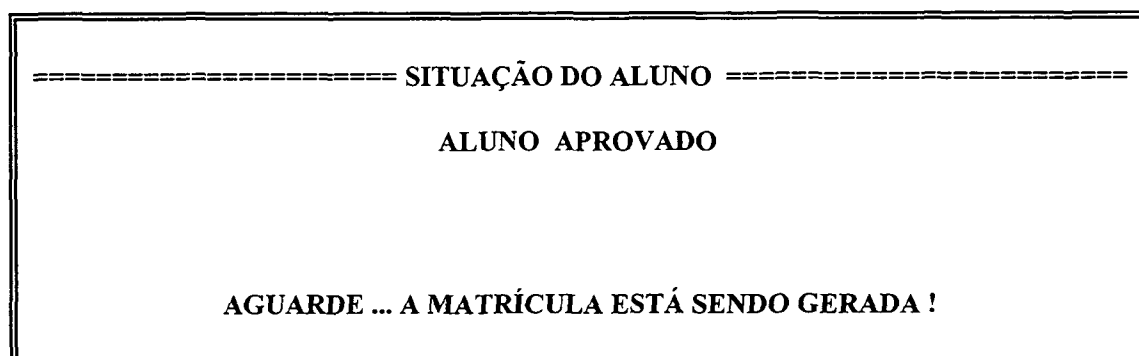


Figura 38 – Agente matriculador informa ao agente usuário a decisão de gerar a matrícula.

Uma vez realizada a tarefa específica de gerar matrícula, essa informação é transferida ao componente Controle do Próprio Processo, onde é identificado então o compromisso do agente em mostrar a matrícula gerada. Para tanto o Componente Gerenciamento de Agentes é ativado para estabelecer uma comunicação com o agente usuário informando a matrícula resultante, como pode ser visto na figura 39.

COMPOSIÇÃO DA MATRÍCULA GERADA			
SÉRIE EM QUE FOI MATRICULADO: 3ª			
DISCIPLINA	CODIGO	TURMA	SITUAÇÃO ATUAL
ANÁLISE DESISTEMAS	203025	B	EM CURSO
ESTATÍSTICA	101055	B	EM CURSO
BANCO DE DADOS	203023	A	EM CURSO
INTELIGÊNCIA ARTIFICIAL	203026	A	EM CURSO
GEOMETRIA ANÁLITICA	101050	B	EM CURSO

Figura 39 – Comunicação ao agente usuário sobre a matrícula gerada.

Concluída a matrícula o Componente Gerenciamento do Mundo é ativado para proceder uma alteração no estado do mundo, atualizando a base de dados referente a nova situação do aluno, série de curso e disciplinas em que foi matriculado. O Componente Gerenciamento de Agentes é ativado em seguida para informar ao agente usuário sobre o término do processo de matrícula (figura 40), e para liberar de sua memória o conhecimento que não é mais necessário. O AGENTE-M torna-se então disponível para iniciar um novo processo de matrícula.

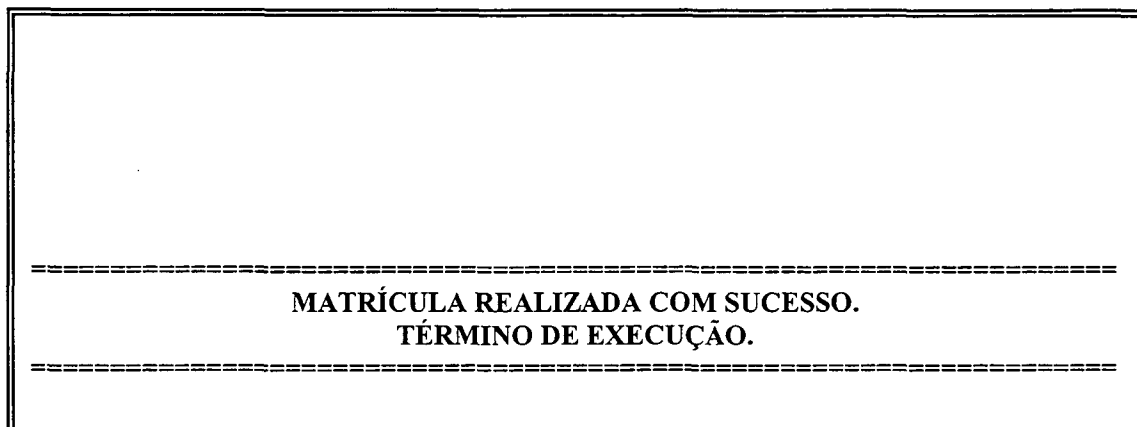


Figura 40 – Comunicação ao agente usuário sobre o término da matrícula.

6.2 Exemplo de uma sessão: matricular aluno aprovado em dependência

Descreveremos nessa seção um exemplo de operação do AGENTE-M matriculando um aluno em regime de dependência, tomando-se por base a seguinte definição:

“Considera-se regime de dependência a faculdade de poder o aluno freqüentar até duras disciplinas em que ficou reprovado simultaneamente com a série imediatamente seguinte, para a qual será regularmente promovido [UEPG99].”

O aluno considerado apresenta a seguinte situação: possui o número de registro acadêmico RA991020029, cursou a primeira série do curso de informática, conforme a grade curricular apresentada na tabela 5, tendo sido reprovado nas disciplinas com códigos 501070 e 101051 e aprovado nas demais, e deseja ser matriculado na segunda série.

Como o aluno apresenta uma situação típica de dependência ele deverá ser matriculado pelo AGENTE-M em todas as disciplinas da segunda série e também nas duas outras em que já reprovou.

BACHARELADO EM INFORMÁTICA

1ª SÉRIE 782 23	Psicologia das Relações Humanas 501070 68 2	Lógica Computacional 203042 102 3	Cálculo Diferencial e Integral 101051 102 3	Introdução à Investigação Científica 203055 68 2	Organização de Computadores 203016 136 4	Inglês Técnico 505027 102 3	Português Instrumental 506044 68 2	Estrutura de Programação 203015 136 4	
2ª SÉRIE 884 25	Planejamento em Informática 203021 102 3	Álgebra Linear e Programação Linear 101054 102 3	Cálculo Numérico 101053 102 3	Sistemas Operacionais 203041 136 4	Laboratório de Hardware 203020 102 3	Estrutura da Informação 203018 136 4	Programação de Computadores 203040 204 6		
3ª SÉRIE 884 26	Análise de Sistemas 203025 170 5	Gerenciamento de Centro de Processamento de Dados 203024 102 3	Estatística 101055 136 4	Legislação Aplicada à Informática 503017 68 2	Banco de Dados 203022 204 6	Inteligência Artificial 203026 102 3	Geometria Analítica 101050 100 3		
4ª SÉRIE 680 20	Projeto 203029 136 4	Estágio Supervisionado 203022 272 8	Computação Gráfica 203030 102 3	Teleprocessamento 203027 102 3	Análise de Algoritmos 203043 68 2				

Em vigor a partir de 02 de janeiro de 1995 (Res. C.A. nº 265/94 e alterado pela Res. CEPE nº 54/98)

Tabela 5 – Grade Curricular do Curso de Informática da UEPG [UEPG99], com destaque para as disciplinas em que o aluno está reprovado.

Descreveremos a seguir o comportamento do AGENTE-M para o exemplo proposto.

Primeiramente, o agente usuário, através de dois comandos em LOOM, informa o seu RA e ativa o AGENTE-M, conforme mostra a figura 41:

```
(SETQ ?RA (GET-INSTANCE 'RA971021129' )
(PERFORM (AGENTE-M ?R))
```

Figura 41 – Tela 1: Comandos para informar o RA e ativar o AGENTE-M.

O aluno será identificado pelo Agente Matriculador que fará uma validação do RA (figura 42). Como o número de registro é válido, o AGENTE-M assume o compromisso de

buscar dados do aluno referente a seu cadastro para mostrar ao agente usuário, e depois verifica a situação do aluno (figura 43). Os dados observados pelo agente matriculador, bem como suas decisões são sempre apresentadas ao agente usuário que acompanha o processo de sua matrícula (figura 44).

Através do conhecimento que possui e da observação do histórico do aluno, o AGENTE-M concluirá que o aluno está em regime de dependência e decidirá matriculá-lo na série seguinte, mas terá que matriculá-lo também nas disciplinas em que está reprovado.

Em um processo de interação com o mundo, o agente matriculador atualizará a base de dados, acrescentando ao histórico do aluno as disciplinas em que foi matriculado, que assumem a situação atual “em curso”. Finalmente o AGENTE-M mostra ao agente usuário a matrícula gerada e anuncia o término do processo de matrícula tornando-se então disponível para uma nova atuação (figura 45).

<p>=====</p> <p>SISTEMA DE MATRICULA DO CURSO DE INFORMÁTICA</p> <p>=====</p> <p>RA DO ALUNO A MATRICULAR: RA971020829</p> <p>RESULTADO DA VALIDAÇÃO DO RA: VÁLIDO</p>

Figura 42 - Tela 2: Identificação e validação do RA.

<p>=====</p> <p>SISTEMA DE MATRICULA DO CURSO DE INFORMÁTICA</p> <p>=====</p> <p>RA DO ALUNO A MATRICULAR: RA971020829 RA VÁLIDO</p> <p>=====</p> <p>DADOS DO ALUNO</p> <p>=====</p> <p>Nome: PAULO DE VASCOENLOS Data Nascimento: 20/10/1980 Série Atual: 1^ª Data da última matrícula: 04/02/1999 Situação Atual: APROVADO EM REGIME DE DEPENDÊNCIA</p>
--

Figura 43 - Tela 3: Dados do aluno e sua situação atual.

=====

SISTEMA DE MATRICULA DO CURSO DE INFORMÁTICA

=====

RA DO ALUNO A MATRICULAR: RA971020829
RA VÁLIDO

=====

HISTÓRICO DO ALUNO

=====

DISCIPLINA	CÓDIGO	SITUAÇÃO	ANO DE CURSO	TURMA
PSICOLOGIA REL. HUMANAS	501070	R	1999	A
LÓGICA COMPUTACIONAL	203042	A	1999	A
CÁLCULO DIFERENCIAL INTEGRAL	101051	R	1999	A
INTR.INVESTIGAÇÃO CIENTÍFICA	203056	A	1999	A
ORGANIZAÇÃO DE COMPUTADORES	203016	A	1999	B
INGLÊS TÉCNICO	505027	A	1999	A
PORTUGUÊS INSTRUMENTAL	506044	A	1999	A
ESTRUTURA DE PROGRAMAÇÃO	203015	A	1999	B

AGUARDE...A MATRÍCULA ESTÁ SENDO GERADA.

Figura 44 - Tela 4: Histórico escolar do aluno e aviso sobre a decisão de gerar matrícula.

=====

SISTEMA DE MATRICULA DO CURSO DE INFORMÁTICA

=====

RA DO ALUNO A MATRICULAR: RA971020829
RA VÁLIDO

=====

COMPOSIÇÃO DA MATRÍCULA GERADA

=====

SÉRIE EM QUE FOI MATRICULADO: 2^a

DISCIPLINA	CÓDIGO	SITUAÇÃO	ANO DE CURSO	TURMA
PLANEJAMENTO EM INFORMÁTICA	203021	EM CURSO	2000	A
ÁLGEBRA LINEAR E PROG.LINEAR	101054	EM CURSO	2000	A
CÁLCULO NUMÉRICO	101053	EM CURSO	2000	A
SISTEMAS OPERACIONAIS	203041	EM CURSO	2000	B
LABORATÓRIO DE HARDWARE	203020	EM CURSO	2000	B
ESTRUTURA DA INFORMAÇÃO	203018	EM CURSO	2000	B
PROGRAMAÇÃO DE COMPUTADORES	203040	EM CURSO	2000	B
PSICOLOGIA REL. HUMANAS	501070	EM CURSO	2000	A
CÁLCULO DIFERENCIAL INTEGRAL	101051	EM CURSO	2000	A

MATRÍCULA REALIZADA COM SUCESSO.
TÉRMINO DE EXECUÇÃO.

Figura 45 - Tela 5: Matrícula gerada para a 2^a série e disciplinas reprovadas.

6.3 Discussões

Com relação aos resultados obtidos com a implementação do AGENTE-M, verificamos que:

A linguagem LOOM permitiu a implementação do agente conforme projetado uma restrição:

- A versão do LOOM que usamos só faz inferência sobre o que já existe em sua base de conhecimento, comprometendo a validação de dados desconhecidos. Assim se o número de RA fornecido não existir na base o LOOM ao invés de continuar a execução do programa informando que o RA é inválido, indica forçosamente um erro que ocorre por não existir tal instância para o conceito RA do aluno. Em literatura específica [BRIL93] da linguagem vimos mencionado que versões futuras buscam sanar esse problema.

A operação do AGENTE-M é bastante simples, precisando apenas de uma informação (RA a matricular) fornecida pelo agente usuário para então assumir o controle da matrícula com total autonomia.

A versão de linguagem utilizada impossibilitou a implementação do AGENTE-M em modo gráfico, o que resultou no uso de telas simples para estabelecer a interação com o agente usuário.

Concluimos, através dos testes realizados, que o agente matriculador raciocina e atua corretamente nas mais diversas situações em que o aluno possa se encontrar.

7 Conclusões

Este trabalho apresenta o desenvolvimento de um agente deliberativo, denominado AGENTE-M, que realiza o processo de matrícula de alunos em uma universidade. Em sua atuação interage com o aluno que deseja ser matriculado (agente usuário) e com a base de dados que contém as informações necessárias (mundo).

Esse agente utiliza as técnicas da IAD e SMA que correspondem ao estudo da solução de problemas através da distribuição de conhecimento entre diversas entidades. Suas metas essenciais são desenvolver mecanismos e métodos que permitam agentes interagir como humanos, e entender a interação entre entidades inteligentes, quer sejam elas computacionais, humanas ou ambos. Salientamos que essas metas estão longe de serem atingidas, existindo muitos desafios a serem vencidos.

O AGENTE-M foi modelado com base na arquitetura BDI proposta pelo DESIRE e implementado na linguagem de representação de conhecimento LOOM.

Projetamos o AGENTE-M utilizando as características do DESIRE, mas não o seu ambiente de desenvolvimento. Concluímos que suas características de modelagem de agentes são viáveis de serem utilizadas mesmo não sendo desenvolvidas em seu próprio ambiente.

O AGENTE-M resultou numa arquitetura composta por quatro componentes principais e oito subcomponentes, com raciocínio explícito sobre o próprio agente, sobre o mundo e sobre o agente usuário.

Concluímos que o projeto do agente em componentes específicos facilita a compreensão e manutenção do seu conhecimento. O uso de componentes específicos de interação com o mundo e agentes contribuiu para tornar explícito o raciocínio do AGENTE-M sobre ele mesmo, sobre o mundo e sobre outros agentes.

A associação de estados mentais ao comportamento do agente é uma tarefa complexa. Adotamos então uma seqüência lógica de raciocínio considerando que desejamos algo com base em crenças. Esses desejos geram intenções que levam a ações que devem ser executadas. As questões de conflitos de raciocínio não foram consideradas em nosso trabalho.

Os estados mentais representados foram: crenças, desejos e intenções. Incorporamos ao agente a capacidade de planejamento que define uma estratégia, ou seja, o conjunto de ações a serem executadas pelo agente para realizar determinado desejo.

As características da linguagem LOOM permitiram a implementação do AGENTE-M de acordo com seu projeto, no sentido em que conseguimos criar uma estrutura distribuída

em componentes, e implementamos as ontologias para conceitualizar os estados mentais do agente e as demais informações pertencentes a sua base de conhecimento.

A dificuldade que tivemos em relação à linguagem LOOM foi o fato não permitir inferência sobre dados que ainda não pertencem a base de conhecimento. Isso dificultou a interação entre o agente matriculador e o usuário, uma vez que as informações trocadas deviam obrigatoriamente pertencer à base previamente.

A operação do AGENTE-M é bastante simples e através dos testes realizados, verificamos que o agente matriculador raciocina e atua corretamente nas diversas situações em que o aluno possa se encontrar.

Em comparação ao agente desenvolvido anteriormente dentro do SISMAT, consideramos que o AGENTE-M possui um poder maior de representação de conhecimento, e que o raciocínio do agente em função de seus estados mentais é mais compreensível. Porém, o tratamento da comunicação entre agentes é melhor no SISMAT e mais fraco no AGENTE-M, muito embora a linguagem KQML utilizada no LALO tenha apresentado limitações. A dificuldade de comunicação no AGENTE-M se justifica pelas limitações da versão de teste da linguagem Common Lisp que utilizamos. Se pudéssemos ter utilizado uma versão completa da ferramenta acreditamos que a comunicação entre os agentes teria sido melhor.

Finalmente, concluímos que a associação de estados mentais ao agente estabelece uma íntima relação entre seu raciocínio e o pensamento humano, uma vez que ao desenvolvermos o AGENTE-M tivemos como maior preocupação representar como uma pessoa agiria e pensaria em relação a tarefa de matricular um aluno. Destacamos porém, que essa representação consiste de metáforas que simulam parcialmente o comportamento humano; uma representação perfeita está longe de ser conquistada.

Como trabalho futuro, propomos o estudo de inserção do AGENTE-M em diferentes ambientes. Consideramos importante verificar em relação ao agente, que tipo de conhecimento é mais afetado ao deparar-se com um novo ambiente, bem como a necessidade ou não de alterações em sua arquitetura. Assim poderemos avaliar o quanto a estrutura que adotamos para o agente realmente facilita sua adaptação a outros meios.

Referências Bibliográficas

- [BOND88] BOND, A. H, Gasser, L.; *An Analysis of Problems and Research in DAI*. In: Distributed Artificial Intelligence. San Mateo: Morgan Kaufmann, 1988.
- [BRA95] BRAZIER, F. M. T., TREUR, J., WLINGAARDS, N. J. E., e WILLEMS, M.; *Formal specification of hierarchically (de)composed tasks*. In B.R. Gaines and M.A. Musen (Eds.). Proceedings of the 9th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, KAW '95, 1995, pp. 25/1-25/20.
- [BRA96] BRAZIER, F.M.T., JONKER, C.M. e TREUR, J.; *Modelling Project Coordination in a Multi-Agent Framework*. Fifth Workshops on Enabling Technology for Collaborative Enterprises, WET ICE'96, IEEE Computer Society Press, 1996, pp. 148-155.
- [BRA96a] BRAZIER, F.M.T., DUNIN-KEPLICZ, B. JENNINGS, N. e TREUR, J.; *Modelling Distributed Industrial Processes in a Multi-Agent Framework*. In: Cooperative Knowledge Processing: The Key Technology for Intelligent Organisations, S. Kirn and G. O'Hare (eds.), Springer Verlag, Computer Supported Cooperative Work Series, 1996, pp. 212-229.
- [BRA96b] BRAZIER, F.M.T., DUNIN-KEPLICZ, B., TREUR, J., e VERBRUGGE, R.; *Beliefs, Intentions and DESIRE*. Proc. of the 10th Banff Knowledge Acquisition for Knowledge-based Systems workshop (KAW'96), Gaines, B.R., and Musen, M.A. (Eds.), SRDG Publications, Department of Computer Science, University of Calgary, Canada, 1996, pp. 22/1-22/15.
- [BRA97] BRAZIER, F.M.T., ECK, P.^oT. van e TREUR, J.; *Modelling Competitive Cooperation of Agents in a Compositional Multi-Agent Framework*. In: Plaza, E. and Benjamins, R. (eds.), Knowledge Acquisition, Modelling and Management, Proc. of the 10th European Knowledge Acquisition Workshop,

EKAW'97, Lecture Notes in AI, volume 1319, Springer-Verlag, Berlin, 1997, pp. 317-322.

- [BRA97a] BRAZIER, F.M.T., DUNIN KEPLICKZ, B., JENNINGS, N. e TREUR, J.; *DESIRE: Modelling Multi-Agent Systems in a Compositional Formal Framework*. International Journal of Cooperative Information Systems, vol. 6, Special Issue on Formal Methods in Cooperative Information Systems: Multi-Agent Systems, (M. Huhns and M. Singh, eds.), 1997, pp.67-94.
- [BRA97b] BRAZIER, F.M.T., ECK, P.A.T. van, e TREUR, J.; *Modelling a Society of Simple Agents: From Conceptual Specification to Experimentation*. In: Conte, R., Hegselmann, R. and Terna, P. (eds.) *Simulating Social Phenomena, Proc. of the International Conference on Computer Simulations and Social Sciences, ICCS&SS'97, Lecture Notes in Economics and Mathematical Systems, Vol. 456*, Springer-Verlag, Berlin, 1997, pp. 103-107.
- [BRA98] BRAZIER, F.M.T., JONKER, C.M., TREUR, J., e WIJNGAARDS, N.J.E.; *On the Role of Abilities of Agents in Redesign*. Em: B.R. Gaines, M.A. Musen (eds.), *Proceedings of the 11th Banff Knowledge Acquisition for Knowledge-based Systems workshop, KAW'98, Calgary: SRDG Publications, Department of Computer Science, University of Calgary*, pp. 16, 1998.
- [BRA98a] BRAZIER, F. M.T. et al.; *Agents Negotiating for Load Balancing of Electricity Use*. Third International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, PAAM'98, The Practical Application Company Ltd, 1998, pp. 551-554.
- [BRA98b] BRAZIER, F.M.T. et al.; *Modelling Internal Dynamic Behaviour of BDI Agents*. Third International Workshop on Formal Models of Agents, MODELAGE'97. Lecture Notes in AI, Springer Verlag, 1997. In press, 1998.
- [BRA99] BRAZIER, F.M.T., e TREUR, J.; *Compositional Modelling of Reflective Agents*. International Journal of Human-Computer Studies. In press, 1999.

- [BRA99a] BRAZIER, F.M.T., JONKER, C.M., JUNGEN, F.J., and TREUR, J.; *Distributed Scheduling to Support a Call Centre: a Cooperative Approach*. Applied Artificial Intelligence Journal, vol. 13, pp. 65-90, 1999.
- [BRIL93] Brill, David.; *LOOM Reference Manual Version 2.0*. University of Southern California, 1993.
- [BRO91] BROOKS, R. A.; *Intelligence without representation*. Artificial Intelligence, 47(1):139-159, 1991.
- [BRT87] BRATMAN, M. E.; *Intentions, Plans, and Practical Reason*. Harvard University Press, Cambridge, MA, 1987.
- [CAS95] CASTELFRANCHI, C.; *Commitments: From individual intentions to groups and organizations*. In: V. Lesser (Ed.), Proc. Of the First International Conference on Multi-Agent Systems, ICMAS-95, MIT Press, Cambridge, MA, pp 41-48, 1995.
- [COH79] COHEN, P. R. e PERRAULT, C. R.; *Elements of a plan based theory of speech acts*. Cognitive Science, 3:177-212, 1979.
- [COH90] COHEN, P.R. e LEVESQUE, H. J.; *Intention is choice with commitment*. Artificial Intelligence 42, pp 213-261. 1990.
- [COR99] CORREA, M.; *A Arquitetura de Diálogos entre Agentes Cognitivos Distribuídos*. Rio de Janeiro: UFRJ, 1994. In: GIRAFFA, L.M.M.; Uma arquitetura de tutor utilizando estados mentais. Tese de Doutorado, Universidade Federal do Rio Grande do Sul, 1999.
- [DAV96] DAVIDSSON, P.; *Autonomous Agents and the Concept of Concepts*. Doctoral Dissertation for the degree of PhD in Computer Science. Lund University, 1996.

- [DEM90] DEMAZEAU, Y. e MULLER, J.; *Decentralized Artificial Intelligence*, First European Workshop on Modeling Autonomous Agents in a Multi-Agent World, Proceedings, Cambridge, North-Holland, 1990.
- [DEM92] DEMAZEAU, Y. e SICHMAN, J.; *When can knowledge based systems be called agents*. XII Congresso da Sociedade Brasileira de Computação - IX simpósio Brasileiro de Inteligência Artificial. Pp. 172-185, Rio de Janeiro, 1992.
- [DUN96] DUNIN-KEPLICZ, B. e VERBRUGGE, R.; *Collective commitments*. In: Proceedings of the Second International Conference on Multiagent Systems, ICMAS-96, AAAI Press, Menlo Park, CA, 1996, pp-56-63.
- [FEN98] FENSEL, D. et al.; *The Knowledge Acquisition and Representation Language, KARL*. IEEE, July/August 1998.
- [FER91] FERBER, J. e GASSER, L.; *Intelligence Artificielle Distribuée*. In: International Workshop on Expert Systems & Their Applications, 11, France, 1991.
- [FERG95] FERGUSON, I. A.; *Integrated control and coordinated behaviour: A case for agent models*. In M. Wooldridge and N. R. Jennings, editors, *Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890)*, pages 203-218. Springer-Verlag: Berlin, Germany, January 1995.
- [FRAN99] Franz, Corporation.; *Allegro CL 4.0. Trial version for LINUX*. Web: <http://www.franz.com>, 1999.
- [GEN92] GENESERETH, M. e FIKES, R.; *Knowledge Interchange Format, Version 3.0 Reference Manual*. Technical report, Computer Science Department, Stanford University, 1992.

- [GEN97] GENESERETH, M., SING, N. e SYED, M.; *A Distributed and Anonymous Knowledge Sharing Approach to Software Interoperation*. ACM Symposium on Applied Computing, February, 1997.
- [GIR97] GIRAFFA, L.M.M.; *Seleção e adoção de estratégias de ensino em Sistemas Tutores Inteligentes*. CPGCC/UFRGS, 1997 (EQ-10).
- [GIR99] GIRAFFA, L.M.M.; *Uma arquitetura de tutor utilizando estados mentais*. Tese de Doutorado. Universidade Federal do Rio Grande do Sul, 1999.
- [GRU92] GRUBER, T.R.; *Ontolingua: A Mechanism to Support Portable Ontologies*. Knowledge Systems Laboratory Stanford University, 1992.
- [GRU93] GRUBER, T.; *A Translation Approach to Portable Ontology Specifications*. Technical Report KSL 92-71, Stanford University, Knowledge Systems Laboratory, 1993.
- [GUA97] GUARINO, N.; *Understanding, Building and Using Ontologies*. International Journal of Human-Computer Studies, 1997.
- [HAY91] HAYES-ROTH, B.; *A integrated architecture for intelligent agents*. SIGART, 2(4):79-81, 1991.
- [HAYE99] HAYES, C. C.; *Agentes in a Nutshell - A Very Brief Introduction*. IEEE Transactions on Knowledge and Data Engineering – IEEE Computer Society, Vol.11: 127-132, 1999.
- [HUH98] HUHNS, M.N. e SINGH, M.P.; *Agents and multiagent systems: Themes, approaches, and challenges*. In M.N. Huhns and M.P. Singh, editors, *Readings in Agents*, pages 1-23. Morgan Kaufmann, San Francisco, CA, 1998.
- [JEN98] JENNINGS, N.R., SYCARA, K., e WOOLDRIDGE, M.; *A roadmap of agent research and development*. *Autonomous Agents and Systems*, 1:7-38, 1998.

- [JON98] JONKER, C.M., TREUR, J.; *Compositional Design and Maintenance o Broker Agents*. IFIP'98 Conference IT&KNOWS'98, J. Cuenca (ed.), Vhapman and Hall, 1998.
- [LAB97] LABROU, Y.; *Semantics for an Agent Communication Language*. Phd thesis, University of Maryland Graduate School, Baltimore, Maryland 21228-5398, 1997.
- [LALO97] LALO (*Language of Agents Logiciel Object*), CRIM – Centre de Recherche Informatique de Montreal, Web: <http://www.crim.ca/sbc/english/lalo>, 1997.
- [LES99] LESSER, V.; *Cooperative Multiagent Systems: A Personal View of the State of the Art*. IEEE, January/February 1999.
- [LOOM91] LOOM Tutorial. ISX Corporation, University of Southern California, 1991.
- [LOOM91a] LOOM Users Guide. ISX Corporation, University of Southern California, 1991.
- [LUCK95] LUCK, M. e DÍNVERNO, M.; *A Formal Framework for Agency and Autonomy*. In Proceedings First International Conference on Multi-Agent Systems. San Francisco, Califórnia, MIT Press, 1995.
- [MAT00] MATHIAS, I.; *SISMAT – Sistema de Matrícula Inteligente*. Dissertação de mestrado, Universidade Federal do Paraná, Brasil, 2000.
- [MAT00a] MATHIAS, I.; *SISMAT – Ferramenta de Desenvolvimento de Agentes – Relatório Técnico*. Departamento de Informática, Universidade Federal do Paraná, Brasil, 2000.

- [MIL82] MILLER, M.L.; *A Structured Planning and Debugging Environment for Elementary Programming*. In: SLEEMAN, D.H.; BROWN, J.S. (Eds.) *Intelligent Tutoring Systems*. New York, Academic press, 1982.
- [MIN85] MINSKY, M.; *The Society of Mind*. New York, Simon & Schuster, Inc., 1985.
- [MOR97] MÓRA, M.C., Lopes, J.G., Coelho, J.G., Viccari, R.; *Modelling dynamic aspects of intentions*. In E.Costa, editor, 8th Portuguese Conference on Artificial Intelligence. Springer-Verlag, 1997.
- [NEW90] NEWELL, A.; *Unified Theories of Cognition*. Harvard University Press, Cambridge, MA, 1990.
- [NIL98] NILSSON, N.J.; *Artificial Intelligence. A New Synthesis*. Morgan Kaufmann Publ., San Francisco, CA, 1998.
- [OLI96] OLIVEIRA, F.; *Inteligência Artificial Distribuída*. In: IV Escola Regional de Informática, SBC, SC, 1996.
- [OLI97] OLIVER, M.D. e GENESERETH, M.R.; *Infomaster – An Information Integration Tool*. Proceedings of the International Workshop on Intelligent Information Integration, Freiburg, Germany, September, 1997.
- [POO98] POOLE, D. MACHWORTH, A. e GOEBEL, R.; *Computational Intelligence*. Oxford University Press, New York, 1998.
- [POZO95] POZO, A., LIMA, W., BARRETO, J. e FERNANDES, J.; *EPIIS: An Epilepsy Intelligent Instruction System Using Multi-Agents Society Paradigm*. In: 38th MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS. R.J., Brasil, Agosto 13-16, 1995.

- [POZO96] POZO, A.T.R.; *Um Sistema de Ensino Inteligente via Sociedade Multiagentes aplicado ao Diagnóstico de Epilepsia*. Tese de Doutorado, Universidade Federal de Santa Catarina, 1996.
- [PRO99] Web: <http://www.logic.stanford.edu/sharing/programs/prologic> , 1999.
- [RAO95] RAO, A. S. e GEORGEFF, M. P.; *An abstract architecture for rational agents*. In C. Rich and B. Nebel, editors, *Proceedings of Knowledge Representation and Reasoning*, (439-449), 1995.
- [RICH93] RICH, E.; *Inteligência Artificial*. São Paulo, Makron Books, 1993.
- [RUS95] RUSSELL, S.J. e NORVIG, P.; *Artificial intelligence: A modern approach*. Prentice-Hall, Englewood Cliffs, New Jersey. 1995.
- [SHO93] SHOHAM, Y.; *Agent-Oriented Programming*. Artificial Intelligence, Elsevier, Amsterdam, v.60, n.1, (51-92), 1993.
- [SHO93a] SHOHAM, Y.; *Artificial Intelligence*. Elsevier. 1993.
- [SIN92] SINGH, M.P.; *On the semantics of protocols among distributed intelligent systems*. In *Proceedings of IEEE International Phoenix Conference on Computers and Communications*, 1992.
- [SMI95] SMITH IRA, A. e COHEN, P.R.; *Toward a semantics for a speech act based agent communication language*. In *Proceedings of the ACM CIKM Intelligent Information Agents Workshop*, 1995.
- [SPI00] SPINOSA, E. e EMER, M.; *Lógica Descritiva*. Trabalho disciplinar. UFPR, 2000.
- [UEPG99] UEPG.; *Catálogo Geral 1999*. Universidade Estadual de Ponta Grossa. Pró-Reitoria de Graduação, Divisão de Ensino, 1999.

- [VAN98] VAN der VET, P. e MARS, N.; *Bottom-Up Construction of Ontologies*. IEEE, July/August 1998.
- [WEI99] WEISS, G.; *Multiagent systems: a modern approach to distributed artificial intelligence*. The MIT Press, Cambridge, 1999.
- [WOO95] WOOLDRIDGE, M. e JENNINGS, N.R.; *Intelligent agents: Theory and practice*. The Knowledge Engineering Review, 10(2):115-152, 1995.
- [WOO97] WOOLDRIDGE, M.; *Agente-based software engineering*. IEE Transactions on Software Engineering, 144(1):26-37, February 1997.