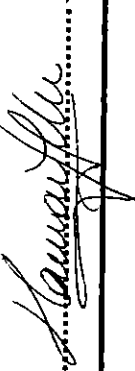


Uma arquitetura para ambientes de
ensino-aprendizagem utilizando o modelo dos
Sistemas Tutores Inteligentes e agentes

Neli Regina da Silveira Almeida Prado

Serviço de Pós-Graduação EESC/USP
EXEMPLAR REVISADO
Data de entrada no Serviço: 31.10.2002
Ass: 

Dissertação apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Mestre em Engenharia de Produção.

ORIENTADOR: Prof. Dr. Edson Walmir Cazarini

São Carlos

2002



DEDALUS - Acervo - EESC



31100037102

Class.	TESE - EESC
Cutt.	4604
Tombo	T026102
Sysno	1276187

31100037102

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca - EESC/USP

P896a

Prado, Neli Regina da Silveira Almeida

Uma arquitetura para ambientes de ensino-
aprendizagem utilizando o modelo dos sistemas tutores
inteligentes e agentes / Neli Regina da Silveira
Almeida Prado. -- São Carlos, 2002.

Dissertação (Mestrado) -- Escola de Engenharia de
São Carlos-Universidade de São Paulo, 2002.

Área : Engenharia de Produção.

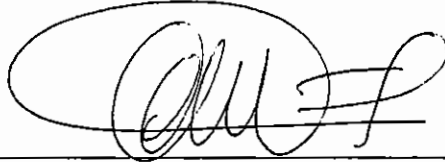
Orientador: Prof. Dr. Edson Walmir Cazarini.

1. Sistemas tutores inteligentes. 2. Inteligência
artificial distribuída. 3. Agentes inteligentes. I.
Título.

FOLHA DE JULGAMENTO

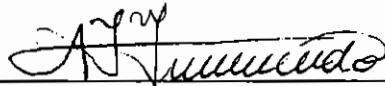
Candidata: Bacharela **NELI REGINA DA SILVEIRA ALMEIDA PRADO**

Dissertação defendida e julgada em 03-10-2002 perante a Comissão Julgadora:



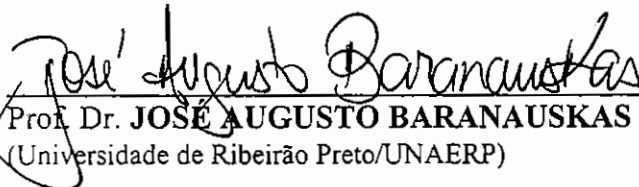
Prof. Dr. **EDSON WALMIR CAZARINI (Orientador)**
(Escola de Engenharia de São Carlos/USP)

Aprovada



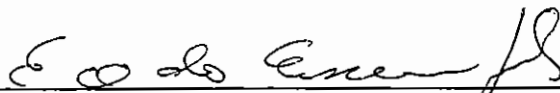
Prof. Dr. **ANTONIO FERNANDO TRAINA**
(Universidade de Ribeirão Preto/UNAERP)

Aprovada

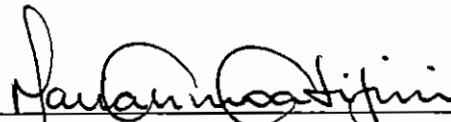


Prof. Dr. **JOSE AUGUSTO BARANAUSKAS**
(Universidade de Ribeirão Preto/UNAERP)

Aprovada



Prof. Doutor **EDMUNDO ESCRIVÃO FILHO**
Coordenador do Programa de Pós-Graduação
em Engenharia de Produção



Profa. Assoc. **MARIA DO CARMO CALIJURI**
Presidente da Comissão de Pós-Graduação da EESC

Ao José e aos meus pais.

Agradecimentos

Ao departamento de Engenharia da Produção da EESC – USP, pela oportunidade da realização deste trabalho.

Ao Professor Edson Walmir Cazarini, pela orientação e sugestão de tema tão fascinante.

Aos professores Antonio Fernando Traina e Fernando Santos que fizeram parte da banca de qualificação e acrescentaram grande contribuição ao trabalho.

À banca de defesa pela disposição e paciência na leitura do texto.

Ao Pacheco pelo eterno incentivo, pela ajuda e paciência em todos os momentos.

À minha família por entender e respeitar meus momentos de ausência.

À Rai por seu suporte à casa e ao José, sem o qual não teria tranquilidade, e mesmo, possibilidade de realizar este trabalho.

Aos meus amigos que de alguma forma — apoiando, incentivando ou simplesmente me agüentando — contribuíram para a realização deste trabalho.

SUMÁRIO

LISTA DE FIGURAS.....	V
LISTA DE TABELAS.....	VI
LISTA DE ABREVIATURAS E SIGLAS.....	VII
RESUMO.....	IX
<i>ABSTRACT</i>	X
1 INTRODUÇÃO.....	1
1.1 Contexto do trabalho.....	2
1.2 Problemas.....	3
1.3 Justificativa.....	4
1.4 Objetivos gerais do trabalho.....	5
1.4.1 Objetivos específicos.....	5
1.5 Delimitação da pesquisa.....	6
1.6 Métodos de pesquisa.....	6
1.6.1 Fases da pesquisa.....	7
1.7 Estrutura do trabalho.....	8
2 O ENSINO-APRENDIZAGEM ASSISTIDO POR COMPUTADORES.....	10
2.1 Evolução dos ambientes de ensino-aprendizagem.....	11
2.2 Considerações finais.....	17
3 OS SISTEMAS TUTORES INTELIGENTES.....	18
3.1 Componentes de um STI.....	19
3.1.1 Módulo de domínio.....	20
3.1.2 Modelo do estudante.....	22
3.1.3 Módulo tutorial.....	26
3.1.4 Módulo de interface.....	29
3.2 Características fundamentais de um STI.....	30

4.4.1 Resolução Distribuída de Problemas <i>versus</i> Sistemas Multi-Agente.....	64
4.6 Considerações finais.....	66
5 PROJETANDO AGENTES.....	67
5.1 Arquiteturas de agentes.....	67
5.1.1 Arquiteturas baseadas em lógica.....	68
5.1.2 Arquiteturas reativas.....	69
5.1.3 Arquiteturas BDI – <i>Belief-Desire-Intention</i>	69
5.1.4 Arquiteturas em camadas.....	70
5.2 Linguagens de comunicação entre agentes.....	73
5.2.1 KQML — Uma linguagem padrão de comunicação de agentes.....	74
5.2.2 Linguagens de comunicação de agentes <i>ad hoc</i>	77
5.3 A construção de agentes.....	78
5.3.1 Linguagens de programação de agentes.....	79
5.4 Considerações finais.....	83
6 ARQUITETURA PROPOSTA.....	84
6.1 As bases de conhecimento do domínio.....	87
6.2 As bases de informações do estudante.....	88
6.2.1 Base de histórico.....	89
6.2.2 Base de comportamento.....	90
6.2.3 Base de aprendizado.....	91
6.2.4 Novas bases de informações a respeito do estudante.....	91
6.3 Módulo de interface.....	92
6.3.1 Módulo de cadastro.....	95
6.3.2 Módulo de operação.....	96
6.3.3 Módulo de controle de atividades.....	96
6.4 Agentes do domínio.....	97
6.5 Agentes tutores.....	100
6.5.1 Agente de planejamento de ensino.....	100
6.5.2 Agente de controle tutor.....	101
6.5.3 Agente de avaliação de desempenho.....	102
6.6 Os agentes de geração de perfil do estudante.....	105
6.6.1 Agentes de classificação.....	105
6.6.2 Agente gerenciador de perfil.....	106

6.7 A comunicação entre os agentes	108
6.8 Considerações finais.....	108
7 CONCLUSÃO.....	111
7.1 Propostas para trabalhos futuros.....	112
BIBLIOGRAFIA.....	114

LISTA DE FIGURAS

FIGURA 1 — Modelo Geral de um STI.....	20
FIGURA 2 — Elementos que compõem o ambiente SIATE.....	33
FIGURA 3 — Arquitetura do sistema FLUTE.....	37
FIGURA 4 — Estrutura pedagógica do domínio.....	38
FIGURA 5 — Representação da arquitetura.....	42
FIGURA 6 — Diagrama esquemático de uma arquitetura BDI genérica.....	70
FIGURA 7 — Fluxos de informação e controle em arquiteturas em camada.....	71
FIGURA 8 — TouringMachines: uma arquitetura de agentes híbrida.....	72
FIGURA 9 — Uma visão abstrata da linguagem KQML.....	76
FIGURA 10 — Componentes da arquitetura.....	85
FIGURA 11 — Componentes do Módulo de Interface.....	95
FIGURA 12 — As interações dos Agentes de Domínio	98
FIGURA 13 — As interações do Agente de Planejamento de Ensino.....	101
FIGURA 14 — As interações do Agente de Controle Tutor.....	102
FIGURA 15 — Interações do Agente de Avaliação e Desempenho.....	103
FIGURA 16 — As interações dos Agentes de Geração de Perfil do estudante.....	106
FIGURA 17 — Representação da arquitetura proposta.....	109

LISTA DE TABELAS

QUADRO 1 — Características dos ambientes de ensino-aprendizagem.....	17
QUADRO 2 — Características dos tipos de representação de conhecimento.....	22
QUADRO 3 — <i>Performatives</i> KQML.....	75
QUADRO 4 — Componentes do Módulo de Interface.....	97
QUADRO 5 — Agentes de Domínio.....	99
QUADRO 6 — Agentes Tutores.....	104
QUADRO 7 — Agentes de Geração de Perfil do estudante.....	107

LISTA DE ABREVIATURAS E SIGLAS

AAD	– Agente de Avaliação e Desempenho
AC	– Agente de Classificação
ACL	– Agent Communication Language
ACT	– Agente de Controle Tutor
AGP	– Agente Gerenciador de Perfil
ALL	– Agent Level Language
APE	– Agente de Planejamento de Ensino
BDI	– Belief-Desire-Intention
CAI	– Computer-Aided Instruction
CBT	– Computer-Based Training
CFG	– Context Free Grammars
CGI	– Common Gateway Interface
CPU	– Central Processing Unit
CSCL	– Computer-Supported Collaborative Learning
CSG	– Context Sensitive Grammars
DCFL	– Deterministic Context-Free Language
DPS	– Distributed Problem Solving
EAD	– Educação à Distância
FLUTE	– Formal Languages and Automata Environment
GET-BITS	– Generic Tools for Building ITS
HTML	– Hypertext Markup Language

IA	– Inteligência Artificial
IAD	– Inteligência Artificial Distribuída
ICAI	– Intelligent Computer Aided Instruction
ILE	– Intelligent Learning Environment
KQML	– Knowledge Query and Manipulation Language
MAS	– Multi-Agent System
MC	– Módulo de Cadastro
MCA	– Módulo de Controle de Atividades
MO	– Módulo de Operação
POA	– Programação Orientada a Agentes
POO	– Programação Orientada a Objetos
RBC	– Raciocínio Baseado em Casos
SBC	– Sistemas Baseados em Casos
SIATE	– Sistema Inteligente de Apoio a Treinamento e Ensino
STI	– Sistemas Tutores Inteligentes
WAN	– Wide Area Network
Web	– Termo usado como referencia ao ambiente Word Wide Web
WWW	– World Wide Web

Resumo

PRADO, N.R.S.A. (2002). *Uma arquitetura para ambientes de ensino-aprendizagem utilizando o modelo dos Sistemas Tutores Inteligentes e agentes*. São Carlos, 2002. 120p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo.

O ensino auxiliado por computador é uma alternativa atrativa de apoio aos estudantes que aprendem acompanhados por tutores humanos e também para os estudantes que aprendem sozinhos. Das diversas ferramentas computacionais de apoio ao ensino, destacam-se os Sistemas Tutores Inteligentes. Pesquisas recentes tem demonstrado a validade de utilizar conceitos e técnicas de inteligência artificial para a construção de ambientes de ensino-aprendizagem. Este trabalho propõe o uso de agentes inteligentes no desenvolvimento de ambientes de ensino-aprendizagem. Parte, a princípio, da conceituação de ambientes de ensino-aprendizagem e de agentes inteligentes. O estudo de agentes inteligentes envolve um variado número de propostas relativas a teorias de agentes, que dizem respeito às especificações das propriedades que os agentes devem possuir, a classificação destes agentes e as formas nas quais os mesmos podem ser organizados. Em seguida, apresenta ferramentas computacionais para apoiar o desenvolvimento de agentes. E, finalmente propõe uma arquitetura que é fundamentada no modelo dos sistemas tutores inteligentes, cujos componentes são implementados através da tecnologia de agentes e organizados a partir de conceitos da inteligência artificial distribuída. A arquitetura proposta busca trabalhar com aspectos relacionados a modularidade, flexibilidade e possibilidades de expansão de tais ambientes.

Palavras-chaves: Sistemas Tutores Inteligentes, Inteligência Artificial Distribuída, Agentes Inteligentes.

Abstract

PRADO, N.R.S.A. (2002). *An Architecture for learning environments based on Intelligent Tutoring Systems' Model and Agents*. São Carlos, 2002. 119p. MSc Dissertation - Escola de Engenharia de São Carlos, Universidade de São Paulo.

Computer-aided teaching is an attractive choice for supporting both tutor assisted and self-learning students. Among the various teaching tools, are the Intelligent Tutoring Systems. Recent research has pointed out the benefits of employing artificial intelligence concepts and techniques for building learning environments. This work proposes the usage of intelligent agents on the development of learning environments. Its starting point is the conceptualisation of learning environments and intelligent agents. The study of intelligent agents encompasses a varied number of proposals with regard to Agent's theories, which deal with the properties' specifications for those agents; their classification and the ways by which they can be organised. Next, the work presents computer tools for supporting the development of agents. Finally, it proposes an architecture based on intelligent tutor systems, whose components are implemented upon the agent's technology and organised from the concepts of distributed artificial intelligence. The proposed architecture considers aspects related to modularity, flexibility and growing potential of those environments.

Key words: intelligent tutoring systems, distributed artificial intelligence, intelligent agents.

1 INTRODUÇÃO

A rápida evolução das tecnologias de informação e comunicação, e a emergência de uma sociedade da informação criaram numerosas novas oportunidades para a melhoria da qualidade da educação. A importância do uso dos computadores no processo de ensino cresceu exponencialmente, seja como ferramenta de apoio ao ensino presencial ou no ensino à distância (EAD)¹.

“O surgimento de novas tecnologias de treinamento e educação, somadas aos requisitos de aprendizagem exigidos pela dinâmica da sociedade atual, faz do ensino à distância uma real necessidade. O ponto principal é a possibilidade de expandir a sala de aula para um universo maior, permitindo educar pessoas em qualquer lugar e a qualquer hora. Assim sendo, é clara a necessidade de novas soluções que permitam realmente um ensino à distância” (CASAS, 1999, s/n.).

A internet é uma das ferramentas que desempenha um importante papel no ensino à distância, uma vez que facilita o acesso a enormes repositórios de informação, materiais e serviços de aprendizagem. A educação a distância promovida através da internet permite, ainda, uma maior flexibilidade de acesso à informação através da exploração de hiperdocumentos, maior autonomia do estudante, possibilitando ao mesmo ter maior controle sobre a construção do seu conhecimento, além de oferecer novas oportunidades aos participantes do processo de ensino-aprendizagem.

Para promover o processo de ensino-aprendizagem à distância de maneira eficaz, obtendo vantagens significativas das oportunidades criadas pelas tecnologias é necessário entender as características intrínsecas às tecnologias disponíveis e como elas podem ser combinadas. Além de, segundo SOARES et al. (2001), associar estas tecnologias às melhores estratégias pedagógicas.

¹ Dentre os objetivos de se educar à distância está o interesse em levar o conhecimento a todas as pessoas que, seja pela distância, pela incompatibilidade de horário ou dificuldades financeiras, são impossibilitadas de participarem de cursos convencionais realizados em locais apropriados.

Tecnologias computacionais como hipermídia, redes, bancos de dados, Inteligência Artificial etc. têm desempenhado papel fundamental na implementação de ambientes de ensino-aprendizagem que proporcionam treinamento e ensino à distância.

Os assuntos abordados nesta pesquisa estão especificamente relacionados à investigação de conceitos envolvendo ambientes de ensino-aprendizagem e a Inteligência Artificial, mais especificamente os agentes inteligentes.

A principal contribuição deste trabalho radica na proposta de uma arquitetura para ambientes de ensino-aprendizagem, que procura através do uso da tecnologia de agentes inteligentes promover melhorias relacionadas a aspectos como a flexibilidade, modularidade e mesmo, a efetividade dos ambientes de ensino-aprendizagem no processo de ensino.

Nas seções a seguir é apresentado o contexto no qual está inserida esta pesquisa, os problemas identificados em ambientes de ensino-aprendizagem, a justificativa para o desenvolvimento deste trabalho, a delimitação do escopo da pesquisa e os objetivos alcançados com a mesma.

1.1 CONTEXTO DO TRABALHO

Uma das etapas do projeto de ambientes de ensino-aprendizagem é a definição de um modelo de arquitetura sobre a qual o ambiente é desenvolvido. A arquitetura define como o ambiente está estruturado, que elementos o compõe e como ocorre a interação entre estes componentes.

A maioria dos ambientes de ensino-aprendizagem pesquisados se baseiam no modelo dos Sistemas Tutores Inteligentes, descrito no capítulo 3, o qual é composto por quatro módulos principais, a saber:

- o módulo do domínio que compreende os fatos e as regras de um domínio específico a ser ensinado;
- o módulo tutor que cria e regula as interações instrucionais com o estudante;

- o modelo do estudante que representa o conhecimento e as habilidades cognitivas do estudante;
- o módulo de interface que controla a interação entre o estudante e o sistema.

Ambientes de ensino-aprendizagem compostos por módulos, como nos Sistemas Tutores Inteligentes, são projetados e em cada um dos módulos são aplicadas várias técnicas de Inteligência Artificial para melhorar sua performance.

O projeto e desenvolvimento de ambientes de ensino-aprendizagem através do uso de técnicas de Inteligência Artificial têm se constituído, há bastante tempo, em objeto de maior investigação por parte dos pesquisadores e busca encontrar novas formas de tornar estes ambientes mais inteligentes e melhorar sua capacidade de ensinar.

Este trabalho está inserido no contexto da investigação da utilização de técnicas de Inteligência Artificial no projeto de ambientes de ensino-aprendizagem. Considerando o modelo de Sistemas Tutores Inteligentes como a base da pesquisa e procurando melhorar a funcionalidade de seus componentes através do uso de técnicas de Inteligência Artificial.

1.2 PROBLEMAS

Pesquisadores² apontam algumas características, inerentes ao projeto, que os Sistemas Tutores Inteligentes devem possuir. Entre elas se destacam:

- a flexibilidade para mudanças, evoluções e atualizações na base de conhecimento ou no conjunto de procedimentos de treinamento, que possam suportar a reusabilidade;
- e uma flexibilidade e adaptabilidade que possa suportar a capacidade de operação multi-usuário e incorporar a diversidade de interfaces de *hardware* e *software* para fornecer recursos e ferramentas solicitados pelo domínio do assunto a ser ensinado.

Entretanto, pesquisas nesta área indicam que maioria dos ambientes de ensino-aprendizagem ainda não possui tais características. Muitos são desenvolvidos para a aplicação de

² Ver: http://ifets.ieee.org/periodical/vol_1_98/formal_discussion1.html.

ensino em um domínio específico, não favorecem incorporação de novas interfaces ou ainda, a reusabilidade.

GIANGRANDI & TASSO (1996) apontam um outro problema apresentado por alguns Sistemas Tutores Inteligentes, o fato de que os mesmos são baseados na hipótese de que o conhecimento do estudante não muda durante uma sessão tutorial e, portanto, o modelo do estudante é construído sem considerar os diferentes momentos nos quais os dados sobre o estudante devem ser coletados e atualizados.

Problemas relacionados ao tempo e ao custo de desenvolvimento de ambientes de ensino-aprendizagem também são apontados por BECK et al. (2001).

Há, portanto, a necessidade de técnicas que ajudem a resolver estas dificuldades. A modularização é uma das abordagens para a simplificação da construção de Sistemas Tutores Inteligentes, ela permite a reutilização de componentes de um sistema no desenvolvimento de outro, o que caracteriza a reusabilidade, e a incorporação de novos componentes ao ambiente, dando flexibilidade e permitindo a evolução do ambiente.

Este trabalho investiga conceitos e técnicas de Inteligência Artificial, como Inteligência Artificial Distribuída e agentes inteligentes, como soluções que possibilitem aos ambientes de ensino-aprendizagem alcançar as características que lhe são desejáveis e superar os problemas encontrados nos mesmos.

1.3 JUSTIFICATIVA

A utilização da abordagem de Inteligência Artificial Distribuída e da tecnologia de agentes inteligentes aparece como promissora para melhoria do projeto e implementação de Sistemas Tutores Inteligentes.

Segundo GIRAFFA (1999), as propostas de utilização agentes inteligentes para modelar Sistemas Tutores Inteligentes trazem uma grande vantagem em relação às arquiteturas tradicionais de Sistemas Tutores Inteligentes, uma vez que apresentam uma flexibilidade maior no tratamento dos elementos que compõem o sistema.

O fato de usar agentes para modelar os componentes de um Sistema Tutor Inteligente possibilita o agrupamento da arquitetura tradicional, um módulo = um agente³, ou na explosão de cada módulo em vários agentes. Essa abordagem aponta vantagens como:

- a modularidade, que facilita a construção de um sistema formado por módulos quase independentes, representa uma boa opção para reduzir o custo destes sistemas e ainda favorece a evolução dos mesmos;
- o uso de raciocínios heterogêneos, onde o problema pode ser formado por vários subproblemas, aos quais correspondem a diferentes técnicas no raciocínio e ainda, as facilidades de interação dos agentes podem ser exploradas na integração de várias ações na tentativa de se alcançar um determinado objetivo.

Na abordagem multi-agente, de acordo com COSTA (1999), agentes específicos podem ainda ser construídos para oferecer serviços adicionais a serem utilizados no processo de ensino à distância, através da internet. Um bom exemplo seria oferecer serviços automatizados de busca de material de referência, filtros inteligentes para correio eletrônico e salas de discussão.

O desenvolvimento deste trabalho se justifica uma vez que sua proposta é investigar como o uso da tecnologia de agentes inteligentes e conceitos de Inteligência Artificial Distribuída favorecem o desenvolvimento e evolução de ambientes de ensino-aprendizagem.

1.4 OBJETIVOS GERAIS DO TRABALHO

Este trabalho analisa e propõe o uso da tecnologia de agentes inteligentes na implementação de ambientes de ensino-aprendizagem, como alternativa para tornar esses ambientes mais flexíveis e efetivos no processo de ensino. Propõe uma arquitetura, fundamentada no modelo dos Sistemas Tutores Inteligentes, define quais funções do sistema são realizadas por agentes, quais as propriedades básicas estes agentes possuem e como os mesmos estão organizados no ambiente. Como contribuição, investiga ainda, a construção de agentes inteligentes através da definição de arquiteturas internas, plataformas e linguagens de programação usadas para o seu desenvolvimento.

³ O termo agente é usado como referência a agentes inteligentes.

1.4.1 Objetivos específicos

Como objetivos específicos tem-se:

- definição do uso dos computadores no processo de ensino-aprendizagem;
- descrição de um modelo para o desenvolvimento de ambientes de ensino-aprendizagem;
- definição de agentes inteligentes e exame das estruturas necessárias para o desenvolvimento de tais entidades;
- definição de uma arquitetura para ambientes de ensino-aprendizagem através do uso da tecnologia de agentes inteligentes;
- descrição da organização destes agentes no ambiente.

1.5 DELIMITAÇÃO DA PESQUISA

O projeto de um ambiente de ensino-aprendizagem implica uma série de tomadas de decisões que vão desde o aspecto técnico até o aspecto pedagógico. Há então a necessidade da delimitação do escopo deste trabalho, o qual está relacionado a aspectos orientados à tecnologia, onde propõe melhorias nos ambientes de ensino-aprendizagem através das características de algumas ferramentas computacionais.

Não faz parte do escopo deste trabalho discutir questões de ordem educacional, ou seja, a real efetividade dos ambientes computacionais no processo de ensino-aprendizagem. Assim como, levantar questões relacionadas especificamente com a implementação de tais ambientes.

1.6 MÉTODO DE PESQUISA

É realizada uma pesquisa exploratória que, segundo GIL (1991), do ponto de vista dos objetivos visa proporcionar maior familiaridade com o problema com vistas a torná-lo explícito ou construir hipóteses. Envolve levantamento bibliográfico; análise de exemplos que estimulem a compreensão. Assume, em geral, as formas de pesquisas bibliográficas e estudos de casos.

A pesquisa bibliográfica tem como objetivo o levantamento histórico do tema tratado — ambientes inteligentes de ensino-aprendizagem —, uma fundamentação teórica para os assuntos abordados e a definição dos limites da pesquisa.

1.6.1 Fases da pesquisa

As fases que compõem este trabalho são descritas a seguir.

Fase 1: Definição do trabalho

A definição do trabalho abrange a justificativa para o desenvolvimento de tal trabalho, a delimitação do tema e o estabelecimento dos objetivos da pesquisa realizada.

Fase 2: Pesquisa bibliográfica

“A pesquisa bibliográfica procura explicar um problema a partir de referências publicadas em documentos. Pode ser realizada independentemente ou como parte da pesquisa descritiva ou experimental. Em ambos os casos, busca conhecer e analisar as contribuições culturais e científicas do passado existentes sobre um determinado assunto, tema ou problema” (CERVO & BERVIAN, 2002, p. 65). Além de evitar duplicidade de pesquisa, evitar problemas ocorridos em trabalhos anteriores e fornecer uma fundamentação teórica.

Esta fase do trabalho, então, procura atingir as finalidades da pesquisa bibliográfica fazendo um levantamento bibliográfico a respeito:

- do uso dos computadores no processo de ensino-aprendizagem;
- de um modelo que suporte o desenvolvimento dos ambientes de ensino-aprendizagem;
- identificação de arquiteturas já propostas para tais ambientes;
- conceitos de agentes inteligentes;
- construção de agentes inteligentes.

Fase 3: Qualificação

Nesta fase da pesquisa é preparado e apresentado o texto para qualificação.

Fase 4: Especificação da arquitetura

Nesta fase é feita a especificação de uma arquitetura para ambientes de ensino-aprendizagem utilizando agentes inteligentes. São discutidas as características desejadas ao ambiente e determinado como o mesmo é organizado:

- Quantos agentes são utilizados, a princípio e
- Como as tarefas são distribuídas entre estes agentes.

Considerando as funções atribuídas aos agentes são determinadas as características desejadas a cada agente, o protocolo de comunicação a serem utilizados e sua interação com os demais agentes.

Fase 5: Exposição dos resultados

Nesta fase são expostos os resultados obtidos nas etapas anteriores, apresentada a conclusão obtida e consideradas as sugestões para futuros trabalhos.

1.7 ESTRUTURA DO TRABALHO

O restante deste trabalho está assim organizado:

No capítulo 2 o objetivo é estabelecer o estado da arte do uso dos computadores como ferramentas de ensino-aprendizagem e como a Inteligência Artificial e outras áreas contribuíram com novas abordagens nos ambientes de ensino-aprendizagem, permitindo a evolução dos mesmos.

No capítulo 3 são apresentados os conceitos que envolvem um Sistema Tutor Inteligente, assim como seus principais componentes. São descritas algumas arquiteturas já propostas para ambientes de ensino-aprendizagem, as quais utilizam os conceitos e componentes dos Sistemas Tutores Inteligentes.

No capítulo 4 é feita a definição do que é um agente inteligente, quais as suas principais propriedades e uma classificação dos mesmos. Assim como a maneira na qual os agentes inteligentes podem estar organizados através de conceitos de Inteligência Artificial Distribuída.

No capítulo 5 são apresentados conceitos relacionados à construção de agentes propriamente dita. As arquiteturas internas de agentes, que representam como estes agentes devem ser estruturados, as plataformas de desenvolvimento necessárias para a implementação de agentes e o processo de comunicação entre os mesmos.

No capítulo 6 é apresentada a arquitetura proposta. São enunciados os pressupostos que levaram a algumas decisões de projeto da mesma, é feita a descrição detalhada das funções de cada um dos seus componentes, assim como, a interação existente entre estes.

No capítulo 7 é feita, a conclusão deste trabalho e consideradas as sugestões para trabalhos futuros.

2 ENSINO-APRENDIZAGEM ASSISTIDO POR COMPUTADORES

No ensino mediado por computador, filosofias contrastantes que vão do objetivismo ao construtivismo estabelecem extremos de um contínuo. A filosofia objetivista está baseada na idéia de que estudantes aprendem por que lhes falam e que o conhecimento é objetivo e existe independentemente dos aprendizes. Como consequência, o material usado para transmissão de conhecimento nos ambientes de ensino-aprendizagem, dentro dessa filosofia, tende a ser mais explícito e declarativo. A filosofia construtivista presume que pessoas aprendem fazendo, ou seja, constroem o conhecimento situando experiências cognitivas em atividades adequadas.

Numa abordagem construtivista, os ambientes de ensino-aprendizagem, procuram em geral, colocar à disposição dos estudantes um conjunto de *softwares* que possibilite o aprender-a-aprender, ou seja, que o estudante se tome mais participativo e possa perceber os efeitos de suas tomadas de decisão no ambiente e, também, possa formular hipóteses sobre o conhecimento que está manipulando.

Os ambientes de ensino-aprendizagem evoluem, buscando apresentar características defendidas pelos novos paradigmas de ensino que surgem. Para que esses ambientes, então, se tornem efetivos no processo de ensino, devem resultar da junção de pesquisas realizadas em diversas áreas como a Educação, a Informática Aplicada a Educação, a Psicologia Cognitiva e outras afins.

O objetivo dessa junção é buscar o desenvolvimento de toda uma metodologia de trabalho que proporcione a identificação das características desejadas nos ambientes de ensino-aprendizagem; identificar as limitações existentes em termos de *hardware* e *software*; viabilizar a implementação de ambientes assim como, sua manutenção e evolução; e ainda, preparar os estudantes de maneira que eles entendam o papel da informática no processo de aquisição e construção do conhecimento.

O processo de evolução dos ambientes de ensino-aprendizagem é constante, uma vez que, novas teorias de ensino surgem e também a disponibilidade de novas tecnologias de *hardware* e *software* permite um incremento nas capacidades destes ambientes.

2.1 EVOLUÇÃO DOS AMBIENTES DE ENSINO-APRENDIZAGEM

Os primeiros sistemas voltados para o ensino através do computador, surgidos na década de 50 tiveram sua origem em projetos na área de Educação. Estes sistemas são conhecidos como Sistemas de Treinamento Baseado em Computador (*Computer-Based Training – CBT*) e Sistemas de Instrução Auxiliada por Computador (*Computer-Aided Instruction – CAI*).

A proposta dos CBTs e CAIs é apresentar um problema ao estudante, registrar sua resposta e avaliar seu desempenho. A seqüência de perguntas e respostas é dirigida pelos acertos e erros dos estudantes. Eles consideram que se a informação for apresentada ao estudante, ele poderá absorvê-la. As informações são transmitidas de uma forma passiva, onde o sistema controla a interação com o estudante.

Os Sistemas de Instrução Auxiliada por Computador nas suas versões iniciais, apresentavam instruções programadas que repetiam na máquina os mesmos métodos utilizados pelo paradigma vigente na época — *Behaviorista*⁴ —, ou seja, o estudante utilizava um conjunto de lições previamente organizadas pelo professor, ou instrutor, de forma seqüencial e com pouca interação. O conteúdo era pré-programado pelo professor baseado num currículo de referência, geralmente único, e elaborado proceduralmente. Nestes ambientes, de acordo com GIRAFFA (2001), existe uma série de passos ou tarefas a serem executados e à medida que o estudante atinge uma determinada marca — rendimento ou aprendizado — novos níveis são disponibilizados.

Os termos CAI e CBT são exemplos de uma variedade de termos similares que se referem a aplicações educacionais que, predominantemente, focalizam o ensino e a pedagogia de exercício e prática. Nestes ambientes a ênfase instrucional está na apresentação de material para a aprendizagem, onde a compreensão do estudante é supervisionada e onde os resultados esperados são pré-definidos, bifurcando-se ao longo de apresentações adicionais ou práticas

⁴ Para o Behaviorismo todo o comportamento pode ser explicado como uma reação motora a um estímulo específico do ambiente.

apropriadas. O projetista pode antecipar toda a seqüência do potencial instrucional e pode completar janelas pré-definidas de informações a serem mostradas; o computador controla o fluxo de material.

Segundo GIRAFFA (2001), no grupo dos Sistemas de Instrução Auxiliada por Computador encontram-se as seguintes modalidades de sistemas:

- programas de reforço ou exercício: o estudante pratica e testa conhecimentos de forma dirigida e procedural. As versões mais atuais destes programas utilizam recursos hipermídia mantendo estas características;
- tutoriais: os quais seguem o padrão de ensino da sala de aula tradicional onde o conteúdo é previamente organizado numa estrutura definida pelo professor e o estudante seleciona dentre as diversas opções disponíveis o conteúdo que deseja estudar. Mesmo aqueles que utilizam estrutura de hipermídia ou foram escritos em versão para a internet, mantêm as características do ensino tradicional e não apresentam grandes revoluções, sob o ponto de vista pedagógico;
- jogos educacionais e as simulações: são modalidades que adquiriram uma projeção maior nos últimos anos devido ao desenvolvimento dos recursos de *hardware* e *software* disponíveis no mercado. As diferenças conceituais entre os jogos e as simulações podem ser caracterizadas pelo fato de que o jogo é um processo intrinsecamente competitivo, em que coexistem a vitória e a derrota, já a simulação é a simples execução dinâmica de um modelo previamente definido.

BECK et al. (2001) enfatiza que nos Sistemas de Instrução Auxiliada por Computador e de Treinamento Baseado em Computador, a instrução não é individualizada de acordo com as necessidades do aprendiz. Em vez disso, as decisões sobre como mover um estudante através do material é como um roteiro, tal como “*se a questão 21 é respondida corretamente, proceda para a questão 54; caso contrário vá para a questão 32*”. As habilidades do aprendiz não são levadas em consideração.

Desta forma, os Sistemas de Instrução e Treinamento Baseados em Computador podem ser pouco efetivo no processo ensino-aprendizagem, uma vez que eles não possuem um conhecimento individualizado de cada estudante e não podem fornecer, conseqüentemente, o

mesmo tipo de atenção e ajuda individualizada que um estudante receberia de um tutor humano. Para um sistema de ensino-aprendizagem baseado em computador fornecer tal atenção, ele tem que, além de raciocinar sobre o domínio a ser ensinado, conhecer o estudante.

A percepção desta deficiência apresentada pelos Sistemas de Instrução e de Treinamento Baseados em Computador, levou a pesquisas em um outro tipo de sistemas, os Sistemas Inteligentes de Instrução Auxiliada por Computador (*Intelligent Computer Aided Instruction – ICAI*), surgidos na década de 70, a partir de projetos de pesquisas da área de Inteligência Artificial (IA), os quais podem ser vistos como uma evolução dos sistemas de instrução auxiliada por computador.

Os ICAI apresentam uma estrutura diferenciada dos CAI, utilizando técnicas de Inteligência Artificial para a representação do conhecimento e os resultados da Psicologia Cognitiva para representar as estratégias ou métodos de ensino a serem utilizados pelo sistema, a partir do conhecimento que este deve ter do estudante. Segundo TURINE et al. (1994), os sistemas ICAI foram projetados como uma tentativa de fazer com que o programa educacional deixe de ser um mero *virador de páginas eletrônico*⁵ e se torne um elemento mais ativo no processo de ensino-aprendizagem.

Uma modalidade de Sistemas Inteligentes de Instrução Auxiliada por Computador, amplamente pesquisada, é o Sistema Tutor Inteligente (STI). Uma característica importante de um Sistema Tutor Inteligente é prover ensino adaptativo, permitindo a realização de atividades apropriadas para que o nível de habilidade do usuário mude, quando acumular experiência no ensino. Os STIs são tratados com maior ênfase no capítulo 3 deste trabalho.

Novas modalidades de ambientes de ensino-aprendizagem foram surgindo à medida que o interesse por pesquisas na área, impulsionadas pela busca de novos métodos de ensino e aprendizagem, aumentou. Diferentes autores focaram suas pesquisas em diferentes aspectos dos Sistemas Tutores Inteligentes, o que resultou em uma variedade de siglas e termos utilizados, os quais muitas vezes se confundem, e, também o uso de diferentes técnicas e abordagens, que dificultam a caracterização destes sistemas. Os ambientes inteligentes de ensino que surgiram a partir destas pesquisas podem ser vistos como uma evolução dos STIs.

⁵ O termo “*virador de páginas eletrônico*” é utilizado como uma crítica aos programas CAI que apenas usam o meio eletrônico para fazer o mesmo que era realizado no papel, sem nenhum ganho significativo em nível de ensino-aprendizagem. O termo constitui-se em um jargão da área de Informática na Educação.

Uma sigla bastante encontrada na literatura é ILE – *Intelligent Learning Enviroment* ou *Interactive Learning Enviroment*, que é usada também como referência a Ambientes Inteligentes de Aprendizagem ou a Ambientes Interativos de Aprendizagem. Os Ambientes Interativos de Aprendizagem têm como principal característica a tentativa de expandir a abordagem tradicional de STIs suportando um aprendizado do estudante onde este tenha mais possibilidades de aprender fazendo, através do uso de componentes mais interativos. Um método de ensino bastante utilizado nos ILEs é o ensino baseado em investigação.

Os principais princípios associados a estes ambientes, de acordo com MCARTHUR et al⁶ apud COSTA (1999), são:

- construção e não instrução: o objetivo é explorar o fato de que estudantes podem aprender mais efetivamente através da construção de seu próprio conhecimento;
- o controle é do estudante e não do tutor: a questão é dar mais liberdade para o estudante controlar suas interações no processo de aprendizado. O tutor deve atuar como um guia, e não como o único detentor do conhecimento;
- a individualização é determinada pelo estudante e não pelo tutor: diferentemente dos STIs, a personalização da informação é o resultado da interação com o ambiente. Esta responsabilidade pode estar também associada ao sistema, mas o estudante ainda terá uma boa parte do controle de sua individualização;
- o conhecimento adquirido pelo estudante é resultado de suas interações com o sistema e não com o tutor: a informação adquirida vem como uma função das escolhas e ações do estudante no ambiente de ensino e não como um discurso gerado pelo tutor.

Estes princípios apresentam uma clara mudança no enfoque de aprendizado dos STIs para ambientes ILE. O processo deixa de ser centrado no tutor e passa a ser centrado no estudante. Sendo assim, torna-se necessária uma nova gama de ferramentas computacionais. Estas ferramentas incluem, freqüentemente, vídeo interativo ou outras representações gráficas, e

⁶ MCARTHUR, D.; LEWIS, M.; BISHAY, M. (1993). The roles of artificial intelligence in education: current progress and future prospects. RAND, Santa Monica, CA, USA, November.

permitem aos estudantes investigar e aprender tópicos de forma livre, sem estarem presos a algum tipo de controle.

Um outro tipo de ambiente de ensino-aprendizagem encontrado na literatura são os Sistemas de Ensino Colaborativo Apoiados por Computador (*Computer-Supported Collaborative Learning – CSCL*). Os CSCL se caracterizam por considerar vários estudantes trabalhando cooperativamente em um mesmo ambiente. São também conhecidos como Sistemas Tutores Cooperativos ou Sistemas de Aprendizagem Social. Estes sistemas combinam aspectos da modularidade dos Sistemas Tutores Inteligentes, o uso de redes e Micromundos⁷ podendo, ainda, agregar os elementos de simulações — jogos educacionais.

O aprendizado colaborativo pode ser entendido como grupos de estudantes trabalhando de forma interativa para resolver problemas. Estes ambientes têm sido vistos como benéficos, tanto em aspectos cognitivos como em aspectos sociais. Nesta situação, o foco não está mais na interação entre professor e estudante, mas em como os estudantes podem interagir entre si e como eles podem ensinar uns aos outros sem a necessidade de um professor (COSTA, 1999).

Um importante aspecto de ambientes colaborativos é que em situações de grupo, nem todos os estudantes possuem as mesmas habilidades. AYALA & YANO (1996) acreditam que a chave está em se encontrar as condições ideais que permitirão uma colaboração efetiva entre os estudantes. Por colaboração efetiva deve-se entender aquela situação onde o estudante pode aprender a partir de outros estudantes, enquanto ele realiza uma tarefa que necessita a aplicação de elementos de conhecimento internalizadas por outros estudantes e também passíveis de serem internalizadas por ele.

A partir de pesquisas em Educação à Distância, buscando usar as potencialidades oferecidas pela internet, surge uma nova modalidade de ambientes de ensino-aprendizagem que são os Web-tutores. Segundo GIRAFFA (2001), os ambientes de ensino-aprendizagem na Web carregam características importantes sob o ponto de vista da abordagem social, uma vez que o desenvolvimento da linguagem de um indivíduo se dá através de um espaço social. Neste

⁷ Os micromundos se baseiam na proposta de aprendizagem de Jean Piaget, em que a ênfase está na construção do conhecimento por parte do estudante e não na mera transmissão de conhecimentos (como ocorre na categoria CAD). Os micromundos foram criados para desenvolver habilidades cognitivas no estudante e para proporcionar o pensamento reflexivo. Trabalham basicamente com a solução de problemas e não apresentam conteúdos pré-fixados e respostas.

espaço, o estudante pode trocar experiências com seus pares e acessar a diferentes fontes de informação em diferentes formatos e organizações. Cada estudante manipula o conhecimento no seu tempo e o atendimento de suas necessidades não está ligado às restrições físicas.

A flexibilidade de acesso à informação, juntamente com a capacidade de exploração de hiperdocumentos faz com que os Web-tutores se tornem bastante atrativos para a utilização como programas de ensino através da recuperação de informação. Sob esta perspectiva, quanto mais controle o usuário tiver sobre o material a ser examinado, maior chance de sucesso haverá na interação com o sistema (FREIRE, 1998).

Seguindo essa linha de raciocínio, a tecnologia hipermídia chama a atenção dos desenvolvedores de ambientes de ensino-aprendizagem, pois o estudante ao contrário de seguir caminhos que o autor pré-estabelece, pode seguir seus próprios caminhos, de acordo com seus interesses e preferências. O que faz com que ele passe a ser ativo, ocupando um papel participante em seu próprio aprendizado (SCHANK⁸ apud FREIRE *ibid.*).

Os atuais ambientes de ensino-aprendizagem utilizam recursos hipermídia, permitem o trabalho individual ou coletivo, utilizam técnicas de Inteligência Artificial em diferentes escalas e, conforme a metodologia utilizada pelo professor, podem ser explorados de diferentes formas. O uso desses diferentes recursos torna os sistemas de ensino-aprendizagem cada vez mais personalizados e altamente interativos, buscando fazer com que o estudante seja mais participativo e criativo.

FREIRE *ibid.* enfatiza que a liberdade total na exploração do material instrucional é uma idéia favorável ao processo de aprendizado, uma vez que possibilita ao estudante ter controle sobre o aprendizado e estudar se torna mais pessoal e interessante. No entanto vários experimentos têm mostrado o contrário, isto é, situações em que o estudante apresenta melhor rendimento são aquelas onde ele é guiado ou orientado de alguma forma.

Portanto, o desafio na construção de ambientes de ensino-aprendizagem está em utilizar as novas tecnologias disponíveis para melhorar a qualidade destes ambientes e encontrar um equilíbrio entre a liberdade que o estudante deve ter ao explorar as informações e o controle que o sistema deve exercer em disponibilizar informações adequadas.

⁸ SCHANK, R.C. (1994) Active learning Through Multimedia. IEEE Multimedia, p. 69-78.

2.2 CONSIDERAÇÕES FINAIS

Um paralelo entre o desenvolvimento da tecnologia computacional e o desenvolvimento da pedagogia educacional pode ser traçado. Da mesma maneira que os sistemas computacionais observaram uma evolução e possibilitaram novos rumos para o projeto e para o desenvolvimento de sistemas de ensino-aprendizagem, a pedagogia passou a considerar outras possibilidades de desenvolver o processo de ensino-aprendizagem (EDGAR⁹ apud GIRAFFA, 2001).

O Quadro 1, demonstra as principais características apresentadas pelos ambientes de ensino-aprendizagem dentro do processo de evolução dos mesmos.

QUADRO 1 — Características dos ambientes de ensino-aprendizagem.

	Técnicas utilizadas	Características	Interação com o estudante
CBT e CAI	Programação procedural	Perguntas e respostas dirigidas por acertos e erros.	<ul style="list-style-type: none"> ▪ Estudante tem papel passivo diante do sistema. ▪ O sistema não considera as características individuais de aprendizado de cada estudante.
ICAI e STI	Técnicas de Inteligência Artificial	Utilização de módulos separados para representação do conhecimento, das estratégias de ensino e o modelo do estudante.	<ul style="list-style-type: none"> ▪ O controle sobre o aprendizado pode ser alternado entre o estudante e o sistema. ▪ Considera as características de cada estudante para fornecer ensino individualizado.
ILE	IA e simulação	Uso de módulos e ênfase na interação com o estudante.	<ul style="list-style-type: none"> ▪ O estudante tem papel ativo no processo de aprendizagem.
CSCL	IA e redes	Uso de módulos e ênfase na colaboração entre vários elementos participantes do processo de aprendizagem.	<ul style="list-style-type: none"> ▪ Existe a colaboração e interação estudante-estudante e estudante-tutor. ▪ Considera o aspecto social da aprendizagem.
Web-Tutor	IA, redes e hipermídia.	Possui características dos sistemas anteriores, aproveitando as vantagens oferecidas pela internet.	<ul style="list-style-type: none"> ▪ Considera grande interação com o estudante. ▪ Aprendizagem proporcionada através do ensino à distância.

⁹ EDGAR, R. (1995). PC is to Piaget as WWW is to Vygotsky. In: SIGGRAPH COMPUTE.R GRAPHICS CONFERENCE, 5., Los Angeles, 1995. *Proceedings*. p.71-95.

3 OS SISTEMAS TUTORES INTELIGENTES

As primeiras pesquisas em Sistemas Tutores Inteligentes (STIs) pretendiam a descoberta de uma forma de tornar o computador um tutor inteligente, que fosse capaz de criar uma representação das fases de conhecimento do estudante, uma vez que a incapacidade de adaptação as reais necessidades do estudante era uma das limitações dos Sistemas de Instrução Baseados em Computador, até então desenvolvidos.

Segundo SLEEMAN & BROWN¹⁰ apud TURINE et al. (1994); CASAS (1999), para que este tipo de sistema tivesse impacto significativo no processo de ensino-aprendizagem, era necessário que tivessem suas próprias habilidades para resolver problemas, seu próprio diagnóstico do estudante ou capacidade de modelagem do estudante e suas próprias capacidades de explicação.

“uma vez que os estudantes não são iguais, nem todos os estudantes requerem a mesma quantidade de reforço para compreender um determinado tópico, idealmente em um STI, o estudante pode progredir exatamente na medida do seu entendimento. Conceitos que o estudante não entende bem poderão ser repetidos, enquanto outros podem ser apresentados apenas uma única vez, conseqüentemente evitando a mentalidade de apostilas para classes padrões” (THOMPSON, 1996, p.10).

As pesquisas em STIs que se seguiram vem ao encontro das modernas correntes da Educação que estão privilegiando a troca de paradigma: o tradicional, modelo de ensino centrado no professor, pelo novo, modelo centrado no estudante e baseado no “*aprender a aprender*”. E onde a participação do estudante no processo de ensino-aprendizagem é incrementada (GIRAFFA, 1999).

Novas tecnologias computacionais como hipermídia e poderosas linguagens de programação, nos paradigmas de orientação a objetos e orientação a agentes, permitiram uma melhora na funcionalidade e desenvolvimento de Sistemas Tutores Inteligentes.

A arquitetura da nova geração de Sistemas Tutores Inteligentes também começou a mudar, foi reforçada pelo uso da tecnologia de agentes inteligentes em sua implementação.

¹⁰ SLEEMAN, D.; BROWN, J. S. (1982). *Intelligent Tutoring Systems*. London, Academic Press, 345p.

3.1 COMPONENTES DE UM STI

Os Sistemas Tutores Inteligentes alcançam sua inteligência através da representação do conhecimento do domínio a ser ensinado, pela representação de decisões pedagógicas sobre como transmitir o material, além de informações sobre o estudante. Sendo, então, capazes de compreender a quem ensinar, o quê e como ensinar.

A partir dos trabalhos realizados sobre STIs e as propostas de arquiteturas apresentadas por estes, observa-se que os mesmos podem variar um pouco quanto à sua arquitetura, ou seja, quais e como são organizados os módulos que a compõem, mas entretanto, essa variação recai sobre as características fundamentais que os Sistemas Tutores Inteligentes devem possuir, que são: a representação do conhecimento do domínio, a representação das estratégias pedagógicas e o conhecimento que o sistema deve ter do estudante.

WOOLF¹¹ apud COSTA (1999), identificou quatro componentes principais em um STI: módulo domínio do conhecimento, modelo do estudante, módulo pedagógico e módulo de comunicação. Outros trabalhos, subdividiram alguns desses componentes definindo um quinto e por vezes um sexto módulo. É considerado neste trabalho o modelo apresentado por WOOLF *ibid.*, por este apresentar um modelo usado na maioria dos trabalhos pesquisados.

A Figura 1, mostra os módulos componentes de um Sistema Tutor Inteligente e suas interações.

¹¹ WOOLF, B. AI in Education. *Encyclopedia of Artificial Intelligence*. Wiley & Sons, Inc., New York, 1992, p. 434-444.

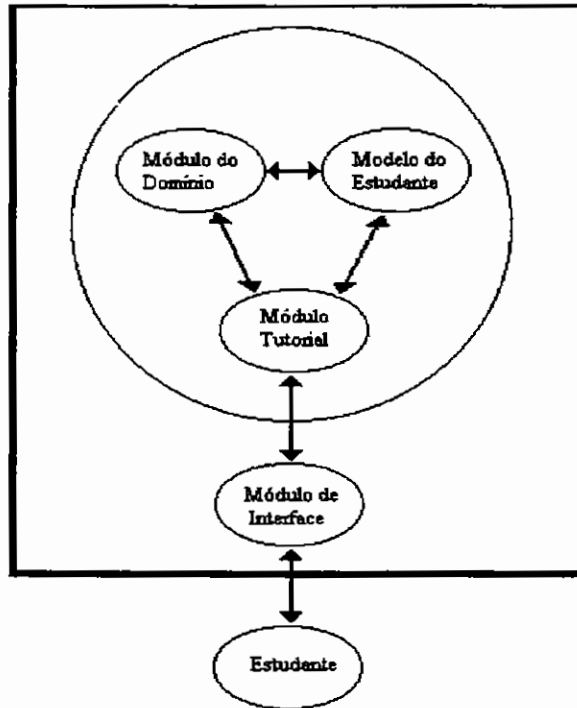


FIGURA 1 — Modelo Geral de um STL

Fonte: NWANA (1990).

A seguir são descritas as características e funcionalidades de cada um dos componentes de um STL. A denominação de cada componente utilizada é a encontrada em vários trabalhos consultados (TURINE et al., 1994); (FREIRE, 1998); (PUGLIESI, 1998).

3.1.1 Módulo do domínio

O módulo do domínio contém o conhecimento do assunto a ser ensinado. O conhecimento é representado separado das informações de controle — de como ensinar.

TURINE et al. (1994) enfatizam que o módulo do domínio exerce duas funções principais:

1. serve como fonte de conhecimento do assunto a ser ensinado para o estudante. Isto inclui, capacidade de geração de explicações e respostas para os estudantes, bem como tarefas e questões a serem realizadas;

2. fornece um padrão para avaliar o desempenho geral do estudante. Para isto, o sistema deve ser capaz de gerar soluções para os problemas no mesmo contexto do estudante, para que suas respectivas respostas sejam comparadas. Segundo WOOLF & MCDONALD¹² apud FREIRE (1998), mesmo que o módulo do domínio não tenha necessariamente habilidades para resolver todos os problemas para o estudante, deve pelo menos reconhecer uma resposta incorreta.

Os principais problemas encontrados no projeto do módulo de domínio são o processo de aquisição de conhecimento, via especialista, e a representação deste conhecimento através de alguma técnica adequada.

A escolha de como representar um determinado conhecimento no domínio de um STI depende do tipo de conhecimento a ser manipulado e do seu uso pretendido no sistema. Em domínios de natureza descritiva e teórica, a representação utilizada é geralmente declarativa — através de Redes Semânticas¹³ ou *Frames*¹⁴. Em domínios orientados à execução de uma determinada tarefa, a representação tende a ser procedimental — Regras de Produção¹⁵.

A escolha da representação do conhecimento deve recair sobre aquele método que melhor e mais facilmente atenda aos requisitos de representação e manipulação do raciocínio. Para CASAS (1999), uma escolha inadequada pode comprometer todo o desempenho do sistema, uma vez que este módulo deve ser capaz de determinar, entre outras coisas, a complexidade e conseqüentemente a forma de apresentação dos conceitos da área de conhecimento em questão. Um resumo das características de tipos de representações e conhecimento é apresentado no Quadro 2.

¹² WOOLF, B.; MCDONALD, D. D. (1984). Building a computer tutor: Design issues. *IEEE Computer*, p. 61-73.

¹³ Nas redes semânticas o conhecimento é representado em termos de nós conectados entre si.

¹⁴ Nos *frames* o conhecimento é representado através do conceito de entidades, onde cada entidade é descrita através de uma coleção de atributos e valores associados a eles.

¹⁵ $\alpha \rightarrow \beta$: se α for verdade implica β .

QUADRO 2 — Características dos tipos de representação de conhecimento.

Representações	Características	Alguns campos de aplicação
Lógica dos predicados	<ul style="list-style-type: none"> ▪ Facilidade de manipular novos fatos a partir de fatos conhecidos. 	<ul style="list-style-type: none"> ▪ Sistemas de dedução natural. ▪ Linguagens de programação declarativas.
Regras de produção	<ul style="list-style-type: none"> ▪ Parecida com o modo das pessoas falarem sobre como resolvem seus problemas. 	<ul style="list-style-type: none"> ▪ Sistemas especialistas. ▪ Engenharia do conhecimento.
Redes semânticas	<ul style="list-style-type: none"> ▪ São orientadas semanticamente. Simulam o modelo psicológico de memória associativa humana. 	<ul style="list-style-type: none"> ▪ Processamento de linguagem natural.
<i>Frames</i>	<ul style="list-style-type: none"> ▪ Estrutura de dados para representar um conceito ou uma situação dada. ▪ Apropriada na interpretação de uma seqüência específica de acontecimentos. ▪ Permite indicar como os acontecimentos se relacionam entre si. 	<ul style="list-style-type: none"> ▪ Interpretação visual. ▪ Compreensão de fala. ▪ Situações estereotipadas.

Fonte: CASAS (1999).

A modelagem do conhecimento a ser disponibilizado é de grande importância para o sucesso do sistema como um todo. É importante no desenvolvimento de um Sistema Tutor Inteligente procurar uma representação do conhecimento que esteja preparada para o crescimento incremental do domínio.

3.1.2 Modelo do estudante

O modelo do estudante armazena informações de cada estudante de forma individual. Este módulo deve conter uma representação do estado atual de conhecimento e do desempenho do estudante sobre o assunto que esta sendo ensinado. As informações do conhecimento e do desempenho do estudante servem como subsídio para o sistema decidir o que e como deve ensinar. Segundo FREIRE (1998), este módulo deve ser constituído por dados estáticos, que são

os dados cadastrais do estudante e seu nível de conhecimento inicial, e dados dinâmicos, que representam o desempenho e comportamento do estudante.

TURINE et al. (1994) afirmam que o modelo do estudante possui seis tipos de funções básicas:

1. elaborativa: para ajudar a complementar o conhecimento do estudante;
2. estratégica: para ajudar a promover mudanças significativas nas estratégias de ensino pelo módulo tutorial;
3. de diagnóstico: para ajudar a diagnosticar os *bugs* do conhecimento do estudante;
4. corretiva: para ajudar a eliminar os *bugs* do conhecimento do estudante;
5. de prognóstico: para ajudar a determinar as prováveis respostas do estudante às ações do tutor;
6. avaliativa: para ajudar a avaliar tanto o estudante como o sistema.

Um modelo realista do estudante implica uma atualização dinâmica, à medida que o sistema avalia o desempenho do estudante. Este módulo tem também por objetivo auxiliar no diagnóstico, correção e prevenção de erros do estudante, e é considerado o módulo principal, o núcleo de qualquer STI (CASAS, 1999).

Todas as informações coletadas pelo modelo do estudante devem ser relevantes e aptas a serem usadas pelo módulo tutorial do sistema, uma vez que o propósito do modelo do estudante é justamente esse, fornecer dados para o módulo tutorial.

COSTA (1999) cita algumas técnicas para a construção do modelo do estudante, como por exemplo:

- incluir um reconhecimento de padrões aplicados à história das respostas fornecidas por ele;
- comparar a conduta do estudante com a de um especialista e verificar os pontos em comum;

- registrar as preferências do estudante;
- indicar seus objetivos particulares;
- registrar as coisas que ele costuma esquecer quando interage com o tutor.

Idealmente, um STI deveria saber o nível de conhecimento do estudante no assunto que está sendo ensinado e o comportamento de aprendizagem do estudante, envolvendo aspectos como: ritmo, os métodos de apresentação aos quais o estudante responde melhor — suas preferências — , o seu nível de interesse e motivação pelo assunto em questão e sua área de especialidade, entre outras (TURINE et al., 1994).

Uma das maneiras que MESQUITA & SILVA (1998), usam para conhecer as preferências do estudante é ao estudante errar um exercício este mesmo exercício é apresentado usando uma outra técnica ou estratégia pedagógica. No momento que estudante acerta o exercício, é anotada no banco de dados do estudante a técnica de exposição que originou o acerto. Quando expondo um novo elemento de conhecimento, o sistema utiliza a técnica de exposição com maior taxa de acerto, antes de utilizar outras.

A tarefa de reunir informações relevantes sobre cada estudante individualmente, é chamada modelagem do estudante. A modelagem do estudante servirá para guiar os procedimentos de ensino a serem utilizados pelo sistema tutor.

Durante o processo de modelagem do estudante, quando quer que seja uma nova parte de informação sobre o comportamento do estudante se torna avaliável, o modelo do estudante deve ser modificado para dar incrementalmente melhores descrições da performance do estudante (GIANGRANDI & TASSO, 1996).

Por enquanto, de acordo com VASSILEVA¹⁶ apud TURINE et al. (1994); FREIRE (1998), ainda não existem técnicas estabelecidas para os modelos de estudante, o que existe é uma coleção de protótipos, e não está claro o quanto eles podem ser generalizados além de alguns poucos domínios recorrentes na literatura, como no ensino de matemática, por exemplo.

¹⁶ VASSILEVA, J. (1990). A classification and synthesis of student modelling techniques in intelligent computer-assisted instruction. In: NORRIE, D.H. & SIX, H.W. eds. *Lecture Notes in Computer Science, Computer Assisted Learning — 3rd International Conference, ICCAL'90*, Springer-Verlag.

Os principais problemas encontrados na modelagem do estudante parecem estar relacionados com o que representar e como representar.

O modelo do estudante, segundo COSTA & WERNECK¹⁷ apud CASAS (1999), pode ser representado apoiando-se em alguns modelos de descrição, a saber:

- modelo diferencial: onde a resposta do estudante é comparada com a base de conhecimento. Nesta modelagem o conhecimento do estudante é somente um subconjunto do conhecimento do especialista;
- modelo de *overlay* ou superposição: o conhecimento do estudante é representado como um subconjunto da base de conhecimento do sistema. Este modelo assume (implícita ou explicitamente) que os erros ou comportamentos anômalos do estudante são sempre devidos à ausência de alguma informação presente na base do domínio;
- modelo de perturbação ou modelo *buggy*: também relaciona o modelo do estudante com a base de conhecimento do domínio. O modelo de perturbação consiste na representação do conhecimento correto e incorreto do estudante, concluídos pelo sistema através de diálogos com o estudante. Este modelo assume que os erros do estudante são decorrentes da concepção errônea de algum conceito ou ausência dele. Neste, existe a base do domínio e uma biblioteca de erros típicos; o modelo do estudante inclui elementos da base do domínio e da biblioteca de erros;
- modelo de simulação: o ambiente possui um modelo de como o estudante pode ou deve se comportar em determinada situação e através deste modelo e permite prever o comportamento futuro do estudante, ou seja, a resposta do estudante baseado no seu comportamento durante a sessão de trabalho;
- modelo de crenças: consiste em um conjunto de crenças refletindo o grau de compreensão do estudante sobre um conceito em particular.

ERICCO & MICARELLI (1994), citam o sistema PROUST, desenvolvido por Lewis Johnson e Soloway¹⁸ para o ensino de programação na linguagem Pascal, o qual usa o modelo

¹⁷ COSTA, R. ; WERNECK, V. (1996). Tutores Inteligentes. Rio de Janeiro: COPPE/UFRJ.

de perturbação. Durante uma sessão do estudante com o sistema, este examina a solução proposta pelo estudante a um dado problema, procura pela descrição correspondente ao problema e cria hipóteses sobre os métodos que podem ser usados para resolver o problema. Se uma destas hipóteses bate com a solução do estudante, o sistema conclui que a solução é correta. Caso contrário, o sistema checa a base de conhecimento de erros comuns para ver se algum deles pode explicar a solução proposta. Portanto, o sistema é capaz de identificar ambos os planos usados pelo estudante para sua solução a um dado problema e a intenção do estudante no uso deste plano.

Segundo THOMPSON (1996), um dos problemas mais difíceis no projeto de um STI é determinar efetivamente e representar o conhecimento atual do estudante sobre um assunto em pauta. Enquanto importantes progressos têm sido feitos no desenvolvimento de STIs, efetivamente modelar o estudante continua a apresentar significantes desafios. Estes desafios surgem primariamente das inconsistências no raciocínio do estudante. Estudantes algumas vezes mudam seu pensamento, se distraem enquanto trabalham e apresentam resultados não tão bons. Eles também aprendem novas matérias ao mesmo tempo em que esquecem matérias já vistas, e cometem erros negligentes. Além disso, eles simplesmente não recordam um conhecimento de um item em um particular no momento que ele é requerido.

Construir portanto, um modelo perfeito do estudante não é trivial. “Provavelmente esta perfeição seja impossível, pois o canal de comunicação sistema estudante usado — teclado ou a tela — é considerado muito restrito. Tutores humanos, professores, normalmente combinam dados de uma grande variedade de fontes, como por exemplo os efeitos da voz ou expressão facial dos seus estudantes” (NWANA, 1990, p. 266).

3.1.3 Módulo tutorial

O módulo tutorial oferece uma metodologia para o processo de aprendizado. Este módulo é formado por um conjunto de estratégias de ensino que permitem ao STI decidir quando revisar conceitos, quando apresentar um novo tópico e quais tópicos apresentar. As entradas deste

¹⁸ JOHNSON, L.W.; SOLOWAY, E. (1985). PROUST: An automatic debugger for pascal programs. *Byte*, v. 10, n.4, p.179-190.

módulo são fornecidas pelo modelo do estudante, as decisões pedagógicas são feitas então, de acordo com as necessidades individuais de cada estudante.

Uma estratégia de ensino pode ser vista como um plano, ou seja, uma estrutura de ações visando atingir determinados objetivos. As estratégias constituem o conhecimento sobre como ensinar. Isto é, como gerar, a partir de informações sobre o estudante — fornecidas pelo modelo do estudante — e de sua própria estrutura tutorial, uma seqüência de atividades pedagógicas capazes de apresentar com sucesso um determinado tópico a um determinado estudante.

No projeto de um Sistema Tutor Inteligente deve ser adotado um método ou estratégia de ensino que possa estimular o estudante e fixar o conhecimento no mesmo. Um método muito utilizado pelos tutores de um modo geral, é o chamado método socrático¹⁹, em que, partindo de conhecimentos que o estudante já domina, o tutor ensina através de perguntas e diálogos, levando-o a tirar suas próprias conclusões. No método socrático o material apresentando é projetado para induzir os estudantes a identificarem enganos e interpretações errôneas a cerca do conteúdo (CASAS, 1999). Este método se ajusta melhor às tarefas de aprendizagem conceituais ou procedurais.

Outro método de ensino empregado em STIs é o *coaching* (treinamento), que emprega atividades de entretenimento como jogos para transmitir conceitos relacionados. Utilizam-se ainda as estratégias pedagógicas de orientação, em que o sistema é explicitamente chamado pelo estudante quando este necessita orientação sobre o que fazer. As estratégias podem incluir orientação para tarefas novas, transições entre tópicos e interação com o estudante para achar opções melhores no contexto.

Um terceiro método faz a utilização de hipertextos, onde o estudante navega numa estrutura de hipertexto e explora o conteúdo a partir dos seus interesses e pré-requisitos. Este documento é organizado de tal maneira que cada subdivisão lógica do assunto está ligada ao documento, através de diversos tipos de ligações, possibilitando que o estudante navegue por diferentes alternativas para explorar o domínio. O modelo de hipertexto, segundo CASAS *ibid.*, abre excelentes perspectivas para a construção de Sistemas Tutores Inteligentes, uma vez que pode abrigar, no mesmo documento, diferentes formas de representação de conhecimento. Além disto, o estudante pode trabalhar de forma mais participativa e dinâmica. Podendo controlar

¹⁹ Socrático - Pretende proporcionar instrução através de diálogo com os estudantes e detectar erros.

dinamicamente as informações, de acordo com sua lógica pessoal, que pode ser diferenciada da lógica do autor do sistema.

A seleção do conjunto de estratégias de ensino que constitui o STI é um aspecto muito importante para garantir a qualidade pedagógica do ambiente. A seleção de uma estratégia depende de fatores como o nível de conhecimento do estudante, a motivação e as características efetivas do mesmo e, ainda, o domínio de conhecimento a ser ensinado. A utilização da mesma estratégia não produz um efeito satisfatório para todos os estudantes. Cada estratégia tem vantagens específicas e é muito útil saber qual a estratégia que fortalece adequadamente o processo de aquisição para um determinado estudante. Professores bons apresentam o mesmo material de modos diferentes. As situações em que isto pode ocorrer não são previsíveis. O que se pode fazer é procurar diversificar o auxílio oferecido ao estudante levando em consideração os indicadores retirados do próprio trabalho progressivo do estudante (CASAS, 1999).

De acordo com as estratégias de ensino que podem ser utilizadas encontramos ainda algumas técnicas. Para cada aspecto do ensino levado em consideração ter-se-á uma respectiva técnica que mais se adequa à situação. Segundo CASAS *ibid.*, existem muitas técnicas de ensino, das quais destacam-se as seguintes:

- expositiva: onde o professor faz uma exposição do assunto da aula;
- perguntas: através de questões cuidadosamente colocadas que levam a um melhor conhecimento do estudante, das suas dificuldades e interesses;
- discussão: consiste na discussão de um tema, por parte dos estudantes, sob a direção do professor;
- demonstração: é usada para comprovar determinadas afirmações, apresentando razões ou fatos logicamente encadeados.

“A construção de um STI com múltiplas estratégias é muito útil para aumentar a qualidade pedagógica dos ambientes” (GIRAFFA, 1999, p. 37).

A forma de divisão do controle, ou seja, como é feito o gerenciamento da iniciativa no controle do aprendizado é outro importante aspecto a ser considerado nos STIs. De acordo com

FREIRE (1998) a divisão do controle é o principal problema enfrentado pelo módulo tutorial e ele pode ocorrer das seguintes maneiras:

- controle total do sistema sobre a forma de aprendizagem. O sistema monitora todas as atividades realizadas pelo estudante, o qual apenas responde às questões formuladas pelo sistema;
- controle do aprendizado é balanceado entre o sistema e o estudante. Ambos podem perguntar e responder questões, e
- controle total do estudante sobre a aprendizagem. O estudante tem a possibilidade de controlar totalmente o sistema. Este controle está presente, por exemplo, em ambientes de aprendizagem por descoberta ou em sistema de treinamento.

3.1.4 Módulo de interface

O módulo de interface controla a interação do sistema com o estudante. Este módulo define como o material é apresentado ao estudante. A complexidade para a implementação do módulo de interface é bastante variável, podendo ser desde simples janelas de diálogo até linguagem natural e reconhecimento de voz. Uma questão a ser considerada no desenvolvimento deste módulo, segundo BECK et al. (2001), é a aplicação de realidade virtual para permitir uma imersão total do estudante no sistema.

Este módulo é responsável por traduzir a representação interna do sistema para uma linguagem de interface que é compreensível para o estudante e, no sentido inverso, buscar respostas do sistema para as intervenções do usuário. Deve acomodar os objetivos do sistema e do estudante, podendo assim, enriquecer esta interação com o uso de vários meios de comunicação — texto, som, imagem —, além de fornecer meios para que seja possível ao professor e/ou especialista alterar e atualizar o sistema. A interface tem que ser construída de uma maneira amigável, simples e que estimule o estudante.

Quanto mais um estudante se sentir inserido no seu ambiente de aprendizagem, mais rica será a sua experiência. Esta inserção pode ser facilitada através da interação com uma interface de múltiplos meios, ou seja, vários recursos de comunicação. “A escolha do recurso de

comunicação entre usuário e sistema, mais adequado ao tópico a ser ensinado e a eficácia da comunicação são portanto, um problema relevante” (NWANA, 1990, p. 272).

VICCARI & OLIVEIRA (1992) enfatizam que a questão da interação cresce de importância nos STIs, pois é na interação que o sistema tutor exerce duas de suas principais funções: a apresentação do material instrucional e a monitoração do progresso do estudante.

Na apresentação do material instrucional é necessário evitar que o estudante se entedie, ou seja, é preciso riqueza de recursos durante a apresentação. As informações são mais bem absorvidas se o estudante for um participante ativo no processo de aprendizado, e se a apresentação envolver vários dos sentidos do estudante (TURINE et al., 1994).

De acordo com VICCARI & OLIVEIRA *ibid.*, são ainda objetivos a serem cumpridos pelo módulo de interface:

- facilitar o diálogo com o estudante, onde este possa intervir facilmente no discurso do sistema tutor e vice-versa;
- dar respostas dentro de limites de tempo aceitáveis;
- realizar a monitoração o máximo possível em *background*, não onerando o estudante com questionários excessivos.

3.2 CARACTERÍSTICAS FUNDAMENTAIS DE UM STI

Como já apresentado, uma das principais características de um Sistema Tutor Inteligente é prover ensino adaptativo, permitindo a realização de atividades apropriadas para que o nível de habilidade do usuário mude quando este acumular experiência no ensino. O STI deve, também, possuir a habilidade para diagnosticar as concepções erradas do estudante durante o processo de ensino-aprendizagem, e baseado nesse diagnóstico fornecer ao estudante o ensino adequado. Outra importante característica destes sistemas é oferecer uma considerável flexibilidade na apresentação do material ao estudante.

VICCARI & MOUSSALLE²⁰ apud TURINE et al. (1994) afirmam que um Sistema Tutor Inteligente ideal deveria possuir ainda, as características gerais apresentadas a seguir. E segundo NWANA (1990), algumas dessas características são as que diferenciam os STIs dos Sstemas de Instrução Auxiliada por Computador:

- o STI deve assegurar uma clara articulação de conhecimento num domínio limitado, isto é, uma representação explícita e aprofundada do domínio;
- o projetista, ou especialista no assunto, deve definir o conhecimento e as regras de inferência, mas não a seqüência da aula, que deve ser determinada pelo sistema;
- possibilitar e incentivar a exploração do assunto em questão pelo estudante;
- possuir um modelo de performance do estudante, que deve ser mantido dinamicamente, e usado para guiar a aula;
- fornecer diagnóstico detalhado de erros do estudante, ao invés de mostrar simplesmente a resposta certa;
- permitir ao estudante questionar o sistema, além de responder as questões formuladas por ele;
- possuir vários métodos de ensino para apresentação do assunto e operar conforme o(s) método(s) de ensino adotado;
- possuir meta-conhecimento²¹ para resolver situações não previstas nas regras que descrevem o conhecimento do tutor;
- possuir mecanismos que permitam a simulação automática e resolução interativa dos problemas;
- possuir mecanismos que descrevam o raciocínio que o estudante e o tutor utilizam ao explorar um determinado assunto;

²⁰ VICCARI, R.M. & MOUSSALLE, N. (1990). Tutores inteligentes para o ensino da linguagem Prolog. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, I., Rio de Janeiro. *Anais*. Rio de Janeiro, v.1, p.111-134.

²¹ Referente à aplicação independente do domínio.

- ser flexível em todos os níveis, — arquitetura, controle, comunicação, adaptação ao estudante;
- possuir capacidade para reconstruir estados passados.

De acordo com VICCARI & MOUSSALLE *ibid.*, através da utilização de vários métodos e recursos oferecidos pela IA, tem-se tentado dotar um sistema tutor com algumas destas características. Cada uma delas já constitui uma tarefa bastante complexa. Contar com todas elas num único sistema compreende muito mais complexidade, trabalho e pesquisas.

Um dos objetivos deste trabalho, como mencionado na introdução, é analisar arquiteturas já propostas para ambientes de ensino-aprendizagem, procurando encontrar nestas, abordagens que se mostrem adequadas para o desenvolvimento de ambientes de ensino-aprendizagem.

Nas seções a seguir são descritas três arquiteturas já propostas, as quais se mostraram relevantes para este trabalho.

3.3 SIATE — SISTEMA INTELIGENTE DE APOIO A TREINAMENTO E ENSINO

O ambiente de treinamento e ensino SIATE, proposto PUGLIESI²² (1998) e FREIRE²³ (1998), em trabalhos complementares, integra características de sistemas tutores, sistemas híbridos de conhecimento, sistemas de raciocínio baseado em casos e hipermídia, utilizando as vantagens de cada uma dessas tecnologias.

Um protótipo do ambiente foi construído para validar a arquitetura proposta, utilizando como domínio de aplicação o tema Aquisição de Conhecimento, que é um dos processos usados para a construção de Sistemas Inteligentes. Tal protótipo consiste de um hiperdocumento para WWW que contém informações não apenas sobre o domínio da aplicação, como também sobre o ambiente de ensino SIATE. Foi implementado utilizando a linguagem HTML (*Hipertext*

²² Sistema híbrido inteligente para suporte a um ambiente de treinamento e ensino.

²³ O sistema tutor de um ambiente inteligente para treinamento e ensino.

Markup Language), juntamente com Applets Java, JavaScript, CGI (Common Gateway Interface) e MiniSQL²⁴.

A Figura 2, apresenta os elementos que compõem a arquitetura do SIATE.

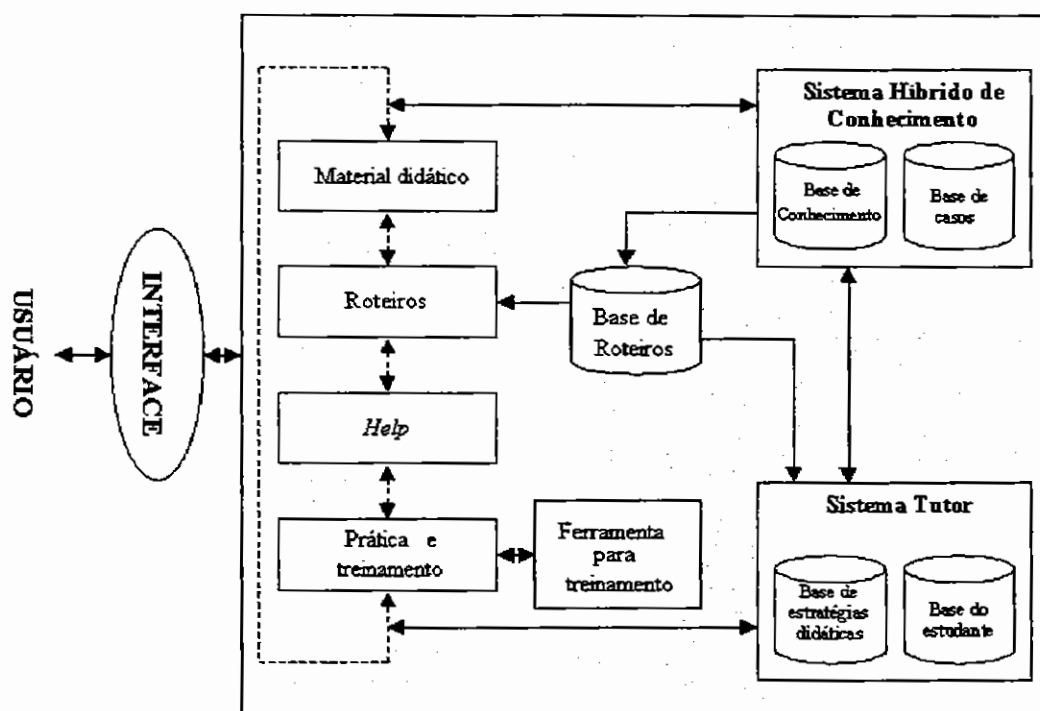


FIGURA 2 — Elementos que compõem o ambiente SIATE.

Fonte: PUGLIESI (1998).

3.3.1 Interface

Através do módulo de interface é possível a interação entre o estudante e o SIATE. O primeiro contato do estudante com o ambiente ocorrerá através de uma página de identificação do estudante. As informações obtidas do estudante são relevantes para a composição da base do estudante, a qual é parte integrante do sistema tutor. Após a identificação do estudante este é

²⁴ MiniSQL foi utilizado para armazenar as informações sobre as páginas que compõem o módulo.

conduzido a um menu principal, onde ele tem acesso a exploração dos módulos material didático, roteiros, prática e treinamento e *help*.

3.3.2 Módulo de material didático

O módulo de material didático tem como objetivo principal inserir o estudante no contexto da arquitetura proposta e também do domínio do sistema. São oferecidas dicas sobre a melhor forma de navegação e utilização do material didático, descrições sobre as características do domínio da aplicação e sobre as características do ambiente, fornecendo ao estudante uma visão geral sobre o ambiente com o qual está interagindo.

As páginas que o estudante visita no módulo de material didático são armazenadas para que o sistema tutor possa fazer sugestões de exercícios e exemplos, baseando-se nas características das páginas pelas quais o estudante passou ou não.

3.3.3 Módulo de roteiros

Neste módulo estão disponíveis os roteiros para navegação direcionada sobre o material didático existente. É composto por dois submódulos que são:

- geração de roteiros, onde o roteiro é gerado a partir de inferências efetuadas pelo sistema híbrido de conhecimento que, baseado no tópico escolhido, nas necessidades e situação atual do estudante, no que diz respeito a sua performance, tenta produzir uma seqüência de páginas que melhor adaptem às necessidades desse estudante; e seleção de roteiros. A geração de roteiros pode ocorrer por iniciativa do estudante ou por sugestão do sistema tutor.
- seleção de roteiros que é efetuada diretamente pelo estudante, o qual selecionará em um menu alguns roteiros definidos pelo autor do material didático.

3.3.4 Módulo de prática/treinamento

Neste módulo estudante pode treinar/praticar dentro do domínio da aplicação, com o apoio especialista do sistema híbrido de conhecimento, utilizando recursos que possibilitem ao estudante praticar, treinar ou até mesmo simular em algum tipo de representação do mundo real.

Na tela desse módulo o estudante poderá acessar informações que estão divididas em quatro classes que são dúvidas, controle, desempenho e utilitários.

3.3.5 Módulo *help*

Neste módulo estão disponíveis informações sobre como utilizar cada um dos módulos do ambiente e também como melhor utilizar os recursos oferecidos pelo sistema tutor e pelo sistema híbrido de conhecimento.

3.3.6 Sistema tutor

O sistema tutor no ambiente SIATE tem como funções:

- identificar através de consulta feita à base do estudante, o nível de conhecimento de cada estudante, para então decidir que nível de informações sugerir a ele;
- analisar o caminho ou páginas percorridas pelo estudante, e decidir se pode tomar alguma decisão sobre aplicar um teste para avaliar o conhecimento do estudante e o que deve conter este teste;
- avaliar através de testes, o desempenho do estudante após este percorrer páginas ou percorrer algum roteiro e armazenar esse desempenho para uso futuro.

O sistema tutor é composto pela base do estudante, base de exercícios, base de exemplos, base de sugestões, gerador de exercícios, gerador de exemplos e gerador de sugestões.

3.3.7 Sistema híbrido de conhecimento

O sistema híbrido de conhecimento oferece apoio especialista sobre o domínio de conhecimento às aplicações do ambiente dentro do módulo prática/treinamento. É responsável por assessorar o estudante, baseado no conhecimento que for explicitado em sua base de conhecimento. O sistema é chamado híbrido por se tratar de uma fusão de técnicas de IA como, Sistemas Baseados em Conhecimento (SBC), técnicas de Raciocínio Baseado em Casos (RBC) e técnicas para a representação de conhecimento como regras, *frames* e casos.

O sistema híbrido de conhecimento do ambiente SLATE é composto por três partes principais a saber: Motor de inferência — agrupa todas as formas de raciocínio contidas no sistema híbrido de conhecimento; uma base de conhecimento — contém o conhecimento do domínio, uma base de casos e o conhecimento para a definição de roteiros dinâmicos e a camada de interação com o usuário — faz perguntas ao estudante, fornece explicações e fornece os roteiros gerados dinamicamente.

3.4 FLUTE

O sistema FLUTE – *Formal Languages and Automata Environment*, é um Sistema Tutor Inteligente desenvolvido para aplicação no domínio de linguagens formais e autômato. O conhecimento no sistema FLUTE é representado a partir da ontologia de STI que tem sido definida dentro de uma arquitetura recentemente desenvolvida e um modelo de STI chamado GET-BITS (*GEneric Tools for Building ITS*)²⁵.

O sistema FLUTE foi implementado como uma aplicação para Windows 98, usando MS Visual C++, projetado como um sistema monousuário, ou seja, uma ferramenta para aprendizado individual.

A arquitetura do sistema FLUTE é mostrada na Figura 3. Segundo o autor nesta figura não estão representados os submódulos de todo os módulos da arquitetura.

²⁵ A arquitetura GET-BITS tem uma visão do Sistema Tutor Inteligente como camadas. Um esquema hierárquico de composição das partes componentes de qualquer STI.

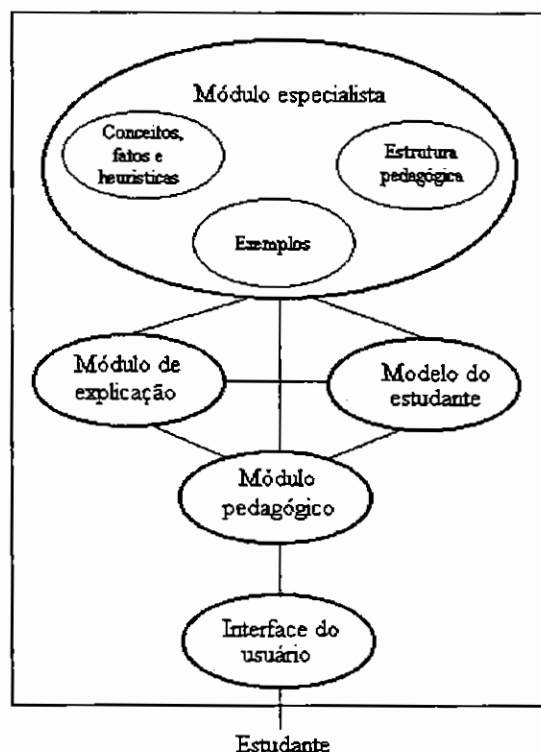


FIGURA 3 — Arquitetura do sistema FLUTE.

Fonte: DEVEDZIC et al. (2000).

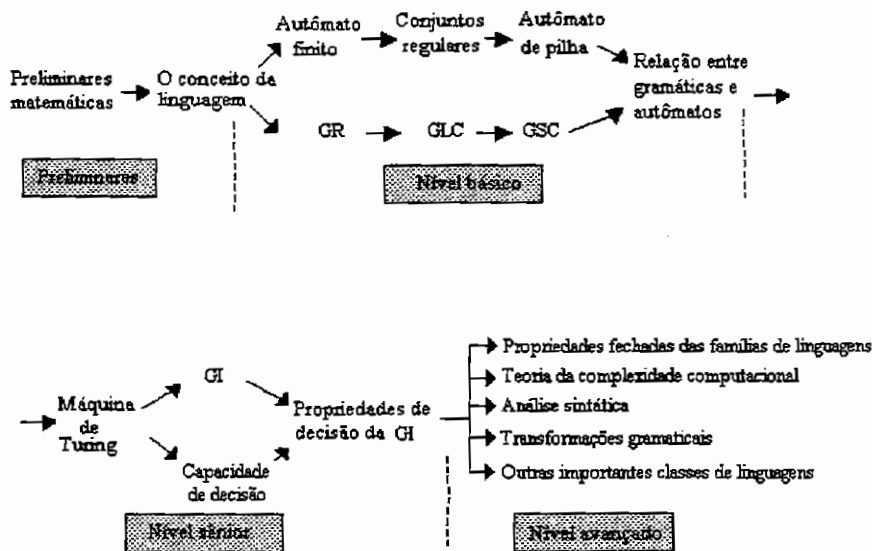
3.4.1 Módulo especialista

O módulo especialista contém o conhecimento do domínio e é composto por três importantes partes. A primeira contém todos os conceitos, tópicos, fatos, conhecimento procedural e a heurística do domínio de o quê o estudante deve aprender. Um significativo subconjunto de tais itens pode ser organizado como uma lição. Alguns atributos importantes de uma lição são estabelecidos como conjunto de objetivos, conjunto de tópicos, conceitos, fatos, teoremas etc., ensinados na qual as lições são um conjunto correspondente de regras de ensino, e um conjunto de problemas associados — testes, questões e exercícios.

Um componente essencial do domínio do conhecimento é uma base de dados de exemplos, os quais são sistematicamente associados a cada tópico, conceito etc., que o estudante

tem que aprender durante a sessão. A apresentação de exemplos para os estudantes esta sobre o controle do módulo pedagógico. Isto torna o processo de ensino, segundo os autores da arquitetura, mais dinâmico e facilita o aprendizado.

A estrutura pedagógica do domínio é definida no FLUTE como um conjunto de gráficos dirigidos que mostram relações de precedência de unidades de conhecimento explicitamente dentro de cada lição e entre os tópicos de lições diferentes. Os gráficos podem definir assuntos de aprendizagem e possíveis caminhos de aprendizagem no domínio. Onde houver uma bifurcação no caminho, o processo de aprendizado pode seguir por qualquer um dos caminhos. Onde houver uma união de dois caminhos, o processo de aprendizagem não pode continuar a menos que o estudante demonstrasse dominar o conhecimento ao longo de ambos os caminhos (isto é conferido pelo módulo pedagógico e é registrado no modelo de estudante).



(GR – Gramáticas regulares; GLC – Gramáticas livres de contexto; GSC – Gramáticas sensíveis ao contexto; GI – Gramáticas irrestritas)

FIGURA 4 — Estrutura pedagógica do domínio.

Fonte: DEVEDZIC et al. (2000).

No sistema FLUTE um gráfico de dependência define o domínio inteiro do sistema, Figura 4. Esse gráfico representa a estrutura dos níveis do domínio específico de linguagens formais e autômatos. São estabelecidos quatro níveis de aprendizagem: preliminares, nível básico, nível sênior e nível avançado. O módulo pedagógico, periodicamente e automaticamente, testa o estudante para conferir seu domínio do conhecimento de domínio associado ao nível que está sendo estudado.

3.4.2 Modelo do estudante

De acordo com os autores deste ambiente, os componentes do modelo do estudante refletem:

- o estado corrente do conhecimento do estudante;
- o progresso corrente do estudante no aprendizado de uma lição particular;
- frequência e duração das tentativas de vários exercícios e solicitações por dicas, explicações e ajuda;
- a performance do estudante na resposta das questões do sistema e resolução de problemas;
- o comportamento do estudante — frequências no seguimento de caminhos corretos/incorretos no aprendizado de um tópico ou um conceito;
- medidas de como continuamente e quão freqüentemente o estudante aprende através do sistema;
- medidas das iniciativas do estudante em trabalhar com o sistema, por exemplo com que frequência o estudante consulta os *links* relacionados aos tópicos.

O modelo do estudante é atualizado depois de cada sessão e é usado no início da próxima sessão. Quando o estudante tem sua primeira sessão com o FLUTE, o agente de modelagem do estudante oferece a ele um conjunto de exercícios e testes antes do processo tutor começar. Deste modo o agente de modelagem do estudante estima um conhecimento de fundo do novo estudante e determina valores iniciais de atributos do modelo do estudante. Embora seja tarefa do agente

de modelagem do estudante atualizar e manter do modelo do estudante, usuários com privilégios especiais, como instrutores humanos, também podem modificar o modelo do estudante eles mesmos.

3.4.3 Módulo pedagógico

O sistema FLUTE sempre opera em um dos seguintes três modos de operação: ensino, exame, consulta. O módulo pedagógico executa um dos agentes pedagógicos a qualquer tempo, levando assim o FLUTE a operar em um dos três modos.

Quando o agente do modo ensino é executado o FLUTE apresenta uma lição para o estudante. No início, o agente executa um planejador baseado em regra e desenvolve um conjunto inicial de ações de ensino. O plano é baseado no conteúdo da lição a ser apresentada, a relevância depende dos gráficos e modelo do estudante. Dependendo das reações do estudante, o agente de modelagem do estudante regularmente atualiza o modelo do estudante. O agente de ensino se comunica com o agente de modelagem do estudante e pode mudar os sub-planos, e se necessário o plano de apresentação de uma lição inteira.

No modo de exame, o agente de exame pode gerar problemas e exercícios para o estudante, oferecer dicas e sugestões para a solução de problemas, e conduzir uma avaliação mais complexa do seu conhecimento, através de uma série de testes e exames. Depois de cada sessão de exame, o agente de exame envia um pedido para o agente de modelagem do estudante para atualizar os atributos relevantes do modelo do estudante (graus, velocidade de resposta, progresso corrente etc.).

O agente do modo de consulta é o que responde as questões do estudante, explica conceitos heurísticos do domínio, repete partes requeridas de uma lição correspondente ao conhecimento atual do estudante, refina soluções de problemas, providencia dicas para o aprendizado futuro etc. O modo de consulta é executado estritamente a pedido do estudante.

3.4.4 Módulo de explicação

O módulo de explicação do FLUTE é um agente baseado em regra. Usa toda informação disponível da base do domínio do conhecimento (conceitos de lições, objetivos, tópicos, gráficos

de dependência e exemplos), bem como as vindas do modelo do estudante, para responder as questões do estudante e prover a explicação desejada. Especificamente, ele é desenhado para aceitar um pedido de explicação vinda do estudante e para dar as seguintes respostas:

- determinar os conteúdos das respostas/explicações;
- decidir sobre o estilo de apresentação da explicação (ilustrações, exemplos, conceitos e referências);
- selecionar o modelo de conhecimento que provê a mais fácil explicação em casos quando múltiplos modelos de conhecimento podem ser usados;
- compor a explicação e ordenar suas declarações em um modo coerente, racional e compreensível.

3.4.5 Interface com o usuário

A interface com o usuário do sistema de FLUTE possui ferramentas de navegação, mostrando gráficos de dependência, esboços da lição e do curso, objetivos, pontos pedagógicos etc., e os parâmetros que modelam o estudante (conhecimento atual e progresso corrente, como posições dentro dos gráficos de dependência correspondentes, desempenho, comportamento de aprendizagem).

3.5 UMA ARQUITETURA BASEADA EM AGENTES PARA SUPORTE AO ENSINO À DISTÂNCIA

A arquitetura apresentada por COSTA (1999) utiliza agentes inteligentes para compor um ambiente de ensino colaborativo para ensino através da internet. No capítulo 2, vimos que os ambientes de ensino colaborativo são ambientes que suportam e mediam trabalho e aprendizado cooperativo em uma rede de computadores.

O ambiente proposto é focado na idéia de que o ensino pode ser alcançado pela exploração de casos e resolução de problemas e também através de troca de explicações e experiências entre os estudantes.

COSTA *ibid.*, propõe uma plataforma multi-agente com diferentes classes de agentes, capazes de interagir entre si e com os seus usuários, permitindo facilidades de colaboração nos níveis agente-agente e agente-usuário. As três classes de agentes definidas, Figura 5, cooperam através de uma rede de computadores e utilizam técnicas diferentes para resolver problemas específicos e interpretar situações. Dois paradigmas foram combinados para orientar o processo de raciocínio dos agentes propostos: sistemas especialistas e raciocínio baseado em casos.

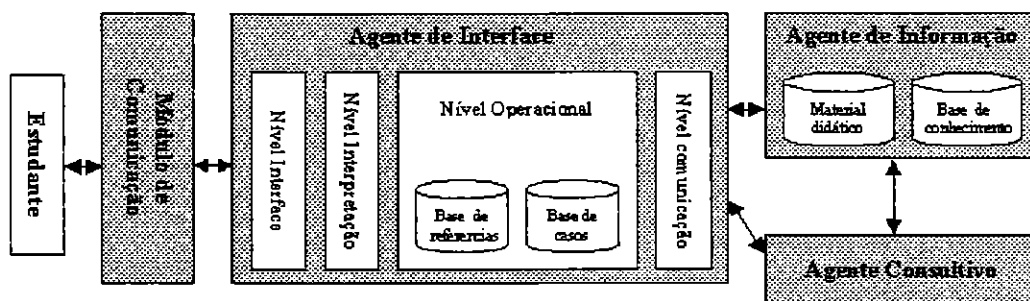


FIGURA 5 — Representação da arquitetura.

A Figura 5 é uma contribuição deste trabalho, e procura representar a arquitetura proposta por COSTA (1999).

3.5.1 O agente de interface

O agente de interface interage com o estudante e coopera com os outros agentes trocando informações sobre capacidades, comprometerimentos e metas de aprendizado dos estudantes. Na estrutura básica de um agente de interface apresentada, Figura 5, é possível verificar quatro níveis bem estruturados:

1. O **Nível de Interface** que oferece a funcionalidade de comunicação com o estudante.
2. O **Nível de Interpretação** que tem a função básica de tradução. Ele procura preparar e ordenar as consultas feitas pelo estudante em uma estrutura de dados interna que é manipulada pelo agente.

3. O **Nível Operacional** que representa a parte principal do agente, pois é nele que reside a funcionalidade de raciocínio. Este nível é responsável pela monitoração das atividades do estudante e pela manutenção do conhecimento deste. Uma base utiliza uma abordagem para aprender e representar a informação sobre preferências do estudante. A segunda base de conhecimento utiliza raciocínio baseado em casos para a construção de uma representação parcial sobre o domínio, ou seja, o conhecimento aprendido.
4. O **Nível de Comunicação** que é a parte do agente que lida com a troca de mensagens com outros agentes. O formalismo adotado neste nível para a representação das mensagens é a linguagem KQML²⁶.

3.5.2 O agente de informação

Neste agente a informação é dividida em duas categorias diferentes: material didático e base de conhecimento. O material didático compreende páginas em formato HTML, imagens, textos e multimídia em geral. Esta categoria de informação é armazenada em um banco de dados relacional e através de um *site* internet.

A base de conhecimento é organizada através de um conjunto de casos e regras. Onde se utilizam regras apenas para a estrutura de recuperação dos casos. Os casos representam as situações a serem exploradas pelos estudantes. Os casos foram modelados através de um formulário com vários atributos.

O agente de interface utiliza as informações do agente de informação como base para a construção de suas próprias interpretações do domínio. O agente consultivo também acessa as informações representadas no agente de informação para guiar o processo de aprendizagem do estudante.

²⁶ Como é apresentado no capítulo 5, a linguagem KQML fornece uma plataforma para programas e agentes trocarem informações e conhecimento, assegurando que dois agentes estejam utilizando a mesma linguagem durante o processo de comunicação.

3.5.3 O agente consultivo

O agente consultivo oferece assistência pessoal ao estudante. Ele assiste o estudante através de uma monitoração constante, indicando a aplicação apropriada do conhecimento. Pode ser consultado indiretamente pelo estudante, através do agente de interface.

Uma importante característica do agente consultivo é que através da monitoração do estudante ele pode reconhecer situações onde é necessário apresentar explicações. E, ainda, a possibilidade de fornecer informações adicionais como exemplos ou contra-exemplos para o mesmo.

3.5.4 O módulo de comunicação

É proposto o uso de um ambiente na forma de um navegador internet inteligente, o qual permite ao estudante acessar o sistema. O ambiente providencia mecanismos que exploram a interação entre os estudantes. O agente de interface passa a ser um componente do navegador inteligente, mantendo toda a responsabilidade de acessar a plataforma multi-agente.

A interface proposta para o navegador inteligente possui quatro áreas básicas:

- **Área de navegação:** espaço na forma de um navegador para a internet. Além de permitir o acesso a qualquer *site* na internet, este navegador também permite abrir arquivos no formato HTML.
- **Área do assistente pessoal:** um personagem animado estabelece a metáfora para a representação do agente de interface e do agente consultivo. O estudante pode interagir com o personagem, solicitando informações adicionais como a recuperação de casos para um determinado tipo de problema.
- **Área de chamada da classe virtual:** esta área apresenta os nomes de todos os estudantes que fazem parte da sala de aula virtual, distinguindo entre os estudantes que estão acessando o ambiente e aqueles que fazem parte do grupo, mas que não estão acessando o ambiente naquele momento.

- **Área de conversação:** o objetivo desta área é registrar todas as interações ocorridas entre o agente e o estudante. Desta forma, o estudante pode, a qualquer momento, revisar o que aconteceu.

3.5.5 Arquitetura de comunicação

Na arquitetura, aqui demonstrada, a comunicação entre os agentes ocorre através da troca de mensagens padronizadas por uma linguagem de comunicação de agentes. Este padrão estabelece que a comunicação não ocorre diretamente entre os agentes, mas através de um programa especial denominado Facilitador. Nessa arquitetura de comunicação são implementadas duas bases de dados locais: na primeira, o Facilitador armazena toda a informação necessária para executar o roteamento das mensagens. Na segunda base ele registra todas as mensagens trocadas.

3.6 CONSIDERAÇÕES FINAIS

Os STIs ainda apresentam deficiências relacionadas a aspectos como a possibilidade de usar de várias estratégias de ensino, o uso de interfaces variadas, um melhor conhecimento do estudante, uma maior autonomia do mesmo. Cada arquitetura, aqui apresentada, tem seu enfoque em um diferente aspecto dos STIs, tentando superar as deficiências aí encontradas.

Os STIs, apresentam ainda, problemas como o alto custo financeiro e o elevado tempo de desenvolvimento, que devem ser amenizados. Abordagens e tecnologias diversas estão sendo usadas para que o desenvolvimento de ambientes de ensino-aprendizagem se torne viável. Na tentativa de reduzir os custos no desenvolvimento, conceitos bem conhecidos da engenharia de *software* como reutilização e modularidade têm sido utilizados. A questão então é desenvolver STIs de forma incremental, permitindo uma evolução contínua (COSTA, 1999).

A análise das arquiteturas para ambientes de ensino-aprendizagem já propostas mostra que as mesmas estão fundamentadas no modelo geral de Sistemas Tutores Inteligentes, uma vez que todas elas apresentam os módulos componentes relativos a tal modelo. Algumas vezes os componentes são apresentados com denominações diferentes, mas com a mesma funcionalidade. E outras vezes, são incluídos novos componentes, mas a função destes, ainda, é a mesma apresentada pelos módulos componentes do modelo geral de um STI.

A finalidade de se analisar arquiteturas já propostas para ambientes de ensino-aprendizagem é encontrar nestas, características e abordagens que se mostrem adequadas para a resolução dos problemas encontrados no desenvolvimento de tais ambientes. As arquiteturas propostas por CASAS (1999) e DEVEDZIC et al. (2000) se mostram interessantes para o desenvolvimento deste trabalho. Estes autores propõem o uso de agentes inteligentes no projeto e construção de ambientes de ensino-aprendizagem na tentativa de melhorar a sua performance.

A arquitetura do ambiente SIATE, proposto por FREIRE (1998) e PUGLIESI (1998), é interessante do ponto de vista de interface com o usuário, uma vez que permite que o mesmo explore livremente o conhecimento do domínio através da navegação em páginas no formato HTML e permite também o acompanhamento das ações executadas pelo estudante nestas páginas.

4 AGENTES INTELIGENTES

As pesquisas em agentes podem ser divididas em duas gerações. A primeira geração de trabalhos abrange o período de 1977-1990, e se concentra principalmente em agentes deliberativos²⁷, os quais possuem modelos simbólicos internos. O foco desta pesquisas estava em assuntos macros tais como a interação e comunicação entre agentes, a decomposição e distribuição das tarefas, coordenação e cooperação, resolução de conflito via negociação etc. O objetivo era especificar, analisar, projetar e integrar sistemas compreendendo a colaboração de múltiplos agentes (NWANA, 1996).

A segunda geração de trabalhos com agentes, que se iniciou 1990 e se segue até os dias atuais é caracterizada por dois extremos: pesquisa e desenvolvimento de teorias, arquiteturas e linguagens de agentes e uma significativa ampliação dos tipos de agentes sendo estudados.

Os agentes, inicialmente desenvolvidos, estavam restritos a um único computador ou no máximo a uma rede homogênea e limitado a tarefas pré-estabelecidas. Atualmente os agentes estão quebrando este confinamento a um só ambiente, apresentando-se móveis e aprendendo a executar tarefas baseados em sua própria experiência.

4.1 DEFINIÇÃO DE AGENTE

O termo agente foi inicialmente definido em trabalhos preliminares na área de Inteligência Artificial, onde pesquisadores dedicavam-se em tentar reproduzir uma entidade artificial que imitasse as habilidades humanas.

Embora usado com frequência em Inteligência Artificial, o termo agente não possui uma noção uniforme e precisa para o seu significado. O termo é vastamente usado por muitos pesquisadores envolvidos com pesquisas em áreas relacionadas. Isto desafia a tentativa de produzir uma definição universalmente aceita, uma vez que, a definição de agente é determinada, em grande parte, pelo campo de pesquisa em questão.

²⁷ Um agente deliberativo possui uma representação explícita de seu comportamento e as decisões (ações a executar) são realizadas através de um raciocínio simbólico.

Serão apresentadas a seguir algumas das diversas definições de agente encontradas na literatura, apenas com o intuito de demonstrar as diferentes definições existentes.

FRANKLIN & GRAESSER (1996) definem agente como sendo um sistema que faz parte de um ambiente onde ele percebe e atua para atender suas próprias metas. Uma definição bastante similar é a apresentada por RUSSEL & NORVIG (1995), onde um agente é definido como sendo algo que pode ser visto como percebendo seu ambiente através de sensores e atuando sobre este ambiente através de efetadores.

Na definição de agentes apresentada por SHOHAM (1997), um agente é uma entidade de *software* funcionando continuamente e de forma autônoma em um ambiente particular, freqüentemente habitado por outros agentes e processos.

NWANA (1996) define agente como sendo um componente de *software* e/ou *hardware* o qual é capaz de atuar exatamente para realizar tarefas em nome do seu usuário.

Uma definição mais completa de agente é dada por WOOLDRIDGE (2000), onde um agente é uma entidade computacional tal como um programa de *software* ou um robô que pode ser visto como percebendo e atuando sobre seu ambiente e é autônomo uma vez que, seu comportamento depende de suas próprias experiências. Como uma entidade inteligente ele opera flexivelmente e racionalmente em uma variedade de circunstâncias ambientais de acordo com seu equipamento perceptual e efetivo. Flexibilidade comportamental e racionalidade são atingidas pelo agente com base em processos chave como a resolução de problemas, planejamento, tomada de decisão e aprendizagem.

As definições de agente encontradas, normalmente estão associadas a diferentes pontos de vista e dependem muito da funcionalidade fornecida pelo agente em questão. Entretanto, a visão de vários conceitos auxilia no processo de entendimento de aspectos importantes que envolvem o termo agente. Um dos conceitos implícitos em todas as definições apresentadas acima é o conceito da autonomia, onde um agente deve agir por si mesmo atuando sobre o ambiente a partir das suas percepções do mesmo.

4.2 PRINCIPAIS CARACTERÍSTICAS DE UM AGENTE

Com base nas definições de agente é possível extrair algumas características que os agentes devem possuir como: estar inseridos em um ambiente; sentir e atuar sobre o ambiente e responder segundo os objetivos para os quais foram projetados. Portanto um agente deve possuir uma série de características ou propriedades que o defina como agente.

As características de agentes podem ser divididas em duas categorias. A primeira está associada a uma noção fraca sobre agentes que define a abordagem inicial das pesquisas na área, e a segunda associa-se a uma noção forte do termo agente (WOOLDRIDGE & JENNINGS, 1994).

4.2.1 Características da noção fraca de agentes

As propriedades que compõem a noção fraca do termo agente são: autonomia, habilidade social, reatividade, pró-atividade e continuidade temporal. Segundo WOOLDRIDGE & JENNINGS (1995), de um modo geral a maioria das vezes em que o termo agente é usado para se referir a um *hardware* ou mais usualmente um *software* ele demonstra as propriedades de noção fraca de agente. A seguir será descrita cada uma destas características.

Autonomia

Autonomia refere-se ao princípio de que os agentes podem agir baseados em seus próprios princípios, sem a necessidade de serem guiados por humanos (NWANA, 1996).

Segundo WOOLDRIDGE (2000), dizer que os agentes são autônomos significa que em alguma extensão eles têm controle sobre seu comportamento e podem atuar sem a intervenção de humanos ou outros sistemas. Os agentes autônomos perseguem seus objetivos e para alcançá-los realizam tarefas seqüenciais, de acordo com o seu conhecimento sobre o domínio.

Uma vez que o usuário ativa o agente indicando algum objetivo a ser realizado, este atua independentemente, selecionando estratégias e monitorando o progresso em busca da meta. Se uma estratégia falhar, usará outra, mas sem a intervenção humana.

Um exemplo da autonomia dos agentes é quando estes são usados para manter uma agenda pessoal independente de seu usuário. Isto requer aspectos de ação periódica, execução espontânea e iniciativa, na qual o agente deve ser capaz de executar ações preemptivas e independentes que irão eventualmente beneficiar o usuário.

Habilidade social

A habilidade social está relacionada à capacidade do agente interagir com outros agentes e, possivelmente humanos, para solicitar informação ou também para expor os resultados obtidos na execução de alguma tarefa agendada (WOOLDRIDGE & JENNINGS, 1994).

A característica de habilidade social nos agentes está diretamente associada com a característica de comunicabilidade. Quando existe mais de um agente envolvido, há uma necessidade óbvia por um modelo de comunicação. Entretanto, o conceito de comunicabilidade não estabelece apenas a troca de informações entre agentes, eles podem se comunicar com outras entidades, incluindo-se humanos e o seu ambiente.

Existem aplicações onde os agentes devem acessar informações de outras fontes, sobre o atual estado do ambiente, durante o curso de suas tarefas. Isto requer a habilidade de comunicar-se com os repositórios destas informações. Estes repositórios podem ser outros agentes ou bases de dados.

Reatividade

Reatividade é a habilidade que um agente deve possuir para sentir o estado do ambiente no qual ele está inserido e em função disto atuar, respondendo de maneira adequada às mudanças ocorridas no ambiente. Portanto, um agente deve ser capaz de perceber seu ambiente e atuar sobre ele. Este atributo está presente em praticamente todas as definições de agente (WOOLDRIDGE & JENNINGS, 1994); (RUSSELL & NORVIG, 1995); (FRANKLIN & GRAESSER, 1996).

O ambiente ao qual o agente deve perceber e reagir pode ser o mundo físico, o usuário via interface gráfica, outros agentes, a internet ou uma combinação de todos estes.

A característica de reatividade está presente nos agentes de filtragem de e-mail, onde, segundo FRANKLIN & GRAESSER (1996), o agente se encontra a maior parte do tempo em

uma espécie de “estado de dormência”, do qual ele é despertado quando algo muda em seu ambiente, ou seja, quando chega um e-mail.

Orientação a objetivos ou pró-atividade

Este atributo pode ser também denominado iniciativa, uma vez que representa um comportamento independente. As ações são dirigidas pelo objetivo e não simplesmente por mudanças no seu ambiente. O agente que implementa este atributo possui maior flexibilidade, pois é capaz de resolver problemas causados por situações inesperadas.

O atributo de pró-atividade em um agente habilita-o a não, apenas simplesmente, reagir em resposta ao seu ambiente, e sim torná-lo capaz de exibir um comportamento orientado a objetivos, tomando a iniciativa.

A pró-atividade é o elemento chave da autonomia, que é a sua habilidade de tomar iniciativas, sem a necessidade de agir em virtude de uma mudança de seu ambiente (WOOLDRIDGE & JENNINGS, 1994).

Continuidade temporal

Possuir continuidade temporal significa que os agentes são processos que estão continuamente rodando, podendo estar ativos ou passivos. Diferente de um programa convencional do qual se conhece o início e o fim, um agente deve se auto-executar até que seja alcançado corretamente o objetivo solicitado, enquanto seu ciclo perdure ou seu usuário não o detenha. A continuidade temporal, segundo TOLOSA & BORDIGNON (1999), é a propriedade que dá vida ao agente, possibilitando que se mantenha alerta a uma solicitação de alguma mudança no ambiente.

4.2.2 Características da noção forte de agentes

A noção forte significa que as propriedades possuídas pelos agentes identificadas na noção fraca de agentes são conceituadas ou implementadas sobre algumas propriedades aplicadas aos humanos. Por exemplo, noções como conhecimento, crença, intenção, obrigação e assim por diante (estados mentais). Várias outras propriedades então, são atribuídas aos agentes, como: aprendizagem, mobilidade, cooperação e racionalidade.

De acordo com WOOLDRIDGE & JENNINGS (1995), os agentes que possuem as habilidades ditas fortes devem possuir uma ou mais das características descritas a seguir.

Aprendizagem

Um dos atributos que mais caracterizam agentes inteligentes é a capacidade de aprender. Segundo WOOLDRIDGE (2000), uma real autonomia só pode estar presente quando um agente possui a habilidade de avaliar as variações de seu ambiente externo e escolher qual a ação mais correta. Entretanto, mesmo quando um agente não reconhece nenhuma ação a ser executada é esperado que ele procure encontrar uma saída. A questão não é acertar sempre, mas aprender continuamente por experiência, seja através de sucessos ou de fracassos.

Um agente é capaz de aprender quando possui a capacidade de acumular conhecimento, onde este pode lhe ser fornecido através de uma seqüência de instruções ou baseado em experiência passada, e partir daí modificar seu comportamento em resposta às novas situações.

Mobilidade

A mobilidade diz respeito à habilidade de um agente em se mover por uma rede de computadores.

A capacidade de poder se mover através de uma rede de computadores parece ser interessante para agentes que auxiliam seus usuários na busca de informações, principalmente na internet. A mobilidade capacita o agente a viajar pelas redes de computadores, de nó a nó, em busca de recursos que o permita cumprir com a sua agenda (NWANA, 1996).

Este atributo, entretanto, pode causar sérios problemas de sobrecarga na rede, uma vez que eles trafegam entre as máquinas conectadas. Um outro problema está relacionado com segurança, uma vez que um agente móvel pode conter problemas de código ou até mesmo estar transportando um vírus de computador. Sendo assim, a implantação de agentes móveis deve ser acompanhada de processos de autorização, além da garantia de que a memória e os recursos da máquina estarão protegidos.

Cooperação

Cooperação pode ser entendida como a capacidade que os agentes têm de trabalhar em conjunto de forma a concluírem tarefas de interesse comum. A idéia é que eles trabalhem juntos para realizar tarefas complexas, aproveitando-se das particularidades de cada um para cumprir seus objetivos.

A cooperação pode ser vista como uma extensão natural da habilidade social ou comunicabilidade, uma vez que a cooperação é resultado da interação entre vários agentes e/ou humanos e que para tal deve ser especificada uma linguagem de comunicação própria.

OLIVEIRA & VICCARI (1996) enfatizam que a cooperação entre agentes é fundamental, sendo a razão principal para a existência de um ambiente multi-agente.

Racionalidade

A racionalidade permite ao agente agir de forma a alcançar seus objetivos e nunca atuar de modo que impeça que esse objetivo seja alcançado. Ou seja, um agente nunca realizará tarefas se estas vão contra as suas crenças.

Para atingir a racionalidade o agente utiliza a sua capacidade de raciocínio ou inferência, que é talvez o aspecto mais importante que distingue um agente inteligente de outros agentes/*softwares* (FRANKLIN & GRAESSER, 1996). Dizer que um agente tem raciocínio significa dizer que ele tem habilidade de inferir, baseado no conhecimento atual e experiências, numa maneira racional e reprodutível.

É importante ressaltar que as características apresentadas não estão em ordem de importância. A escolha de quais características devem estar presentes em um agente depende da funcionalidade que o projetista pretende dar ao agente.

Segundo, FRANKLIN & GRAESSER *ibid.*, nenhuma destas características sozinha estabelece que o *software* é um agente inteligente. E que a maioria dos agentes possuem as seguintes propriedades: autonomia, reatividade, pró-atividade e continuidade temporal. WOOLDRIDGE & JENNINGS (1995) acrescentam ainda que um agente precisaria também

possuir a habilidade social. As outras propriedades produzem novas classes de agentes em potencial.

4.3 TIPOS DE AGENTES

A análise das características que estão presentes nos agentes tem sido utilizada pelos pesquisadores para organizar os agentes em tipos, classificando os agentes por características que estes têm em comum.

A classificação dos tipos de agentes existentes é apontada por diversos autores, baseada na funcionalidade do agente e na forma de atuação.

NWANA & NDUMU (1998) propõem uma tipologia de agentes que identifica diferentes dimensões de classificação. Eles acreditam que agentes podem ser classificados de acordo com:

1. Sua mobilidade, podem ser estáticos ou móveis. No caso de ser móvel, o agente pode ainda estar residente na máquina do cliente ou temporariamente no servidor;
2. A presença ou não de um modelo de raciocínio simbólico, ou seja, um agente pode ser deliberativo ou reativo. Um agente deliberativo possui um modelo de raciocínio simbólico interno, e ele se ocupa do planejamento e negociação com outros agentes para alcançar suas metas. Um agente reativo não tem qualquer modelo simbólico interno do seu ambiente, e ele atua usando um comportamento tipo estímulo/resposta respondendo ao estado presente do ambiente no qual ele está inserido;
3. A presença de atributos mínimos que eles deveriam exibir, como: autonomia, cooperação e aprendizagem. Combinando estas três características, quatro tipos de agentes podem ser derivados: agentes colaborativos, agentes colaborativos com capacidade de aprendizagem, agentes de interface e agentes verdadeiramente inteligentes. É importante notar que os limites desta classificação não devem ser interpretados como linhas bem definidas. Na verdade, o fato de agentes cooperativos terem mais ênfase em cooperação e autonomia que agentes com capacidade de aprendizagem não exclui a possibilidades daqueles desenvolverem características de aprendizado;

4. Pelos papéis que desempenham (particularmente, se os papéis são principais), por exemplo agentes de *gathering*²⁸ de informação na internet. Essencialmente, tais agentes ajudam a gerenciar uma vasta quantia de informação em redes de longa distância como a internet. Pode-se referir a esta classe de agentes como agentes de informação ou agentes de internet. Agentes de informação podem ser estáticos ou móveis, e deliberativos ou reativos;
5. Pela combinação de duas ou mais filosofias de agentes em um único agente, a qual comporia a categoria de agentes híbridos.

Após estabelecer as diferentes dimensões para a classificação de agentes, NWANA & NDUMU *ibid.* estabeleceram sete categorias de agentes: agentes colaborativos, agentes de interface, agentes móveis, agentes de informação, agentes reativos, agentes híbridos e agentes verdadeiramente inteligentes.

A seguir serão apresentados as características, papéis e alguns exemplos de aplicações de cada uma das seis primeiras classes de agentes estabelecidas. As características dos agentes verdadeiramente inteligentes não são apresentadas, por estes ainda se tratarem, segundo NWANA & NDUMU *ibid.*, de uma aspiração dos pesquisadores e não uma realidade

4.3.1 Agentes colaborativos

Os agentes colaborativos enfatizam autonomia e cooperação com outros agentes para executar tarefas para seus usuários, onde cada agente contribui com sua própria técnica inteligente para a solução de um problema complexo.

Os agentes colaborativos tendem a ser agentes estáticos. Eles podem aprender, mas este aspecto não é tipicamente uma ênfase principal de sua operação, entretanto alguns executam aprendizagem de hábitos. Para coordenar suas atividades, eles podem ter que negociar para alcançar acordos mutuamente aceitáveis e para que essa negociação ocorra é clara a necessidade de se definir uma linguagem comum para a comunicação entre agentes, seção 5.2.

²⁸ Esses agentes buscam e juntam informações espalhadas na internet.

A razão para se ter sistemas de agente colaborativos é uma especificação do objetivo da Inteligência Artificial Distribuída, seção 4.4. A IA Distribuída pode ser vista como responsável pela criação de sistemas que interconectam agentes colaborativos separadamente desenvolvidos e que permite que o conjunto funcione além das capacidades de quaisquer de seus integrantes. Algumas outras motivações para pesquisas em IA Distribuída, e conseqüentemente a pesquisa nos agentes colaborativos, incluem:

1. Resolver problemas que são muito grandes ou complexos para um único agente centralizado resolver, devido às limitações de recursos ou o risco de ter um sistema centralizado;
2. Permitir a interconexão e interoperação de sistemas já existentes, por exemplo sistemas especialistas, sistemas de apoio de decisão, programas convencionais etc;
3. Fornecer soluções para problemas inerentemente distribuídos. Por exemplo, soluções as quais necessitam de fontes de informação distribuídas como fontes de informação *on-line* distribuídas ou redes de sensores distribuídos.

4.3.2 Agentes de interface

Agentes de interface enfatizam a autonomia e aprendizado para executar tarefas para seus usuários. O agente de interface pode ser metaforicamente chamado de um assistente pessoal, o qual colabora com o usuário no mesmo ambiente de trabalho.

Este tipo de agente atua normalmente em *background*, analisando as ações do usuário, encontrando padrões repetitivos e automatizando estes padrões com a aprovação do usuário. O agente observa e monitora as ações realizadas pelo usuário na interface, aprende novos atalhos, e sugere modos melhores de fazer a tarefa. Os agentes de interface aprendem a ajudar os seus usuários, tipicamente, nos seguintes quatro modos: (i) observando e imitando o usuário; (ii) recebendo *feedback* positivo e negativo do usuário; (iii) recebendo instruções explícitas do usuário e (iv) pedindo conselho a outros agentes.

Segundo NWANA & NDUMU (1998), são especificadas duas precondições a serem cumpridas através de programas de aplicação para agentes de interface. Primeiro, que haja um

significativo comportamento repetitivo no uso da aplicação (caso contrário, o agente não poderá aprender qualquer coisa) e, segundo, que este comportamento repetitivo seja potencialmente diferente para usuários diferentes.

Os agentes de interface oferecem alguns benefícios gerais como: proporcionar menos trabalho para o usuário final; poder se adaptar, com o passar do tempo e de acordo com as preferências e hábitos de seu usuário; compartilhar a perícia de diferentes usuários em uma comunidade.

São encontrados na literatura, subtipos de agentes de interface, devido as diferentes tarefas para as quais eles são construídos. Os mais comuns são:

1. Assistentes: trabalham realizando tarefas típicas como o gerenciamento da agenda do usuário, ajudando-o a planejar as reuniões. Suas ações incluem negociar, aceitar ou recusar reuniões.
2. Filtros: sua tarefa principal é a de analisar informações segundo um conjunto de regras dadas pelo usuário. Uma aplicação típica é o filtro de mensagens de correio eletrônico.
3. Guias: assistem o usuário no uso de uma aplicação, monitoram as ações do usuário e tentam sugerir passos a serem seguidos para se alcançar um objetivo.

A principal crítica de agentes de interface é que eles tendem a funcionar em modo *stand-alone* ou, no máximo, só se engajam em comunicação de tarefa restrita e específica com colega idêntico. Isto não é necessariamente ruim mas seria útil ter agentes de interface que pudessem negociar com seus colegas como com agentes colaborativos (Nwana & Ndumu *ibid.*).

4.3.3 Agentes móveis

Agentes móveis são *softwares* capazes de percorrer redes de longa distância (WANs) como a internet, interagindo com *hosts* externos, juntando informação em nome dos seus usuários e retornando com as tarefas a eles estabelecidas executadas. Estas tarefas podem variar desde fazer uma reserva de vôo a administrar uma rede de telecomunicações.

Na tecnologia de agentes o conceito de mobilidade permite que um agente se mova de um ambiente hospedeiro, um *host*, para outro carregando consigo seu código e seu estado para outra máquina, sendo capaz de terminar sua execução nesta. Para que um agente possa ser executado em diferentes *hosts* em uma rede, cada *host* deve fornecer suporte de *software* especializado. Este *software* permite que o agente acesse os recursos do *host* que ele precisa para comunicação, computação e navegação. Ou seja, deve haver uma infraestrutura de transporte que mova o código do agente de uma localização para outra.

A hipótese chave que está por trás dos agentes móveis é a idéia de que em certas aplicações, eles provêem um número vantagens que escapam aos agentes estáticos. Por exemplo, imagine ter que carregar muitas imagens para escolher apenas uma. Não é mais natural adquirir um agente para ir a aquela localização, fazer uma busca local e só transferir a imagem escolhida através da rede?

Os principais desafios da pesquisa e desenvolvimento de agentes móveis, segundo NWANA & NDUMU (1998) incluem:

1. Locomoção: como um agente faz se empacota e se move de lugar para outro?
2. Autenticação: como se assegurar que o agente é quem diz ser, e está representando quem reivindica ser representado?
3. Segredo: como assegurar que o agente mantenha sua privacidade? Como assegurar que outros não leiam seu agente pessoal e o executem para seus próprios benefícios? Como assegurar que seu agente não seja eliminado e seu conteúdo apagado?
4. Segurança: como se proteger contra vírus? Como impedir que um agente invasor entre em um *loop* infinito e consuma todos os ciclos de CPU?
5. Desempenho: qual seria o efeito de estar usando centenas, milhares ou milhões de tais agentes em uma WAN?

4.3.4 Agentes de informação

Os agentes de informação, também conhecidos como agentes de internet ou agentes Web, executam o papel de administração, manipulação ou conferência de informação de forma inteligente em muitas fontes de informação distribuídas. É importante ressaltar que estes agentes não devem ser confundidos com simples mecanismos de busca utilizados na internet. O objetivo não é simplesmente encontrar informações que satisfaçam um conjunto de palavras-chave, mas espera-se que este tipo de agente possa reconhecer padrões de informação e encontrar aquelas mais relevantes. Além disso, este agente deve poder operar em modo autônomo, realizando filtragens e em alguns casos aplicando inferências. Neste caso, o agente consegue transformar pedaços de informação em conhecimento altamente produtivo para seu usuário.

Agentes de informação possuem métodos para permitir o rápido acesso e recuperação de informação relevante. Podem ter mecanismos de busca e navegação flexíveis e algoritmos de classificação poderosos.

“a motivação para o desenvolvimento de agentes de informação se dá por dois motivos. Primeiro, há simplesmente uma necessidade por ferramentas para gerenciar a explosão de informação na internet. Segundo, haverá enormes benefícios financeiros para quem construir o primeiro gerente de informação utilizável na internet que seja pró-ativo, dinâmico, adaptativo e cooperativo” (NWANA & NDUMU, 1998, p. 37)

Os agentes de internet poderiam ser móveis porém, esta não é ainda uma regra. São tipicamente estáticos uns são embutidos dentro de um navegador de internet e usam um *host* de ferramentas de administração de internet como *Spiders*²⁹ e máquinas de busca para juntar a informação.

O principal problema com agentes de informação estáticos está na manutenção atualizada de seus índices, uma vez que o ambiente da internet é muito dinâmico. Por isto, é provável que a maioria de futuros agentes de informação seja da variedade móvel. Eles poderão navegar na rede e armazenar sua topologia em um banco de dados, ou seja, no seu *site* local.

²⁹ Os *Spiders* criam uma rede de fontes de informação, em um *host* específico, e percorrem essas fontes em busca de informações solicitadas.

Espera-se que os agentes de informação sejam uma das principais áreas de crescimento nos próximos anos.

4.3.5 Agentes reativos

Agentes reativos representam uma categoria especial de agentes que não possuem modelos internos, simbólicos dos seus ambientes; ao invés disto eles respondem de uma maneira de incentivo/resposta ao estado presente do ambiente no qual eles são embutidos.

As três idéias chave que suportam os agentes reativos, de acordo com NWANA & NDUMU *ibid.*, são:

1. Funcionalidade emergente: os agentes reativos são relativamente simples e eles interagem com outros agentes de modos básicos. Não obstante, padrões complexos de comportamento emergem destas interações quando o conjunto de agentes é visto globalmente.
2. Decomposição de tarefa: um agente reativo é visto como uma coleção de módulos que operam autonomamente e são responsáveis por tarefas específicas (por exemplo sentido, controle motor, computações etc.). A comunicação entre os módulos é minimizada e de natureza de baixo nível. Nenhum modelo global existe dentro de quaisquer dos agentes e, conseqüentemente, o comportamento global tem que emergir.
3. Agentes reativos tendem a operar em representações as quais são próximas a um rude sensor de dados, em contraste às representações simbólicas de alto nível que abundam nos outros tipos de agentes.

A hipótese de agentes reativos pode ser declarada como segue: “podem ser desenvolvidos sistemas de agentes inteligentes com agentes simples que não têm modelos simbólicos internos, e dos quais a inteligência deriva do comportamento emergente das interações dos vários agentes” (NWANA & NDUMU, 1998, p. 38).

4.3.6 Agentes híbridos

Segundo NWANA & NDUMU (1998), é difícil e até improdutivo um debate sobre qual dos tipos de agentes, colaborativos, de interface, moveis, de informação e reativos, é o melhor. Mesmo porque a escolha do tipo de agente e quais propriedades são exibidas pelo agente depende da aplicação ou papel que o mesmo realiza. Considerando também, que cada tipo tem ou promete suas próprias vantagens e deficiências, uma solução é maximizar as vantagens e minimizar as deficiências da técnica mais pertinente para um propósito em particular. Um modo de fazer isto é adotar uma aproximação híbrida, onde os agentes híbridos são aqueles cuja constituição é uma combinação de duas ou mais filosofias de agentes dentro de um agente singular.

Agentes podem também ser definidos operacionalmente em termos de domínios no qual fornecem seus serviços, incluindo busca por informação, filtragem de dados, ajuda sensível ao contexto, assistência em tempo real, além da execução de tarefas conforme as necessidades de um usuário e outros. Neste contexto, uma abordagem possível para definir agentes pode ser através de sua organização, seguindo algum tipo de classificação.

4.4 INTELIGÊNCIA ARTIFICIAL DISTRIBUÍDA

Agentes operam e existem em algum tipo de ambiente, estes ambientes podem ou não conter outros agentes. Embora existam situações onde um agente pode operar utilmente por si mesmo, o aumento da interconexão e a operação de computadores em rede estão tornando tais situações raras. O interesse por paralelismo e distribuição na IA, tornou a Inteligência Artificial Distribuída ou IA Distribuída, uma área emergente nas últimas décadas.

Segundo OLIVEIRA & VICCARI (1996), a atenção da IA Distribuída é focada na interação, cooperação, (isto é, distribuição dinâmica de controle e tarefas) e fluxo do conhecimento entre entidades logicamente distintas. A IA Distribuída pode ser entendida como o estudo do comportamento inteligente computacionalmente alcançado através da interação de múltiplas entidades, cada uma capaz de algum grau de ação autônoma. Tais entidades são usualmente referidas como agentes.

“A Inteligência Artificial Distribuída se propõe a estudar e desenvolver técnicas que permitam a um grupo de entidades, fracamente acopladas, distribuídas ao longo de uma rede de computadores, colaborar na resolução de um problema global. A complexidade das entidades que compõem um sistema de IA Distribuída varia muito, podendo ser um simples elemento de processamento ou um programa complexo capaz de comportamento racional” (PRADO, 1996, s/n).

Segundo HUHNS & STEPHENS (2000), existem aplicações nas quais a abordagem centralizada se torna impossível, necessitando-se de uma abordagem distribuída. Nestas aplicações a informação envolvida é necessariamente distribuída e reside em sistemas que são grandes e complexos em diversos sentidos:

- eles podem estar geograficamente distribuídos;
- eles podem ter muitos componentes;
- eles podem ter um enorme conteúdo;
- eles podem ter um amplo escopo, abrangendo a maior parte de um domínio significativo. Além disso, os componentes podem ser heterogêneos. Neste tipo de sistema onde o problema a ser resolvido tende a ser essencialmente distribuído, a computação distribuída é algumas vezes mais fácil de ser entendida e mais fácil de ser desenvolvida.

Ainda segundo HUHNS & STEPHENS *ibid.*, há quatro grandes técnicas para trabalhar com o tamanho e a complexidade de alguns sistemas: modularidade, distribuição, abstração e inteligência. O uso de inteligência e módulos distribuídos combina todas essas quatro técnicas. Essa combinação é fornecida pela abordagem da Inteligência Artificial Distribuída.

Em um sistema de Inteligência Artificial Distribuída, cada agente é capaz de resolver problemas e trabalhar isoladamente. Porém, existem problemas que são de solução impossível sem a cooperação dos demais agentes da rede. A cooperação é necessária porque um único agente pode não reunir os recursos necessários para resolver o problema em questão. Estes recursos podem ser de *hardware*, *software* ou informações sobre o domínio da aplicação. Desta maneira, diferentes agentes podem ter os recursos necessários para resolver diferentes partes de

um problema mais geral. Durante o processo de resolução, os agentes cooperam usando seus recursos para resolver problemas locais, encontrando soluções que são, posteriormente, integradas para compor a solução do problema global.

“As pesquisas em IA Distribuída podem ser caracterizadas e entendidas através do seguinte paradigma: existe um conjunto de agentes tentando resolver um único problema. Embora haja um estado e um objetivo global para o problema, cada agente tem destes apenas uma visão parcial e imprecisa. Sendo assim, cada agente procura entender, prever e influenciar o estado global segundo sua visão local dos objetivos do sistema. A comunicação entre os agentes é permitida e é através dela que são trocadas soluções, tarefas, objetivos e estimativas do estado global do problema” (PRADO, 1996, s/n)

A Inteligência Artificial Distribuída é dividida em duas grandes áreas de pesquisa, de acordo com PRADO (1996):

1. sistemas para **resolução distribuída de problemas** (DPS - *Distributed Problem Solving*): que estuda como o trabalho para resolver um problema pode ser dividido em um conjunto de nós, módulos ou agentes, que dividem e compartilham conhecimento sobre o problema e sobre o estado de sua solução;
2. sistemas **multi-agente** (MAS - *Multi-Agent Systems*): que procura modelos e técnicas que coordenem as atividades de um grupo de agentes autônomos, de forma que trabalhem de forma inteligente e cooperativa.

Nos sistemas de Resolução Distribuída de Problemas, inicialmente define-se um objetivo global e o problema passa a ser projetar entidades distribuídas capazes de executar este objetivo global. Em Sistemas Multi-Agente, definem-se inicialmente as entidades autônomas e descentralizadas. Estuda-se, então, como estas entidades podem executar tarefas, que podem ser individuais ou podem ser de interesse de várias entidades.

Antes de se prosseguir com a descrição destas duas áreas, é necessário ressaltar que, como enfatizado por PRADO *ibid.*, os termos Resolução Distribuída de Problemas e Sistemas Multi-Agente, foram criados para rotular uma série de sistemas e pesquisas, que possuem aspectos em comum. Desta forma, é difícil determinar os limites desta classificação. Na verdade, não há

ainda um consenso na comunidade de IA Distribuída se estas duas áreas são complementares ou se uma delas está contida na outra.

4.4.1 Resolução Distribuída de Problemas *versus* Sistemas Multi-Agente

Na Resolução Distribuída de Problemas a atividade de solução de um problema em particular pode ser distribuída por vários agentes que cooperam em nível de divisão e compartilhamento do conhecimento sobre o problema e sobre o desenvolvimento de uma solução. Nesta abordagem, os agentes são designados para resolver um problema específico. Embora haja identificável estado global e um objetivo global, cada agente tem uma incompleta e possivelmente inexata visão do problema. Cada agente visa reconhecer, prognosticar e transformar o estado global em um modo que sua visão local dos objetivos globais é satisfeita. Quando na construção do sistema, a principal preocupação do projetista do sistema é alcançar a melhor performance na resolução do problema através da exploração do paralelismo intrínseco para o problema e/ou da solução do problema (OLIVEIRA & VICCARI, 1996).

De outro lado, em Sistemas Multi-Agente o projetista não está primariamente preocupado com o problema específico. A abordagem é sobre coordenação do comportamento inteligente entre um conjunto de agentes autônomos (possivelmente pré-existent). A busca por autonomia rende arquiteturas mais flexíveis, onde a organização do sistema é assunto para mudanças de acordo com as variações no ambiente e/ou no problema a ser resolvido.

De acordo com PRADO (1996), um Sistema de Resolução Distribuída envolve, tipicamente, um conjunto de processos sendo executados simultaneamente de forma concorrente, compartilhando recursos físicos ou arquivos em geral, sendo este o principal motivo da interação dos processos. O objetivo destes sistemas é preservar a ilusão de que cada processo é único e está sendo executado em um sistema dedicado. A decomposição das tarefas é feita de forma que cada agente raramente necessita do auxílio de outro, para executar a tarefa a ele atribuída.

Nos MAS os agentes especialistas na atividade que desempenham não trabalham isolados, mas de forma cooperativa tentando resolver um problema da melhor forma possível (HUHNS & STEPHENS, 2000).

Estas diferenças se tornam mais claras quando se estudam as interações dos agentes que ocorrem em ambas as abordagens, PRADO *ibid.* Em um sistema de Resolução Distribuída de Problemas tem-se:

- diferentes processos executados de forma concorrente disputando recursos físicos e lógicos, sendo esta a principal razão da interação;
- objetivo de preservar a ilusão de que cada processo está sendo executado separadamente em um sistema dedicado.

Em um Sistema Multi-Agente tem-se:

- os diversos processos executados simultaneamente, compartilhando recursos e dividindo tarefas;
- como objetivo tirar proveito da distribuição, espacial ou funcional, dos agentes da rede, deve ser possível a comunicação entre estes agentes e, estes por sua vez, devem se tornar responsáveis pelo controle das atividades.

Segundo DURFEE & ROSENSCHEIN³⁰ apud COSTA (1999), sistemas de Resolução Distribuída procuram trabalhar na base da confiança, sinceridade, divisão de tarefas e outras propriedades que são muito difíceis de serem alcançadas em uma coleção de indivíduos. Por outro lado, os Sistemas Multi-Agente se baseiam na teoria dos jogos para demonstrar que um agente precisa ser racional, ou seja, todas as suas ações devem ser no sentido de maximizar o seu próprio custo/benefício.

Entretanto, COSTA (1999) afirma que o trabalho de Durfee & Rosenschein não procurou separar totalmente sistemas de Resolução Distribuída dos Sistemas Multi-Agente, mas apresentou três visões de relacionamento entre eles:

1. Em uma primeira proposta, sistemas DPS podem ser vistos como sendo um subconjunto dos Sistemas Multi-Agente. Nesta situação, um MAS assegura algumas propriedades como metas comuns e projeto centralizado.

³⁰ DURFEE, E.; ROSENSCHEIN, J.S.A. (1994). Distributed Problem Solving and Multi-Agent Systems: Comparisons and Examples. INTERNATIONAL WORKSHOP ON DISTRIBUTED ARTIFICIAL INTELLIGENCE, 13., Washington, 1994, p.17-19.

2. Na segunda proposta, os Sistemas Multi-Agente fornecem a base para os sistemas DPS. Sendo assim, um MAS procura trabalhar com a individualidade e o interesse próprio de cada agente para definir como serão feitas as interações com outros agentes. O sistema DPS procura então, verificar como estas interações podem ser exploradas de forma a resolver algum problema mais global.
3. Na terceira proposta, os Sistemas Multi-Agente são vistos como sendo complementares aos sistemas DPS. Neste caso, um MAS procura responder de que forma algumas propriedades coletivas podem ser percebidas em um ambiente particular. Enquanto, sistemas DPS procuram responder como uma coleção particular de agentes pode obter algum desempenho eficiente se as propriedades do ambiente são dinâmicas e não controláveis.

4.5 CONSIDERAÇÕES FINAIS

O desenvolvimento de uma aplicação na qual se pretende utilizar a tecnologia de agentes envolve inicialmente, a identificação das características necessárias a esses agentes. Essas características serão mais facilmente identificadas se houver uma classificação dos agentes necessários de acordo com a função a ser desempenhada por estes.

Através dos conceitos de IA Distribuída — abordagem Multi-Agente ou Resolução Distribuída de Problemas —, é possível organizar esses agentes na composição do ambiente, para que os mesmos interajam, colaborem e cooperem na resolução dos problemas e permitindo que o objetivo do sistema como um todo seja alcançado.

É importante ressaltar que o termo MAS tem sido utilizado, atualmente, para definir quaisquer sistemas onde múltiplos agentes interagem. Entretanto, a divisão entre Sistemas Multi-Agente e Resolução Distribuída de Problemas apresentada procura mostrar os diferentes tipos de problemas abordados por cada abordagem, além de tentar evitar o mau uso do termo.

Para que haja a interação entre os vários agentes que comporão um ambiente é necessário definir um padrão de comunicação entre estes. As linguagens de comunicação entre agentes são discutidas na seção 5.2 deste trabalho.

5 PROJETANDO AGENTES

O projeto de sistemas implementados a partir de agentes envolve a especificação e implementação da arquitetura interna de cada um dos agentes e do modelo de comunicação entre os mesmos. A implementação da arquitetura escolhida pode ser feita em diferentes tipos de linguagens de programação. Contudo, nos últimos anos vêm sendo propostas linguagens específicas para este fim. Tais linguagens já incorporam características das principais arquiteturas e das linguagens de comunicação.

Neste capítulo são expostas algumas arquiteturas, linguagens de comunicação entre agentes e linguagens de programação de agentes.

5.1 ARQUITETURAS DE AGENTES

Uma arquitetura de agente é um mapa dos detalhes internos do agente, suas estruturas de dados, as operações que pode realizar sobre estas estruturas e o fluxo de controle entre as estruturas de dados. As arquiteturas de agente representam um movimento entre a especificação das características de um agente, fornecida pela teoria de agentes e a qual fornece uma visão abstrata de um agente, para o início da especificação da estrutura interna e funcionamento do mesmo.

“Uma arquitetura de agente é uma metodologia particular para a construção de agentes. Ela especifica como (...) o agente pode ser decomposto e construído a partir de um conjunto de módulos componentes e como estes módulos desenvolvidos podem interagir. O conjunto total de módulos e suas interações têm que fornecer uma resposta da questão de como o sensor de dados e o estado interno atual de um agente determinam as ações (...) e o futuro estado interno do agente. Uma arquitetura possui técnicas e algoritmos que suportam esta metodologia” (MAES³¹ apud WOOLDRIDGE & JENNINGS, 1995, p.18).

³¹ MAES, P. (1991). The agent network architecture (ANA). *SIGART Bulletin*, 2 (4), p. 115-120.

As arquiteturas apresentadas a seguir representam as quatro maiores abordagens para a construção de agentes, segundo WOOLDRIDGE (2000).

5.1.1 Arquiteturas baseadas em lógica

A abordagem tradicional de construção de sistemas de Inteligência Artificial, conhecida como IA simbólica, sugere que o comportamento inteligente pode ser gerado em um sistema dando a este uma representação simbólica do seu ambiente e do comportamento desejado. Ou seja, os agentes possuem uma representação explícita, em modelos simbólicos³², do mundo e sobre a qual as decisões, por exemplo sobre que ações realizar, são tomadas através de raciocínio lógico, baseado na comparação de padrões e manipulação simbólica.

Agentes de planejamento³³ são um exemplo de agentes que utilizam modelos simbólicos em suas arquiteturas. Planejamento é essencialmente programação, onde é projetado um curso de ação que, quando executado, resultam no alcance de algum objetivo desejado. Este tipo de sistema tem uma descrição simbólica do mundo e o objetivo desejado previamente fixado, além de um conjunto de descrições de ações, as quais caracterizam as pré-condições e pós-condições associadas com várias ações. Os agentes então, tentam encontrar uma seqüência de ações a qual os leva a alcançar o objetivo.

Há dois problemas a serem resolvidos na visão de se construir um agente com modelos simbólicos, segundo SHARDLOW³⁴ apud WOODRIDGE & JENNINGS (1995), que são:

- o problema de conversão/tradução: como traduzir o mundo real em uma descrição simbólica precisa e adequada, a tempo de essa descrição ser utilizável;
- o problema da representação/raciocínio: como representar simbolicamente uma informação sobre entidades e processos complexos de um mundo real, e como

³² Um sistema simbólico é definido como sendo um conjunto de entidades físicas (símbolos), fisicamente possíveis de serem construídas, que podem ser combinadas para formar estruturas, e as quais são capazes de executar processos que operam estes símbolos de acordo com o conjunto de instruções que foi simbolicamente codificado.

³³ Dentro da comunidade de pesquisa em AI simbólica tem se assumido que alguma forma de sistemas de planejamento será um componente de qualquer agente inteligente.

³⁴ SHARDLOW, N. (1990). *Action and agency in cognitive science*. Thesis (Master). Department of Psychology, University of Manchester.

obter dos agentes que raciocinem com estas informações a tempo dos resultados serem utilizáveis.

5.1.2 Arquiteturas reativas

As arquiteturas reativas de agentes são assim chamadas pois estes simplesmente reagem ao ambiente, sem executar nenhum processo de raciocínio simbólico. Tais arquiteturas são também conhecidas como comportamentais³⁵ ou situadas³⁶. Estas arquiteturas surgiram como alternativa às suposições da abordagem simbólica, alguns pesquisadores acreditavam que essas suposições não eram suficientes para construir agentes que pudessem operar em ambientes de tempo restrito.

As pesquisas para uma abordagem alternativa partiram da idéia de que o comportamento inteligente é produto da interação do agente com seu ambiente, não necessitando então, de um modelo pré-estabelecido e que o comportamento inteligente emerge a partir da interação de vários comportamentos mais simples.

5.1.3 Arquiteturas BDI – *Belief-Desire-Intention*

As arquiteturas BDI são baseadas em raciocínio prático, no qual o processo de decisão sobre o que fazer assemelha-se ao tipo de raciocínio que seres humanos costumam realizar no dia-a-dia. O raciocínio prático envolve dois importantes processos: decidir qual objetivo queremos atingir e como iremos atingir esse objetivo.

WOOLDRIDGE (2000) afirma que a arquitetura BDI é atrativa por algumas razões. Primeiro, ela é intuitiva; todos nós reconhecemos o processo de decisão sobre o que fazer e como fazer e, temos ainda, um entendimento informal das noções de crença, desejo e intenção. Ela apresenta uma decomposição funcional clara que indica quais tipos de subsistemas são necessários para a construção de um agente. A maior dificuldade como sempre, é saber como implementar estas funções eficientemente.

³⁵ Criação e combinação de comportamento individual.

³⁶ Agentes realmente situados em algum ambiente.

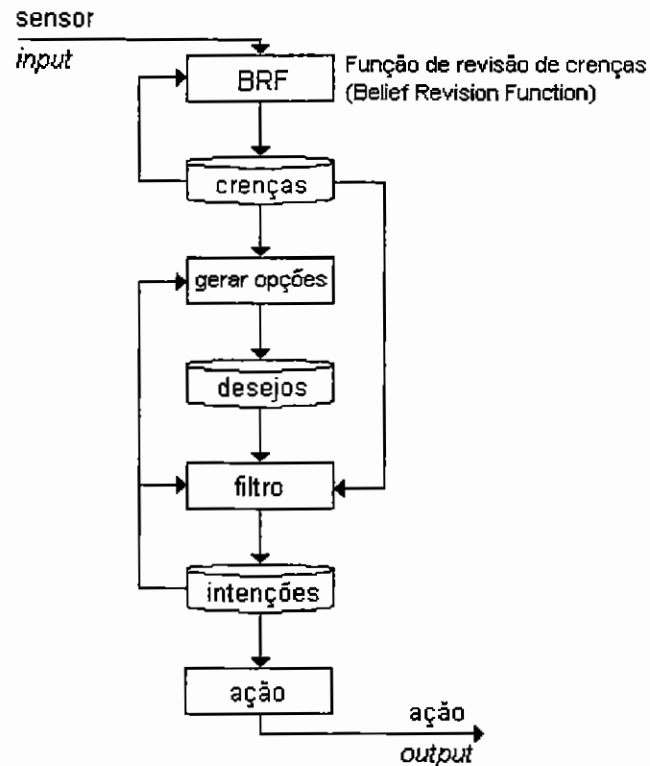


FIGURA 6 — Diagrama esquemático de uma arquitetura BDI genérica.

Fonte: WOOLDRIDGE (2000).

5.1.4 Arquiteturas em camadas

Considerando que um agente é capaz de comportamento reativo e pró-ativo, uma decomposição torna-se óbvia e envolve a criação de subsistemas separados para tratar destes diferentes tipos de comportamento (WOOLDRIDGE, 2000). Esta idéia leva naturalmente a uma classe de arquitetura na qual vários subsistemas são arranjados em uma hierarquia de camadas.

Dois tipos de fluxo de controle podem fluir entre as camadas e de acordo com estes fluxos as arquiteturas em camadas podem ser organizadas em:

- horizontalmente em camadas: Todos os níveis de *software* são conectados à entrada sensorial e ao atuador, Figura 7a. De fato, cada camada age como um agente sugerindo qual ação executar. Para garantir a consistência geralmente é incluída uma

função de mediação, responsável por definir qual camada tem o controle do agente a cada instante. Se existem n camadas em uma arquitetura e cada uma delas é capaz de sugerir m ações possíveis então existem m^n interações possíveis a considerar;

- verticalmente em camadas: A entrada e saída são conectadas a no máximo uma camada. Esta arquitetura pode ser dividida em arquitetura unidirecional (*one pass*) e bidirecional (*two pass*). Em arquiteturas de controle unidirecional, Figura 7b., o controle flui através de cada uma das camadas até a última, produzindo a saída. Em arquiteturas de controle bidirecional, Figura 7c., a informação flui para cima e o controle para baixo.

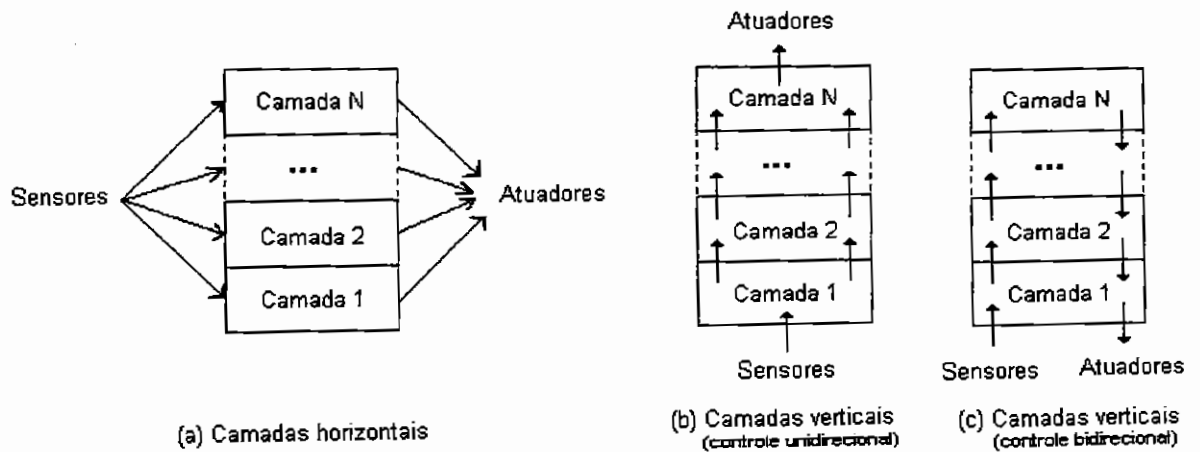


FIGURA 7 — Fluxos de informação e controle em arquiteturas em camada.

Fonte: WOOLDRIDGE (2000).

Um exemplo de arquitetura de agentes organizada horizontalmente em camadas é a **TOURINGMACHINES**³⁷, desenvolvida por Ferguson (FERGUSON, 1992a³⁸; FERGUSON,

³⁷ A nomenclatura desta arquitetura não tem relação com a conhecida Máquina de Turing, desenvolvida pelo matemático A. Turing.

³⁸ FERGUSON, I.A. (1992a). *TouringMachines: An Architecture for Dynamic, Rational, Mobile Agents*. Thesis (Doctor), Clare Hall, University of Cambridge, UK.

1992b³⁹ apud (WOODRIDGE & JENNINGS, 1995)); (WOOLDRIDGE, 2000). Tal arquitetura, Figura 8, consiste em subsistemas de percepção e ação, os quais tem interface diretamente como o ambiente do agente e uma estrutura de controle, com camadas embutidas, a qual faz o intermédio entre as camadas. Cada camada é um processo produtor de atividade independente e executado concorrentemente.

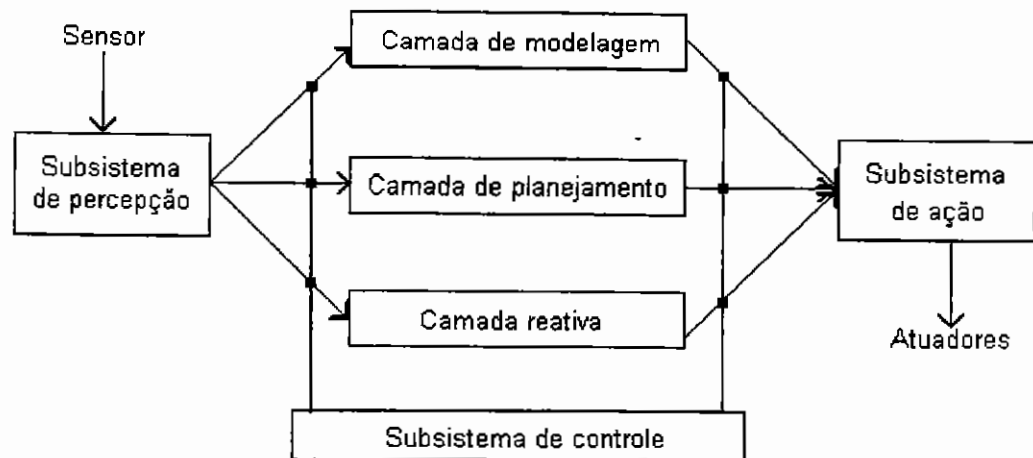


FIGURA 8 — TouringMachines: uma arquitetura de agentes híbrida.

Fonte: WOOLDRIDGE (2000).

A camada reativa gera potenciais cursos de ação em resposta aos eventos que acontecem muito rapidamente, os quais as outras camadas não conseguem tratar. Ela é implementada como um conjunto de papéis de situação-ação.

A camada de planejamento constrói planos e seleciona ações para serem executadas de maneira a alcançar os objetivos do agente.

³⁹ FERGUSON, I.A. (1992b). Towards an architecture for adaptive, rational, mobile agents. In: WERNER AND DEMAZEAU eds. *Decentralized AI 3*. Elsevier Science Publishers B.V.: Amsterdam, p. 249-262.

A camada de modelação contém representações simbólicas do estado cognitivo de outras entidades no ambiente do agente. Estes modelos são manipulados para identificar e resolver conflitos de objetivos — situações onde um agente pode não alcançar seus objetivos.

As três camadas são capazes de se comunicar, via troca de mensagens, e são embutidas em uma estrutura de controle. O propósito desta estrutura é fazer a mediação entre as camadas, e em particular, para negociar as ações conflitantes de diferentes camadas.

5.2 LINGUAGEM DE COMUNICAÇÃO ENTRE AGENTES

Uma das razões chave para as pesquisas em Sistemas Multi-Agente e Resolução Distribuída de Problemas é que um conjunto de agentes possa fornecer valor agregado aos resultados obtidos e que tais resultados não poderiam ser obtidos através do uso de um único agente (HUHNS & STPHENS, 2000). Este valor agregado é tipicamente alcançado através da cooperação. Esta cooperação é praticamente impossível de ser conseguida, na grande maioria dos Sistemas Multi-Agente, sem que os agentes se comuniquem. Então, para que haja cooperação eficiente entre os agentes é requerido destes que se comuniquem uns com os outros. E uma vez que, os agentes necessitam se comunicar, eles devem individualmente entender alguma linguagem de comunicação de agente (*agent communication language* – ACL).

As linguagens de comunicação de agentes são projetadas especificamente para facilitar as comunicações entre dois ou mais agentes, permitindo que estes comuniquem informações, intenções, objetivos e assim por diante. Uma ACL possibilita ao agente solicitar ajuda de outros agentes, para que possa alcançar seus objetivos, confiando a estes a realização de tarefas, monitorando execuções, relatando progressos, sucessos e falhas, recusando alocações de tarefas etc.

As linguagens de comunicação de agentes podem ser classificadas em duas categorias: linguagens de comunicação de agentes padrão e linguagens de comunicação de agentes *ad hoc*⁴⁰ (NWANA & WOOLDRIDGE, 1996).

⁴⁰ São desenvolvidas especificamente para uma aplicação.

Muitas ACLs, padrão ou *ad hoc*, são inspiradas na teoria do discurso que foi desenvolvida por lingüistas em uma tentativa de entender como os humanos usam a linguagem em situações do seu cotidiano.

5.2.1 KQML — Uma linguagem padrão de comunicação de agentes

A *Knowledge Query and Manipulation Language* (KQML) é parte de um amplo esforço de pesquisa para desenvolver uma metodologia para a distribuição de informação entre diferentes aplicações de agentes. O protocolo KQML permite a comunicação entre agentes e entre agentes e outras aplicações. Esse esforço de pesquisa tentar focar na habilidade dos agentes interoperar eficientemente através da comunicação de informação e conhecimento, a despeito dos problemas introduzidos pela heterogeneidade de plataformas, diferenças nas tecnologias de implementação e diferenças nos ambientes de operação e desenvolvimento. Esta comunicação requer uma linguagem comum.

A KQML segue abordagem declarativa, uma outra abordagem existente para comunicação de agentes é a abordagem procedural⁴¹. Na KQML o projeto da linguagem é baseado na idéia de que a comunicação pode ser mais bem modelada pela troca de comandos declarativos e deve ser suficientemente expressiva para comunicar uma ampla gama de informações. No cerne da linguagem KQML estão mais de trinta e seis *performatives*, apresentadas no Quadro 3, as quais define os discursos que são permitidos aos agentes usarem. Cada *performative* tem o objetivo implícito de realizar alguma ação específica. Usando *performatives* os agentes podem fazer perguntas sobre informações, comunicar fatos, divulgar seus serviços e solicitar serviços de outros agentes.

⁴¹ Na abordagem procedural a comunicação acontece através de diretivas. Tais linguagens são simples e poderosas, permitindo que programas transmitam não somente comandos, mas programas inteiros. Apesar destas linguagens serem eficientemente executadas elas trazem a desvantagem das *procedures* serem unidirecionais, e muitas vezes os agentes precisam compartilhar informações em ambas as direções.

QUADRO 3 — Performatives KQML.

<i>Categoria das performatives</i>	<i>Nomes de performatives reservados</i>
Informação básica	<i>Tell, deny, untell, cancel.</i>
Consulta básica	<i>Evaluate, reply, ask-if, ask-about, ask-one, ask-all, sorry.</i>
Consulta com múltiplas respostas	<i>Stream-about, stream-all.</i>
Efetuaadores básicos	<i>Achieve, unachieved.</i>
Geradoras	<i>Standby, ready, next, rest, discard, generator.</i>
Definição de capacidades	<i>Advertise</i>
Notificação	<i>Subscribe, monitor.</i>
De rede	<i>Register, unregister, forward, broadcast, pipe, break.</i>
Facilitação	<i>Broker-one, broker-all, recommend-one, recommend-all, recruit-one, recruit-all.</i>

Fonte: NWANA & WOOLDRIDGE (1996).

A *performative* especifica que o conteúdo de uma mensagem pode ser uma pergunta, uma afirmação ou quaisquer uma das outras categorias de *performatives*. O conjunto de *performatives* de KQML, segundo GENESERETH & KETCHPEL (1994), não é totalmente necessário, algumas aplicações não suportam todas as performativas, e nem é suficiente para todas as aplicações de agentes. Entretanto, desenvolvedores de aplicações de agentes são encorajados a usá-las para melhorar a interoperabilidade através das aplicações.

A KQML adota o uso de ontologias em suas mensagens. Ontologias são um conjunto de especificações explícitas de significado, conceitos e relacionamentos aplicáveis a algum domínio específico. A utilização de uma ontologia permite a definição de um contexto único, eliminando-se a ambigüidade.

Para entender a mensagem em si, os agentes devem ter acesso a antologia e entender a linguagem na qual a mensagem foi expressa. Uma mensagem pode conter código Prolog, Lisp, código C, expressões da linguagem natural etc.

Mensagens KQML codificam informação em três diferentes camadas: a camada do conteúdo, a camada da mensagem e camadas de comunicação, como mostrado na Figura 9.

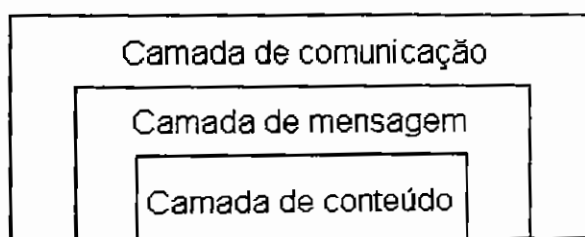


FIGURA 9 — Uma visão abstrata da linguagem KQML.

Fonte: NWANA & WOOLDRIDGE (1996).

Um exemplo de uma mensagem KQML é apresentado abaixo.

(ask-one

: sender user1

: receiver user2

: language Prolog

: ontology OOP.COURSE

: content concept (object, X)

...)

No exemplo acima, o agente *user1* solicita ao agente *user2* o conceito de objeto, onde este conceito deve estar inserido no contexto de um curso de programação orientada a objeto. O conteúdo da mensagem está especificado em Prolog.

A *performative* KQML utilizada é *ask-one*. O parâmetro *:content* define o nível de conteúdo da mensagem. No nível de conteúdo é usada a estrutura da linguagem Prolog que espera retornar em X o conceito de objeto. Os parâmetros *:sender* e *:receiver* especificam a informação no nível de comunicação. O nome da *performative*, a especificação da linguagem (*:language*) e o nome da ontologia usada (*:ontology*) fazem parte do nível de mensagem. Segundo COSTA (1999), é importante notar que a mesma mensagem poderia ser enviada com um conteúdo e linguagem diferente, ou ainda, exatamente a mesma mensagem com uma ontologia diferente.

Outros termos como *:reply-with* e *:in-reply-to* podem ser usados como parâmetros para a troca de mensagens.

A KQML deve operar em uma infraestrutura de comunicação que permite aos agentes localizarem uns aos outros. A infraestrutura não é parte da especificação da KQML, os sistemas implementados usam geralmente, programas feitos especificamente para a aplicação, chamados facilitadores, para realizar esta função.

5.2.2 Linguagens de comunicação de agentes *ad hoc*

Atualmente muitas aplicações baseadas em agentes usam um conjunto de *performatives ad hoc* dentro de *ad hoc* linguagens de comunicação de agentes. Muitas outras não tem uma linguagem de comunicação explícita; os componentes ou agentes se comunicam através do depósito de informação em alguma estrutura de dados compartilhada. Aplicações com suas próprias linguagens de comunicação são na maioria baseadas na teoria do discurso e possuem um conjunto de *performatives* similar a linguagem KQML, mas são usualmente especificadas diferentemente e usam protocolos diferentes.

A negligência de tais abordagens *ad hoc* é que estas tornam não triviais, ou até impossível, a ocorrência de uma interoperação entre aplicações de agentes construídas por diferentes desenvolvedores. O caso de se ter uma linguagem de comunicação de agentes padrão como a

KQML parece forçar os desenvolvedores a usá-la, garantindo assim, a interoperação de aplicações de agentes construídas separadamente (NWANA & WOOLDRIDGE, 1996).

5.3 A CONSTRUÇÃO DE AGENTES

No início das pesquisas em agentes os pesquisadores implementavam sistemas de agentes através de linguagens de programação de Inteligência Artificial convencionais, como Lisp e Prolog. As pesquisas com o passar dos anos, mostraram que a integração de arquiteturas de agentes bem definidas, com linguagens especializadas pode tornar o desenvolvimento e a manutenção de tais sistemas mais fáceis. A partir destas pesquisas e do crescente interesse em aplicações baseadas em agentes, um expressivo número de plataformas de desenvolvimento que fornecem suporte tecnológico para facilitar o projeto e desenvolvimento de aplicações baseadas em agentes tem se tornado disponível.

HEXMOOR et al. (1999) enfatiza que a propriedade essencial de uma plataforma de desenvolvimento de agentes é permitir integração flexível dos componentes, suportando a interação com o ambiente (incluindo outros agentes, sistemas não baseados em agentes, sensores e atuadores físicos e humanos) e o gerenciamento desta interação, incluindo raciocínio, aprendizado, coordenação etc.

A escolha da plataforma de desenvolvimento para uma aplicação de agentes é uma decisão de extrema importância, pois cada plataforma, muito provavelmente, terá alguma forma de especialização e otimização a respeito das representações e algoritmos. Algumas vezes estas especializações são por razões acadêmicas e teóricas, algumas vezes elas são por razões pragmáticas, orientadas a indústria. As especializações podem resultar em um produto superior ou facilitar o desenvolvimento para alguns desenvolvedores, mas podem debilitar um produto ou dificultar o desenvolvimento para outros.

Devido ao grande interesse por aplicações com o uso agentes para a internet, um crescente número de plataformas de desenvolvimento de agentes para tais aplicações tem se tornado disponível. Estas são um exemplo da especialização que pode haver nas plataformas de desenvolvimento de agentes, uma vez que por conta das características e estruturas dos agentes necessários a tais aplicações, dificilmente estas plataformas se adequarão a outro tipo de aplicação.

Segundo HEXMOOR et al. *ibid.*, na escolha de uma plataforma de desenvolvimento de agentes, uma análise racional do problema específico e dos requerimentos compreendidos pela aplicação deve levar em conta as seguintes considerações:

- grau de flexibilidade;
- granularidade desejada;
- agentes requeridos e suas tarefas;
- poder computacional disponível e
- tempo requerido para implementação.

5.3.1 Linguagens de programação de agentes

As linguagens de programação de agentes são linguagens que podem incorporar vários princípios propostos pela teoria de agentes. Elas definem como podemos programar agentes, quais são as primitivas corretas para estas tarefas e como podemos efetivamente compilar e executar um programa de agente.

Linguagens de Programação Orientadas a Objeto (POO) como, Smalltalk, C++ e Java, são amplamente utilizadas na construção de agentes. Além do fato de o conceito de agente não ser tão distante do conceito de objeto, o que torna mais clara a construção de um agente por estas linguagens. As linguagens POO tornam mais fácil a construção de sistemas de agentes em comparação com linguagens como Pascal, Lisp, C ou Prolog e ainda, proporcionam vantagens com relação a reusabilidade e modularidade.

Entretanto, segundo AMANDI et al. (1998), as linguagens POO apresentam uma limitação na representação de complexas atitudes mentais tais como crenças, preferências, intenções, compromissos etc., dos agentes. Esta limitação, por outro lado, é possível de ser resolvida através de vários formalismos da Programação Lógica.

Na tentativa de manter a vantagens da modularidade e reusabilidade fornecidas pelas linguagens orientadas a objeto e a inferência de conhecimento fornecido pelas linguagens

lógicas, várias alternativas tem sido propostas. Estas alternativas podem ser caracterizadas em duas linhas principais:

- I) Extensão de Programação Lógica com conceitos de Programação Orientada a Objeto. Esta alternativa proporcionaria modularidade aos programas de lógica, a qual amenizaria os problemas de performance em grandes sistemas desenvolvidos com Programação Lógica e proporcionaria a possibilidade de reuso destes programas, o que atualmente não é possível devido a complexidade destes programas.
- II) Extensão de linguagens de Programação Orientada a Objeto com conceitos de Programação Lógica. Esta alternativa permitiria a criação de uma base de conhecimento, em uma forma declarativa usando lógica, em cada objeto e o gerenciamento deste conhecimento através de um conjunto de facilidades oferecidas pelos conceitos de Programação Orientada a Objeto.

MENEZES (2001) afirma que a Programação Orientada a Agentes (POA) enunciada por Shoham⁴², define uma nova forma de construir programas que facilita a implementação de aplicações que necessitam apresentar comportamentos como autonomia, pró-atividade, aprendizado, capacidade para interação social etc.

A idéia chave da Programação Orientada a Agentes é a de programar os agentes diretamente em termos de noções mentais (tais como crenças, desejos e intenções), os quais os teóricos de agente desenvolveram para representar as propriedades de agentes. A motivação por trás desta proposta é que os humanos usam tais conceitos como um mecanismo de abstração para representar as propriedades de sistemas complexos. Da mesma forma que podemos usar estas noções mentais para descrever e explicar o comportamento dos humanos, isto também pode ser útil para programar máquinas.

As ferramentas propostas para facilitar o desenvolvimento de aplicações orientadas a agentes podem se apresentar em duas formas distintas: linguagens de Programação Orientadas a Agentes e bibliotecas de agentes que implementem características de agentes inteligentes como,

⁴² SHOHAM, Y. (1990). *Agent-oriented programming*. Computer Science Department, Stanford University, Stanford. /Technical Report STAN-CS-1335-90.

mobilidade, raciocínio, comunicação etc., em uma linguagem hospedeira, geralmente orientada a objetos como Java ou C++ (MENEZES *ibid.*).

A seguir serão apresentados dois ambientes para o desenvolvimento de aplicações de agentes, ambos foram projetados para ambientes com múltiplos agentes.

Telescript

Telescript é o nome de uma família de conceitos e técnicas que suportam o desenvolvimento dos produtos da General Magic, empresa do Vale do Silício. O Telescript fornece uma complexa linguagem de Programação Orientada a Objeto e um modelo de segurança também complexo. Considerada comercialmente o primeiro ambiente de programação para *softwares* baseados em agentes, a sua primeira versão comercial foi apresentada no final de 1994. Desde então, é amplamente citada na literatura de agentes como exemplificação de conceitos como agentes móveis, Sistemas Multi-Agente, comunicação entre agentes, agentes na Web, ferramenta para desenvolvimento de agentes, entre outros.

Há dois conceitos chaves na tecnologia Telescript: lugares e agentes. Os lugares são localizações virtuais que são ocupados por agentes. Os agentes são processos de *software* e são móveis. Eles são capazes de se mover de um lugar para outro, neste caso o programa e o estado do agente são codificados e transmitidos através de uma rede, de um lugar para outro, no qual a execução recomeça. Os agentes são capazes de se encontrar e se conectar uns com os outros para troca de informações.

Quatro componentes foram desenvolvidos para suportar a tecnologia Telescript. O primeiro é a linguagem Telescript, a qual foi projetada para suportar tarefas complexas de comunicação como, navegação, transporte, autenticação, controle de acesso e assim por diante. O segundo componente é a máquina Telescript, a qual atua como um interpretador para a linguagem Telescript, mantém lugares, horários para execução de agentes, gerencia a comunicação e transporte de agentes e fornece ainda, uma interface com outras aplicações. O terceiro componente é o protocolo Telescript, o qual primariamente trata da codificação e decodificação de agentes para possibilitar o transporte entre os lugares. E o quarto componente é um conjunto de ferramentas de *software* para suportar o desenvolvimento de aplicações Telescript.

Embora os conceitos do Telescript fossem, segundo JOHNSON (2001), brilhantes ele mal sobreviveu ao seu lançamento comercial, pois exatamente quando a internet estava penetrando na consciência popular, a General Magic havia apostado tudo num protocolo de comunicação próprio, incompatível com os padrões abertos da internet. Independente do fracasso da primeira versão do Telescript, a General Magic afirma estar desenvolvendo uma versão do *software* compatível com a internet. PARUNAK (2000) enfatiza que muitas outras firmas estão implementando os conceitos da Telescript em outras linguagens de programação como, por exemplo, Java.

HOMAGE – Heterogeneous Object Multi Agent Environment

O HOMAGE é um ambiente que permite o desenvolvimento de sistemas multi-agente composto por agentes distribuídos na internet e usa diferentes protocolos para a comunicação entre eles. O HOMAGE permite ao usuário programar em dois diferentes níveis de abstração, em alto nível através de uma linguagem orientada a agente, ALL (*Agent Level Language*), e em baixo nível através três conhecidas linguagens orientadas a objeto, que são C++, Common Lisp e Java.

A linguagem ALL oferece um modelo de agente baseado em um estado, um conjunto de ações, um motor e um conjunto de regras.

O estado mantém o conhecimento do agente sobre ele mesmo, sobre o ambiente externo e sobre os outros agentes. O conhecimento não é estático, um agente pode adquirir conhecimento através da percepção e comunicação com outros agentes.

O conjunto de ações permite ao agente agir sobre si mesmo e sobre o mundo externo. Ele tem três subconjuntos de ações: um para gerenciar seu estado e seus objetos internos; um para se comunicar com outros agentes e um para criar outros agentes.

O comportamento de um agente é dirigido por um motor que fornece mensagens de entrada através de regras divididas em conjuntos.

Uma interessante característica no HOMAGE, é que um agente pode dinamicamente criar outros agentes. Ele pode criar novas instâncias de agentes ou pode criar cópias de si mesmo, isto é, outros agentes com suas capacidades e estados correntes.

5.4 CONSIDERAÇÕES FINAIS

A construção de agentes parte da definição da arquitetura interna, a qual é um mapa dos detalhes internos de um agente. É através da arquitetura do agente que se define se um agente terá um modelo simbólico de representação de conhecimento e comportamento, se o seu conhecimento será incremental, construído a partir de conhecimentos mais simples, como será o processo de decisão deste agente e ainda, se o mesmo será composto por subsistemas separados que tratarão de diferentes tipos de comportamento.

Linguagens de programação nas abordagens de Programação Orientada a Objeto e Programação Orientada a Agente são utilizadas no desenvolvimento de aplicações de agentes. A proposta da Programação Orientada a Agentes é facilitar a construção de programas de agentes, uma vez que estes podem ser desenvolvidos a partir do comportamento que devam exibir.

Entretanto, a Programação Orientada a Objeto é, mais amplamente, utilizada no desenvolvimento de aplicações de agentes. Isso se dá pelo fato da POO ser uma abordagem mais conhecida e natural para os desenvolvedores, tornando assim a construção de agentes mais intuitiva.

As plataformas as linguagens de desenvolvimento de agentes mais recentemente propostas, já incorporam características das principais arquiteturas de agentes e das linguagens de comunicação que estabelecem os padrões de comunicação entre estes.

6 ARQUITETURA PROPOSTA

A arquitetura proposta utiliza a metodologia que vem sendo aplicada em projetos de Sistemas Tutores Inteligentes (STIs), onde os módulos componentes do ambiente são implementados através da tecnologia de agentes inteligentes.

O que se espera, ao usar agentes inteligentes no projeto e construção de ambientes de ensino-aprendizagem é alcançar objetivos relacionadas a efetividade desses ambientes no processo de ensino e à solução de problemas relacionados a modularidade e flexibilidade dos mesmos.

A arquitetura proposta, Figura 10, fundamenta-se no modelo de Sistemas Tutores Inteligentes, apresentado no capítulo 3 deste trabalho, os quais são compostos pelo módulo de domínio, módulo tutor, modelo do estudante e módulo de interface. Adotou-se uma organização diferente destes componentes dentro do ambiente, assim como na sua implementação, entretanto as funções dos mesmos são equivalentes às encontradas nos componentes de um STI.

A arquitetura é composta por dois conjuntos principais de bases de informações. As Bases de Conhecimento do Domínio que armazenam informações a respeito do domínio a ser ensinado e as Bases de Informações do Estudante. Esses conjuntos de bases de informações contêm as principais informações que são utilizadas pelos outros componentes da arquitetura.

As Bases de Conhecimento do Domínio são montadas na fase de construção do ambiente, de acordo com um domínio específico a ser ensinado e a partir do conhecimento de especialistas ou fontes de informação. À medida que se deseje que o ambiente trabalhe com outros domínios de conhecimento, novos conjuntos ou bases de informações podem ser inseridos.

As Bases de Informações do Estudante contêm dados que são fornecidos pelo estudante durante o cadastro no ambiente, e informações que surgem naturalmente durante as interações do estudante com o ambiente.

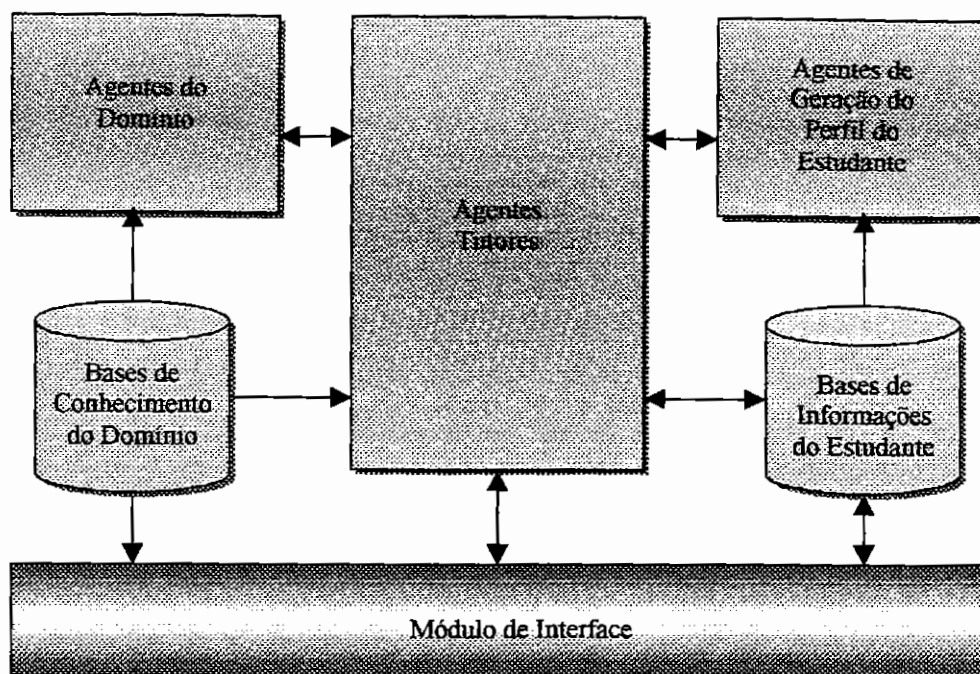


FIGURA 10 — Componentes da arquitetura.

A interação do estudante com o ambiente é realizada através do Módulo de Interface. Essa interação se dá através de um navegador internet que permite ao estudante realizar atividades, explorando páginas no formato HTML (*Hypertext Markup Language*) e recursos hipermídia⁴³.

A arquitetura possui ainda três outros módulos, os quais são compostos por agentes inteligentes. As funções desses agentes são equivalentes a funções realizadas pelos componentes de um STI. Os módulos são:

- **Módulo de Agentes do Domínio:** é composto por agentes que possuem acesso às Bases de Conhecimento do Domínio, inferem e buscam informações adequadas às situações de ensino.

⁴³ Uso simultâneo de dados em diferentes formas de mídia.

- **Módulo de Geração de Perfil do estudante:** é composto por agentes que são responsáveis por traçar perfis do estudante, a partir de inferências feitas sobre as informações contidas nas Bases de Informações do Estudante.
- **Módulo de Agentes Tutores:** é composto por agentes responsáveis pela elaboração de planos de ensino, pela definição das estratégias de ensino adequadas e o acompanhamento do desempenho do estudante.

Uma das principais características presentes nos agentes que compõem a arquitetura é a autonomia. O uso de agentes autônomos é considerado porque as atividades que os agentes exercem não necessitam de supervisão externa constante e, não existe uma autoridade central para controlar todas as interações entre os mesmos (PEREIRA et al., 1998).

Algumas decisões de projeto diferenciam a arquitetura aqui proposta, de outras arquiteturas, que também usam de agentes inteligentes, encontradas na literatura. Entre estas diferenças destacam-se as seguintes:

- I) As Bases de Conhecimento de Domínio e as Bases de Informações do Estudante são implementadas independente dos agentes.
- II) Nos componentes implementados através do uso de agentes, cada agente é responsável por uma função ou atividade específica do componente. Diferente de outras arquiteturas onde um único agente é responsável por realizar as funções de todo o módulo.
- III) Os agentes são organizados no ambiente de acordo com conceitos da abordagem de Resolução Distribuída de Problemas, seção 4.4.1, onde cada agente executa atividades isoladas e apenas comunicam seus resultados. Outras arquiteturas põem a abordagem de sistemas multi-agente.

A abordagem multi-agente, é largamente utilizada na implementação de ambientes de ensino-aprendizagem que usam a tecnologia de agentes. Entretanto, a princípio, o uso desta abordagem não se mostrou necessário na arquitetura aqui proposta, uma vez que os agentes são especialistas em suas atividades, recebem e enviam solicitações de tarefas, executam suas tarefas e enviam resultados alcançados.

Nas seções a seguir é apresentada uma descrição mais detalhada de cada um dos componentes da arquitetura.

6.1 AS BASES DE CONHECIMENTO DO DOMÍNIO

Nos Sistemas Tutores Inteligentes o ensino é apoiado em uma grande base de conhecimento a respeito do domínio a ser ensinado. Esse conhecimento está, aqui dividido em diferentes subdomínios independentes, e cada um destes contém informações pertinentes a um tipo de atividade que é realizada pelo estudante, quando em interação com o ambiente. Esta divisão considera quatro micro-especialidades⁴⁴, que são:

1. Conceitos do domínio;
2. Exemplos;
3. Exercícios;
4. Avaliações.

A divisão do conhecimento, permite que o ensino ocorra através do modelo socrático, seção 3.1.3, no qual as estratégias ou planos de ensino são compostos por várias atividades diferentes que o estudante deve realizar, como exploração de conceitos, exemplos, realização de exercícios etc.

Cada uma das bases que compõe o conhecimento do domínio é utilizada por agentes que trabalham especificamente com as informações destas bases, seção 6.4, e pelo Módulo de Interface que busca as informações a serem apresentadas ao estudante.

O conhecimento do domínio possui ainda uma base com a ementa e o cronograma do curso. Esta base de informação é usada por um agente responsável pela geração de planos de ensino — Agente de Planejamento de Ensino, seção 6.5.1.

⁴⁴ O termo micro-especialidades é usado por COSTA apud GIRAFFA (1999), na especificação da arquitetura do ambiente MATHEMA.

De acordo com o domínio a ser ensinado as informações podem ser armazenadas como arquivos em formato HTML, arquivos de imagens e multimídia em geral. De acordo com COSTA (1999), a melhor abordagem para armazenar essas informações é um banco de dados orientado a objetos, a qual permite ganhos em relação a um melhor armazenamento dos recursos multimídia e a organização das páginas como objetos.

6.2 AS BASES DE INFORMAÇÕES DO ESTUDANTE

O conhecimento do estudante é crítico para a individualização do processo instrucional e freqüentemente faz a diferença entre professores experientes e inexperientes. Professores experientes tem modelos bem definidos para classes como um todo e para cada estudante individualmente. Professores são capazes de lembrar ações realizadas pelo estudante e construir perfis dos mesmos. De acordo com GIRAFFA (1999), professores experientes harmonizam o estudante, o domínio do conhecimento e as estratégias de ensino.

Na construção de STI procura-se dotar o sistema com estas mesmas capacidades, habilitando-o a construir um modelo do estudante a partir das informações que possui deste, dos registros de suas ações e ainda, criar um perfil que represente as características de aprendizagem e personalidade do estudante. Este processo, de construção inicial de um modelo do estudante e a manutenção deste durante as execuções do sistema, é chamado de modelagem do estudante.

Para que o sistema possa criar estratégias de ensino adequadas, a modelagem do estudante tem que ser um processo dinâmico onde as informações sejam constantemente atualizadas. Estas atualizações são feitas a partir das entradas (*inputs*) do estudante, em qualquer que seja a forma: respostas dadas aos questionários, resultados de exercícios e avaliações, páginas e conteúdos explorados etc.

Considerada a importância do modelo do estudante para um ambiente de ensino-aprendizagem e a análise de diferentes modelos de estudante propostos na literatura, assim como, a verificação dos limites apresentados por estes, optou-se pela construção de uma base que armazene o maior número de informações relevantes⁴⁵ a respeito do estudante.

⁴⁵ As informações aqui consideradas relevantes são aquelas que sistema está apto a usar.

Ao trabalhar com maior número de informações do estudante pretende-se permitir a criação de um modelo do estudante que considere diferentes aspectos como comportamento, conhecimento, desempenho alcançado nas atividades etc. Com isso o processo de seleção e adoção de estratégias de ensino por parte dos Agentes Tutores pode contemplar estes diferentes aspectos do estudante.

As Bases de Informações do Estudante armazenam informações sobre diferentes aspectos do estudante. A princípio as bases de informações propostas são: Base de Histórico, Base de Comportamento e Base de Aprendizado do estudante. À medida que novos aspectos do estudante desejem ser considerados, novas bases de informações podem ser criadas.

As informações contidas nestas bases são utilizadas por agentes específicos, — Agentes de Geração de Perfil, seção 6.6 —, que geram perfis do estudante considerando suas características de conhecimento, aprendizado e comportamento.

6.2.1 Base de Histórico

Na Base de Histórico do estudante são armazenadas informações como *login* e senha, informações pessoais, nível de conhecimento sobre o domínio, — conhecimento inicial e conhecimento atual —, grau de dificuldade de aprendizagem em cada nível e notas atingidas nas avaliações.

O nível de conhecimento inicial do estudante sobre o domínio é determinado a partir de uma avaliação aplicada durante o cadastro do estudante no curso. Esta avaliação leva em consideração a área e o nível formação do estudante, e é realizada pelo Agente de Avaliação e Desempenho, seção 6.5.3.

A Base de Histórico do estudante é atualizada sempre que o mesmo realiza exercícios ou avaliações. Quando o estudante realiza exercícios, o Agente de Avaliação e Desempenho determina, considerando erros e acertos do estudante e parâmetros internos, o grau de dificuldade de aprendizagem apresentado pelo estudante no nível (etapa) do curso. Essa informação é armazenada na Base de Histórico.

Quando o estudante realiza uma avaliação para a mudança de nível, esta avaliação é corrigida pelo Agente de Avaliação e Desempenho e de acordo com parâmetros de avaliação,

uma nota é estabelecida à mesma. O nível de conhecimento do estudante é atualizado e nota atingida na avaliação é registrada na Base de Histórico do estudante.

6.2.2 Base de Comportamento

A Base de Comportamento é alimentada a partir de informações enviadas pelo Módulo de Interface, seção 6.3, à medida que o estudante interage com o ambiente. Um dos objetivos da Base de Comportamento é permitir que os Agentes de Geração de Perfil do estudante tracem um perfil de comportamento deste em relação ao ambiente.

À medida que o estudante interage com o ambiente durante as seções, as seguintes informações são armazenadas na Base de Comportamento:

- registro do número de acessos do estudante às opções de conceitos, exemplos, exercícios durante cada seção;
- tempo gasto na interpretação de conceitos, análise de exemplos e na resolução de cada exercício;
- outras informações temporais como quanto tempo durou a seção do estudante com o ambiente, o horário e o dia em que ocorreu a seção.

Um outro objetivo de se acompanhar o comportamento do estudante nas seções com o ambiente é a possibilidade de identificar que ações que o mesmo executa que resultam em um efetivo aprendizado. Isto é, toda vez que o estudante realiza corretamente um exercício o sistema identifica as ações que foram realizadas antes deste exercício e registra esta seqüência como uma seqüência que, além de indicar as preferências do estudante, também proporciona resultados favoráveis no aprendizado.

Um comportamento do estudante só é identificado de forma ideal, após algumas interações do mesmo com o ambiente.

6.2.3 Base de Aprendizado

A Base de Aprendizado contém informações a respeito dos conceitos e exemplos explorados, e de exercícios e avaliações realizadas pelo estudante. As informações que compõem a Base de Aprendizado são enviadas pelo Agente de Avaliação e Desempenho e pelo Módulo de Interface. O objetivo é representar o conhecimento que o estudante já explorou no ambiente.

São armazenados na Base de Aprendizado os nomes e endereços de arquivos, como por exemplo páginas HTML, extraídos das Bases de Conhecimento do Domínio e os quais, de alguma forma foram explorados pelo estudante.

A Base de Aprendizado é dividida em dois componentes, Base de Aprendizado do Domínio e a Base de Aprendizado da Seção. A cada interação do estudante durante uma seção com o ambiente são registradas as informações exploradas (conceitos, exemplos, exercícios e avaliações) na Base de Aprendizado da Seção. A Base de Aprendizado do Domínio é atualizada ao final de cada seção do estudante com o ambiente, a partir das informações da Base de Aprendizado da Seção.

Entre as finalidades da Base de Aprendizado esta possibilidade do estudante realizar, a qualquer momento, uma revisão do que já foi visto a respeito do domínio. E também, fornecer informações sobre o que o estudante já explorou a respeito do domínio, ao agente responsável pela geração de planos de ensino — Agente de Planejamento de Ensino, seção 6.5.1.

Os exercícios e avaliações realizados incorretamente não são armazenados na Base de Aprendizado, assim como os conceitos e exemplos relacionados a estes.

6.2.4 Novas bases de informações a respeito do estudante

Novas bases contendo informações a respeito do estudante podem ser inseridas na arquitetura a qualquer momento que se deseje considerar outras características do estudante na composição de um perfil do mesmo. Um exemplo de um outro aspecto do estudante que o sistema pode considerar, é um perfil psicológico do mesmo.

PEREIRA et al (1998) sugerem uma classificação do estudante determinando perfis psicológicos a partir de um questionário aplicado no início do curso. Essa classificação é

baseada os estudos de Carl Jung⁴⁶. De acordo com Jung, a partir de atitudes as pessoas podem ser classificadas em quatro pares opostos: extrovertido – introvertido, sensitivo – intuitivo, emotivo – racional ou perceptivo – filosófico/julgador. Pela combinação destes quatro pares é possível obter 16 tipos diferentes de personalidades.

Uma base de informações então, é criada a partir de um questionário específico aplicado ao estudante durante o processo de cadastro. Essas informações são utilizadas para a classificação do estudante dentro de um perfil psicológico, como sugerido por PEREIRA et al (1998).

Uma outra base de informações pode ainda ser criada para armazenar os erros cometidos pelo estudante durante a realização de exercícios e avaliações. Essa base pode ser utilizada com a finalidade de se identificar padrões nas dificuldades apresentadas pelo estudante.

As estratégias de ensino então podem ser adequadas considerando mais estes aspectos do estudante.

6.3 MÓDULO DE INTERFACE

O estudante interage com o ambiente através de um navegador internet. A navegação é feita através da exploração de páginas no formato HTML que permitem a utilização de recursos de hipertexto e hipermedia.

“uma das facilidades do navegador é a sua habilidade de funcionar como uma ferramenta tradicional de treinamento através da internet, permitindo ao estudante navegar livremente (de modo não linear) pelas informações” (COSTA, 1999, s/n).

De acordo com SCHÄR & KRUEGER⁴⁷ apud MAGALHÃES (2001), o hipertexto e a hipermedia permitem a organização do material didático em nós, utilizando texto ou outras mídias, conectados por ligações, que permitem o acesso não estruturado através desses nós. A aquisição do conhecimento ocorre de forma construtivista, permitindo ao estudante estabelecer a estratégia.

⁴⁶ Carl Jung (1875 — 1961) — psiquiatra suíço que ocupou lugar preponderante no movimento psicanalítico.

⁴⁷ SHÄR, S.G.; KRUEGER, H. (2000). Using new learning technologies with multimedia. *IEEE multimedia*, 7 (3): 40-51.

As páginas HTML do Módulo de Interface oferecem as opções de atividades que o estudante pode explorar na interação com o sistema. Estas opções são:

- I) **Sugestões de uso:** apresenta as sugestões iniciais do sistema a respeito de como o estudante pode proceder e que passos são indicados a serem seguidos para que o aprendizado através do sistema seja mais eficiente.
- II) **Conceitos:** de acordo com o nível de conhecimento do estudante, são disponibilizados os conceitos referentes ao domínio.

Os conceitos são apresentados por sugestão do Agente de Controle Tutor seguindo um roteiro pré-definido que apresenta as informações em uma seqüência lógica para proporcionar o aprendizado dentro de um determinado nível. Este roteiro contém *links* para informações complementares, ou para maior detalhamento dos conceitos. O estudante pode explorar os conceitos de uma maneira livre, através de tópicos apresentados a ele, onde o estudante escolhe os tópicos que pretende explorar e em que seqüência. Após algumas interações do estudante com o ambiente os Agentes Tutores são capazes de sugerir um roteiro que se adeque melhor às características de aprendizado do estudante.

- III) **Exemplos:** os exemplos são apresentados para complementar ou reforçar conceitos. Os exemplos seguem a mesma maneira de apresentação dos conceitos, de maneira pré-definida e podem ser explorados pelo estudante de acordo com tópicos.
- IV) **Exercícios:** os exercícios são apresentados com a finalidade de proporcionar a prática e treinamento a respeito do domínio. Os exercícios podem ser realizados em uma seqüência pré-definida ou podem ser escolhidos pelo estudante de acordo com tópicos.
- V) **Revisões:** sempre que o estudante o deseje, pode revisar os conceitos já vistos, assim como os exemplos e exercícios realizados. Para esta revisão o estudante seleciona o nível que pretende revisar e os conceitos, exemplos e exercícios realizados são apresentados como *links* a serem explorados.

VI) **Avaliação:** a opção de avaliação permite ao estudante realizar uma avaliação para mudança de nível de conhecimento.

Essas opções são disponibilizadas ou não, ao estudante de acordo com o plano de ensino gerado. Um exemplo é quando um plano de ensino propõe a apresentação de conceitos e exemplos ao estudante, as opções de conceitos e exemplos são então disponibilizadas, — ativas —, na tela para que o estudante possa explorá-las. As outras opções, exercícios e avaliações encontram-se desativadas, neste momento.

O Módulo de Interface tem ainda as seguintes funções:

- permitir ao estudante que se cadastre no curso;
- apresentar durante a etapa de cadastro o questionário que determinará em que nível de conhecimento o estudante se encontra inicialmente;
- identificar o estudante através de *login* e senha a cada acesso;
- apresentar ao estudante as informações solicitadas por este ou sugeridas pelos Agentes Tutores;
- receber e enviar informações aos Agentes Tutores e as Bases de Informações do Estudante;
- registrar informações de cada página acessada pelo estudante, assim como o tempo gasto na exploração do conteúdo de cada uma delas.

“O registro eletrônico do percurso de navegação pelas páginas adotado pelo estudante e o tempo despendido em cada link são informativos tanto para avaliar os pressupostos pedagógicos que nortearam a elaboração desse material, como para investigar uma série de questões teóricas sobre ensino e aprendizagem através da WWW. Dados sobre o tempo despendido em cada link, por exemplo, permitem fazer previsões sobre graus de dificuldade que o estudante encontra na execução das diferentes tarefas propostas. Já os dados sobre o percurso de navegação são indícios relevantes para refletirmos sobre a construção do conhecimento a partir do material disponibilizado na

tela. Mais especificamente, os caminhos de navegação podem indicar relações que o estudante estabelece entre os diferentes segmentos oferecidos pelo material (...). O acesso ao registro eletrônico do tempo e percurso de navegação também pode ser útil para estudos interessados em investigar possíveis relações entre as opções de caminhos privilegiadas” (MAGALHÃES, 2001, p. 8).

As funções do Módulo de Interface são divididas entre outros três módulos: Módulo de Cadastro, Módulo de Operação e Módulo de Controle de Atividades, Figura 11. Cada um dos módulos é descrito nas seções a seguir.

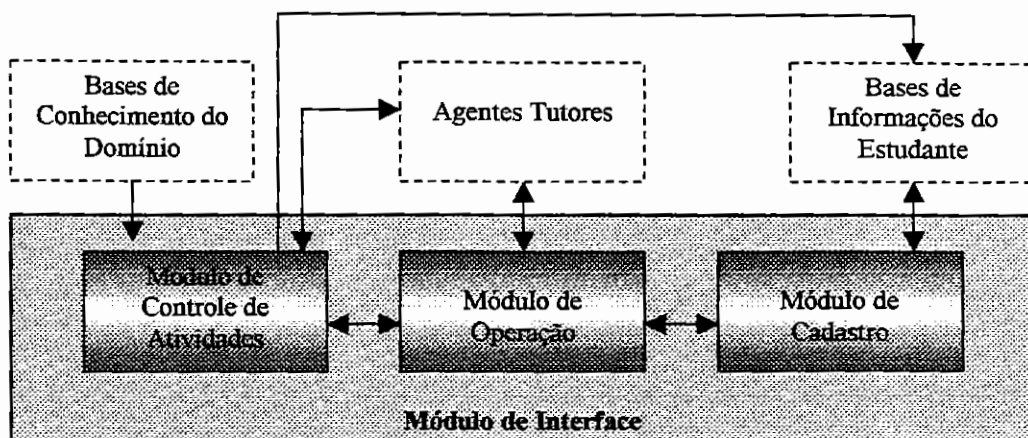


FIGURA 11 — Componentes do Módulo de Interface.

6.3.1 Módulo de Cadastro

O Módulo de Cadastro é responsável por obter as informações para o cadastro do estudante em seu primeiro acesso ao sistema. Apresenta ao estudante um questionário inicial a ser preenchido com dados pessoais, formação e o curso no qual pretende se cadastrar, e aplica a avaliação que determina o nível de conhecimento inicial que o estudante possui sobre o domínio. Estas informações iniciais alimentam a Base de Histórico do estudante.

Uma vez o estudante já cadastrado, o Módulo de Cadastro é responsável pela identificação do estudante através de um *login* e senha, nos acessos do mesmo ao ambiente. Este módulo também envia informações como a data e o horário que o estudante acessou o ambiente, à Base de Comportamento do estudante.

6.3.2 Módulo de Operação

O Módulo de Operação tem a função de, a cada ação realizada pelo estudante no ambiente, ativar o Agente de Controle Tutor, seção 6.5.2, para a geração de um novo plano de ensino para o estudante. A partir das informações recebidas, então, do Agente de Controle Tutor, disponibiliza, ou não, opções na tela.

A partir das opções selecionadas pelo estudante o Módulo de Operação também ativa o Módulo de Controle de Atividades para que os arquivos correspondentes às opções selecionadas sejam buscados nas Bases de Conhecimento de Domínio.

6.3.3 Módulo de Controle de Atividades

O Módulo de Controle de Atividades é ativado pelo Módulo de Operação quando o estudante seleciona a opção da atividade que deseja realizar. Ele busca então, nas Bases de Conhecimento de Domínio as páginas ou arquivos que são apresentados ao estudante.

Este módulo, tem ainda como função, registrar na Base de Comportamento do estudante informações sobre páginas percorridas, ações executadas e tempo gasto em cada uma. E enviar à Base de Aprendizado do estudante nomes e endereços de arquivos com os conceitos e exemplos explorados.

Quando por exemplo, um estudante seleciona a opção conceitos na tela. O Módulo de Controle de Atividades recebe do Módulo de Operação o endereço e o nome do arquivo correspondente, que é uma página HTML, e busca então este arquivo na Base de Conceitos. As informações a respeito da página acessada, os *links* percorridos e o tempo gasto são registradas na Base de Comportamento e de Aprendizado do estudante.

O Quadro 4, mostra resumidamente os componentes do Módulo de Interface, suas funções e objetivos principais, e os componentes da arquitetura com os quais se comunicam.

QUADRO 4 — Componentes do Módulo de Interface.

Módulo	Função	Objetivos	Comunicação
Módulo de Cadastro (MC)	Permite ao estudante que se cadastre e tenha acesso ao ambiente.	Identifica estudante e envia informações à Base de Comportamento do estudante.	Comunicação com as Bases de Informações do Estudante.
Módulo de Operação (MO)	Ativa Agente de Controle Tutor, o MCA e disponibiliza opções ao estudante.	A partir das opções do estudante dispara ações para que o conhecimento seja apresentado adequadamente.	Comunicação com o Agente de Controle Tutor e com MCA.
Módulo de Controle de Atividades (MCA)	Busca arquivos nas Bases de Conhecimento do Domínio e acompanha as ações do estudante.	Apresenta o conhecimento ao estudante, envia informações ao Agente de Controle Tutor, às bases de Aprendizado e de Comportamento do estudante.	Comunicação com o Agente de Controle Tutor, Base de Aprendizado e Base de Comportamento do estudante.

6.4 AGENTES DO DOMÍNIO

Os agentes deste módulo são responsáveis por decidir que conteúdo específico do curso deve ser apresentado ao estudante, em um determinado momento. Esses agentes identificam, a partir de informações recebidas do Agente de Controle Tutor, de processos de inferência e de buscas nas Bases de Conhecimento do Domínio, os conteúdos adequados à situação.

Cada um dos agentes é capaz de receber e interpretar mensagens vindas do Agente de Controle Tutor. Estas mensagens contêm informações do estudante, que guiam as inferências e buscas por informações dos Agentes do Domínio. Após identificar informações adequadas os Agentes do Domínio enviam resposta ao solicitante, — Agente de Controle Tutor, Figura 12.

As mensagens enviadas pelos Agentes de Domínio ao Agente de Controle Tutor contêm nomes de arquivos, normalmente no formato HTML, e endereços contidos nas Bases de Conhecimento do Domínio.

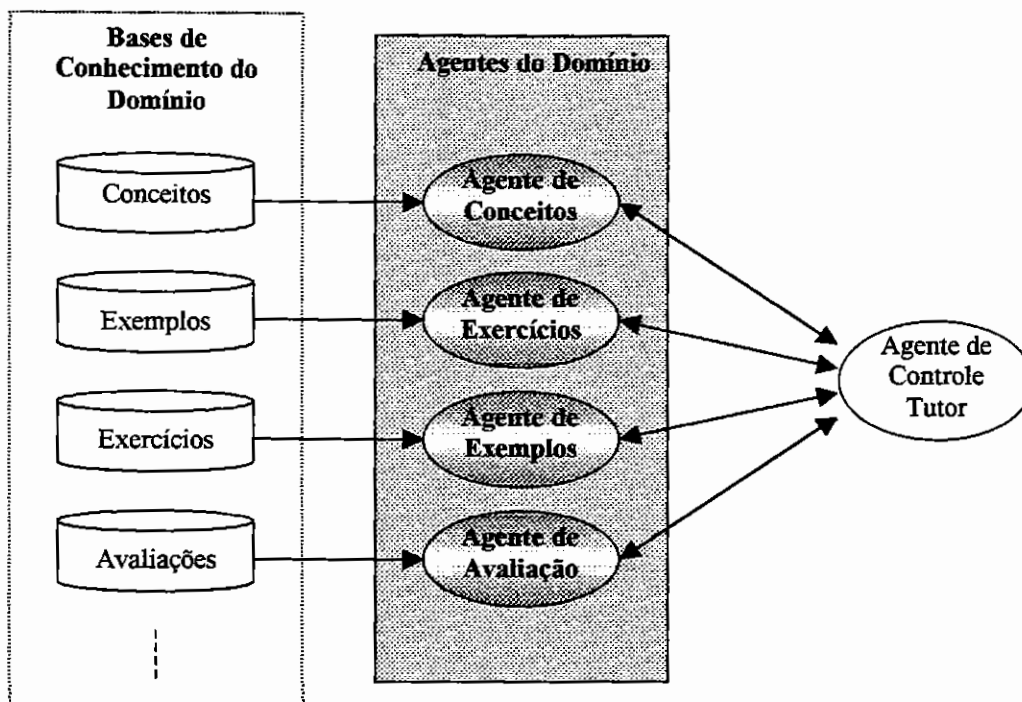


FIGURA 12 — As interações dos Agentes do Domínio.

Os Agentes do Domínio, a princípio, trabalham independentemente, cada um com uma Base de Conhecimento específica e recebem mensagens enviadas especificamente a eles, não tendo que ser comunicar ou trocar informações uns com os outros. Estes agentes são identificados como:

- Agente de Conceitos;
- Agente de Exemplos;
- Agente de Exercícios;

- Agente de Avaliação.

Uma técnica de IA bastante adequada à funcionalidade dos Agentes do Domínio é o Raciocínio Baseado em Casos (RBC), onde os casos representam situações que podem ocorrer e nas quais determinadas ações são sugeridas. O poder dessa técnica vem de sua habilidade para recuperar casos relevantes de forma rápida e precisa. Para uma discussão detalhada desta técnica ver SOVAT & CARVALHO (1999).

O Quadro 5, demonstra as funções principais dos Agentes do Domínio, o objetivo dos mesmos, componentes da arquitetura com os quais se comunicam e as propriedades básicas de agente que possuem.

QUADRO 5 — Agentes do Domínio.

Agente	Função	Objetivos	Comunicação	Propriedades básicas
Agente de Conceitos	Inferir a partir das solicitações do Agente de Controle Tutor sobre conceitos a apresentar.	Identificar e retornar informações adequadas ao nível e perfil do estudante.	Trocamos mensagens com o Agente de Controle Tutor e com a base de informação específica com que trabalham.	Autonomia, habilidade social, raciocínio, orientação a objetivo e aprendizagem.
Agente de Exemplos	Inferir a partir das solicitações do Agente de Controle Tutor sobre exemplos a apresentar			
Agente de Exercícios	Inferir a partir das solicitações do Agente de Controle Tutor sobre exercícios a apresentar.			
Agente de Avaliação	Inferir a partir das solicitações do Agente de Controle Tutor sobre a avaliação a apresentar.			

6.5 AGENTES TUTORES

As funções do módulo tutor são divididas entre três agentes. O uso de mais de um agente neste módulo tem como objetivo diminuir a complexidade dos mesmos e obter agentes mais especialistas. Os agentes tutores são:

- Agente de Planejamento de Ensino;
- Agente de Controle Tutor;
- Agente de Avaliação e Desempenho.

6.5.1 Agente de Planejamento de Ensino

O Agente de Planejamento de Ensino é responsável por elaborar um plano de ensino a ser sugerido ao estudante. Planos de ensino são solicitados pelo Agente de Controle Tutor durante a seção do estudante com o ambiente. À medida que o conhecimento do estudante muda durante uma seção, novos planos devem ser gerados.

Quando o Agente de Planejamento de Ensino, Figura 13, recebe a solicitação de um plano de ensino, ele busca informações na Base de Ementa e Conteúdo do curso e na Base de Aprendizado do estudante, com o objetivo de identificar o que do conteúdo do curso já foi explorado pelo estudante. A partir do cruzamento das informações de o que o estudante deve aprender com informações do que ele já explorou no domínio, o Agente de Planejamento de Ensino traça um plano de ensino.

O plano de ensino contém informações a respeito dos conceitos ou exemplos que o estudante deve explorar, os exercícios que devem ser realizados e também, indica uma avaliação caso o estudante esteja apto a realizá-la. Uma vez montado, o plano de ensino é enviado ao Agente de Controle Tutor.



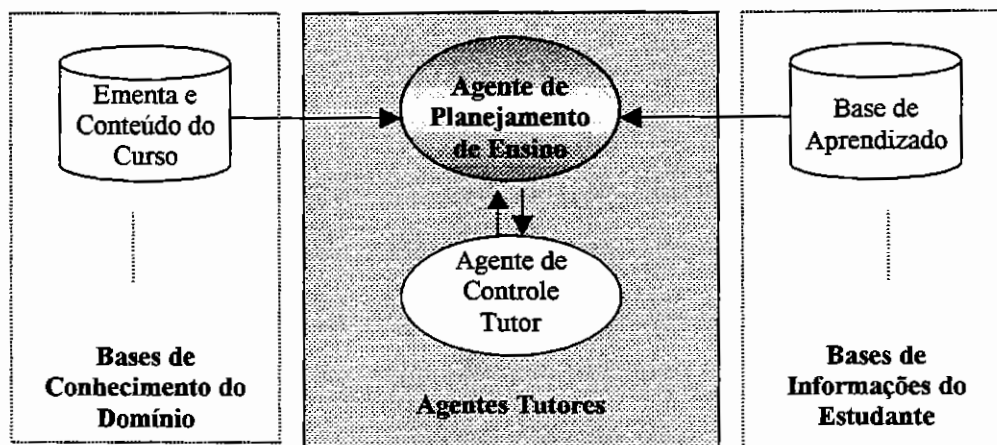


FIGURA 13 — As interações do Agente de Planejamento de Ensino.

O Agente de Planejamento de Ensino pode ser visto como um sistema de planejamento apoiado em IA e para a sua implementação devem ser aplicados conceitos e técnicas desta área. Uma técnica passível de ser usada é o Planejamento Baseado em Casos⁴⁸.

6.5.2 Agente de Controle Tutor

A cada ação realizada pelo estudante no ambiente, o Módulo de Operação, ativa o Agente de Controle Tutor para que um novo plano de ensino seja gerado. O Agente de Controle Tutor então, executa algumas ações para que possa retornar ao Módulo de Operação informações sobre quais opções de atividades devem ser disponibilizadas ao estudante, Figura 14.

O Agente de Controle Tutor inicialmente solicita um perfil do estudante ao Agente Gerenciador de Perfil, seção 6.6.2, e um plano de ensino ao Agente de Planejamento de Ensino. De acordo com o que propõe o plano de ensino, — apresentação de conceitos e exemplos, apresentação de conceitos, exemplos e exercícios, ou apenas exercícios e avaliação —, o Agente de Controle Tutor envia o perfil do estudante a específicos Agentes do Domínio e solicita a estes, indicação de conteúdos adequados. Ao receber as respostas dos Agentes do Domínio envia-as ao Módulo de Operação para que esse disponibilize as opções ao estudante.

⁴⁸ TONINDANDEL, F.; RILLO, M. (2001). Sistemas de planejamento baseado em casos. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 21., Fortaleza, 2001. *Anais*. Ceará, v.1, s/n.

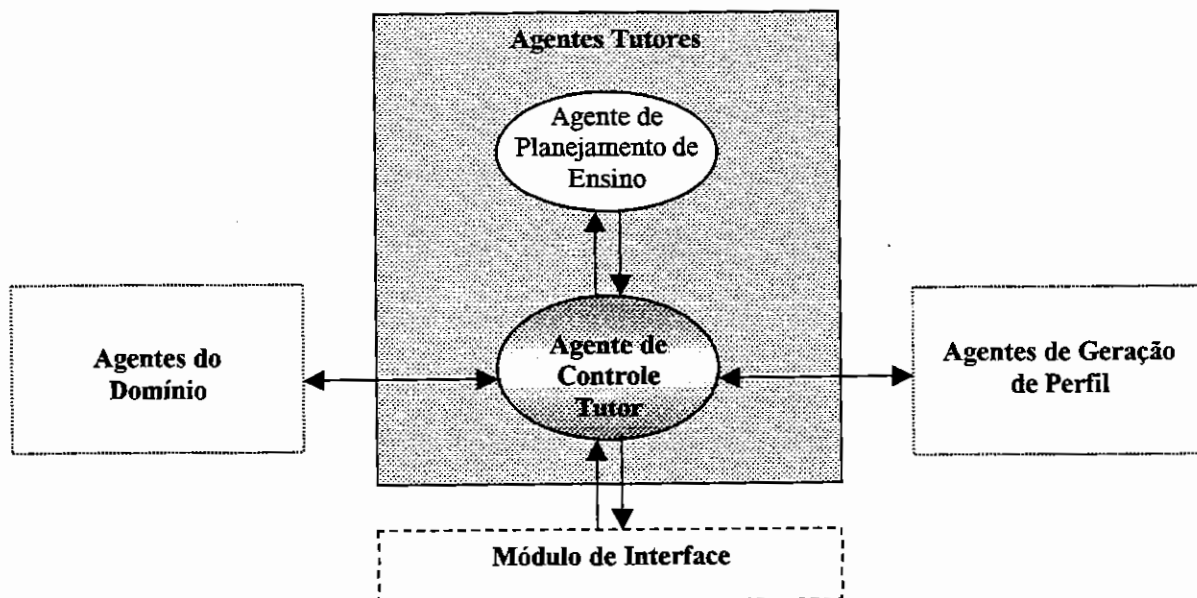


FIGURA 14 — As interações do Agente de Controle Tutor.

6.5.3 Agente de Avaliação e Desempenho

O Agente de Avaliação e Desempenho tem como funções corrigir os exercícios e avaliações realizadas pelo estudante, atualizar o nível de conhecimento do mesmo e identificar o grau de dificuldade de aprendizado apresentado pelo estudante em cada nível.

A correção de exercícios e avaliações é feita comparando as respostas fornecidas pelo estudante às respostas contidas nas Bases de Exercícios e Avaliações, Figura 15. E, de acordo com parâmetros de avaliação, o Agente de Avaliação e Desempenho é capaz de avaliar o desempenho do estudante na realização de uma avaliação e estabelecer uma nota para a mesma. Se a nota atingida pelo estudante permite a mudança do nível de conhecimento no sistema, a Base do Histórico do estudante é atualizada e a nota alcançada na avaliação é registrada.

No caso da realização de exercícios, o Agente de Avaliação e Desempenho, considerando os acertos e erros do estudante e parâmetros internos, classifica o estudante em graus de

dificuldade de aprendizagem dentro de cada nível. A informação sobre o grau de dificuldade de aprendizagem do estudante também é enviada à Base do Histórico do estudante.

Informações sobre os exercícios realizados corretamente e avaliações na qual o estudante conseguiu atingir nota para mudar de nível, assim como as respectivas respostas são enviadas à Base de Aprendizado do estudante.

Os erros cometidos pelo estudante nas realizações de exercícios e avaliações podem ser armazenados em uma nova Base de Informações do estudante, como sugerido na seção 6.2.4.

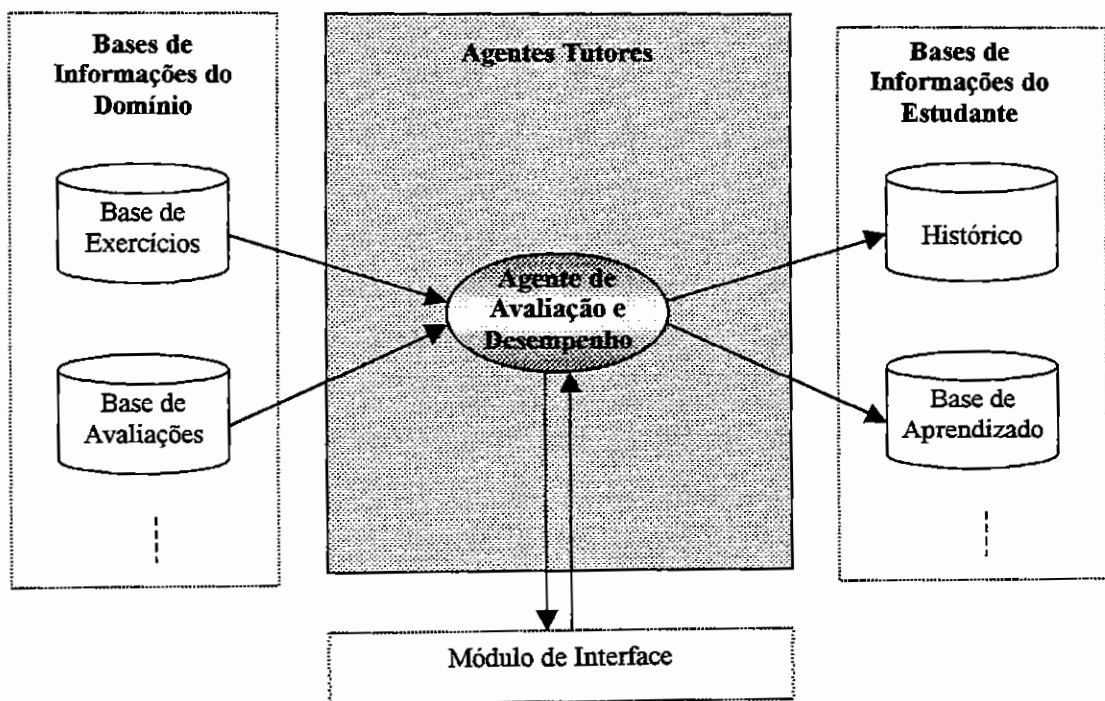


FIGURA 15 — Interações do Agente de Avaliação e Desempenho.

O Quadro 6, demonstra a função de cada um dos Agentes Tutores, os objetivos a serem alcançados por eles, os outros agentes ou componentes com os quais se comunicam e quais as propriedades básicas de agente apresentam.

QUADRO 6 — Agentes Tutores.

Agentes	Funções	Objetivos	Comunicação	Propriedades básicas
Agente de Controle Tutor (ACT)	Solicitar um perfil do estudante ao Agente Gerenciador de Perfil.	Obter um perfil do estudante para ser enviado aos Agentes de Domínio.	Troca de mensagens com os agentes Gerenciador de Perfil, Planejamento de Ensino, do Domínio e Módulo de Operação.	Autonomia, habilidade social, raciocínio e orientação a objetivos.
	Solicitar um plano de ensino ao Agente de Planejamento de Ensino.	Obter um plano de ensino adequado ao estudante.		
	Solicitar informações aos Agentes do Domínio.	Obter informações do que apresentar ao estudante.		
	Reunir as informações recebidas dos Agentes de Domínio.	Enviar informações do que disponibilizar ao estudante, para o Módulo de Operação.		
Agente de Planejamento de Ensino (APE)	Consultar Base de Aprendizado do estudante.	Comparar informações da Base de Aprendizado com informações a respeito da ementa e conteúdo do curso.	Troca de mensagens com o Agente Controle Tutor e com Base de Ementa e Conteúdo do curso e Base de Aprendizado.	Autonomia, habilidade social, raciocínio, orientação a objetivos e aprendizagem.
	Consultar Base de Ementa e Conteúdo do curso.			
	Inferir sobre que atividades o estudante deve realizar.	Traçar um plano de ensino para o estudante.		
Agente de Avaliação e Desempenho (AAD)	Classificar o estudante em graus de dificuldade de aprendizagem.	Alimentar a Base do Histórico do estudante.	Troca de mensagens com as Base de Exercícios, de Avaliação, Base do Histórico, Aprendizado e com o Módulo de Operação.	Autonomia, habilidade social, raciocínio, orientação a objetivos e aprendizagem.
	Avaliar o desempenho do estudante nas avaliações.	Atualizar o nível de conhecimento do estudante na Base do Histórico.		
	Solicitar o registro de informações sobre exercícios e avaliações realizadas pelo estudante.	Alimentar a Base de Aprendizado do estudante.		
	Consultar as Bases de Exercícios e Avaliações.	Obter respostas para correção dos exercícios e avaliações.		

6.6 OS AGENTES DE GERAÇÃO DE PERFIL DO ESTUDANTE

O propósito de um modelo do estudante é descrever o estudante de alguma maneira. Fornece meios de identificar o conhecimento do estudante, conhecer suas preferências, seu comportamento e guiar as decisões a respeito de como adaptar o processo de ensino ao mesmo.

A proposta aqui, é descrever o estudante através de um perfil que represente suas características. O perfil do estudante leva em consideração informações diversas a seu respeito, e é gerado pelo trabalho conjunto dos agentes que compõem este módulo.

A princípio, este módulo é composto por dois agentes, aqui chamados de Agentes de Classificação e um outro chamado Agente Gerenciador de Perfil.

6.6.1 Agentes de Classificação

Cada um dos Agentes de Classificação tem como função traçar um perfil do estudante, a partir de uma específica Base de Informações do estudante, Figura 16.

O Agente de Classificação 1 fornece, a partir da Base de Histórico do estudante, um perfil do desempenho do estudante, considerando informações como o nível de conhecimento que possui, grau de dificuldade de aprendizagem no nível que se encontra, notas atingidas etc.

O Agente de Classificação 2 fornece, a partir da Base de Comportamento do estudante, um perfil de comportamento em relação ao ambiente, considerando informações como, duração de cada seção, preferências quanto à execução das atividades, seqüências de atividades que levaram a um bom desempenho etc.

Outros agentes de classificação podem ser inseridos na arquitetura e novos perfis do estudante podem ser gerados, a partir da criação de novas Bases de Informações que considerem outros aspectos do estudante, como sugerido na seção 6.2.4.

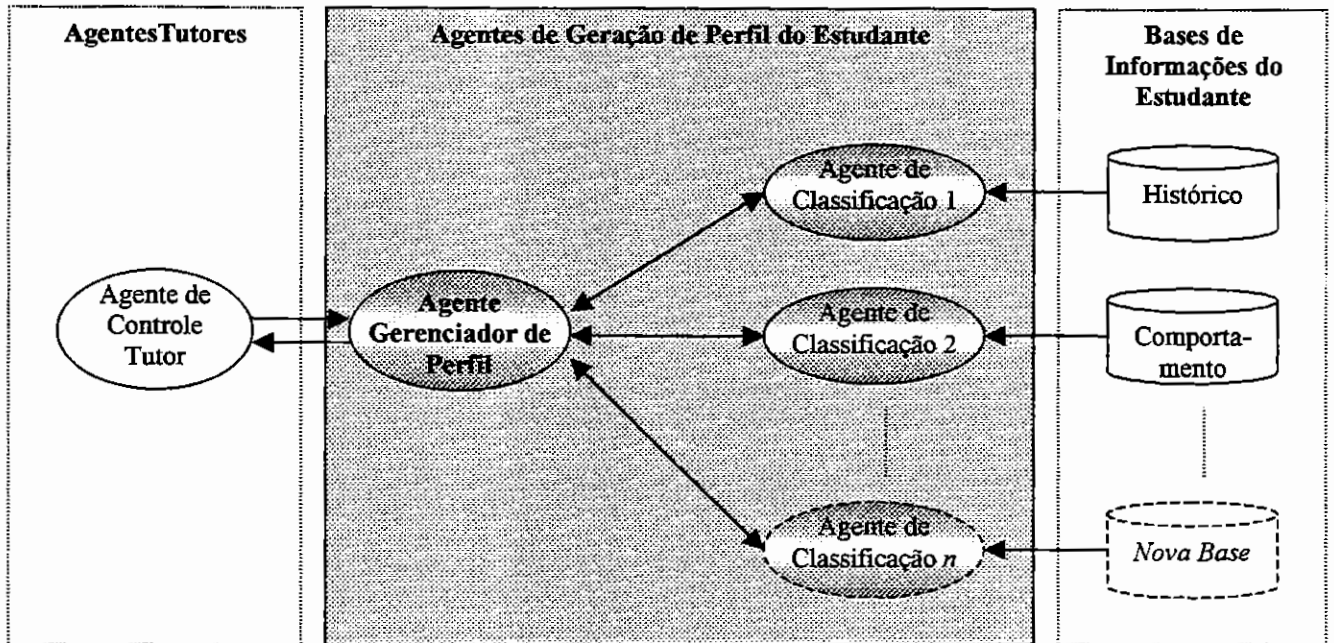


FIGURA 16 — As interações dos Agentes de Geração de Perfil do estudante.

6.6.2 Agente Gerenciador de Perfil

O Agente Gerenciador de Perfil recebe solicitação do Agente de Controle Tutor por um perfil do estudante que está em seção com o ambiente. Solicita então, aos Agentes de Classificação que estes gerem perfis do estudante. Combinando os perfis fornecidos pelos Agentes de Classificação, o Agente Gerenciador de Perfil é capaz de traçar um perfil mais completo do estudante, Figura 16.

O perfil do estudante reflete as constantes atualizações que ocorrem nas Bases de Informações do Estudante. De acordo com as ações do estudante no ambiente as informações a respeito do mesmo vão sendo armazenadas e atualizadas e novos perfis podem ser gerados a partir destas.

Cada um dos agentes de geração de perfil pode ser implementado de acordo com as técnicas que se desejar testar. Uma técnica de IA que se adequa a funcionalidade destes agentes são as Redes Neurais⁴⁹.

O Quadro 7, mostra as funções e objetivos principais de cada agente do Módulo de Geração de Perfil do estudante, a comunicação com os outros elementos da arquitetura e as propriedades básicas que estes agentes possuem.

QUADRO 7 — Agentes de Geração de Perfil do estudante.

Agente	Função	Objetivos	Comunicação	Propriedades básicas
Agente Gerenciador de Perfil (AGP)	Traça um perfil do estudante a partir das informações dos Agentes de Classificação.	Fornecer um perfil do estudante que contemple diferentes aspectos do mesmo ao Agente de Controle Tutor.	Troca de mensagens com os Agentes de classificação e com o Agente de Controle Tutor.	Autonomia, orientação a objetivos, raciocínio, habilidade social e aprendizagem.
Agente de classificação 1 (AC 1)	Classifica o estudante a partir da Base do Histórico.	Fornecer um perfil do conhecimento e desempenho do estudante.	Trocam informações com o Agente Gerenciador de Perfil e com as respectivas Bases de informações do estudante com as quais trabalham.	
Agente de classificação 2 (AC 2)	Classifica o estudante a partir da Base de Comportamento.	Fornecer um perfil do comportamento do estudante.		

⁴⁹ RIBEIRO, J.; VASCONCELOS, G. (1999). Redes neurais construtivas para classificação de padrões. In: CONGRESSO NACIONAL DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 19., Rio de Janeiro, 1999. *Anais*. Rio de Janeiro, v.4, p.53-64.

6.7 A COMUNICAÇÃO ENTRE OS AGENTES

A arquitetura é composta ainda por uma estrutura de comunicação que lida com a troca de mensagens entre os agentes. Essa troca de mensagens ocorre através de um protocolo padrão para a comunicação entre agentes. O protocolo estabelece que a comunicação não ocorre diretamente entre os agentes, mas através de um módulo de comunicação responsável pelo gerenciamento da troca de mensagens.

A comunicação entre os componentes da arquitetura consiste em enviar e receber mensagens para registro, solicitar e enviar informações. Portanto, todos os agentes definidos na arquitetura, assim como os outros componentes, devem possuir a habilidade de enviar, receber e interpretar as mensagens.

O formalismo adotado para a representação das mensagens é a linguagem KQML, descrita na seção 5.2.1. Uma característica interessante da linguagem KQML é que ela pode oferecer uma forma de acesso a informação, mesmo para programas que não sejam agentes.

6.8 CONSIDERAÇÕES FINAIS

Após a análise de características relevantes em ambientes de ensino-aprendizagem e recursos computacionais utilizados com bons resultados, chegou-se a proposta de uma arquitetura para ambientes de ensino-aprendizagem que se fundamenta no modelo dos STIs, utiliza recursos de hipermídia e tecnologia de agentes inteligentes.

Os agentes estão organizados na arquitetura com base nos conceitos da Resolução Distribuída de Problemas, uma vez que cada agente é responsável pela realização de tarefas específicas e isoladas, se comunicando uns com os outros apenas para solicitação de informações e troca de resultados alcançados.

O uso de múltiplos agentes na composição dos módulos de domínio e tutor permite uma maior qualidade e rapidez da manipulação do conhecimento a ser ensinado, assim como, nas decisões pedagógicas. Uma vez que, diferente de outras arquiteturas onde um agente é responsável pela função do módulo como um todo, aqui as funções de cada agente são menos complexas, e os mesmos podem ser mais especialistas.

Uma visão geral da arquitetura é mostrada na Figura 17, onde é possível observar os agentes e outros componentes da arquitetura, e a interação entre estes.

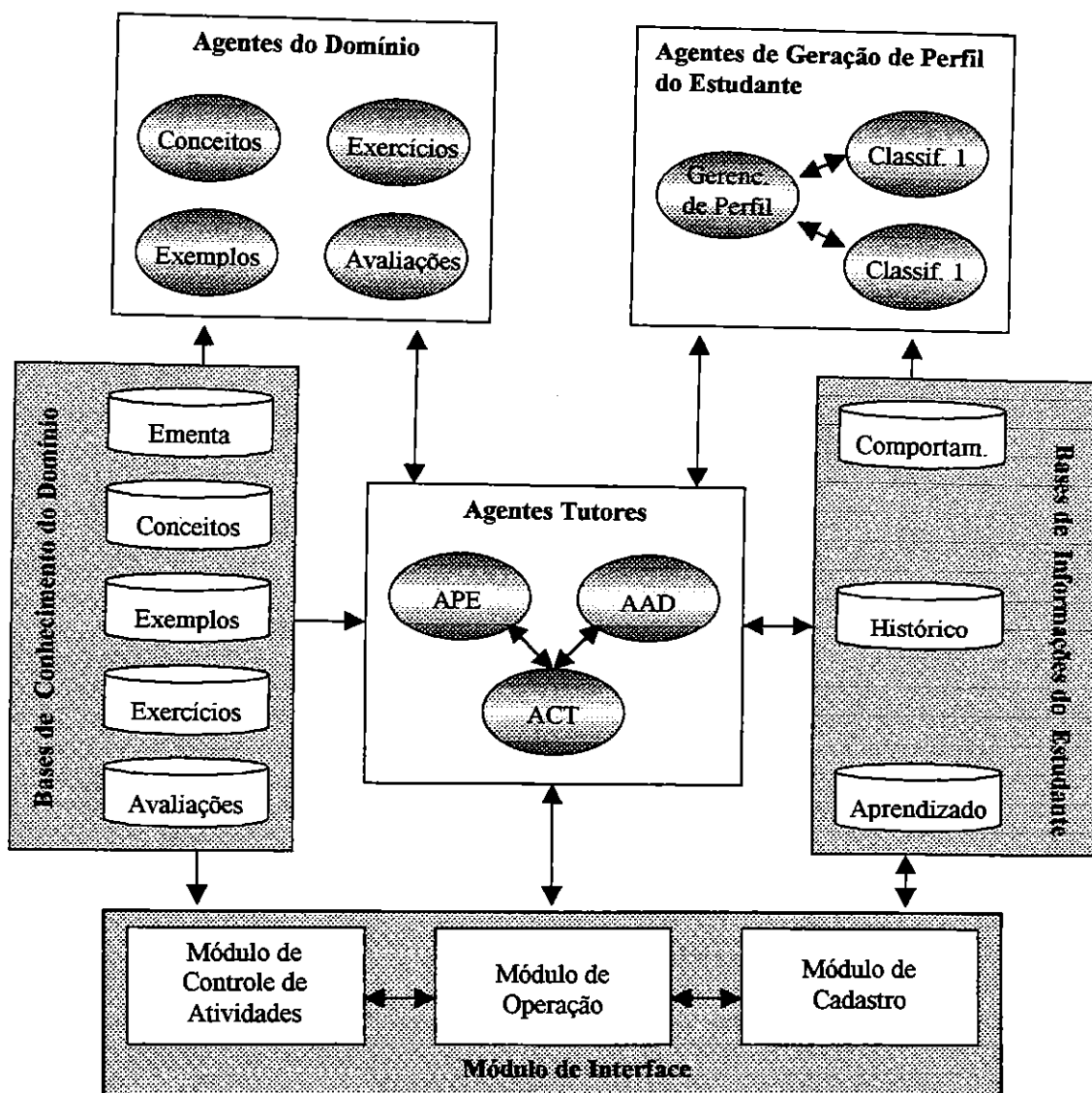


FIGURA 17 — Representação da arquitetura proposta.

A arquitetura proposta busca proporcionar flexibilidade a um ambiente de ensino-aprendizagem, uma vez que a inclusão de novas Bases de Conhecimento de Domínio, novas Bases de Informações do Estudante, agentes com outras funcionalidades e modificações em

quaisquer componentes do ambiente, podem ocorrer sem a necessidade de reformular o ambiente como um todo. No caso de inclusão de novos componentes na arquitetura, os atuais componentes e agentes terão apenas que ser habilitados a trocar informações com os novos componentes inseridos.

O número de agentes presentes na arquitetura e os diferentes tipos de funcionalidades que os agentes tem, indicam a necessidade da aplicação de mais de uma técnica de inteligência artificial. Algumas técnicas de IA foram sugeridas para o desenvolvimento de alguns agentes da arquitetura por se mostrarem adequadas às funcionalidades de tais agentes. Entretanto, há a necessidade de se testar as mesmas.

A arquitetura proposta não considera o ensino em um domínio específico, portanto a indicação de uma forma adequada para a representação do conhecimento domínio é falha, uma vez que a representação do conhecimento depende das características desse conhecimento.

7 CONCLUSÃO

O interesse no desenvolvimento de ambientes de ensino-aprendizagem tem crescido exponencialmente, devido às novas possibilidades que a internet oferece ao ensino à distância. Pesquisas realizadas em diferentes áreas procuram estabelecer premissas para o desenvolvimento de ambientes computadorizados que realmente sejam efetivos no processo de ensino.

As pesquisas realizadas pela área computacional tentam resolver as deficiências, ainda apresentadas, pelos ambientes de ensino-aprendizagem através do uso de novas tecnologias computacionais.

Este trabalho procura contribuir com uma proposta de arquitetura para o desenvolvimento de ambientes de ensino-aprendizagem que suportem o ensino à distância, utilizando a tecnologia de agentes inteligentes. A visão que se tem sobre os agentes inteligentes é a de usuário, na tentativa de melhorar a performance dos ambientes de ensino-aprendizagem a partir de algumas características que estas entidades podem apresentar, não considerando aspectos técnicos relacionados ao alcance de tais características.

As decisões de projeto da arquitetura consideram, principalmente, o aspecto da modularidade, visando proporcionar características relacionadas à flexibilidade do ambiente.

A flexibilidade apresentada pela arquitetura está relacionada às possibilidades de:

- inclusão de novos domínios de conhecimento;
- expansão do conjunto de procedimentos de ensino;
- inclusão de novos agentes, oferecendo novas funcionalidades ao ambiente;
- aplicação dos agentes da arquitetura no desenvolvimento de outros ambientes.

A arquitetura proposta atende a requisitos mínimos de projeto, uma vez que muitas variáveis do processo de ensino não são consideradas. Entretanto a principal preocupação na

elaboração da mesma, como já mencionado, é a flexibilidade apresentada, permitindo uma evolução do ambiente que permita o tratamento de variáveis não consideradas.

Algumas limitações apresentadas pela arquitetura são:

- a falta de um modelo que permita aos Agentes Tutores explicar os erros cometidos pelo estudante. Isso poderia ser resolvido incluindo uma base de erros comuns no conjunto de Bases de Conhecimento do Domínio e um agente responsável pelas inferências nesta base;
- o conhecimento que se tem do estudante é incompleto, devido à ausência da representação de outros aspectos do estudante como estado de ânimo e motivação, entre outras. Estes aspectos a respeito do estudante são difíceis de se conhecer;
- a incapacidade do ambiente em oferecer ajuda ao estudante, quando necessário. Um agente responsável por fornecer ajuda ao estudante durante a interação com o ambiente, poderia fazer parte do módulo de interface.

O modelo do estudante, nesta, assim como em outras arquiteturas, ainda é a parte mais frágil. Mesmo com o uso de técnicas de IA é difícil, senão impossível, reproduzir aspectos relacionados ao comportamento de aprendizagem do estudante. Portanto as decisões em relação à construção do modelo do estudante são, em sua maioria, baseadas em experimentos empíricos, em observações ou questões intuitivas.

Arquiteturas pesquisadas propõem, em relação a modelagem do estudante, o uso de um agente que represente o conhecimento e o comportamento do mesmo. A arquitetura aqui proposta difere destas, uma vez que, propõe o uso de bases que armazenam as informações a respeito do estudante, e que estas informações sejam usadas por vários agentes na tentativa de classificar ou identificar características de comportamento e conhecimento do estudante.

7.1 Proposta para trabalhos futuros

Os resultados obtidos com esse trabalho permitiram identificar elementos importantes para continuação desta pesquisa. Em um primeiro momento, propõe-se a definição e a inclusão de novos elementos que permitam diminuir as limitações apresentadas pela arquitetura.

Em seguida, propõe-se a implementação da arquitetura para a comprovação das hipóteses apresentadas pela mesma. Para a implementação é necessária a investigação de:

- técnicas de Inteligência Artificial adequadas a construção dos agentes, considerando as características apresentadas por cada um deles;
- abordagens apropriadas para o desenvolvimento das Bases de Conhecimento do Domínio, considerando, a princípio, um domínio de aplicação específico.
- abordagens apropriadas para o desenvolvimento das Bases de Informações do estudante;
- técnicas e ferramentas para a construção da estrutura de comunicação da arquitetura.

A criação de ambientes de ensino-aprendizado é um assunto complexo que demanda experiência e participação de diversas fontes de saber. Mostrando-se necessário o emprego de trabalho de equipes multidisciplinares, com especialistas de diversos campos como: pedagogia, psicologia cognitiva, educação à distância e computação. Dentro da computação, ainda, diversas áreas como Banco de Dados, IA e tecnologias hipermídia, devem ser englobadas. A idéia da interdisciplinaridade nestes projetos é um fato concreto, sem o qual não se podem obter resultados satisfatórios.

BIBLIOGRAFIA

- AMANDI, A.; ITURREGUI, R.; ZUNINO, A. (1998). Object-agent oriented programming. <http://citeseer.nj.nec.com/295478.html> (07 Jan.).
- AYALA, G.; YANO, Y. (1996). Intelligent agents to support the effective collaboration in a CSCL environment. In: WORLD CONFERENCE ON EDUCATIONAL COMMUNICATIONS, Boston, 1996. *Proceedings*. Massachusetts, v.1, p.19-24.
- BECK, J.; STERN, M.; HAUGSJAA, E. (2001). Applications of IA in Education. <http://www.acm.org/crossroads/xrds3-1/aied.html> (07 Set.).
- BICA, F.; SILVEIRA, R.A.; VICCARI, R.M. (1998). ELETROTUTOR III – Uma abordagem multiagentes. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 9. <http://www.lia.ufc.br/sbie98> (12 Jan.).
- BOTICARIO, J.G.; GAUDIOSO, E. (1999). A multiagent architecture for a web-based adaptive educational system. <http://citeseer.nj.nec.com/390570.html> (02 Abr.).
- BUSH, G.; PURVIS, M.; CRANEFIELD, S. (2000). Experiences in the development of an agent architecture. In: ZHANG AND SOO, eds. *Design and application of intelligent agents*. Lecture notes in artificial intelligence 1881, Springer-Verlag, p. 76-87.
- CARDIERI, M.A.C.A. (1998). Agentes inteligentes: Noções gerais. <http://www.dca.fee.unicamp.br/courses/IA368F/1s1998/Monografias/angelica.html> (12 Dez.).
- CASAS, L.A.A. (1999). *Contribuições para a modelagem de um ambiente inteligente de educação baseado em realidade virtual*. Florianópolis. Tese (Doutorado) – Departamento de Engenharia de Produção e Sistemas, Universidade Federal de Santa Catarina.
- CERVO, A.,L.; BERVIAN, P. A. (2002). *Metodologia Científica*. São Paulo, Prentice Hall.
- COSTA, M.T.C. da (1999). *Uma arquitetura baseada em agentes para suporte ao ensino à distância*. Florianópolis. Tese (Doutorado). Departamento de Engenharia de Produção e Sistemas, Universidade Federal de Santa Catarina.

- DEVEDZIC, V. ; DEBENHAM, J.; POPOVIC, D. (2000). Teaching formal languages by an intelligent tutoring system. *Education Technology & Society*, v.1.3, n.2, Apr. http://ifets.ieee.org/periodical/vol_2_2000/devedzic.html (03 Nov.).
- ERRICO, B.; MICARELLI, A. (1994). From student modeling to user modeling. In: INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS. <http://citeseer.nj.nec.com.70213.html> (02 de Abr.).
- FERREIRA, J.S.; LABIDI, S. (1998). Modelagem do aprendiz baseado no paradigma de ensino cooperativo. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 9. <http://www.lia.ufc.br/sbie98> (12 Jan.).
- FRANKLIN, S.; GRAESSER, A. (1996). Is it an agent, or just a program?: a taxonomy for autonomous agents. In: MULLER et al. *Intelligent agents III: Agents theories, architectures, languages*. Lecture notes in artificial intelligence 1193, Springer-Verlag, p.21-35.
- FREIRE, M.E.P. (1998). *O sistema tutor de um ambiente inteligente para treinamento e ensino*. São Carlos. Dissertação (Mestrado) – Instituto de Ciências Matemáticas de São Carlos, Universidade de São Paulo.
- GENESERETH M.R.; KETCHPEL S.P. (1994). Software agents. *Communications of the ACM*, v. 37, n. 7, July. <http://portal.acm.org>. (11 Set.).
- GIANGRANDI, P.; TASSO, C. (1996). Modeling the temporal evolution of student's knowledge. In: EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 1, Lisboa, 1996. *Proceedings*. Portugal. <http://www.cbl.leeds.ac.uk/~euroaied/sessions.html> (14 abr.).
- GIL, A.C. (1991). *Métodos e técnicas de pesquisa social*. São Paulo, Atlas.
- GIRAFFA, L.M.M. (2001). A classificação dos softwares educacionais: Nós necessitamos mesmo de uma taxonomia? <http://www.inf.purs.br/~giraffa> (05 Ago.).
- GIRAFFA, L.M.M; (1999). *Uma arquitetura de tutor utilizando estados mentais*. Porto Alegre. Tese (Doutorado) – Instituto de Informática, Universidade Federal do Rio Grande do Sul.
- GIRAFFA, L.M.M; VICCARI, R.M. (1998). Estratégias de ensino em sistemas tutores inteligentes modelados através da tecnologia de agentes. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 9. <http://www.lia.ufc.br/sbie98> (12 Jan.).

- GOMES, A.S.R. (2000). *Contribuições ao estudo de redes de agentes*. Campinas. Dissertação (Mestrado) – Faculdade de Engenharia Elétrica e Computação, Universidade Estadual de Campinas.
- HERMANS, B. (1997). Intelligent software agents on the internet. http://www.firstmonday.dk/issues/issues2_3/index.html (22 Ago.).
- HEXMOOR, H.; HUBER, M.; MÜLLER, J.P.; POLLOCK, J.; STEINER, D. (1999). On the evaluation of agent architectures. In: JENNINGS AND LESPÉRANCE, eds. *Intelligent agent IV: Agent theories, architectures, and languages*. Lecture Notes in Artificial Intelligence 1757, Springer-Verlag, p.106-116.
- HUHNS, M.N.; STEPHENS, L.M. (2000). Multiagent systems and societies of agents. In: WEISS, G., ed. *Multiagent Systems: a modern approach to distributed artificial intelligence*. The MIT Press, p.79-120.
- JENNINGS, N.R.; WOOLDRIDGE, M. (1998). Applying agent technology. Applied artificial intelligence. <http://citeseer.nj.nec.com/jennings98applying.html> (06 Jan.).
- JOHNSON, S. (2001). *Cultura da interface: como o computador transforma nossa maneira de criar e comunicar*. Rio de Janeiro, Jorge Zahar Ed.
- JOHNSON, W.L. (2001). Pedagogical agents in virtual learning environments. <http://www.isi.edu/isd/johnson.html> (29 Ago.).
- KARAGIANNIDIS, C.; SAMPSON, D.; CARDINALI, F. (2001). Integrating adaptive educational content into different courses and curricula. *Educational Technology & Society*, v.4, n.3. http://ifets.ieee.org/periodical/vol_4_2000/karagiannidis.html (24 Mai.).
- LÉVY, P. (1993). *As tecnologias da inteligência*. Rio de Janeiro, editora 34.
- LOPES, H.S.; NASSARI, S.M.; RAMIREZ, A.; LIMA, W.C. (1993). Arquiteturas híbridas integrando paradigmas conexionistas e simbolistas. In: SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE, 1., Rio Claro, 1993. *Anais*. São Paulo, v.3, p.10-18.
- MAGALHÃES, L.P (2001). *Projeto Sapiens*. Relatório final de atividades, processo no. 97/12807-1. <http://www.dca.fee.unicamp.br/projects/sapiens/documentos.html> (04 Mar.).
- MATHIAS, I.M.; POZO, A.T.R. (2001). Programação orientada a agentes e sua aplicação em sistemas tutores inteligentes. <http://www.portal.ufpr.br> (02 Nov.).

- MENEZES, L.; RAMALHO, G.; MOURA, H. (2001). Descrição modular de linguagens orientadas a agentes utilizando semântica de ações. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 21., Fortaleza, 2001. *Anais*. Ceará, v.1, s/n.
- MESQUITA, L.M.; SILVA, W. T. (1998). Ferramenta de autoria para o domínio de conhecimento de STI baseado em frames – FASTI. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 9. <http://www.lia.ufc.br/sbie98> (12 Jan.).
- NWANA, H.S. (1990). Intelligent Tutoring Systems: An overview. *Artificial Intelligence Review*, v. 4, p.251-277.
- NWANA, H.S. (1996). Software agents: An overview. *Knowledge Engineering Review*, v.11, n.3, p.1-40. www.labs.bt.com/projects/agents/publish/papers/review1.htm (08 Ago.).
- NWANA, H.S.; NDUMU, D.T. (1998). A brief introduction to software agent technology. In: JENNINGS AND WOOLDRIDGE eds. *Agent Technology: Foundations, Applications and Markets*. Lecture Notes in Artificial Intelligence 1855, Springer-Verlag, p.29-47.
- NWANA, H.S.; NDUMU, D.T. (1999). A perspective on software agents research. <http://citeseer.nj.nec.com/nwana99perspective.html> (08 Ago.).
- NWANA, H.S.; WOOLDRIDGE, M. (1996). Software agent technologies. <http://citeseer.nj.nec.com/29693.html> (19 Jul.).
- OLIVEIRA, F.M.; VICCARI, R.M.; COELHO, H. (1994). A topological approach to equilibration of concepts. In: BRASILIAN SYMPOSIUM ON ARTIFICIAL INTELLIGENCE, 11., Fortaleza, 1994, *Anais*. Ceará, v.1, p.52-61.
- OLIVEIRA, F.M. de; VICCARI, R.M. (1996). Are learning systems distributed or social systems? In: EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, I, Lisboa, *Proceedings*. Portugal. <http://cbl.leeds.ac.uk/~euroaied/papers/Oliveira/> (27 Ago.).
- PARUNAK, H.V.D. (2000). Industrial and Pratical Applications of DAI. In: WEISS, G. ed. *Multiagent Systems: a modern approach to distributed artificial intelligence*. The MIT press, p.377-421.
- PEREIRA, A.S.; D'AMICO, C.B.; GEYER, C.F.R. (1998). Uma aplicação de ensino orientada a agentes. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 9. <http://www.lia.ufc.br/sbie98> (12 Jan.).

- POGGI, A.; ADORNI, G. (1996). A multi language environment to develop multi agent applications. In: MULLER et al. *Intelligent Agents III: agent theories, architectures, languages*. Lecture notes in artificial intelligence 1193. Springer-Verlag, p.325-339.
- PRADO, J.P.A. (1996). *Uma arquitetura para inteligência artificial distribuída baseada em lógica paraconsistente anotada*. São Paulo. Tese (Doutorado) – Escola Politécnica da Universidade de São Paulo.
- PUGLIESI, J.B. (1998). *Sistema híbrido inteligente para suporte a um ambiente de treinamento e ensino*. São Carlos. Dissertação (Mestrado) – Instituto de Ciências Matemáticas de São Carlos, Universidade de São Paulo.
- RAGNEMALM, E.L. (1995). Student diagnosis in practice; Bridging a gap. <http://citeseer.nj.nec.com/ragnemalm95student.html> (02 Abr.).
- RICH, E.; KNIGHT, K. (1993). *Inteligência Artificial*. São Paulo, Makron Books.
- ROMANI, L.A.S.; GOMES, S.S. (2002). Professor virtual baseado em sistemas multi-agentes. www.cnpia.embrapa.br/~luciana/disciplinas/mo642/versao_final.html (24 Mai.).
- RUSSEL, S.J.; NORVIG, P. (1995). *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, NJ, Prentice Hall.
- SELF, J. (1994). *Dormobile: a vehicle for metacognition*. Lancaster. /Technical report Computing Department – Lancaster University. <http://www.lancs.ac.uk/computing/research/aai-aied/> (03 Nov.).
- SHAW, E.; JOHNSON, W.L.; GANESHAN, R. (2001). Pedagogical agents on the web. <http://www.isi.edu/isd/carte> (29 Ago.).
- SHOHAM, Y. (1997). An overview of agent-oriented program. In: BRADSHAW, J. ed. *Software Agentes*. AAAI Press, p. 216-289.
- SILVA, E.L.; MENEZES, E.M. (2001). *Metodologia de pesquisa e elaboração de dissertação*. Florianópolis: Laboratório de Ensino a Distância da UFSC. <http://www.led.ufsc.br> (10 Ago.).
- SILVA, E.Q.; MOREIRA, D.A. (2001). Agente gerenciador de cursos a distância via internet. In: WORKSHOP DE TESES E DISSERTAÇÕES, 6., ICMC/USP São Carlos. *Anais*. São Paulo, v.1, s.n.

- SOARES, J.M.; SERRA, A.B.; OLIVEIRA, A.M.B.; BARROSO, G.C. (2001). Análise das dimensões críticas da educação à distância como base para especificação de um sistema de gestão. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 21., Fortaleza, 2001. *Anais*. Ceará, v.1, p. 24.
- SOVAT, R.B.; CARVALHO, A.C.P.L.F. (1999). Um ambiente para desenvolvimento de sistemas de raciocínio baseado em casos. In: CONGRESSO NACIONAL DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 19., Rio de Janeiro, 1999. *Anais*. Rio de Janeiro, v.4, p.133-148.
- THOMPSON, J.E. (1996). *Student modeling in an intelligent tutoring system*. Thesis (Master). Graduate School of Engineering of the Air Force Institute of Technology Air University. <http://citeseer.nj.nec.com/thompson96student.html> (16 de Abr.).
- TOLOSA, G.H.; BORDIGNON, F.R.A. (1999). Revisión: tecnologia de agentes de software. *Ci. Inf.*, Brasília, v.28, n.3, p.302-309, set/dez. www.scielo.br/pdf/ci/v28n3/v28n3a8.pdf (22 Set.).
- TURINE, M.A.S.; MALTEMPI, M.V.; HASEGAWA, R. (1994). *Sistemas tutores inteligentes: uma revisão descritiva*. São Carlos. / Relatório Técnico ICM – USP.
- URBAN-LURAIN, M. (2001). Intelligent Tutoring Systems: An historic review in the context of the development of artificial intelligence and educational psychology. <http://aral.cse.msu.edu/Publications/TTS/its.htm> (28 Ago.).
- VASSILEVA, J. (1997). Goal-based pedagogical agents. In: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 8., Kobe, 1997. *Proceedings*. Japan, v.8, s.n.
- VICCARI, R.M.; OLIVEIRA, F.M. (1992). Sistemas tutores inteligentes. <http://www.ufrs.br/pos> (28 Ago.).
- VIRVOU, M.; MOUNDRIDOU, M. (2000). A web-based authoring tool for algebra-related intelligent tutoring systems. *Education Technology & Society*. v. 3, n.2. April. http://ifets.ieee.org/periodical/vol_2_200/vivou.pdf (03 Nov.).
- WOO, C. W. (1991). *Instructional planning in an intelligent tutoring system: combining global lesson plans with local discourse control*. Thesis (Doctor) – Graduated School of Computer Science, Illinois Institute of technology.
- WOOLDRIDGE, M. (2000). Intelligent Agents. In: WEISS, G. ed. *Multiagent Systems: a modern approach to distributed artificial intelligence*. The MIT press, p. 27-77.

WOOLDRIDGE, M.; JENNINGS, N.R. (1994). Agent theories, architectures, and languages: a survey. In: WOOLDRIDGE AND JENNINGS eds. *Intelligent Agents*. Springer-Verlag, p.1-22.

WOOLDRIDGE, M.; JENNINGS, N.R. (1995). Theory and practice. *The knowledge engineering review*. <http://citeseer.nj.nec.com/wooldridge95theoryandpractice.html> (22 Set.).

http://ifets.ieee.org/periodical/vol_1_98/formal_discussion1.html. The necessity and an outline of an integral approach to designing intelligent tutoring systems of a new generation. (12 Jan.).

<http://www.uow.edu.au/auc/conf98/papers/eklund.html>. The role of student knowledge in the design of computer-based learning environments. (02 Abr.).