

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

**Arquitetura de um Agente Identificador de
Fatores Motivacionais e Afetivos em um
Ambiente de Ensino e Aprendizagem.**

por

HARRY ERWIN MOISSA

Dissertação submetida à avaliação, como requisito parcial para a obtenção do grau de
Mestre em Ciência da Computação.

Prof^a. Dra. Rosa Maria Viccari
Orientadora

Porto Alegre, agosto de 2001.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Moissa, Harry Erwin

Arquitetura de um Agente Identificador de Fatores Motivacionais e Afetivos em um Ambiente de Ensino e Aprendizagem por Harry Erwin Moissa. – Porto Alegre: PPGC da UFRGS, 2001.

78p.

Dissertação (Mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2001. Orientadora: Viccari, Rosa Maria.

1. Fatores motivacionais. 2. Sistemas Tutores Inteligentes. 3. Agentes. I Viccari, Rosa Maria. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora : Profa. Wrana Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Superintendente de Pós-Graduação: Prof. Philippe Olivier Alexandre Navaux

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Carlos Alberto Heuser

Bibliotecária – Chefe do Instituto de Informática: Beatriz Haro.

Agradecimentos

Agradeço a todos que de alguma forma contribuíram para a conclusão desta dissertação.

A Deus por ter me dado força e saúde.

À Zelmi, minha esposa, pelo apoio, paciência e compreensão.

À Prof^a. Rosa, minha orientadora, pela amizade e presteza com que sempre me orientou.

À Prof^a. Magda, pela amizade, pelas dicas, sugestões e críticas.

Aos colegas do grupo de pesquisa de Inteligência Artificial, Francine, Adriana e Ricardo, pelas dicas e material de pesquisa.

À União de Tecnologia e Escolas de Santa Catarina, pelo apoio financeiro.

Sumário

Lista de Abreviaturas.....	6
Lista de Figuras.....	7
Lista de Tabelas.....	8
Abstract	10
1 Introdução	11
2 Sistemas Tutores Inteligentes	13
2.1 Estrutura Funcional	13
2.1.1 Base de conhecimento do Domínio.....	15
2.1.2 Modelo do aluno.....	15
2.1.3 Módulo Pedagógico.....	16
2.1.4 Interface	16
2.1.5 Módulo de Controle.....	16
2.2 Arquiteturas	16
2.2.1 Arquitetura Aberta.....	16
2.2.2 Arquitetura por dimensões de relação	17
2.2.3 Arquitetura Orientada a Objetos	18
2.2.4 Arquitetura baseada em sociedade de agentes	19
3 Agentes.....	22
3.1 Definição de Agente.....	22
3.2 Arquitetura Básica de um Agente.....	23
3.3 Características de um agente	24
3.4 Classificação de Agentes.....	24
3.5 Sistemas Multiagentes	25
3.5.1 Categorias de Sistemas Multiagentes	26
3.6 Sistemas de Comunicação entre Agentes.....	26
3.6.1 Arquitetura de Quadro Negro	27
3.6.2 Troca de Mensagens	28
3.7 KQML.....	29
4 Afetividade e Emoções	30
4.1 Afetividade.....	30
4.2 Emoções	30
4.2.1 Teoria de James-Lange	30
4.2.2 Teoria de Cannon e Bard	31
4.2.3 Teoria de Woodwort.....	31
4.2.4 Teoria de Schachter-Singer	31
4.3 Dispositivos Sensores de sinais afetivos humanos	31
4.3.1 Prototype Sensing System	32
4.3.2 Sensor GSR	32

4.3.3	Sensor BVP.....	33
4.3.4	Sensor de respiração.....	33
4.3.5	Sensor EMG (Electromyogram).....	34
4.4	Interface Sensora de sinais afetivos	34
4.5	Sistemas que utilizam afetividade.....	34
4.6	Modelagem dos fatores afetivos	35
4.7	A abordagem afetiva usada neste trabalho	36
5	Estratégia de Identificação	37
5.1	Fatores Motivacionais	38
5.2	Modelo de Confiança.....	38
5.3	Modelo de Esforço.....	39
5.4	Modelo de Independência	40
6	Descrição da arquitetura proposta.....	41
6.1	Funções dos agentes.....	41
6.2	Requisitos.....	42
6.2.1	Organização do Domínio	42
6.2.2	Sistema de ajuda.....	42
6.2.3	Modelo de aluno com afeto.....	43
6.2.4	Eventos Relevantes.....	44
6.3	Agente Percepção	46
6.4	Sistema de comunicação.....	47
6.5	Estudos de casos	48
6.6	A Interface	52
7	Conclusões	58
7.1	Limitações.....	58
7.2	Trabalhos Futuros.....	59
7.3	Contribuições.....	59
	Anexo 1 Fonte do Objeto Emoção	61
	Anexo 2 Fonte do Objeto Painel Gráfico (Simulador de Eventos)	69
	Bibliografia	75

Lista de Abreviaturas

STI	Sistemas Tutores Inteligentes
ITS	Intelligent Tutoring System
TI	Tutores Inteligentes
ICAI	Intelligent Computer Assisted Instruction
CAI	Computer Aided Instruction
IA	Inteligência Artificial
SMA	Sistemas Multiagentes
MAS	Multi Agent System
KQML	Knowledge Query and Manipulation Language
ANS	Autonomic Nervous System
GSR	Galvanic Skin Response
BVP	Blood Volume Pulse
EMG	Electromyogram
GIA	Grupo de Inteligência Artificial
UFRGS	Universidade Federal do Rio Grande do Sul
Hz	Hertz
CD	Compact Disc
JDBC	Java Data Base Connectivity
BDI	Belief, Desire and Intention
RMI	Remote Method Invocation
CBR	Case Base Reasoning

Lista de Figuras

FIGURA 2.1 - Arquitetura clássica de um sistema Tutor.....	14
FIGURA 2.2 - Arquitetura proposta por Viccari et al.....	15
FIGURA 2.3 - Arquitetura aberta de um ITS.....	17
FIGURA 2.4 - Arquitetura por dimensões de relações.....	18
FIGURA 2.5 - Arquitetura de ITS orientado a objetos	19
FIGURA 2.6 - Arquitetura de ITS baseado em uma sociedade de agentes autônomos.....	20
FIGURA 3.1 - Diagrama de um agente genérico.....	23
FIGURA 3.2 - Categorias de Sistemas Multiagentes	26
FIGURA 3.3 - Arquitetura de quadro-negro	27
FIGURA 3.4 - Sistema de troca de mensagens	28
FIGURA 3.5 - Estrutura de uma mensagem KQML	29
FIGURA 4.1 - Protótipo do sistema de sensoriamento fisiológico.....	33
FIGURA 4.2 - Diagrama de um módulo reconhecedor de padrão afetivo.....	35
FIGURA 6.1 - Arquitetura alterada do Eletrotutor IV	41
FIGURA 6.2 - Modelo de Aluno com Afeto	44
FIGURA 6.3 - Esquema hierárquico dos eventos e atributos relevantes.....	45
FIGURA 6.4 - Arquitetura do Agente Percepção	46
FIGURA 6.5 - Arquitetura do Ambiente	47
FIGURA 6.6 – Tela de Abertura.....	53
FIGURA 6.7 – Tela de Detalhamento das Emoções	55
FIGURA 6.8 – Tela de Simulação de Eventos	56
FIGURA 6.9 – Tela de Conteúdo	57

Lista de Tabelas

TABELA 5.1 - Modelo de Confiança.....	39
TABELA 5.2 - Modelo de Esforço	40
TABELA 5.3 - Modelo de Independência	40
TABELA 6.1 - Inicia Tutor.....	48
TABELA 6.2 - Caso 1	48
TABELA 6.3 - Caso 2	48
TABELA 6.4 - Caso 3	49
TABELA 6.5 - Caso 4	49
TABELA 6.6 - Caso 5	50
TABELA 6.7 - Caso 6	50
TABELA 6.8 - Caso 7	50
TABELA 6.9 - Caso 8	51
TABELA 6.10 - Caso 9	51
TABELA 6.11 - Caso 10.....	52

Resumo

Este trabalho está inserido no grupo de Inteligência Artificial da UFRGS e contribui com os estudos que estão sendo realizados pela Prof^a. Magda Bercht, ambos sob orientação da Profa. Rosa Maria Viccari. Situa-se na área de Inteligência Artificial, Inteligência Artificial Aplicada à Educação, Sistemas Tutores Inteligentes e Agentes. O objetivo deste trabalho é propor a arquitetura de um agente capaz de identificar fatores motivacionais e afetivos pela monitoração das ações do usuário através da interface de um STI. Esta proposta inclui a descrição de um protótipo e descrição em detalhes de diversos itens necessários à integração, entre os quais estão: a arquitetura do STI utilizado, o sistema de comunicação utilizado pela sociedade de agentes, a interface e os requisitos necessários. Inicialmente, apresentam-se algumas estruturas de STI e alguns conceitos de agentes, seguidos de conceitos sobre emoção e afetividade. Em seguida apresentam-se as estratégias de identificação de três importantes fatores motivacionais e afetivos: Independência, Confiança e Esforço. Também são apresentados os requisitos mínimos necessários para aplicação das estratégias de identificação e integração do agente em um STI.

Palavras-Chave: Fatores Motivacionais, Sistemas Tutores Inteligentes, Agentes.

TITLE: AN AGENT'S ARCHITECTURE THAT IDENTIFIES MOTIVATIONAL AND AFFECTIVE FACTORS IN A LEARNING ENVIRONMENT.

Abstract

This work is inserted in the group of Artificial Intelligence of UFRGS and it contributes with the studies that are being accomplished by Magda Bercht. These studies are both under the guidance of Rosa Maria Viccari. They are focus the areas of Artificial Intelligence, Artificial Intelligence applied to the education, Intelligent Tutoring Systems, and Agents. The objective of this work is to propose an agent's architecture able to identify motivational and affective factors for the monitoring of the user's actions through the interface of an ITS. This proposal includes the description of a prototype and details of several items necessary to the integration. Among these are: the used ITS architecture, the communication system used by the agents' society, the interface, and the necessary requirements. Initially, some structures of ITS and some agents' concepts, followed by concepts about emotion and affectivity are presented. After, we present identification of identification of three important motivational and affective factors: Independence, Confidence and Effort. The necessary minimum requirements for application of the identification strategies and the agent's integration in an ITS are also presented.

Keywords: Motivational Factors, Intelligent Tutoring Systems, Agents.

1 Introdução

Os Sistemas Tutores Inteligentes têm evoluído em uma velocidade surpreendente, e têm gerado inúmeras arquiteturas diferentes, que favorecem determinadas funcionalidades ou implementações. Outro fator importante e decorrente dessa evolução é o aumento de funcionalidade que vem sendo agregada aos STI's, que vão desde melhorias ergonômicas na interface [ASC 2000] até a necessidade e uso de novas metodologias de avaliação de desempenho do aluno [TAR 99], [MEN 98].

Na década de 60, Herbert Simon demonstrou ligações entre emoções e o modelo cognitivo [PIC 97], ou seja, a capacidade de aprendizagem do aluno é diretamente influenciada pelo estado emocional do aluno, mas somente nos últimos anos é que a emoção e a afetividade começaram a aparecer com mais frequência nas pesquisas de Inteligência Artificial, conforme cita Bercht et al em [BER 99a].

Nos estudos mais recentes, encontram-se diversos trabalhos que buscam a identificação das emoções através de sistemas computadorizados. Entre eles, pode-se citar: o trabalho de diagnose da motivação em STI proposto por Angel de Vicente e Helen Pain [VIA 98], o trabalho de implementação de táticas motivacionais de Teresa del Soldato e Benedict du Boulay em [SOL 95], o trabalho de modelagem humana e sistemas motivacionais em que Yukihiro Matsubara e Mitsuo Nagamachi em [MAT 96] propõem um projeto ergonômico, com enfoque nos estados psicológicos internos do estudante, e o projeto JANUS que busca identificar as emoções através da interpretação das expressões faciais citado em [KEA 93].

O presente trabalho concentra-se na identificação de fatores motivacionais e afetivos, com a finalidade de fornecer subsídios ao STI, de forma que o mesmo possa utilizar estes fatores detectados como um complemento da avaliação. Esta avaliação complementar é de suma importância na escolha de uma estratégia de ensino personalizada, adequada e motivadora.

O estudo dos agentes e de suas características confirma a adequação de um agente como recurso ideal para solucionar a necessidade de identificação dos fatores motivacionais e afetivos. A implementação de algumas destas características faz com que a arquitetura do agente se torne simples e funcional, possibilitando ainda a utilização de outros recursos próprios de agentes, tais como, a comunicação entre agentes e a deliberação.

A definição dos fatores afetivos e a estratégia de identificação destes fatores são apresentadas sob dois aspectos: um que está relacionado à interface e à arquitetura do STI e outro que se refere aos conceitos e à implementação das estratégias.

As definições de “emoção” apresentam um estreito relacionamento entre as emoções e as reações fisiológicas, conseqüentemente, os sensores fisiológicos são importantes ferramentas no auxílio à identificação dos fatores afetivos. Diversos sensores fisiológicos foram identificados para verificar suas características e uma possível utilização.

Atualmente existem inúmeros trabalhos que se utilizam da identificação de fatores emocionais. A finalidade destes trabalhos também é diversa e pode ser vista em [AFF 2000]. Neste trabalho, porém, não foi possível a utilização de dispositivos especiais, visto que seu objetivo é funcionar a partir de uma página em um *browser* padrão e integrar-se a um ambiente distribuído de ensino-aprendizagem inteligente, baseado em uma arquitetura multiagente que está sendo desenvolvido no GIA/UFRGS, o Eletrotutor III [SIL 2001].

O agente Percepção que aqui se propõe foi projetado para fazer parte de um ambiente que se preocupa com a integração de fatores afetivos a uma representação do modelo de aluno com fatores afetivos projetado por Bercht [BER 2000].

A implementação do protótipo proposto neste trabalho no ambiente do Eletrotutor, servirá como ferramenta para validação do trabalho de avaliação pedagógica e estratégias de ensino que está sendo desenvolvido por Bercht [BER 2000], e inclui a descrição detalhada de diversos itens necessários à integração, entre os quais destaca-se a arquitetura do STI utilizado, o sistema de comunicação utilizado pela sociedade de agentes e outros requisitos necessários.

Este trabalho está organizado da seguinte forma:

- Capítulo 2 apresenta a estrutura básica dos Sistemas Tutores Inteligentes e alguns modelos de arquiteturas;
- Capítulo 3 apresenta uma introdução ao conceito de agentes, suas características, classificação e sistema de comunicação;
- Capítulo 4 introduz as teorias da emoção, apresenta um conjunto de dispositivos utilizados no sensoriamento de sinais afetivos e algumas linhas de pesquisa ligadas à emoção;
- Capítulo 5 descreve a estratégia de identificação utilizada;
- Capítulo 6 descreve a arquitetura proposta para o agente Percepção e seus requisitos;
- Capítulo 7 apresenta as conclusões, limitações e trabalhos futuros.

2 Sistemas Tutores Inteligentes

Os Sistemas Tutores Inteligentes (STI), ou apenas TIs (Tutores Inteligentes), são também encontrados na literatura como ICAI (*Intelligent Computer Assisted Instruction*) e são resultantes da evolução do CAI (*Computer Aided Instruction*).

Sistemas Tutores Inteligentes são programas capazes de identificar as intervenções promovidas pelo aluno. Podem aprender a dialogar, a diagnosticar as deficiências do aluno e orientá-lo quando necessário [MOU 95].

Os Sistemas Tutores Inteligentes são evoluídos em relação aos CAI, visto que são dotados de técnicas de Inteligência Artificial e possuem uma concepção de projeto diferente. Um STI deve apresentar três características: primeira, o especialista embutido deve ser capaz de realizar inferências ou resolver problemas do/no domínio; segundo, ter a capacidade de deduzir o intervalo entre o aluno e o conhecimento almejado; terceiro, possuir a capacidade de implementar estratégias de ensino de forma inteligente a fim de reduzir a distância entre o conhecimento do aluno e do especialista, conforme Burns citado em [BER97].

Nas duas últimas décadas, a evolução tecnológica na área da computação proporcionou um salto na capacidade computacional dos microcomputadores, sem que houvesse um aumento substancial nos preços. Esse aumento na capacidade computacional unido ao processo de popularização destas máquinas, criou um ambiente favorável aos ambientes de ensino baseados em Sistemas Tutores Inteligentes. Porém, somente na última década, é que as tecnologias de rede e de comunicação evoluíram o suficiente para que o acesso à Internet se tornasse popular.

Paralelamente a esse último período, o Ensino a Distância também evoluiu e passou a incorporar cada vez mais os recursos computacionais. Entre os recursos incorporados destacam-se a Inteligência Artificial e os Sistemas Distribuídos.

2.1 Estrutura Funcional

Os STI não possuem uma arquitetura padrão. Estas arquiteturas podem variar de uma implementação para outra, principalmente em se tratando de domínios diferentes, mas em geral possuem os seguintes componentes funcionais básicos em comum: base de conhecimento ou do domínio, modelo pedagógico ou estratégia de ensino, interface com o usuário e modelo do aluno, representados na arquitetura clássica de um sistema tutor [MOU 96], [BER 97]. A figura 2.1 apresentada por [BER 97], representa uma arquitetura clássica.

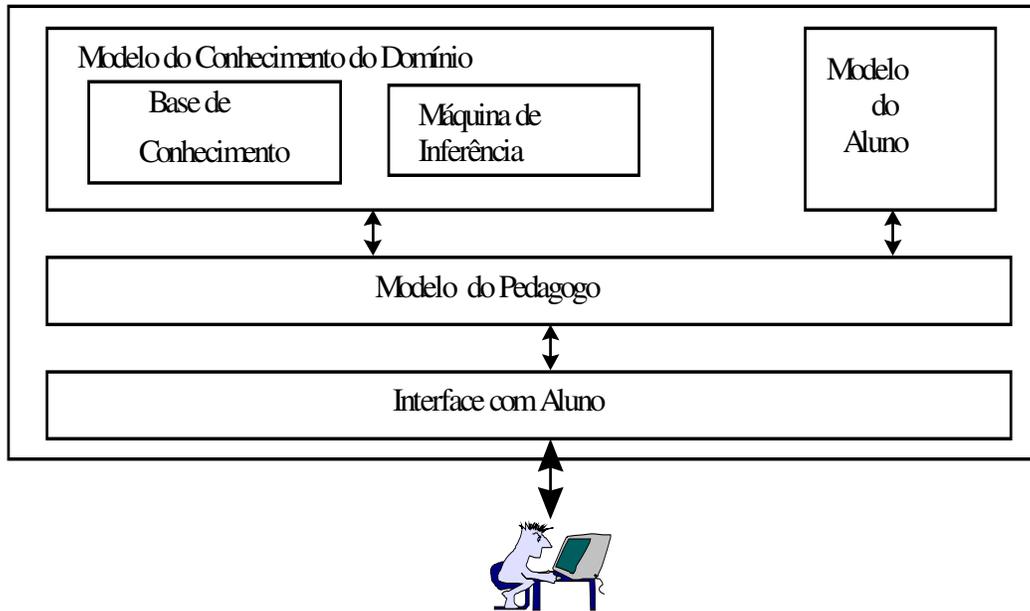


FIGURA 2.1 - Arquitetura clássica de um sistema Tutor

Em todos os domínios pode ser vista uma estrutura comum composta por módulos funcionais, alterando-se a forma de implementação, comunicação, estrutura de armazenamento, inferência, etc. Permanece a funcionalidade que pode ser agrupada a partir de cinco elementos [NUN 98], [VIC 92]. Caracteriza-se por:

- Base de conhecimento ou do domínio;
- Modelo do aluno;
- Módulo pedagógico ou estratégia de ensino;
- Módulo de comunicação ou Interface;
- Módulo de Controle.

O diagrama apresentado na figura 2.2 representa a estrutura proposta por Viccari em [VIC 92].

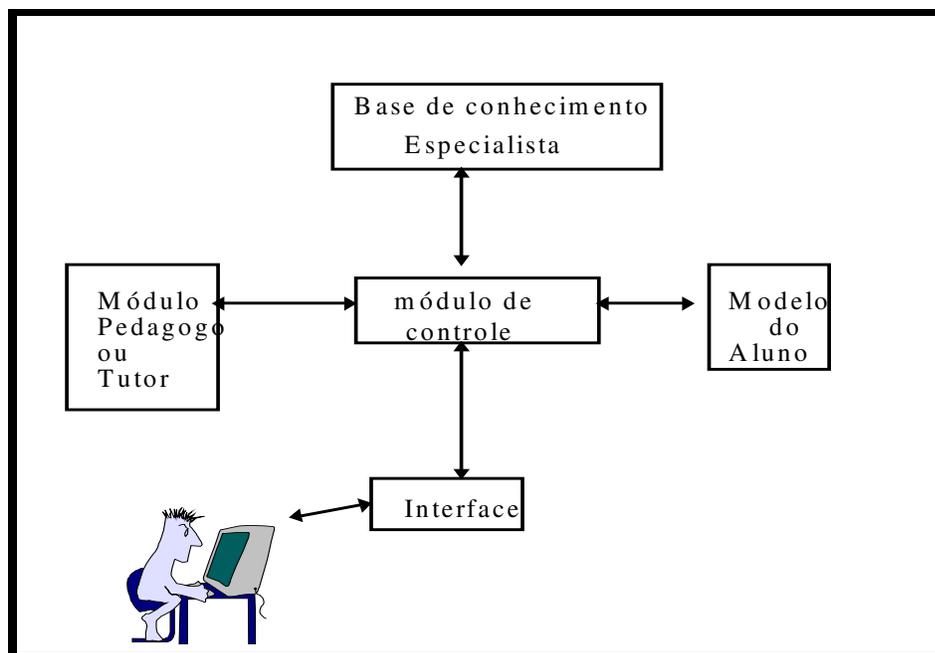


FIGURA 2.2 - Arquitetura proposta por Viccari et al.

2.1.1 Base de conhecimento do Domínio

A base de conhecimento do domínio ou Especialista mantém o conteúdo a ser ensinado, isto é, mantém todo o material instrucional adquirido do especialista, tais como: conhecimentos, fatos e regras do domínio que será trabalhado com o aluno.

2.1.2 Modelo do aluno

É neste módulo que o sistema armazena as informações relevantes referente ao aluno, ou seja, é a representação do aluno sob o ponto de vista do sistema. Este modelo contém as informações relevantes, crenças, o conhecimento que o sistema possui sobre o aluno, conhecimento este que representa do ponto de vista do sistema, o que o aluno conhece a respeito do domínio, de outros domínios afins, o que o aluno desconhece, conhece erradamente e o que ele pensa que conhece. Este módulo deve ainda ser dinâmico e refletir as alterações que ocorrem no estado cognitivo do aluno no decorrer das interações [BER 97].

É através do modelo do aluno que o tutor faz uma avaliação do aluno, e segundo Viccari citado em [BER 97], este modelo pode ainda ser acrescido dos seguintes componentes: especificação dos objetivos do aluno, modelo de intenções do aluno, planos do aluno, atitudes do aluno, procedimentos de inferência do aluno relevantes para a aplicação.

O modelo do aluno é o módulo que representa, dinamicamente, as características do aluno. Quanto mais fiel for a representação do aluno, melhor será a possibilidade do tutor definir uma estratégia de ensino adequada e personalizada ao aluno específico ora representado.

2.1.3 Módulo Pedagógico

Este módulo mantém estratégias para aplicação do material instrucional ao aluno, determina os diversos estágios no processo de ensino, representa o conhecimento de como se deve apresentar o conteúdo ao aluno, a forma, ou seja, as táticas referentes ao ensino, o modelo pedagógico.

2.1.4 Interface

A interface é responsável pela comunicação aluno-máquina e tem como funções: monitoramento de desempenho, comportamento do aluno e apresentação do material instrucional. Deve ser adaptável a diferentes usuários, agradável, rápido e objetivo, procurando sempre características que incentivem a aprendizagem do aluno.

Segundo Maenza citado em [BER 97], as interfaces dos sistemas educacionais evoluíram com a utilização de sistemas de hipertexto, tecnologia de hipermídia e de língua natural.

2.1.5 Módulo de Controle

É o módulo responsável pelo controle dos módulos já descritos e suas funções podem variar em função da estrutura escolhida para a arquitetura do tutor.

2.2 Arquiteturas

São muitas as anatomias propostas para os STI e tem-se como referência, as estruturas já apresentadas nas figuras 2.1 e 2.2, mas nenhuma delas pode ser considerada padrão.

2.2.1 Arquitetura Aberta

Entre outras propostas tem-se a de arquitetura aberta proposta por Oliveira e Viccari citada em [BER 97] e apresentada na figura 2.3, que é estruturada em módulos de conhecimento explícito: de teorias do ambiente, de ensino, de aprendizagem, do domínio, do modelo do aluno.

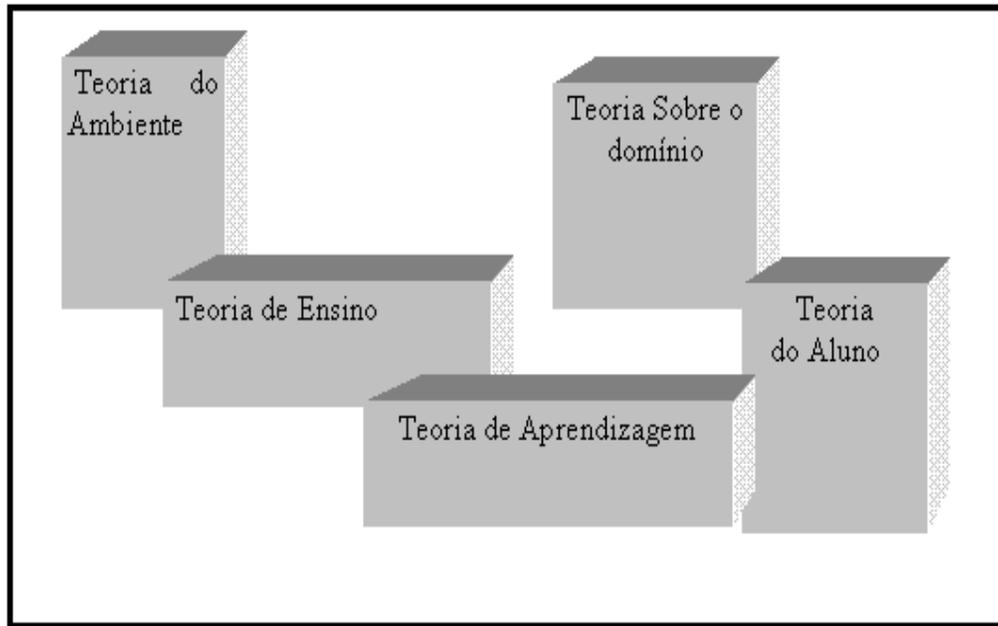


FIGURA 2.3 - Arquitetura aberta de um ITS

2.2.2 Arquitetura por dimensões de relação

Outra proposta é a apresentada por Burns e outros em [BER 97] e apresentada na fig. 2.4, que se diferencia das outras pelo destaque que dá aos relacionamentos de suas funções. Estas se caracterizam em três dimensões:

- Dimensão da comunicação homem-máquina - abrange a integração, os módulos de interface e base de domínio;
- Dimensão instrucional - abrange a comunicação, individualização e integração das funções do especialista instrucional, interface e modelo do aluno;
- Dimensão do conhecimento - envolve o relacionamento dos módulos: interface inteligente, simulação e domínio.

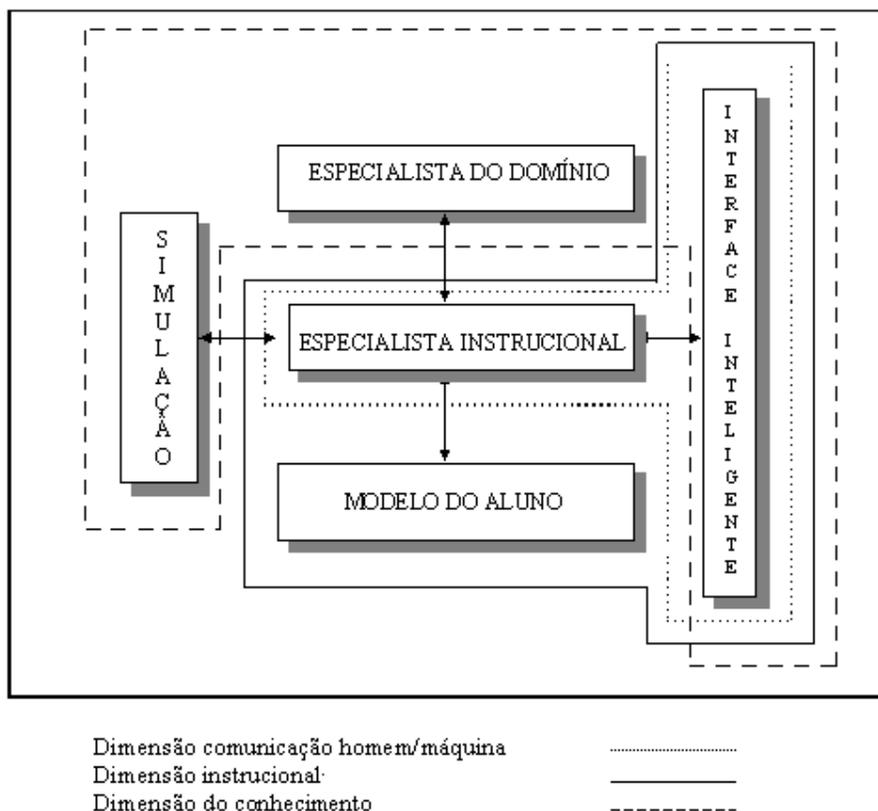


FIGURA 2.4 - Arquitetura por dimensões de relações

2.2.3 Arquitetura Orientada a Objetos

Na figura 2.5 é apresentada a proposta de Warren et al, citada em [BER 97], que está baseada na tecnologia orientada a objetos. Os componentes são entidades independentes e se comunicam entre si através de mensagens.

Nesta proposta, o Executivo é o componente chave, visto que implementa o controle sobre todo processo do sistema.

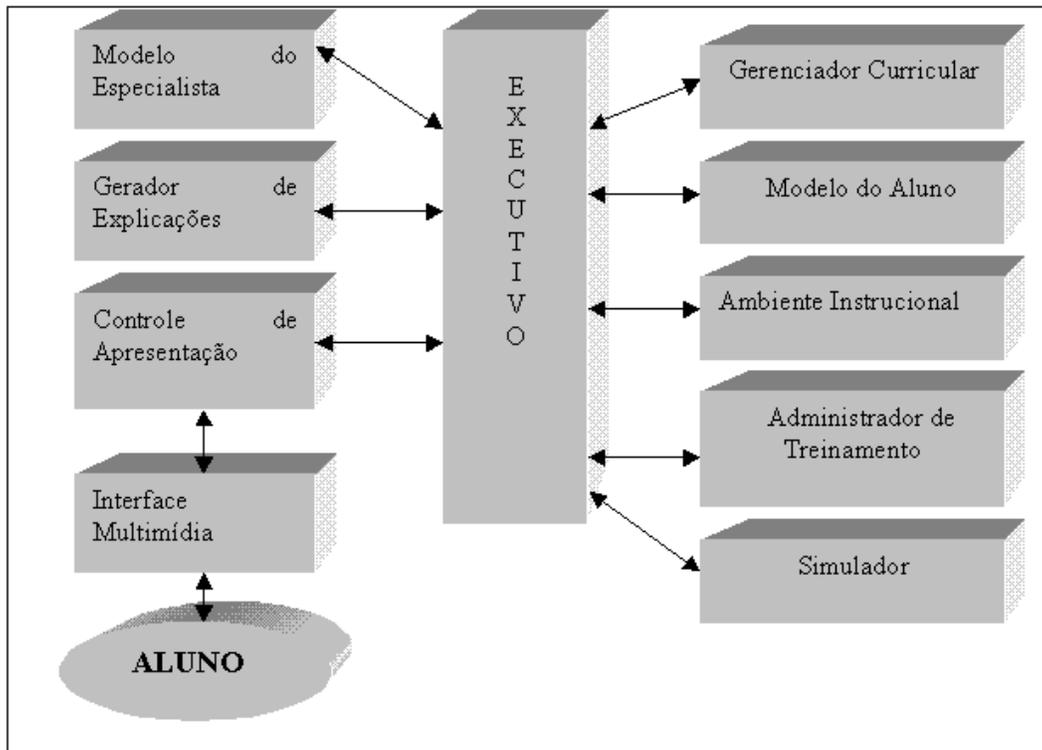


FIGURA 2.5 - Arquitetura de ITS orientado a objetos

2.2.4 Arquitetura baseada em sociedade de agentes

A arquitetura apresentada na figura 2.6, foi implementada na quarta versão do Eletrotutor [SIL 2001] e está baseada em uma sociedade de agentes autônomos que podem se comunicar uns com os outros. Nesta arquitetura, os agentes são executados de maneira contínua e se comunicam através de trocas de mensagens KQML.

O alto nível de independência de cada agente que compõe esta arquitetura, resulta em um sistema multiagente altamente flexível, no que tange à expansão da arquitetura, ou seja, se cada agente é autônomo e tem sua função bem definida, para se agregar uma nova função, são necessárias três ações: incluir um novo agente que execute esta função; preparar os agentes existentes para reconhecer e utilizar os resultados proporcionados pela nova função; incluir o resultado da nova função no sistema de comunicação do SMA.

Outra característica que merece destaque nesta arquitetura é o sistema de comunicação entre agentes. O uso do *Java Remote Method Invocation* (RMI) implementa o protocolo KQML, proporciona um sistema de comunicação extremamente flexível, com grande capacidade de distribuição e relativamente padronizado.

Esta arquitetura possui, basicamente, quatro tipos de agentes: o primeiro tipo de agente, denominado Gerenciador do Modelo do Aluno, é responsável pelo controle total de todo o sistema multiagente; o segundo tipo de agente, denominado Gerenciador de Comunicação, é responsável pela comunicação entre os agentes; o terceiro tipo de agente, denominado Agente Pedagógico, é responsável pelas diversas tarefas relacionadas à tática de ensino, sendo que existe um agente para cada uma das diferentes táticas de ensino; o quarto tipo de agente, denominado Agente de Interface, é responsável pelo controle do *browser* no ambiente do aluno.

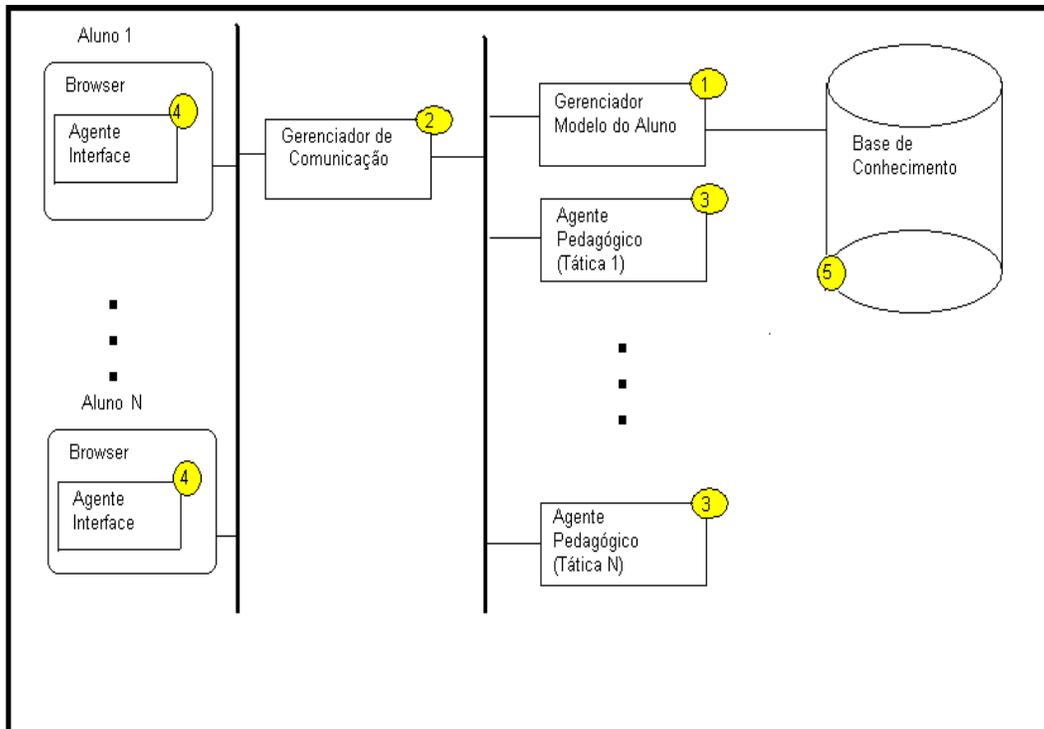


FIGURA 2.6 - Arquitetura de ITS baseado em uma sociedade de agentes autônomos

O agente Gerenciador do modelo do aluno (1) é responsável pela criação e manutenção de uma base do conhecimento (5), que mantenha a representação cognitiva de cada um dos alunos que estejam ou já estiveram conectados ao sistema. Esta manutenção compreende não só a recuperação dos dados do aluno no momento em que ele efetua o login, como também a atualização do histórico do aluno sempre que o Agente Pedagógico enviar dados para esta função.

O Agente Gerenciador de Comunicação (2) é o responsável pela gerência de toda comunicação que ocorre entre o Agente de Interface e os demais agentes do sistema. Todas estas mensagens são recebidas e enviadas através de Java RMI.

O Agente Interface do Aluno (4) é responsável pelo controle do *browser* no ambiente do aluno. Existe um Agente de Interface para cada aluno. O Agente de interface controla as atividades realizadas na interface, repassando as atividades ao

Agente Gerenciador de Comunicação, para que este encaminhe aos outros agentes; ou recebendo as mensagens do Agente Gerenciador de Comunicação, e repassando ao aluno.

O Agente Pedagógico (3) é responsável pela tática de ensino. Existe um para cada tática de ensino definida pelo especialista do curso e suas tarefas podem variar conforme a necessidade de cada agente, porém, algumas tarefas são previamente definidas a todos os agentes genéricos, entre as quais estão: mostrar o conteúdo atual, avançar para o próximo conteúdo, voltar ao conteúdo anterior, atualizar histórico e comunicação.

A flexibilidade proporcionada pela independência dos agentes, aliada à relativa padronização da comunicação utilizada, foram fatores importantes para que esta arquitetura fosse utilizada como base para o desenvolvimento deste trabalho.

A descrição das alterações necessárias para a inserção do agente proposto neste trabalho, os detalhes da implementação e a arquitetura após a implementação podem ser vistas em detalhes no capítulo 6.

3 Agentes

Encontram-se na literatura muitos conceitos para o termo agente, que apesar de muito utilizado, não possui uma definição única e não há consenso sobre o que é ou significa, embora a essência seja comum a todos.

3.1 Definição de Agente

Agentes são entidades reais ou virtuais, inteligentes e agem de acordo com suas metas e conhecimentos, conforme Frozza citado em [CAZ 96].

"Os agentes são sistemas computacionais formados por múltiplos componentes, pequenos e manejáveis, que podem agir sobre eles próprios, sobre o ambiente e cooperar entre si", segundo [MOU 95].

"Agentes autônomos são sistemas computacionais que habitam algum ambiente dinâmico complexo, percebem e atuam autonomamente neste ambiente, e perfazem um conjunto de objetivos ou tarefas para qual foi designado", segundo Maes apud [FRA 96].

"Agentes inteligentes executam três funções: percebem as condições dinâmicas no ambiente; atuam para afetar as condições no ambiente; e raciocinam para interpretar as percepções, resolver os problemas, desenhar inferências e determinar ações". [HAY 95].

Segundo Ferber apud [SIC 95], um agente é uma entidade real ou virtual, capaz de: perceber e representar parcialmente este ambiente; comunicar-se com outros agentes; possuir um comportamento autônomo que é uma consequência de suas observações, de seus conhecimentos e de suas interações com outros agentes.

Segundo Russel [RUS 95], "Um agente é algo que percebe seu meio ambiente através de sensores e age neste mesmo ambiente através de atuadores", conforme figura 3.1.

A definição apresentada por Stuard Russel, apresenta de forma clara e objetiva, o que é um agente e serve como definição ao agente proposto neste trabalho.

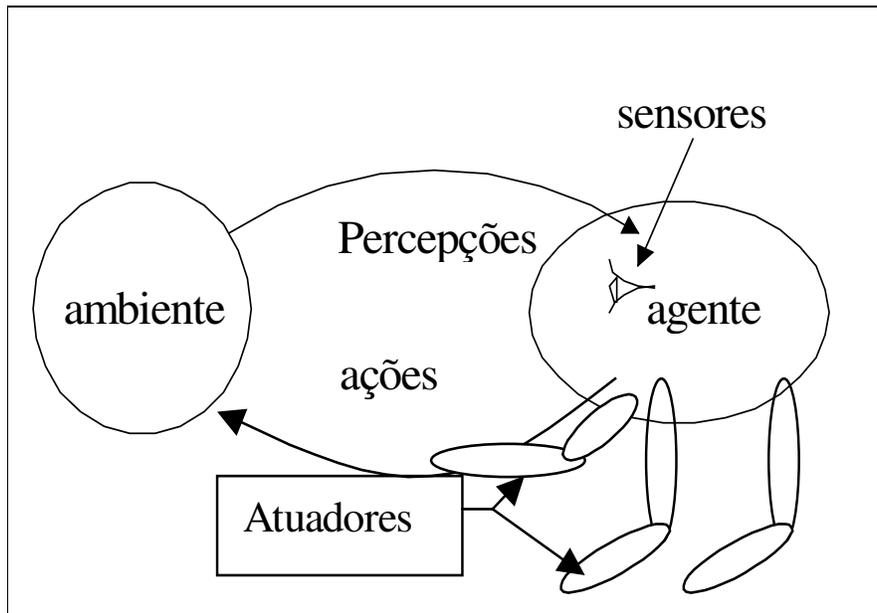


FIGURA 3.1 - Diagrama de um agente genérico

3.2 Arquitetura Básica de um Agente

A arquitetura de agentes é uma técnica de representação utilizada na construção de agentes. É nela que se especifica como o agente será decomposto, quais serão os módulos componentes e como estes módulos se comunicarão.

Segundo Corrêa, citado em [DAM 95], um agente caracteriza-se por suas possíveis interações com o mundo, pelos processos internos que possibilitam a realização destas interações e pela maneira como estas interações acontecem.

As arquiteturas propostas para assegurar um nível particular de autonomia e inteligência aos agentes diferenciam-se entre si e podem ser classificadas de acordo com os mecanismos utilizados pelos agentes para seleção de uma ação, conforme citam [DAM95], [NUN 98]:

- Arquitetura deliberativa: é aquela que apresenta algum tipo de raciocínio lógico, está baseada em modelo simbólico do mundo e tomam decisões baseadas em reconhecimento de padrões ou sobre informações simbólicas;
- Arquitetura reativa: Arquitetura cujas ações estão baseadas em regras pré-definidas, possui um mecanismo de controle simples, não se utiliza um modelo simbólico e é incapaz de responder alternativamente a uma situação não prevista em seu código;
- Arquitetura híbrida: é aquela cujas ações são determinadas através de recursos combinados, característicos das arquiteturas deliberativa e reativa.

3.3 Características de um agente

Assim como na definição de agente, não há um consenso sobre quais características um agente deve possuir. Os pesquisadores Michael Wooldridge e Nicholas Jennings apresentam diferentes listas de características para suas noções forte e fraca de agente. As características estão fortemente relacionadas à classificação dos agentes e freqüentemente são utilizadas para esse fim.

Wooldridge & Jennings [WOO 95], na sua definição fraca de agentes relacionam as seguintes características:

- **Autonomia:** Os agentes operam sem a intervenção direta de um ser humano e possuem algum tipo de controle sobre suas ações e estados internos;
- **Habilidade Social:** Capacidade de interagir com outros agentes por meio de algum tipo de linguagem de comunicação de agentes, segundo Genesereth e Ketchpel [GEN 94];
- **Reatividade:** Capacidade do agente em perceber o seu ambiente, seja ele o mundo físico, o usuário via interface gráfica, uma coleção de outros agentes, a Internet, ou qualquer combinação entre eles;
- **Pró-atividade:** Capacidade de demonstrar comportamento dirigido a objetivos, e não apenas uma atuação em resposta a percepção do ambiente.

Já em sua noção forte de agentes, são apresentadas as seguintes características:

- **Mobilidade:** Capacidade de um agente em se mover por uma rede eletrônica;
- **Veracidade:** Concepção de que o agente não vai propagar falsas informações de forma consciente, isto é, sabendo que as mesmas não são verdadeiras;
- **Benevolência:** Concepção de que os objetivos do agente não sejam conflitantes, e que todos os agentes sempre irão tentar fazer o que lhes é solicitado;
- **Racionalidade:** Concepção de que o agente vai atuar às solicitações para alcançar os objetivos, e que não vai atuar caso este caminho impeça que os objetivos sejam alcançados.

3.4 Classificação de Agentes

Existem diversas formas de se classificar agentes. Segundo Nwana apud [BIC 99], os agentes podem ser classificados como: móveis, estáticos e reativos. Russel

e Norwig [RUS 95] classificam os agentes em quatro tipos: agentes de reflexo simples, agentes que acompanham o mundo, agentes baseados em metas e agentes baseados em utilidade. Brustoloni apud [STO 98] propõe uma classificação dos agentes em três tipos: agentes regulatórios, agentes planejadores e agentes adaptativos.

A classificação de agentes em geral é uma tipificação baseada em um conjunto de características observáveis e é uma forma que ajuda a compreender as possibilidades e limites desta tecnologia. Este tipo de classificação permite estabelecer quais agentes são mais adequados a uma determinada tarefa. Existe ainda uma proposta de classificação baseada na complexidade dos agentes conforme propõe Storb et al em [STO 98].

As diferentes capacidades para a resolução de problemas permitem que os agentes sejam divididos em categorias. Sichman [SIC 95] dividiu-os em duas categorias: agentes reativos e agentes cognitivos. Atualmente, segundo Jaques [JAQ 98a], uma outra categoria tem demonstrado grande importância, a categoria agentes de software.

Os agentes reativos se comportam conforme o modelo estímulo resposta, isto é, ele não armazena as ações já realizadas, criando uma memória das ações já efetivadas no passado. Estes agentes, internamente, apenas realizam associações de entrada e saída, não utilizando qualquer tipo de deliberação. Os agentes reativos estão baseados no modelo de organização, ou seja, sua atuação sozinha não é capaz de realizar tarefas muito inteligentes, porém, em um sistema multiagente a tarefa realizada em conjunto o é.

Segundo Corrêa apud [NUN 98], os agentes cognitivos possuem a capacidade de compreender o ambiente em que atuam, discriminam e identificam objetos e indivíduos, propriedades e relações que ocorrem entre os indivíduos em um determinado ambiente e possuem meios para manipular, extrair e utilizar a informação obtida.

Jaques, citando Janca [JAQ 98a], define um agente de software como: "um software que sabe como fazer coisas que você provavelmente faria se tivesse tempo". Este tipo de agente pode ainda ser dividido em outros dois subtipos: agentes do usuário e agentes do operador.

3.5 Sistemas Multiagentes

Um sistema multiagente é um ambiente habitado por um conjunto de agentes que interagem entre si para alcançar seus objetivos. Baseia-se, principalmente, na interação social de seus indivíduos, ou seja, cada agente é concebido como um indivíduo autônomo.

Os agentes de um sistema multiagente devem implementar um sistema de comunicação eficiente, coordenar o conhecimento existente, as metas, planos e habilidades para alcançar a realização dos objetivos individuais e, conseqüentemente, a realização da meta global do sistema.

3.5.1 Categorias de Sistemas Multiagentes

As categorias de sistemas multiagentes estão baseadas na capacidade de resolução de problemas por parte do agente e pela arquitetura do agente. Segundo Nunes em [NUN 98], existem três grandes categorias de sistemas multiagentes: sistemas multiagentes reativos, sistemas multiagentes cognitivos e sistemas multiagentes híbridos, que são apresentados na figura 3.2.

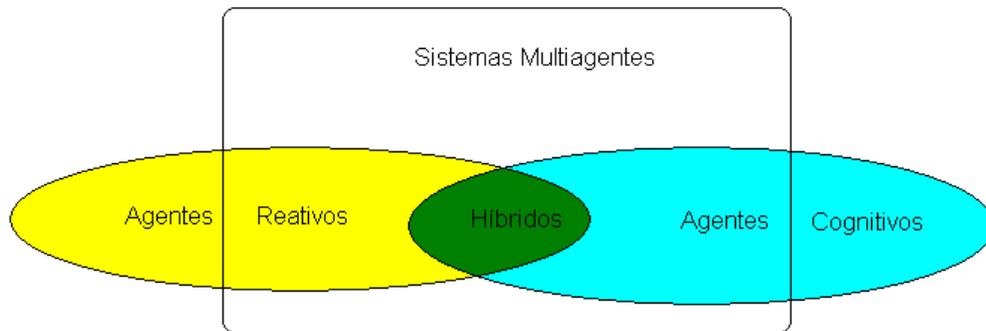


FIGURA 3.2 - Categorias de Sistemas Multiagentes

Em um sistema multiagente (SMA), algumas capacidades são imprescindíveis para que os agentes possam interagir nesse ambiente. Os agentes devem: ter conhecimento de sua existência, ter conhecimento da existência de outros agentes, ser capazes de se comunicar, possuir conhecimentos e habilidades para executar uma determinada tarefa e para cooperar na realização do objetivo global.

3.6 Sistemas de Comunicação entre Agentes

O processo de comunicação é fundamental para um SMA. Os diversos tipos de agentes presentes em uma sociedade devem comunicar-se mutuamente, para que haja interação. A comunicação pode se dar em diferentes níveis e para diversos propósitos como: decomposição, alocação de tarefas, cooperação, transmissão de resultados, resolução de conflitos, negociação, etc.

Quando os agentes necessitam se comunicar com os outros, eles podem fazer isto de várias formas. Eles podem se comunicar diretamente com o outro, desde que eles falem a mesma linguagem. Ou eles podem dialogar através de um interpretador ou facilitador, desde que eles saibam como se comunicar com o interpretador, e este possa dialogar com o outro agente.

Existem dois níveis de linguagens: um básico e outro avançado. O básico compreende o formato e a sintaxe da mensagem e o avançado, o sentido ou a semântica. A sintaxe, normalmente, é facilmente entendida enquanto a semântica não. A semântica precisa de um vocabulário de palavras e seus significados. Este vocabulário é chamado ontologia.

As comunicações podem ser classificadas em direta e indireta.

- Comunicação direta: os agentes se conhecem e, por isso, trocam informações diretamente entre si;
- Comunicação indireta: os agentes não se conhecem e, desta maneira, a comunicação irá ocorrer através de uma estrutura de dados compartilhada.

3.6.1 Arquitetura de Quadro Negro

O sistema de quadro-negro, mais conhecido como *blackboard*, foi utilizado pela primeira vez em um sistema de IA no projeto de compreensão da fala HEARSEY-II, segundo Erman et al. [RIC 93], [BIG 97]. A arquitetura de quadro-negro é bastante simples. Nesta arquitetura a comunicação ocorre por meio de uma estrutura única e compartilhada que recebe a escrita e leitura das informações que são acessadas pelos agentes durante a realização das tarefas. Como não há comunicação direta entre os agentes, eles devem consultar a estrutura periodicamente para verificar se existe alguma informação postada para ele.

Segundo Roth, citado por [NUN 98], a estrutura compartilhada geralmente é dividida em regiões ou níveis. Os agentes podem ler ou escrever em um ou mais níveis e operam sob a supervisão de um escalonador. A arquitetura de quadro-negro pode ser visualizada na figura 3.3.

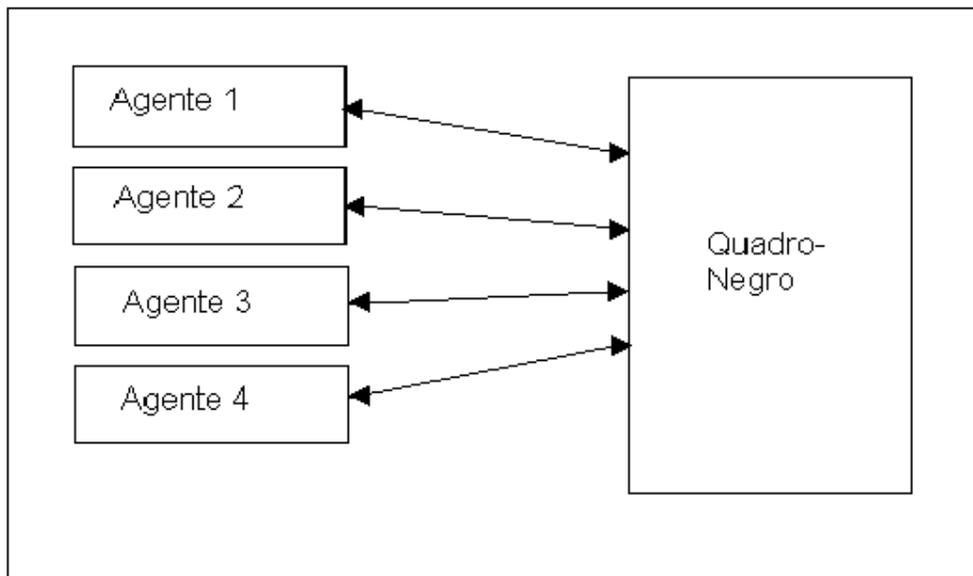


FIGURA 3.3 - Arquitetura de quadro-negro

3.6.2 Troca de Mensagens

Para que a troca de mensagens ocorra de maneira direta, os nomes dos agentes devem ser explicitamente conhecidos, os agentes devem possuir uma representação dos outros agentes da sociedade, incluindo suas capacidades, objetivos e possivelmente suas crenças e conhecimentos, de modo que possam comunicar-se entre si. O sistema de troca de mensagens pode ser visualizado na figura 3.4.

Wooldridge citado em [PAS 99], apresenta três modelos para troca de mensagens entre agentes em um SMA: ponto-a-ponto, *broadcast* e *multicast*.

- **Ponto-a-ponto:** neste modelo as mensagens são enviadas para um endereço específico que é conhecido pelo transmissor. Isto representa vantagens, uma vez que o transmissor sabe para quem a mensagem está sendo enviada. Deste modo os controles de segurança são mais facilmente implementados;
- **Broadcast:** não se baseia na passagem para um endereço específico, mas para todos os agentes que participam do sistema. Contudo, este tipo de passagem de mensagem não é seguro, visto que todos os agentes podem obter o conteúdo das mensagens;
- **Multicast:** diferente do *broadcast* este método visa à passagem de mensagens para um grupo determinado de agentes. Desta maneira quando se deseja enviar uma mensagem a um agente, ela é enviada apenas para o grupo ao qual o agente pertence.

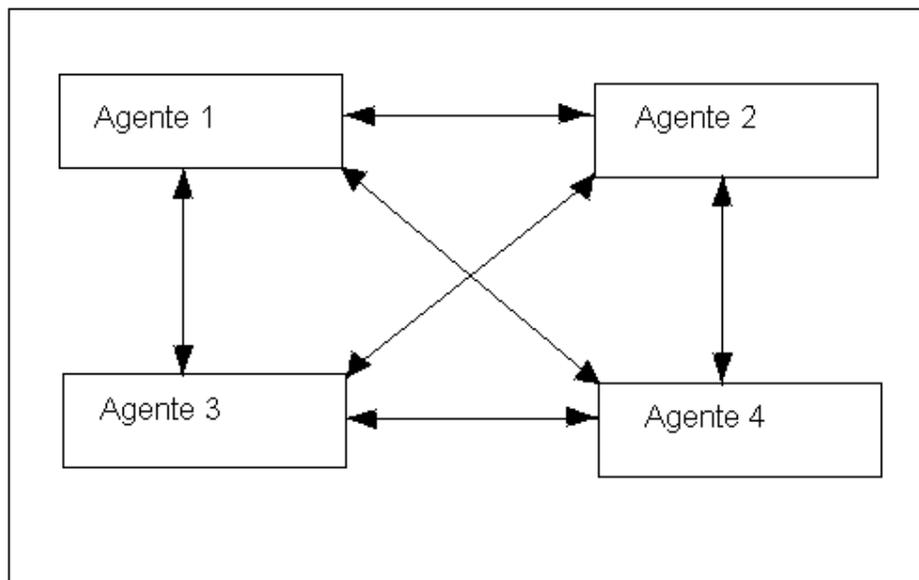


FIGURA 3.4 - Sistema de troca de mensagens

3.7 KQML

A linguagem KQML (*Knowledge query and manipulation language*) fornece um agente com sintaxe padrão para mensagens e um número de performativas que definem o motivo da mensagem,[WOO 95].

A linguagem está dividida em três camadas: a camada de conteúdo, a camada de mensagem e a camada de comunicação. A camada de conteúdo não é conhecida pela KQML e contém as descrições referentes ao conteúdo, tais como linguagem e ontologia; a camada de mensagem contém informações pertinentes a mensagem, tais como tipo de mensagem que se quer passar e o tipo de interação que vai ocorrer, sua função primária é identificar o protocolo a ser utilizado e fornecer a performativa que o emissor vai unir ao conteúdo; a camada de comunicação descreve os parâmetros de nível mais baixo como a identidade do emissor e do receptor da mensagem e o identificador da mensagem. A estrutura de uma mensagem KQML pode ser visualizada na figura 3.5 [JAQ 98].

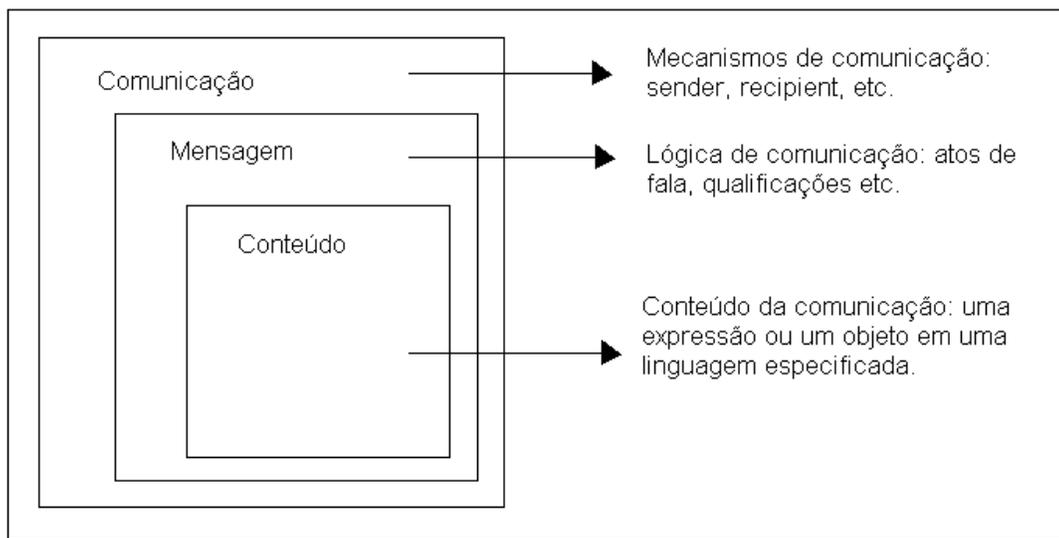


FIGURA 3.5 - Estrutura de uma mensagem KQML

A estrutura interna da camada de conteúdo não faz parte da definição do protocolo KQML, porém, é na camada de conteúdo que estão as informações que vão dar origem aos eventos relevantes. A ontologia, citada no item 3.6, através das palavras e seus significados, fornecerá ao módulo responsável pela comunicação na interface, subsídios suficientes para gerar os eventos relevantes ao agente Percepção.

Por exemplo, uma mensagem que comunica o acerto, ou não, de uma resposta fornecida pelo aluno ao STI, resultará em um evento relevante, do grupo lição, que está relacionado à tabela 5.1 que contém as regras do modelo de confiança.

4 Afetividade e Emoções

4.1 Afetividade

Ceccim, citado por [BER 99], conceitua a afetividade como todo o domínio das emoções propriamente dita, dos sentimentos das emoções, das experiências sensíveis, e principalmente da capacidade em se poder entrar em contato com sensações. Spinoza considera que se pode reunir os sentimentos humanos (paixões), como os de alegria (aqueles que compõem a vida) e os de tristeza (aqueles que decompõem a vida, os que desestimulam a viver). A afetividade implica reconhecer que os seres humanos são afetados pelo que aprendem, pelas experiências que vivenciam e pelo que desejam.

4.2 Emoções

A importância das emoções tem sido, há muito tempo, reconhecida pelos psicólogos, porém, só recentemente percebeu-se que o êxito ou o fracasso de um indivíduo em sua profissão ou em sua vida particular, depende não só da sua capacidade intelectual, mas principalmente de seu equilíbrio emocional [GAR 79] [GOL 95].

Para Daniel Goleman, existem centenas de emoções, incluindo suas combinações, variações, mutações e matizes. Na verdade, diz ele, “*existem mais sutilezas de emoções do que as palavras que temos para defini-las.*”

Na Psicologia não há consenso sobre como se define “emoção”, nem mesmo um termo abrangente que inclua todos os fenômenos que sejam de alguma forma emocionais, Gross apud [FRC 2000]. Em [AFF 2000], encontra-se algumas teorias clássicas da emoção que serão descritas a seguir.

4.2.1 Teoria de James-Lange

A famosa teoria James-Lange foi proposta por William James em 1884 e também, de forma independente, por C. Lange em 1885.

De acordo com esta teoria, as ações precedem as emoções e o cérebro interpreta ações ditas como emoções. A situação ocorre e o cérebro interpreta a situação, causando a característica reação fisiológica. Esta reação pode incluir alguma ou todas as seguintes características fisiológicas: suor, elevação dos batimentos cardíacos, expressão facial e gesticulação.

Estas reações reflexivas ocorrem antes da pessoa estar ciente de que está experimentando uma emoção; somente quando o cérebro avalia cognitivamente a fisiologia é que ela é rotulada como “emoção”.

A opinião comum, que diz que uma pessoa chora quando está triste ou ri quando está feliz ou alegre; a expressão corporal sucede a tristeza ou a alegria. Henry Garret [GAR 79] cita que, contrariando essa opinião comum, Willian James escreveu:

“Minha tese, ao contrário, é de que as mudanças corporais sucedem diretamente à percepção do fato excitante, e que nossa sensação das mesmas mudanças, à medida que ocorrem, é a emoção” .

4.2.2 Teoria de Cannon e Bard

Cannon Woodwort e Bard opõem-se à teoria de James-Lange por declarar que a emoção é percebida primeiro, e então ações seguem da apreciação cognitiva. Na visão dele, o tálamo e amígdala representam a função central; interpretando uma emoção - provocando situação e simultaneamente enviando sinais para o ANS (autonomic nervous system) e para o córtex cerebral que interpreta a situação cognitivamente [AFF 2000].

4.2.3 Teoria de Woodwort

Segundo Woodwort citado em [GAR 79], uma emoção é um estado de agitação do organismo, em que existe um impulso ou atitude consciente, levando a uma atividade definida. Salienta-se que o termo “agitação” significa um estado de atividade ou excitação.

Nesta teoria, surge em primeiro lugar o estímulo; depois o impulso ou disposição motora para escapar ou lutar, a qual por sua vez é seguida de mudanças fisiológicas.

Um exemplo típico é o do automóvel que avança sobre um indivíduo, imediatamente ele salta para a calçada, o medo toma conta dele no momento em que salta, ou imediatamente após o salto.

4.2.4 Teoria de Schachter-Singer

Schachter e Singer concordam com James e Lange que a experiência das emoções aparecem da classificação cognitiva de uma sensação fisiológica. Entretanto, eles também acreditam que esta não foi suficiente para explicar a mais sutil diferença na emoção autopercepção, por exemplo, a diferença entre raiva e medo. Desta forma, eles propuseram que uma pessoa vai obter informação de uma situação imediata (por exemplo: um perigo próximo) e usa isto para qualificar a sensação.

4.3 Dispositivos Sensores de sinais afetivos humanos

Sensores são uma parte importante dos sistemas de computação que são utilizados no reconhecimento de pistas afetivas. Eles podem acumular dados de forma contínua sem ter que interromper o usuário e provêm informações sobre o comportamento ou estado físico de quem os usa. A seguir encontra-se uma descrição de alguns dispositivos utilizados para reconhecer sinais afetivos [AFF 2000b].

4.3.1 Prototype Sensing System

O protótipo do Sistema de Sensoriamento fisiológico foi desenvolvido para ser usado em sistemas operacionais baseados em UNIX, figura 4.1. O sistema inclui um sensor GSR (*Galvanic Skin Response*), um sensor BVP (*Blood Volume Pulse*), um sensor de respiração e um sensor EMG (*Electromyogram*). O sistema é leve, portátil e relativamente robusto para mudança de posição no usuário; o sistema é totalmente externo, o que ajuda a manter a privacidade de quem o usa.

O protótipo do sistema de sensoriamento é a combinação de quatro sensores: BVP, GSR, EMG e sensor de respiração. Cada um destes sensores é descrito em detalhes. Todos os sensores estão ligados a unidade codificadora, um dispositivo que recebe o sinal e os traduz para o formato digital.

4.3.2 Sensor GSR

O GSR é um aparelho que mede a condutividade da pele entre dois elétrodos. Os elétrodos são pequenas placas de metal aplicadas de forma segura, imperceptivelmente uma pequena voltagem passa através da pele. Os elétrodos são tipicamente fixados sobre o dedo da pessoa. Para medir a resistência, uma pequena voltagem é aplicada sobre a pele e a condutividade da pele é medida.

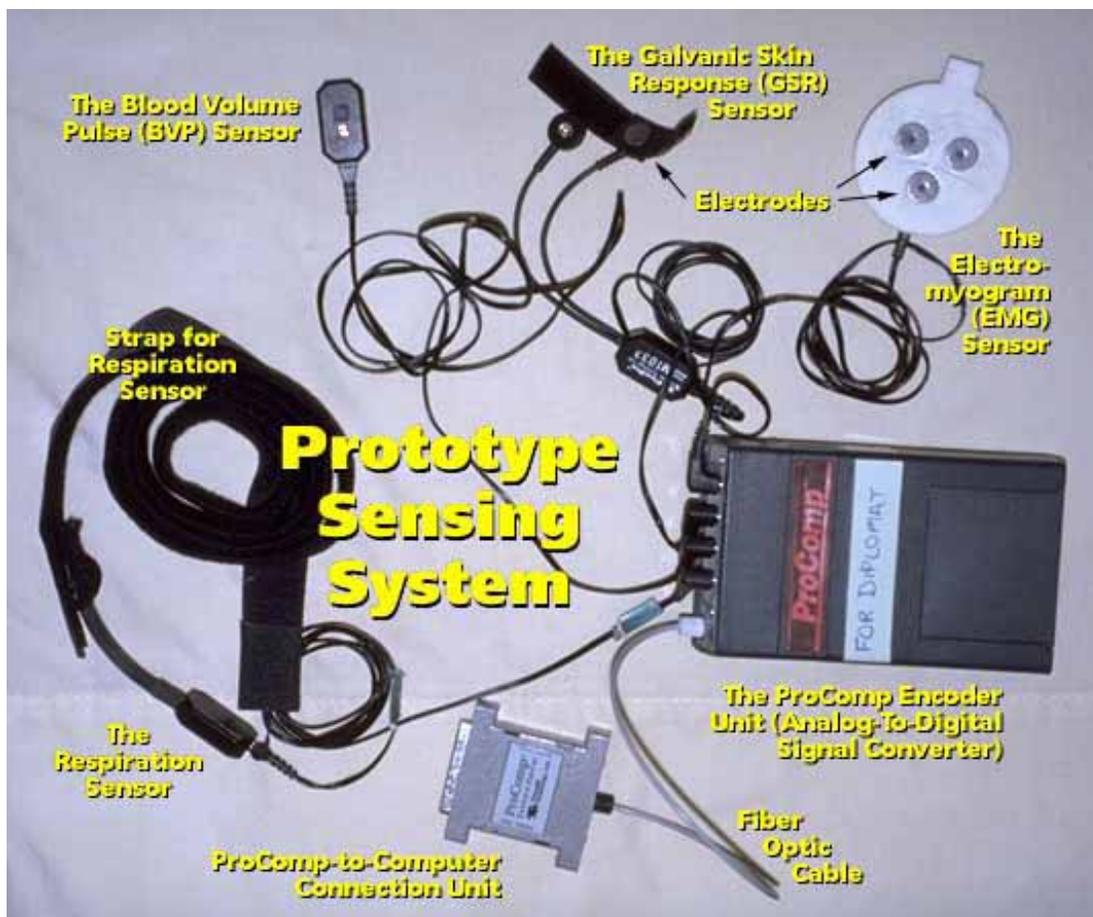


FIGURA 4.1 - Protótipo do sistema de sensoriamento fisiológico

4.3.3 Sensor BVP

O sensor de pulso sangüíneo usa o *photoplethysmography* para detectar a pressão do sangue nas extremidades. *Photoplethysmography* é um processo em que se aplica uma luz e mede-se a luz refletida pela pele. A cada contração do coração, o sangue é forçado através dos vasos periféricos, produzindo o enchimento dos vasos sob o efeito da luz e conseqüentemente modificando a quantidade de luz refletida ao fotosensor.

4.3.4 Sensor de respiração

O sensor de respiração pode ser colocado tanto sobre o esterno, para o monitoramento torácico, como sobre o diafragma. O sensor consiste principalmente de um grande cinto de velcro que circunda a cavidade do tórax e de um pequeno elástico que alonga com a expansão do tórax. O alongamento do elástico é medido em volts e a sua variação é gravada.

4.3.5 Sensor EMG (Electromyogram)

O sensor EMG mede a atividade do músculo (atividade elétrica produzida pelo músculo quando ele está contraindo), amplificando o sinal e enviando para um codificador. No codificador, o sinal é submetido a um filtro. O sensor opera na faixa de 0 a 400 microvolts e o filtro na faixa de 20 a 500 Hz.

4.4 Interface Sensora de sinais afetivos

Muito esforço tem sido feito no sentido de compreender as relações da emoção e como esta pode ser identificada pelo computador. Mesmo quando se medem alguns fatores físicos como temperatura ou a pressão sanguínea não é possível reconhecer as emoções das pessoas. As pesquisas voltadas ao reconhecimento da emoção estão limitadas em correlacionar as manifestações emocionais que podem ser percebidas pelo computador, tais como: fisiologia, comportamento e seleção regular de palavras usadas durante um diálogo [AFF 2000b].

Na avaliação da emoção do usuário, pode-se também medir o auto-relato de cada indivíduo e como eles se sentem. Entretanto, isto é frequentemente considerado não confiável, visto que o auto-relato varia com o pensamento e situações como: “É conveniente dizer como me sinto?” ou “Como descrever como eu me sinto agora, de qualquer jeito?”. Muitas pessoas têm dificuldade para reconhecer e/ou expressar verbalmente suas emoções, especialmente quando estas são uma mistura de emoções ou quando as emoções não são descritíveis. Em muitas situações também é inapropriado interromper o usuário para obter um auto-relato. Apesar de tudo, pensa-se que isto é importante se o usuário quiser contar verbalmente para o sistema seus sentimentos. O sistema deve facilitar isto.

O interesse está nas expressões emocionais, sejam estas expressas de forma verbal ou outro meio qualquer, não interessa exatamente o que foi dito, mas como foi dito, a simples escolha da palavra pode revelar um fator afetivo escondido.

O uso isolado da interface restringe consideravelmente, a capacidade de identificar sinais afetivos, porém, dependendo do sistema que o utiliza esta limitação se faz necessária. A proposta de integração do agente Percepção ao STI Eletrotutor é um exemplo típico, desta restrição. Porém, com o uso de uma interface, de um conjunto de regras e de um mecanismo que monitore as ações realizadas pelo aluno e pelo STI, é possível revelar importantes sinais afetivos, que estavam escondidos nesse processo.

4.5 Sistemas que utilizam afetividade

Atualmente, a identificação de fatores afetivos tem sido utilizada em diversas linhas de pesquisa, com os mais diferentes propósitos, como por exemplo, os projetos citados em [AFF 2000a]:

- *Affective Toys (brinquedos afetivos)*. O *Affective Tigger* é um brinquedo que expressa reações, ele reage às jogadas de seu oponente exibindo sua emoção, baseado na sua percepção sobre o tipo da jogada;

- *Computer Response to User Frustration* (Resposta de computador a frustração do usuário), O agente de interação homem-máquina foi projetado e construído para dar suporte aos usuários em suas habilidades de recuperação de fatores emocionais negativos, particularmente à frustração;
- *Driver's affect* (afetividade do motorista), é uma tentativa em que o carro e o motorista são equipados com vários sensores, com o objetivo de reconhecer importantes fatores afetivos como a raiva ou o estresse;
- *Orpheus, the affective CD player* (Orpheus, um tocador de CD afetivo), escolhe as músicas que serão reproduzidas em função do estado emocional do usuário e suas preferências;
- *Affective Tutor* (Tutor afetivo), utiliza as estratégias de ensino mais indicadas de acordo com o estado afetivo atual do aluno.

4.6 Modelagem dos fatores afetivos

A modelagem dos fatores afetivos pode ser adaptada para refletir os aspectos afetivos de um usuário em particular. No entanto, a noção de sistemas que aprendem com a interação do usuário, pode ser importada de um sistema robusto para solução de problemas de reconhecimento de padrões afetivos. A figura 4.2 apresenta os estágios de um processo de reconhecimento [AFF 2000a].

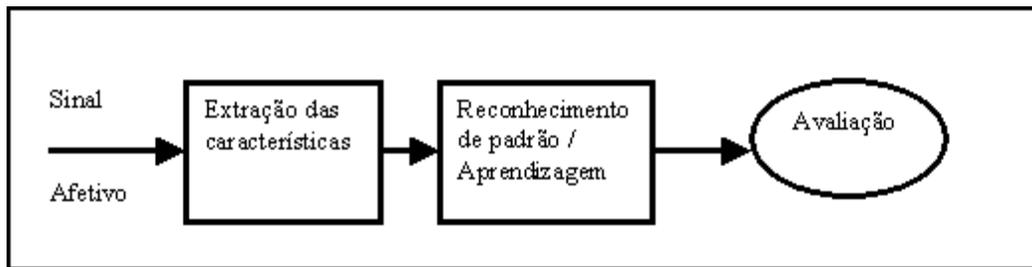


FIGURA 4.2 - Diagrama de um módulo reconhecedor de padrão afetivo

O problema de reconhecimento de padrão consiste da classificação dos dados observados sobre um conjunto de fatores (classes ou categorias) que correspondem a estados emocionais distintos. Quais ferramentas são mais adequadas para acompanhamento dos tipos de estados dos sinais observados.

O primeiro bloco, denominado extração das características, recebe todos os sinais recebidos do ambiente, seleciona os relevantes e repassa ao bloco seguinte, denominado reconhecimento de padrão aprendizagem. É ele que delibera sobre os sinais captados e selecionados, infere suas regras e apresenta o resultado da avaliação que será utilizada por outro módulo do sistema.

4.7 A abordagem afetiva usada neste trabalho

O conceito de afetividade usado neste trabalho, item 4.1, abrange toda a parte do domínio das emoções, como os sentimentos e as experiências sensíveis mas, de modo especial, a capacidade em se poder fazer contato com as sensações.

Os conceitos de emoção, citados neste capítulo, servem para demonstrar que apesar de não haver um consenso entre os pesquisadores da área sobre o que é emoção, todos os pesquisadores estabelecem uma relação entre as emoções e as reações fisiológicas [GAR 79][AFF 2000].

Apesar da forte ligação entre emoções e reações fisiológicas, as reações fisiológicas, sozinhas, não fornecem subsídios suficientes para identificar as emoções e quase sempre estão associadas a aspectos comportamentais.

Os aspectos comportamentais, por sua vez, são tão evidentes, que em alguns casos, mesmo sendo usados sem auxílio dos sensores fisiológicos, são capazes de indicar fortes indícios de alguma manifestação emocional.

O uso de uma interface como sensor de sinais afetivos, neste trabalho, tem como base, as características que ela apresenta. Entre estas características destacam-se: utilização de aspectos comportamentais para detecção dos sinais afetivos, facilidade de distribuição, baixo custo, não causa incômodos. A distribuição pode ser feita pela Internet, o único custo é o do desenvolvimento e não há necessidade de se fixar qualquer dispositivo no usuário.

5 Estratégia de Identificação

Apesar de um grande número de pesquisas sobre a motivação humana, ainda não se sabe corretamente como diagnosticar a motivação nos outros (empatia). Conforme Lepper apud [BER 97], o diagnóstico de emoções normalmente é retirado de padrões comportamentais, isto é, um professor analisa além das expressões verbais, o tom em que elas foram verbalizadas (comportamento), a conduta da postura e das expressões faciais, e as expressões e comportamentos em geral de um aluno frente a tarefas ou interações de ensino aprendizagem.

A descoberta do estado motivacional do estudante é efetivamente restringida pelas limitações da interface. Entretanto, o esforço do estudante em lugar de desempenho é uma indicação razoavelmente fidedigna de motivação intrínseca segundo Keller apud [SOL 95]. Um modo paralelo de obter informações sobre o estudante é a aplicação regular de questionários durante a interação, que concluam em sua própria auto-avaliação e em uma avaliação sobre o comportamento do sistema. O sistema também deve explorar procedimentos de reação repetitivos como, por exemplo, quando os alunos pedem ajuda antes de tentar resolver um problema (indicando, possivelmente, baixa confiança), ou em caso contrário, na ausência total de pedidos de ajuda durante a interação inteira (indicando confiança extremamente alta, possivelmente).

A análise motivacional de um aluno dá-se a princípio por quatro fontes diferentes, segundo [SOL 95]: questionário no início da sessão; comunicação com o estudante durante a interação; auto-avaliação do estudante de seus estados motivacionais durante a interação; solicitações de ajuda e perseverança para completar a tarefa.

A utilização do questionário no início da sessão é uma importante forma de se obter dados para a análise do estado motivacional do aluno, porém, esta se dá no início da sessão; sabendo-se que o estado motivacional é dinâmico, ter-se-ia que aplicar esta técnica durante a utilização do tutor, interrompendo a seqüência de ação do usuário e até mesmo alterando o estado emocional do mesmo. Por este motivo, esta técnica não será usada neste trabalho.

A comunicação com o estudante durante a interação pode ser utilizada de várias formas, sempre com perguntas rápidas, objetivas e pertinentes ao contexto em que o aluno se encontra. Esta técnica foi pouco explorada neste trabalho, visto que seu resultado, normalmente, depende da sinceridade do aluno na resposta e serve principalmente para confirmar o estado motivacional do aluno já detectado por outro método. Neste trabalho utiliza-se a comunicação com o estudante para verificar seu nível de independência. Por exemplo: o tutor verifica que o aluno errou uma resposta e sugere o uso da ajuda, que pode ser aceita pelo estudante, indicando uma redução na independência ou pode ser recusada, indicando um alto nível de independência.

A auto-avaliação do estudante, durante a interação, também não foi utilizada. A sua funcionalidade depende, quase sempre, da vontade do estudante em fornecer o seu estado motivacional. E mesmo quando fornecido, não é totalmente seguro, considerando-se que o aluno pode não estar consciente do seu estado motivacional ou pode estar tentando burlar a estratégia motivacional do tutor.

O acompanhamento das atividades realizadas pelo aluno, como o uso de ajuda, número de tentativas para realizar uma determinada tarefa e a desistência antes de completá-la, também são indicadores usados para identificação dos fatores motivacionais e afetivos.

A partir da monitoração das ações executadas pelo aluno e das reações realizadas pelo STI, pode-se obter os estados motivacionais e/ou afetivos do aluno, cujo comportamento se está observando.

5.1 Fatores Motivacionais

Dentre os fatores afetivos citados por Melchior em [BER 97] e dos estados motivacionais propostos por Keller, Malone e Lepper em [SOL 95], destacam-se os seguintes fatores motivacionais / afetivos que serão identificados:

- Confiança;
- Esforço;
- Independência.

Confiança – É a descoberta do nível de confiança do estudante que é atribuída principalmente às convicções que ele tem sobre sua eficácia em executar a tarefa proposta.

Estudantes menos confiantes estão mais propensos a: evitar tarefas recebidas como difíceis; desistir de uma tarefa antes de tentar executá-la.

Esforço - É considerado, neste trabalho, como o trabalho despendido com o objetivo de alcançar o resultado, não importando se o resultado alcançado está correto ou errado. O enfoque dado ao esforço refere-se ao trabalho que pode ser observado, como por exemplo, o número de tentativas para resolver o problema.

Segundo Keller citado em [SOL 95], "esforço é um indicador direto de motivação".

Independência - O estado de independência do estudante relaciona-se com a sensibilidade em perceber a necessidade ou não de usar a ajuda do sistema tutor para realizar a tarefa. Por exemplo, quando o sistema tutor intervém em excesso, seja por iniciativa própria, em prover sugestões, ou para atender os pedidos de ajuda solicitados pelo usuário corresponde a um modelo de baixa independência. Ao contrário, o modelo de independência é aumentado quando o estudante é liberado para explorar seus próprios problemas ou quando ele rejeita a ajuda oferecida pelo sistema tutor.

5.2 Modelo de Confiança

A confiança é representada como um valor na faixa de 0 a 10. A atribuição do valor inicial deste fator é feita, no início da sessão, com o valor médio 5.

A atualização é feita através de incrementos ou decrementos que podem ser de um ou dois pontos. Em qualquer momento da interação, a confiança do aluno é um valor entre 0 e 10. O valor 0 representa uma confiança muito baixa e o valor 10 representa confiança máxima. O valor que representa a confiança do aluno é atualizado dinamicamente através das regras descritas na tabela 5.1.

TABELA 5.1 - Modelo de Confiança

Regra	Situação	Estado de Ajuda	Confiança
CONF1	nenhuma tentativa	Solicita ajuda	-1
CONF2	Acerta	Com ajuda	+1
CONF3	Acerta	Sem ajuda	+2
CONF4	Erra	Sem ajuda	-1
CONF5	Erra	Com ajuda	-2

5.3 Modelo de Esforço

A tabela 5.2 apresenta um modelo para classificação do esforço do estudante em função de sua persistência na resolução de problemas e solicitações de ajuda para resolver a tarefa proposta. A classificação aceita para os diversos níveis de esforço estão definidas como: nenhum, mínimo, pequeno, médio, grande e máximo. A atribuição do valor inicial deste fator é feita, no início da sessão, com o valor 3 que correspondente ao esforço médio.

Assume-se, que a persistência na resolução do problema pode ser medida pelo número de tentativas de solução ou pelo número de passos realizados, sendo que o um elevado número de passos reflete um alto grau de esforço por parte do aluno. A quantificação do número de passos, para a classificação de muitos ou poucos passos, deve ser definida por um especialista do domínio, em função do grau de dificuldade do problema. Um aluno que solicita ajuda com poucos passos, realiza um esforço pequeno, em comparação a outros que continuam tentando a solução da tarefa sem ajuda do tutor. O resultado da realização da tarefa é outro fator importante. A desistência de uma dada tarefa sem a tentativa de solução, obviamente, demonstra uma baixa persistência.

TABELA 5.2 - Modelo de Esforço

Regra	Estado da Tarefa	Passos	Ajuda	Esforço
ESF1	Desistiu	Nenhum	-	Nenhum
ESF2	Desistiu	Poucos	Com ajuda	Mínimo
ESF3	Desistiu	Poucos	Sem ajuda	Pequeno
ESF4	Realizou	Poucos	Com ajuda	Pequeno
ESF5	Realizou	Poucos	Sem ajuda	Médio
ESF6	Desistiu	Muitos	Com ajuda	Médio
ESF7	Desistiu	Muitos	Sem ajuda	Grande
ESF8	Realizou	Muitos	Com ajuda	Grande
ESF9	Realizou	Muitos	Sem ajuda	Máximo

5.4 Modelo de Independência

O modelo de independência é similar ao modelo de confiança em muitos aspectos. A independência é representada como um valor numérico, tem um valor mínimo e um máximo e é atualizada através de incrementos e decrementos. A faixa assumida é de 0-10 e o valor inicial atribuído no início da sessão é 5. Considerar-se-á quatro situações básicas: o tutor oferece ajuda, o tutor fornece ajuda genérica, o tutor fornece ajuda específica, o aluno rejeita a oferta de ajuda. A tabela 5.3 relaciona as regras para incremento e decremento do valor de independência.

TABELA 5.3 - Modelo de Independência

Regra	Estado da ajuda	Tipo de ajuda	Independência
INDE1	Sugerida	-	-1
INDE2	Fornecida	Genérica	-1
INDE3	Fornecida	Específica	-2
INDE4	Não Utilizada	-	+1
INDE5	Negada	-	+1
INDE6	Rejeitada	-	+2

6 Descrição da arquitetura proposta

O protótipo atual foi desenvolvido tendo como base a arquitetura de agentes autônomos implementados no Eletrotutor III [BIC 99] e posteriormente adaptada para o Eletrotutor IV [SIL 2001]. A principal alteração realizada na arquitetura do STI foi: a inclusão de um novo agente, denominado Percepção. O novo agente foi inserido junto ao, já existente, agente Interface, conforme a figura 6.1.

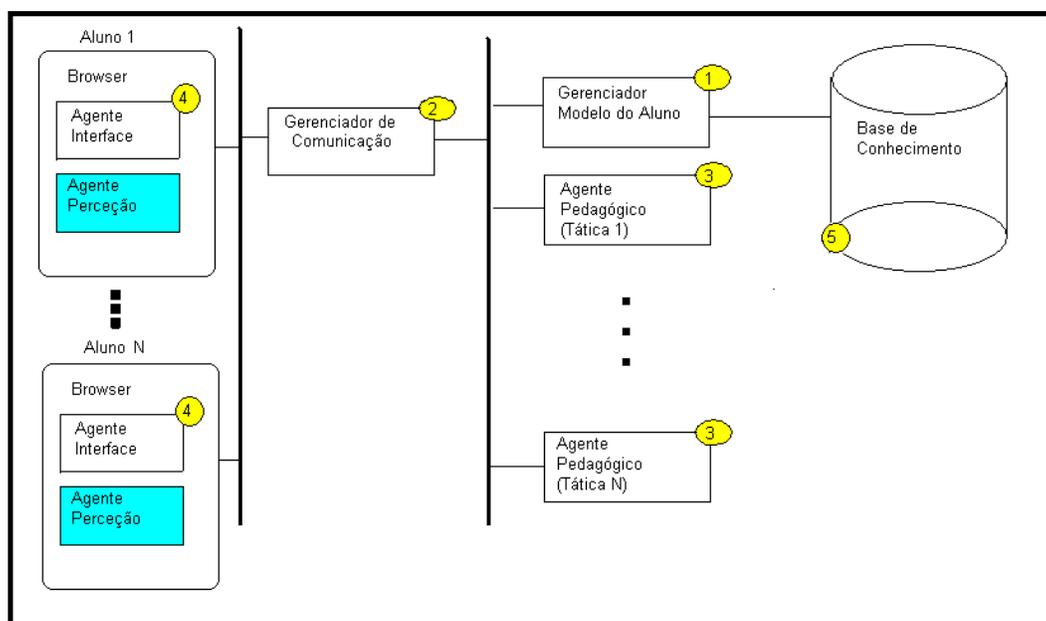


FIGURA 6.1 - Arquitetura alterada do Eletrotutor IV

6.1 Funções dos agentes

As funções dos agentes enumerados entre um e três permanecem inalteradas e mantendo as características descritas em [SIL 2001]. O agente quatro é a proposta deste trabalho. O agente Gerenciador de comunicação (agente 2) é o responsável pela gerência de toda comunicação que ocorre entre o Agente de Interface e os demais agentes do sistema. Todas estas mensagens são recebidas e enviadas através de Java RMI. O agente pedagógico (agente 3) é responsável pela tática de ensino. Existe um para cada tática de ensino definida pelo especialista do curso e suas tarefas podem variar conforme a necessidade de cada agente, porém algumas tarefas são previamente definidas a todos os agentes genéricos, entre as quais estão: mostrar o conteúdo atual, avançar para o próximo conteúdo, voltar ao conteúdo anterior, atualizar histórico e comunicação. O agente gerenciador do modelo do aluno (agente 1) é responsável pela criação e manutenção de uma base do conhecimento (5), que mantenha a representação cognitiva de cada um dos alunos que estejam ou já estiveram conectados ao sistema. Esta manutenção compreende não só a recuperação dos dados do aluno no momento em que ele efetua o login, como também a atualização do histórico do aluno sempre que o Agente Pedagógico enviar dados para esta função. Além dessas funções, nesta versão,

passa a agregar um novo conjunto de funções que darão suporte aos estados motivacionais identificados pelo agente percepção, conforme descrito no item 6.2.3 modelo do aluno com afeto.

6.2 Requisitos

Para que o agente Percepção possa desempenhar sua tarefa com sucesso faz-se necessário que o STI apresente, compulsoriamente, uma série de requisitos que serão classificados em quatro grupos: requisitos quanto à organização do domínio, requisitos quanto ao sistema de ajuda, requisitos quanto às estratégias de ensino e requisitos quanto aos eventos relevantes, ou seja, um modelo de aluno com afeto.

Neste trabalho será considerada como Lição, um conjunto de atividades que se subdivide em passos e que possua pelo menos uma avaliação.

Considera-se como ajuda somente aquelas referentes ao conteúdo pedagógico, não são consideradas as ajudas referentes ao uso do STI. Início/Término é um grupo de eventos que controla a entrada e saída do usuário do sistema.

6.2.1 Organização do Domínio

É necessário que o domínio esteja dividido em lições e que cada lição seja composta por um conjunto de atividades. Esse conjunto de atividades deve conter pelo menos um exercício que seja passivo de avaliação. Também é necessário que o sistema armazene, para cada lição, um valor que represente a quantidade prevista do número de passos necessários para execução, com sucesso, do exercício da lição. Sendo que este número deve ser fornecido pelo especialista do domínio.

Caso o aluno resolva o exercício, corretamente, utilizando-se de menos passos que o previsto, considera-se que ele se utilizou de poucos passos, caso contrário, considera-se que foram utilizados muitos passos.

6.2.2 Sistema de ajuda

É requisito que o STI disponha de um sistema de ajuda ao aluno. Esta ajuda, para ser válida, deve ser voltada para o conteúdo do domínio e deve estar disponível em dois níveis: genérico e específico. A classificação da ajuda depende do domínio do conteúdo que está sendo ensinado e deve ser feita pelo especialista do domínio.

A ajuda genérica é teórica, dada em forma de conceito. Por exemplo: se o aluno solicita a ajuda durante a resolução de um problema, a ajuda genérica provê a fórmula necessária, traz uma explicação sobre as variáveis utilizadas e pode até informar sobre detalhes de procedimentos para se resolver a equação, porém, nunca se deve fazer referência específica ao problema que está sendo solucionado.

A ajuda específica, ao contrário da anterior, é prática, dada em forma de exemplo, ou seja, faz referência ao problema que se está resolvendo. Sempre que

possível é um exemplo bem próximo do problema proposto, podendo até ser o mesmo problema com valores diferentes, ou em casos excepcionais, com os mesmos valores.

Neste trabalho as ajudas genérica e específica diferenciam-se, ainda, pela estrutura de apresentação. Sempre que uma ajuda é solicitada ou sugerida apresenta-se a ajuda genérica que obrigatoriamente contém *links* para uma ou mais ajudas específicas.

A estrutura de organização da ajuda que é utilizada neste trabalho, faz com que toda ajuda sugerida ou solicitada a partir da resolução do problema, seja do tipo genérica, e toda ajuda específica seja solicitada a partir do próprio ambiente de ajuda. Em outras palavras, duas ou mais solicitações de ajuda, seguidas, resultam em ajuda específica, a primeira ajuda de uma seqüência ou uma ajuda isolada resulta em ajuda genérica.

6.2.3 Modelo de aluno com afeto

O modelo do aluno não executa a identificação dos fatores afetivos. Entretanto, este modelo deve receber a informação do fator afetivo atual do aluno e construir a “visão” afetiva para o diagnóstico do estado global do aluno. Isto faz com que o modelo do aluno seja um dos requisitos mais relevantes, pois sem o modelo de aluno com afeto, os fatores afetivos até poderiam ser detectados, mas não haveria o devido tratamento por parte do STI e conseqüentemente não haveria reação.

O modelo do aluno deve estar apto a receber e reagir aos fatores motivacionais e afetivos definidos no item 5.1 deste trabalho: esforço, confiança e independência. O esforço pode assumir seis estados distintos: nenhum, mínimo, pequeno, médio, grande e máximo. A confiança pode assumir um valor inteiro entre 0 (zero) e 10, onde o valor 0 (zero) representa o nível de confiança mais baixo que um aluno pode assumir, e o valor 10 representa o nível mais alto, ou seja, confiança máxima. A independência segue a mesma representação numérica utilizada para confiança, ou seja, pode assumir um valor inteiro entre 0 (zero) e 10, onde o 0 (zero) representa o nível mais baixo de independência e o 10 representa o nível máximo de independência do aluno.

Magda Bercht propõe um modelo de aluno com afeto, em [BER 2001], que possui dois esquemas distintos, um para tratamento dos dados relativos à parte cognitiva e outro para tratamento dos dados relativos à parte afetiva. Os esquemas são nomeados, respectivamente, esquema cognitivo e esquema afetivo. O modelo prevê ainda a conciliação dos resultados de cada esquema, que resulta em um estado global do aluno e servirá para a definição das ações tutoriais a serem realizadas. A representação deste modelo pode ser visualizada na figura 6.2.

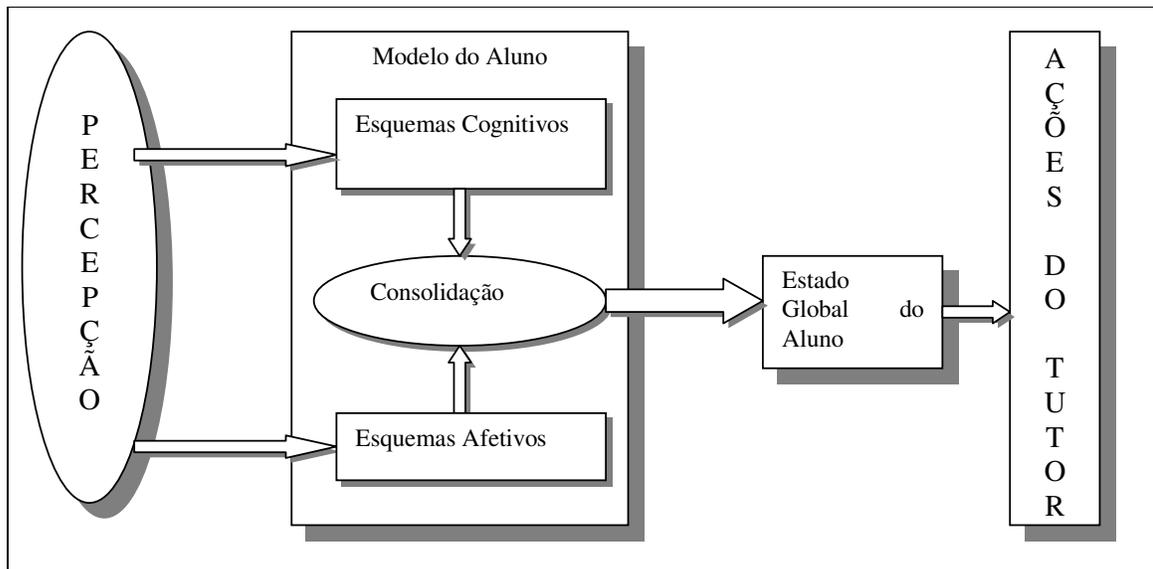


FIGURA 6.2 - Modelo de Aluno com Afeto

Os esquemas cognitivos apresentados na figura 6.2, representam o conjunto de funções e regras, que tratam das condições do aluno em relação ao domínio, e são formados pela performance do aluno, seus erros e acertos, os tópicos que já domina e aqueles que ele deve desenvolver. Os esquemas cognitivos são, em outras palavras, um conjunto de controles, cujo propósito é determinar a crença do tutor em relação ao conhecimento que o aluno possui sobre o domínio.

Os esquemas afetivos representam o conjunto de controles e regras, relativos às informações afetivas determinadas na interface. Em um primeiro momento serão analisados os fatores motivacionais e afetivos como um todo. Como resultado dessa análise, os esquemas afetivos irão determinar quais fatores afetivos devem ser modificados e qual a prioridade de cada um. Esse resultado será repassado ao módulo de conciliação.

O módulo de conciliação recebe os resultados provenientes dos esquemas afetivos e cognitivos e, nesta etapa, os dados resultantes dos dois esquemas são consolidados e enviados como estado global do aluno para o módulo do tutor responsável pela determinação das ações do tutor.

6.2.4 Eventos Relevantes

Os eventos relevantes ao processo de identificação são as ações, reações e comportamentos que possam contribuir para alterar um ou mais fatores afetivos. Para facilitar o estudo, eles foram divididos em três grupos: Lição, Ajuda e Controle de sessão. Para que o agente percepção possa exercer sua função corretamente é fundamental que o STI gere e permita a identificação dos eventos relacionados a cada

grupo. A figura 6.3 apresenta um esquema hierárquico dos eventos e atributos relevantes.

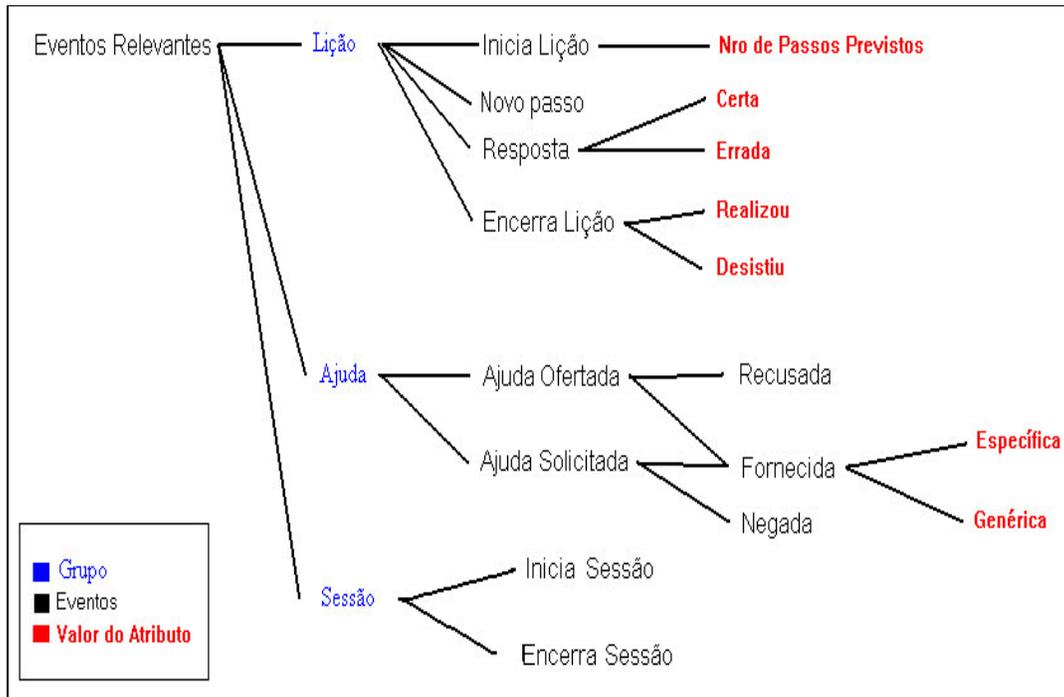


FIGURA 6.3 - Esquema hierárquico dos eventos e atributos relevantes.

Para o grupo lição os eventos relevantes são: inicia lição, novo passo, resposta e encerra lição. Quando for gerado um evento inicia lição. Este deve vir acompanhado do valor que representa o número de passos previstos para conclusão da lição. Este número deve ter sido fornecido pelo especialista do domínio, conforme descrito no item 6.2.1. O evento encerra lição possui uma estrutura um pouco mais complexa, além de necessitar de uma informação adicional informando se a tarefa foi encerrada no fim (completou) ou foi encerrada antes do fim (tarefa abandonada pelo aluno); caso ela tenha sido encerrada porque chegou ao fim o STI deve gerar um outro evento, denominado resposta, que vai reportar uma resposta afirmativa ou negativa relativa ao sucesso do aluno em relação à lição, em outras palavras certo ou errado. O evento novo passo não necessita de informação adicional.

Para o grupo ajuda os eventos relevantes são: ajuda sugerida e ajuda solicitada. A ajuda ofertada é quando o STI exibe uma mensagem para o aluno questionando se ele aceita a ajuda do tutor. E ajuda solicitada é quando o aluno solicita ajuda através de algum recurso disponibilizado na interface, seja um botão específico ou a tecla padrão <F1>. Quando a ajuda é ofertada, o STI deve aguardar um outro evento como resposta, que pode ser aceitou ou recusou. Se a ajuda foi solicitada o STI deve verificar se o nível de independência do aluno está muito baixo e o nível de confiança do aluno está muito alto. Neste caso ele deve negar a ajuda, caso contrário, a ajuda deve ser fornecida normalmente. Se a ajuda foi ofertada e aceita ou foi solicitada, o sistema

gera um evento que será chamado de ajuda fornecida e deverá ser acompanhada de um atributo que qualifique o tipo de ajuda. Os valores possíveis são: genérica e específica.

Para o grupo controle de sessão os eventos possíveis são: inicia sessão e encerra sessão. O evento inicia sessão serve para o agente atribuir os valores iniciais aos seus controles, e o evento encerra sessão chama o evento encerra lição podendo caracterizar ou não o abandono da lição. Este evento faz-se necessário porque o usuário pode encerrar o tutor de várias formas, por exemplo, fechando o *browser*, e neste caso antes de encerrar o STI, é necessário salvar o histórico e outros dados que sejam necessários.

6.3 Agente Percepção

O agente Percepção, apresentado na figura 6.4, monitora todas as comunicações que ocorrem entre a interface e o STI. As informações relevantes são armazenadas em um histórico (log), onde são aplicadas regras que identificam os fatores motivacionais e afetivos do aluno, durante todo o tempo em que o Tutor é utilizado. Cada vez que um dos três fatores é alterado, o agente Percepção envia esta informação para que os esquemas afetivos, do modelo do aluno, sejam consolidados e gerem as ações tutoriais necessárias.

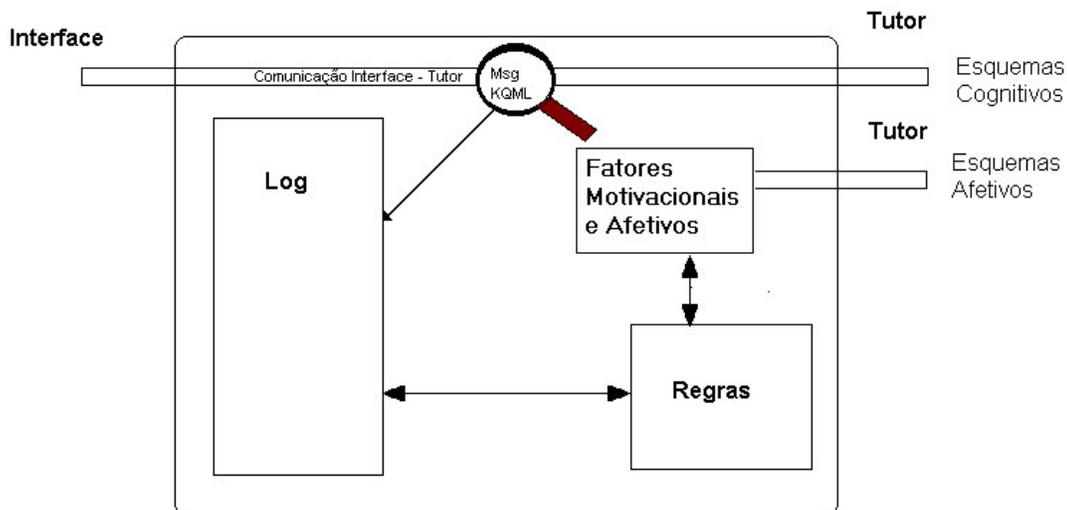


FIGURA 6.4 - Arquitetura do Agente Percepção

O funcionamento do agente Percepção dá-se da seguinte forma:

Toda mensagem que é transmitida entre o Tutor e a interface, em qualquer sentido, é verificada. Se ela gerar um dos eventos relevantes descritos no item 6.2.4 uma inserção com os dados do evento é inserida na lista de históricos. A cada inserção de um item no histórico, executa-se uma ou mais regras que,

conseqüentemente, podem alterar ou não o valor atribuído a um dos fatores motivacionais e afetivos. Se algum dos fatores sofrer alteração em decorrência da aplicação de alguma regra, uma mensagem correspondente é enviada ao Tutor.

As mensagens enviadas do agente Percepção para o STI têm como destino o modelo do aluno que, neste caso, deve estar preparado para receber tanto as informações referentes ao esquema cognitivo quanto às informações referentes ao esquema afetivo.

6.4 Sistema de comunicação

O sistema de comunicação utilizado no STI é de extrema importância ao funcionamento do agente Percepção, uma vez que este utiliza o conteúdo das mensagens com sua fonte de informação.

O Eletrotutor IV utiliza-se de dois métodos de comunicação: o JDBC-Thin da Oracle que é utilizado na comunicação entre os agentes e o banco de dados, e o RMI Java que é utilizado para a comunicação entre agentes. A figura 6.5 é uma adaptação da arquitetura do ambiente do Eletrotutor III, apresentado em [BIC 99] e [SIL 2001].

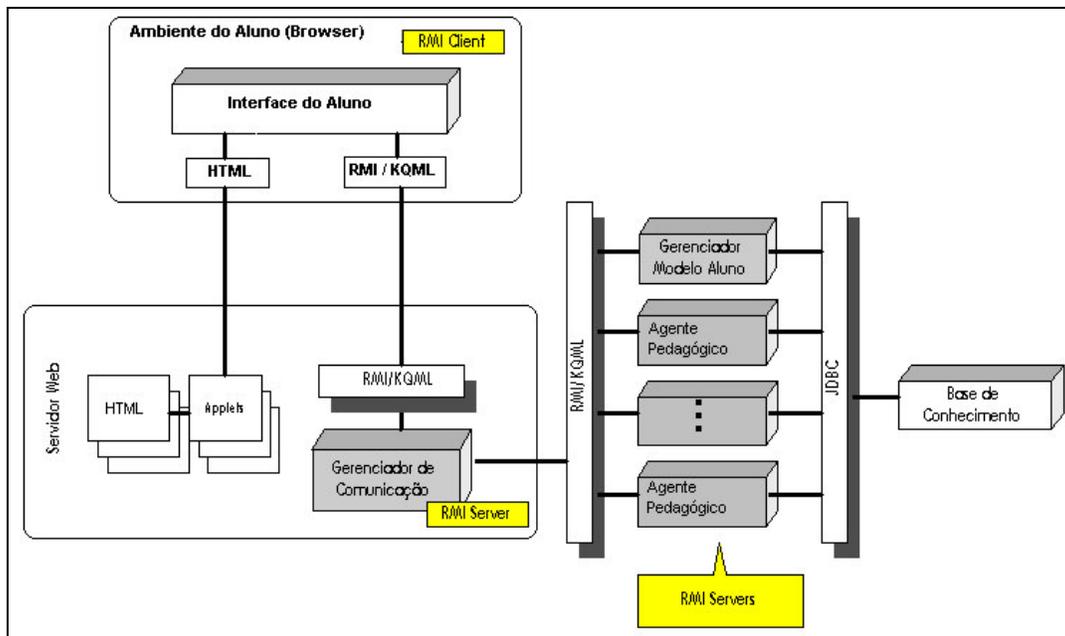


FIGURA 6.5 - Arquitetura do Ambiente

Na figura 6.5 pode-se observar que todas as comunicações feitas entre o agente Interface e os demais agentes do Eletrotutor são do modelo ponto-a-ponto, utilizam RMI Java e um protocolo baseado em KQML.

6.5 Estudos de casos

Para a verificação da identificação dos fatores motivacionais e afetivos, foram feitos alguns estudos de casos que representam a aplicação das regras em função do histórico. Cada estudo de caso refere-se aos eventos de uma lição e os valores dos fatores motivacionais e afetivos do caso anterior são assumidos como sendo os valores iniciais para cada caso. Excepcionalmente, o primeiro caso apresenta valores fixos inseridos pelo evento inicia sessão, quando se inicia o Tutor.

TABELA 6.1 - Inicia Tutor

Evento	Parâmetro	Regra	Valor dos Fatores
Inicia Sessão			Confiança = 5 Esforço = Médio Independência = 5

A tabela 6.1 representa o evento Inicia Sessão que insere valores iniciais para os indicadores dos fatores motivacionais e afetivos do aluno.

TABELA 6.2 - Caso 1

Evento	Parâmetro	Regra	Valor dos Fatores
Inicia Lição	5		
Encerra Lição	Desistiu	ESF1	Esforço = Nenhum

A tabela 6.2 representa a situação em que o aluno inicia uma lição e, em seguida, solicita outra ou encerra o tutor sem qualquer tentativa de solução.

TABELA 6.3 - Caso 2

Evento	Parâmetro	Regra	Valor dos Fatores
Inicia Lição	3		
Novo Passo			
Resposta	Acerta	CONF3	Confiança = 7
Encerra Lição	Realizou	ESF5 INDE4	Esforço = Médio Independência = 6

A tabela 6.3 representa uma situação desejável, isto é, o aluno acerta o exercício utilizando-se de um número de passos inferior ao número de passos médio, sugerido pelo especialista do domínio. A confiança e a independência do aluno aumenta e o esforço se mantém no nível médio.

TABELA 6.4 - Caso 3

Evento	Parâmetro	Regra	Valor dos Fatores
Inicia Lição	3		
Novo Passo			
Novo Passo			
Resposta	Erra	CONF4	Confiança = 6
Sugere Ajuda		INDE1	Independência = 5
Ajuda Fornecida	Genérica	INDE2	Independência = 4
Resposta	Acerta	CONF2	Confiança = 7
Encerra Lição	Realizou	ESF8	Esforço = Grande

A tabela 6.4 representa uma situação em que o aluno erra a questão proposta, imediatamente, o tutor intervém oferecendo ajuda, que é aceita pelo aluno. Esta é uma situação em que a redução do nível de independência é justificada pela recuperação do nível de confiança. Observa-se também que o esforço despendido pelo aluno foi maior, o que pode indicar uma falha na aprendizagem ou um maior grau de dificuldade do exercício.

TABELA 6.5 - Caso 4

Evento	Parâmetro	Regra	Valor dos Fatores
Inicia Lição	5		
Novo Passo			
Solicita Ajuda			
Ajuda Negada		INDE5	Independência = 5
Resposta	Acerta	CONF3	Confiança = 9
Encerra Lição	Realizou	ESF5	Esforço = Médio

O caso 4 apresentado na tabela 6.5, representa um aluno com baixo nível de independência. Esta situação é percebida quando o aluno possui um nível de independência baixo (4) e um nível de confiança alto (7). Se nesta situação o aluno solicitar ajuda, a reação do tutor será negá-la. A negativa é acompanhada de uma mensagem informando ao aluno que a ajuda estará disponível após sua tentativa de resposta.

TABELA 6.6 - Caso 5

Evento	Parâmetro	Regra	Valor dos Fatores
Inicia Lição	3		
Novo Passo			
Resposta	Erra	CONF4	Confiança = 8
Sugere Ajuda		INDE1	Independência = 4
Rejeita Ajuda		INDE6	Independência = 6
Novo Passo			
Novo Passo			
Resposta	Erra	CONF4	Confiança = 7
Encerra Lição	Desistiu	ESF7	Esforço = Grande

O caso 5, apresentado na tabela 6.6, representa o aluno com um nível de independência em elevação. No momento em que o aluno erra a questão, o tutor oferece ajuda. Se o aluno recusá-la é uma indicação de que ele quer resolver a questão sem o auxílio do tutor, representando uma atitude de independência por parte do aluno.

TABELA 6.7 - Caso 6

Evento	Parâmetro	Regra	Valor dos Fatores
Inicia Lição	3		
Novo Passo			
Sugere Ajuda		INDE1	Independência = 5
Ajuda Fornecida	Genérica	INDE2	Independência = 4
Resposta	Erra	CONF5	Confiança = 5
Encerra Lição	Desistiu	ESF6	Esforço = Médio

A tabela 6.7 representa o caso em que o aluno utiliza-se de muitos passos sem tentar uma resposta. Neste caso o tutor interfere, sugerindo ajuda, que é aceita pelo aluno. Na seqüência o aluno erra e desiste. Este caso deve ser interpretado pelo tutor como um exercício muito difícil.

TABELA 6.8 - Caso 7

Evento	Parâmetro	Regra	Valor dos Fatores
Inicia Lição	3		
Solicita Ajuda		CONF1	Confiança = 4
Ajuda Fornecida	Genérica	INDE2	Independência = 3
Encerra Lição	Desistiu	ESF2	Esforço = Mínimo

O caso 7, apresentado na tabela 6.8, representa um aluno que possui um nível de esforço mínimo, visto que ele desiste após o fornecimento da ajuda sem nem mesmo tentar resolver.

TABELA 6.9 - Caso 8

Evento	Parâmetro	Regra	Valor dos Fatores
Inicia Lição	4		
Novo Passo			
Solicita Ajuda			
Ajuda Fornecida	Genérica	INDE2	Independência = 2
Solicita Ajuda			
Ajuda Fornecida	Específica	INDE3	Independência = 0
Resposta	Acerta	CONF2	Confiança = 5
Encerra Lição	Realizou	ESF4	Esforço = Pequeno

O caso 8, apresentado na tabela 6.10, representa um aluno que além de possuir um nível de independência baixo (3), apresenta também um nível de confiança baixo (4). Este contexto faz com que o tutor não recuse a ajuda solicitada pelo aluno, mesmo estando com seu nível de independência muito baixo.

TABELA 6.10 - Caso 9

Evento	Parâmetro	Regra	Valor dos Fatores
Inicia Lição	3		
Novo Passo			
Resposta	Erra	CONF4	Confiança = 4
Encerra Lição	Desistiu	ESF3	Esforço = Pequeno

O caso 9, descrito na tabela 6.10, representa o caso do aluno que desiste em poucos passos sem ajuda. A aplicação das regras neste caso indica uma redução no nível de confiança e um pequeno esforço.

TABELA 6.11 - Caso 10

Evento	Parâmetro	Regra	Valor dos Fatores
Inicia Lição	3		
Novo Passo			
Ajuda Sugerida			
Ajuda Rejeitada		INDE6	Independência = 2
Resposta	Acerta	CONF3	Confiança = 6
Encerra Lição	Realizou	ESF9	Esforço = Máximo

O caso 10, apresentado na tabela 6.11, representa o caso de um aluno com esforço máximo e um crescente nível de independência. Estas características são identificadas quando o aluno utiliza-se de muitos passos, realiza a tarefa e não utiliza a ajuda.

Os dez casos acima representam uma pequena parte dos possíveis casos possíveis em uma interação real, mas representam a aplicação de todas as regras previstas.

6.6 A Interface

Para validar o agente Percepção, descrito no item 6.3 deste capítulo, foi feita uma implementação parcial do mesmo, no ambiente do Eletrotutor [BIC99] e [SIL2001]. Esta implementação contempla um mecanismo discreto de acompanhamento visual, gráfico, dos fatores emocionais e afetivos, uma tela com o detalhamento dos fatores emocionais e afetivos e uma terceira tela com os controles necessários à emulação dos eventos relevantes, necessários ao funcionamento do agente Percepção.

O mecanismo discreto de acompanhamento visual é um pequeno gráfico que se localiza dentro de um *frame*, inserido no Eletrotutor especialmente para manter o gráfico, no canto inferior direito da tela do navegador. O gráfico é composto de três barras horizontais nas cores: vermelha, amarela e azul. A barra de cor vermelha é precedida da letra “E” e representa o nível do fator “Esforço”, a barra de cor amarela é precedida da letra “C” e representa o nível do fator “Confiança” e a barra de cor azul é precedida da letra “I” e representa o nível do fator “Independência”. Quanto maior as barras no gráfico, melhor a situação do aluno com referência aos fatores motivacionais e afetivos que apresenta, ou seja, o tamanho da barra do gráfico é diretamente proporcional à qualidade do nível afetivo do aluno. Na tela de abertura, figura 6.6, pode-se verificar o gráfico em sua situação inicial, ou seja, imediatamente após o início da execução do Tutor.

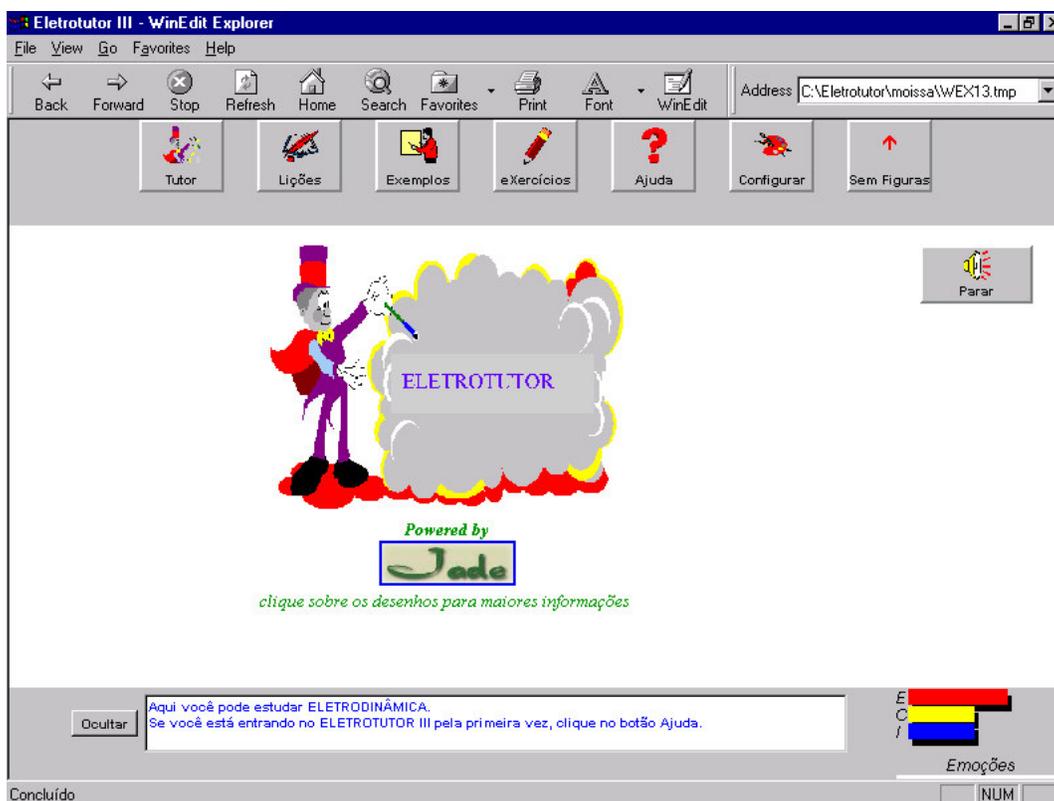


FIGURA 6.6 – Tela de Abertura

A segunda tela apresenta os fatores motivacionais e afetivos em representação numérica, uma lista dos eventos relevantes, na ordem em que ocorreram, e outros nove importantes atributos utilizados durante a interação. Sua exibição se dá através de um duplo clique sobre o gráfico de acompanhamento visual.

A tela está dividida em duas partes: na parte esquerda estão os rótulos e os valores dos principais atributos utilizados pelo agente e na parte direita encontra-se uma lista dos eventos relevantes gerados durante a sessão de uso conforme apresentado na figura 6.7.

O primeiro atributo apresentado representa o esforço atual do aluno que é quantificado por um número que pode variar entre 0 (zero) e 5 (cinco), sendo que o 0 indica nenhum esforço e 5 representa o esforço máximo. Para facilitar a implementação, o agente Percepção, trata os níveis de esforço, apresentados na tabela 5.2, através de números inteiros mantendo, porém, a relação de um número para cada nível de esforço.

O segundo atributo apresentado representa o nível de confiança do aluno e é quantificado por um número que pode variar entre 0 (zero) e 10 (dez), onde zero indica o mínimo de confiança e 10 representa o máximo de confiança.

O terceiro atributo apresentado representa o nível de independência do aluno e é quantificado por um número que pode variar entre 0 (zero) e 10 (dez), onde o valor 0 indica a menor independência e o valor 10 a maior independência.

O quarto atributo apresentado mostra um número que representa a quantidade de passos que o agente irá considerar como poucos passos. Neste protótipo este número é obtido pela diminuição de um passo do número passado como quantidade de passos médios, mas pode ser facilmente adaptado para receber este valor da base de dados alimentada pelo especialista do domínio.

O quinto atributo é um valor numérico recebido como parâmetro do evento inicia lição e indica o número de passos médios necessários para a resolução da lição. A definição deste número deve ser feita pelo especialista do domínio no momento em que a lição é cadastrada.

O sexto atributo é similar ao quarto atributo, todavia representa o que o agente irá considerar como muitos passos. A forma de obtenção deste número também é semelhante a do quarto atributo, porém, ao invés de diminuir um passo, soma-se um.

O sétimo atributo apresenta o número atual de passos dados pelo aluno e serve exclusivamente para monitoração das ações realizadas pelo agente Percepção.

O oitavo atributo apresenta o número de acertos na lição atual.

O nono atributo apresenta o número de acertos consecutivos na lição atual.

O décimo atributo representa a condição de acerto do último resultado apresentado pelo aluno.

O décimo primeiro atributo representa a condição de término da última lição iniciada.

O décimo segundo atributo representa o tipo de ajuda utilizada na lição atual.

No lado direito da Tela de Detalhamento das Emoções, apresentada na figura 6.7, encontra-se uma lista que armazena os eventos recebidos pelo agente e seus parâmetros.

São possíveis onze eventos, distribuídos em três grupos: Lição, Ajuda e Início/Término. Para o grupo Lição são possíveis quatro eventos, dos quais três exigem parâmetro. Para o grupo Ajuda são possíveis cinco eventos, dos quais dois são primários e três são secundários, isto é, são gerados como consequência de um evento primário, destes, somente um evento, do tipo secundário, exige parâmetro. Para o grupo Início/Término são possíveis dois eventos e nenhum deles exige parâmetro.

No grupo Lição o primeiro evento é denominado “Inicia Lição” e exige como parâmetro o número de passos médios previstos para a conclusão da mesma pelo aluno. O segundo evento é denominado “Novo Passo” e representa cada tarefa realizada ou resposta dada pelo aluno durante o estudo da lição; não exige parâmetro. O terceiro evento é denominado “Resposta” e exige como parâmetro um qualificador que indica se a resposta está correta ou não. O quarto parâmetro é denominado “Encerra Lição” e exige um parâmetro que indique se o encerramento se deu porque chegou ao fim ou se foi abortado por algum outro motivo.

No grupo Ajuda pode-se ter dois tipos de eventos, os primários e os secundários. Entre os primários estão a “Ajuda Sugerida” e a “Ajuda Solicitada” e entre os secundários estão a “Ajuda Recusada”, a “Ajuda Negada” e a “Ajuda Fornecida”, sendo que esta última exige um parâmetro que a identifique como genérica ou específica.

No grupo Início/Término são possíveis dois eventos: “Inicia Tutor” e “Encerra Tutor”, sendo que nenhum deles exige parâmetro.

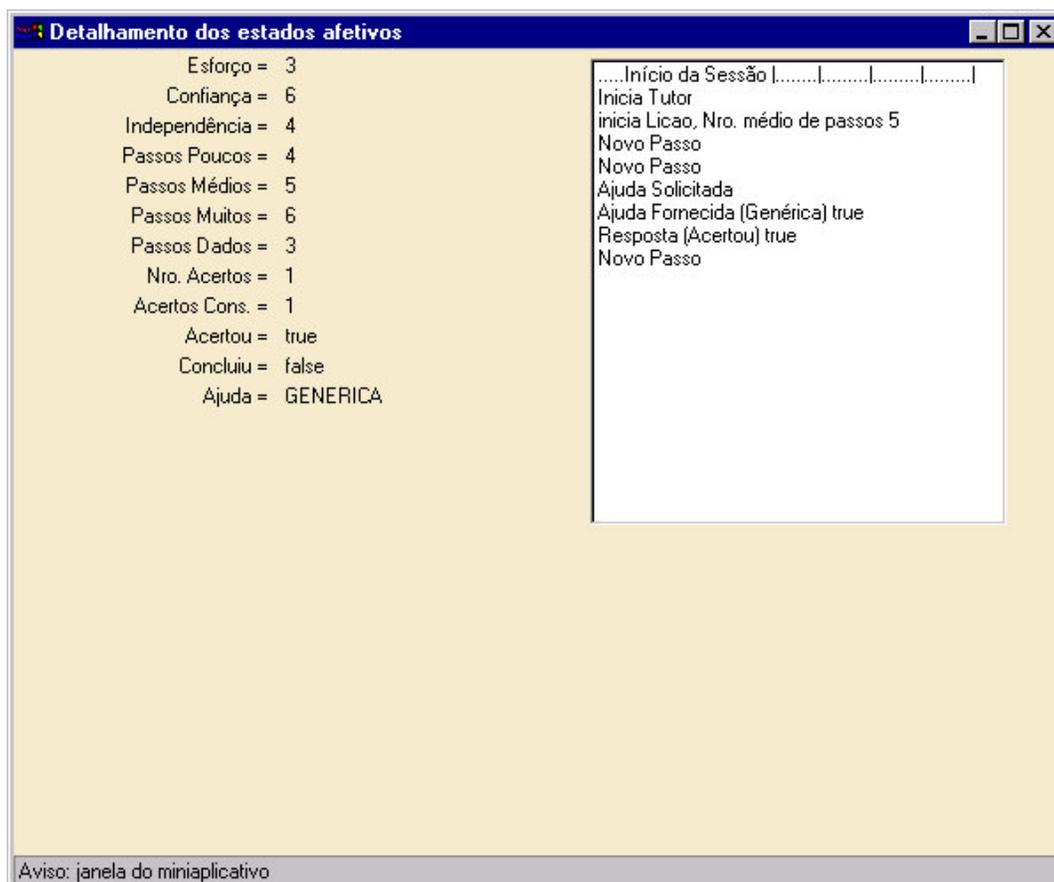


FIGURA 6.7 – Tela de Detalhamento das Emoções

A figura 6.8 apresenta a tela de simulação de eventos. Esta tela é composta de uma caixa de edição de texto e doze botões, sendo que os onze primeiros representam os eventos relevantes.

O funcionamento desta tela é simples e possibilita apenas dois tipos de operação: gerar evento sem parâmetro e gerar evento com parâmetro. Os eventos que operam sem parâmetro são: ajuda sugerida, inicia tutor, ajuda recusada, encerra tutor, ajuda negada, novo passo e ajuda solicitada. Os eventos que operam com parâmetro são: inicia lição, resposta, encerra lição e ajuda fornecida.

Para gerar um evento sem parâmetro, basta pressionar o botão, Para gerar um evento com parâmetro, deve-se digitar o valor do parâmetro na caixa de texto e pressionar o botão do evento correspondente.

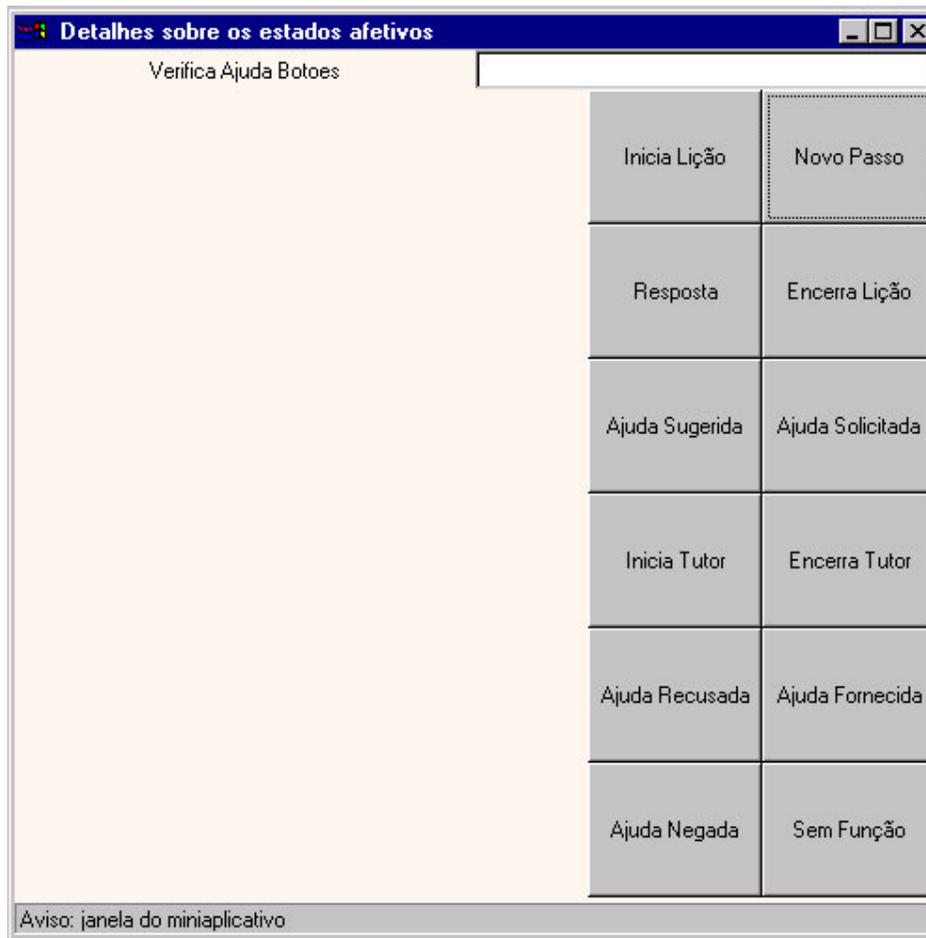


FIGURA 6.8 – Tela de Simulação de Eventos

O evento inicia lição requer como parâmetro um número inteiro que indica o número de passos médios previstos para a conclusão da lição pelo aluno.

O evento resposta aceita os números zero e um, onde zero representa uma resposta errada e um representa uma resposta certa.

O evento encerra lição aceita como parâmetro os números zero e um, onde o zero indica que a lição foi encerrada porque foi concluída e um indica que a lição foi abortada pelo usuário.

O evento ajuda fornecida aceita os números zero e um, onde o zero indica que a ajuda fornecida é do tipo específica e o um indica que a ajuda fornecida é do tipo genérica.

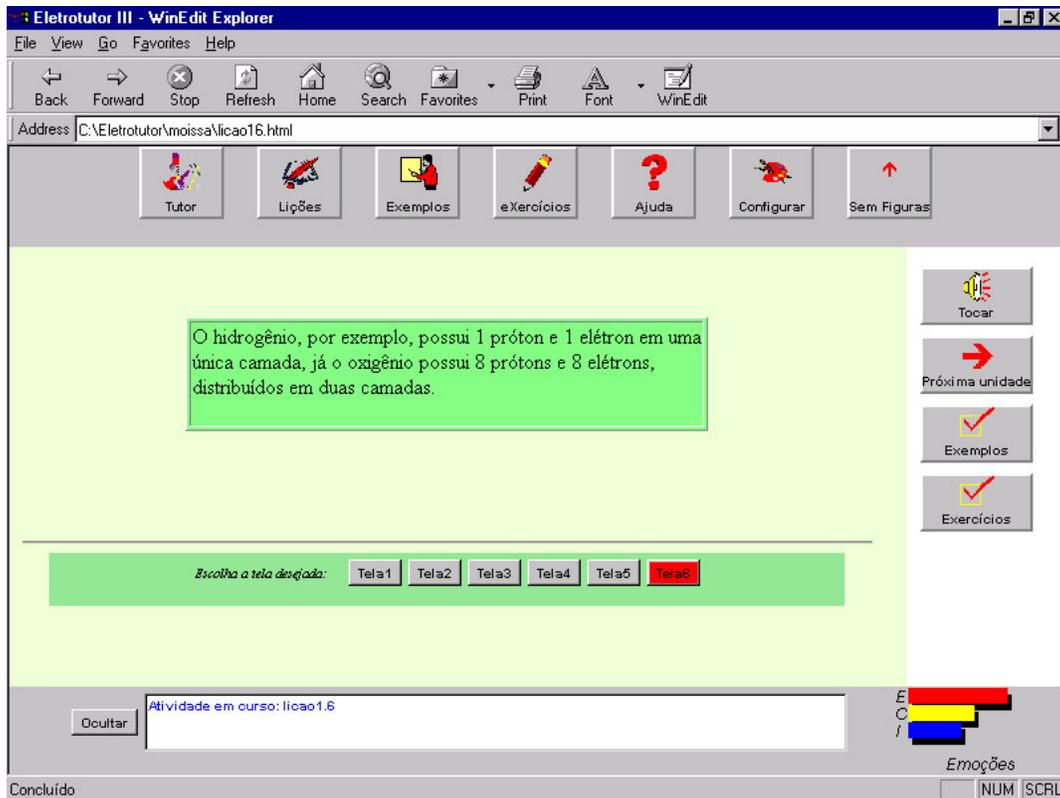


FIGURA 6.9 – Tela de Conteúdo

A figura 6.9, tela de conteúdo, apresenta a tela do tutor após uma seqüência de passos realizados pelo aluno durante uma sessão de uso.

7 Conclusões

A popularização da Internet, a evolução dos processadores e a necessidade de difundir o ensino em escala cada vez maior fizeram aumentar significativamente as pesquisas ligadas ao ensino à distância. Na área da Inteligência Artificial, mais especificamente na área de Sistemas Tutores Inteligentes, a evolução tem sido rápida e constante.

A estrutura funcional dos Sistemas Tutores Inteligentes tem se mantido relativamente estável, mas a arquitetura dos STI's que mantém esta estrutura tem evoluído em todos os aspectos. Os componentes funcionais se tornaram agentes de uma sociedade. O fato dos agentes serem independentes facilitou a evolução de cada módulo funcional, de forma independente, e permitiu a inserção de novos módulos que se mostrassem necessários.

O sistema de troca de mensagens entre os agentes passou a ser um item de fundamental importância. A troca de mensagens ponto a ponto apresenta mais segurança e em geral é a mais utilizada. A comunicação entre agentes freqüentemente tem se baseado na linguagem KQML.

A personalização do ensino, embora não tenha alcançado o nível desejado para os STIs, é uma característica de grande importância e está diretamente relacionada ao sistema de avaliação e as estratégias de ensino. O sistema de avaliação citado não se restringe à avaliação do conteúdo do domínio. Ele deve ser tão abrangente quanto possível, isto é, deve avaliar o conhecimento prévio do aluno, as preferências, a capacidade de compreensão, a adaptação às estratégias de ensino utilizadas, os fatores afetivos, enfim, tudo que possa indicar a estratégia de ensino mais adequada.

O estudo de outros sistemas que identificam sinais afetivos permitiu adaptar e definir um conjunto de regras, que forma a estratégia de identificação de fatores motivacionais e afetivos a partir da interface, bem como, propor a inserção do agente Percepção em uma arquitetura distribuída e descrever os requisitos necessários à identificação destes fatores.

7.1 Limitações

Algumas das limitações observadas durante o desenvolvimento deste trabalho foram:

- Alguns requisitos obrigatórios para as regras de identificação dos fatores motivacionais e afetivos não estarem implementadas nos STI disponíveis. A implementação deste agente requer um sistema de ajuda que diferencie ajuda genérica de ajuda específica. O modelo de aluno deve estar preparado para responder às mudanças de estado motivacional do aluno a fim de manter o máximo de motivação e aprendizado.

- Os STIs que tiveram seus códigos estudados, com a finalidade de se implementar o agente proposto neste trabalho, não possuem estratégias de ensino voltadas para diferentes estados motivacionais.
- Os sistemas de ajuda existentes são muito limitados, tanto em recurso como em classificação do tipo de ajuda.
- A proposta de utilizar-se de um simples *browser* para visualização de páginas na WEB, o que restringiu o uso de qualquer tipo de sensor específico para identificação de sinais afetivos.

7.2 Trabalhos Futuros

A característica multidisciplinar deste trabalho amplia as possibilidades de prosseguimento dos estudos aqui iniciados, proporcionando várias propostas, tais como:

- Implementação de um modelo de aluno que utilize, os fatores motivacionais e afetivos identificados, em seu sistema de avaliação e aplique uma estratégia de ensino personalizada, com o objetivo de estimular a manutenção dos fatores mais favoráveis à aprendizagem.
- Elaboração de um questionário inicial para traçar o perfil do aluno e a sua respectiva consolidação ao sistema de avaliação.
- Estabelecer novas estratégias de identificação e ampliar o número de fatores motivacionais afetivos detectados pelo agente Percepção.
- Projetar uma interface dinâmica que seja sensível aos fatores motivacionais e afetivos.
- Elaborar uma ajuda, que tenha pelo menos dois níveis de abrangência, que seja mais completa e mais adequada às estratégias de ensino motivacionais.
- Agregar outros recursos para identificação de estados emotivos como, por exemplo, interpretação de expressões faciais, proposto por [KEA 93].

7.3 Contribuições

A conclusão deste trabalho proporciona as seguintes contribuições:

- A definição de um conjunto de regras para identificação de fatores motivacionais e afetivos a partir da monitoração da interface de um STI que contemple os requisitos mínimos.
- A definição dos requisitos mínimos necessários para aplicação da estratégia de identificação proposta.
- Proposta de uma arquitetura de agente, compatível com a arquitetura do sistema Eletrotutor.
- Implementação de um protótipo que servirá como ferramenta para validação do trabalho de avaliação pedagógica e estratégias de ensino que vem sendo desenvolvido por Bercht [BER 97].

Anexo 1 Fonte do Objeto Emoção

```

/**
 *
 * Objeto Emoção
 * Versao 1.1 01/11/2000
 * Revisado em 01/05/2001
 * @author: Harry Moissa
 *
 * O objeto Emoção é exibido em um frame
 * a partir de um clique sobre o gráfico exibido
 * na interface do Tutor.
 */

import java.lang.*;
import java.awt.*;
import java.awt.event.*;
import java.applet.Applet;
import java.awt.Dimension;
import javax.swing.JOptionPane;

public class Emocao extends Frame implements ActionListener, WindowListener
{
    private int    esforco=0;
    private int    confianca=0;
    private int    independencia=0;
    private int    passosPoucos=0;
    private int    passosMedios=0;
    private int    passosMuitos=0;
    private int    passosDados=0;
    private int    nroAcertos=0;
    private int    nroAcertosConsecutivos = 0;
    private boolean acertou = false;
    private boolean concluiu = false;
    private String ajuda = "";
    private Image  graf;
    private Graphics grafico;
    Label          lbesforco,
                  lbconfianca,
                  lbindependencia,
                  lbpassosPoucos,
                  lbpassosMedios,
                  lbpassosMuitos,
                  lbpassosDados,
                  lbnroAcertos,
                  lbnroAcertosConsecutivos,

```

```

        lbacertou,
        lbconcluiu,
        lbajuda,
        lbvesforco,
        lbvconfianca,
        lbvindependencia,
        lbvpassosPoucos,
        lbvpassosMedios,
        lbvpassosMuitos,
        lbvpassosDados,
        lbvnroAcertos,
        lbvnroAcertosConsecutivos,
        lbvacertou,
        lbvconcluiu,
        lbvajuda;

Panel    painelsup,
        painelinf,
        paineldir,
        painelesq,
        painelsupesq,
        painelsupdir;

List     lista;

public Emocao(String s)
{
    this.setTitle(s);
    painelsup        = new Panel();
    painelinf        = new Panel();
    paineldir        = new Panel();
    painelesq        = new Panel();
    painelsupesq     = new Panel();
    painelsupdir     = new Panel();
    lbvesforco       = new Label( "Esforço = ");
    lbconfianca      = new Label( "Confiança = ");
    lbdependencia    = new Label( "Independência = ");
    lbpassosPoucos   = new Label( "Passos Poucos = ");
    lbpassosMedios   = new Label( "Passos Médios = ");
    lbpassosMuitos   = new Label( "Passos Muitos = ");
    lbpassosDados    = new Label( "Passos Dados = ");
    lbvroAcertos     = new Label( "Nro. Acertos = ");
    lbvroAcertosConsecutivos = new Label( "Acertos Cons. = ");
    lbacertou        = new Label( "Acertou = ");
    lbconcluiu       = new Label( "Concluiu = ");
    lbajuda          = new Label( "Ajuda = ");
    lbvesforco       = new Label(esforco + " ");
    lbvconfianca     = new Label(confianca + " ");
    lbvindependencia = new Label(independencia + " ");
    lbvpassosPoucos  = new Label(passosPoucos + " ");

```

```

lbvpassosMedios      = new Label(passosMedios + " ");
lbvpassosMuitos     = new Label(passosMuitos + " ");
lbvpassosDados      = new Label(passosDados + " ");
lbvnroAcertos       = new Label(nroAcertos + " ");
lbvnroAcertosConsecutivos = new Label(nroAcertosConsecutivos + " ");
lbvacertou          = new Label(acertou + " ");
lbvconcluiu         = new Label(concluiu + " ");
lbvajuda            = new Label(ajuda + " ");

```

```

lbesforco.setAlignment( Label.RIGHT );
lbconfianca.setAlignment( Label.RIGHT );
lbindependencia.setAlignment( Label.RIGHT );
lbpassosPoucos.setAlignment( Label.RIGHT );
lbpassosMedios.setAlignment( Label.RIGHT );
lbpassosMuitos.setAlignment( Label.RIGHT );
lbpassosDados.setAlignment( Label.RIGHT );
lbnroAcertos.setAlignment( Label.RIGHT );
lbnroAcertosConsecutivos.setAlignment( Label.RIGHT );
lbacertou.setAlignment( Label.RIGHT );
lbconcluiu.setAlignment( Label.RIGHT );
lbajuda.setAlignment( Label.RIGHT );
lbvesforco.setAlignment( Label.LEFT );
lbvconfianca.setAlignment( Label.LEFT );
lbvindependencia.setAlignment( Label.LEFT );
lbvpassosMedios.setAlignment( Label.LEFT );
lbvpassosDados.setAlignment( Label.LEFT );
lbvnroAcertos.setAlignment( Label.LEFT );
lbvnroAcertosConsecutivos.setAlignment( Label.LEFT );
lbvacertou.setAlignment( Label.LEFT );
lbvconcluiu.setAlignment( Label.LEFT );
lbvajuda.setAlignment( Label.LEFT );

```

```

setLayout(new GridLayout(1,2));
add(painelesq);
add(paineldir);

```

```

painelesq.setLayout(new GridLayout(2,1));
painelesq.add(painelsup);
painelesq.add(painelinf);
painelsup.setLayout(new GridLayout(1,2));
painelsup.add(painelsupesq);
painelsup.add(painelsupdir);

```

```

painelsupesq.setLayout(new GridLayout(13,1));
painelsupesq.add(lbesforco);
painelsupesq.add(lbconfianca);
painelsupesq.add(lbindependencia);
painelsupesq.add(lbpassosPoucos);

```

```

painelsupesq.add(lbpassosMedios);
painelsupesq.add(lbpassosMuitos);
painelsupesq.add(lbpassosDados);
painelsupesq.add(lbnroAcertos);
painelsupesq.add(lbnroAcertosConsecutivos);
painelsupesq.add(lbacertou);
painelsupesq.add(lbconcluiu);
painelsupesq.add(lbajuda);

```

```

painelsupdir.setLayout(new GridLayout(13,1));
painelsupdir.add(lbvesforco);
painelsupdir.add(lbvconfianca);
painelsupdir.add(lbvindependencia);
painelsupdir.add(lbvpassosPoucos);
painelsupdir.add(lbvpassosMedios);
painelsupdir.add(lbvpassosMuitos);
painelsupdir.add(lbvpassosDados);
painelsupdir.add(lbvnrAcertos);
painelsupdir.add(lbvnrAcertosConsecutivos);
painelsupdir.add(lbvacertou);
painelsupdir.add(lbvconcluiu);
painelsupdir.add(lbvajuda);

```

```

lista = new List(20, false);
paineldir.add(lista);
lista.addItem("Início da Sessão1.....10.....20.....30.....40");
//graf      = createImage(150, 150);
//grafico   = graf.getGraphics();
//painelsupesq.add(graf);
resize(600,500);
addWindowListener(this);
setVisible(true);
}

```

```

public void reexibe()
{
lbvesforco.setText(" "+esforco);
lbvconfianca.setText(" "+confianca);
lbvindependencia.setText(" "+independencia);
lbvpassosPoucos.setText(" "+passosPoucos);
lbvpassosMedios.setText(" "+passosMedios);
lbvpassosMuitos.setText(" "+passosMuitos);
lbvpassosDados.setText(" "+passosDados);
lbvnroAcertos.setText(" "+nroAcertos);
lbvnroAcertosConsecutivos.setText(" "+nroAcertosConsecutivos);
lbvacertou.setText(" "+acertou);
lbvconcluiu.setText(" "+concluiu);
lbvajuda.setText(" "+ajuda);
}

```

```
public void AtualizaEstado()
{
// Aqui vai o código que envia a emoção para o Tutor
}

public int getConfianca()
{
return confianca;
}

public int getEsforco()
{
return esforco;
}

public int getIndependencia()
{
return independencia;
}

public void setEsforco(int c)
{
if (c >= 0 && c <= 5)
    esforco = c;
AtualizaEstado();
}

public void setConfianca(int c)
{
if (c < 0)    {confianca = 0;}
else if (c > 10) {confianca = 10;}
else        {confianca = c;}
AtualizaEstado();
}

public void setIndependencia(int c)
{
if (c < 0)    {independencia = 0;}
else if (c > 10) {independencia = 10;}
else        {independencia = c;}
AtualizaEstado();
}

public void iniciaLicao(int p)
{
passosPoucos  = p - 1;
passosMedios  = p;
passosMuitos  = p + 1;
passosDados   = 0;
}
```

```

nroAcertos = 0;
nroAcertosConsecutivos = 0;
acertou = false;
ajuda = "";
if (!concluiu) encerraLicao(true); // desiste da lição anterior
concluiu = false;
lista.addItem("inicia Licao, Nro. médio de passos "+ p);
}

```

```

public void encerraLicao(boolean desistiu)
{
lista.addItem("Encerra Licao (Desistiu) "+ desistiu);
if (desistiu)
{
if (passosDados <= passosPoucos)
if (ajuda != "") setEsforco(1);
else setEsforco(2);
if (passosDados >= passosMuitos)
if (ajuda != "") setEsforco(3);
else setEsforco(4);
}
else
{
if (passosDados <= passosPoucos)
if (ajuda != "") setEsforco(2);
else setEsforco(3);
if (passosDados >= passosMuitos)
if (ajuda != "") setEsforco(4);
else setEsforco(5);
}
if (passosDados == 0) setEsforco(0);
concluiu = true;
reexibe();
}

```

```

public void novoPasso()
{
lista.addItem("Novo Passo ");
passosDados ++;
reexibe();
}

```

```

public void evento(String s)
{
reexibe();
}

```

```

public void resposta(boolean r)
{
lista.addItem("Resposta (Acertou) "+ r);
}

```

```

novoPasso();
if (r)
{
    acertou = true;
    if (ajuda != "") {
        setConfianca(getConfianca() + 1);
    }
    if (ajuda == "")
    {
        setConfianca(getConfianca() + 2);
    }
    nroAcertos ++;
    nroAcertosConsecutivos ++;
}
else
{
    acertou = false;
    if (ajuda != ""){
        setConfianca(getConfianca() -2);
    }
    if (ajuda == ""){
        setConfianca(getConfianca() -1);
    }
    nroAcertosConsecutivos = 0;
}
}

public void ajudaSugerida()
{
    lista.addItem("Ajuda Sugerida ");
    setIndependencia(getIndependencia() -1);
}

public void ajudaSolicitada()
{
    lista.addItem("Ajuda Solicitada");
    if (passosDados == 0){
        setConfianca(getConfianca() -1);
    }
}

public void iniciaTutor()
{
    lista.addItem("Inicia Tutor");
    setConfianca(5);
    setEsforco(3);
    setIndependencia(5);
    concluiu = true;
}

```

```

public void ajudaRecusada()
{
    lista.addItem("Ajuda Recusada ");
    setIndependencia(getIndependencia() + 2);
}

public void ajudaFornecida(boolean generica)
{
    lista.addItem("Ajuda Fornecida (Genérica) "+ generica);
    if (generica)
    {
        setIndependencia(getIndependencia() -1);
        ajuda = "GENERICA";
    }
    else
    {
        setIndependencia(getIndependencia() - 2);
        ajuda = "ESPECIFICA";
    }
}

public void ajudaNegada()
{
    lista.addItem("Ajuda Negada ");
    setIndependencia(getIndependencia() +1);
}

public void encerraTutor()
{
    lista.addItem("Encerra Tutor");
    encerraLicao(concluiu);    //concluiu contém estado da última lição
}

```

```

// OS METODOS A SEGUIR SAO DE DECLARAÇÃO OBRIGATORIA, DEVIDO
A ESPECIFICACAO DA
// INTERFACE WindowListener

```

```

public void itemStateChanged(ItemEvent evt){ }
public void actionPerformed(ActionEvent evt) { }
public void windowClosing(WindowEvent evt) {setVisible(false);}
public void windowOpened(WindowEvent evt){}
public void windowIconified(WindowEvent evt) {}
public void windowDeiconified(WindowEvent evt) {}
public void windowClosed(WindowEvent evt) {}
public void windowActivated(WindowEvent evt) {}
public void windowDeactivated(WindowEvent evt) {}

}

```

Anexo 2 Fonte do Objeto Painel Gráfico (Simulador de Eventos)

```

/**
 *
 * PainelGraf
 * Versao 1.2
 * 01/11/2000
 * @author: Harry Moissa
 *
 * É acionada através do botão Chartn na classe PainelInferior
 */

import java.lang.*;
import java.awt.*;
import java.awt.event.*;
import java.applet.Applet;
import java.awt.Dimension;

public class PainelGraf extends Frame implements ActionListener, WindowListener
{
    Panel    painelTitulo, painelBT, painelLista;
    Button    BT01, BT02, BT03, BT04, BT05, BT06, BT07, BT08, BT09, BT010,
    BT011, BT012;
    Label    labelTitulo;
    Emocao   estado;
    TextArea displayArea;
    TextField R;
    int      v;

    public PainelGraf(String s )
    {
        R = new TextField(5);
        R.setEditable(true);
        estado= new Emocao("Detalhamento dos estados afetivos");
        setLayout(new BorderLayout());
        addWindowListener(this);
        this.setTitle(s );
        this.setBackground(new Color(255, 245, 235));
        painelTitulo = new Panel();
        painelTitulo.setLayout ( new GridLayout(1,2) );
        labelTitulo = new Label( "Verifica Ajuda Botoes" );
        labelTitulo.setAlignment( Label.CENTER );
        painelTitulo.add( labelTitulo );
        painelTitulo.add(R);
        painelLista = new Panel();
        painelLista.setLayout( new GridLayout() );
    }

```

```
panelBT = new Panel();  
panelBT.setLayout( new GridLayout(6,2) );
```

```
BT01 = new Button( "Inicia Lição" );  
panelBT.add( BT01 );  
BT01.addActionListener(this);
```

```
BT02 = new Button( "Novo Passo" );  
panelBT.add( BT02 );  
BT02.addActionListener(this);
```

```
BT03 = new Button( "Resposta" );  
panelBT.add( BT03 );  
BT03.addActionListener(this);
```

```
BT04 = new Button( "Encerra Lição" );  
panelBT.add( BT04 );  
BT04.addActionListener(this);
```

```
BT05 = new Button( "Ajuda Sugerida" );  
panelBT.add( BT05 );  
BT05.addActionListener(this);
```

```
BT06 = new Button("Ajuda Solicitada");  
panelBT.add(BT06);  
BT06.addActionListener(this);
```

```
BT07 = new Button("Inicia Tutor");  
panelBT.add(BT07);  
BT07.addActionListener(this);
```

```
BT08 = new Button("Encerra Tutor");  
panelBT.add(BT08);  
BT08.addActionListener(this);
```

```
BT09 = new Button("Ajuda Recusada");  
panelBT.add(BT09);  
BT09.addActionListener(this);
```

```
BT010 = new Button( "Ajuda Fornecida" );  
panelBT.add( BT010 );  
BT010.addActionListener(this);
```

```
BT011 = new Button( "Ajuda Negada" );  
panelBT.add( BT011 );  
BT011.addActionListener(this);
```

```
BT012 = new Button( "Sem Função" );  
panelBT.add( BT012 );  
BT012.addActionListener(this);
```

```

addWindowListener(this);
add ( "North", panelTitulo);
add ( "East", panelBT);
add ( "West", panelLista);
setBounds(1,100,500,500);
setVisible(true);
} // fim do metodo construtor

/**
 * Método que trata os eventos dos botoes
 * @param evt
 */

public void actionPerformed(ActionEvent evt)
{
    if(evt.getActionCommand().equals("Inicia Lição"))
    {
        String entrada = R.getText();
        R.setText("");
        try
            {v = Integer.parseInt(entrada);}
        catch (NumberFormatException e)
            {v = 0;}
        estado.iniciaLicao(v);
        estado.evento("Inicia Lição");
    }
    if(evt.getActionCommand().equals("Novo Passo"))
    {
        estado.evento("Novo Passo");
        estado.novoPasso();
    }
    if(evt.getActionCommand().equals("Resposta"))
    {
        String entrada = R.getText();
        R.setText("");
        try
            {v = Integer.parseInt(entrada);}
        catch (NumberFormatException e)
            {v = 0;}
        if (v > 0) {estado.resposta(true);}
        else      {estado.resposta(false);}
        estado.evento("Resposta!");
    }
    if(evt.getActionCommand().equals("Encerra Lição"))
    {
        String entrada = R.getText();
        R.setText("");
        try
            {v = Integer.parseInt(entrada);}

```

```

    catch (NumberFormatException e)
        {v = 0;}
    if (v > 0) {estado.encerraLicao(true);}
    else      {estado.encerraLicao(false);}
    estado.evento("Encerra Lição");
}
if(evt.getActionCommand().equals("Ajuda Sugerida"))
{
    estado.ajudaSugerida();
    estado.evento("Ajuda Sugerida");
}
if(evt.getActionCommand().equals("Ajuda Solicitada"))
{
    estado.ajudaSolicitada();
    estado.evento("Ajuda Solicitada");
}
if(evt.getActionCommand().equals("Inicia Tutor"))
{
    estado.iniciaTutor();
    estado.evento("Inicia Tutor!");
}
if(evt.getActionCommand().equals("Encerra Tutor"))
{
    estado.encerraTutor();
    estado.evento("Encerra Tutor");
}
if(evt.getActionCommand().equals("Ajuda Recusada"))
{
    estado.ajudaRecusada();
    estado.evento("Ajuda Recusada");
}
if(evt.getActionCommand().equals("Ajuda Fornecida"))
{
    String entrada = R.getText();
    R.setText("");
    try
        {v = Integer.parseInt(entrada);}
    catch (NumberFormatException e)
        {v = 0;}
    if (v > 0) {estado.ajudaFornecida(true);}
    else      {estado.ajudaFornecida(false);}
    estado.evento("Ajuda Fornecida");
}
if(evt.getActionCommand().equals("Ajuda Negada"))
{
    estado.ajudaNegada();
    estado.evento("Ajuda Negada");
}
}

```

```

public void windowClosing(WindowEvent evt)
{
    setVisible(false);
}

// OS METODOS A SEGUIR SAO DE DECLARARAÇÃO OBRIGATORIA,
DEVIDO A ESPECIFICACAO DA
// INTERFACE WindowListener

/**
 * Este metodo é de declaracao obrigatoria(nao faz nada) devido a especific.da
Interface WindowListener
 * @param evt
 */
public void windowOpened(WindowEvent evt)
{
}

/**
 * Este metodo é de declaracao obrigatoria(nao faz nada) devido a especific.da
Interface WindowListener
 * @param evt
 */
public void windowIconified(WindowEvent evt)
{
}

/**
 * Este metodo é de declaracao obrigatoria(nao faz nada) devido a especific.da
Interface WindowListener
 * @param evt
 */
public void windowDeiconified(WindowEvent evt)
{
}

/**
 * Este metodo é de declaracao obrigatoria(nao faz nada) devido a especific.da
Interface WindowListener
 * @param evt
 */
public void windowClosed(WindowEvent evt)
{
}

/**
 * Este metodo é de declaracao obrigatoria(nao faz nada) devido a especific.da
Interface WindowListener
 * @param evt
 */

```

```
public void windowActivated(WindowEvent evt)
{
}

/**
 * Este metodo é de declaracao obrigatoria(nao faz nada) devido a especific.da
Interface WindowListener
 * @param evt
 */
public void windowDeactivated(WindowEvent evt)
{
}
}
```

Bibliografia

- [AFF 2000] AFFECTIVE Computing: Research on Human Emotion. [S.l.:s.n.], 2000. Disponível em: <http://www.media.mit.edu/affect/AC_research/emotions.html> Acesso em: 22 ago. 2000.
- [AFF 2000a] AFFECTIVE Computing: Affect Pattern Recognition. [S.l.:s.n.], 2000. Disponível em: <http://www.media.mit.edu/affect/AC_research/recognizing.html> Acesso em: 22 ago. 2000.
- [AFF 2000b] AFFECTIVE Computing: Sensing Human Affect. [S.l.:s.n.], 2000. Disponível em: <http://www.media.mit.edu/affect/AC_research/sensing.html> Acesso em: 22 ago. 2000.
- [BER 97] BERCHT, M. Avaliação Pedagógica como fator para a Construção de Estratégias de ensino e Aprendizagem Computadorizados. Porto Alegre: CPGCC/UFRGS, 1997.101p. Exame de Qualificação. (EQ-14).
- [BER 99] BERCHT, M.; CECCIM, R. B.; VICCARI, R. M. **Afetividade em Máquina:** uma possibilidade? Non-sense completo? Pós-modernidade na Inteligência Artificial? Porto Alegre: UFRGS, 1999. Draft.
- [BER 99a] BERCHT, M; MOISSA, H. E.; VICCARI R. M. Identificação de fatores motivacionais e afetivos em um ambiente de ensino e aprendizagem. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, SBIE, 10., 1999. **Anais...** Curitiba: Universidade Federal do Paraná, 1999.
- [BER 2001] BERCHT, Magda. **Uma Arquitetura para Fatores Emocionais e Motivacionais em Agentes Pedagógicos.** Porto Alegre: CPGC/UFRGS, 2001. 54p. Proposta de Tese (Doutorado).
- [BIC 98] BICA, F.; SILVEIRA, R. A.; VICCARI, R. M. Eletrotutor III: Uma abordagem Multiagentes. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, SBIE, 9., 1998. **Anais...** Fortaleza: Universidade Federal do Ceará, 1998.
- [BIC 99] BICA, F. **Eletrotutor III – Uma Abordagem Multiagente para o ensino à Distância.** Porto Alegre: CPGCC/UFRGS, 1999. 78p. Dissertação de Mestrado.
- [BIG 97] BIGUS, J. P.; BIGUS J. **Constructing Intelligent Agents with java:** a programmer's guide to smarter applications. New York: John Wiley & Sons, 1997. 379p.

- [CAZ 96] CAZELLA, S.C. Um Estudo Sobre Decomposição de Problemas em Inteligência Artificial Distribuída. Porto Alegre: CPGCC/UFGRS, 1996. 48p. (TI-509).
- [DAV 96] DAVIDSSON, P. **Autonomous Agents and the Concept of Concepts**. Lund: Department of Computer Science Lund University, 1996. 221p. Doctoral Dissertation.
- [DAM 95] DAMICO, C. et. al. **Inteligência Artificial: uma abordagem de agentes**. Porto Alegre: CPGCC/UFRGS, 1995. 87 p. (RP-257).
- [FRA 96] FRANKLIN, S; GRAESSER, A. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In: INTERNATIONAL WORKSHOP ON AGENT THEORIES, ARCHITECTURES, AND LANGUAGES, 3., 1996. **Proceedings...** Berlin: Springer-Verlag, 1996.
- [FRC 2000] FRANKEL C. B.; RAY, R. D. **Emotion, Intention and the Control Architecture of Adaptively Competent Information Processing**. San Francisco, 2000. Disponível em: <<http://www.cs.bham.ac.uk/~axs/aisb2000/abstracts/papers/frankelray.html>>. Acesso em: 19 ago.2000.
- [GAR 79] Garret, Henry Edward. **Grandes experimentos da Psicologia**. Tradução e notas de Maria da Penha Pompeu de Toledo. 4. ed. São Paulo: Ed. Nacional, 1979. 201-231.
- [GEN 94] GENESERETH, M. R.; KETCHPEL, S. P. Software Agents. **Communications of the ACM**, New York, v.37, n.7, p.48-53, July 1994.
- [GIR 98] GIRAFFA, L. M. M.; VICCARI, R. M. Estratégias de Ensino em Sistemas Tutores Inteligentes modelados através de tecnologia de agentes. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, SBIE, 9., 1998. **Anais...** Fortaleza: Universidade Federal do Ceará, 1998.
- [GOL 95] GOLEMAN, Daniel. **Inteligência Emocional: A Teoria Revolucionária que Redefine o que é Ser Inteligente**. Rio de Janeiro: Objetiva, 1995. 370 p.
- [HAY 95] HAYES-ROTH, B. An Architecture for Adaptive Intelligent Systems. **Artificial Intelligence: Special Issue on Agents and Interactivity**, [S.l.], v.72, p.329-365, 1995.
- [JAQ 98] JAQUES, P. A. **Agentes de Software para Monitoração da Colaboração em Ambientes Telemáticos de Ensino**. Porto Alegre: Pontifícia Universidade Católica do Rio Grande do Sul, 1998. 90p. Dissertação de Mestrado.

- [JAQ 98a] JAQUES, P. A.; OLIVEIRA, F. M. Agentes de Software para Análise das Interações em um Ambiente de Ensino à Distância. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, SBIE, 9., 1998. **Anais...** Fortaleza: Universidade Federal do Ceará, 1998.
- [KEA 93] KEARNEY, G. D.; MCKENZIE, S. Machine Interpretation of Emotion: Design of a Memory-Based Expert System for Interpreting Facial Expressions in Terms of Signaled Emotions. **Cognitive Science**, [S.l.], v.17, n.4, p.589-622, 1993.
- [LIE 9?] LIEBERMAN, H. **Attaching Interface Agent Software to Applications**. Cambridge: Media Laboratory Massachusetts Institute of Technology, 199?.
- [MAT 96] MATSUBARA, Y.; NAGAMACHI, M. Motivation System and Human Model for Intelligent Tutoring. In: INTERNATIONAL CONFERENCE INTELLIGENT TUTORING SYSTEMS, 3., 1996, Montreal. **Proceedings...**Berlin: Springer-Verlag, 1996. p.139-147.
- [MEN 98] MENEZES, R. A.; FUKS, H.; GARCIA, A. C. B. Utilizando Agentes no Suporte à Avaliação Informal no Ambiente de Instrução Baseada na Web - AulaNet. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, SBIE, 9., 1998. **Anais...** Fortaleza: Universidade Federal do Ceará, 1998.
- [MOU 95] MOUSSALE, N. M.; VICCARI, R. M. Observação dos estados mentais de agentes durante um diálogo. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, SBIE, 6., 1995. **Anais...** Florianópolis: SBC: UFSC, 1995.
- [MOU 96] MOUSSALE, N. M. **Interações Tutor-Aluno Analisadas através de Seus Estados Mentais**. Porto Alegre: CPGCC/UFRGS, 1996. 231 p. Dissertação de Mestrado.
- [NUN 98] NUNES, M. A. S. N. **Modelagem de um Agente Cognitivo em um Ambiente de Simulação utilizando uma arquitetura híbrida de Sistema Multiagente**. Porto Alegre: CPGCC/UFRGS, 1996. 168 p. Dissertação de Mestrado.
- [PAS 99] PASSOLD, F. R. **Agentes Inteligentes**. Joinville: UDESC, 1999. Trabalho de Diplomação.
- [PIC 97] PICARD, R. **Affective Computing**. Cambridge: MIT Press... 1997. 292p.
- [RIC 93] RICH, E.; KNIGHT, K. **Inteligência Artificial**. 2.ed. São Paulo : Makron Books, 1993. 492-512.
- [RUS 95] RUSSELL, Stuart J.; NORVIG, Peter. **Artificial Intelligence: A Modern Approach**. Englewood Cliffs, NJ: Prentice Hall, 1995.

- [SIC 95] SICHMAN, J.; DEMAZEU, Y; BOISSIER, O. How can knowledge based systems be called agents? In: SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICIAL, 9., 1995. **Anais...**[S.l.:s.n.], 1995.
- [SIL 2001] SILVEIRA, Ricardo Azambuja. **Modelagem Orientada a Agentes Aplicada a Ambientes Inteligentes Distribuídos de Ensino: JADE Java Agent framework for Distance learning Environments.** Porto Alegre: PPGC/UFRGS, 2001. 126 p. Tese de Doutorado.
- [SIL 9?] SILVEIRA, R. A.; VICCARI, R. M. Distance learning: from the single intelligent tutor paradigm to a distributed multiagent tutor arquitetura. [Sl.: s.n.], 199?. Draft.
- [SOL 95] SOLDATO, T.; BOULAY, B. Implementation of Motivational Tactics in Tutoring Systems. **Journal of Artificial Intelligence in Education**, VA, v.6, n.4, p.337-378, 1995.
- [STO 98] STORB, B. H. et al. Usando Conceito de Complexidade para Classificação de Agentes. Florianópolis: UFSC, 1998.
- [TAR 99] TAROUCO, L. M. R.; HACK L. E. A Avaliação na Educação à Distância; o modelo de Kirkpatrick. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, SBIE, 10., 1999. **Anais...** Curitiba: Universidade Federal do Paraná, 1999.
- [VIA 98] VICENTE, A.; PAIN, H. Motivation Diagnosis in Intelligent Tutoring Systems. In: INTERNATIONAL CONFERENCE INTELLIGENT TUTORING SYSTEMS, 4., 1998, San Antonio. **Proceedings...**Berlin: Springer-Verlag, 1998. p.86-95.
- [VIC 92] VICCARI, R.M.; OLIVEIRA, F. **Sistemas Tutores Inteligentes.** Porto Alegre: CPGCC da UFRGS, 1992. 63p. (RP-205).
- [WOO 95] WOOLDRIDGE, M.; JENNINGS, N. **Intelligent Agents: Theory and Practice.** [S.l.:s.n.], 1995. p. 62. Submitted to Knowledge Engineering Review.