

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

**Uma arquitetura de tutor
utilizando estados mentais**

por

LUCIA MARIA MARTINS GIRAFFA

Tese de doutorado submetida como requisito parcial para
a obtenção do grau de Doutor em
Ciência da Computação

Dra. Rosa Maria Viccari
Orientadora

Porto Alegre, maio de 1999

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Giraffa, Lucia Maria Martins

Uma arquitetura de tutor utilizando estados mentais/Lucia Maria Martins Giraffa. – Porto Alegre: PPGC da UFRGS, 1999.

177. f.:il.

Tese (Doutorado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 1999. Orientador: Viccari, Rosa Maria.

1. Sistemas Tutores Inteligentes. 2. Sistemas Multiagentes. 3. Agentes pedagógicos. 4. Jogos Educacionais. I. Viccari, Rosa Maria. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Prof^a. Wrana Panizzi

Pró-Reitor de Pós-Graduação: Prof. Franz Rainer Semmelmann

Diretor do Instituto de Informática: Prof. Philippe Navaux

Coordenador do CPGCC: Prof. Carla Dal Sasso Freitas

Bibliotecária Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

Para André, Pedro, Helcio e Gelsa, com todo meu amor.

Agradecimentos

Aos meus amigos da PUCRS e da UFRGS pelo carinho, apoio e amizade ao longo desta caminhada.

Aos colegas e funcionários do CBLU de Leeds e em especial para John Self, por toda sua atenção e contribuições.

Ao professor Dr. Milton Madeira da Faculdade de Psicologia da PUCRS, pela sua ajuda na elaboração dos instrumentos de avaliação e organização dos resultados.

À PUCRS pela licença parcial remunerada que viabilizou o meu doutorado.

À CAPES pelo auxílio parcial para realização do período de doutorado sanduíche na Universidade de Leeds, Inglaterra.

A todas as coordenadoras das escolas que participaram dos experimentos e me abriram as portas e facilitaram o meu trabalho junto aos professores.

Ao André Raabe por toda sua colaboração, disponibilidade e parceria desde o início do programa de doutorado.

Ao Michael Móra por todo seu apoio e parceria ao longo dos últimos dois anos de tese. Valeu amigo!

Aos meus amigos de fora do ambiente universitário que acompanharam toda a longa trajetória deste trabalho e sempre me deram muito apoio e compreensão.

Aos meus familiares que tanto me incentivaram e torceram por mim ao longo destes anos e souberam pacientemente entender o motivo pelos quais eu não fui a muitos almoços, jantares, churrascos e festinhas da família sempre com a famosa frase: “*..não dá para ir, eu tenho que trabalhar na tese.*”

À minha orientadora e amiga muito querida, Rosa, pelo seu exemplo, dedicação e profissionalismo. Um mundo com mais “Rosas” certamente seria melhor.

Ao meu filho Pedro pela inspiração que sempre foi na minha vida.

Aos meus queridos pais, Helcio e Gelsa, que sempre estiveram comigo me apoiando e incentivando em todas as horas. Vocês são muito especiais e pais maravilhosos.

Finalmente, mas não com menos importância, um agradecimento muitíssimo especial ao meu marido André que sempre acreditou em mim e me brindou com seu carinho, amor e compreensão em todas as horas.

Sumário

Lista de abreviaturas.....	7
Lista de figuras.....	8
Lista de tabelas.....	8
Resumo.....	10
Abstract.....	12
1 Introdução.....	14
1.1 O CONTEXTO DESTA TRABALHO.....	14
1.2 MOTIVAÇÃO.....	19
1.3 OBJETIVO.....	20
1.4 CONTRIBUIÇÕES.....	21
1.5 ORGANIZAÇÃO.....	23
2 SISTEMAS TUTORES INTELIGENTES.....	25
2.1 OS PROGRAMAS EDUCACIONAIS.....	25
2.2 O MODELO DO ALUNO NOS STI.....	33
2.3 O COMPORTAMENTO DO TUTOR NOS STI.....	37
2.4 A QUESTÃO DAS MÚLTIPLAS ESTRATÉGIAS DE ENSINO EM STI.....	39
3 STI MODELADO ATRAVÉS DE UMA ARQUITETURA MULTIAGENTE.....	42
3.1 IAD E AGENTES.....	43
3.2 STI MODELADOS ATRAVÉS DE UMA SOCIEDADE DE AGENTES.....	45
<i>3.2.1 O ambiente MATHEMA.....</i>	<i>46</i>
<i>3.2.3 AME-A: Ambiente Multiagente de Ensino-Aprendizagem.....</i>	<i>50</i>
<i>3.2.3 A arquitetura SEM.....</i>	<i>52</i>
<i>3.2.4 Ambiente de Aprendizagem Interativa multi-agente.....</i>	<i>53</i>
3.4 A ARQUITETURA MULTIAGENTE PROPOSTA PARA CONSTRUÇÃO DE UM STI.....	54
4 AGENTES PEDAGÓGICOS MODELADOS ATRAVÉS DE ESTADOS MENTAIS.....	60
4.1 TEORIAS MENTALÍSTICAS DE AGENTES.....	61
<i>4.1.1 Teorias formais de agentes.....</i>	<i>70</i>
4.2 O FORMALISMO LÓGICO UTILIZADO NA TESE [MOR99].....	73
4.3 OS AGENTES COGNITIVOS MODELADOS COM ARQUITETURA BDI.....	77
4.4 A COREOGRAFIA DOS ESTADOS MENTAIS.....	79
5 O TUTOR MODELADO ATRAVÉS DE ESTADOS MENTAIS.....	84
5.1 ESTRATÉGIAS DE ENSINO DO TUTOR.....	85

5.2 AS ESTRATÉGIAS DE ENSINO DO TUTOR E SUA REPRESENTAÇÃO NO FORMALISMO.....	89
6 EXPERIMENTOS REALIZADOS COM O PROTÓTIPO.....	97
6.1 HISTÓRICO E ETAPAS REALIZADAS.....	97
6.2 DEFINIÇÃO DA INTERFACE.....	98
<i>6.2.1 Metodologia utilizada para melhoria da interface.....</i>	<i>99</i>
6.3 POTENCIAL PEDAGÓGICO DO AMBIENTE SOB A PERSPECTIVA DOS PROFESSORES	101
<i>6.3.1 Metodologia utilizada para validação pedagógica.....</i>	<i>102</i>
6.4 O MCOE NO AMBIENTE DE SALA DE AULA	103
6.5 OS RESULTADOS DO EXPERIMENTO	104
7 CONCLUSÕES, LIMITAÇÕES E TRABALHOS FUTUROS.....	107
7.1 LIMITAÇÕES.....	109
7.2 TRABALHOS FUTUROS	110
<i>7.2.1 Utilização da arquitetura.....</i>	<i>110</i>
<i>7.2.2 Experimento envolvendo a aprendizagem dos alunos.....</i>	<i>111</i>
<i>7.2.3 Trabalhos a serem realizados com o protótipo</i>	<i>111</i>
ANEXO 1 – EVOLUÇÃO DO MCOE	113
ANEXO 2 – MANUAL DO USUÁRIO DO MCOE	116
ANEXO 3 – QUESTIONÁRIO	140
ANEXO 5 – EXEMPLO DO CONTEÚDO DO ARQUIVO DO TUTOR	148
ANEXO 6 - EXEMPLO DE ARQUIVO DO ALUNO	155
ANEXO 7 - PUBLICAÇÕES RELACIONADAS COM A TESE.....	162
BIBLIOGRAFIA.....	166

Lista de Abreviaturas

AEAC	Ambientes de Ensino-aprendizagem computadorizados
AIED	Artificial Intelligence in Education
CAI	Computer Assisted Instruction
CBLU	Computer Based Learning Unit
CC	Ciência da Computação
DAI	Distributed Artificial Intelligence
2D	Duas dimensões (espaço \mathfrak{R}^2)
GIA	Grupo de Inteligência Artificial
H&S	Hardware e Software
IA	Inteligência Artificial
IAD	Inteligência Artificial Distribuída
ICAI	Intelligent Computer Assisted Instruction
IE	Informática na Educação
ILE	Intelligent Learning Environment
STI	Intelligent Tutoring Systems
MAS	Multi-Agent Systems
MCOE	Multi-Agent Co-operative Environment
MOO	MUDs Object Oriented
MUDs	Multi-User Dimensions
PUCRS	Pontifícia Universidade Católica do Rio Grande do Sul
SE	Sistemas Especialistas
SEM	Sociedade dos Estados Mentais (Mental States Society)
SMA	Sistemas Multiagentes
SMAC	Sistemas Multiagentes Cognitivos
SMAH	Sistemas Multiagentes Híbridos
SMAR	Sistemas Multiagentes Reativos
STI	Sistemas Tutores Inteligentes
3D	Três dimensões (espaço \mathfrak{R}^3)
UFRGS	Universidade Federal do Rio Grande do Sul
WEB	World Wide Web

Lista de Figuras

FIGURA 1.1 - TAXONOMIA PARA PROGRAMAS EDUCACIONAIS	26
FIGURA 1.2 - ARQUITETURA CLÁSSICA DE UM STI.....	29
FIGURA 2.1 - ARQUITETURA DO MATHEMA.....	47
FIGURA 2.2 - STI COM ARQUITETURA MULTIAGENTE.....	55
FIGURA 2.2 - A DISTRIBUIÇÃO DOS AGENTES NO MCOE.....	56
FIGURA 2.3 - DISTRIBUIÇÃO DOS OBJETOS.....	57
FIGURA 2.4 - MODELO DO ALUNO	58
FIGURA 4.1 - ELEMENTOS DA LINGUAGEM DO X-BDI.....	76
FIGURA 5.1 - ARQUITETURA DO TUTOR.....	84
FIGURA 6.1 - EVOLUÇÃO DA INTERFACE DO SISTEMA.....	100

Lista de Tabelas

TABELA 5.1 - Exemplos de Mensagem do Tutor.....	88
--	-----------

Resumo

Esta tese situa-se na área de IA (Inteligência Artificial) aplicada à educação incluindo características interdisciplinares tanto da própria IA como de IE (informática na Educação). Faz-se também necessário constarem, aspectos referentes à Ciência da Computação e Educação a fim de melhor situar a complexidade e a dimensão do trabalho desenvolvido.

A utilização de técnicas de IA na elaboração e no desenvolvimento de ambientes de ensino-aprendizagem computadorizados tem se constituído em objeto de investigação por parte dos pesquisadores da área de Informática aplicada à Educação, devido as suas potencialidades. A utilização de agentes na modelagem e no projeto de STI permite-nos resgatar antigos problemas em aberto, como por exemplo a melhoria da interação entre tutor e aluno e a possibilidade de investigação dos processos mentais em nível mais estratificado.

A arquitetura descrita nesta tese utiliza a metodologia que vem sendo aplicada ao projeto de STI, onde são contempladas diferentes formas de se trabalhar com um determinado conhecimento (estratégias de ensino e táticas associadas), levando-se em consideração o tipo de usuário que está interagindo com o sistema. A arquitetura, elaborada segundo uma abordagem construtivista, prevê que o tutor seja menos diretivo e menos controlador das ações do aluno. O controle é feito na forma de monitoração para que o tutor funcione como um parceiro, ou seja, como facilitador do trabalho do aluno. Contudo, devido às características da modalidade escolhida para construção do protótipo (jogo educacional), precisamos ter algumas atitudes no tutor que garantam que o sistema não entre em colapso. O que inviabilizaria o trabalho do aluno. Nestas situações críticas, o tutor vai se comportar de maneira mais diretiva. Cabe salientar que a abordagem construtivista não significa dar liberdade total ao aluno nem privá-lo de qualquer tipo de assistência. Portanto, o que deve ser destacado é o grau de interferência do tutor, i.e., o quanto ele interfere no trabalho do aluno e se ele permite ou não que o aluno siga um caminho alternativo àquele que ele tem como o ideal para resolver o problema (heurísticas do tutor sobre o problema e forma de solução).

Além destes aspectos educacionais inerentes a todo o projeto de *software* educacional (necessários num trabalho desta natureza), esta tese está inserida no contexto da pesquisa em agentes cognitivos modelados através de seus estados mentais.

É importante salientar que os estados mentais utilizados neste trabalho (crenças, desejos, intenções e expectativas) funcionam como uma metáfora dos estudos mentais humanos. Por exemplo, quando se coloca a crença de um aluno a respeito de “lago”, na realidade está se colocando a crença que temos a respeito da crença que o aluno possui a respeito de “lago”. O mesmo acontece com os outros estados mentais aqui utilizados.

Os diálogos reais foram registrados através de observação direta e posteriormente analisados a fim de se identificar os estados mentais relacionados. Perguntas adicionais foram feitas no sentido de obterem-se mais elementos para auxiliar na inferência do conjunto de estados mentais que o aluno possui naquele momento em que estava jogando. Após a observação de vários alunos jogando, identificou-se um certo padrão nas suas atitudes quando executavam uma ação. Observações sucessivas permitiram delinear o conjunto de estados mentais associados à ação do aluno. Tal conjunto foi

utilizado como base para elaboração da coreografia. Estes dados servem de entrada para a construção do modelo do aluno mediante a interação com o tutor.

No presente trabalho, nós apresentamos a modelagem de um STI através do uso da tecnologia de agentes utilizando a arquitetura de SMA (Sistemas Multiagentes). O STI é concebido como um SMA híbrido composto por um ambiente reativo (SMAR - Sistema Multiagente Reativo) e um “kernel” cognitivo (SMAC - Sistema Multiagente Cognitivo). O SMAR e o SMAC interagem entre si de para ampliar as informações quantitativas e qualitativas oferecidas aos alunos que utilizam o sistema. Estas informações disponíveis é que irão permitir ao tutor selecionar estratégias de ensino mais adequadas a um determinado tipo de aluno. A principal contribuição desta tese está centrada no “kernel” cognitivo. Nós propomos uma arquitetura para o tutor que permitirá a monitoração de dois alunos trabalhando conjuntamente. Além disso, propomos uma forma de selecionar o comportamento do tutor para oferecer auxílio personalizado aos alunos considerando o perfil de cada um.

Esta arquitetura pretende ser uma alternativa de solução para uma questão importante na área de STI:

Como o tutor pode selecionar, entre várias estratégias de ensino, a mais adequada para cada perfil de aluno?

O grupo de pesquisa no qual este trabalho está inserido (GIA/UFRGS, sob orientação da Prof.a. Rosa Maria Viccari) tem realizado algumas avaliações experimentais, usando STI tradicionais e STI projetados e modelados através de sistemas multiagentes

As contribuições científicas listadas no texto deste trabalho possibilitaram que o grupo avançasse sua pesquisa na abordagem mentalística através da criação de uma arquitetura para o tutor e favorecesse a integração do trabalho desenvolvido por Móra et al. [MOR97; MOR98]. A utilização do modelo computacional de agentes criado por Móra et al. gerou a implementação do “kernel” cognitivo. Os desafios inerentes a implementação da arquitetura proposta para o tutor ampliaram as características do ambiente criado por Móra et al. e favorecem a junção de dois trabalhos de tese supervisionados pela mesma orientadora [MOR99]. Portanto, a nova arquitetura proporcionou ganhos tanto para tais pesquisas, como para o avanço das pesquisas desenvolvidas pelo nosso grupo.

Palavras-chave: sistemas tutores inteligentes, sistemas multiagentes, arquiteturas BDI, estratégias de ensino, modelagem de aluno.

TITLE: “ A tutor architecture using mental states.”

Abstract

The present thesis has been elaborated within the AI (Artificial Intelligence) applied to Education realm, and it brings specific contributions to the STI (Intelligent Tutoring System) area.

The use of AI techniques has been investigated by researchers of Computer Science applied to Education, due to its potentialities to improve educational systems. The agents' techniques used in the design of STI allow us to solve old problems opened in the area. For instance, the improvement of the interaction between tutor and student, and the possibility of tracing the mental processes in a more stratified way.

The architecture described in this thesis uses the methodology applied to the modern STI projects: multiple strategies for the tutor (i.e., teaching strategies and associated tactics). This approach considers different forms of working with a certain piece of knowledge, and is taken into consideration to create the user profile, as well as to monitor the student interaction with the system.

The architecture, designed according to a constructivist approach, expects the tutor to be less directive, and less controller of the student's actions. The control is made by an observation of students' actions by the tutor.

The tutor works either as a student's partner or as a facilitator. However, due to the characteristics of the modality chosen for construction of the prototype (educational game), we needed to take some attitudes in the tutor in order to avoid the system to collapse. What would make unfeasible the student's work. In these critical situations, the tutor will behave in more directive way. It fits to point out that the constructivist approach does not mean to give total freedom to the student or to deprive it of any kind of attendance. Therefore, what should be note here is the degree of the tutor's interference, i.e., how it interferes with the student's work using its own set of heuristics.

Besides these educational aspects, inherent to a project of Educational software, this thesis is inserted in the context of the research in cognitive agents modelled through their mental states (Believe, Desire, Intention, and Expectation).

It is important to point out that the mental states used in this work are as a metaphor of the human mental states. For example, when the student's has a believe regarding " lake ", in fact we area talking about belief that we have regarding the belief that the student possesses regarding about " lake ". The same happens here with the other mental states used.

The real dialogues were registered through direct observation in real situation (students playing with the game). They were analysed in order to identify the mental states connected with the student's actions. Additional questions were asked to obtain more elements to aid us to inference the group of mental states possessed by the student when he/she was playing. Successive observations allowed us to delineate the group of mental states associated to the student's action. Such group was used as a base for the choreography. These data were used as input for the construction to the student's model during the interaction with the tutor.

In this work, we have presented the design of an STI with the use of architecture of MAS (Multi-agent System Architecture). The ITS is conceived as a hybrid MAS composed by a RMAS (Reactive Multi-agent System) and a "cognitive kernel " using the CMAS (Cognitive Multi-agent System). The RMAS and the CMAS interact with each other to enlarge the quantitative and qualitative information offered to the students that uses the system. These available information allow the tutor to select teaching strategies more adapted to a certain student type.

The main contribution of this thesis is centred in the " cognitive kernel ". We propose an architecture for the tutor that will allow the two students to work together. Besides, we propose a way to select the tutor behaviour in order to aid the students considering their personal profile. This architecture intends to be an alternative solution for an important question in ITS research:

How can the tutor select, among several teaching strategies, the one that is more suitable for each student profile?

The research group in which this work is placed (GIA/UFRGS – Artificial Intelligence Group), under the supervision of Profa. Rosa Maria Viccari, has been doing executing some experimental evaluations, using traditional ITS and ITS modelled through multi-agents systems techniques.

The scientific contributions listed in this work allow the group to achieve some interesting results in the research of STI using a mental approach. The tutor architecture favoured the integration of the X-BDI (eXecutable Belief Desire, and Intention model)) developed by Móra et al.

The use the X-BDI allowed us to implement the " cognitive kernel ". The inherent challenges posed by the implementation of the architecture of the tutor refined the X-BDI environment. It favours the junction of two thesis works supervised done under the same supervisor[MOR99]. Therefore, the new architecture provided many gains for both researches, as well as for the progress of the research developed by our group.

Keywords: Intelligent Tutoring Systems, Multi-agent systems, BDI architectures, teaching strategies, student modelling.

1 Introdução

1.1 O contexto deste trabalho

Esta tese situa-se na área de IA (Inteligência Artificial) aplicada à educação incluindo características interdisciplinares tanto da própria IA como de IE (informática na Educação). Faz-se também necessário constarem, aspectos referentes à Ciência da Computação e Educação a fim de melhor situar a complexidade e a dimensão do trabalho desenvolvido.

O contexto de uma tese desta natureza é intrinsecamente amplo, devido à interdisciplinaridade, e pontual, uma vez que traz específica contribuição à área de programas educacionais da modalidade STI (sistema Tutor Inteligente).

A utilização de técnicas de IA na elaboração e no desenvolvimento de ambientes de ensino-aprendizagem computadorizados tem se constituído em objeto de investigação por parte dos pesquisadores da área de Informática aplicada à Educação, devido às suas potencialidades. Infelizmente, as promessas projetadas por Carbonel [CAR70] e seus colegas para os STI ainda não foram possíveis de serem atingidas. A idéia de termos um STI, em que o aluno receba auxílio personalizado conforme suas necessidades específicas, ainda está distante do que realmente gostaríamos de fazer. As razões para tais limitações residem tanto nos aspectos de *hardware* e *software* (H&S), quanto nos aspectos psicológicos e pedagógicos. Os recursos de H&S, apesar do grande avanço da tecnologia moderna, ainda são muito limitados e não conseguem reproduzir na sua totalidade a complexidade dos processos mentais de uma pessoa. As pesquisas nas áreas de Psicologia e Educação não oferecem ainda teorias computáveis que possamos aplicar e fazer simulações mais adequadas. O que fazemos são adaptações e combinações de soluções geradas pelas três áreas, a fim de melhorarmos os ambientes computadorizados de ensino-aprendizagem.

As tecnologias de hipermídia, de redes e do paradigma de agentes na pesquisa de *software* educacional foram rapidamente absorvidas pelos pesquisadores da área, especialmente na área de IA aplicada à Educação. Alguns STI foram construídos utilizando estas técnicas. Citamos como exemplos os trabalhos de Arkhas e Self [AKH96; AKH97], Colazzo [COL97], Dillenbourg [DIL97], Ikeda [IKE97], Khuwaja [KHU96], Leroux [LER96], Lester [LES97], Luckin [LUC98], Marsella e Johnson [MAR98], Mengelle et al. [MEN98], Self [SEL95], Viccari [VIC90] e Yin et al. [YIN98]. O modelo do estudante, no entanto, ainda é a parte mais frágil de tais sistemas. A fim de construir um STI com um modelo de estudante ideal precisamos entender o processo que ocorre na cabeça do estudante durante a interação com o tutor artificial e reproduzi-lo na máquina. Vários trabalhos têm sido desenvolvidos a fim de criar um modelo aproximado da coreografia intrínseca aos processos mentais do estudante ao longo do processo de aprendizagem. Os trabalhos de Winograd e Flores [WIN84], Devlin [DEV91], Dillenbourg [DIL96], Gagne [GAG96], Corrêa [COR95], Aimeur [AIM96] e Moussalle [MOU96] são exemplos de resultados neste sentido .

A utilização de agentes na modelagem e no projeto de STI permite-nos resgatar antigos problemas em aberto, como por exemplo a melhoria da interação entre tutor e aluno e a possibilidade de investigação dos processos mentais em nível mais estratificado. O estado da arte em STI pode ser avaliado pelos trabalhos de Andre

[AND97], Corbett [COR98a], Collazo [COL97], Costa [COS97], Cook [COO97], Dillenbourg [DIL97], Elliot [ELL97], Frasson [FRA97], Hayes-Roth [HAY97], Hietala [HIE97], Lewis [LEW97], Luckin [LUC98], Marsella e Johnson [MAR98], Mengelle et al. [MEN98], Mitsuru [MIT97], Morin [MOR97a], Roberts et al. [ROB98].

Os *software* educacionais podem ser divididos em dois grandes grupos: *CAI* (*Computer Aided Instruction*), fundamentados na teoria behaviorista-comportamentalista e os *Ambientes de Aprendizagem Interativa* fundamentados na teoria cognitivista-conceitual piagetiana. Os CAI¹ são programas que possuem uma modelagem menos complexa do que os programas da modalidade interativa.

As diferenças mais profundas entre os ICAI² (*Intelligent Computer Aided Instruction*) e os CAI tradicionais estão nas formas com que se concebem os seus projetos. Os CAI induzem o aluno a uma resposta correta mediante uma série de estímulos cuidadosamente planejados. Por outro lado, os ICAI pretendem simular algumas das capacidades cognitivas do aluno e utilizar os resultados como base das decisões pedagógicas a serem tomadas.

Muitos bons programas de CAI permitem ao aluno vivenciar uma série de simulações numéricas, físicas e biológicas que podem ser examinadas e testadas.

A técnica de IA diferencia-se da técnica utilizada em programas CAI. IA apresenta uma modelagem qualitativa. O que a torna vantajosa em relação à CAI, uma vez que podemos simular o processo de solução de problemas.

Desta forma podemos permitir que o aluno utilize um modelo de resolução de problemas e que este possa servir de base para se entender como aquele resolve os problemas propostos, utilizando parcialmente a assistência do ambiente.

Aos CAI se critica, principalmente, sua incapacidade de manter um diálogo aberto com o aluno. No caso dos STI, se pretende que o próprio computador adote uma forma de conduta inteligente que permita controlar o processo de aquisição do conhecimento por parte do aluno. Assim, segundo [SLE82], os STI são programas de computador que utilizam técnicas procedentes da IA para representar o conhecimento e levar a termo uma interação com o aluno. A programação com técnicas de IA permite uma excepcional graduação da generalidade [CLA90].

O objetivo fundamental dos STI é proporcionar uma instrução adaptada ao aluno, tanto no conteúdo como na forma, superando desta maneira alguns dos problemas mais cruciais do *software* educativo na atualidade.

Neste sentido, os STI deveriam apresentar um comportamento o mais próximo possível ao de um professor humano. Porém, a realidade está muito distante de alcançar tal propósito. Existem muitas razões para que isto ocorra, entre elas referimos à limitações em nível de *hardware* e *software* que não nos permitem colocar dispositivos que possam trabalhar com aspectos relativos aos sentidos do olfato, tato e visão. Um professor humano pode e leva em consideração tais estímulos para poder organizar seu trabalho junto ao aluno e, além disto, utiliza as saídas destes sentidos para fins de

¹ Os programas da modalidade CAI englobam os tutorias, exercício e prática, demonstrações, simulações e jogos. Todos concebidos

² ICAI e STI são utilizados como sinônimos por uma grande maioria de autores de IE. Nesta tese estamos utilizando num sentido mais amplo como será apresentado adiante.

feedback (retroalimentação). O fato é que desconhecemos a maneira com a qual nós humanos efetivamente processamos a informação dentro do nosso cérebro.

Temos um conjunto de teorias (umas mais trabalhadas do que outras) mas não temos ainda, uma resposta de como armazenamos, processamos e recuperamos informações no nosso cérebro. A resposta de como se estruturam tais aspectos é vital para a construção de um modelo de aluno e por consequência importantíssima para a construção de um STI. Segundo Asanome [ASA91], a dificuldade na construção deste modelo se encontra na falta de conhecimentos necessários para modelar o processo de aprendizagem do aluno.

As dificuldades em implementar o que já conhecemos a respeito do processamento de informação humano, com os ambientes de que dispomos (Prolog, Lisp, CLIPS e outros) têm apresentado restrições que implicam perda do modelo mental humano que conhecemos em favor de um modelo computacional, com perdas significativas de qualidade. Esta é uma questão que continuará em aberto até que surjam alternativas em nível de modelagem e implementação.

Segundo Viccari [VIC90], os STI são programas que modificam suas bases de conhecimento, percebem as intervenções do aluno e são dotados da capacidade de aprender e adaptar suas estratégias de ensino com o aluno mediante a interação. Caracterizam-se principalmente por construir um Modelo Cognitivo do Aluno através da interação, da formulação e da comprovação de hipótese sobre o estilo cognitivo do aluno. Baseiam-se, também, no procedimento do aluno, no seu nível de conhecimento do assunto, nas suas estratégias de aprendizagem e na capacidade de formular estratégias de ensino-aprendizagem adequadas ao aluno considerando a situação do momento. No processo de ensino através de um tutor inteligente *o aluno aprende fazendo*[VIC90].

Esta abordagem vem ao encontro das modernas correntes da Educação [CHA91] que estão privilegiando a troca de paradigma: o tradicional, modelo de ensino centrado no professor, pelo novo, modelo centrado no aluno e baseado no "aprender a aprender". Neste último, o aspeto da meta-cognição passa a ser relevante.

A meta-cognição sobre o processo de aquisição do conhecimento muda a forma de como encarar a aprendizagem, ou seja, transforma a visão de mera transferência de conteúdos do professor para o aluno, para fornecer-lhe condições e habilidades que lhe permitam explorar o seu meio, comparar e sintetizar conhecimentos.

As formas só fazem sentido na medida em que viabilizam o domínio de determinados conteúdos. Portanto, o que se precisa são novas formas de tratar o processo de aquisição de conhecimento [SAV90]. O que se propõe é uma mudança nas estratégias de ensino utilizando os conteúdos como uma forma do aluno assimilar o método e entender o processo a fim de que possa fazer transferência deste aprendizado para novas situações.

Segundo Hendley [HEN92], o uso de técnicas de IA na construção de *software* educacionais torna-se cada vez mais importante, uma vez que seu estudo fará com que os mesmos tenham mais qualidade. Destacam-se a seguir alguns aspectos de qualidade dos sistemas educacionais:

- os sistemas necessitam incorporar componentes de IA e estes estão longe do trivial no que concerne à implementação;

- a modelagem do domínio do conhecimento, particularmente no que se relaciona às habilidades trabalhadas proceduralmente e de forma acessível ao estudante, está longe de apresentar uma solução satisfatória no que refere à representação do conhecimento. Se um sistema é usado para provar um nível do problema, então isto pode ser visto como uma fonte de resolução do problema e os resultados podem ser usados pelo módulo de ensino;
- o uso de técnicas de IA no que diz respeito aos aspectos pedagógicos ainda não é eficaz. Os sistemas necessitam capacidades que os permitam construir uma estrutura com estratégia explícita de navegação selecionada a partir do aluno. Uma vez obtida esta estrutura, os resultados serão mais eficazes;
- tornar o conhecimento explícito e facilmente modificável para que o comportamento dos sistemas seja mais consistente e racional. Ao estudante pode ser permitida a interação com a base de conhecimento através da leitura, depuração e complementação de informações.

Acrescentar um **I** à sigla **CAI** não significa apenas agregar técnicas de IA para a construção de sistemas tutores, mas inclui trabalhar de forma interdisciplinar com as conquistas que outras áreas de pesquisa obtiveram em relação ao conhecimento da comunicação inteligente, tais como os avanços da psicologia e da pedagogia.

A arquitetura descrita nesta tese utiliza a metodologia que vem sendo aplicada ao projeto de STI, onde são contempladas diferentes formas de se trabalhar com um determinado conhecimento (estratégias de ensino e táticas associadas), levando-se em consideração o tipo de usuário que está interagindo com o sistema. A arquitetura, elaborada segundo uma abordagem construtivista, prevê que o tutor seja menos diretivo e menos controlador das ações do aluno. O controle é feito na forma de monitoração para que o tutor funcione como um parceiro, ou seja, como facilitador do trabalho do aluno. Contudo, devido às características da modalidade escolhida (jogo educacional), precisamos ter algumas atitudes no tutor que garantam que o sistema não entre em colapso, o que inviabilizaria o trabalho do aluno. Nestas situações críticas, o tutor vai se comportar de maneira mais diretiva. Cabe salientar que a abordagem construtivista não significa dar liberdade total ao aluno nem privá-lo de qualquer tipo de assistência. Portanto, o que deve ser destacado é o grau de interferência do tutor, i.e., o quanto ele interfere no trabalho do aluno e se ele permite ou não que o aluno siga um caminho alternativo àquele que ele tem como o ideal para resolver o problema (heurísticas do tutor sobre o problema e forma de solução).

Além destes aspectos educacionais inerentes a todo o projeto de *software* educacional (necessários num trabalho desta natureza), esta tese está inserida no contexto da pesquisa em agentes cognitivos modelados através de seus estados mentais.

Embora o termo “agente” tenha sido usado com frequência na Inteligência Artificial, não há uma noção uniforme e precisa para o seu significado. “Agente” pode ser entendido como uma entidade que exhibe alguns aspectos da inteligência humana. De um modo geral, para a Inteligência Artificial, agente é mais frequentemente associado a uma entidade que funciona contínua e autonomamente em um ambiente no qual existem outros processos e agentes [COR94]. A autonomia significa que as atividades de agentes não exigem intervenção humana contínua ou que elas possuem poder de autodeterminação, em um meio ambiente. A autonomia de um agente requer inteligência devido à necessidade de sobreviver em um ambiente real, dinâmico e nem sempre benigno. Está intimamente relacionada com o tipo de arquitetura do sistema;

assim, podemos ter agentes com diferentes graus de autonomia. O grau de autonomia e a inteligência que o agente possui permitem distinguí-lo de um simples componente de hardware ou *software* usual.

O agente também pode ser uma entidade à qual se atribuem estados, denominados de estados mentais, tais como crenças, decisões, capacidades, objetivos, intenções, compromissos e expectativas (conceitos análogos ou similares aos humanos) [SHO93]. Logo, a questão “o que é um agente” é substituída pela questão “que entidades podem ser vistas como tendo estado mental”. Isto leva à distinção entre a legitimidade de atribuir qualidades mentais para as máquinas e a sua utilidade. Atribuir certas crenças, intenções, conhecimentos, habilidades ou desejos para uma máquina ou programa de computador é legítimo quando a mesma informação expressa sobre a máquina é também expressa sobre uma pessoa usuária do sistema. Esta relação estabelece-se no uso do mesmo mecanismo de inferência pela máquina e pelo agente humano. Atribuir estados mentais para a máquina é útil quando ajuda a compreender a estrutura da máquina, o seu comportamento, o seu passado e futuro e como melhorá-la. Estes itens foram pesquisados e desenvolvidos por Daniel Dennett, John McCarthy, conforme citado em Shoham [SHO93].

Segundo essa abordagem, o *agente* é uma entidade, cujo estado consiste de componentes mentais como crenças, capacidades, escolhas e compromissos; por essa razão o estado de um agente é chamado de *estado mental*³. Esses componentes mentais devem ser definidos de uma forma precisa e ter uma correspondência direta com o seu uso no senso comum. Nessa visão, o que faz qualquer componente de hardware ou *software* ser um agente é precisamente o fato dele poder ser analisado e controlado em termos destes componentes mentais. A questão “o que é um agente” fica substituída; portanto, pela questão de quais entidades podem ser vistas como tendo estados mentais. neste sentido, a caracterização de agente constitui o que chamamos de *abordagem mentalística* [GIR98; GIR98e; GIR98f; GIR98f].

É importante salientar que os estados mentais utilizados neste trabalho funcionam como uma metáfora dos estados mentais humanos. Por exemplo, quando se coloca a crença de um aluno a respeito de “lago”, na realidade está se colocando a crença que temos a respeito da crença que o aluno possui a respeito de “lago”. O mesmo acontece com os outros estados mentais aqui utilizados.

Os diálogos reais foram registrados através de observação direta e posteriormente analisados a fim de se identificar os estados mentais relacionados. Perguntas adicionais foram feitas no sentido de obterem-se mais elementos para auxiliar na inferência do conjunto de estados mentais que o aluno possui naquele momento em que estava jogando. Após a observação de vários alunos jogando, identificou-se um certo padrão nas suas atitudes quando executavam uma ação. Observações sucessivas permitiram delinear o conjunto de estados mentais associados à ação do aluno. Tal conjunto foi utilizado como base para elaboração da coreografia. Estes dados servem de entrada para a construção do modelo do aluno mediante a interação com o tutor.

³ A modelagem de agentes utilizando estados mentais, juntamente com os detalhes da arquitetura BDI, são abordados em detalhes no capítulo 3.

1.2 Motivação

Quando observamos os programas educacionais que estão disponíveis no mercado, constatamos a pouca oferta de ambientes de ensino-aprendizagem que permitam sua exploração tanto por parte do aluno, como do professor. Os ambientes geralmente são fechados e não permitem a testagem de hipóteses acerca das possibilidades que apresentam. Por exemplo, quando o aluno manipula parâmetros do programa (o que podemos considerar como manipulação do domínio utilizado pelo sistema), uma interação mais dinâmica entre os ambientes e o usuário poderia ser explorada. No entanto, isto não é observado. Além disso os ambientes não oferecem nenhum tipo de auxílio personalizado ao estudante no sentido de orientá-lo levando em consideração suas características pessoais.

O trabalho que desenvolvemos nos últimos anos na área de *software* educacional buscaram oferecer uma proposta alternativa para a criação de ambientes educacionais que pudessem ser explorados de forma mais criativa por parte de alunos e professores [GIR93; SOU94; OLI95; GIR96; RAA96; ROS96; LUZ97; ALV97; GIR97; GIR98c; GIR98g]. Quanto a criação de programas educacionais, as limitações das técnicas convencionais de modelagem destes ambientes fizeram com que a aproximação da área de IA fosse inevitável devido ao potencial do uso destas técnicas (especialmente o uso de agentes no projeto de sistemas educacionais modalidade STI) no que diz respeito à melhoria dos seus aspectos qualitativos.

Os STI transcendem os enfoques de treinamento e prática, pois estão desenhados para assumir a responsabilidade de ensinar, utilizando um diálogo de iniciativa mista. Neste sentido, os seres humanos e a máquina são tratados como processadores de conhecimento.

Os processos da máquina só podem operar baseando-se em operações explícitas e regras. A interação natural entre humanos (professor-aluno) envolve afetividade, o que já não se processa da mesma forma na interação entre máquina-humano (computador-aluno). Os computadores ainda não podem relacionar-se afetivamente com os seres humanos⁴, cabendo às pessoas, na maioria das vezes, se adaptar ao “estilo da máquina” em vez de adaptar-se à natureza do indivíduo, fato que ocorre naturalmente quando existe a interação entre humanos (professor-aluno). Os STI procuram esta adaptação e utilizam todas as técnicas de IA disponíveis para tanto.

A destreza e o conhecimento humano é baseado em crenças e juízo que, que apesar de serem difíceis de se decompor, são fundamentais para modelar o processamento de informação humano. Sendo assim quanto mais utilizamos computadores como ferramentas intelectuais mais dependemos das características formais e menos das dimensões táticas e interpretativas do conhecimento.

Podemos pensar, segundo [STR85], que tais dimensões táticas e interpretativas do conhecimento não se dissociam; assim sendo, podemos ocultar a dimensão tática para melhor podermos trabalhar e podermos processar a informação através de regras.

⁴ A pesquisa na área de afetividade em máquina e afetividade de máquina ainda se encontra em estágio inicial no sentido de aplicação. O trabalho de Bercht [BER99] apresenta uma visão geral dos esforços despendidos neste sentido e os resultados preliminares. Segundo Bercht, os computadores não podem se relacionar afetivamente de uma forma natural e a pesquisa na área neste momento, trazem aspectos de funcionalidade que as emoções podem trazer.

Isto é o que observamos no caso das estratégias de ensino utilizadas nos modelos de STI. Tais estratégias são pensadas *ad hoc* devido a falta de uma compreensão e representação suficiente das teorias de aprendizagem humana. O realmente possuímos são os paradigmas de aprendizagem e não propriamente as teorias de aprendizagem, que deveriam ser aplicadas na construção de STI. Ou seja, não possuímos teorias de aprendizagem que possam ser descritas através de um processo formal.

[BOY92] salienta que é necessário que a Ciência da Computação realmente contribua para melhoria da Educação com a assistência de máquinas quase-inteligentes para o desenvolvimento de aprendizados cooperativos e conversações de ensino, não automáticas e estáticas, mas sim inteligentes. Maddux(in [WAX93, p.21]) no seu artigo *Past and Future stages in Educational Computing Research*, salienta que "...a pesquisa computacional está apenas começando e, deve continuar, enfatizando o tratamento das interações de aprendizagem".

Logo, o campo de pesquisa dos STI na abordagem de agentes, utilizando uma arquitetura SMA, implicou num trabalho de cunho interdisciplinar e no desenvolvimento de uma metodologia para o projeto, implementação e validação pedagógica de um STI, construído a partir da teoria construtivista/conceitual de se fazer educação.

Os resultados do presente trabalho juntamente com as possibilidades apresentadas pelas pesquisas interdisciplinares são fontes de motivação constante para o desenvolvimento de pesquisas. A crença de que os avanços na abordagem mentalística para agentes cognitivos melhorarão a qualidade dos ambientes de ensino-aprendizagem também se constituem motivação forte para s investigações.

1.3 Objetivo

No presente trabalho, nós apresentamos a modelagem de um STI através do uso da tecnologia de agentes utilizando a arquitetura de SMA. O STI é concebido como um SMA híbrido composto por um ambiente reativo (SMAR) e um "kernel" cognitivo (SMAC). O SMAR e o SMAC interagem entre si para ampliar as informações quantitativas e qualitativas oferecidas aos alunos que utilizam o sistema. Estas informações disponíveis é que irão permitir ao tutor selecionar estratégias de ensino mais adequadas a um determinado tipo de aluno. A principal contribuição desta tese está centrada no "kernel" cognitivo. Nós propomos uma arquitetura para o tutor que permitirá a monitoração de dois alunos trabalhando conjuntamente. Além disso, propomos uma forma de selecionar o comportamento do tutor para oferecer auxílio personalizado aos alunos considerando o perfil de cada um.

Esta arquitetura pretende ser uma alternativa de solução para uma questão importante na área de STI:

Como o tutor pode selecionar, entre várias estratégias de ensino, a mais adequada para cada perfil de aluno?

Após a análise dos diferentes modelos de alunos propostos na literatura (vide capítulo 1, item 1.2) e a verificação dos limites apresentados em tais modelos, optamos por esta questão através de um modelo qualitativo que agregue mais informações sobre o aluno. O que se deseja ao adicionar mais informação é permitir trabalhar num nível de granularidade mais fina a cerca do processo de construção de conhecimento por parte do tutor sobre o aluno. Com isto, o processo de seleção e adoção de estratégias de ensino por parte do tutor pode ser obtido através de um modelo que contemple outros estados mentais acrescentados às crenças que tradicionalmente eram utilizadas. Com este modelo composto de vários estados mentais, espera-se verificar o quanto a adição de desejos, intenções e expectativas⁵ pode contribuir para melhoria do comportamento do tutor.

1.4 Contribuições

O grupo de pesquisa no qual este trabalho está inserido (GIA/UFRGS, sob orientação da Profa. Rosa Maria Viccari) tem realizado algumas avaliações experimentais, usando STI tradicionais e STI projetados e modelados através de sistemas multiagentes utilizando redes locais e a WEB (Silveira e Viccari, 1997). Estas experiências permitem identificar as vantagens e restrições de utilizarmos novas tecnologias para construir os STI. Os resultados sugerem que a simples mudança do paradigma convencional para uma arquitetura cliente-servidor e o uso de uma interface gráfica não são suficientes para obterem-se resultados significativos. O uso de recursos de multimídia não garante por si só mais eficiência no processo [SIL97]. Por outro lado, as restrições destes sistemas podem ser minimizadas ao integrarmos a noção de cooperação no processo de ensino-aprendizagem através do enfoque de multiagentes. Estas técnicas e métodos nos permitem projetar um ambiente mais complexo, levando em conta agentes humanos externos e agentes internos modelados na máquina (computador).

Portanto, as contribuições principais desta tese são as seguintes:

- a utilização de um “kernel” cognitivo (baseado na arquitetura BDI) que processa as informações dos alunos sem a necessidade de usar as modelagens tradicionalmente utilizadas em STI (estereótipos, *overlay*, *buggy*). Desta forma, estende-se o modelo cognitivo do aluno constituído apenas de crenças agregando-se outros estados mentais (desejos, intenções e expectativas, estas tratadas como crenças futuras). Esta arquitetura trabalha com a construção de um modelo do aluno dinâmico e forte à medida que ele vai interagindo com o sistema. Isto permite ao tutor melhor identificar qual a maneira mais adequada de auxiliar o aluno considerando suas características pessoais;
- modelo do aluno baseado em estados mentais permite uma descrição qualitativa do aluno. Temos uma descrição do aluno e não uma mera classificação baseada em regras pré-definidas. Podemos falar a respeito do aluno evitando de apenas quantificá-lo sob o ponto de vista de valores ou percentagens. Se comparamos este modelo aos modelos utilizados na construção dos STI já desenvolvidos por

⁵ As expectativas não fazem parte da arquitetura BDI e, como estão sendo tratadas como crenças futuras, foram incorporadas ao conjunto de estados mentais. O uso da expectativa é importante em qualquer sistema de ensino-aprendizagem pois a expectativa está associada a execução dos planos e seu resultado [COR98].

[ASA91;AND97;FRA97;LEW97;PAV96; PAV97; WER95], observa-se que os modelos são eminentemente quantitativos. Ou seja, o aluno pode ser avaliado através do seu potencial de rendimento ou, ainda, através de comparações entre um modelo de aluno idealizado pelo tutor e o resultado apresentado pelo aluno após a interação. Alguns destes ambientes utilizam sofisticados recursos multimídia, como no caso de ADELE [LEW97], SEC-Tutor [WER95], ou modelagem de realidade virtual STEVE [AND97]. Estas tecnologias ampliam a interação do aluno com o sistema e possibilitam um tratamento muito sofisticado do domínio. No entanto, não possuem nenhuma descrição qualitativa no sentido de permitir ao tutor comparar o aluno em relação a ele mesmo. Diferentemente destes modelos, nós propomos que o tutor detenha informações passadas pelo próprio aluno que se modificam ao longo da sessão de trabalho à medida em que o aluno modifica ou não o seu comportamento. Esta possibilidade diferencia-se dos modelos clássicos que medem o desempenho do aluno.

- A possibilidade de analisarmos o arquivo gerado durante a sessão de trabalho com os alunos e o tutor e observarmos os estados mentais que foram aparecendo permitiriam o acesso a informação adicional sobre os alunos. Também tornou-se viável melhorar a forma com que o tutor decide o seu comportamento (conjunto de heurísticas utilizadas pelo tutor). Na verdade, os dados quantitativos são usados como indicadores quantitativos. São indicadores de como funciona o comportamento do aluno e poderão ser utilizados para depurar o comportamento do tutor. Tais dados vão agregar ao conjunto de crenças do tutor novas possibilidades de comportamento mediante as reações do aluno. Sabemos que a primeira vez que um professor ministra uma disciplina, ele traz consigo um conjunto determinado de crenças. Ao longo do curso, tais crenças são revisadas diante da reação dos alunos. Desta forma, este mesmo professor passará a dispor de um novo conjunto de crenças podendo incluir e/ou excluir aquelas que dispunha no início do curso em relação a mesma disciplina. Isto faz parte do processo de amadurecimento que o tutor humano sofre. Com esta arquitetura este processo pode ser incorporado ao tutor artificial, uma vez que isto não implica em redesenhar o sistema e sim colocar na base do tutor mais informação.
- A arquitetura do tutor possibilita aos programas tradicionais de ensino receber o “kernel” cognitivo e transformá-lo em STI (não sendo esta uma tarefa trivial, mas que pode ser realizada). Isto resulta da independência que a arquitetura multiagente do tutor possui em relação às propostas tradicionais de STI. Também permite sua generalização para um framework para criação de STI;
- A arquitetura geral, proposta para projetar um STI com interface de jogo educacional, permite que áreas de aplicação (domínio) que requeiram uma modelagem mais complexa possam se utilizar de um SMAR. A construção de simulações baseadas em agentes reativos e suas propriedades podem acrescentar um ganho qualitativo na exploração do ambiente. Por outro lado, as técnicas convencionais não favorecem tal exploração. Além disso, a utilização de um SMAR apresenta repercussões positivas na adequação dos ambientes para representação de fenômenos naturais que requerem maior dinamismo.

As contribuições científicas listadas acima possibilitaram que o grupo avançasse sua pesquisa na abordagem mentalística através da criação de uma arquitetura para o tutor e favorecesse a integração do trabalho desenvolvido por Móra et al. [MOR97; MOR98]. A utilização do modelo computacional de agentes criado por Móra et al. gerou a

implementação do “kernel” cognitivo. Os desafios inerentes à implementação da arquitetura proposta para o tutor ampliaram as características do ambiente criado por Móra et al. e favorecem a junção de dois trabalhos de tese supervisionados pela mesma orientadora [MOR99]. Portanto, a nova arquitetura proporcionou ganhos tanto para tais pesquisas, como para o avanço das pesquisas desenvolvidas pelo nosso grupo. Nós avançamos na direção de uma proposta de *framework* para elaboração de STI que será detalhada posteriormente neste trabalho.

Além das contribuições sob o ponto de vista da IA e da engenharia de *software* utilizada em STI, ressaltamos os seguintes aspectos pedagógicos da proposta:

- difere de STI que utilizam textos como forma de interação entre o aluno e o sistema, nossa proposta é oferecer um ambiente visual.
- Os alunos aprendem manipulando elementos gráficos (i.e., objetos) diretamente na tela e verificam sua imediata reação diante das suas ações. O aluno parte da ação para chegar ao conhecimento;
- utiliza de múltiplas estratégias de ensino consideradas sob o ponto de vista construtivista implicou em um projeto e implementação mais sofisticados da interface. Desta forma, os recursos tradicionalmente aplicados a jogos comerciais foram adaptados para um ambiente educacional;

não existe a necessidade de se redefinir o tutor para que ele possa trabalhar com alunos diferentes. O conjunto de estados mentais associados às ações do aluno podem mudar e isto não implica em redefinir o comportamento do tutor. Ele vai se comportar baseado nos dados que possui adequando-se aos comportamentos diferenciados de cada aluno.

- a possibilidade dos alunos trabalharem de forma cooperativa para solucionar problemas comuns é mais um elemento utilizado pelo tutor para auxiliar o aluno. O tutor percebe através das ações dos alunos quando existe a cooperação ou não.

Estas contribuições serão abordadas em detalhes ao longo dos capítulos.

1.5 Organização

Esta tese constitui-se de oito capítulos, sendo o primeiro capítulo constituído da presente introdução. Em cada capítulo serão abordados os seguintes assuntos:

Capítulo 2:

- considerações sobre ambientes de ensino-aprendizagem;
- os pressupostos pedagógicos e psicológicos que envolvem o projeto de um programa educacional;
- visão geral sobre STI.

Capítulo 3:

- apresenta os STI modelados através do uso de agentes pedagógicos;
- a arquitetura geral proposta para construção de STI utilizando SMAH.

Capítulo 4:

- teorias mentalísticas de agentes;
- teorias formais de agentes;
- formalismo lógico utilizado nesta tese;
- os agentes cognitivos modelados na arquitetura BDI.

Capítulo 5:

- estratégias de ensino do tutor;
- exemplos da coreografia realizada durante a interação entre o tutor e os alunos.

Capítulo 6:

- os experimentos realizados com o protótipo;
- a investigação realizada com os professores de séries iniciais a fim de validar a proposta pedagógica intrínseca na arquitetura.

Capítulo 7:

- apresenta as conclusões obtidas com esta pesquisa;
- apresenta as limitações deste trabalho;
- perspectivas para futuras investigações que serão realizados a partir dos resultados alcançados com esta tese.

Os anexos apresentam o histórico do desenvolvimento do protótipo, o manual do usuário do MCOE, o instrumento de pesquisa utilizado para coleta de dados na escola, exemplo do arquivo utilizado na modelagem do tutor, exemplo dos estados mentais associados às ações dos alunos e as publicações relacionadas ao trabalho da tese.

2 Sistemas Tutores Inteligentes

O projeto de um programa educacional implica numa série de tomadas de decisões que vão desde o aspecto técnico até o aspecto pedagógico. A escolha da modalidade de programa educacional que se deseja implementar traz consigo uma série de pressupostos psico-pedagógicos que devem ser observados pelos projetistas. Estes pressupostos refletem a crença que o projetista possui a respeito de Educação. Além disso, tais pressupostos influenciam diretamente no projeto do ambiente educacional. Para escolher qual a modalidade que deseja implementar, o projetista encontra uma série de dificuldades ao utilizar as taxonomias encontradas na literatura. Os novos sistemas apresentados utilizam recursos hipermídia, permitem o trabalho individual ou coletivo, utilizam técnicas de IA em diferentes escalas e, conforme a metodologia utilizada pelo professor, podem ser explorados de diferentes formas. Todas estas possibilidades fazem com que a classificação dos novos sistemas em uma única modalidade seja bastante complicada.

Portanto, retomar esta discussão e apresentar uma taxonomia revisada que leve em consideração a tendência atual de classificar os programas pelo tipo de aprendizagem é uma questão importante para um melhor entendimento dos AEAC (Ambientes de Ensino-Aprendizagem Computadorizados) e, em especial, para a construção da arquitetura proposta nesta tese.

Este capítulo trata das questões referentes aos aspectos pedagógicos e psicológicos envolvidos no projeto de um ambiente de ensino-aprendizagem computadorizado. No decorrer deste capítulo, os aspectos relacionados à abordagem construtivista e social utilizadas na concepção da arquitetura proposta para o tutor serão enfatizados.

2.1 Os programas educacionais

“Todo o programa pode ser considerado um programa educacional desde que utilize uma metodologia que o contextualize no processo de ensino-aprendizagem”. Esta é uma conceituação cada vez mais consensual na comunidade de IE. Além disso, permite que uma série de programas desenvolvidos para outras aplicações venham a ser utilizados como programas educacionais. Esta flexibilidade em conceituar os programas educacionais, ao mesmo tempo que amplia o conjunto de programas que podem ser utilizados por educadores, faz com que aumente a necessidade de revisarmos as taxonomias tradicionalmente utilizadas na literatura.

Outra questão que surge é por que precisamos classificar os programas educacionais. A idéia de classificar e rotular um programa através de suas características estruturais e funcionais é válida no sentido de organizar os programas para facilitar o seu acesso em uma biblioteca de programas escolares. Outra vantagem desta classificação é favorecer sua análise sob o ponto de vista de projeto e/ou pedagógico.

Uma taxonomia deve ser útil para auxiliar os projetistas a determinarem as características básicas que irão predominar no sistema e para determinar o processo de modelagem e implementação do ambiente através de uma metodologia adequada. Com a evolução dos recursos de H&S, estamos diante de ambientes educacionais mais complexos, tanto no aspecto de projeto do programa, como na forma de sua utilização.

Se os estilos de aprendizagem dos alunos diferem, então não podemos pensar em ensinar todos os alunos da mesma maneira [PIL96]. Devemos oferecer uma diversidade de atividades que contemplem todos os alunos ao longo do processo de aprendizagem. Para atingir este objetivo, devemos utilizar recursos em que as abordagens se adequem aos diferentes tipos de alunos. Um dos recursos que apresenta um futuro promissor neste final de milênio é a aprendizagem adaptada às necessidades do aluno através de AEAC.

As taxonomias mais tradicionais utilizadas para classificar os programas educacionais não contemplam as modalidades que utilizam técnicas de IA e os ambientes cooperativos. Inclusive, muitas delas não levam em consideração um aspecto fundamental: o tipo de aprendizagem proporcionada pelo ambiente. A necessidade de se fazer uma reflexão neste sentido originou a figura 1.1 que apresenta uma taxonomia dividindo os programas educacionais em dois grandes grupos:

- Programas com a aprendizagem do aluno dirigida a um conjunto de habilidades específicas;
- Programas para aprendizagem de habilidades cognitivas amplas.

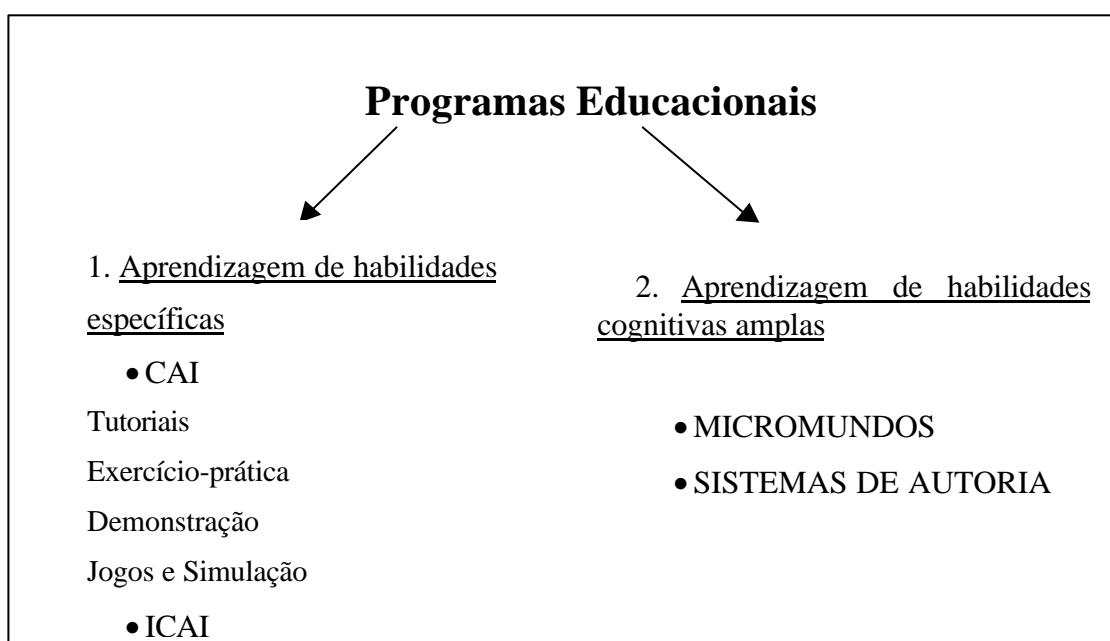


FIGURA 1.1- Taxonomia para Programas Educacionais

A taxonomia apresentada na figura 1 será discutida no decorrer deste trabalho de maneira a explorar as características de cada categoria sem compromisso cronológico. Gostaríamos de ressaltar que nos deteremos mais em alguns aspectos mencionados, uma vez que estes apresentam maior potencial pedagógico, fator este fundamental para nossa proposta.

Na categoria 1 estão os programas onde a aprendizagem proporcionada pelo ambiente está centrada na aquisição de habilidades específicas (motricidade fina (manuseio do mouse), percepção, identificação e outras).

Os programas educacionais desta modalidade se dividem em dois grandes grupos:

- os CAI, surgidos na década de 50 e oriundos de projetos da área de Educação;
- os ICAI, surgidos na década de 70, a partir de projetos de pesquisas da área de IA.

O fato destes dois grupos serem originados de grupos de pesquisas diferenciados (Educação e IA) faz com que a identificação das modalidades dos programas que constituem cada categoria apresentem mais ou menos consenso na comunidade de IE. Os CAI, por serem oriundos da Educação, apresentam propostas baseadas nas teorias aceitas pela comunidade de educadores. No entanto, cabe salientar que o que é considerado como teoria na área de Educação, nem sempre satisfaz os requisitos de uma teoria formal na área de CC. A maioria das teorias consideradas em Educação não apresentam um modelo computacional necessário sob o ponto de vista da CC[GIR95].

Os CAI nas versões iniciais apresentavam instruções programadas que repetiam na máquina os mesmos métodos utilizados pelo paradigma vigente na época (behaviorista), ou seja, o aluno utilizava um conjunto de lições previamente organizadas pelo professor (instrutor) de forma seqüencial e com pouca interação. Os CAI foram criados para oferecer suporte ao ensino de habilidades específicas sem a utilização do modelo do aluno para orientar a forma de interação. O conteúdo é pré-programado pelo professor baseado num currículo de referência (geralmente único) e elaborado proceduralmente. Nestes ambientes existem uma série de passos (tarefas) a serem executados e à medida que o aluno atinge uma determinada marca (rendimento ou “aprendizado”) novos níveis são disponibilizados.

No grupo dos CAI vamos encontrar as seguintes modalidades:

- *Programas de reforço ou exercício* - o aluno pratica e testa conhecimentos de forma dirigida e procedural. As versões mais atuais destes programas utilizam recursos hipermídia mantendo estas características;
- *Tutoriais* - seguem o padrão de ensino da sala de aula tradicional onde o conteúdo é previamente organizado numa estrutura definida pelo professor e o aluno seleciona dentre as diversas opções disponíveis o conteúdo que deseja estudar. Mesmo aqueles que utilizam estrutura de hipermídia ou foram escritos em versão WEB, mantêm as características do ensino tradicional e não apresentam grandes revoluções, sob o ponto de vista pedagógico;
- *Jogos educacionais e as simulações* - são modalidades que adquiriram uma projeção maior nos últimos anos devido ao desenvolvimento dos recursos de H&S disponíveis no mercado. As diferenças conceituais entre os jogos e as simulações podem ser caracterizadas pelo fato de que o jogo é um processo intrinsecamente competitivo (em que co-existem a vitória e a derrota) e uma simulação é a simples execução dinâmica de um modelo previamente definido.
- Os *jogos educacionais* são *exemplos* de ambientes de resolução de problemas que podem ser projetados e explorados com uma abordagem construtivista, o que os colocaria na categoria 2. Nos ambientes concebidos na abordagem construtivista, existem ferramentas e possibilidades de combinações para que o aluno resolva um determinado problema, como no caso do protótipo desenvolvido para esta tese. Apesar disso, a interface

inspirada nos jogos é apenas um recurso a mais para ampliar o tipo de habilidade que se deseja proporcionar ao aluno. Os jogos apresentam restrições para a atividade do aluno; por exemplo, a limitação de tempo e o objetivo a atingir. Está implícito um sentido de competição que pode ser explorada positivamente.

- A *simulação* implica em um modelo computacional que procura dar conta dos eventos que acontecem no ambiente. Os eventos acontecem de forma contínua em relação ao tempo e de forma discreta em relação às ações, uma vez que são interrompidos e retomados durante o processo. Muitos ambientes educacionais não são simulações, mas emulações da realidade, uma vez que não se utilizam de modelos computacionais.

A emulação é uma instância da simulação. Esta diferenciação não é considerada no âmbito da Informática na Educação, pois se utiliza simulação como sinônimo de emulação. Isto se torna um problema, uma vez que os leitores da área de Ciência da Computação não entendem o sentido da palavra “simulação” empregado na descrição de vários ambientes educacionais descritos na literatura de IE. Outro aspecto a considerar é a diferença entre a simulação determinística (o resultado é sempre o mesmo quando determinados parâmetros são colocados) e a estocástica (existe uma aleatoriedade nos parâmetros que faz com que a simulação produza resultados diferenciados). Apesar das diferenças existentes, tanto os jogos quanto as simulações, apresentam uma seqüência de passos semelhantes no que diz respeito ao planejamento e aplicação destas técnicas.

No planejamento de jogos e simulações é de vital importância a definição e fixação dos objetivos da atividade, a determinação do contexto desejado para a mesma, a identificação dos recursos utilizáveis para se alcançar os objetivos finais e a determinação da seqüência de interações. Além da vantagem de ordem motivacional que os jogos e as simulações apresentam, há vantagem prática. Ou seja, os jogos e as simulações facilitam a solução de problemas de forma intuitiva, incentivando desta forma o desenvolvimento do raciocínio intuitivo. Os jogos e as simulações favorecem formas de raciocínio alternativas, que diferem do raciocínio analítico usual. Estas formas alternativas estão intimamente relacionadas ao desenvolvimento da autoconfiança e auto-estima do aluno. As vantagens apresentadas vem despertando o interesse dos educadores.

Com a evolução vertiginosa dos recursos computacionais e a popularização dos computadores, observamos uma melhoria na qualidade das interfaces e um aumento nas possibilidades de interação entre o usuário (aluno) e o programa educacional.

O grupo dos ICAI apresenta uma estrutura diferenciada dos CAI. Os programas utilizam técnicas de IA e os resultados da Psicologia Cognitiva com uma arquitetura modular. Baseiam-se no conteúdo e independe do método de ensino utilizado (estratégias e táticas empregadas na interação com o aluno) em relação ao domínio. Os sistemas ICAI foram projetados como uma tentativa de fazer com que o programa educacional deixe de ser um mero *virador de páginas eletrônico*⁶ e se torne um elemento mais ativo no processo de ensino-aprendizagem.

⁶ Este termo *virador de páginas eletrônico* aparece com freqüência na literatura da área, como por exemplo em [OLI94]. É utilizado como uma crítica aos programas CAI que apenas usam o meio eletrônico para fazer o mesmo que era realizado no papel, sem nenhum ganho significativo em nível de

Dentro dos ICAI encontramos a modalidade STI. Nos STI existe um modelo de aluno que objetiva personalizar o trabalho conforme as diferenças individuais de cada usuário. Os modelos de aluno (forte e fraco) determinam se o STI é um *Tutor* ou um *Assistente*. Os tipos de modelo do aluno pretendem reproduzir o tipo de aluno que está interagindo com o sistema (ambiente) através da utilização de diversas técnicas de modelagem. Estas técnicas consideram desde modelos estereotipados simples até modelos sofisticados envolvendo estados mentais.

As estratégias de ensino contêm a forma de comportamento que o sistema vai utilizar para auxiliar o aluno na utilização do conhecimento. Este comportamento está intrinsecamente conectado ao paradigma educacional que suporta as idéias do projetista do sistema. A maneira de se comportar do tutor pode ser mais ou menos diretiva dependendo dos objetivos educacionais do programa.

A base do domínio armazena o conteúdo a ser trabalhado com o aluno. Atualmente, a base do domínio pode apresentar um grau de sofisticação bastante complexo e diferenciado. Os sistemas mais modernos utilizam recursos de hipermídia e simulação para que o aluno acesse e manipule os conteúdos. As interfaces estão cada vez mais sofisticadas, a ponto de algumas utilizarem os recursos de realidade virtual.

O módulo de controle funciona como um articulador e coordenador dos demais módulos a fim de garantir um sincronismo adequado entre todas as partes. Nos ambientes mais modernos, esta função fica transparente para o usuário. Em sistemas que utilizam uma arquitetura multiagentes, o controle fica distribuído entre os vários agentes. A figura 1.2 apresenta a arquitetura clássica utilizada nos STI.

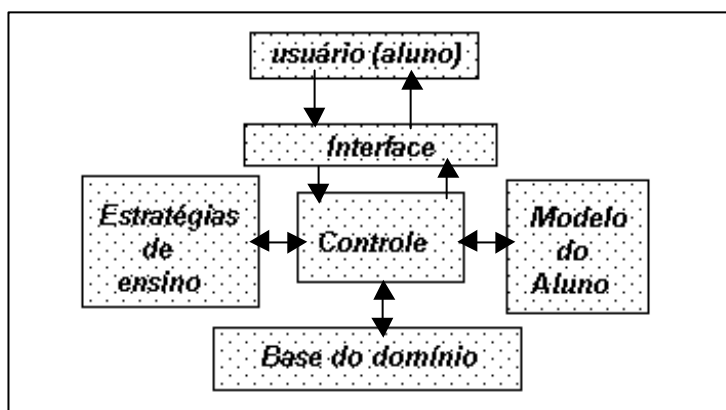


FIGURA 1.2 - Arquitetura clássica de um STI

Com a utilização de outros tipos de programas para fins educacionais, foram surgindo novas categorias para o grupo de programas. Assim, a taxonomia dos ICAI deixou de ser meramente sinônimo de STI. Ao examinarmos a literatura da área, observamos que não existe consenso a respeito da taxonomia dos ICAI, isto se justifica pela inexistência de uma teoria geral para explicar o processamento da informação humana e representá-la na máquina [NUN93].

Para Clancey [CLA89] e Fischer [FIS90], os ICAI e os STI são a mesma modalidade de programas, ou seja, são sinônimos.

Segundo [KEL87], existem três tipos de ICAI:

- Sistemas Tutores Inteligentes – com o uso do diálogo socrático (uso de perguntas indiretas considerando o nível de compreensão do aluno);
- Sistemas de Treinamento Inteligente - baseiam-se no desempenho individual do aluno e fornecem conselhos para o desempenho em uma área específica;
- Ambientes de Ferramentas Inteligentes - trabalham junto com o usuário.

Pinto [PIN95] faz uma divisão do STI baseado nas considerações de Clancey [CLA89] e divide os Sistemas Tutores Inteligentes em Tutores Inteligentes e Assistentes Inteligentes.

Os Assistentes Inteligentes são Sistemas Tutores Inteligentes implementados em microcomputadores (em rede ou *stand alone*⁷) que utilizam uma estratégia não-invasiva, ou seja, não interferem na dinâmica do aluno. O número de intervenções que possam eventualmente se suceder é pequeno e acontece para dar apoio ao aluno na forma de observações e/ou informações novas pouco conhecidas

Na taxonomia apresentada na figura 1 consideramos os SE e os STI como ICAI. Os STI foram divididos em tutores inteligentes e assistentes inteligentes, considerando-se [PIN95] e [KEL87].

A categoria 2 contempla os AEAC que permitem a aprendizagem de habilidades cognitivas amplas. Deseja-se que os alunos obtenham um nível mais elevado de aprendizagem que ultrapasse as habilidades mais simples. Segundo Kneller [KNE88], um indivíduo criativo deverá desenvolver as seguintes habilidades:

- consciência sobre o conhecimento (i.e., o indivíduo deverá ser sensível aos pormenores do conteúdo e suas aplicações);
- fluência nas suas idéias;
- flexibilidade para perceber alternâncias nas situações que se apresentam ao longo do trabalho;
- capacidade de reorganização;
- capacidade de desenvolver sua originalidade (i.e., gerar idéias novas, sem imitar a realidade conhecida).

Neste grupo estão todos os programas educacionais que utilizam como referencial teórico o paradigma que enfatiza o trabalho do aluno e o “*aprender a aprender*”.

A categoria 2 dos programas educacionais está dividida da seguinte forma:

- Micromundos, que constituem o grupo dos ambientes de aprendizagem cognitivas amplas. Surgiram na década de 60 através dos trabalhos de Papert [PAP80] e se caracterizaram por ser uma proposta contrária aos CAI. Neste sentido, o ambiente permite que o aluno trabalhe de forma diversificada, segundo seu próprio ritmo e ainda possibilita que ele construa sua solução utilizando recursos de programação inerentes ao ambiente. Os micromundos se baseiam na

⁷ *Stand alone*: o microcomputador trabalha de forma isolada.

proposta de aprendizagem de Jean Piaget, em que a ênfase está na construção do conhecimento por parte do aluno e não na mera transmissão de conhecimentos (como ocorre na categoria CAI). Os micromundos foram criados para desenvolver habilidades cognitivas no aluno e para proporcionar o pensamento reflexivo. Além disso, possuem uma linguagem de programação com sintaxe simplificada (e não por isto menos poderosa) para que o aluno construa soluções para os problemas propostos. Trabalham basicamente com a solução de problemas e não apresentam conteúdos pré-fixados e repostas;

- Sistemas de Autoria - ferramenta de criação que possibilita ao aluno viabilizar seu projeto de trabalho. Utilizado desta maneira, um sistema de autoria oferece ao aluno a possibilidade de explorar um conjunto amplo de habilidades cognitivas exercendo sua criatividade;
- Jogos educacionais - apresentam uma concepção diferenciada daqueles apresentados na categoria 1. Nestes ambientes, existe um modelo de simulação onde o tipo de ação executada pelo aluno fará diferença no resultado do jogo. Não é mais uma mera questão de ganhar ou perder. O aluno atinge ou não um determinado grau de controle do jogo através do resultado que aparece na tela no final do tempo da partida. São ambientes mais sofisticados envolvendo um maior grau de complexidade tanto no seu projeto, como na sua implementação. Requerem uma equipe interdisciplinar mais diversificada do que os jogos da categoria 1;
- ILE - ambientes de aprendizagem social que utilizam técnicas de IA no seu projeto e no seu desenvolvimento. Se caracterizam por considerar mais de um aluno ou mais de um tutor trabalhando no mesmo ambiente. Podem incluir características das modalidades encontradas na categoria ICAI. Os ILE também são conhecidos como Sistemas Tutores Cooperativos ou Sistemas de Aprendizagem Social. Combinam aspectos das modalidades STI e Micromundos podendo agregar os elementos de simulações (jogos educacionais).

Alguns autores tradicionalmente citados na área de IE [TAY80; LUC94] consideram na suas taxonomias os programas que foram projetados para outra finalidade que não a pedagógica. Esses programas podem ser considerados como ferramentas educacionais através do uso de uma metodologia que os contextualize no processo de ensino-aprendizagem. Nesta categoria encontramos os programas aplicativos de uso geral (como processadores de texto, planilhas eletrônicas, sistemas gerenciadores de banco de dados e outros). A possibilidade de explorar o aspecto pedagógico de uma ferramenta não é algo novo em educação e seu uso para sistemas computacionais é uma extensão. Dessa forma, o uso desse recurso não permite que esses sistemas sejam tratados na classificação proposta na figura 1, onde se agrupa os sistemas intencionalmente desenvolvidos para fins educacionais.

A história da IE está ligada à evolução dos AEAC e esses são um reflexo do estado da arte da tecnologia e da pesquisa de grupos interdisciplinares. Segundo Edgar [EDG95], podemos traçar um paralelo entre o desenvolvimento da tecnologia computacional e o desenvolvimento da pedagogia educacional. Da mesma maneira que os sistemas computacionais observaram uma evolução e possibilitaram novos rumos para o projeto e para o desenvolvimento de programas educacionais, a pedagogia passou a considerar outras possibilidades de desenvolver o processo de ensino-aprendizagem.

A popularidade da abordagem construtivista fundamentada no trabalho de Piaget [PIA70; PIA74; PIA76; PIA78; PIA80] favoreceu a criação de programas mais abertos em que o aluno explorar o conhecimento de uma forma mais pessoal. A idéia de se ter apenas um conjunto de instruções organizadas num bloco único dá espaço para novas propostas, onde vários programas são integrados formando um ambiente de ensino-aprendizagem, como no caso dos micromundos..

Com o advento das redes de computadores e da WEB, surgem as versões de AEAC que favorecem o trabalho colaborativo e/ou cooperativo⁸. Cabe salientar que os programas educacionais podem ser trabalhados de forma colaborativa ou cooperativa independente de serem projetados para WEB ou para ambiente de rede.

Alguns dos programas educacionais na WEB mantêm as características básicas encontradas em suas versões anteriores incorporando aspectos sociais da aprendizagem. Estes aspectos sociais (descritos por Vygotsky [VYG84]) passaram a ser úteis e adequados para se modelar os novos ambientes distribuídos e intercomunicativos.

Os AEAC na WEB carregam características importantes sob o ponto de vista da abordagem social, uma vez que o desenvolvimento da linguagem de um indivíduo se dá através de um espaço social. Neste espaço, o aluno pode trocar experiências com seus pares e aceder a diferentes fontes de informação em diferentes formatos e organizações. Cada aluno manipula o conhecimento no seu tempo e o atendimento de suas necessidades não está ligado às restrições físicas. A única restrição que ele vai encontrar na rede é a linguagem utilizada para representar o conhecimento.

A tendência dos modernos AEAC é apresentar o uso de recursos de hipermídia, interfaces gráficas de baixo custo cognitivo, sistemas de auxílio cada vez mais personalizados e sistemas altamente interativos. Nesta nova geração de programas, busca-se fazer com que o aluno seja mais participativo e criativo. Uma maneira de classificar os AEAC neste sentido deve contemplar o tipo de aprendizagem que o ambiente proporciona. O tipo de aprendizagem está diretamente ligado ao conjunto de crenças que o projetista do programa possui e professa a respeito de Educação.

A evolução do programa educacional é consequência da evolução da própria CC e das necessidades que professores e alunos estão vivenciando na mudança de paradigma (de behaviorista para cognitivista). Os discursos educacionais indicam que as mudanças estejam ocorrendo; no entanto, sabe-se que a realidade escolar nem sempre reflete tais mudanças. Mas esta é uma consideração que não trataremos. Vamos considerar que esta mudança é mais do que uma tendência, ou seja, um fato na Educação deste final de milênio.

⁸ Cooperação é diferente de colaboração. Colaboração é compartilhar informação sem modificar a informação recebida. Cooperação é compartilhar informação e poder interferir e modificar a informação recebida, e atuar de forma conjunta para construir algo em comum. Toda a cooperação envolve colaboração.

2.2 O modelo do aluno nos STI

O modelo do aluno representa o conhecimento e as habilidades cognitivas do aluno em um dado momento. É constituído por dados estáticos e dados dinâmicos[VIC90] que serão de fundamental importância para o tutor poder comprovar hipóteses a respeito do aluno. Contêm uma representação do estado do conhecimento do aluno no momento em que interage com o STI. A partir desse modelo e do conteúdo a ser ensinado, o sistema deve ser capaz de inferir a melhor estratégia de ensino a ser utilizada. Um modelo realista do aluno implica em uma atualização dinâmica enquanto o sistema avalia o desempenho do estudante.

Os dados dinâmicos referem-se ao desempenho do aluno diante das questões formuladas pelo tutor e confrontadas com as hipóteses elaboradas pelo aluno. Os conhecimentos novos que o aluno traz podem o tutor. A partir disso, o tutor é capaz de interpretar o modelo do aluno

Muitas técnicas são utilizadas para construir o modelo do aluno. Listamos a seguir algumas delas:

- incluir um reconhecimento de padrões aplicados ao histórico das respostas fornecidas por ele;
- comparar a conduta do aluno com a de um especialista e verificar os pontos em comum;
- acrescentar as preferências do aluno;
- incluir seus objetivos particulares;
- observar as coisas que o aluno sempre costuma esquecer quando interage com o tutor;

O modelo do aluno, segundo [COS96], pode ser representado apoiando-se em alguns modelos de descrição, tais como:

- **Modelo diferencial** – consiste da comparação da resposta do aluno com a base de conhecimento. Esta modelagem, segundo [DAM95], compara a performance do especialista com a do estudante e não o conhecimento deles. A modelagem por diferenciação divide o conhecimento em duas classes: aquele que se espera do estudante e o outro que não se espera. Nesta modelagem o conhecimento do aluno é somente um subconjunto do conhecimento do especialista
- **Modelo de Overlay ou superposição** – consiste da representação do conhecimento do aluno como um subconjunto da base de conhecimento do sistema tutor. Isso implica na representação de conhecimento utilizada no modelo do aluno ser a mesma na base do domínio.
- **Modelo de Perturbação** - consiste em assumir que os erros do aluno são decorrentes da concepção errônea de algum conceito ou ausência dele. Este modelo também relaciona o modelo do aluno com a base de conhecimento do domínio. Foi proposto inicialmente por Brown e Burton [BRO82] e recebeu o nome de modelo BUGGY.
- **Modelo de Simulação** – consiste em colocar no ambiente um modelo de como o estudante pode ou deve comportar-se em determinada situação. Através deste

modelo é possível prever o comportamento futuro do estudante, ou seja, a resposta do aluno somente com base no seu comportamento durante a sessão de trabalho [ASA91].

- *Modelo de Crenças* - consiste em um conjunto de crenças refletindo o grau que supõe-se o estudante entender sobre um conceito em particular.

Em Inteligência Artificial, é comum chamar de "conhecimento" às informações que o sistema possui a respeito do domínio da aplicação. No entanto, uma noção epistemologicamente mais rigorosa do termo define "conhecimento" como sendo "crença verdadeira e justificada" [NIL87] (o que em geral não é o caso da informação armazenada em um sistema computacional). O "conhecimento" de um sistema especialista, por exemplo, é quase sempre sujeito a revisão e frequentemente não-justificado (em geral, a palavra do especialista é justificativa suficiente). Assim, a rigor, seria mais correto falar em "bases de crenças" ao invés de falar em "bases de conhecimento". Na maioria dos sistemas, essa diferença não tem conseqüências práticas, pois o comportamento do sistema não é afetado. No entanto, a distinção é fundamental para a modelagem do aluno.

- *Modelo de Agentes* - consiste em tratar o modelo do aluno como um sistema de crenças, desejos e intenções. A interação entre aluno e sistema tutor é uma interação entre dois agentes inteligentes (ou, pelo menos, dotados de algum comportamento cognitivo). Considerar o aluno como um agente implica em considerar o modelo do aluno como um modelo de agente. Esta consideração apresentará conseqüências para a estrutura do modelo. Em particular, o modelo do aluno enquanto um modelo de agente, deve incluir, de um modo ou de outro, três componentes: a base de crenças, a base de motivações e o modelo de inferência.

A base de crenças compreende os conhecimentos/crenças do agente sobre o domínio (ou domínios) da aplicação. Sua caracterização como crenças deriva naturalmente do fato de que todo o aluno é capaz de aprender através de processos não-montônicos. Salienta-se que toda a informação representada na base de crenças é, em princípio, passível de revisão.

Na base de motivações está representada tanto as intenções primárias do agente (seus objetivos básicos) como as intenções derivadas (obtidas a partir das intenções primárias, seja por conseqüência direta, ou pela interação das intenções com as informações presentes na base de crenças).

O modelo de inferência pode ser pensado como um conjunto de regras de inferência. Cada uma das regras modela um tipo de inferência ou mecanismo de aprendizagem disponível para o agente, podendo atuar tanto sobre a base de crenças como sobre a base de motivações. No primeiro caso (que é o mais comum), temos a transformação do conhecimento que o agente possui do domínio. No segundo caso, temos a representação da criação/alteração de planos e modificação das intenções. É interessante notar, no entanto, que esta arquitetura gera um terceiro caso: as regras de transformação se aplicam sobre o próprio modelo de inferência. Desta forma, seria possível modelar agentes com sofisticadas capacidades de auto-modificação (meta-raciocínio).

- O *modelo overlay* assume (implícita ou explicitamente) que os erros ou comportamentos anômalos do aluno são sempre devidos à ausência de alguma informação presente na base do domínio. Este pressuposto psicológico é excessivamente simplista, uma vez que muitos comportamentos incorretos

originam-se da presença de concepções incorretas na mente do aluno. Com base nisso, muitos sistemas adotam modelos de perturbação [SEL88].

- No *modelo de perturbação*, existe a base do domínio e uma biblioteca de erros típicos. O modelo do aluno dentro da perspectiva do modelo de perturbação incluirá, por sua vez, elementos da base do domínio e da biblioteca de erros. Estes modelos permitem um tratamento mais inteligente dos comportamentos incorretos do estudante; no entanto, ainda são limitados, principalmente pela estrutura da biblioteca de erros.

Segundo Viccari e Oliveira [OLI92;OLI94], supor que o modelo do aluno é uma base de conhecimento implica em supor que o aluno não possui concepções incorretas (o que não é razoável) e em supor que o sistema é capaz de inferir corretamente o estado cognitivo do aluno (o que não parece uma suposição realista; na melhor das hipóteses, o modelo do aluno é uma aproximação útil).

O comportamento lógico das concepções do aluno assemelha-se muito mais a uma lógica de crenças do que a uma lógica de conhecimento. As concepções estão sujeitas a revisão, uma vez que podem ser inconsistentes (por exemplo, o aluno pode dispor de uma teoria inadequada). Além disso, o modelo do aluno é na verdade um conjunto de crenças aninhadas (crenças do tutor a respeito das crenças do aluno). Este conjunto está sujeito a revisão, pois não somente as hipóteses do tutor podem estar incorretas, como também o próprio estado cognitivo do aluno pode mudar com o tempo, em consequência da interação entre aluno e tutor.

Quando projetamos programas educacionais baseado em princípios gerais de aprendizagem e em recursos para suportar diferentes tipos de aprendizagem, os estilos individuais de aprendizagem podem ser contemplados. Para tanto, precisamos explorar os diferentes estilos individuais de aprendizagem e explorá-los através de várias estratégias de ensino. Neste sentido, para se poder adotar um modelo personalizado do estudante (abordagem baseada no perfil do aluno, proposta por Bloom [BLO83]), teremos de trabalhar com um modelo mais sofisticado do aluno.

O que dispomos atualmente são de modelos muito mais empíricos do que realmente gostaríamos de utilizar. A modelagem do aluno através de seus estados mentais proporciona um refinamento da coreografia destes estados mentais associadas com as ações dos alunos. A forma de captar fidedignamente estes estados mentais ainda é uma questão em aberto. Sabe-se que os diálogos entre os alunos são fontes extremamente ricas de dados sobre o processo de organização da mente. Por esta razão, os estados mentais apresentados neste trabalho foram extraídos de diálogos entre alunos em laboratório utilizando a versão preliminar do ambiente de teste (maiores detalhes serão abordados nos capítulos 4 e 5).

As lacunas e mal entendimentos em tarefas de aprendizagem construtivas podem acontecer durante o processo. Neste sentido, o diálogo pode ajudar os estudantes a reconhecer os elementos que compõem um problema e as diferentes formas de resolvê-lo. É provável que a interação de diálogo seja mais efetiva quando um dos interlocutores (um mais maduro e experiente) saiba favorecer a interação. De acordo com Vygotsky [VYG84], a função do tutor é incentivar o estudante a desenvolver representações mais abstratas e especializadas. Não podem esperar que os estudantes gerem os próprios padrões, uma vez que eles estão em fase de construção do seu conhecimento. O tutor auxiliaria o aluno a organizar suas próprias idéias, mas sempre observando as características individuais de cada aluno.

Segundo [SAL91], falar de aprendizagem equívale a destacar o processo de construção de significados como elemento central do processo de ensino/aprendizagem. O aluno aprende um conteúdo qualquer (um conceito, uma explicação de um fenômeno, um procedimento para resolver uma equação, etc.) quando é capaz de atribuir-lhe um significado. No sentido estrito, o aluno também pode aprender estes conteúdos de uma forma puramente memorística e ser capaz de repeti-los ou de utilizá-los mecanicamente sem entender em absoluto o que está dizendo ou fazendo. O que geralmente acontece é que o aluno atribui significados parciais ao que aprende. Isto não significa que o professor possua os mesmos significados. Ou seja, o que o professor ensinou não necessariamente corresponda ao que o aluno aprendeu. Ambos não têm a mesma compreensão sob o apreendido. Isto quer dizer que o significado da aprendizagem não é uma questão de tudo ou nada e sim de grau de apropriação do conhecimento. Em vez de querermos que os alunos atinjam aprendizagens significativas, talvez fosse mais adequado, segundo [SAL91], tentaremos tornar as aprendizagens mais significativas em função do grau de escolaridade.

Segundo Ausubel e seus seguidores [AUS80], a aprendizagem humana classifica-se através de duas distinções: a aprendizagem por recepção versus aprendizagem por descoberta por um lado, e a aprendizagem automática versus aprendizagem significativa por outro. Na aprendizagem por recepção, o aluno recebe o conteúdo a ser aprendido e deve interiorizá-lo ou incorporá-lo ao seu repertório pessoal. A aprendizagem receptiva corresponde a grande parte do ensino proporcionado aos estudantes nas escolas. Essa aprendizagem receptiva verbal não é necessariamente mera memorização, sem compreensão do significado. Na verdade, a aprendizagem verbal significativa é o meio principal de aquisição de grande parte de nossos conhecimentos.

O processo que envolve a interação de uma nova estrutura de conhecimento, é definido por Ausubel como baseado em conceitos subsunçores ou apenas *subsunçores*. Os subsunçores são os elementos de que o aluno dispõe. O aluno baseia-se neles para formar uma nova estrutura de aprendizagem.

A proposta de Norman [NOR85] baseia-se num conceito semelhante às redes de subsunçores de Ausubel. Para este autor, o conhecimento humano é armazenado na forma de redes semânticas. Neste sentido, o aluno ao estudar um novo conhecimento faz uma varredura nas suas estruturas mentais e procura alguma estrutura onde um conjunto de conceitos que compõem uma determinada informação possa ser duplicado. Sendo a estrutura adequada para representar o novo conhecimento, os nodos desta rede semântica recebem as novas informações. Caso o aluno não possua nenhuma estrutura que sirva para armazenar o novo conhecimento, ele deverá criá-la através da repetição. A repetição provocaria uma reação em nível fisiológico e mental para armazenar novas informações. Cabe salientar que a repetição não implica na utilização da mesma estratégia ou técnica para provocar este efeito. A diversidade de estratégias e táticas é desejável para que se amplie a eficácia do processo de ensino-aprendizagem.

A importância de termos um modelo qualitativo para o aluno reflete exatamente estas questões pedagógicas abordadas acima. A busca deste modelo vai ao encontro do tipo de trabalho que se deseja ao mudarmos do paradigma tradicional (ênfase no ensino centrado no professor) para o paradigma centrado no aluno (aprender a aprender).

A mudança de paradigma implica em uma nova concepção dos programas educacionais (especialmente os STI). Conforme já mencionado anteriormente, a parte mais fraca dos STI ainda se manifesta no modelo do aluno. Isto acontece devido à inexistência de um modelo formal que expresse os processos mentais. As máquinas

(computadores) que dispomos nos dias de hoje precisam dispor deste modelo mais formal de processos mentais.

2.3 O comportamento do tutor nos STI

Um STI concebido na abordagem construtivista não pretende deixar o estudante sem qualquer supervisão (auxílio). Pretende sim orientá-lo sem desconsiderar o seu conhecimento prévio e as tomadas de decisão realizadas ao longo da sessão de trabalho, entre várias formas, na forma de uma mensagem (conteúdos diferenciados), de avisos sonoros (indicador de situação crítica), de exemplos, de apresentação de uma regra.

A seleção do conjunto de estratégias de ensino que constituirá o STI é um aspecto muito importante para garantir a qualidade pedagógica do ambiente. A seleção de uma estratégia depende de vários fatores, tais como: o nível de conhecimento do estudante, o domínio, a motivação e as características afetivas do mesmo. A utilização da mesma estratégia não produz um efeito satisfatório para todos os estudantes. Segundo Frasson [FRA97], cada estratégia têm vantagens específicas. É muito útil saber qual a estratégia que fortalecerá adequadamente o processo de aprendizagem para um determinado estudante. Bons professores apresentam o mesmo material de formas diferentes. As situações em que isto pode ocorrer não são previsíveis (até mesmo se o domínio for previsível). O que se pode fazer é procurar diversificar o auxílio oferecido ao aluno considerando os indicadores retirados do próprio trabalho pregresso do estudante.

A construção de um STI com múltiplas estratégias é muito útil para aumentar a qualidade pedagógica dos ambientes. No entanto, a aquisição de conhecimento sobre multi-estratégias pedagógicas de forma separada da aquisição de perícias de domínio, ainda é um grande desafio para os projetistas de STI [VAS97; MEN98]. As estratégias pedagógicas dispõem do conjunto de regras e/ou planos para alcançar metas específicas. Ohlsson [OHL87] e Keller [KER87] sugeriram que o tutor deve ter dois níveis de planeamento: táticas pedagógicas e estratégias pedagógicas.

As estratégias pedagógicas têm o conhecimento sobre como ensinar e as táticas pedagógicas contêm as ações para efetivar a estratégia selecionada. As estratégias inteligentes usadas em STI são baseadas em regras explícitas por carregarem decisões instrucionais.

Os STI podem ser classificados em três diferentes grupos: behaviorista (controle estabelecido pelo tutor, exemplo [CAR70; BUR82]), ambientes cognitivos (controle em poder do estudante [FRA97; MEN98; MOU96; GIR98]) e ambientes mistos (o controle se alterna entre o sistema e o estudante [VIC90]).

O ponto de ruptura entre os grupos é o modelo do estudante. O modelo de estudante projetado é que guiará o comportamento dos STI. Daí a importância no tratamento do modelo do aluno. Quanto maior for o conjunto de informações (quantitativas e qualitativas) que o sistema possui sobre o aluno, melhor será o seu comportamento.

As estratégias usadas em STI podem ser classificadas nas seguintes categorias:

- *Socráticos* - o material apresentando é projetado para induzir os estudantes a identificarem enganos e interpretações errôneas a cerca do conteúdo. A estratégia socrática utiliza uma seqüência de perguntas encadeadas baseadas no erro do

estudante. Esta estratégia se ajusta melhor às tarefas de aprendizagem conceituais ou procedurais.

Exemplos:

SCHOLAR [CAR70], primeiras versões do SOPHIE [BRO78; BRO82], e WHY [STE77].

- *Reativos* - as lições reagem às perguntas do estudante e apresentam hipóteses que simulam os efeitos das idéias do estudante, apresentando as respectivas implicações baseadas em regras prévias. O conhecimento deriva da simulação dos trabalhos do estudante no sistema.

Exemplos:

SOPHIE [BRO82], WEST [BUR79], WUMPUS [GOL82], SPADE [MIL82], GUIDON [CLA82], ADVISOR [GEN77], STEAMER [STE77] and PERSONA [AND97];

- *Treinamento (Coaching)* - o sistema utiliza um conjunto de regras de produção para escolher a forma mais apropriada de instrução para um determinado estudante. As estratégias podem incluir orientação para tarefas novas, transições entre tópicos e interação com o estudante para achar opções melhores no contexto. Estes ambientes podem ser mais diretivos dependendo do projetista e das suas convicções educacionais teóricas. Segundo Kearsley [KEA87], um STI treinador observa o desempenho do estudante e provê conselhos que ajudarão o estudante a executar melhor uma determinada atividade. A resolução de problemas tipo simulações e jogos são mais adequados para ambientes orientados.

Exemplos:

BUGGY [BRO78], WEST [BUR79], WUMPS [GOL82], SPADE [MIL82], COACH [GEN77], ALGEBRA [LAN83], TUTOR PROLOG [VIC90], ELETRO-TUTOR [SIL92], and PERSONA [AND97].

- *Colaborativo (ajuda)* - o tutor age como um participante em uma conversação com estudantes e auxilia-os a clarificar suas idéias utilizando suas próprias perícias em momentos chaves. Trabalha de forma semelhante a um treinador, mas de fato é um perito que não insiste em suas perícias. O sistema tem menos controle do que as ações da categoria anterior, mas ainda tem algum tipo de controle sobre a seção de trabalho com o aluno, especialmente em jogos educacionais. Segundo Kearsley [KEA87], o tutor “depura” o trabalho do estudante. Eles são adequados para ambientes com ênfase em solução de problemas.

Exemplos:

QUEST [WHI86], WUMPUS [GOL82] e PROUST [SOL84].

Muitos dos sistemas citados na literatura apresentam mais de uma característica da classificação apresentada acima. Isto acontece porque os projetistas dispõem de várias metas para construir o sistema. Eles consideram os princípios instrucionais diversificados, os métodos diferentes para estruturação do conhecimento, os vários formatos instrucionais para áreas de assuntos diferentes. É muito raro achar um STI puro na sua concepção pedagógica, isto é, com apenas um enfoque. Isso acontece por causa da natureza interdisciplinar destes ambientes (que contemplam as áreas de Ciência da Computação, Psicologia Cognitiva, e Educação) e devido a heterogeneidade

do processo de ensino-aprendizagem. É um processo dinâmico que altera situações de maior e menor controle por parte do tutor.

Dos pontos de vista educacional e psicológico, nós temos que pensar como os sistemas dirigirão as interações de ensino-aprendizagem e como serão feitas as sessões entre o tutor e o estudante. A seção seguinte discute a relação entre estas duas áreas e suas implicações para o projeto de um STI.

2.4 A questão das múltiplas estratégias de ensino em STI

A evolução dos STI contempla vários aspectos, entre eles, a estruturação e a representação do conhecimento (para melhorar a comunicação com o estudante), a utilização de estratégias diversificadas para domínios específicos e as melhorias na modelagem do estudante (ênfase nos aspectos qualitativos).

No final dos anos 80, foi dada mais ênfase aos aspectos pedagógicos e foi incrementada a participação do estudante no processo de ensino-aprendizagem. Esta nova geração de STI rompeu com a tradicional abordagem behaviorista contida nos primeiros STI e abriu caminho para a construção de STI com uma abordagem construtivista.

A arquitetura destes sistemas começou a mudar com esta nova abordagem e foi reforçada pelo uso das técnicas de agente para sua modelagem e implementação. O uso de arquiteturas multiagente permite, aos projetistas, criar STI onde a abordagem tradicional (módulos funcionais) foram substituídos por agentes ou sociedade de agentes [BEC98;COS92;COR94;COS96; DIL95;DAM96; FRA97;LUC98;MOU96; MUR97]. Contudo, estes sistemas ainda mantêm a arquitetura básica. Eles têm uma base de domínio, um modelo de estudante, um módulo de ensino, e um controle distribuído (em maior ou menor grau).

Os ambientes que se utilizam de agentes apresentam duas estratégias básicas de aprendizagem: o aluno (agente humano) aprende com o auxílio de um colega (agente artificial) para resolver problemas que parecem durante a interação; e o aluno (agente humano) aprende através da atividade de ensinar outro agente artificial.

Aimeur e Frasson [AIM96; SHI98] propuseram uma terceira estratégia denominada de aprendizagem por perturbação, em que o aluno é desafiado por um agente artificial (TroubleMaker⁹) a resolver os questionamentos colocados por este agente artificial.

As características destas três estratégias são:

- *Aprendizagem com um colega* - a idéia é apresentar um colega (agente artificial) no sistema que discutirá as tarefas com o aluno e o auxiliará na percepção do conhecimento fazendo com que o resultado da interação seja mais um processo de construção do que um processo de transmissão de conhecimento. Neste âmbito, o estudante poderá cooperar com um colega que tem objetivos e nível de conhecimento bastante semelhantes ao seu. O conhecimento de cada agente é diferente e o tutor tem um conhecimento maior do que o conhecimento do colega, e este, por sua vez, um conhecimento maior do que o aluno. O papel do tutor é

⁹ Encenqueiro

alternar problemas presentes e fazer críticas às soluções apresentadas pelo estudante. O colega artificial, apresentado ao estudante, vai diluir a carga normalmente atribuída ao agente tutor, fazendo com que parte dos questionamentos e das tarefas de ensino sejam realizadas pelo colega artificial. Desta forma, o aluno não sente tanto a presença e a orientação do tutor. Esta situação se assemelha mais ao tipo de parceria que normalmente se estabelece numa situação regular de aula.

- *Aprendizagem através do ensino*- a idéia é pôr um estudante humano para ensinar o companheiro de aprendizagem (agente artificial) provendo exemplos e explicando porque a solução dada pelo companheiro não é adequada. Esta aproximação está baseada na teoria de Gagne [GAG96] que afirma que a aquisição de conhecimento é alcançada quando um estudante pode explicar a solução de uma tarefa. O aluno utiliza seu próprio mecanismo de conclusão baseado na sua forma individual de aquisição de conhecimento. Ao explicar para o colega como ele resolve o problema, ele necessitará estar mais consciente da forma da solução deste e não somente basear-se na sua intuição. O aluno terá de elaborar a resposta num decodificar e re-codificar de informações que fará com que ele consiga organizar o seu pensamento, ampliando não só o seu conhecimento mas a forma de manipular este conhecimento.

- *Aprendizagem por perturbação* - três agentes (tutor, um encrenqueiro, e um estudante) compartilham o mesmo espaço de aprendizagem. O nível de competência do encrenqueiro é superior ao do estudante para proporcionar uma competição razoável ao estudante. Um problema é submetido ao estudante e ao encrenqueiro. O encrenqueiro pode ter comportamentos diferentes (dar respostas erradas, oferecer a solução certa, dar sugestões, e assim por diante). O estudante explica sua opinião ao encrenqueiro sob o controle do tutor. Se o comportamento do estudante mostrar que está impossibilitado de apresentar uma alternativa correta, o tutor lhe dá a solução correta. O encrenqueiro funciona como um elemento questionador que pode estar colaborando com o aluno ou não. O fato do aluno não saber qual a intenção do encrenqueiro faz com que ele tenha de analisar as intervenções deste agente . O aluno compara as repostas e verifica qual está mais adequada para o problema proposto. O encrenqueiro, ao mesmo tempo que desestabiliza o aluno com possíveis intervenções erradas, auxilia-o a pensar sobre alternativas de solução que poderiam ocorrer para a resolução do problema.

Consideramos esta possibilidade do aluno trabalhar com a assistência diferenciada por parte do tutor e com a colaboração de um outro colega humano, como uma quarta possibilidade para construir os STI em uma abordagem cooperativa. Cabe salientar que, neste trabalho, a cooperação ocorre fora do controle do sistema. Os alunos trocam mensagens e o tutor vai perceber o resultado disto através do conjunto de estados mentais apresentados pelos alunos. Neste momento o grau de complexidade e a quantidade de questões em aberto (falta de modelo coletivo para grupo, resolução de conflitos nas negociações entre os agentes pedagógicos oriundos das contradições em seus estados mentais e outros) implicariam em outro trabalho. Futuramente, pretendemos trabalhar com estas questões através da análise de modelos de grupo, onde as questões referentes ao processo de negociação e resolução de conflitos podem ser mensuradas.

A utilização de multi-estratégias adicionou um grau de complexidade maior ao tutor. O fato de o tutor utilizar um conjunto de estados mentais associados às ações de cada

aluno fornece o desenvolvimento de um trabalho com orientação personalizada. Além disso, este avanço implica em um projeto mais sofisticado da interface

Este capítulo tratou dos aspectos ligados à Informática na Educação necessários para melhor contextualizar o trabalho da tese e oferecer uma visão geral do estado da arte da área de STI. O capítulo a seguir trata dos STI modelados com agentes, onde serão abordados os aspectos referentes à IAD e à SMA.

3 STI modelado através de uma arquitetura multiagente

A modelagem de tutores inteligentes como sistemas multiagentes já vem sendo pesquisada, como demonstram os trabalhos de [COS97], [DAM99],[DIL94;DIL95] e [MOU96]. O uso de agentes tutores em ambientes de realidade virtual também pode ser verificado em [SOU97], onde o agente tenta se passar por ser humano (tendo sido, inclusive vencedor do concurso “Teste de Turing”), agindo como uma espécie de fonte de informações em um micromundo textual em rede (MUD). Ele é capaz de dar informações sobre os diversos *locais* do MUD, e sobre outros visitantes que lá estiveram. O agente é, inclusive, capaz de lembrar de conversas anteriores e citá-las a quem quiser.

Esta nova visão de STI traz algumas vantagens evidentes sobre a abordagem tradicional de considerar a existência de apenas um tutor no ambiente de aprendizagem. Em especial, citamos como desvantagens dos sistemas tradicionais os seguintes pontos:

- tutor detém todo o conhecimento, que precisa ser programado nele previamente;
- aprendiz vê o tutor como detentor de todo o conhecimento. O aprendiz em geral não se sente a vontade para questionar o tutor e, assim, desenvolve pouca autonomia [SOU97];

Em relação a estes problemas, a abordagem multiagentes provê as seguintes vantagens:

- conhecimento pode ser distribuído entre vários “tutores”, cada um com suas crenças, desejos, objetivos, emoções e planos de ação. Esta distribuição cria maiores oportunidades de variar técnicas pedagógicas. Possibilita ao aprendiz escolher o tutor que mais lhe convém e ao projetista criar uma sociedade de agentes tutores que se comunicam entre si e determinam quem é o mais apto a realizar determinada tarefa ou auxiliar determinado perfil de aluno [COS97];
- aprendiz interage com um tutor de forma mais flexível, do jeito que como interagiria com outro aprendiz. O tutor é apenas mais um personagem na história [AND97; AIM.97;GIR98a;GIRb;GIR98c;GIR98d];
- aprendiz pode passar conhecimentos ao tutor que serão repassados a outros aprendizes. Ele também pode passar este conhecimento diretamente aos outros aprendizes.

Segundo [SOU97], esta abordagem possivelmente elevará os níveis de autonomia do aprendiz, levando-o a questionar as proposições do tutor e/ou de outros aprendizes. Melhores níveis de autonomia também contribuirá para que se consiga uma real melhoria nos aspectos qualitativos dos STI, principalmente no que concerne aos aspectos pedagógicos normalmente muito restritos com o uso das técnicas convencionais.

Este capítulo tem por objetivos apresentar as definições adotadas neste texto no que tange aos conceitos relacionados com os sistemas multiagentes e apresentar uma descrição geral dos ambientes utilizados nos estudos preliminares realizados para compor a arquitetura multiagente. Serão apresentados em detalhes a arquitetura proposta para a modelagem de um STI na abordagem MAS, a arquitetura dos agentes

reativos e, uma idéia geral da modelagem dos agentes cognitivos (abordados posteriormente nos capítulos 4 e 5).

As definições e conceitos são muito necessários a fim de posicionar o trabalho diante da não uniformidade destes na área de IAD.

3.1 IAD e agentes

A IA clássica (origem psicológica) tem seu estudo centrado no comportamento individual do ser humano. Os pesquisadores buscam respostas para a melhoria dos seus sistemas no modelo de inteligência do indivíduo, utilizando os diversos pelos métodos de representação do conhecimento e de inferência. Os STI projetados nesta abordagem refletem exatamente estas características [CAR70; BRO78; BUR79; CLA82; STE77].

A IAD (origem sociológica) utiliza um modelo de inteligência baseado no comportamento social, com ênfase nas ações e interações de agentes que podem ser entidades reais ou virtuais imersas num ambiente sobre o qual são capazes de agir [ALV97]. A IAD se divide em duas áreas, são elas: resolução distribuída de problemas e sistemas multiagentes.

O termo agente é uma noção central e fundamental para área de IAD. Este termo vem sendo utilizado para denotar simples processos de hardware e/ou software até entidades sofisticadas capazes de realizar tarefas complexas. Esta diversidade reflete o estado atual da área, onde não se possui um consenso do que realmente um agente é. Cada grupo de pesquisa segue uma determinada linha apresentando sua definição personalizada do termo agente de acordo com seus próprios objetivos. Ao fazermos um extenso levantamento de todas as tendências da área [GIR98h], utilizamos como referencial o trabalho de Russell & Norvig [RUS95], que definem um *agente* como um sistema capaz de perceber através de sensores e agir em um dado ambiente através de atuadores. Esta definição é enriquecida através da descrição de um conjunto de propriedades usualmente aceitas e que os agentes devem exibir.

As propriedades básicas de um agente são:

- Autonomia - escolhe a ação a tomar baseado mais na própria experiência do que no conhecimento embutido pelo projetista. As ações do agente não requerem interferência humana direta;
- Temporário ou contínuo – opta por permanecer ou não no ambiente;
- Comunicativo - troca informações com o ambiente e com os outros agentes;
- Aprendizagem - aprende com as informações oriundas do ambiente ou de outros agentes;
- Mobilidade - capaz de se deslocar para ambientes diferentes do original;
- Flexibilidade - aceita a intervenção de outros agentes;
- Proatividade - capaz de, além de responder a estímulos do ambiente, exibir um comportamento orientado a objetivos. Ou seja, ser capaz de prever como atingir ou evitar um determinado estado ou objetivo. Pensar no futuro, antecipar, agindo em função de prever.

O ambiente e seu respectivo projeto (design) irá determinar as propriedades que o agente terá. Além destas propriedades básicas podemos incluir a *habilidade social* (via algum tipo de comunicação entre agentes), a *reatividade* (reage a estímulos recebidos de outros agentes ou do ambiente), a *racionalidade* (dependente do grau de sucesso e percepção que o agente possui), *adaptabilidade* (capacidade de se adaptar a modificações no ambiente, alterando planos previamente concebidos ou aprendendo através da interação com o ambiente) e *representação através de personagens* (tanto em 2D, 3D e através de avatares¹⁰).

Recentemente na literatura apareceu a noção de *agentes pedagógicos* devido a muitos sistemas desenvolvidos para fins educacionais adotarem o paradigma de agentes. Estes agentes incorporam muitas características dos agentes utilizados em ambientes lúdicos (jogos).

Agentes pedagógicos podem ser divididos em *goal-driven*¹¹ (tutores, mentores e assistentes) e *utility-driven*¹² (MOO¹³ e agentes da WEB).

Os agentes *utility-driven* são utilizados como agentes que realizam tarefas auxiliares ligadas às atividades pedagógicas, i.e., entre outras atividades, ajudam os estudantes a encontrar programas específicos, arquivos, diretórios, a agendar encontros de grupos, a lembrá-lo de atividades a serem entregues como tarefas de classe e outras. Estes agentes executam tarefas para o estudante ou para o tutor.

Os agentes *goal-driven* na WEB possuem mobilidade e operam em diferentes contextos (textuais, hipermídia, realidade virtual, por exemplo). A interação acontece baseada no processo cooperativo ou competitivo no qual agentes artificiais e agentes humanos se comunicam e realizam atividades.

A principal razão de se introduzir agentes para projetarmos STI e ILE reside nas suas propriedades que permitem ganho de qualidade sob o ponto de vista pedagógico, como por exemplo a proatividade, a habilidade social e a flexibilidade. Incorporar agentes a um programa educacional é intensificar os aspectos pedagógicos desejáveis no ambiente.

Segundo Vassileva [VAS97], um agente pedagógico tem de agir num mundo onde existam outros agentes, porque muitos destes agentes têm objetivos que requerem a ajuda de outro agente. Este relacionamento entre os agentes pode ser visto como uma nova fonte de recursos para atingir os objetivos.

Os agentes pedagógicos podem atuar como tutores virtuais, estudantes virtuais ou companheiros virtuais de aprendizagem. O objetivo destes agentes é auxiliar os estudantes no processo de aprendizagem.

¹⁰ Avatar é um ícone gráfico que representa uma pessoa real num sistema do espaço virtual.

¹¹ Goal-driven : guiados por objetivos

¹² Utility-driven: guiados pela sua utilidade no ambiente

¹³ Inicialmente conhecidos como *Multi User Dungeon*, os MUDs são ambientes virtuais acessíveis via rede de computadores, onde pessoas e objetos interagem por meio de uma interface baseada em textos conectada a uma base de dados compartilhada que lhes permitem executar jogos. Os MOOs são uma evolução dos MUDs. Utilizam a orientação a objetos associada a uma poderosa linguagem interna de programação como forma de criação de objetos e ações mais sofisticadas entre os seus usuários. Os participantes de um MOO são conhecidos como jogadores. Eles devem conectar-se ao ambiente através de *Telnet* ou algum outro *software* cliente especializado. [DIL97].

As propriedades fundamentais dos agentes pedagógicos são as seguintes: autonomia, habilidade social, proatividade e persistência. Estes agentes não se demovem dos seus objetivos e continuam atuando no ambiente. Além destas propriedades, os agentes pedagógicos podem ser reativos, de performance contínua, serem capazes de aprender e, na maioria dos casos serem representados por um personagem. Estas propriedades básicas foram identificadas através da análise minuciosa de publicações científicas na área de AIED e detalhadas em [GIR98h].

3.2 STI modelados através de uma sociedade de agentes

O estudo das sociedades de agentes em IAD tem assumido grande importância. Segundo [EUS95], as vantagens apontadas para este tipo de sistema são as seguintes: a *modularidade* (facilidade na construção de um sistema formado por módulos quase independentes), a mudança da arquitetura nos computadores (o computador tende a ser formado, não por um único processador com uma grande memória, mas por um grupo de processadores, cada um com a sua própria memória, o que implica uma melhor adaptabilidade destes sistemas), os raciocínios heterogêneos (o problema pode ser formado por vários subproblemas, aos quais correspondem diferentes formalismos na representação do conhecimento), as perspectivas múltiplas (muitos problemas são mais facilmente resolvidos se podem ser vistos segundo várias perspectivas), os problemas distribuídos (os problemas podem ter os seus dados disponíveis em várias localizações físicas distintas) e a confiança (a resolução de um problema pode continuar, ainda que haja uma falha da parte de um dos agentes).

Segundo [DAM95], as limitações do estilo “paternalista” da ação tutorial encontradas nas implementações de STI que utilizam o modelo tradicional se devem ao fato de que as interações são prerrogativas do sistema. Tais limitações têm feito com que a pesquisa na busca de ambientes mais cooperativos seja objeto de estudo na área de IA-ED. Nos últimos anos, pesquisadores têm projetado STI para se ajustar às necessidades de cada agente humano que deseja aprender. Os tutores são projetados para considerar não apenas o comportamento do agente cognitivo, mas também o comportamento como parte de um sistema cooperativo. O sistema muda de maneira a se ajustar às capacidades do agente-aprendiz e, como consequência disto, a aprendizagem melhora.

A modelagem de STI numa arquitetura funcional de agentes é mais do que uma abordagem generalista. Teoricamente a modelagem não apresenta limites para o número de agentes que podem participar do processo de aquisição do conhecimento. Processo este que ocorre através de negociação dos papéis dos agentes tanto tutores, como aprendizes. Portanto, a modelagem de STI situa-se em uma abordagem mais restrita.

As propostas de utilização de arquiteturas MAS em STI trazem uma grande vantagem em relação as arquiteturas tradicionais de STI: apresentam uma flexibilidade maior no tratamento dos elementos que compõem o sistema [GAG96],[LEM96]. Além disso, o fato de usarmos agentes para modelar os seus componentes possibilita o agrupamento da arquitetura tradicional (um módulo = um agente) ou na explosão de cada módulo. Neste último caso, o refinamento pode chegar até os estados mentais de um agente [MOU96].

Para delinear nossa pesquisa acerca de STI em arquitetura MAS, escolhemos alguns ambientes recentemente citados na literatura.. Estes ambientes são detalhados nos itens a seguir e o critério de seleção deu-se em função das suas características e das suas contribuições para o nosso trabalho. Ao final da descrição de cada ambiente, ressalta-se o tipo de contribuição que este trabalho ofereceu para os nossos estudos que definiram a arquitetura proposta nesta tese.

3.2.1 O ambiente MATHEMA

O MATHEMA, proposto por [COS96], é um ambiente interativo concebido para apoiar atividades que venham a favorecer um ensino adaptativo e seus desdobramentos no processo de aprendizagem. Trata-se de um ambiente que propõem uma nova abordagem no suporte à concepção de ILE, configurado como uma solução alternativa e efetiva no problema de adaptação.

O ensino (adaptativo) é aqui visto como consequência do processo de interações cooperativas envolvendo os seus componentes (aluno e tutor) em atividades baseadas em resolução de problemas. Segundo Costa [COS97], a noção de problema significa uma situação na qual, a partir de certas condições iniciais ou estado inicial do problema, tenta-se atingir uma dado objetivo através de um conjunto de ações. A aprendizagem é favorecida e decorre de atividades provenientes do ensino adaptativo. Assumimos que aprendizagem significa aquisição de conhecimento.

O MATHEMA foi concebido numa estrutura definida para suportar técnicas e funcionalidades de IAD através de uma abordagem baseada nos SMA. O modelo de agente adotado investiga um possível aprendiz humano em uma sociedade de agentes tutores artificiais, visando envolvê-lo numa situação de aprendizagem.

A arquitetura do MATHEMA é composta de seis componentes, descritos a seguir de forma sucinta:

- Aprendiz humano - agente interessado em aprender algo sobre um determinado domínio trabalhado no ambiente MATHEMA. Ele desempenhará essencialmente o papel de uma agente ativo envolvido em atividades baseadas na resolução de problemas num domínio de aplicação, sob a assistência especializada de um agente tutor (imerso numa sociedade de agentes cooperativos);
- Motivador externo - entidades humanas externas que desempenham o papel de motivadoras do aprendiz para trabalhar no MATHEMA. Alguns motivadores podem ser o próprio professor ou seus próprios colegas;
- Sociedade de agentes tutores artificiais (SATA) - conjunto de agentes (ou micro-agentes) que algumas vezes cooperam entre si a fim de promover a aprendizagem de um dado aprendiz na atividade de ensino. Cada um desses agentes é especializado em subdomínios contidos em um dado domínio do conhecimento. Esta idéia é inspirada no trabalho de MINSKY [MIN87], em que a inteligência emerge da combinação de agentes mentais, cada um responsável por um pequeno processo;
- Sociedade de especialistas humanos (SEH) - fonte de conhecimento externa ao sistema computacional (um agente que funciona como um oráculo) para a SATA. Dessa sociedade depende a manutenção da SATA (inclusão, exclusão, alterações

no conhecimento dos agentes) e mais a assistência aos aprendizes no caso da falha da SATA. A SEH pode analisar o desenvolvimento das interações entre o aluno enquanto a SATA pode avaliar seu desempenho e promover melhorias sempre que necessário, independente de ter sido ou não requisitada pela SATA.

- Agente Interface (AI) - elo de ligação entre o aprendiz humano e a SATA.
- Agente manutenção (AM) - elo de ligação (interface) entre a SEH e a SATA, encarregando-se de prover a interação entre elas.

As flechas mostradas na figura 2.1 representam os relacionamentos interativos entre os componentes do MATHEMA. Baseado nesses relacionamentos, será dada a seguir uma visão geral do cenário de funcionamento das interações (mais de um ponto de vista abstrato e externo) que podem ocorrer no MATHEMA.

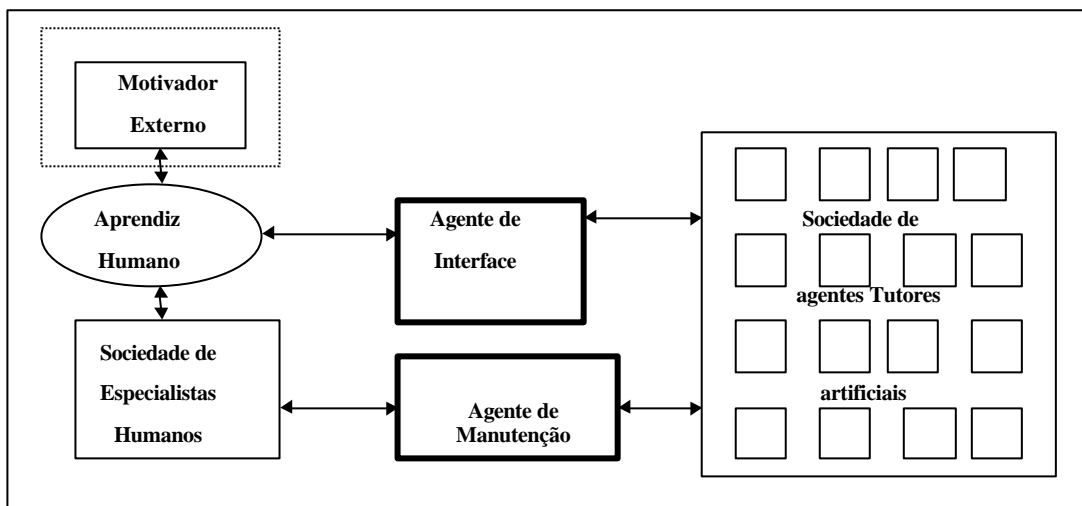


FIGURA 2.1 - Arquitetura do MATHEMA

O funcionamento do MATHEMA se inicia com um dado Aprendiz Humano, supostamente incentivado por um Motivador Humano Externo que começa a interagir com o Agente Interface. Nesse caso, o AI apresenta ao Aprendiz algo sobre o ambiente computacional de ensino e ajuda, mediante uma análise de seus objetivos ao escolher seu Supervisor em SATA. A partir daí, inicia-se um processo de interação cooperativa e didática entre o Aprendiz e o Agente Supervisor. O Agente Supervisor será então o responsável pelo Aprendiz, oferecendo-lhe a orientação necessária à promoção do seu aprendizado. Durante esta interação, situações com diferentes complexidades podem ocorrer. A mais elementar é a que envolve apenas a interação entre o Aprendiz e o seu Supervisor. A complexidade, entretanto, começa a aumentar à medida em que a demanda do Aprendiz passa a requerer a participação de outros agentes, além do seu Supervisor. Isso poderá evoluir a uma situação extrema na qual os agentes em SATA não conseguirão atender à requisição do Aprendiz. Nesse caso, o Agente Supervisor informa o aprendiz sobre a impossibilidade em atendê-lo, pelo menos momentaneamente, informando que irá consultar sua entidade oracular e agendando um novo horário. Assim, ele informa a SEH sobre o ocorrido, através do Agente de Manutenção. A SEH, tendo resolvido o problema, informa o agente supervisor da sua

disponibilidade em interagir com o Aprendiz. Além disso, caso julgue necessário, SEH realiza uma operação de manutenção em SATA.

A componente SATA é constituída por um conjunto de agentes (micro-agentes) que podem cooperar entre si através de um protocolo, a fim de promover a aprendizagem de uma dado estudante, numa dada situação de resolução de problemas. Esta sociedade é definida em função de um domínio de conhecimento, que por sua vez é organizado em uma série de subdomínios. A cada subdomínio está associado um micro-agente. Para trabalhar essa visão, foram feitas algumas reflexões e considerações sobre o domínio de conhecimento.

Foi definido um esquema de organização do conhecimento em que o domínio de conhecimento é dividido em diferentes subdomínios, preferencialmente independentes. Esta divisão está comprometida com micro-especialidades distribuídas em três dimensões de conhecimento: uma para profundidade (que diz respeito ao refinamento na linguagem de percepção), outra para amplitude (lateralidade) e a última, composta por diferentes pontos de vista sobre o domínio de conhecimento e sobre o domínio pedagógico. Em SATA, o domínio associado a um ponto de vista particular origina uma série de micro-agentes comprometidos com as dimensões de profundidade e lateralidade. Essas três dimensões são o alicerce para se buscar um compromisso entre uma boa estruturação e riqueza de conhecimento. Com isso e mais um mecanismo de interação "poderoso", pretende-se dispor de um ambiente mais adequado para promover aprendizagem cooperativa e, assim, conseguir um maior potencial de adaptabilidade do Tutor a um estudante em particular. A idéia; portanto, é combinar estes fatores em tal ordem que venham a favorecer o surgimento de um comportamento adaptativo na interação com o Aprendiz.

Cada agente é uma entidade inteligente em algum subdomínio dispondo de conhecimento necessário para ensinar e, particularmente, resolver problemas nestes subdomínios. Ele pode também contribuir para um comportamento social através de suas capacidades. A cada agente estão associadas as seguintes capacidades:

- Resolução de problemas - um agente deve não somente propor problemas ao aprendiz, como parte do processo interativo inerente, mas também deve resolver ou ajudar a resolvê-los.
- Aprendizagem - cada agente possui três modalidades de aprendizagem: a primeira diz respeito à possibilidade dele aprender com a SEH, a segunda considera a possibilidade dele aprender com o aprendiz (por exemplo, aprender novas soluções para um dado problema) e sobre o aprendiz (constituindo-se no que se denomina Modelo do Estudante), e finalmente a terceira trata da possibilidade de aprendizagem sobre outros agentes (um tipo de aprendizagem social, em que um agente em cooperação com o outro, pode especializar mais o seu conhecimento sobre a capacidade desse agente).
- Resolução de Conflitos Cognitivos - essa situação diz respeito à possibilidade de discordâncias que possam surgir entre o conhecimento do agente tutor e o do aprendiz.

Esses conflitos podem ser identificados em quatro situações referentes a um processo de resolução de problemas. Tais situações são descritas a seguir:

- tutor está certo e o aprendiz está errado. Essa situação ocorre no momento em que o aprendiz apresenta sua argumentação referente à solução do problema e o tutor detecta algum erro ou falsa concepção no processo de solução;
- aprendiz está certo e o tutor está errado. Essa situação pode ocorrer, mas apenas será percebida e resolvida pela SEH;
- ambos estão certos. Essa situação ocorre quando o tutor reconhece que a solução do aprendiz está correta, mas não se trata de uma solução conhecida pelo tutor;
- ambos estão errados. Essa situação pode ocorrer, mas apenas será percebida e resolvida pela SEH;

O modelo do agente inclui uma organização social. Segundo o modelo proposto por SCHIMAN et al. [SIC92], uma organização social é caracterizada por meio de suas interações. Um agente tutor é constituído por duas camadas principais, a saber: camada do sistema de ensino e camada de cooperação.

A *Camada do Sistema de Ensino* é responsável pelas atividades de ensino adaptativo, tomando por base uma aprendizagem calcada em resolução de problemas nos termos já discutidos. É através dela que ocorrem as interações entre o Aprendiz e o Agente Tutor. O Aprendiz é colocado numa situação de resolução de problemas em interação com o Tutor. Nesse processo, o Agente Tutor acompanha o Aprendiz durante a interação, monitorando suas ações e provendo-lhe um feedback adequado. Os tipos de interações entre o Aprendiz e o Tutor, incluem aspectos didáticos em uma situação de troca de mensagens. O tutor envia ao Aprendiz uma mensagem com conteúdo X e este retorna uma outra com conteúdo Y.

A *Camada de Cooperação* é composta por conhecimentos e mecanismos necessários para a viabilização do trabalho cooperativo na sociedade de tutores. Para isso, cada tutor possui nesta camada seis módulos, são eles:

- Autoconhecimento (AC) - é uma estrutura de conhecimento representando as capacidades do agente.
- Conhecimento Social (CS) - é uma estrutura de conhecimento representando as capacidades dos outros agentes tutores na sociedade. Esse conhecimento é usado para identificar agentes tutores especializados em uma dada tarefa. Isso é necessário quando as capacidades de um agente são insuficientes para lidar com um dado problema e o agente necessita da cooperação de um outro.
- Encaminhamento - é um módulo responsável pela seleção de agentes para iniciar uma cooperação. Esta seleção é feita baseada no conhecimento social, descrito acima. Algumas estratégias e regras de resolução de conflitos são utilizadas por este módulo para solucionar possíveis impasses que possam surgir durante o processo de eleição do(s) agente(s).
- Protocolos - são coleções de possíveis protocolos que podem ser usadas no processo de cooperação entre os agentes. Um protocolo é um conjunto de regras que visa restringir o diálogo entre os agentes com o intuito de possibilitar o processo de construção/interpretação das mensagens trocadas entre eles. Estes protocolos podem ser aceitados a partir de uma biblioteca de protocolos, para atenderem certas exigências da camada de ensino.

- Controle - é um módulo responsável pela coordenação dos protocolos, pelo preparo das mensagens a enviar e pelo processamento das mensagens recebidas.
- Comunicação - esse módulo é responsável pelo envio e recebimento de mensagens, através do ambiente de comunicação.

O ambiente do MATHEMA possui características interessantes, especialmente a possibilidade de acompanhar os desdobramentos do processo de aprendizagem considerando os diferentes tipos de alunos. No MATHEMA o tratamento dos diferentes estilos de aprendizagem do aluno é tratado através da sociedade de tutores artificiais onde existe uma coleção de tutores que podem ser alocados para tarefa de trabalhar com um aluno específico. A análise desta proposta mostrou a possibilidade de tratar alunos com perfis diferenciados e a utilização do trabalho de Minsky [MIN87] para organizar a arquitetura dos agentes reativos. No capítulo 5, apresentamos a forma como resolvemos esta questão na arquitetura proposta.

3.2.3 AME-A: Ambiente Multiagente de Ensino-Aprendizagem

D'Amico [DAM99] apresenta uma proposta de uma sociedade de agentes, em que todos são autônomos e se comunicam através de mensagens. D'Amico utiliza uma abordagem psico-pedagógica no seu trabalho. Segundo ela, o modelo de aluno é definido por um agente *Modela_Aprendiz*, que identifica o nível de conhecimento do aprendiz, seus objetivos de aprendizagem, motivação e as suas características psico-pedagógicas.

Os agentes no ambiente de ensino-aprendizagem apresentam os seguintes aspectos:

- as atividades de cada agente não requerem uma constante supervisão externa;
- não existe uma autoridade central que seja designada para controlar todas as interações e performances entre os agentes.

Nesta sociedade proposta por [DAM99] é permitido a coexistência de vários agentes-aprendizes. A sociedade de agentes é composta por um conjunto de agentes envolvendo dois processos básicos: ensino e aprendizagem. Cada agente é definido por um sextupla $(W_A, A_A, G_A, P_A, S_A, F_A)$, onde:

- W_A : conjunto de possíveis estados de mundo do agente A;
- A_A : conjunto de ações que A pode fazer;
- G_A : conjunto de objetivos de A ;
- P_A : conjunto de planos hipotéticos para os objetivos de A;
- S_A : conjunto de planos nos quais implica a execução de A, isto é, o conjunto de planos que for escolhido para ser executado por A;
- F_A : conjunto de funções usadas para controlar o laço básico do agente A.

O laço básico do agente A foi definido por Steiner [STE93, Apud in DAM96, p.89] em quatro fases: a ativação do objetivo, o planejamento, o agendamento (sincronismo temporal) e a execução/avaliação. Deste modo, funções são utilizadas para ativação de objetivos, localização de planos e execução de ações. Os primeiros três elementos da

sextupla definem a funcionalidade da sociedade e os últimos três definem a operacionalidade.

O processo de ensino envolve os seguintes agentes:

- agente *seleção_objetivos* - seleciona o próximo objetivo e envia o endereço do material para ser enviado ao agente *material_atual*. A velocidade e a ordem na qual o material vai ser usado depende da performance e do interesse do agente que está aprendendo;
- agente *material_atual* - atende o estudante durante o processo de aprendizagem através da apresentação de material instrucional selecionado pelo agente *seleciona_objetivo* de diversas maneiras. A seleção apresentada aqui justifica-se pelas preferências do *agente_aprendiz*;
- *inic_agente* - define a área de dados para o *agente_aprendiz*, mostra os objetivos que podem ser selecionados e define o melhor caminho no grafo dos objetivos;
- agente *suprimento_prática* - dispõem de diferentes tipos de exercícios, exemplos e aspectos de modelo para o *agente_aprendiz* e interfere na escolha feita pelo aluno;
- agente *controle_de_resultado* - supervisiona a performance do *agente_aprendiz* durante os testes e execução dos exercícios de maneira a prover um feedback. Ele também distingue os erros de deslize (erros que ocorrem quando alguém comete um engano por não entender o problema). Se o agente detecta um possível desapontamento ou entusiasmo ele passa esta informação ao agente *dar_ajuda*;
- agente *dar_ajuda* – auxilia o *agente_aprendiz* se sua performance não for muita boa ou se ele estiver desconcentrado. O agente *dar_ajuda* oferece novos exemplos, retoma o a o agente *material_atual*. E indica o uso do navegador (procedimento que orienta o aluno no seu trabalho).

O processo de aprendizagem envolve os seguintes agentes:

- *agente_aprendiz* - é o próprio aluno que utiliza o sistema;
- *agente_navegação* – é o agente que acessa o material armazenado em um hiperdocumento para esclarecer dúvidas, oferecer exemplos, anexar lembretes e apresentar conceitos básicos quando o aluno requisitar.
- agente *seleção_estratégia* – é o agente que seleciona a estratégia de ensino-aprendizagem mais adequada ao *agente_aprendiz*. As estratégias são em número fixo e ativadas em função da quantidade de parâmetros válidos entre o agente a estrutura da estratégia.
- *agente_performance* – é o agente que analisa a história do *agente_aprendiz*, o seu conhecimento, o seu caminho e a sua preferência para tentar ensinar algo que ele ainda não está apto a aprender ou não necessita.

O ambiente proposto por D'Amico envolve também as múltiplas estratégias, porém elas são selecionadas em função dos parâmetros que o agente *seleção_estratégia* recebe dos outros agentes. O trabalho de D'Amico foi importante por, por além de outras coisas, também considerar mais de uma estratégia em função de um modelo diferenciado de aluno, reforçando a importância do tema para a área de pesquisa.

3.2.3 A arquitetura SEM

A arquitetura **SEM** (Sociedade dos Estados Mentais) proposta por Corrêa em [COR94], que é uma arquitetura genérica de agente cognitivo, deliberativo e autônomo, cuja característica fundamental é a especificação do agente através dos estados mentais crença, desejo, intenção e expectativa. A compreensão de cada agente é obtida através da determinação destes estados mentais..

A mudança de uma arquitetura tradicional de STI possibilitou maior riqueza na observação do comportamento dos agentes através das mudanças que ocorrem nos estados mentais.

A representação formal e simbólica dos estados mentais dos agentes se baseia na Teoria das Situações desenvolvida por Barwise e Perry [BAR88] e Devlin [DEV91].. Esta teoria favorece o poder de representação dos estados mentais, aumenta o poder explicativo das interações e como consequência fornece uma melhor avaliação das transformações que ocorrem nos processos de ensino e aprendizagem.

Os agentes locais interagem entre si comunicando-se através de mensagens. os agentes autônomos constituem o que Corrêa [COR94] chamou de “núcleo”. A partir do núcleo define-se a arquitetura **SEM**. Cada agente possui uma arquitetura formada por uma 8-tupla, cujo simbolismo é baseado na Teoria das Situações, semelhante ao agente global. Cada um dos agentes, ditos agentes locais, tem funções específicas decorrentes das definições, das propriedades e das relações dos estados mentais (crença, desejo, intenção, expectativa). A descrição do formalismo utilizado para representar os estados mentais será apresentado no capítulo 4.

O agente global tutor, através do seu agente local desejo, envia uma mensagem para o agente global estudante que a recebe através do seu agente local intenção. O agente local intenção verifica se o desejo está conectado à intenção trazendo para a ação o agente local desejo. A intenção, através do processo de inferência, verifica se existe uma estratégia para preencher a associação entre desejo e intenção.

Ao lado do agente local crença, o agente local intenção, que já possui uma estratégia de ensino determinada, controla a execução e envia ao agente global tutor uma mensagem dizendo que a ação requisitada foi realizada (com sucesso ou não) e que esta foi recebida pelo agente local desejo. Este, por sua vez, conduz para ação os agentes remanescentes para verificar se o desejo foi ou não preenchido enviando uma nova mensagem ou esperando por uma nova que será enviada pelo estudante. O agente global tutor pode receber do agente global estudante uma ação que é a resposta do pedido feito pelo tutor para o estudante através da mensagem enviada pelo agente local desejo (tutor) e a intenção (estudante), ou uma ação que é a requisição feita pelo estudante ao tutor.

O estudo da arquitetura SEM mostrou as possibilidades existentes na modelagem de agentes cognitivos com uso de estados mentais e evidenciou o potencial desta abordagem para tratar as questões de múltiplas estratégias na construção de um tutor que possa oferecer assistência personalizada levando em consideração o estilo cognitivo de cada aluno.

3.2.4 Ambiente de Aprendizagem Interativa multi-agente

O trabalho de [EUS95] foi elaborado na perspectiva da interação multiagentes. O seu objetivo é a construção de um Ambiente de Aprendizagem Inteligente, capaz de ajudar o professor na difícil tarefa de ensinar com sucesso. O ambiente permite a simulação em computador da interação entre os vários componentes intervenientes no processo de ensino-aprendizagem numa situação de aula. A simulação inclui também alguns aspectos relacionados à aprendizagem dos alunos, como o raciocínio e o tipo de assimilação. O professor e o aluno são representados, no Ambiente de Aprendizagem Inteligente (AAI), pelo agente professor (módulo de programação) e pelo agente aluno, respectivamente. O objetivo do agente professor é o de "ensinar" os agentes alunos, a partir de objetivos predefinidos pelo utilizador.

Cada agente é modelado em termos do estado de crenças. Em cada momento o agente é caracterizado por uma estrutura de conhecimento incluindo sua própria informação (organizada através dos seus elementos mais simples denominados crenças e um conjunto de regras de inferência, que delimitam a sua capacidade de raciocínio) e a informação e regras de inferência que julga serem possuídas pelos outros agentes da sociedade. Neste último caso, o conhecimento sobre os outros agentes da sociedade possibilita-lhe ter em cada momento um modelo sobre os outros agentes e ser capaz de escolher, durante o processo de interação qual o agente a quem fazer, por exemplo, um pedido.

A escolha da linguagem interna para representar as crenças do agente foi determinada com a finalidade de ser utilizar em conteúdos programáticos na área da Matemática, embora o seu uso possa ser mais geral. Assim, por exemplo, uma das estruturas utilizada tem o predicado "teorema" (com o objetivo de trazer alguma semelhança com os denominados teoremas da linguagem Matemática). Para a escolha do tipo de "raciocínio", isto é, do conjunto das regras de inferência disponíveis para cada agente, foi feito um inquérito com a finalidade de conhecer mais sobre o tipo de raciocínio aplicado na aprendizagem por alunos já num estado adulto (ao redor dos 20 anos de idade).

A motivação do agente é construída em função dos seus interesses e desejos futuros, o que implica a sua existência. O professor deve procurar uma ligação do contexto lecionado com esses interesses e objetivos do aluno.

O modelo do *agente professor* é formado por dois sub-modelos, o próprio modelo do agente e um modelo de ensino. Estes modelos incluem algumas técnicas de ensino.

Numa segunda fase do desenvolvimento do trabalho de [EUS95], foi ampliado o modelo do agente. Assim, associou-se ao modelo do agente um "tempo de vida" que permite o aparecimento das noções "mais antigo" "mais novo" (uma crença é mais antiga que outra, um agente é mais novo que outro,...). O tempo de vida favoreceu a introdução da denominada "memória" do agente construída, procurando alguma semelhança com o caso humano. Além de um componente temporal, a memória do agente depende de uma componente denominada "motivação".

O modelo do agente é uma ampliação do modelo proposto por Costa et al. [COS92], introduzindo as noções de tempo de vida, motivação e memória. Introduziram também os conceitos tipo de professor, objetivos e plano de aula. O tempo de vida possibilita a avaliação da "idade" do agente com o significado do tempo total utilizado na interação, no raciocínio e na organização de conhecimento do agente.

Podemos, por exemplo, determinar o tempo gasto na resolução de um exercício por agentes diferentes. A motivação implica uma preocupação do utilizador na ligação entre os objetivos de uma aula e os interesses dos alunos. A preparação de uma aula não pode terminar sem uma investigação sobre os interesses dos alunos e sobre a forma como estes podem ser motivados. Somente assim, os objetivos dessa aula poderão ser atingidos.

A memória dependente de um tempo fixo e da motivação do agente (Contexto, Subcontexto), possibilita uma aproximação à memória humana no sentido em que crenças diferentes podem ser guardadas de forma diferente. Há crenças que ficam quase logo "invisíveis" para o agente, há outras que são guardadas de forma permanente e há outras ainda que vão sendo "esquecidas" ao longo do tempo.

O tipo de professor está associado ao tipo de método de ensino utilizado na execução da aula e também à maneira como o professor exerce o seu papel de líder da classe.

O melhoramento do modelo do agente passa pela introdução de um sub-modelo de ações para que possamos utilizar o termo "fazer" para além do termo "conhecer".

Salientamos que a realização da simulação pode ser melhorada com a introdução de técnicas de ensino como elementos isolados que poderão ser reunidos para formar vários tipos de métodos de ensino. Quanto aos componentes do processo de ensino-aprendizagem deverá ser realizada uma revisão no aspecto de melhoria da avaliação da aprendizagem. Estes aspectos foram considerados quando elaboramos a nossa arquitetura apresentada no item a seguir.

3.4 A arquitetura multiagente proposta para construção de um STI

O projeto de um STI que incorpore agentes na sua concepção se situa na subárea de SMA. A pesquisa na área de SMA trata do estudo de agentes autônomos em um universo multiagente [DEM90].

Um arquitetura baseada num SMA possui as seguintes características [ALV97]:

- os agentes devem ser concebidos como uma entidade capaz de realizar um determinado processamento e não como uma entidade capaz de realizar este processamento no contexto de uma aplicação particular;
- a concepção das interações é realizada independente de uma aplicação particular e os protocolos de interação podem ser reutilizados em várias aplicações similares. Novos agentes podem ser inseridos no ambiente utilizando o mesmo protocolo de apresentação e comunicação com o ambiente. Evidentemente, instanciações destas interações se fazem necessárias a fim de viabilizar a funcionalidade destes novos elementos no ambiente;
- o controle global do sistema é implementado de forma descentralizada nos agentes. Durante a fase de resolução, os agentes utilizam suas representações locais dos protocolos de interação e do projeto da organização para raciocinar e agir.

Os SMA híbridos que sustentam a arquitetura geral proposta para modelar um STI são subdivididos em dois módulos: SMAR (constituídos de agentes reativos) e SMAC (constituídos de agentes cognitivos). A divisão entre agentes reativos e cognitivos está

associada à didática para explicar o grau de conhecimento e racionalidade que o agente possui. Este grau não é medido de forma binária, uma vez que varia entre intervalos em que as características dos extremos podem ser adotadas em diferentes intensidades.

A figura 2.2 apresenta a arquitetura geral proposta para o STI modelado utilizando o SMA.

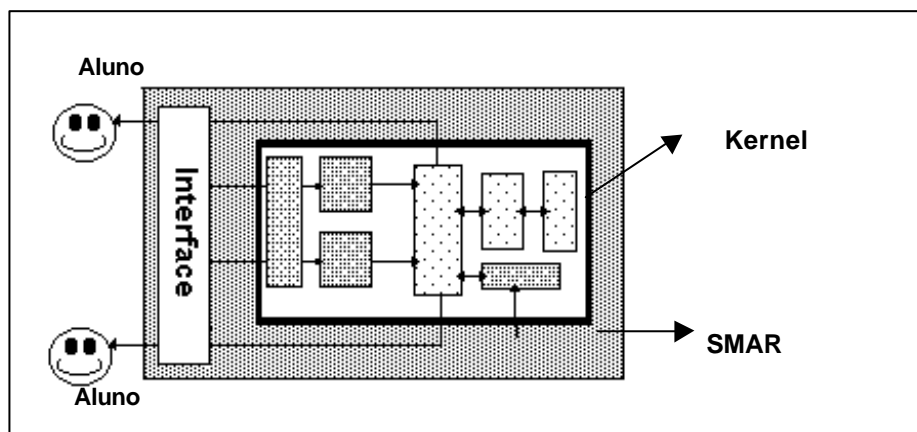


FIGURA 2.2 - STI com arquitetura multiagente

Estamos propondo tratar a base do domínio do STI através de um cenário interativo onde as ações dos alunos transcorrem de forma dinâmica e com resposta imediata. A utilização de agentes reativos permite a visualização (leitura direta) do que está acontecendo no ambiente. Domínios com características similares à do protótipo podem utilizar esta proposta de maneira direta. A possibilidade de modelar o domínio da aplicação na forma de um SMAR permite ampliar a interação, a representação da complexidade do domínio e suas inter-relações, ampliando assim as possibilidades da exploração do conteúdo associado a este domínio.

A fim de demonstrar a viabilidade da arquitetura proposta e proporcionar uma série de testes para depuração e validação das idéias contidas nesta tese, foi desenvolvido um protótipo denominado de MCOE. Este sistema será utilizado como exemplo para as considerações a serem feitas a respeito da arquitetura proposta.

No MCOE, o SMAR é composto pelos elementos do cenário do jogo, constituindo-se de um conjunto de entidades simples (agentes reativos) que não possuem representação do seu ambiente. Segundo Alvares e Sichman [ALV97], as principais características dos agentes dos sistema multiagentes reativos são:

- conhecimento dos agentes é implícito e se manifesta através do seu comportamento;
- seu comportamento se baseia no que é percebido a cada instante do ambiente, mas sem uma representação explícita dele;
- os agentes reativos não mantêm um histórico de suas ações, de forma que o resultado da ação passada não exerce nenhuma influência sobre suas ações futuras;

- a forma de organização dos agentes reativos é similar à dos animais, em oposição à organização social dos agentes cognitivos;
- os sistemas multiagentes reativos têm, em geral, um grande número de agentes; Podendo ser da ordem de dezenas, centenas ou até mesmo milhões de agentes.

Os elementos do cenários do MCOE (peixes, microorganismos e plantas) possuem todas as características acima listadas, além de serem entidades autônomas que existem no ambiente e sofrem com as interações que realizam com os outros elementos do cenário. Estes elementos apresentam um comportamento autônomo, uma vez que eles possuem independência com relação às ações dos agentes externos (outros agentes reativos e agentes cognitivos do ambiente). Possuem também estados internos que se auto-organizam no ambiente. Suas ações emergem para uma ação coletiva devido ao princípio da cadeia alimentar existente na sua modelagem. Seu comportamento é baseado no comportamento de animais reais. Além de todas estas características, os elementos do cenário do MCOE possuem estados internos ligados à satisfação (sentem fome e se reproduzem) e percepção quanto aos limites do seu ambiente (limites do cenário).

O ambiente do MCOE é uma sociedade heterogênea e fechada baseado no controle distribuído (conhecimento, dados e habilidades) entre diferentes agentes. Os agentes cognitivos são orientados por objetivos sendo seu objetivo principal a manutenção do equilíbrio ecológico. A figura 2.2 apresenta a distribuição destes agentes no sistema.

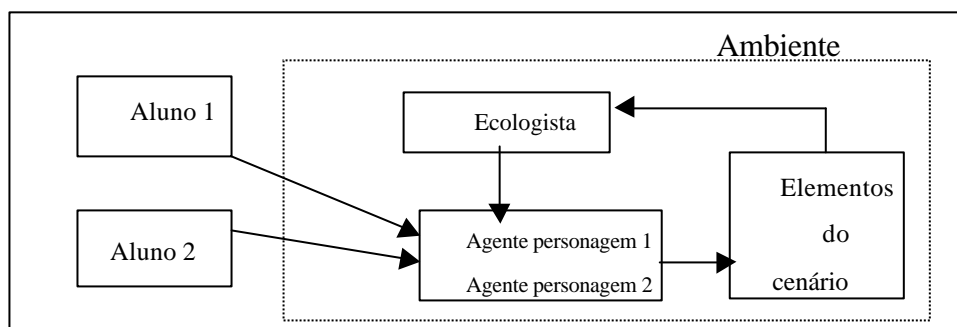


FIGURA 2.2 - A distribuição dos agentes no MCOE

Os agentes reativos foram modelados através do uso de técnicas de orientação para objetos. A especificação do problema (modelo da simulação) foi realizada de forma declarativa representando os dados sob uma forma matemática. Esta especificação contém a definição das entidades envolvidas, a definição das classes e inicializações utilizadas na simulação e a parte das regras (internas, externas, ações e condições).

A estrutura utilizada para os objetos é mostrada na figura 2.3.

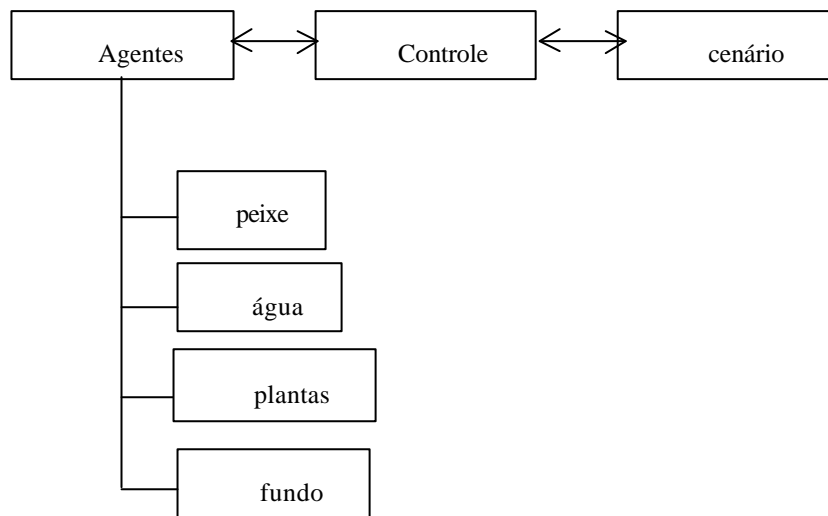


FIGURA 2.3 - Distribuição dos objetos

O objeto *agentes (header)* é responsável pelas ações básicas dos agentes (apresentação, nível de energia, posição, velocidade, aceleração e outros estados físicos). Este objeto também é responsável pelo comportamento final do agente, possuindo uma estrutura que possibilita a inclusão de outros agentes no ambiente. Esta é a estrutura básica utilizada para a implementação.

O objeto *controle* coordena os diferentes elementos do sistema, controla o comportamento dos agentes durante a simulação e trabalha como se fosse um mediador na sociedade de agentes (entre os agentes e suas relações com o ambiente).

O objeto *cenário* é responsável pela representação das condições do ambiente e todos os efeitos que aparecem no ambiente, como por exemplo a aparência da água que é pré-determinada por um conjunto de regras.

O SMAR contém a interface do sistema que possibilita os alunos realizarem suas ações repassadas posteriormente ao kernel cognitivo que processa toda a informação dos agentes cognitivos do sistema (alunos e tutor) e também as informações recebidas do ambiente. O kernel cognitivo é um SMAC com as características do modelo proposto por Ferber e Gasser [FER91]. Ou seja, o kernel cognitivo está baseado num modelo organizacional humano (sociológico) mantendo uma representação explícita do seu ambiente e de outros agentes da sociedade. Ele possui memória do passado, i.e., um histórico de suas ações realizadas nas interações. A comunicação entre os agentes é feita de modo direto, através do envio de mensagens. O kernel cognitivo apresenta ainda um mecanismo de controle deliberativo em que os agentes raciocinam sobre quais objetivos devem alcançar, que planos seguir e quais ações devem ser executadas em um determinado momento. Uma última característica apontada por [FER91] para o kernel cognitivo é que este é uma sociedade que contém tipicamente poucos agentes.

No caso da arquitetura proposta, temos os três agentes cognitivos (dois alunos e um tutor) e uma sociedade de estados mentais que compõem a parte de raciocínio de cada agente cognitivo composta por um conjunto de estados mentais expressos numa

arquitetura BDI¹⁴ que forma outra sociedade de agentes. O capítulo 4 apresenta estas considerações em detalhe.

Sob o ponto de vista organizacional, os agentes da arquitetura proposta possuem um modelo *descendente* apresentado na classificação de Conte e Castelfranchi [CON92]. Os agentes têm a priori um problema a resolver. Isto é típico de uma aplicação educacional e também intrínseco num STI. A cooperação é preestabelecida como uma hipótese de partida, que pode ou não se confirmar. As interações sociais dos agentes são limitadas por uma organização preexistente, que orienta os agentes para atingir o objetivo para o qual o sistema foi concebido. Segundo [ALV97] este tipo de modelo se adequa às estruturas em que a fase de organização é preestabelecida no sistema. No caso da arquitetura proposta para os STI, a organização é uma condição para o estabelecimento das relações entre os agentes.

O modelo do aluno apresentado na figura 2.4, é expresso através dos seguintes conjuntos:

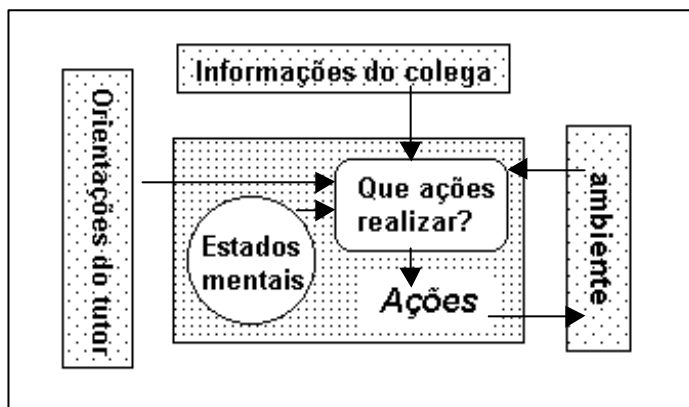


FIGURA 2.4 - Modelo do aluno

- um conjunto de estados mentais (desejos, crenças, intenções e expectativas) a respeito do ambiente (chamamos isto de conhecimento a respeito do ambiente);
- um conjunto de estados mentais sobre si mesmo;
- um conjunto de estados mentais sobre o tutor;
- sensores para receber informações sobre o ambiente;
- informações que o aluno recebe do seu colega;

O aluno utiliza estas informações para decidir qual ação ele deve realizar.

O conjunto de estados mentais considerados na arquitetura interna de cada agente cognitivo, possui uma granularidade que permite o rastreamento do processo de raciocínio dos agentes.

¹⁴ Manteremos o termo BDI apesar deste texto estar escrito em português a fim de facilitar a compreensão das considerações apresentadas, uma vez que a sigla BDI é amplamente utilizada pelos pesquisadores da área de IAD.

Desta maneira podemos estudar o comportamento dos agentes e utilizar estas informações para auxiliar a entender o processo de raciocínio dos alunos. A granularidade vai permitir, também, favorecer a melhoria no conjunto de heurísticas utilizadas pelo tutor, uma vez que poderemos fazer uma análise dos estados mentais que aparecem em situações específicas de ensino-aprendizagem.

O kernel cognitivo do sistema é o local onde está o tutor. A arquitetura do tutor possui um módulo de inferência que identifica os estados mentais associados a cada ação do aluno. Estes estados mentais são extraídos do diálogo dos alunos e usados na teoria dos atos de fala de Searle [SEA84]. As informações sobre os dois alunos são armazenadas no módulo de representação interna que o tutor possui para cada aluno. Estas informações são construídas de maneira dinâmica e em tempo real ao longo das interações e são fornecidas pelo ambiente quando o aluno realiza uma ação. Além destas informações, existe um módulo com o conjunto de estados mentais que o tutor possui acerca do seu próprio comportamento, do ambiente e do módulo seletor. Este último, o módulo seletor, possui o conjunto de heurísticas a respeito do comportamento do tutor (estratégias e táticas que irá utilizar).

O capítulo a seguir, analisa em detalhes a abordagem mentalística utilizada neste trabalho. Este nível de detalhamento faz-se necessário para a contextualização e entendimento da modelagem dos agentes cognitivos.

4 Agentes Pedagógicos modelados através de estados mentais

Desde o final dos anos 30, os pesquisadores buscam um paradigma para a investigação precisa e unificada da mente humana a fim de lançar bases para a fundação de uma ciência da mente. A idéia de termos processos mentais estudados à luz de um modelo computacional é uma alternativa para dilemas metodológicos que a psicologia enfrenta. Estes dilemas se polarizam entre o comportamentalismo (com a questão dos estímulos e respostas para explicar o processamento humano) e o introspeccionismo (exame dos pensamentos e sentimentos envolvidos no pensamento humano). Segundo Teixeira [TEI98], a busca destes modelos computacionais postula a existência de estados mentais e representações de estados internos que operam entre os *inputs* e os *outputs* que a mente recebe. O esforço interdisciplinar de simular processos mentais humanos requer a contribuição de todas as áreas ligadas ao estudo da mente (Psicologia, Lingüística, Filosofia, Neurologia e outras). A contribuição destas áreas se dá de forma mais direta ou indireta dependendo da natureza do modelo computacional considerado.

O desenvolvimento da IA, especialmente nas últimas décadas, permitiu a constituição das bases utilizadas para compor uma ciência da mente. Através da IA, os cientistas refletem sobre o significado do que é ser inteligente, ter vida mental, ter consciência e outros conceitos empregados nas áreas afins. Se a IA não conseguiu até agora criar um sistema computacional inteligente, ela pelo menos auxiliou na reflexão dos cientistas sobre os processos mentais e seu funcionamento.

Mesmo que a inteligência seja complexa, assim como apontado por Corrêa [COR98], nós podemos achar um modo de entender melhor o que está acontecendo na mente de uma pessoa através de reações mentais simuladas por computador. Através destas simulações possivelmente teremos indícios de como o processamento humano funciona e inferiremos algumas possibilidades acerca da estrutura básica de trabalho da mente. A partir disso, poderemos projetar melhores simulações. A idéia por trás desta coreografia de estados mentais está baseada nas interações de dois espaços principais: o espaço de declaração e o de arquiteturas dos agentes.

Apesar de ainda desconhecermos aspectos importantes sobre o funcionamento da mente, temos os componentes chaves dos estados mentais de um agente (crenças, desejos e intenções) e muitos atributos como satisfação, insatisfação, certeza ou urgência que atuam como os *infos* da Teoria da Situação [DEV91]. Tais componentes combinam tipos particulares com ajuda de certas leis e controles, como no caso do trabalho de Corrêa [COR94].

Ainda existe uma grande distância entre o nosso conhecimento sobre como a mente natural funciona e o que realmente acontece. Entretanto, através da observação da estrutura dos estados mentais sugeridos como uma metáfora do que possivelmente ocorre na mente humana, podemos construir modelos computacionais (software) para tentar entender como esses sistemas biológicos funcionam. Tais modelos são reveladores de segredos de uma variedade de reações artificiais que podem ser um recurso para o projeto de agentes cognitivos sintonizados com problemas de domínios particulares.

A idéia principal desta abordagem mentalística, segundo [COR94], se concentra no fato de que o agente cognitivo possui estados internos que se relacionam com o estado

do ambiente com o qual interage. Estes estados seriam correspondentes aos estados mentais humanos, que apresentam um vínculo com o mundo em termos da sua existência e significância..

As idéias subjacentes à abordagem social introduzem conceitos promissores para a compreensão de como construir agentes com habilidades especiais, tais como, aprender e ensinar. A aquisição do conhecimento pelos agentes numa sociedade multiagente emerge de suas interações em que cada agente se comporta como um tutor e como um aluno. De fato não existe um tutor e um aluno, mas apenas atitudes de tutor e atitudes de aluno, conforme Corrêa, Coelho e Viccari [COR98].

O cérebro humano tem um sofisticado sistema neuronal que pode armazenar, interpretar e processar informação em diferentes graus. O filósofo Daniel Dennett [DEN78; DAN87] denominou este sistema interno de noções intencionais. Segundo Dennett, este sistema intencional pode ser simulado usando um método de atribuição de crenças, desejos e perspicácia de raciocínio. Assim, o sistema tem níveis diferentes de intenção: um sistema de primeira-ordem com crenças e desejos (mas nenhuma crenças e desejos sobre eles) e um sistema de segunda-ordem mais sofisticado (com crenças e desejos sobre crenças e desejos), i.e., um sistema que trabalha como um meta-nível sobre as crenças e desejos.

O antropomorfismo para construir os agentes artificiais é útil para a descrição da instância intencional, desde que nós entendamos suficientemente o mecanismo essencial a fim de termos uma descrição mais simples da mecânica de seu comportamento [SHO90].

Este capítulo analisa as questões relativas à abordagem mentalística utilizada neste trabalho e os conceitos a ela relacionados. Além disso, serão descritas as características do ambiente lógico utilizado para a modelagem dos agentes cognitivos e apresentados exemplos de sua aplicação no protótipo construído para esta tese.

4.1 Teorias Mentalísticas de Agentes

Muitos agentes modelados em IA são baseados no sistema físico simbólico de Simon e Newell [NEW76], no qual se assume que os agentes mantêm uma representação interna do seu mundo composta por um conjunto de estados mentais que podem ser modificados de alguma forma por um raciocínio simbólico. O agente possui estados internos que se relacionam com o estado do ambiente com o qual interagem. Esses estados mentais devem corresponder aos estados mentais humanos, mas essa correspondência não precisa ser exata.

Estados mentais humanos têm um vínculo com o mundo através do qual estabelecem sua existência e sua significância. Essa característica pela qual os estados mentais humanos “são acerca de”, “referem-se a objetos ou situações do mundo”, ou “se dirigem a” é chamada de intencionalidade. Crença, desejo, expectativa e intenção são exemplos de estados intencionais. Em certas situações específicas, os estados mentais de alegria e tristeza não são intencionais.

Um segmento da pesquisa em IA tem explorado modelos de agentes baseados em crenças, desejos e intenções. As arquiteturas que seguem este paradigma são conhecidas como arquiteturas BDI (*Belief, Desire, and Intention*). As idéias básicas da abordagem BDI são descrever o processamento interno do estado de um agente utilizando um

conjunto de categorias mentais (crença, desejo e intenções) e definir uma arquitetura de controle através da qual o agente seleciona racionalmente o curso de suas ações. Algumas abordagens de arquitetura BDI agregam as noções de planos e objetivos. Os trabalhos de Bratman et al. [BRA84;BRA87;BRA89], Rao e Georgeff [RAO92] são exemplos desta suplementação. Os conceitos individuais associados à estas categorias são apresentados a seguir sob diferentes perspectivas encontradas na literatura em forma de itens. Além disso, a visão particular destes conceitos utilizada neste trabalho é apresentada no item 3.3 .

- Crenças:

Segundo Müller [MUL96], as crenças do agente expressam suas expectativas sobre o estado atual do mundo e sobre a chance de uma ação ou conjunto de ações atingir um certo efeito.

Nos trabalhos de Wooldridge [WOO95] e Halpern e Moses [HAL92], as crenças são modeladas usando a semântica de mundos possíveis. Nesse sentido, um conjunto de mundos possíveis é associado com cada situação, denotando os mundos que o agente acreditar serem possíveis. Na visão de Bratman [BRA89], crenças são as visões que o agente possui sobre o ambiente em que ele se encontra.

Para Corrêa [COR94], crença é um estado mental intencional fundamental para as interações dos agentes, com noção idêntica a de *conhecimento*, cujo conteúdo externo é uma proposição. Pode ser representado, utilizando-se a Teoria das Situações, como:

$C \models \{ \langle Bel, A, P, l, t, v \rangle \}$, onde:

- Bel é a relação para a representação de crença;
- A representa o agente;
- P é uma proposição (i.e., uma situação);
- l é uma localização espacial (local)
- t é um instante de tempo
- v 1, se o agente A acredita que a proposição P ocorre no local l e no tempo t
- 0, se o agente A não acredita que a proposição P ocorre no local l e no tempo t

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

$B_a^t \varphi$, significando que o agente 'a', no tempo 't', acredita na sentença φ .

- Desejos:

Segundo Müller [MUL96], desejo é uma noção abstrata que especifica as preferências acerca dos estados futuros do mundo ou o curso das ações que o agente possivelmente quer que se verifiquem. Uma questão importante relacionada ao desejo é que ao agente é permitido ter desejos inconsistentes, que ele não precisa acreditar que possam ser alcançados. Ter desejos não significa que o agente agirá para atingí-los. É um dos estados mentais humanos que não tem uma definição precisa, sendo, portanto, usado e explicado de diversas maneiras. Um estado mental é motivador se é um mecanismo ou representação que tende a produzir, modificar ou selecionar ações à luz das crenças. Desejo é intencional e motivador apresentando as seguintes características:

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

Desejos são estados disposicionais, conforme terminologia empregada por Sloman [Apud COR 94]. As ações são realizadas através das *intenções* causadas pelos desejos.

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

A arquitetura SEM, com base na Teoria das Situações, apresenta o desejo D de uma agente A como um estado mental intencional e motivador, representado pela seguinte situação:

$D \models \{ \langle \text{Des}, A, P, e, l_0, t_0, v \rangle \}$, em que:

- Des é a relação para a representação de desejo,
- A representa o agente que possui o desejo,
- P é uma proposição,
- e é uma situação que informa sobre a satisfação, não satisfação, urgência, intensidade e insistência do desejo D
- l_0 é a localização espacial (local) associada à ocorrência do desejo D
- t_0 é o tempo associado à ocorrência do desejo D

- v 1, se ocorre o desejo D ao agente A
 0, se não ocorre o desejo D ao agente A

• Intenções:

Assim como os desejos, as intenções contêm a representação dos estados que o agente quer que se verifiquem.

A base para a definição do conceito de intenção está fortemente associada aos trabalhos filosóficos de Michael Bratman [BRA84; BRA89]. Este autor claramente distingue o conceito de fazer coisas intencionalmente (ação) e possuir intenção de fazê-las (estado mental). Por exemplo:

1- “Escrevo esta tese”. Intencionalmente escrevo esta sentença caracterizando a ação no conceito de intenção;

2- “Escrevi esta tese para ilustrar uma nova idéia sobre STI”. A intenção de exemplificar caracteriza o estado da mente no conceito de intenção. Ter ‘intenção de’ não garante que a ação será realizada, mas apenas que se intenciona realizá-la.

Estes exemplos identificam as duas formas distintas embutidas no uso cotidiano da palavra intenção: ações (físicas, verbais, ..) e ações mentais (estados da mente). A tradição de estudos filosóficos sobre a mente e a ação diz que o agente mostra o que intenciona fazer ao especificar os papéis funcionais característicos da intenção. Para tanto, é necessário articular as relações sistemáticas entre intenção, outros estados psicológicos (como crença e desejos), processos e atividades psicológicos relevantes (tais como raciocínio prático) e entradas (percepções) e saídas (ações) cruciais.

Como a ação intencional e a intenção de agir são relacionadas? Existem duas abordagens comuns para esse problema: modelo *desejo-crença* e visão simplista.

No modelo *desejo-crença*¹⁵, a ‘ação intencional’ está relacionada aos desejos e crenças do agente, enquanto a ‘intenção para agir’ é vista como redutível a certos desejos e crenças. A vinculação entre elas se reduz ao problema da relação do complexo de desejos e crenças constituinte da ‘intenção para agir’ e dos desejos e crenças necessários para ‘ação intencional’.

Na visão de Bratman [BRA89] essa abordagem do modelo *desejo-crença* está errada, uma vez que seres humanos constróem planos. Esses planos ajudam a guiar a conduta das pessoas e coordenar suas atividades no tempo, de uma forma que seus desejos e crenças comuns não fazem. Nesse contexto intenções são vistas como estados da mente distintos que não podem ser reduzidos a conjuntos de desejos e crenças.

Na visão simplista¹⁶, para executar uma ação intencional A é necessário ter a intenção de executar A; os estados mentais do agente, no instante da ação, necessariamente devem conter A entre as coisas que ele intenciona. Bratman rejeita essa

¹⁵ Postulado por diversos autores, dentre eles R. Audi, M Beardsley e P. Churchland [Apud BRA 84, p. 375].

¹⁶B. Aune e J. Searle [Apud BRA 84, p.377] são filósofos que advogam a visão simplista.

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

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

Uma intenção direcionada para o presente é aquela na qual se tem a intenção de realizar a ação imediatamente. Por outro lado, uma intenção direcionada para o futuro é aquela em que se tem a intenção de realizar a ação mais tarde.

Bratman apresenta a idéia de intenção como um estado mental associado a planos de coordenação e a ação intencional. Intenções direcionadas para o futuro são elementos típicos de grandes planos e auxilia a coordenação das atividades ao longo do tempo. Para coordenar as atividades do agente, o plano deve ser consistente internamente, i.e., deve ser possível realizá-lo por completo. As intenções são *fracamente consistentes* quando todas elas podem ser colocadas juntas em um plano internamente consistente. As intenções são ditas *fortemente consistentes* quando relação as crenças do agente, quando todas elas podem ser colocadas juntas em um plano consistente com as suas crenças.

Para auxiliar na coordenação, as intenções do agente necessitam ser fortemente consistente com relação às suas crenças. Logo, existe um demanda racional para que as intenções direcionadas para o futuro sejam fortemente consistentes com às crenças do agente. Essa demanda deverá ser respeitada no raciocínio prático e no planejamento. Ela distingue intenções (racionais) de simples desejos.

A visão simplista não é aceita, porque não garante que as intenções sejam fortemente consistentes. Ela impõe uma forte ligação entre intenção e ação intencional, mas essa ligação é insensível a diferenças na demanda do raciocínio prático. Assim, para Bratman ter a intenção (racional) de executar A, somente demanda que as crenças do agente não sejam inconsistentes com a crença que executará A. O requisito “fortemente consistente” parece estar solidamente ligado à característica básica das intenções, independente do seu papel na coordenação de planos.

A intenção formada de alguma forma influencia a ação futura do agente. A intenção atual persiste e guia o que será a ação executada, embora não seja irrevogável. Tal irrevogabilidade é irracional, pois as coisas mudam e o agente nem sempre antecipa o futuro com exatidão.

Na literatura dominante, na linha representada por Donald Davidson [DAV80], o raciocínio prático inicia com um conjunto definido de desejos e crenças do agente e gera uma decisão, uma escolha ou uma ação. O raciocínio prático é um meio para ponderar considerações a favor e contra em que as considerações relevantes são providas pelo o que o agente *deseja/valoriza/importa-se* e pelo o que acredita. Portanto, raciocínio prático consiste em ponderar desejo-crença a favor e contra cursos alternativos de ação. Esse modelo não prevê papel especial para as intenções direcionadas para o futuro do agente como entradas para o raciocínio prático. Este paradigma é adotado pelas teorias de decisão.

Bratman enfatiza o papel de intenções prévias como entradas para o raciocínio. Nesse sentido, o raciocínio é realizado a partir de intenção prévia e crenças relevantes para derivar intenções sobre meios, passos preliminares, ou mais especificamente, cursos de ação. Esse paradigma é utilizado no “problema de planejamento”.

Essas linhas de raciocínio são relacionadas através das seguintes idéias e conceitos:

- deliberar é um processo que demanda tempo e outros recursos (o que impõe limites para a extensão de uma deliberação, no tempo da ação). Através de intenções direcionadas para o futuro, o agente usa as deliberações presentes para conformar a ação futura. Para alcançar objetivos complexos, o agente deve coordenar suas atividades individuais presentes, futuras e as dos outros agentes. As intenções direcionadas para o futuro facilitam tanto a coordenação interna como a coordenação entre agentes;
- as intenções direcionadas para o futuro são tipicamente elementos de grandes planos de coordenação. Planos são estados mentais que envolvem um conjunto apropriado de compromissos para agir; o agente tem um plano para executar A somente se é verdade que ele planeja executar A;
- sendo os agentes limitados, seus planos são tipicamente parciais. Algumas decisões específicas são deixadas para posterior deliberação considerando a consistência com os demais planos do agente e evitando a necessidade de antecipar o futuro. Também pode ser considerada a complexidade da coordenação de atividades intra e entre agentes;
- planos tipicamente possuem estrutura hierárquica. Planos concernentes aos fins englobam planos relativos aos meios e passos preliminares. Intenções mais genéricas englobam as mais específicas. O agente pode deliberar sobre partes (p. ex. meios ou passos preliminares) enquanto mantém fixas outras partes (intenções finais). Nesse tipo de raciocínio, o agente satisfaz planos parciais que guiam, com sucesso, a sua conduta para a satisfação da intenção final;
- planos devem ser consistentes internamente com as crenças do agente (podem ser executados juntos, com sucesso, em um mundo no qual as crenças do agente são verdadeiras) e apresentar coerência de meios-fins. Cada um deles deve ser preenchido com subplanos concernentes a meios, passos iniciais e cursos de ações relativamente específicos. O subplano deve ser tão extenso quanto o agente acredita ser necessário no momento para executar o que ele planeja.

Em suma, intenções e planos desempenham dois papéis diretos como entradas para o raciocínio prático. Primeiro, as intenções prévias freqüentemente colocam decisões para deliberação futura ao se considerar a necessidade de coerência de meios-fins. O mais cedo possível, o agente necessita preencher o seu plano, com especificações para poder executá-lo. Segundo, dado a necessidade da consistência (interna e com as crenças), as intenções prévias restringem futuras intenções.

Intenções prévias colocam problemas para deliberação, estabelecendo padrões de relevância para as opções em consideração. Além disso, restringem as soluções para esses problemas providenciando um filtro de admissibilidade para as opções. Dessa forma, intenções prévias e planos ajudam a tornar o processo de deliberação tratável por agentes com substancial limitação de recursos.

O modelo que surge, então, pode ser visto como uma relação entre dois tipos de raciocínio prático: a ponderação de razões oriundas de desejo-crença, pró e contra

opções conflitantes, e o raciocínio a partir de uma intenção prévia para derivar intenções concernentes aos meios e formas para a execução do plano. As intenções prévias são vistas como elementos de planos parciais que provêm uma estrutura de fundo para a ponderação das razões oriundas de desejo-crença. É essa estrutura que coloca problemas para deliberação posterior e restringe suas possíveis soluções.

O raciocínio prático possui dois níveis:

- os planos parciais colocam problemas e providenciam um filtro de admissibilidade nas opções que são suas soluções potenciais; e
- as razões oriundas de desejo-crença entram como considerações a serem ponderadas na deliberação entre opções relevantes e admissíveis.

As crenças possuem grau de confiança (probabilidade subjetiva) variando entre 0 e 1. Aquelas com grau de confiança muito elevado (próximos a 1) são chamadas de crenças *flat-out*. Essas crenças, combinadas com os planos do agente, tornam certas opções inadmissíveis. A estrutura de fundo para o raciocínio prático inclui crenças *flat-out*, intenções prévias e planos.

As intenções prévias e planos devem ter uma relativa estabilidade, resistindo a reconsiderações. Se constantemente o agente reconsidera méritos, intenções prévias e planos, a utilidade na coordenação e no auxílio na sua luta contra os recursos limitados será de pouca utilidade. Entretanto, é irracional tratar os planos prévios como irrevogáveis se não há expectativas de que eles serão realizados. Essas considerações sugerem que os agentes racionais necessitam políticas gerais e hábitos que governem a reconsideração de intenções prévias e planos. A não reconsideração de intenções prévias e planos devem ser tomadas como *default*, mas um *default* a ser alterado diante de certos tipos especiais de problemas não previstos.

Bratman [BRA89] afirma que o modelo sobre o papel das intenções direcionadas para o futuro como entrada para raciocínio posterior é dependente da idéia de que as intenções do agente sejam consistentes com suas crenças. É esta demanda de consistência que dirige a operação do filtro de admissibilidade de opções. Por outro lado, existe uma segunda idéia implícita no modelo baseada na assunção de que intenções prévias possuem estabilidade relativa e, junto com planos, desempenham papel crucial de coordenação: toda vez que o agente tiver a intenção de executar A no futuro, ele necessariamente deve ter a crença que irá executar A.

Intenções, desejos e valoração são exemplos de pró-attitudes, já crenças não são pró-attitudes. Pró-attitudes determinam um papel motivacional. Na conjunção com crenças, elas podem levar o agente a agir. A maioria delas são apenas influenciadoras potenciais de conduta, mas a intenção é uma pró-atitude controladora. O agente pode esperar para executar suas intenções em momento que considerar apropriado. Isso é possível, devido às intenções serem controladoras de conduta. As intenções podem ser executadas pelo agente enquanto persistirem. Como elas são relativamente estáveis, o agente pode contar com que elas persistam até o momento exato de agir.

Essas são as características que permitem explicar como as intenções dirigidas para o futuro normalmente auxiliam a coordenação dando suporte às expectativas de sucesso associadas às suas execuções. As intenções dirigidas para o futuro deverão ser consistentes internamente. Além disso, elas deverão considerar as crenças do agente e oferecer suporte às crenças para serem executadas com sucesso. Em casos normais, quando o agente tem a intenção de A, ele também acredita que ele A.

Outro fator importante a ser considerado no raciocínio prático de agentes racionais com recursos limitados é a diferença entre intenção e escolha. Intenções são tipicamente fundamentadas em deliberação prévia incluindo atitudes relativamente estáveis que funcionam como entradas para o raciocínio prático. Como pró-atitude controladora de condutas, a intenção está intimamente relacionada com o esforço, o comprometimento e a ação. Já as escolhas de um agente racional são restritas por pressões holísticas (expressas através de clareza de padrões de raciocínio e honestidade intelectual, ou seja ausência de má fé). Escolha e intenção são diferentemente afetadas por padrões de bom raciocínio, embora sejam ambas relacionadas com o raciocínio posterior e ação. Assim não podemos assumir que um agente intenciona todas as suas escolhas realizadas com base no raciocínio prático.

Uma vez definidos estes três estados mentais, os modelos existentes para uma arquitetura BDI estabelecem as relações entre os estados mentais. Segundo Móra [MOR99], essas relações podem acontecer de duas formas. Primeiro, as intenções adotadas por um agente são um subconjunto consistente dos desejos deste agente. Segundo, as intenções não podem ser conflitantes com crenças dos agentes. Ou seja, o agente não pode possuir intenções que ele acredite sejam impossíveis de satisfazer.

Até o presente momento, nós discutimos sobre as crenças, os desejos e as intenções. Resta ainda, discutirmos sobre o conceito de “expectativa”.

- Expectativa:

Expectativa é o quarto estado mental associado aos agentes cognitivos que é tratado como uma crença futura do agente.

Schegloff e Sacks [Apud COR 94] analisaram registros de conversas naturais e observaram que a estrutura das conversas consistem, em grande parte, de pares adjacentes que ocorrem seqüencialmente no diálogo. Esses pares adjacentes são os pares de frases proferidas por diferentes interlocutores.

Tais situações ocorrem em troca de cumprimentos, pergunta-resposta, oferta-aceite, oferta-recusa, etc. Normalmente as frases da segunda parte estão restritas às condições impostas pela primeira. Não é necessário que os pares de frases da primeira e da segunda parte sejam contíguos. O fundamental é que uma parte seja resposta da outra.

Schegloff e Sacks observam ser possível resolver os seguintes problemas:

- como podemos dizer que dois itens num diálogo fazem parte de um par adjacente ou de duas unidades separadas, cada uma podendo ocorrer depois da outra?
- como podemos detectar a ausência de um item relevante?

Schegloff e Sacks observaram que ao controlar as conversas através da aplicação do princípio da relevância condicional obtém-se respostas às questões levantadas:

“Por relevância condicional de um item acerca de outro queremos dizer: dado o primeiro, o segundo é esperado; e no caso deste ocorrer, este pode ser visto como o segundo item do primeiro; no caso de não ocorrer ele é considerado como oficialmente ausente. Tudo isto é providenciado pela ocorrência do primeiro item que cria uma expectativa à respeito do que deveria vir a seguir.” [Apud COR 94, p.105]:

Segundo Corrêa [COR94], o fenômeno dos pares adjacentes é mais geral e ocorre em qualquer outro tipo de interação dos agentes. O princípio de relevância condicional pode ser estendido a qualquer seqüência de ações realizadas pelo agente: *a realização de uma ação pode causar um conjunto de expectativas a respeito das possíveis situações que vão ocorrer no futuro*. Esse autor considera expectativa como um estado mental intencional e motivador, que pode ser representado pela Teoria de Situações da seguinte forma:

$E \models \{ \langle \text{Exp}, A, P, e, l_0, t_0, v \rangle \}$, onde:

- Exp é a relação para a representação de expectativa,
- A representa o agente que possui o desejo,
- P é uma proposição,
- e é uma situação que informa sobre a satisfação, não satisfação, urgência, intensidade e insistência da expectativa E
- l_0 é a localização espacial (local) associada à ocorrência da expectativa E
- t_0 é o tempo associado à ocorrência da expectativa E
- v 1, se ocorre a expectativa E ao agente A
0, se não ocorre a expectativa E ao agente A

A expectativa apresenta as propriedades descritas a seguir:

a) *Uma crença e a realização de uma ação podem causar uma expectativa*. Por exemplo, considere o teste proposto por Samarão [Apud COR 94] a ser realizado com crianças entre 9 e 12 meses de idade, às quais é dada uma bola de borracha, e uma oca, que possui um orifício através do qual salta um boneco quando a bola é apertada. Deixa-se a criança apertar a bola e observar o boneco saltar. Após alguns instantes, entrega-se um outro brinquedo e por fim uma outra bola semelhante à primeira, mas que não contenha o boneco em seu interior. Considera-se o teste positivo quando o bebê demonstrar grande espanto e desapontamento por não ver o boneco saltar de dentro da bola, ao espremê-la;

b) *Uma expectativa pode causar uma ação*. No exemplo acima, em decorrência da expectativa frustrada, o bebê aperta novamente a bola.

c) *Uma expectativa pode causar uma crença*. Por exemplo, se um agente solicita a ajuda de outro agente e este não responde, após alguma insistência, o primeiro agente pode passar a acreditar que o segundo não quer mais colaborar com ele.

d) *Uma expectativa pode causar um desejo*. Por exemplo, suponha que desejamos saber um número de telefone de determinada pessoa e ligamos para o número 102 (serviço de informações). Discamos o número. Nesse caso, temos a expectativa de que o alguém atenda. Caso o ninguém atenda, a expectativa é frustrada e pode surgir o desejo de postergar a ligação ou mesmo ligar para outra pessoa afim de conseguir a informação desejada;

Conforme essas propriedades, as expectativas, têm a função de contribuir para aumentar a eficiência da satisfação das intenções, visto que a reação do agente para

satisfazer uma expectativa quando ela é frustrada, evita a necessidade imediata de rever a estratégia traçada inicialmente, mesmo quando ela é parcial.

O problema com a representação dos estados mentais está no seu contexto, i.e., os estados mentais observados são colecionados de experiências baseadas em situações reais. O conjunto de dados destes estados mentais pode variar e pode depender muito do ponto de vista do investigador.

4.1.1 Teorias formais de agentes

Existem duas abordagens fundamentais ao problema sintático (*formalismos*): a primeira utiliza uma abordagem modal (com operadores modais) e a segunda utiliza uma *meta-linguagem* (uma linguagem de primeira-ordem que contém condições que denotam fórmulas de algum outro objeto de linguagem). As lógicas modais utilizadas geralmente não são tratáveis computacionalmente. Segundo Móra [MOR99], isso faz com que estas lógicas modais possam ser usadas como linguagem de especificação, mas não como ferramenta básica para implementar agentes. Os modelos que trabalham com a segunda abordagem (perspectiva de especificação) enfatizam a definição das propriedades que os agentes idealmente deveriam possuir sem se preocupar em como elas seriam construídas nos agentes.

Todo o formalismo é conectado com teorias de agentes que combinam os aspectos sintáticos e semânticos na arquitetura do agente. O aspecto semântico é importante porque nós precisamos orientar a dinâmica dos agentes. Uma teoria de agente completa tem que definir como os atributos dos agentes estão relacionados.

A teoria mais utilizada de agentes é a proposta por Cohen e Levesque [COH89], cujo formalismo utilizou uma teoria de intenção e, posteriormente, uma teoria de atos de fala.

Em [COH87] é descrito um formalismo para modelar estados mentais, em especial, as intenções. A sintaxe deste formalismo tem os conectivos usuais da Lógica de Primeira Ordem com igualdade e operadores de Lógica Modal¹⁷, bem como algumas extensões listadas a seguir:

¹⁷Lógica Modal pode ser descrita resumidamente como a lógica da necessidade e possibilidade do que “deve ser” e do que “pode ser” [HUG 68, Apud COS 92].

Uma assertiva que é forçada a ser verdadeira em todas as situações em que pudermos verificar a sua condição de validade é dita necessariamente verdadeira, ou simplesmente necessária. Assim uma verdade necessária é aquela que não poderia ser falsa em nenhuma circunstância.

Se pudermos verificar que para todas as situações uma assertiva é falsa, dizemos que ela é necessariamente falsa, ou simplesmente impossível. Se uma assertiva não é nem impossível e nem necessária, dizemos que ela é Contingência. Assim uma verdade contingente é aquela que poderia ser falsa em circunstâncias específicas. Admitindo a noção de mundos possíveis onde a lógica é aplicada, podemos afirmar que uma verdade necessária é verdade em todos os mundos possíveis e uma verdade contingente é verdade no mundo atual, mas não se mantém verdade em todos os mundos possíveis.

Se uma assertiva não é impossível, dizemos que ela é possível. Essas quatro noções (necessidade, impossibilidade, contingência e possibilidade) definem os modos básicos nos quais uma assertiva pode ser falsa ou verdadeira e são os principais objetos de estudo da lógica modal.

São usados os operadores lógicos L para representar necessidade e M para representar possibilidade; os quais, combinados com a negação, expressam as modalidades básicas de impossibilidade e contingência.

(BEL $x p$) x tem a crença p

(GOAL $x p$) x tem a meta p

(AGT $x e$) x é o único agente da seqüência de eventos e , onde $e_1 < e_2$ e e_1 é uma seqüência inicial de e_2

(HAPPENS a) a seqüência de eventos da ação a acontecerá

(DONE a) a seqüência de eventos da ação a aconteceu, em que:

Ação a é uma seqüência de eventos que pode ser:

. composta: $a;b$

. escolha não determinística: $a|b$

. testada: $p?$

. repetida: a^*

Ações:

(DONE $x a$) $\stackrel{\text{def}}{=} (\text{DONE } a) \wedge (\text{AGT } x a)$

(HAPPENS $x a$) $\stackrel{\text{def}}{=} (\text{HAPPENS } a) \wedge (\text{AGT } x a)$

Eventualidade:

$M p \stackrel{\text{def}}{=} \exists e (\text{HAPPENS } e;p?)$

$M p$ é verdadeiro se existe uma ação que acontece (inclusive a ação nula) após p se realizar.

Depois:

(LATER p) $\stackrel{\text{def}}{=} \sim p \wedge L p$

Sempre:

$L \stackrel{\text{def}}{=} \sim M \sim p$

p é verdadeiro através da ocorrência de todos os eventos desde agora até o futuro.

Antes:

(BEFORE $p q$) $\stackrel{\text{def}}{=} \forall (\text{HAPPENS } c;q?) \exists a(a \leq c) \wedge (\text{HAPPENS } a;p?)$

A wff p tornar-se-á verdadeira não depois que q

Conhece:

$$(\text{KNOW } x \text{ p}) \stackrel{\text{def}}{=} p \wedge (\text{BEL } x \text{ p})$$

Competência:

$$(\text{COMPETENT } x \text{ p}) \stackrel{\text{def}}{=} (\text{BEL } x \text{ p}) \supset p$$

Agentes que são competentes em relação a alguma proposição têm somente crenças corretas sobre elas. É razoável assumir que os agentes são competentes com respeito às suas próprias crenças, objetivos e realização de objetivos iniciais (eventos primitivos). Os autores também apresentam a semântica dessa linguagem baseados no modelo de mundos possíveis¹⁸ para crenças, metas (objetivos) e eventos.

A lógica utilizada em [COH87] foi usada para a análise de conflito e cooperação em diálogo multiagente e vários estudos nas fundações teóricas do problema de cooperação encontradas no trabalhos de Galliers [GAL88], Castelfranchi [CAS95] e Jennings [JEN95]. No trabalho de Cohen e Levesque [COH89] são apenas considerados as crenças e objetivos.

Rao e Georgeff [RAO95] desenvolveram uma teoria baseada em três modalidades primitivas: crenças, desejos, e intenções (BDI). Esse trabalho baseia-se na noção de realismo que deduz a pergunta como através de uma crença de um agente sobre o futuro. Além disso, essa noção afeta os desejos e intenções do agente. No entanto, Rao e Georgeff consideraram a possibilidade de agregar planos ao seu formalismo. A descrição das crenças, desejos e intenções são feitas através dos operadores modais, da definição da semântica dos mundos possíveis para estes operadores e, de uma axiomatização definindo o inter-relacionamento e propriedades dos operadores BDI. Em contraste com a maioria das teorias filosóficas, Rao e Georgeff trataram intenções como um conceito com o mesmo estatuto de crenças e desejos. Isso permitiu a representação de diversos tipos de comprometimento racional baseados em diferentes propriedades sobre a persistência das crenças, objetivos e intenções. O mundo é modelado usando uma estrutura temporal com ramificações futuras e passadas denominadas de árvore do tempo. As situações são definidas como pontos de tempo particular num mundo particular. Pontos de tempo são transformados em outros pontos de tempo, através de eventos. Existem eventos primitivos e não-primitivos. Os planos são subdivididos numa estrutura hierárquica de subplanos e estes subdivididos, por sua vez, de forma mais refinada até alcançar as ações primitivas. Existe uma distinção entre escolha (a habilidade e uma agente deliberadamente selecionar uma ação de um conjunto de alternativas) e chance (incerteza associada a ocorrência das ações). A determinação do que ocorrerá é feita pelo ambiente. A linguagem formal utilizada para a descrição das estruturas é uma variação da “árvore Lógica Computacional” proposta por Emerson [EME89, Apud in MUL97].

¹⁸ Os mundos possíveis são representados por uma cadeia de eventos temporários que se estende de forma infinita (o passado e no futuro), caracterizando um possível caminho do mundo que pode ter sido ou será.

Existem dois operadores modais definidos na estrutura da formalismo: *opcional* e *inevitável*. Os operadores modais *always*, *eventually*, *next* e *until* são também definidos. Os operadores modais BEL, GOAL, e INTEND são introduzidos para crenças, objetivos e intenções, respectivamente.

O trabalho de Singh [SIN94], apresenta uma aproximação bastante diferente em que os agentes, são modelados através de uma família de lógicas para representar atitudes de informação (crenças e conhecimento) e pró-attitudes (conhecimento, perícia, e comunicação).

Werner [WER83] descreve um modelo geral de agentes mas o seu trabalho não foi investigado a fundo nesta tese, uma vez que não foram encontradas implementações baseadas no mesmo.

Wooldridge [WOO95] desenvolveu uma família de lógicas por representar as propriedades de MAS, porém não desenvolveu uma estrutura geral para teoria de agentes.

Wooldridge e Jennings [WOO95a] sugerem que a lógica de um agente precisa incorporar algum mecanismo para representar as ações do agente.

Segundo Móra[MOR99], os modelos formais tem dois papéis a apresentar, são eles:

- inicialmente devem servir como ferramenta para definir agentes;
- definir propriedades que estes agentes devem possuir e provar que os agentes modelados de fato possuem as propriedades desejadas.

Devem, também, fornecer subsídios para a implementação de agentes. Os modelos existentes são bastante adequados para a descrição formal de agentes e suas propriedades e para serem usados como ferramenta de verificação de propriedades que o agente possui. A razão para a distância existente entre a especificação e implementação está na escolha do formalismo utilizado para construir tais modelos. O caminho seguido no presente trabalho, possui um enfoque diferenciado dos trabalhos citados anteriormente. O modelo do agente que utilizamos parte da análise de Bratman [BRA89] sobre intenções, seu papel em raciocínio racional e sua relação à crenças e desejos, conforme explicado anteriormente. O modelo de Móra et al. (*X-BDI – Executable BDI*) não inclui um agente completo, mas sim um agente com sua estrutura cognitiva.

Seu detalhamento é apresentado no item a seguir, onde as descrições foram elaboradas utilizando o texto original de Michael Móra [MOR99]. Foram mantidas algumas descrições tal qual aparecem no volume, a fim de manter-se fidelidade às idéias apresentadas pelo autor.

4.2 O formalismo lógico utilizado na tese [MOR99]

A escolha do formalismo proposto por Móra et al. deve-se a dois fatores:

- é uma teoria de agentes computacional disponível para uso imediato. Embora questões como performance e eficiência não sejam consideradas, a partir das especificações utilizadas para formalizar o agente é possível executá-las e verificar os seus comportamentos; e

- é uma teoria que possui baixa complexidade para descrição dos agentes. Se define formalmente um agente apenas em termos dos seus estados mentais (crença, desejos e intenções). Além disso, se verifica se as propriedades que se deseja que o agente possua acontecem de fato.

Este segundo aspecto facilita a construção de outras aplicações utilizando a arquitetura proposta na nossa investigação, necessitando do projetista apenas o conhecimento da sintaxe dos comandos da linguagem do ambiente X-BDI.

Segundo Móra [MOR99], sua proposta disponibiliza um sistema formal cuja linguagem seja adequada para a representação de conhecimento. Esse sistema deve prover suporte para diversos tipos de raciocínios, tratados de forma computacional, além de fornecer as ferramentas necessárias para se modelar os vários estados mentais.

O ambiente base utilizado para a construção do X-BDI usa programação em lógica estendida com negação explícita – ELP (*Extended Logic Programming with explicit negation*) e com a semântica bem fundada estendida – WFSX (*Well-Founded Semantics eXtended for explicit negation*). Para este formalismo lógico, existe um procedimento de derivação descendente, o SLX (*Selected Linear resolution for eXtended programs*), que é completo e correto em relação a WFSX. Segundo [MOR99], esse procedimento permite provar que determinado literal pertence ou não ao modelo do programa, sem exigir o cálculo do modelo completo desse programa. Existe um algoritmo bem definido para o procedimento de revisão do mesmo, sobre o qual se baseiam os diversos tipos de raciocínio.

O requisito básico para a lógica selecionada é que ela possa ser utilizada tanto para a especificação como para implementação do sistema. E, de fato, esta é a maneira usual como a linguagem é vista na programação em lógica. A programação lógica estendida acrescenta uma segunda negação, a *negação explícita* à programação em lógica normal (utilizando as cláusulas de Horn aumentadas pelo operador *not*) O ELP estende a lógica tradicional do programa, obtida da negação habitual (chamada de *negação implícita* no contexto de ELP), a esta segunda negação denominada de *explícita*.

Essa extensão nos permite representar a informação negativa explicitamente (como uma convicção que uma propriedade que P não suporta, ou uma intenção que uma propriedade P não deveria suportar) e aumenta o poder expressivo da linguagem. Quando nós introduzimos a informação negativa, nós podemos lidar com os programas contraditórios [ALF96]. O X-BDI favorece que se lide com essas contradições de forma genérica. Assim, quando as contradições surgem, o ambiente fornece mecanismos para removê-las alterando o valor verdade de um subconjunto predefinido de literais. Tal mecanismo de remoção de contradições é obtido através de uma variante da SLX que computa as combinações possíveis de alterações de valores verdade que garantem a remoção. Através desse mecanismo de remoção é possível realizar diferentes tipos de raciocínio não-monotônicos.

A estrutura da ELP, além de prover procedimentos computacionais a prova de teorias que se expressam em sua linguagem, também provê um mecanismo para determinar como minimizar a remoção de contradições. Os benefícios da utilização de um modelo com estas características são providas pelo formalismo. Então, inicialmente são definidos esses três estados mentais e as relações estáticas entre eles, isto é constrangimento em consistência entre esses estados mentais. Depois, avança-se com a definição de aspectos dinâmicos de estados mentais, ou seja, como o agente escolhe suas intenções, e quando e como revisa suas intenções.

A lógica de programação estendida (ELP) é um conjunto de regras

$$H \leftarrow B_1, \dots, B_n, \text{not } C_1, \dots, \text{not } C_m \quad (m, n \geq 0)$$

onde $H, B_1, \dots, B_n, C_1, \dots, C_m$ são literais objetivos. Um literal objetivo é também um átomo A ou sua negação explícita $\neg A$. O símbolo *not* significa a negação por omissão ou negação por falha, e *not L* é uma literal por omissão. Literais podem ser literais objetivos ou literais omissão, sendo que $\neg \neg L \equiv L$.

A linguagem também permite tratar da consistência das restrições através de $A \leftarrow B_1, \dots, B_n, \text{not } C_1, \dots, \text{not } C_m$ ($m, n \geq 0$) onde $A, B_1, \dots, B_n, C_1, \dots, C_m$ são literais objetivos, instanciando que A pode suportar se o seu corpo $B_1, \dots, B_n, C_1, \dots, C_m$ suporta também. Particularmente, quando $A = \perp$, onde \perp instancia por *contradição*, isso significa uma *contradição* aparece quando o corpo da *contradição* é válida.

Quando raciocinamos sobre pró-attitudes, tais como desejos e intenções, necessitamos lidar com propriedades que possam ser suportadas em um certo instante de tempo e com ações que devem ser executadas num determinado tempo.

Entretanto, para representá-las e raciocinarmos a seu respeito, necessitamos de um formalismo lógico que lide com ações e tempo. Móra utiliza uma versão modificada do Event Calculus (EC) proposto por [MOR 95]. O predicado *holds_at(P,T)*, define que a propriedade P é verdadeira no tempo T . O predicado *happens(E,T)* significa que o evento E ocorreu num tempo T ; *initiates(E,P)* significa que o evento E inicia a propriedade P no tempo que o evento E ocorre; *terminates(E,P)* significa que o evento P terminou; *persists(Te,P,T)* significa que P persiste desde Te até T (pelo menos).

Existe uma variável especial denominada de *Now* que representa o tempo presente. Observe que a propriedade P é verdadeira num tempo T (*holds_at(P,T)*) se tiver um evento prévio que inicia em P e se P persiste até T . P persiste até T se não pode ser provado por omissão a existência de outro evento que termina P antes do tempo T .

Essas são as providências originais de EC pelo predicado *holds_at(P,T)* que nos permite argumentar sobre o futuro, assumindo uma sucessão de ações representada hipoteticamente pelos predicados *happens(E,T)* e *act(E,A)* e pela verificação das propriedades que os assegurariam. Também nos permite argumentar sobre o passado. Para saber se uma determinada propriedade P no momento T , o EC confere que propriedades permanecem válidas depois da execução das ações que aconteceram antes de T . o EC assume que as propriedades só mudam como consequência das ações executadas pelo agente. Essa não é uma suposição razoável, uma vez que nós gostaríamos de permitir a ação de outros agentes, bem como também permitir ações que não são notadas pelo agente. O papel de *sense(P,T)* significa que a propriedade de P é percebida pelo agente no momento T .

A figura 8, apresenta alguns aspectos da linguagem do X-BDI. Maiores detalhes podem ser obtidos em [MOR97; MOR98; MOR99].

$\neg \text{holds} \quad _ \text{AT} (P , T)$	\leftarrow	$\text{not holds} \quad _ \text{at} (P , T) .$
$\text{holds} \quad _ \text{at} (P , T)$	\leftarrow	$\text{initially} \quad (P) ,$ $\text{persists}(\quad 0 , P , T) .$
$\text{holds_at}(P \quad , T)$	\leftarrow	$\text{happens} \quad (E , T_e) ,$ $\text{initiates} \quad (E , P) ,$ $T_e < T ,$ $\text{persists} \quad (T_e , P , T) .$
$\text{holds} \quad _ \text{at} (P , T)$	\leftarrow	$\text{senses} \quad (P , T_s) ,$ $T_s < T ,$ $\text{persists} \quad (T_s , P , T) .$
$\text{persists} \quad (T_e , P , T)$	\leftarrow	$\text{not clipped} \quad (T_e , P , T) .$
$\text{clipped} \quad (T_e , P , T)$	\leftarrow	$\text{happens} \quad (I_e , T_{Ie}) ,$ $\text{terminates} \quad (I_e , P) ,$ $\text{not out} \quad (T_{Ie} , T_e , T) .$
$\text{out} \quad (T_{Ie} , T_e , T)$	\leftarrow	$T \leq T_{Ie} .$
$\text{out} \quad (T_{Ie} , T_e , T)$	\leftarrow	$T_{Ie} < T_e .$

FIGURA 4.1- Elementos da linguagem do X-BDI.

Segundo Móra [MOR99], ELP com o EC provê os elementos que são precisos para se modelar estados mentais. Não obstante, o uso de linguagem de mais alto nível permitiria ter uma descrição mais simples e mais clara de ações e seus efeitos. Assim sendo, Móra et al. lançam a linguagem de ELP e as primitivas de EC em uma sintaxe mais simples que é semelhante a linguagem ONE proposta por Gelfond e Lifschitz [GEL92, Apud in QUA97]. Nesse idioma são traduzidas as orações ELP e as primitivas de EC, durante a argumentação segundo [QUA97]. As orações são as seguintes:

- *A causa F if P_1, \dots, P_n* – ação A causa propriedade F se suporta a proposição P;
- *F after A if P_1, \dots, P_n* – suporta a propriedade F após a ação A se suporta a proposição P;
- *A occurs_in E* – ação A ocorre quando o evento E ocorre;
- *E preceeds E'* – o evento E ocorre antes do evento E'.

A linguagem ainda provê meios para referência de crenças que deveriam conter o futuro ou que deveriam ter dentro de si. Isso é feito através do operador *next(P)* declarando que a propriedade P deveria ser suportada no próximo momento de tempo. Uma vez tendo o formalismo lógico fixado, pode-se definir o modelo de BDI.

Construída a base formal necessária, passa-se à definição dos estados mentais. Inicialmente são definidos os aspectos estáticos do modelo, ou seja, quais são os três estados mentais básicos, seu conteúdo propocisional, e as condições e restrições que tais estados devem satisfazer. Um exemplo disso, é considerar que o conjunto de crenças e intenções devam ser consistentes e as respectivas crenças também.

Após definir-se os aspectos estáticos, definem-se os aspectos dinâmicos dos estados mentais, como por exemplo, como são criadas as intenções, como são produzidas as ações a partir das intenções (no caso do tutor do MCOE as ações estão restritas à execução de um plano), como são resolvidos os conflitos nos desejos, etc.

A arquitetura se apoia na mesma preocupação central do ambiente X-BDI, ou seja, o comportamento dos agentes a partir dos estados mentais pró-ativos diretamente ligados a produção de ações.

O modelo possui duas características :

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

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

4.3 Os agentes cognitivos modelados com arquitetura BDI

A estrutura do agente cognitivo é uma tupla em que B é o conjunto de crenças do agente, D é o conjunto de desejos do agente, I é o conjunto das intenções do agente e T é o conjunto de axiomas de tempo, como definido acima. Os desejos do agente são um conjunto de orações $DES(Ag,P,Atr)$, em que Ag é a identificação de agente, P é uma propriedade e Atr é uma lista de atributos.

Desejos são relacionados eventualmente ao estado de mundos que o agente quer provocar. Na abordagem utilizada neste trabalho, desejos não dirigem necessariamente o agente para agir. Isto é, o fato de um agente ter um desejo não significa agir para o satisfazer. Significa que antes de um determinado agente decidir o que fazer, ele passa por um processo de racionalização e confronta os seus desejos (o estado de eventos que quer provocar) com suas convicções (as circunstâncias atuais e restrições que o mundo impõe). O agente escolherá os desejos que são possíveis de acordo com algum critério. No caso deste trabalho, o agente escolhe os desejos em função das crenças válidas no momento a respeito do ambiente.

Crenças constituem a informação sobre a atitude do agente. Elas representam as informações que o agente tem sobre o ambiente e sobre si próprios. O conjunto B contém orações que descrevem o domínio do problema e usam ELP. Um agente A acredita em uma propriedade P , num tempo T se, a partir de B e T , o agente pode deduzir $BEL(Ag,P)$ durante o tempo T . Assume-se que o agente atualiza continuamente

suas crenças para refletir mudanças que detecta no ambiente e que, sempre quando uma nova crença é somada ao conjunto de crenças, a consistência é mantida.

Intenções são caracterizadas por uma escolha de um estado de eventos a alcançar. Assim, as intenções são vistas como um compromisso que o agente assume com um específico futuro possível. Isso significa que, diferentemente dos desejos, uma intenção não pode ser contraditória com outras intenções. Além disso, significa que não seria racional para um agente agir para alcançar estados incompatíveis. Intenções deveriam ser apoiadas pelas convicções do agente, i.e., não seria racional para um agente pretender algo que não acredita ser possível. Uma vez que uma intenção é adotada, o agente procurará satisfazer aquela intenção e planejará ações para realizá-la. O replanejamento de ações ocorre sempre quando um fracasso acontece. Essas ações também devem ser adotadas como intenções por parte dos agentes.

A definição de intenções obriga criarmos restrições para a racionalidade: um agente não deveria pretender algo num tempo passado; um agente não deveria pretender algo que acredita já estar satisfeito ou que será satisfeito sem esforços por parte do agente; e um agente só pretende algo que acredita ser possível de ser alcançado. Nesse último caso, pode haver um curso de ações que conduzem ao estado intencional associado ao evento. Quando projetamos um agente, nós especificamos as suas crenças e os seus desejos. O agente é que vai escolher apropriadamente suas intenções a partir de seus desejos. Essas restrições de racionalidade também devem ser garantidas durante esse processo de seleção [MOR98].

Agentes escolhem as intenções de duas fontes diferentes: de seus desejos e de um refinamento de outras intenções. Por definição, não há nenhuma restrição nos desejos do agente. Então, um agente pode ter desejos contraditórios (conforme já explicado anteriormente). Por outro lado, intenções são restringidas através de restrições de racionalidade. Assim, os agentes têm que selecionar só os desejos relacionados com aquelas restrições. Primeiro, é necessário determinar esse subconjunto dos desejos que são pertinentes de acordo com as convicções atuais do agente. Depois, é necessário determinar desejos que são conjuntamente realizáveis. Em geral, pode haver mais de um subconjunto de desejos pertinentes. Nós deveríamos indicar de alguma maneira os subconjuntos preferidos e adotá-los como intenções através de uma relação de preferência definida nos atributos dos desejos. De acordo com a teoria definida em [MOR97, MOR98], o agente deveria preferir satisfazer os desejos mais importantes primeiro. Para que isso ocorra, o agente adota a maior quantidade de desejos que puder. A seleção que combina as diferentes formas de raciocínio não-motônico são providas pelo formalismo lógico.

Assim que o agente adotar suas intenções, ele começará a planejar como realizá-las. Durante o planejamento, o agente formará as intenções que são relativas ou preexistentes, significando que as intenções existentes são refinadas. Esse refinamento pode ser feito de vários modos, como por exemplo, um plano que inclui uma ação que não é diretamente executável envolve um conjunto de ações com uma ordem temporal [KON93, Apud in MOR99]. Com o agente se compromete com as intenções adotadas, as intenções prévias constroem a adoção de novas. Ou seja, durante a elaboração de planos, é adotada só uma nova intenção potencial não contraditória com as intenções e com crenças existentes .

O próximo passo é definir quando o agente deve executar o raciocínio sobre as intenções. Não é suficiente declarar que um agente deve revisar suas intenções quando acredita que acontece uma certa condição, assim como acreditar que uma intenção foi

satisfeita ou que não é possível satisfazê-la. O que o agente precisa fazer é verificar constantemente suas crenças. Ao invés de instanciar as intenções, é necessário definir um mecanismo que ative o processo de raciocínio sem impor um fardo adicional ao agente. A aproximação disponibilizada por Móra et al. permite definir as condições que fazem o agente começar a raciocinar sobre as intenções como restrições em cima de suas crenças. Sempre que fatos novos são incorporados, o agente tem que manter constantemente suas crenças consistentes. Da mesma forma, sempre que uma contradição aparece, o agente revisa suas crenças. O processo de revisão de intenções é ativado quando uma das restrições é violada.

4.4 A coreografia dos estados mentais

A coreografia que ocorre na interação entre os dois alunos e o tutor se baseia no conjunto de ações dos alunos e nas reações do tutor em face ao conjunto de informações que o tutor recebe dos alunos e do ambiente. Tais informações são passadas ao tutor através das ações de cada aluno que envolve a externalização do diálogo existente entre eles e reflete a estratégia de ação acordada naquele momento pelos dois.

Durante uma partida, que consideramos uma seção de trabalho entre os alunos e o tutor, são inicialmente carregados para o “kernel” cognitivo o conjunto de informações a respeito do ambiente (níveis de energia, quantidade de seres vivos, etc.) que serão utilizados pelo tutor para desencadear o processo de acompanhamento dos alunos e do ambiente. Depois dos alunos configurarem o ambiente com todas as rotinas necessárias (maiores detalhes no anexo 4), a partida é iniciada. A cada ação dos alunos, o conjunto de estados mentais associados à ação é enviado ao “kernel”. Essas informações são passadas ao tutor através de uma cadeia de strings onde estão os estados mentais associados, escritos no formalismo computável de Móra et al. (maiores detalhes sobre a implementação dessa interface entre o ambiente e o kernel se encontram no anexo 5). Cada vez que uma ação é executada, as informações acerca do ambiente são também enviadas para o kernel. Dessa maneira, o tutor possui sempre um conjunto de informações atualizadas que lhe permitem ter um histórico do ambiente e de cada aluno.

A fim de ilustrar a dinâmica da coreografia, selecionamos um diálogo pequeno que é complexo bastante para mostrar como o framework lógico é utilizado.

O sistema coloca muita poluição na tela e cria uma situação complexa. Os elementos poluidores são 2 dragas, 2 lanchas, 2 esgotos industrial, 1 poluição química e 2 postos de gasolina. Esses elementos aparecerão ao longo do jogo em intervalos pré-fixados no arquivo de configuração (vide anexo 4 para maiores detalhes sobre essa configuração).

O aluno 1 escolhe o personagem Prefeito e o aluno 2 escolhe a Mãe Natureza. Tanto o Prefeito, como a Mãe Natureza são personagens com bastante recursos para combater a poluição. No princípio, o sistema ativa a draga e o posto de gasolina. A energia apresenta diminuições niveladas: Plantas -40%, Microorganismo -50%, Assente -50%, Pesque -20%, Molhe -20%. Nesse momento, o ecômetro mostra o nível de energia e a cor de algumas mudanças de elementos de paisagem.

O tutor tem somente um objetivo: ajudar o estudante a manter um nível de energia adequado no ambiente. Acredita que pode ajudar enviando as mensagens aos alunos. Os conteúdos dessas mensagens dependerão da estratégia adotada no momento pelo tutor. Esses desejos e crenças são modelados como se segue :

DES(tutor,ajudar_alunos).
BEL(tutor,ajudar_alunos) if
BEL(tutor, envia_mensagem).

Uma vez que esse é o único desejo do tutor, ele torna-se o único candidato à intenção. No entanto, será adotado como intenção, se o tutor acreditar que há uma sucessão de ações que possam ser executadas para que essa intenção se realize. Neste momento, o tutor dependerá do conhecimento sobre os estados mentais dos alunos para decidir o que fazer. Como ainda ele não dispõe dos conhecimentos estes estados mentais, ele aguarda. Enquanto isso, ele insere em seus gatilhos (*triggers*) de crenças suas opções que farão com que seja reconsiderado quando a interação acontecer. Por exemplo, o tutor tem a seguinte crença:

BEL(tutor, envia_mensagem) if
BEL (tutor,next(BEL (aluno, receber_ajuda))),
BEL(energia_nivel_ecometro >= 70),
Mensagem (16).

(1)

Acredita-se que o estudante espera um pouco de ajuda e que o nível de energia está sobre 70. A única ajuda que pode oferecer é aconselhar o estudante a prestar atenção no nível de energia. Essa crença colocaria o gatilho seguinte:

$\perp \leftarrow BEL(tutor,next(BEL(aluno,rceber-ajuda)))$
BEL(energia_ecometro (Ee)),
Ee >= 70.

(2)

Quando as pré-condições para a ação de enviar uma mensagem 16 forem satisfeitas, o tutor estará em condições de ajudar o aluno. O modelo do agente transforma o desejo em uma intenção e a satisfaz enviando a mensagem 16.

Uma vez que o tutor tem essa intenção, ele usa suas crenças sobre o ambiente (ferramenta, energia dos elementos do cenário, poluidores), sobre como lutar contra a poluição (estratégias diferentes no uso das ferramentas para eliminar poluição), sobre como aconselhar os alunos para ajudá-los a controlar o ambiente e assim por diante.

Vamos analisar agora a coreografia tomada em tempos diferentes. Observe que o tempo utilizado no X-BDI é o tempo em que ocorre uma ação, aquele em que acontece algo novo enviado ao “kernel”. Não é utilizado um sistema de tempo de horas, minutos, segundos, etc.

(Tempo Ponto 1)

Aluno 1: A poluição começou e eu aplicarei a ferramenta multa.

O conjunto de estados mentais mostra a expectativa de que "o nível de energia crescerá" e "a ferramenta multa combate a draga" .

Aluno 2: Eu acelerarei a degradação.

O conjunto de estados mentais mostra a expectativa de que "o nível de energia crescerá." "A ferramenta acelera o combate do esgoto industrial"

Tutor: Verifica que os alunos não mudaram suas crenças e o nível de energia aumentou. Fica observando e não envia nenhuma mensagem.

Nenhum dos gatilhos fixados pelo tutor é ativado por esta interação. Então, ele não adota qualquer intenção e espera.

(Tempo Ponto 2)

O nível de energia aumentou mas não é 100%.

Aluno 1: Olhar isto! Tudo vai bem!

Aluno 2: Sim está funcionando!

Tutor: Espera.

Nesse momento, os alunos não fazem nenhuma ação, pois o ecômetro está no nível de energia que não indica perigo ou necessidade de intervenção por parte do tutor. O tutor acredita que quando os alunos não fazem nenhuma ação e o nível do ecômetro esta acima de 70, ele deve aguardar.

(Tempo Ponto 3)

Então aparecem os poluidores: a draga, a lancha e o lixo industrial. O nível de energia dos elementos do cenário decresce na seguinte proporção: plantas -40%, microorganismo -40%, fundo -50%, peixe -30%, água -20%. Os alunos observam a mudança imediata na cor do ecômetro e o comportamento dos elementos do cenário.

Aluno 1: Eu usarei remove a licença para aumentar a energia rapidamente nivelada.

O conjunto de estados mentais apresenta as seguintes expectativas: "a energia do ecômetro subirá", "a ferramenta licença combate draga e lixo industrial",

Aluno 2: Eu intensificarei o raio de sol, porque colocará mais energia rapidamente.

O conjunto de estados mentais apresenta as seguintes expectativas: "a energia do ecômetro subirá", "a ferramenta intensifica combate draga e lixo industrial".

Tutor: Verifica que o nível de energia aumentou e os alunos não estão usando a regra para controlar o número de vezes que uma ferramenta pode ser usada.

Os estados mentais dos alunos não apresentam a crença " número de ferramentas deveria ser controlado ". Esse estado mental é associado a cada ação do aluno em verificar a quantidade de ferramentas restantes para cada personagem. O tutor, então, emite uma mensagem de alerta para os dois alunos nessa situação.

O Tutor executará essa ação quando a quantia restante de alguma das ferramentas é igual ao valor 1. No conjunto de crenças do tutor aparece a crença " número de ferramentas deve ser controlado " e isso ativa o desejo " controlar as ferramentas".

Neste passo, o tutor só analisa a interação. De fato, de acordo com o modelo de agente, o tutor ampliou pouco suas crenças. Mas, tais crenças novas não ativam nenhum dos gatilhos. Isso significa que nenhum dos desejos do tutor é pertinente. De fato, não é considerado racional agir para alcançar intenções se as crenças refletem que a intenção será satisfeita sem qualquer intervenção. Nesse caso, se o tutor pensa que os alunos estão executando bem, não seria racional fornecer qualquer conselho a eles.

Esse é o comportamento habitual quando o modelo de agente é usado. O tutor mantém seu conjunto de crenças consistentes. Quando a mudança no conjunto de crenças ativa algum gatilho, o agente tentará adotar intenções, construir planos e assim por diante. Nesse caso, a possibilidade para construir planos não é usada. De fato, o tutor deve ter um plano de ação que leve em consideração as interações, especialmente a última interação. Ele é um tutor cognitivo que apresenta de alguma maneira um comportamento reativo. Não é uma simples aplicação de regras de reação ao se levar em conta o histórico da interação e a expectativa sobre a próxima interação. Além de ser interessante do ponto de vista computacional [MOR 98a], também respeita a abordagem pedagógica que é utilizada, i.e., a abordagem construtivista [GIR97].

(Tempo Ponto 4)

Nesse momento, mais poluidores aparecem: 1 posto de gasolina e 1 esgoto industrial. O nível de energia decresce na seguinte proporção: Plantas -65%, microorganismos -35%, Fundo -65%, Peixes -25%, Água -80%.

Aluno 1: O lago está morrendo. Eu não tenho a ferramenta mais intensa e mais forte, eu tenho a ferramenta acelerar. Eu vou jogar com esta. E tu deves jogar com a tua mais poderosa.

O conjunto de crenças apresenta: " ferramentas fortes devem ser usadas para o nível de energia crescer rápido", "se o número de uma ferramenta é igual a zero então a ferramenta está indisponível ", "devo controlar o número de vezes que posso usar uma ferramenta " e, "devo controlar o tempo do jogo ".

O desejo " aluno 2 cooperar com aluno 1 " se torna uma intenção. Existem as seguintes expectativas: " aluno 2 vai cooperar " e " a energia do ecômetro vai subir".

Total de energia depois desta intervenção: Plantas -50%, microorganismos -20%, Assente -50%, Pesque -10%, Molhe -65%.

Aluno 2: Está bem, eu usarei proibição. O nível de energia aumentará.

O conjunto de estados mental apresenta as crenças: " ferramentas fortes devem ser usadas para o nível de energia crescer rápido", "se o número de uma ferramenta é igual a zero, então a ferramenta está indisponível ", deveria controlar o número de tempos que podem usar uma ferramenta " e, "controlar o número de vezes que posso usar uma ferramenta ".

O desejo "cooperar com o aluno 1" é selecionado como intenção. Existem as expectativas: " aluno 2 vai cooperar " e " a energia do ecômetro vai subir".

Total de energia no tempo depois dessa intervenção: Plantas -35%, microorganismos -10%, Assente -40%, Pesque -5%, Molhe -55%.

Tutor: Verifica que os dois alunos estão cooperando. Verifica o tempo restante do jogo e os níveis de energia. Quando a cooperação existe, o Tutor tende a reforçar o comportamento dos alunos. De fato, o tutor tem uma crença sobre a cooperação existente entre os alunos:

BEL(tutor,aluno_cooperando) if BEL(tutor,envia_mensagem).

BEL(envia_mensagem) se BEL(tutor, before(BEL(tutor,suspende(aluno))))),

Nesse momento, o gatilho (2) conectou-se com crença (1) e esta é ativada. Então, o tutor adota uma intenção (ajudar os alunos) que será satisfeita enviando uma mensagem, nesse caso uma mensagem de reforço.

O diálogo procede desse modo até a partida terminar. Ao final do tempo previsto, os alunos controlam a poluição com o auxílio do tutor.

O capítulo a seguir, trata dos aspectos relativos ao agente cognitivo tutor sendo abordados os aspectos de IE e IA da sua arquitetura e detalhando como estão expressos os estados mentais do tutor. Serão utilizados como exemplos, alguns dos estados mentais citados na coreografia acima.

5. O tutor modelado através de estados mentais

A proposta da arquitetura do tutor leva em consideração o paradigma de ensino-aprendizagem centrado no aluno. O que se deseja é aplicar técnicas e procedimentos da IA para ampliar as possibilidades de adaptação às necessidades dos alunos no ambiente educacional. O perfil do tutor deverá ser capaz de inferir procedimentos a partir de um conjunto de heurísticas pré-definidas baseado na experiência e na observação de alunos trabalhando no domínio específico.

A diferença dessa arquitetura para as demais é a apresentação de um modelo do aluno construído de forma dinâmica, interativa e em tempo real. Nesse sentido, o tutor não utilizará os seguintes critérios:

- O conjunto de estereótipos para classificar o aluno;
- A medida para diferenciar um aluno ideal e aquele que está interagindo com o sistema para determinar como irá se comportar com ele; e
- O conjunto de erros inferidos para avaliar o quanto o aluno está ou não alcançando os objetivos do sistema.

Por outro lado, o tutor utilizará as seguintes ferramentas:

- As informações armazenadas referentes ao próprio aluno ao longo de cada interação desde o início do jogo;
- O conhecimento que possui sobre o domínio; e
- As informações que recebe do ambiente.

A figura 5.1 apresenta a arquitetura do tutor que será detalhada a seguir.

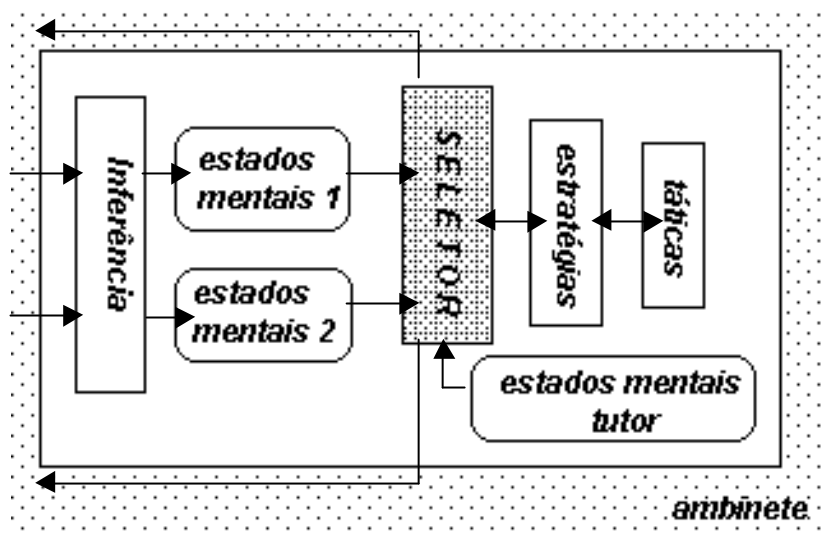


FIGURA 5.1- Arquitetura do tutor

5.1 Estratégias de ensino do tutor

A seleção do conjunto de estratégias de ensino que constituirão o STI é muito importante para garantir a qualidade pedagógica do ambiente. A seleção de uma estratégia depende de vários fatores, tais como: o nível de conhecimento do estudante, o domínio, a motivação e as características afetivas do mesmo. A utilização da mesma estratégia não produz um efeito satisfatório para todos os estudantes. Segundo Frasson [FRA97], cada estratégia tem vantagens específicas. É muito útil saber qual a estratégia que fortalecerá adequadamente o processo de aquisição para um determinado estudante. Professores competentes apresentam o mesmo material de modos diferentes. As situações em que isso pode ocorrer não são previsíveis (até mesmo se o domínio for). O que se pode fazer é procurar diversificar o auxílio oferecido ao aluno levando em consideração os indicadores retirados do próprio trabalho pregresso do estudante.

A construção de um STI com múltiplas estratégias é muito útil para se aumentar a qualidade pedagógica de tais ambientes. No entanto, a aquisição de conhecimento sobre multiestratégias pedagógicas de forma separada da aquisição de perícias de domínio ainda é um grande desafio para os projetistas de STI.

As estratégias pedagógicas são um conjunto de regras e/ou planos para alcançar metas específicas. Keller [KEL87] e Murray [MUR97] sugeriram que o tutor deve ter dois níveis de planejamento: táticas pedagógicas e estratégias pedagógicas. As estratégias pedagógicas têm o conhecimento sobre como ensinar e as táticas contêm as ações para efetivar a estratégia selecionada.

Os STI apresentam diferentes tipos de estratégias conforme abordado no capítulo 2. Muitos dos sistemas citados na literatura apresentam mais de uma característica conforme a classificação apresentada naquele capítulo. Essa forma adotada pelos projetistas se justifica pela múltipla escolha de objetivos, de princípios educacionais diversificados, de métodos diferentes para a estruturação do conhecimento, de vários formatos instrucionais para áreas de assuntos variados e de outras possíveis variantes. É muito raro achar um STI puro na sua concepção pedagógica. Isso acontece devido à natureza interdisciplinar desses ambientes e a forma heterogênea do processo de ensino-aprendizagem. O STI possui um processo dinâmico que altera situações de maior e menor controle por parte do tutor.

No final dos anos 80, enfatizou-se os aspectos pedagógicos e incrementou-se a participação do estudante no processo de ensino-aprendizagem. Essa nova geração de STI é conhecida como Sistemas de Aprendizagem Cooperativos ou Sistemas de Aprendizagem Social. A arquitetura de tais sistemas começou a mudar com essa nova abordagem e foi reforçada pelo uso das técnicas de agente para sua modelagem e implementação.

O uso de arquiteturas Multi-agente favoreceu os projetistas a substituir os módulos funcionais (abordagem tradicional) por agentes ou sociedade de agentes, citamos como exemplos os trabalhos de [AND97;COR94;COS97;DAM97;FRA97;MOU96;MUR97]. Apesar disso, estes sistemas ainda mantêm a arquitetura básica comum. Eles têm uma base de domínio, um modelo de estudante, as estratégias tutoriais e o controle distribuído (em maior ou menor grau).

O tutor possui três estratégias de comportamento (guia, reativo e assistente). Cada estratégia traz implícita um determinado grau de interferência no trabalho do aluno. Quando o tutor é um guia, ele sugere de forma mais direta o que o aluno deve fazer e

sugere ações específicas, independente do estado dos elementos do ambiente. Ele se comportará assim, caso ele perceba (através do conjunto de estados mentais e ações) que o aluno não possui uma linha de trabalho e/ou não possui um entendimento do que fazer dentro do jogo.

Um ambiente construtivista é por concepção menos diretivo e menos controlador das ações do aluno. O controle existe na forma de monitoração para que o tutor funcione como um parceiro e um facilitador do trabalho do aluno. Porém, devido às características da modalidade escolhida (jogo educacional), precisamos ter algumas atitudes no tutor que garantam que o sistema não entre em colapso e não inviabilize o trabalho do aluno. Nessas situações críticas, o tutor vai se comportar de maneira mais diretiva. Cabe salientar que a abordagem construtivista não significa dar liberdade total ao aluno sem nenhuma assistência. O que muda é o grau de interferência do tutor, i.e., o quanto ele interfere no trabalho do aluno e permite ou não que o aluno siga um caminho alternativo àquele que ele tem como o ideal para resolver o problema (conjunto de regras do tutor sobre o problema e forma de solução).

Na estratégia reativa, o tutor se comporta em função dos pontos críticos (problemas) que vão surgindo. Cada problema gera um conjunto de ações selecionadas (i.e. táticas) para ajudar os estudantes a construir as próprias soluções para um determinado problema. O estudante é induzido a pensar em possibilidades de solução através do trabalho cooperativo com o seu colega. Na estratégia assistente, o tutor se comporta como um parceiro que sugere ações de forma menos invasiva e procura reforçar o aluno a descobrir caminhos e refletir sobre o que está acontecendo no sistema.

Como o ambiente é um jogo há a restrição do tempo. O fato de existir um tempo determinado para as ações ocorrerem fez com que os comportamentos do tutor ficassem associados ao controle do tempo, do estado do ambiente e das ações do aluno. Quando existe tempo suficiente para o aluno explorar as possibilidades de combinações com as ferramentas, o tutor tende a se comportar como assistente. À medida que o tempo corre, ele tende a ser mais diretivo e reagir conforme os problemas do ambiente. No caso de um colapso eminente, o tutor se torna mais diretivo e passa a guiar as ações do aluno. Essas prioridades podem mudar caso o tutor perceba que o aluno está com problemas (crenças conflitantes) ou apresenta a intenção explícita de ser guiado pelo tutor. A TABELA 5.1 apresenta exemplos de regras associadas ao tipo de mensagem utilizada pelo tutor.

As três estratégias utilizadas pelo tutor se associam a um conjunto de táticas não excludentes, i.e., existem táticas que são compartilhadas por mais de uma estratégia. Isso se deve ao fato de algumas táticas funcionarem de forma diferente conforme o contexto em que são utilizadas. Segundo [PIL96], o bom professor não usa sempre a mesma estratégia, mas pode repetir táticas num contexto diferente fazendo com que o seu resultado se modifique. A figura 5.2 apresenta essa composição de forma esquematizada.

A utilização de multiestratégias adicionou um grau de complexidade ao tutor. Enquanto, a utilização do conjunto de estados mentais associados às ações de cada aluno permitiu que se trabalhasse com uma maior granularidade do processo de orientação personalizada à cada aluno.

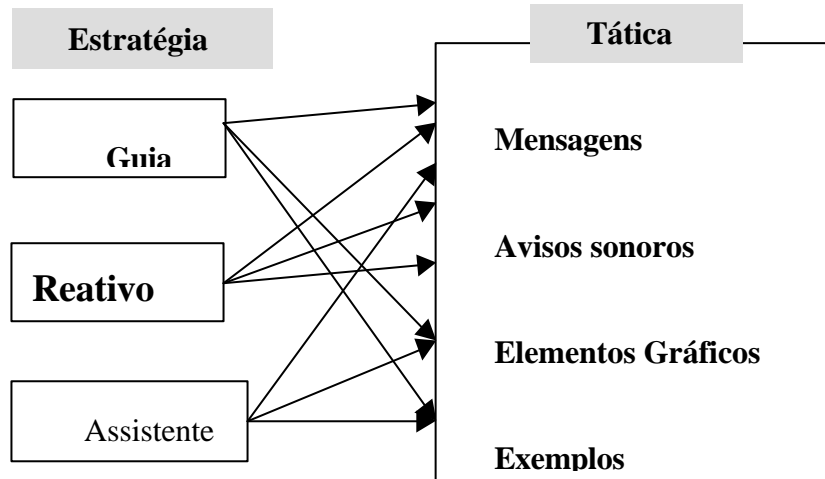


FIGURA 5.2 - Estratégias e Táticas do MCOE

É importante salientar que, mais do que uma questão metodológica, a seleção de estratégias e táticas de ensino está diretamente ligada à didática a ser empregada o ensino de cada conteúdo. Cada área do conhecimento possui suas peculiaridades; assim podemos ter uma metodologia para trabalhar com vários conteúdos, utilizando uma didática diferenciada com relação à natureza dos mesmos [BAL97].

A aplicação escolhida determinou a seleção deste conjunto específico de estratégias e táticas, porque essas foram julgadas mais adequadas pelo especialista que auxiliou a modelagem da base de domínio [RAA96].

TABELA 5.1- Exemplos de mensagens do tutor

Número	Conteúdo	Situação
1	Verifique a energia de cada elemento do jogo	O Ecômetro está com nível de energia superior a 50%, porém existe um elemento do cenário com energia crítica (vide regras de cada elemento) Tempo de jogo superior a 1 minuto.
2	Verifique o nível de energia dos peixes	O nível de energia dos peixes está abaixo do normal e o tempo de jogo superior a 1 minuto.
3	Verifique o nível de energia das plantas	O nível de energia das plantas está abaixo do normal e o tempo de jogo superior a 1 minuto.
4	Verifique o nível de energia dos microorganismos	O nível de energia dos microorganismos está abaixo do normal e o tempo de jogo o tempo superior a 1 minuto.
5	Verifique o nível de energia do fundo do lago	O nível de energia do fundo do lago está abaixo do normal e o tempo superior a 1 minuto.
6	Verifique o nível de energia da água	O nível de energia da água está abaixo do normal e o tempo de jogo superior a 1 minuto.
7	Verifique o número de ferramentas restantes	Existe uma ferramenta que está com pouca ou apenas uma opção de uso e o tempo de jogo superior a 1 minuto.
8	Cuidado uma ferramenta não poderá ser mais utilizada	Existe uma ferramenta que está com pouca ou apenas uma opção de uso e o o tempo de jogo superior a 1 minuto.
9	Consulte o sistema de ajuda e leia as regras do jogo e de seu funcionamento	O aluno não utiliza nenhuma ferramenta e ecômetro está com energia < 60
10	Use a ferramenta que coloca mais energia no cenário	O tempo de jogo superior a 2 minutos .e os níveis de energia do ecômetro estão abaixo de 50%.
11	Você não pode usar mais uma das ferramentas	Uma das ferramentas está zerada
12	Você não pode usar mais duas das ferramentas	Duas das ferramentas estão zeradas
13	Você só pode usar uma das ferramentas	Três ferramentas estão zeradas
14	Você não pode mais jogar	Todas as ferramentas estão zeradas
15	Observe o que acontece no lago e consulte o sistema de ajuda sobre as regras do jogo	Todas as ferramentas estão zeradas e ainda ressa tempo de jogo superior a 2 min
16	Você está jogando bem, continue atento ao jogo	Energia do ecômetro > 70 e aluno deseja receber ajuda
17	Procure trocar idéias com o seu colega	Somente um aluno envia mensagens e o outro não envia nenhuma mensagem de volta

5.2 As estratégias de ensino do tutor e sua representação no formalismo

O tutor possui um conjunto de crenças e desejos a respeito do ambiente, dos alunos e do seu comportamento. A fim de exemplificar como tais crenças são expressados no formalismo, selecionou-se algumas crenças relacionadas com o diálogo apresentado no item 4.4 do capítulo anterior.

Exemplos de regras de funcionamento da simulação:

bel(tutor, e_um_personagem(prefeito)).

bel(tutor, e_um_personagem(maenatureza)).

bel(tutor, e_um_elementocenario(Peixe)) if

bel(tutor, peixe(Peixe)).

bel(tutor, e_um_elementocenario(Planta)) if

bel(tutor, planta(Planta)).

bel(tutor, e_um_elementocenario(micro)).

bel(tutor, e_um_elementocenario(fundo)).

bel(tutor, e_um_elementocenario(agua)).

bel(tutor, e_um_elementocenario(ecometro)).

bel(tutor, retira_energia(draga)).

bel(tutor, retira_energia(lixourbano)).

bel(tutor, retira_energia(lancha)).

bel(tutor, retira_energia (esgotoindustrial)).

bel(tutor, retira_energia (produtostoxicos)).

bel(tutor, prefeito(lixeias)).

bel(tutor, prefeito(proibicao)).

bel(tutor, prefeito(multa)).

bel(tutor, prefeito(licenca)).

bel(tutor, maenatureza(acelera)).

bel(tutor, maenatureza(intensifica)).

bel(tutor, maenatureza(trocafundo)).

bel(tutor, maenatureza(adicionaplanta)).

bel(tutor, energia(ecometro,Ee)) if
bel(tutor, energia(peixeA, PA)),
bel(tutor, energia(peixeB, PB)),
bel(tutor, energia(peixeC, PC)),
bel(tutor, energia(plantaA, Pa)),
bel(tutor, energia(plantaB, Pb)),
bel(tutor, energia(micro,M)),
bel(tutor, energia(fundo,F)),
bel(tutor, energia(agua, A)),
Ee is (PA+PB+PC+Pa+Pb+M+F+A)/8.

estrutura de funcionamento das ferramentas de cada personagem:

bel(tutor,energia(plantaA,N)) if
before(bel(tutor,multa)),
*N is N+(N * 0.15).*

bel(tutor,energia(plantaA,N)) if
before(bel(tutor,licenca)),
*N is N+(N * 0.15).*

bel(tutor,energia(plantaA,N)) if
before(bel(tutor,acelera)),
*N is N+(N * 0.2).*

bel(tutor,energia(plantaA,N)) if
before(bel(tutor,intensifica)),
*N is N+(N * 0.15).*

como as ferramentas e poluidores afetam os elementos do cenário:

bel(tutor,energia(peixeA,N)) if
before(bel(tutor,gasolina)),
*N is N-(N * 0.2).*

bel(tutor,energia(peixeB,N)) if
before(bel(tutor, draga)),
*N is N-(N * 0).*

- desejos do tutor em relação ao controle de cada elemento do cenário

des(tutor, controlar_energia(X)).

bel(tutor, controlar_energia(Peixe)) if
bel(tutor, peixe(Peixe)),
bel(tutor,energia(peixeA, PA)),
PA<100.

bel(tutor, controlar_energia(Peixe)) if
bel(tutor, peixe(Peixe)),
bel(tutor,energia(peixeB, PB)),
PB<100.

bel(tutor, controlar_energia(Peixe)) if
bel(tutor, peixe(Peixe)),
bel(tutor,energia(peixeC, PC)),
PC<100.

bel(tutor, controlar_energia(Planta)) if
bel(tutor, planta(Planta)),
bel(tutor,energia(plantaA, Pa)),
Pa<100.

bel(tutor, controlar_energia(Planta)) if
bel(tutor, planta(Planta)),
bel(tutor,energia(plantaB, Pb)),
Pb<100.

bel(tutor, controlar_energia(agua)) if
bel(tutor,energia(agua, A)),

$A < 100$.

bel(tutor, controlar_energia(micro)) if

bel(tutor, energia(micro, M)),

$M < 100$.

bel(tutor, controlar_energia(fundo)) if

bel(tutor, energia(fundo, F)),

$F < 100$.

O tutor guia o seu comportamento baseado nas informações que ele recebe do ambiente (valores relativos à energia dos elementos do cenário, do tempo e dos estados mentais associados às ações dos alunos). Enquanto essas informações são verificadas, o conjunto de crenças que suporta os desejos candidatos torna-se o conjunto de intenções. No caso do tutor do MCOE, o conjunto de planos (ações) associados a uma intenção se restringe a um. Ou seja, o tutor vai escolher uma ação a realizar ao tomar uma decisão.

Um agente cognitivo que apresenta atitudes de tutor, conforme Corrêa, Coelho e Viccari [COR98], possui um conjunto de estados mentais básicos que o caracteriza. Por exemplo, no caso do tutor do MCOE, ele possui desejos de corrigir o aluno, de cooperar com ele, de verificar o seu conhecimento, desejos esses típicos de um agente dessa natureza. Isto é exemplificado a seguir:

des(tutor, cooperar(Aluno)).

bel(tutor, cooperar(Aluno)) if

bel (tutor, INT_THAT(Aluno, receber_auxilio)).

bel(tutor, cooperar(Aluno)) if

bel(tutor, tempo(T)),

$T > 0$.

des(tutor, verificar_conhe(Aluno)) if

bel(tutor, energia(ecometro, Ee)),

$Ee < 100$.

des(tutor, corrigir_conhe(Aluno)) if

bel(tutor, energia(ecometro, Ee)),

$Ee < 70$.

O agente tutor deve ser preparado a enfrentar as situações típicas que ocorrem no processo de ensino-aprendizagem que devem ser previstas pelo projetista da aplicação.

No contexto do MCOE podemos citar como exemplo a situação em que o aluno não realiza nenhuma ação. Determinar o que significa no contexto do trabalho “o aluno não fazer nenhuma ação” envolve uma análise mais complexa dessa atitude. A heurística a ser utilizada para compor o conjunto de ações possíveis do tutor nesse contexto está diretamente ligada à natureza da aplicação. Como objetivo do jogo é manter o equilíbrio ecológico do sistema, o aluno vai constantemente monitorar os níveis de energia que serão os indicadores do que está acontecendo no ambiente. Essa monitoração pode levar o aluno a uma atitude de espera para verificar as conseqüências de suas ações (e do seu colega), representada por:

bel(tutor, nao_faz_nada) if
bel(tutor, aluno_suspende).

bel (tutor, aluno_suspende) if
before(bel(tutor, tempo (T1)),
bel(tutor, tempo (T2)),
T1=T2.

Momentaneamente o aluno acredita que está tudo bem e sua intenção é esperar para ver o que acontece. Uma vez que o sistema está equilibrado, ele não necessita fazer qualquer tipo de intervenção. Este comportamento do aluno é representado no tutor por:

bel(tutor, aluno_sem_acao) if
bel (tutor,INT_THAT(Aluno,espera)).

Outra possibilidade que o tutor considera quando o aluno está sem ação é a impotência de realizar qualquer intervenção por ele não possui mais ferramentas disponíveis para jogar. Esta possibilidade é representada por:

bel(tutor, aluno_sem_acao) if
bel(tutor, aluno_e_um_personagem(Aluno, P1)),
P1= "prefeito",
before(bel(tutor, numero_ferramenta(lixeira, Nan))),
bel(tutor, numero_ferramenta(lixeira, Nat)),

```

Nan=Nat,
before(bel(tutor, numero_ferramenta(proibicao, Nan))),
bel(tutor, numero_ferramenta(proibicao, Nat)),
Nan=Nat,
before(bel(tutor, numero_ferramenta(multa, Nan))),
bel(tutor, numero_ferramenta(multa, Nat)),
Nan=Nat,
before(bel(tutor, numero_ferramenta(licenca, Nan))),
bel(tutor, numero_ferramenta(licenca, Nat)),
Nan=Nat.

/* o que significam as ferramentas zeradas */
bel(tutor, zerada(F, 0)) if
  bel(tutor, numero_ferramenta (F, No)),
  No= 0.

bel(tutor, numero_ferramenta_zerada_prefeito(No)) if
  bel(tutor, zerada(lixeira, L)),
  bel(tutor, zerada(proibicao, P)),
  bel(tutor, zerada(multa, Mu)),
  bel(tutor, zerada(licenca, Li)),
  No is L+ P + Mu + Li.

bel(tutor, numero_ferramenta_zerada_maenatureza(No)) if
  bel(tutor, zerada(accelera, Ac)),
  bel(tutor, zerada(intensifica, In)),
  bel(tutor, zerada(trocafunido, Tf)),
  bel(tutor, zerada(adicionaplanta, Ap)),
  No is Ac+ In + Tf + Ap.

```

No caso do MCOE, o conjunto de planos associados às intenções é unitário. O tutor vai escolher uma ação a realizar dentro do seu conjunto de comportamentos possíveis. Cada estratégia tem seu conjunto de táticas associadas e, por exemplo, a escolha do envio de uma mensagem específica representada a seguir:

```

bel(tutor, envia_mensagem(16)) if

```

bel(tutor, aluno_e_um_personagem(Aluno, P1)),
P1= "maenatureza",
bel(tutor, numero_ferramenta_zerada_maenatureza(4)),
bel(tutor, tempo_folga_1(Tx1)).

Em um ambiente mais complexo, como o modelado no protótipo, tivemos de considerar um conjunto bastante grande de variáveis que afetam o comportamento do tutor e dos estados mentais associados às ações dos alunos. Dada a natureza da modalidade de programa educacional escolhida (jogo), o tempo se torna fundamental. Esse fator foi tratado de forma diferenciada no contexto do jogo. Apresentamos alguns exemplos a seguir:

bel(tutor, tempo_restante(Tr)) if
bel(tutor, tempo_jogo(Tj)),
bel(tutor, tempo (T),
Tr is Tj-T.

bel(tutor, tempo_restante(Tr)) if
bel(tutor, tempo_partida(Tp),
bel(tutor, tempo(T),
Tr is Tp-T.

bel(tutor, tempo_folga_1(Tx1))if
bel(tutor, tempo_partida(Tp),
*Tx1 is Tp*0.8.*

bel(tutor, tempo_folga_2(Tx2))if
bel(tutor, tempo_partida(Tp),
*Tx2 is Tp*0.2.*

bel(tutor, tempo_folga_3(Tx3))if
bel(tutor, tempo_partida(Tp),
*Tx3 is Tp*0.5.*

des(tutor, alertar_tempo) if
bel(tutor, tempo_restante(Tr)),
Tr <= Tx2.

O “kernel” cognitivo produz uma descrição das interações do estudante com o tutor, a evolução dos estados mentais favorecendo ao professor verificar como os alunos perceberam e trabalharam o contexto. A escolha de ferramentas inadequadas ou de baixo poder de combate em uma situação crítica indica que o estudante ainda entende mal o problema e necessita de uma orientação adicional.

Importante salientar que o tutor tem um modelo pessoal de cada aluno, mas ainda não possui um modelo coletivo. O aspecto da modelagem coletiva está em aberto tanto em Ciência da Computação como na Psicologia. Os trabalhos de Paiva [PAI96;PAI97] e Hoppe [HOP95] apresentam alguns resultados obtidos até o momento e parecem ser promissores para se modelar grupos em ambientes computacionais. Investigações em andamento no CBLU, sob a supervisão do Dr. John Self, ainda estão em fase inicial e não possuem resultados finais publicados [TED98].

6. Experimentos realizados com o protótipo

A decisão de explorar as possibilidades de se projetar um STI utilizando uma arquitetura de SMA implicou na criação de um ambiente para testar idéias e verificar limitações e possibilidades da arquitetura proposta. A construção do protótipo desta tese foi elaborada ao longo de três anos. Na medida em que cada fase do projeto foi sendo concluída, tornou-se possível verificar alguns elementos da arquitetura confirmando hipóteses e delineando aspectos a serem considerados em trabalhos futuros.

Este capítulo apresenta o processo de construção do protótipo, os experimentos realizados a fim de validar aspectos de IA e IE e os resultados da validação pedagógica da proposta realizada com professores de escolas da nossa cidade.

6.1 Histórico e etapas realizadas

A decisão de construir um protótipo para possibilitar a observação da arquitetura pretendida para o ambiente e, em particular, para o agente tutor, necessitava de um ambiente de partida com um domínio de aplicação mais complexo do que o trabalho realizado anteriormente por Moussalle e Viccari [MOU96] no nosso grupo de pesquisa (GIA/UFRGS).

Moussalle e Viccari [MOU96] desenvolveram um STI para suporte ao ensino de Matemática (operação de divisão no conjunto dos números naturais, destinado a alunos de 5ª série do 1º grau). Os agentes cognitivos (tutor e aluno) foram modelados com a abordagem mentalística e utilizada a arquitetura SEM de [COR94]. O ambiente trata de um domínio muito específico, utiliza um conjunto reduzido de táticas de ensino e interface voltada a caracter, foi necessário pensar-se em um outro tipo de ambiente em função do tipo de investigação que desejava-se levar a termo.

Para tanto foram analisados os projetos de software educacional desenvolvidos pela autora desta tese e decidiu-se ampliar o projeto do jogo educacional denominado de Eco-Lógico [RAA96].¹⁹ O jogo Eco-Lógico foi redesenhado para uma arquitetura MAS e mantido apenas o seu princípio básico de funcionamento (vide anexo 2).

O projeto do novo ambiente passou por uma fase intermediária denominada de Multi-Ecológico [GIR97; GIR97a; GIR97b], em que o primeiro ambiente foi redesenhado para compor uma sociedade de agentes heterogêneos (reativos e cognitivos), mas ainda para apenas um usuário. Ou seja, criou-se um sistema utilizando agentes, mas não um sistema multiusuário. Nessa fase, foram projetadas a arquitetura dos agentes reativos e cognitivos e iniciou-se o estudo para determinação de qual formalismo seria utilizado para a modelagem dos agentes. Foi desenvolvido um protótipo para testar a arquitetura dos agentes reativos incluindo os elementos que deveriam compor a interface para suportar a arquitetura do tutor. Tais elementos foram incluídos, porque o uso de múltiplas estratégias exigiriam uma complexidade maior no projeto geral do sistema. Maiores detalhes vide anexo 3.

¹⁹ A autora desta tese participou como orientadora deste projeto.

Durante o período de estudos na Universidade de Leeds (UK), com a supervisão da orientadora. Dra. Rosa Maria Viccari, e do professor John Self decidiu-se ampliar a complexidade do sistema no que concerne aos agentes cognitivos colocando dois alunos trabalhando ao mesmo tempo e jogando a mesma partida. Essa nova abordagem fez com que o sistema apresentasse uma potencial de atividade de trabalho cooperativo por parte dos alunos e o ambiente ganhasse uma estrutura multiusuário. Além disso, as sociedades de agentes que fazem parte da arquitetura geral do sistema e do tutor foram definidas em detalhe. O protótipo foi renomeado para MCOE (*Multi Co-Operative Environment*) [GIR98b; GIR98c], fazendo com que o seu nome fosse mais representativo do que realmente o sistema representava.

6.2 Definição da interface

Os aspectos de ergonomia dos elementos da tela e a sobrecarga cognitiva inerente às interfaces de jogos foram estudadas em detalhe. Em geral as aplicações envolvendo projetos de STI, especialmente em trabalhos de tese, tratam da questão da interface do sistema como uma tarefa secundária. Optamos por incluir esse tópico em paralelo ao projeto da arquitetura devido à sua importância quando se trabalha com domínios mais complexos. A influência da interface é muito grande e quanto maior sua qualidade, maior as possibilidades de interação do usuário com o sistema e, também, a riqueza das respostas que o tutor pode enviar para o aluno. Um ambiente meramente textual restringe muito as possibilidades de táticas que um tutor pode utilizar.

O projeto da interface foi desenvolvido utilizando o método da avaliação heurística de Nielsen [NIE95]. Essa avaliação é realizada através da inspeção sistemática da interface considerando sua usabilidade. O objetivo é encontrar problemas de usabilidade no projeto, através de testes sucessivos com especialistas em interface ou da área de aplicação do sistema e potenciais usuários do programa. Os problemas identificados fazem parte do projeto interativo da interface que vai se adaptando conforme as observações do usuário e avaliadores.

A interface do MCOE sofreu várias alterações ao longo da implementação do protótipo devido ao uso da metodologia associada à avaliação heurística.

A primeira interface (figura 6.1–B) projetada foi baseada no modelo de tela criado para o Multi-Ecológico (figura 6.1-A). Os elementos do cenário estavam desenhados em 2D e utilizou-se um padrão de interface do Windows tradicional. Esta interface apresentou os seguintes problemas sob o ponto de vista dos avaliadores:

- cor da água diferente da real e muito escura;
- pouco realismo para mostrar a cadeia alimentar e as interações entre os elementos do cenário; e
- lago muito pequeno em relação a toda tela.

Na fase seguinte, fez-se alterações na cor da água e aumentou-se o tamanho da área de exibição do lago. O ecômetro ainda estava incorporado ao cenário como nas versões anteriores (figura 6.1 –C). Essa interface apresentou os seguintes problemas sob o ponto de vista dos avaliadores:

- pouco realismo para mostrar a cadeia alimentar e as interações entre os elementos do cenário; e

- o lago estava muito pequeno em relação a toda tela;

O cenário não motivava os usuários a sentir-se num ambiente de jogo, pois os elementos da interface estavam muito separados.

A figura 6.1-D apresenta a interface com a primeira versão do cenário modelado em 3D. A opção por uma mudança drástica na forma de modelagem dos elementos do cenário e como seria implementado o SMAR da arquitetura sucederam-se devido às observações dos avaliadores de que os elementos da interface estavam interferindo entre si. Essa interferência envolvia a falta de clareza entre o destaque do que era principal e o que estava acontecendo dificultando a percepção do usuário. Além disso, tínhamos as necessidades em nível de projeto para implementar as múltiplas estratégias do tutor.

Nessa versão, optou-se por dispor de um número menor de elementos no cenário e o ecômetro foi colocado na barra de comandos para ser acessado quando o usuário desejasse. O ecologista (tutor) permaneceu sendo apresentado na interface como nas versões anteriores. Essa simplificação da interface recebeu as seguintes críticas:

- o ecologista (tutor) não deveria aparecer na interface pois, ele não é um personagem do jogo (ele não joga só observa). A sua presença dá idéia de que ele pode agir no ambiente, diferente do que se desejava;
- a ausência do Ecômetro prejudica a avaliação do usuário a cerca do ambiente. Ele é um elemento central para o usuário guiar-se acerca do estado geral do sistema; e
- o cenário utilizado para o contorno da cidade e o fundo do lago não estavam adequados.

A figura 6-E apresenta a interface com o ecômetro e o cenário alterados. No entanto, ainda residia um problema em aberto: os poluidores. Os avaliadores sugeriram que os poluidores deveriam estar presentes na tela do jogo, em que mediante suas manifestações aconteceria algo diferente no ambiente. Dessa maneira a interface foi novamente modificada e a versão mais recente é apresentada na figura 6.1-F.

Os artigos que foram sendo escritos ao longo do trabalho de tese utilizaram diferentes telas como exemplo, pois eles refletiam o estado atual do projeto no momento em que foram escritos. Os artigos citados estão comentados no anexo 7.

6.2.1 Metodologia utilizada para melhoria da interface

O uso da avaliação heurística pressupõe a existência de um protótipo em fase de implementação que permita aos avaliadores julgarem a interface de acordo com os objetivos do programa. O teste foi dividido em duas etapas: testagem da parte reativa do sistema e testagem da interface com os agentes cognitivos (especialmente o tutor). Essa divisão foi realizada para favorecer a definição da forma com que as intervenções do tutor fossem apresentadas sem muita quebra no ritmo da interação dos alunos.

A primeira etapa teve a participação de oito professores que trabalham com o desenvolvimento de programas educacionais (implementação), onze professores de séries iniciais, quatro professores de Ciências e dois professores de Ecologia, todos pertencentes a escolas de nossa cidade. Foram utilizados como base para as entrevistas os seguintes critérios:

- distribuição dos elementos na tela;
- uso das cores;
- uso do som;
- textos utilizados no sistema de ajuda;
- botões associados às ferramentas; e
- sequenciação das telas que antecedem o início do jogo.

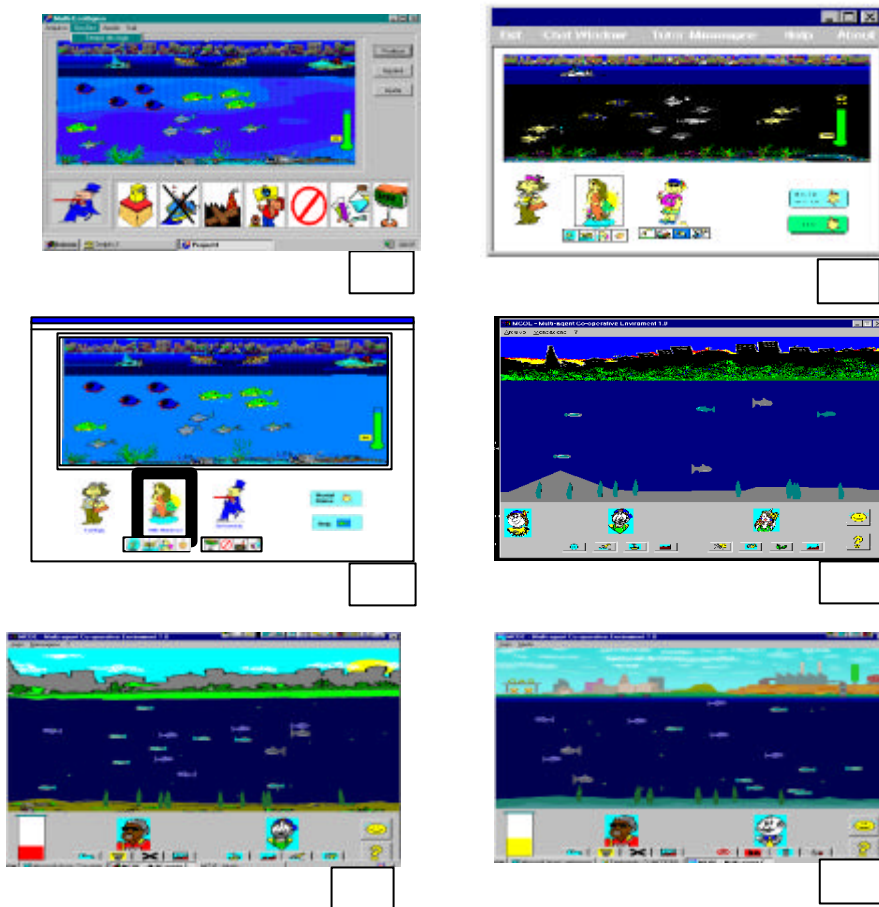


FIGURA 6.1 - Evolução da interface do sistema

Os especialistas foram entrevistados pela autora desta tese sem o uso de um instrumento específico e seguindo as orientações dos experimentos de Nielsen, em que os entrevistados relatam suas impressões a respeito da interface de forma intuitiva e baseados na usabilidade. Coube ao entrevistador registrar as observações e depois analisá-las de maneira a identificar as opiniões que eram comuns. Esse grupo de observações comuns acrescidas daquelas que foram devidamente justificadas pelos especialistas quando eles declaravam que aquilo era muito relevante para o bom

entendimento do ambiente, foram levadas em consideração para as mudanças feitas na interface do protótipo.

Dentre as alterações realizadas destacam-se as seguintes:

as alterações na cor de vários elementos do cenário para dar mais contraste aos seus elementos;

- o texto todo em segunda pessoa do singular;
- o uso de som associado aos botões das ferramentas dos alunos;
- a diferenciação racial nos personagens;
- a possibilidade de configuração do tempo do jogo; e
- a possibilidade de configuração número de ferramentas..

Algumas observações importantes não foram incorporadas ao protótipo devido a sua complexidade de execução no tempo disponível para conclusão da tese. Entre elas destacamos as seguintes:

- a possibilidade de se dispor de mais ferramentas para cada personagem oferecendo mais opções de escolha para o aluno;
- um sistema de graduação do ecômetro que refletisse o somatório da energia (atualmente esta informação é obtida pressionando-se o botão direito do mouse); e
- informações sobre cada elemento do cenário com o pressionar do ponteiro do mouse em cima deste elemento (atualmente apenas o nível de energia é fornecido).

A segunda etapa para determinação da forma como seriam apresentadas as intervenções do tutor teve a participação dos professores que fizeram mais observações e demonstraram maior interesse em contribuir com o projeto. Observou-se um perfil comum a estes professores: apresentavam maior experiência no uso de programas educacionais como atividade complementar e/ou introdutória com seus alunos, estavam sempre à procura de opções para exploração do conteúdo a ser trabalhado com os alunos, se mostraram voluntários para reuniões extraclasse, não manifestaram nenhuma consideração sobre uma possível forma de remuneração por parte da escola para participar dos experimentos e todos disseram que a oportunidade de participar da construção de um programa educacional estava sendo uma experiência muito rica.

O perfil identificado ao longo do trabalho de campo é aqui apresentado por julgarmos importante para os projetistas de programas educacionais que desejam fazer futuras validações pedagógicas de seus sistemas.

6.3 Potencial pedagógico do ambiente sob a perspectiva dos professores

Um ambiente projetado sob o ponto de vista construtivista, em que se deseja que o aluno construa seu conhecimento através da interação com os elementos que constituem este conhecimento, deve permitir um certo grau de liberdade ao professor e aos alunos no que concerne a sua exploração sob o ponto de vista pedagógico.

O ambiente deve permitir a exploração de objetivos do professor e dos objetivos específicos dos alunos. Além disso, deve favorecer a testagem de hipóteses construídas tanto pelo professor (em relação à aprendizagem dos alunos e forma de exploração do conteúdo), quanto pelos alunos (manipulação do conteúdo).

Todo programa pode ser considerado educacional desde que seja utilizada uma metodologia que o contextualize na ação docente e discente. Essa afirmação tem sido constantemente enfatizada em várias publicações conforme enfatizado no capítulo 2 desta tese. Um programa que seja projetado especificamente para fins educacionais sob a perspectiva construtivista deve poder se adaptar às várias metodologias e formas de trabalho de diferentes professores que utilizariam o ambiente. Portanto, é necessário que seja investigado o potencial pedagógico do ambiente. A testagem com os alunos dependerá da forma com que o professor pretende utilizar o ambiente.

O protótipo foi testado com 6 professores de séries iniciais de duas escolas (uma pública e outra privada), a fim de obter-se uma validação a respeito do potencial pedagógico do protótipo e investigar-se os ganhos advindos da utilização de uma arquitetura de SMA híbrida no projeto de um ambiente educacional, especialmente no uso de múltiplas estratégias por parte do tutor.

Esse experimento foi possível de ser efetuado, pois foram desenvolvidos dois protótipos com técnicas diferentes. Além disso, tornou-se possível a comparação e a identificação dos ganhos em utilizar-se as técnicas e procedimentos de IA e a verificação seu impacto no potencial pedagógico do ambiente.

6.3.1 Metodologia utilizada para validação pedagógica

Para obter resultados percentuais da comparação entre os dois ambientes e a testagem das hipóteses associadas ao projeto do ambiente utilizando SMA, foi elaborado um questionário (vide anexo 1), sob a supervisão de um psicólogo especialista em Psicologia Cognitiva experimental.

Os objetivos da aplicação do instrumento foram os seguintes:

- verificar como as duas interfaces utilizadas nos dois programas educacionais (Eco-Lógico e MCOE) influíram na percepção dos professores quanto às possibilidades de uso dos programas como ferramenta auxiliar ao ensino de poluição;
- verificar se a utilização de agentes aumentou a possibilidade de exploração do domínio da aplicação (poluição da água) com o uso de agentes, visto que os dois ambientes (Eco-Lógico e MCOE) possuem o mesmo objetivo (restaurar o equilíbrio do ecossistema através do uso das ferramentas dos personagens) e o mesmo conteúdo.

As hipóteses associadas aos objetivos do questionário são as seguintes:

- as possibilidades de exploração do domínio (sob o ponto de vista pedagógico num STI construído usando arquitetura de SMA são maiores do que a arquitetura convencional tradicionalmente utilizada em STI;
- o entendimento do objetivo do jogo aumenta utilizando a arquitetura de SMA;

- a interface usando agentes é mais dinâmica e interativa do que aquela implementada com a ferramenta Toolbook²⁰;

A amostra utilizada para esse experimento foi constituída de todos os professores de séries iniciais das escolas investigadas que trabalham com o conteúdo: poluição. Foram visitadas onze escolas na grande Porto Alegre (oito privadas e três públicas) que utilizavam o computador como uma ferramenta auxiliar no processo de ensino-aprendizagem e trabalhavam na perspectiva construtivista. O objetivo das visitas era identificar quais professores poderiam participar do experimento, uma vez que a atividade de avaliação dos ambientes demandava tempo adicional por parte dos docentes.

Das escolas visitadas, apenas três se manifestaram interessadas em participar do experimento. Muitas alegaram a proximidade do término do ano escolar (o experimento foi realizado entre outubro de 1998 a janeiro de 1999) como impedimento para realização dos testes devido a sobrecarga de trabalho dos docentes. Algumas, apesar de mostrarem-se interessadas, não confirmaram na prática o trabalho relatado pelas coordenadoras, fato esse que fez com que fossem excluídas do experimento.

Os dois programas e seus respectivos funcionamentos foram explicados aos professores pelo entrevistador (a própria autora desta tese). Além disso, um manual do usuário foi disponibilizado a todos professores. Os professores foram convidados a usar os dois programas tendo em mente sua possível escolha como ferramenta de auxílio, isto é, elemento a ser integrado nas suas atividades de ensino junto aos seus alunos. Após essa etapa os professores recebiam o questionário e o respondiam sem a presença do entrevistador. Os questionários foram recolhidos após o experimento.

6.4 O MCOE no ambiente de sala de aula

O grupo GIA/UFRGS realizou um experimento envolvendo a validação de um STI para o estudo de conteúdos de Física do segundo grau (lei de Ohm)) elaborado por Silveira e Vicari [SIL97], em que foram utilizadas três turmas de alunos utilizando a metodologia de grupos de controle. As turmas foram divididas em dois grupos, metade dos alunos utilizou o sistema de Silveira e outra metade não. Foram aplicados pré testes e pós-testes a fim de identificar se houve ou não um ganho significativo de aprendizagem por parte dos alunos que utilizaram o sistema. O teste foi uma prova elaborada pelos professores e atribuído grau numérico de zero a dez.

Apesar de termos interesse em aplicar esse tipo de validação com os alunos de nossa mostra, essa metodologia não utilizada. A avaliação dos alunos na faixa etária considerada no protótipo utilizam como forma de avaliação o parecer descritivo ao qual não é atribuído uma nota numérica ou equivalente. Os pareceres descrevem a evolução do aluno e comentam se eles atingiram os objetivos propostos nas atividades ou não. É feita uma avaliação mais de Mórada com duração de um a dois bimestres letivos e, em alguns casos apenas é descrito o grau de participação, o envolvimento e o interesse do aluno na atividade, sendo medido mais adiante o quanto ele reteve do conhecimento trabalhado.

²⁰ Toolbook é marca registrada da empresa Assimatrix.

As escolas que trabalham com metodologias centradas no aluno, em que o objetivo maior é a construção do conhecimento, não se preocupam em valorar e quantificar esse conhecimento a fim obter um dado mais preciso (por exemplo, tal atividade fez com que o aluno tivesse 20% a mais de resposta positiva ou equivalente). Os alunos são avaliados num processo contínuo. Os professores avaliam as atividades realizadas com os alunos para saber o que eles acharam de positivo ou negativo no tipo de trabalho desenvolvido. Sob essa perspectiva é que nossa avaliação foi desenvolvida.

Dada a impossibilidade de realizarmos um experimento de longo prazo devido às restrições de tempo inerentes ao projeto de tese, foi realizado apenas um experimento empírico com as crianças a fim de observar sua reação ao ambiente no que diz respeito ao entendimento do jogo, sua percepção sobre o conteúdo trabalhado e a forma com que eles interagiram com o sistema.

Um grupo de 16 alunos de 3ª série do 1º grau utilizou o ambiente no laboratório da escola. O jogo foi explicado em sala de aula e o seu manual utilizado como base para a professora informar os alunos sobre o conteúdo do jogo. A turma foi dividida em dois grupos e a atividade foi complementada e realizada em três etapas. Enquanto alguns se preparavam para o ensaio das festividades de final de ano letivo, alguns voluntários foram ao laboratório utilizar o ambiente. Essa distribuição ocorreu devido ao número de equipamentos disponíveis no laboratório da escola (4 microcomputadores). Apesar de não haver rigor metodológico nesse experimento, ele foi utilizado para termos um indicador de interesse dos alunos pelo ambiente e para a professora verificar se o sistema poderia ser utilizado ou não no ano escolar de 99 como uma de suas possibilidades para o estudo da poluição.

As crianças demonstraram muito entusiasmo pelo ambiente, identificaram alguns elementos que julgaram complexos de entender num primeiro momento e perguntaram como o jogo foi feito. Os alunos manifestaram interesse em ter uma cópia do jogo para levar para casa e perguntaram se teriam oportunidade de jogar novamente no reinício do ano escolar. Devido a esses resultados, a professora se prontificou a fazer parte do experimento que desejamos elaborar como parte dos trabalhos futuros a serem realizados a partir desta tese.

Esse experimento será detalhado no próximo capítulo, capítulo 7, quando da apresentação dos trabalhos futuros a serem realizados a partir dos resultados desta tese.

6.5 Os resultados do experimento

Os professores entrevistados quando selecionam *software* educacionais para utilizar com seus alunos, levam em consideração as seguintes preferências:

- selecionam *software* como elemento integrante de atividade complementar aos trabalhos realizados na classe, em detrimento de outros recursos (vídeos, televisão, etc.) (60%);
- usam *software* mais de 5 vezes ao ano nas atividades complementares (66%);

Com relação aos programas escolhidos, os professores manifestaram opinião de que os programas devem apresentar as seguintes características:

- escritos em língua portuguesa (83%);

- modalidade jogos *are* educacionais (83%);
- uso de som e animações (50%);
- permitam aos alunos realizarem atividades interativas (83%);
- armazenem os resultados dos alunos para posterior discussão (66%);
- o ambiente se apresente de forma diferenciada cada vez que o aluno vai utilizá-lo (83%);
- possuir um sistema de ajuda para entender o funcionamento do sistema (83%);
- possuir um sistema de ajuda que apresente explicações sobre o domínio da aplicação (conteúdo) (66%).

A comparação entre os dois ambientes utilizados (Eco-Lógico e MCOE) nos forneceram resultados importantes para verificar nossas hipóteses:

- representação mais adequada do ecossistema de um lago: MCOE (33%), Eco-Lógico(33%), ambos (16%);
- melhor representação da vida num lago: MCOE(100%);
- conteúdos mais significativos para serem explorados pelos alunos: MCOE(100%);
- dinâmica dos elementos da interface: MCOE (33%), Eco-Lógico(33%), ambos (16%);
- mais fácil de ser entendido na primeira vez que o aluno joga: MCOE (83%), Eco-Lógico(17%);
- mais adequado para trabalhar com seus alunos: ambos os sistemas (100%).

Um dos resultados mais significativos para nosso trabalho foi os 100% alcançados nos conteúdos significativos. O domínio é o mesmo, as regras do jogo são iguais. Entretanto, os professores tiveram a percepção que o MCOE era mais significativo que o outro jogo. O Eco-Lógico usa apenas uma estratégia criada pelo especialista para trabalhar com os alunos. No MCOE temos múltiplas estratégias intrínsecas na concepção do projeto. Usando estas múltiplas estratégias para modelar o comportamento do tutor tivemos de sofisticar a interface de maneira a suportar os aspectos pedagógicos inerentes às múltiplas estratégias.

O uso da técnicas de agentes nos auxiliou a alcançar o grau de sofisticação necessária para o projeto do sistema. No acaso de STI os agentes são considerados *agentes pedagógicos* e possuem propriedades fundamentais: autonomia, habilidade social, pró-atividade e persistência. Alguns deles podem ser reativos, contínuos, representar personagens e ser capazes de aprender. Os agentes do MCOE possuem todas estas propriedades, exceto a capacidade de aprender (no presente momento).

Este conjunto de propriedades traz características importantes para os agentes pedagógicos sob o ponto de vista educacional. Nos agentes reativos, estas propriedades melhoraram a qualidade da representação do domínio, aumentam o *feedback* sobre as ações dos alunos e suas estratégias para resolverem os problemas.

O MCOE é um ambiente de resolução de problemas onde os alunos possuem múltiplos desafios durante a mesma partida. A possibilidade do ambiente apresentar-se com diferentes aspectos (interface) cria uma sensibilidade nos alunos de que o sistema

se modifica à cada vez que eles jogam. O jogo nunca se apresenta da mesma forma. O número de peixes no lago, plantas e outros elementos do cenário variam, bem como os desafios que o aluno deve enfrentar. Isto ocorre mesmo que o personagem escolhido seja o igual ao da partida anterior.

Considerando-se os agentes cognitivos, a autonomia garante as condições dos alunos atingirem seus objetivos pessoais. Num ambiente educacional o agente tem de analisar suas próprias ações e seus reflexos no ambiente (propriedade pró-ativa), antes de fazer uma determinada ação. A habilidade social está implícita nas ações executadas no teclado quando o aluno pressiona determinado botão associado a uma ferramenta de combate da poluição e, também, codificada no conjunto de crenças ligadas às ações do aluno.

7. Conclusões, limitações e trabalhos futuros

A construção de STI utilizando arquitetura de sistemas multiagentes permite um ganho de qualidade tanto no aspecto de representação do domínio, aumentando as possibilidades pedagógicas do ambiente, quanto nas possibilidades de se construir modelos de aluno e tutor mais robustos.

A utilização da abordagem mentalística possibilitou o resgate da questão em aberto na área de STI:

Como o tutor pode selecionar, entre várias estratégias de ensino, a mais adequada para cada perfil de aluno?

e a construção de uma arquitetura em que o projetista apenas tem de descrever o conjunto de estados mentais que devem ser associados aos agentes cognitivos. Apesar das limitações já constadas na atual fase da arquitetura (vide item 7.1 deste capítulo) os resultados obtidos nos permitiram avançar de forma significativa os resultados até então alcançados no nosso grupo de pesquisa.

A flexibilidade da arquitetura favorece sua aplicação em outros ambientes de ensino inteligente, cujo domínio requiera uma modelagem mais complexa. Citamos como exemplo direto de aplicação o trabalho de doutorado de Casas [CAS99], utilizando como base o exame de qualificação e a proposta de tese de [GIR98], em que é apresentada uma arquitetura multiagente para construção de um ambiente educacional inteligente baseado em RV. Casas utilizou a proposta geral da arquitetura de STI para compor um ambiente interativo em que o tutor toma decisões baseadas no modelo que tem do ambiente, de si próprio e do aluno. Entretanto, Casas não utiliza o “kernel” cognitivo como proposto nessa tese. Ele utiliza um outro modelo de aluno, compostos por aspectos mais tradicionais e sem o uso de estados mentais. O trabalho disponibilizado na Internet, permitiram a Casas delinear seu modelo multiagente utilizando o modelo geral proposto por Giraffa e Viccari.

Além do aspecto geral da arquitetura multiagente, o “kernel” cognitivo pode ser acoplado a outros ambientes multiagente fazendo com que os aspectos intrínsecos a um STI sejam incorporados ao sistema original. Para tanto o projetista deve elaborar o conjunto de estados mentais associados às ações dos alunos e o conjunto de estados mentais associados ao comportamento do tutor. Faz-se necessária uma interface que comunique o ambiente ao kernel cognitivo, utilizando um protocolo de comunicação entre a linguagem do ambiente de origem e o kernel implementado em Prolog.

A inclusão do modelo de estudante construiu usando arquitetura de BDI fornece mais do que informação sensorial para o tutor. Descreve o estado atual do estudante e ao mesmo tempo não o rotula no que concerne a estereótipos prévios. O estudante está sendo considerando a partir do mapeamento que o tutor faz dele na sessão de trabalho. Pode-se dizer que o aluno é comprado a ele mesmo, sendo usado como medida o seu próprio seu desempenho. O tutor projetado com uma arquitetura de estados mentais também se torna mais robusto e ao mesmo tempo flexível. Pois, adaptar o tutor para uma nova situação e contexto, significa excluir ou incluir novos estados mentais a cerca do ambiente e dos alunos.

Pelo lado de agentes cognitivos, a autonomia garante as condições para alcançar as suas metas pessoais. A propriedade de proatividade que os agentes possuem trazem de

forma implícita algo que é considerado muito importante num ambiente educacional: o pensamento reflexivo. O aluno pensar sobre a sua ação e as conseqüências que advirão desta ação. A habilidade social dos agentes está implícita nas ações de teclado quando o aluno pressiona uma ferramenta selecionada do seu personagem.

O uso de múltiplas estratégias para modelar o comportamento do tutor implicou na sofisticação da interface. A interface tem de permitir um ambiente interativo, rico em possibilidades para o aluno e com alternativas para se poder expressar as múltiplas táticas associadas às estratégias do tutor. O uso das técnicas de agente possibilitou-nos alcançar esse grau de sofisticação necessário para projeto de um STI com essas características. Um dos resultados mais importantes obtidos com o experimento realizado junto aos professores, foi os 100% alcançados em conteúdos significativos. O domínio é o mesmo, e as regras de funcionamento são iguais. No entanto, todos os professores tiveram a percepção que MCOE tem conteúdos mais significativos que o outro jogo. Um explicação para esse resultado reside no fato de que o Eco-Lógico só usa a estratégia criada pelos especialistas que auxiliaram a modelar o sistema, logo a interface apresenta uma maneira do aluno perceber o que está acontecendo no sistema e interagir.

Os agentes reativos utilizados aumentaram a qualidade da representação do domínio e ampliaram as possibilidades do aluno observar o resultado de suas ações no ambiente. O resultado final é um ambiente rico e que permite um *feedback*²¹ instantâneo do que está acontecendo no sistema, fazendo com que o aluno possa ter mais elementos para poder montar sua estratégia de ação. É importante lembrar que o jogo (MCOE) é um ambiente de resolução de problemas em que os estudantes têm desafios múltiplos durante a mesma sessão. A possibilidade de exibir o ambiente do jogo com aspecto diferente a cada vez que o aluno inicia uma nova partida, cria a sensação de um novo desafio. Fica muito difícil o aluno conseguir saber de forma antecipada o que vai acontecer. O número de peixes que irá compor o cenário, qual a situação do sistema após a execução de um determinado conjunto de ações, permitem criar um ambiente que não têm uma configuração fixa e exibe uma situação diferenciada a cada vez que é inicializado.

Este trabalho, além das contribuições científicas, favoreceu a combinação e a exploração de várias tecnologias que vinham sendo utilizadas de forma separada em muitos projetos existentes no II/CPGCC. É um trabalho que mostrou sua adequação ao solucionar os problemas decorrentes da arquitetura proposta ao garantir a interação dos aspectos pedagógicos, de qualidade da interface e de performance do sistema (SMAR).

A arquitetura multiagente apresentada nesta tese também será utilizada como base para o desenvolvimento de uma arquitetura geral para STI na área de saúde. Maiores detalhes vide item 7.2 deste capítulo).

Os resultados acima apresentados e as diversas publicações obtidas com essa tese (vide anexo 7), constituem-se num indicador importante das possibilidades dessa arquitetura.

²¹ *Feedback*: retroalimentação. O aluno recebe um conjunto de informações significativas que o permitem refletir a cerca da conseqüência de suas ações.

7.1 Limitações

Os resultados obtidos com esse trabalho nos permitem identificar elementos importantes para continuar nossa pesquisa com agentes pedagógicos modelados na abordagem mentalística. Podemos construir interfaces mais poderosas e realistas para simulações de ambientes educacionais na modalidade jogos com a arquitetura de um STI e ampliar as possibilidades de exploração no que concerne aos aspectos pedagógicos. Entretanto, algumas limitações se apresentam e necessitam de um estudo mais amplo.

As limitações existentes hoje no modelo proposto são as seguintes:

- Não estamos trabalhando o aspecto do conflito que pode existir quando os dois agentes alunos trabalham de forma cooperativa;
- O número de estados mentais que está sendo considerado no momento (apenas 4 estados).

A cooperação entre os alunos ainda é trabalhada de forma à parte (off-line) no sistema. O tutor percebe o que acontece entre os alunos através do conjunto de estados mentais associados às suas ações e não monitora ou participa do processo de negociação para solução do conflito que pode emergir entre os alunos. Esse aspecto não foi considerado nesta etapa da pesquisa devido à sua complexidade que demanda um estudo específico e baseado em modelos coletivos de grupo e, também a solução de conflitos através do processo de negociação entre os agentes. A modelagem coletiva de grupos é uma área de estudo bastante complexa e tem sido objeto de estudo de vários pesquisadores de IA, especialmente de STI. Os trabalhos de Laurillar [LAU92], Roschelle [ROS92], Katz e Lesgol [KAT96], Fuchs et al. [FUC94], Shapiro et al. [SHA95], Grasser e Person [GRA96], Tedesco e Self [TED98] são exemplos do estado da arte nessa área: modelos coletivos para agentes.

Através da parceria existente entre a orientadora desse projeto, Dra. Rosa Maria Viccari, e o Prof. John Self do CBLU da Universidade de Leeds, pretende-se utilizar os resultados obtidos com as pesquisas em andamento, verificar a viabilidade de tratar tais aspectos e propor uma melhoria na arquitetura do tutor. Nesse sentido, o objetivo é contemplar os tópicos discutidos no ambiente em que o tutor oferece auxílio ao aluno. Além de considerar as características individuais do aluno o tutor levaria em consideração o estado da negociação existente nos alunos e o perfil de cada agente durante o processo de negociação. O aluno que apresentar um perfil mais dominador vai necessitar de um tipo diferenciado de auxílio daquele que demonstrar uma atitude mais passiva no processo de negociação. Esses aspectos, se considerados, vão enriquecer o modelo do aluno e ampliar a qualidade do auxílio fornecido pelo tutor e, por consequência, implicarão numa ampliação do conjunto de estratégias e táticas utilizadas pelo tutor.

Corrêa, Coelho e Viccari [COR98], estão desenvolvendo um trabalho no sentido de identificar quais os estados mentais (além dos 4 utilizados como consenso) seriam necessários para se modelar agentes pedagógicos. Num estudo preliminar foram identificados os seguintes estados mentais: decisões, capacidades, compromissos e urgência. Esses são possíveis estados a serem incorporados ao modelo de agentes

pedagógicos. Todos baseados em conceitos análogos ou similares aos humanos sustentados pelas idéias de [SHO93].

Com a utilização de mais estados mentais para enriquecer o modelo do aluno (aspectos quantitativos e principalmente qualitativos), poderemos ampliar a eficácia do tutor para oferecer um auxílio cada vez mais personalizado ao tipo de aluno que está interagindo com o ambiente (STI).

Além de tais limitações, consideramos a não validação do aspecto relativo à aprendizagem do aluno. O prazo utilizado por uma tese requer o estabelecimento de prioridades e metas a serem atingidas. Chega-se ao final de uma etapa com muitas idéias em aberto e que não são exequíveis no prazo regulamentar, uma vez que muitas delas são decorrentes dos resultados obtidos com a pesquisa da própria tese. Esse é o caso desse tipo de validação pedagógica do ambiente proposto. Será realizado um experimento para validar o sistema junto aos alunos. Maiores detalhes serão apresentados no item a seguir.

7.2 Trabalhos futuros

7.2.1 Utilização da arquitetura

Os resultados obtidos nesta pesquisa e no trabalho de Móra et al. permitiram que fossem elaborados dois projetos de pesquisa envolvendo a arquitetura:

- O primeiro projeto está inserido no escopo do edital em parceria entre os Programas de Tecnologia da Informação (PTI) e de Educação (PEDU) para formação de grupos de Pesquisa em Informática na Educação. O edital poderá ser encontrado no site <http://www.cnpq.br/dpe/protem-cc/edital-pti.htm>. O projeto pretende desenvolver um framework (arcabouço) para o desenvolvimento de STI baseado na arquitetura apresentada nesta tese e utilizando o X-BDI.
- O segundo projeto atende ao edital do programa PROTEM-CC, na área de IE, envolvendo o Departamento de Computação da Universidade Estadual de Londrina e a Faculdade de Informática da PUCRS. O projeto submetido denomina-se ATHIFIP - ambiente tutor hipermídia inteligente aplicado ao ensino da área de saúde. A idéia básica do projeto é o desenvolvimento de uma arquitetura multiagente de STI Hipermídia aplicado ao Ensino da área de Saúde. A arquitetura suporta um ambiente multiusuário (2 alunos por máquina, como é de praxe nas instituições educacionais) ou um ambiente distribuído (com alunos trabalhando em espaço físico diferente). A validação dessa arquitetura será feita através de experimentos a serem realizados com um protótipo na área de Fisioterapia. Essa aplicação se justifica devido à existência de um tutorial hipermídia já testado e desenvolvido pelo grupo de Londrina. A área de fisioterapia será utilizada para a construção da base de domínio do STI.

7.2.2 Experimento envolvendo a aprendizagem dos alunos

O experimento será executado ao longo do ano escolar de 1999. A professora irá trabalhar os conteúdos utilizados para modelagem da base de domínio do MCOE e, depois, fará uma ampla e detalhada explicação do funcionamento do sistema. Os alunos irão utilizar o ambiente com orientação da professora a fim de poderem testar hipóteses oferecidas pela professora e depois num segundo momento, para exploração livre conforme o interesse dos alunos.

A professora vai elaborar um questionário referente aos conteúdos trabalhados para ser aplicado aos alunos e será solicitado a eles fazer um desenho que reflita o que aprenderam sobre a poluição.

Ao mesmo tempo, em paralelo, outra turma da escola estará trabalhando o mesmo conteúdo com outra professora. Ambas as professoras possuem o mesmo conteúdo a ser trabalhado, só que uma das turmas vai interagir com o sistema e a outra não. Os mesmos testes serão aplicados aos dois grupos de alunos.

No final do ano escolar, as professoras irão aplicar os mesmos testes nas duas turmas e os dados serão comparados. Desejamos verificar se a turma que utilizou o sistema apresentou performance diferente da outra turma que não o utilizou. Ou seja, o quanto o sistema auxiliou ou não os alunos a reterem informações significativas dos conteúdos estudados (poluição da água). O desenho vai mostrar o quanto do ambiente ficou retido na memória das crianças. Espera-se que ao serem solicitadas a desenhar sobre a poluição, muitas das crianças que interagiram com o sistema façam uma associação com o ambiente gráfico do jogo mostrando um indicador da participação do sistema como auxiliar na aquisição de conhecimento por parte dos alunos.

Esse experimento e o tratamento dos dados contará com a supervisão do mesmo psicólogo que auxiliou a elaboração do instrumento utilizado na validação pedagógica do sistema, uma aluna de mestrado em psicologia e outra do mestrado em Educação. A supervisora pedagógica da escola irá supervisionar o andamento do experimento. Os alunos serão filmados enquanto utilizam o sistema e suas observações sobre a usabilidade do jogo serão registradas, a fim de obtermos indicadores para melhoria da interação dos alunos e do manual do sistema.

Os resultados serão publicados na forma de relatório técnico e artigo científico.

7.2.3 Trabalhos a serem realizados com o protótipo

O sistema proposto para teste das idéias dessa tese (MCOE) recebeu uma ótima aceitação por parte dos professores entrevistados que manifestaram seu interesse em utilizá-lo como recurso complementar de suas atividades com os alunos. Devido ao potencial do ambiente e sua aceitação, decidiu-se continuar implementando as sugestões colhidas ao longo dos experimentos da seguinte forma:

- colocar mais ferramentas para cada personagem e permitir que o usuário escolha as ferramentas que deseja utilizar;
- colocar mais personagens disponíveis para o usuário;

- ampliar o dicionário do sistema de ajuda e permitir a inclusão de novos termos por parte do aluno e do professor ;
- disponibilizar no sistema de ajuda links interessantes para sites que falem sobre a poluição de águas; e
- salvar as interações de todos os alunos de forma mais acessível e gerar um relatório para que professores e alunos possam analisar resultados e trabalhar melhor suas hipóteses e respectivas testagens.

Além dessas melhorias que se deseja colocar no sistema, está sendo realizado um trabalho por Callegari e Oliveira, [CAL99], no Curso de Mestrado em Informática da PUCRS, denominado de “*Uma proposta de extensão ao ambiente MCOE utilizando aprendizagem por reforço*”, cujos objetivos são:

- fazer com que os agentes reativos do sistema MCOE alterem seus padrões de comportamento em função das mudanças no ambiente;
- estudar a necessidade de se propor mudanças na arquitetura do ambiente para acomodar agentes reativos que aprendem via aprendizagem por reforço;
- propor mudanças na arquitetura interna dos agentes reativos usando aprendizagem por reforço; e
- implementar as alterações propostas no protótipo do MCOE para visualizar os efeitos das alterações.

Espera-se que essas alterações venham a tornar o ambiente mais dinâmico gerando um conjunto maior de efeitos próximos aos que acontecem na natureza (adaptação). Além disso, as mesmas poderão influenciar a forma como os alunos resolvem os problemas, uma vez que as regras do sistema terão de ser adaptadas e o jogo ganhará um nível de complexidade maior através da aprendizagem. Isso poderá ampliar a faixa etária de utilização e a forma com que o sistema pode ser explorado do ponto de vista pedagógico. Assim, o trabalho visa contribuir para a área de IA aplicada à Educação, ao estudar a aplicação de Aprendizagem por Reforço em Sistemas Multiagentes em um ambiente educacional.

Anexo 1 – Evolução do MCOE

1.1 O ambiente do Eco-Lógico

O Eco-Lógico é um jogo educacional desenvolvido utilizando Toolbook Multimedia. A aplicação de domínio é Ecologia (poluição de água). Neste ambiente ecológico o estudante escolhe um personagem para jogar usando seis opções: Mãe Natureza, Prefeito, Cidadão, Pescador, Ecologista, e Turista. A figura 1 apresenta a tela de escolha dos personagens.

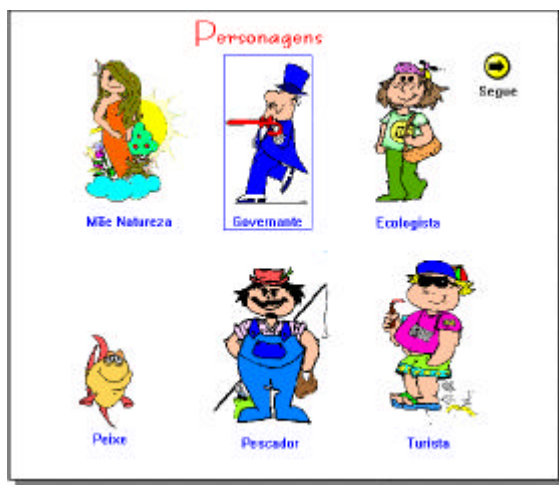


FIGURA 1: A tela de escolha dos personagens do Eco-Lógico

Depois disso, o estudante define a configuração de jogo escolhendo quantos elementos estranhos ao ecossistema que causarão poluição serão colocados no lago. A figura 2 apresenta os poluidores disponíveis para seleção.

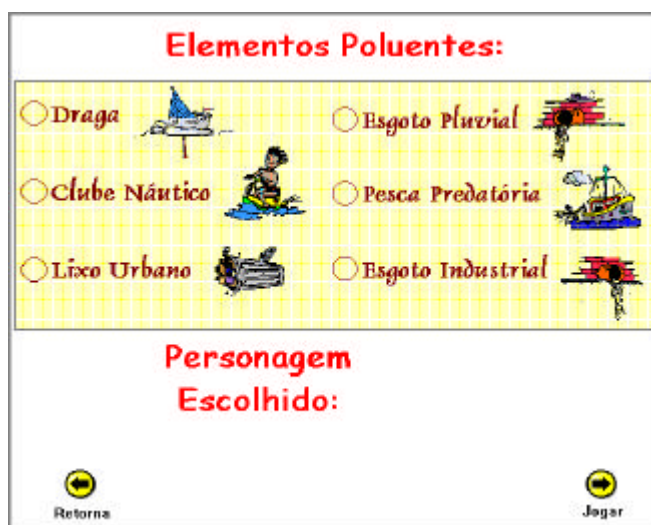


FIGURA 2: A tela de escolha dos elementos interventores.

Os conteúdos apresentados foram revisados pelos especialistas na área de Ecologia Ambiental da Universidade Federal do Rio Grande do Sul, levando em consideração o fato de que o Eco-Lógico é uma ferramenta de auxílio ao desenvolvimento de atividades em sala de aula, na qual o papel do professor é o de orientador nas atividades do aluno (quando questionado) e, principalmente, trabalhar com os resultados obtidos por estes propiciando um reforço dos conteúdos. O Eco-Lógico pode ser utilizado em atividade extra-classe como um ambiente meramente lúdico ou como parte de uma tarefa complementar proposta ao aluno.

Depois de 30 segundos o poluidor escolhido aparece e começa a mostrar a sua ação (retirar energia dos elementos do cenário).

Um medidor de energia denominado de ecômetro ajuda o usuário a observar as condições do lago. Uma situação bem equilibrada é mostrada quando a medida está cheia e verde. Como o equilíbrio está perdido, a cor muda para amarelo gradualmente e, finalmente para vermelho quando uma situação perigosa é alcançada. O número de peixes que nadam no lago também indica a situação do ecossistema.

O desafio do estudante é manter o equilíbrio e lutar contra a ação destes elementos poluentes através de quatro ferramentas. Cada personagem tem suas próprias ferramentas com poder diferenciado para lutar contra a poluição. O período de tempo do jogo é cinco minutos.

O jogo é modelado para apenas um aluno jogar. Há efeitos de sons durante as animações, e podem ser obtidas informações sobre os elementos do cenário pressionando o botão direito do mouse. Ao final do jogo o estudante recebe uma mensagem explicando se alcançou ou não o equilíbrio do sistema.

Maiores informações visite a homepage do projeto:

<http://www.inf.pucrs.br/~raabe/eco-logico/>

1.2 O ambiente do Multi- Ecológico

A modelagem do Multi-Ecológico [GIR97; GIR97a; GIR97b], foi uma etapa intermediária para testagem da arquitetura de SMAR e lançou as bases para o MCOE. No Multi-Ecológico, a interface era fortemente baseada no Eco-Lógico pois ainda se considerava apenas um aluno jogando. Foram mantidas as mesmas regras de funcionamento do Eco-Lógico e restringido o número de personagens para quatro opções e os alunos ainda podiam escolher os poluentes através de um processo de seleção. Houve mudança na interface em decorrência do ambiente de programação utilizado (Delphi). A figura 3 apresenta um exemplo da interface utilizada nesta fase.

O sistema não foi finalizado e apenas ficou numa versão de teste, uma vez que a proposta foi modificada substancialmente com a inclusão de mais um aluno para jogar e a definição de múltiplas estratégias associadas utilizadas na arquitetura do tutor implicaram num projeto mais complexo da interface. Nesta fase do projeto uma dissertação de mestrado foi ligada ao projeto a fim de auxiliar as definições da implementação do SMAR e a versão inicial do que seriam os arquivos de teste utilizados pelo tutor.

Após os resultados parciais obtidos e as novas considerações agregadas ao projeto, optou-se por trocar o ambiente de desenvolvimento em função das futuras necessidades

de ligação entre o kernel cognitivo e o SMAR. Estes resultados parciais foram redimensionadas e rescritos em função das definições posteriores implementadas pelo refinamento da arquitetura e da proposta geral do sistema.



FIGURA 3: Interface do Multi-Ecológico

Anexo 2 – Manual do Usuário do MCOE

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



MANUAL DO USUÁRIO

Elaborado por

LUCIA MARIA MARTINS GIRAFFA

Orientação

DRA. ROSA MARIA VICCARI

PORTO ALEGRE

1999

Pretendemos, com nosso trabalho, auxiliar a conscientização da importância da Educação Ambiental, numa fase onde a criança forma o seu caráter, personalidade, e seus valores sociais. Nosso objetivo é que o software possa se tornar mais uma alternativa para a conscientização ecológica e seja mais um recurso disponível para os professores da área.

Agradecimentos

Alexandre Arenzon, e Sandra Maria Hartz, pelos conceitos e ensinamentos sobre ecologia, e pela ajuda fundamental na definição do ecossistema do lago na versão do Eco-Lógico que foi utilizada como base para o MCOE.

Edson Lindner, pela idéia do lago, e pelo interesse e prontidão sempre que foi preciso.

Às coordenadoras e professoras das escolas pesquisadas pela disponibilidade e riqueza das suas contribuições.

À André Raabe pela disponibilidade, comprometimento e colaboração ao longo de todos estes anos de trabalho.

À Leandro Bernsmüller pela dedicação, competência e colaboração ao longo da implementação do MCOE.

Às crianças que auxiliaram a testar o MCOE.

Sumário

1. EQUIPE DE PESQUISA E DESENVOLVIMENTO	121
2. OBJETIVO DO JOGO E PÚBLICO ALVO	121
3. A CADEIA ALIMENTAR DE UM LAGO	121
4. OS PERSONAGENS DO JOGO	122
5. OS POLUENTES	124
6. O ESQUEMA DA ENERGIA NO SISTEMA	126
6.1 <i>Esquema da reprodução</i>	127
6.2 <i>Retirada e reposição de energia</i>	127
7. COMO CONFIGURAR O MCOE PARA JOGAR	129
8. A ESCOLHA DOS PERSONAGENS.....	133
9. O AMBIENTE DO JOGO	134
9.1 <i>Cálculo do Nível de Energia do Ecômetro:</i>	135
10. O SISTEMA DE AJUDA.....	113
11. REQUISITOS DE HARDWARE E SOFTWARE PARA UTILIZAR O MCOE	137
11.1 <i>Requisitos de hardware</i>	137
11.2 <i>Requisitos de software</i>	137
12. O TUTOR DO MCOE (APENAS NA VERSÃO 2.0)	137
13. INFORMAÇÕES SOBRE O GRUPO DE PESQUISA EM IA-ED	138
14. BIBLIOGRAFIA DO MANUAL DO USUÁRIO.....	138

1. Equipe de pesquisa e desenvolvimento

Lucia Maria Martins Giraffa (FACIN/PUCRS – CPGCC/UFRGS)

Concepção e projeto do MCOE, desenhos 2D e arquitetura do tutor e Orientadora do projeto Eco-Lógico.

Leandro Bernsmüller II/UFRGS

Programação, modelos 3D e sonoplastia do MCOE

André Raabe (FACIN/PUCRS)

Programação do Eco-Lógico, auxiliar na coleta de dados nas escolas e organizador do texto sobre o referencial teórico deste manual.

Orientação projeto MCOE: Prof.a. Dra. Rosa Maria Viccari (CPGCC/UFRGS)

2. Objetivo do jogo e público alvo

No MCOE, o aluno tem um ambiente onde aparecem inúmeros problemas ao longo da sua sessão de trabalho e deverá solucioná-los utilizando seu conhecimento prévio e combinar ferramentas na construção de uma estratégia de ação.

O jogo é composto por um lago onde existe um ecossistema formado por peixes, plantas, água e microrganismos que possuem um sistema de reprodução em equilíbrio até a intervenção de poluentes que provocam alterações no seu estado normal. Estes poluentes aparecem de forma aleatória ao longo do jogo e são combatidos através de ferramentas do personagem escolhido por cada aluno.

O aluno tem a liberdade de escolher a ferramentas que quiser e vai interagir com um colega para construir uma estratégia comum para resolver o problema da poluição do lago.

O jogo foi concebido para alunos que estejam freqüentando a 3^a e 4^a série do primeiro grau (ensino fundamental).

Podendo, entretanto, ser utilizado por crianças d outras faixas etárias através da configuração de seus parâmetros e da utilização de uma metodologia que permita a contextualização do ambiente nas atividades realizadas pelos alunos.

3. A Cadeia Alimentar de um lago

Um lago pode ser considerado como sendo um ecossistema de pequeno porte, pois os seres vivos ali inseridos convivem teoricamente em equilíbrio entre si e com o meio.

A transferência da energia alimentar, desde a fonte nos autótrofos (plantas-produtores), até os consumidores, ocorre através de uma série de organismos que consomem e são consumidos, e esta transferência é chamada de Cadeia Alimentar ou Cadeia Trófica.

Os seres vivos presentes em um lago formam uma cadeia alimentar composta de todos os níveis de consumidores (primário, secundário, terciário), produtores, decompositores, enfim todos os elementos necessários para que seja possível a situação de equilíbrio.

As cadeias alimentares aquáticas, como lagos, tem um comportamento específico. As algas são a base da cadeia alimentar, elas fixam a energia solar através da fotossíntese. O zooplâncton (microorganismos animais), que se alimentam das algas são os consumidores primários. Os peixes que se alimentam diretamente das algas, também são consumidores primários, por sua vez, os peixes que se alimentam de zooplâncton são consumidores secundários. Por fim, os peixes predadores, que se alimentam de outros peixes, são os consumidores terciários.

A quantidade de peixes em cada nível trófico obedece a relação de aproveitamento da energia potencial ao longo da cadeia alimentar.

É claro que um lago não existe isoladamente na natureza, como um ecótono (pequeno ecossistema) ele interage com outros ecossistemas e ecótonos formando um complexo maior.

Em um lago a biodiversidade, quantidade de seres vivos de diferentes espécies, não é muito grande se comparada com o resto da natureza, por este motivo a capacidade de recuperação de um lago frente a problemas ecológicos, como a poluição, se dá de forma lenta e gradativa, e em alguns casos o problema pode se tornar irreversível.

Concluí-se então que um lago é um ambiente adequado para representação dos problemas ecológicos a nível didático pois permite que, sem muitas restrições, se represente uma situação muito próxima da realidade.

No software estamos simulando o ambiente de um lago situado a beira de uma cidade em crescimento, onde o desafio proposto é o de manter o equilíbrio do ecossistema do lago. Algumas espécies de peixes e algas estão presentes no jogo de forma a ilustrar a interação dos elementos interventores (poluentes), com a teia alimentar do lago.

4. Os Personagens do jogo

O objetivo de o jogo ter diferentes personagens parte de uma comparação com a realidade, sendo que dentro das limitações de enfoque e modelagem, buscamos atribuir ferramentas aos personagens que façam que o aluno identifique qual o papel de cada um no contexto social.

É importante salientar que não está se afirmando que, na vida real, os personagens tem realmente os devidos poderes atribuídos a eles no jogo, apenas estamos enumerando os diferentes tipos de poluição existentes, e algumas das possíveis soluções, relacionando que elementos da sociedade tem uma maior proximidade com essas soluções.

Podemos afirmar que num âmbito geral as ações dos personagens possuem dois objetivos pedagógicos básicos:

- Chamar atenção para os poluentes mais comuns a nossa realidade, e possíveis soluções para eles.
- Permitir que o usuário identifique as ações que são positivas ou negativas para o equilíbrio do ambiente.

O usuário escolhe qual o seu personagem no início do jogo. Conforme ilustra a figura 1. Cada personagem possui ferramentas com objetivos específicos que explicamos a partir de agora.

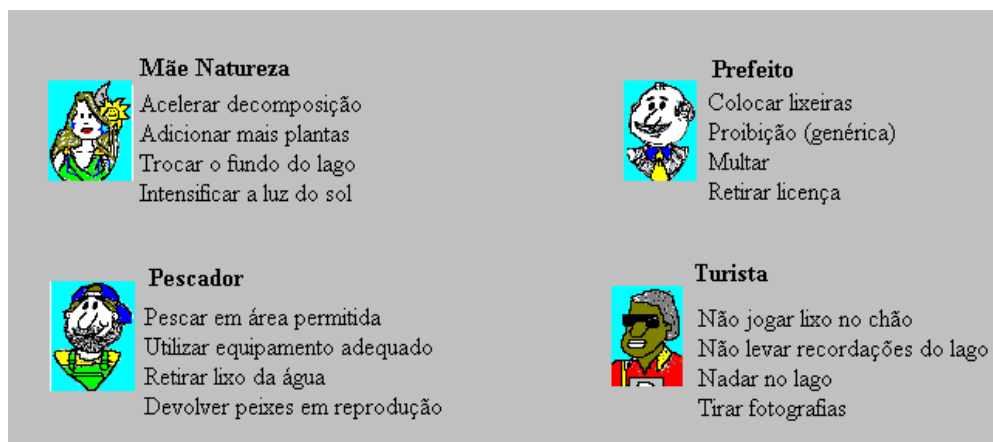


FIGURA 2: personagens e suas ferramentas.

Mãe Natureza

- *Acelerar degradação*: Mostrar que o processo de degradação é a única maneira de o meio ambiente se recompor de certos poluentes e que este é um processo natural, porém lento.
- *Colocar Aguapés*: Mostrar que os aguapés são algas que colaboram positivamente no estancamento da poluição de um lago ou rio.
- *Trocar o Fundo do lago*: Mostrar que a poluição fica mais concentrada na areia do fundo do lago.
- *Abrir Sol*: Mostrar que o sol aumenta a fotossíntese e conseqüentemente a quantidade de oxigênio no lago.

Prefeito

- *Colocar Lixeiras*: Mostrar que um governante pode colocar lixeiras diminuindo a quantidade de lixo que vai para o lago induzindo as pessoas a terem um hábito de procurar guardar o lixo em local adequado.

- *Proibição*: Mostrar que um governante pode proibir algumas atividades que são poluentes. Tais como a navegação motorizada (lanchas e Jet-Ski) e a utilização da draga.
- *Multar*: Mostrar que o governante pode multar uma ação danosa ao meio ambiente e isto pode fazer com que as pessoas parem para pensar no que estão fazendo;
- *Retirar licença*: Mostrar que o governante pode obrigar uma empresa a instalar tratamento de fluídos, através da retirada de sua licença e que essa é a uma saída para o problema do desequilíbrio causado pelas indústrias que não se preocupam com o meio ambiente sejam elas de que tamanho forem);

Pescador

- *Pesca predatória*: Mostrar que a pesca de arrastão é uma prática predatória e que prejudica o ambiente.
- *Pescar em Área protegida*: Mostrar que as áreas protegidas possuem espécies em extinção e conseqüentemente não devem ser pescadas.
- *Pescar para se alimentar*: Mostrar que o pescador que pesca para se alimentar não afeta o equilíbrio do meio ambiente.
- *Pescar em época permitida*: Mostrar que existem épocas para pesca definidas por especialistas e que devem ser respeitadas.

Turista

- *Não Jogar Lixo*: Identificar o lixo, por menor que seja, como um elemento poluente, e mostrar que não se deve jogar lixo em lugar inadequado. Não jogar lixo é uma ação boa para manutenção do equilíbrio do lago;
- *Não Levar Lembrança do Lago*: Mostrar que a prática, comum de muitos os que estão praticando o turismo, de levar pedras, conchas, plantas, animais, enfim qualquer coisa que faça parte do ecossistema do lugar é prejudicial ao mesmo pois afeta o seu equilíbrio.
- *Nadar no lago*: Mostrar que nadar no lago não prejudica o ecossistema, e que a prática de um esporte é sempre positiva a menos que sejam utilizados equipamentos poluidores (tanto sonoros como químicos). Ex.: Jet-Ski.
- *Tirar fotografia*: Mostrar que documentar as paisagens através da máquina fotográfica não oferece perigo nenhum ao meio ambiente, e inclusive pode levar ao conhecimento de outras as pessoas as belezas de um local.

5. Os Poluentes

Dentre os muitos poluentes existentes elegeu-se aqueles que se relacionam melhor com as ferramentas dadas aos personagens e que conseqüentemente representam aqueles que mais com mais freqüência os ecossistemas de lagos e rios a beira das cidades. A figura 3 apresenta os poluentes utilizados no sistema.

O sistema escolhe por sorteio os elementos poluentes que estarão presentes no ambiente. O usuário (aluno ou professor) pode fazer a configuração da quantidade de poluentes e quando eles irão aparecer através do arquivo de configuração que os sistema usa para jogar (veja mais adiante no manual como se faz isto).

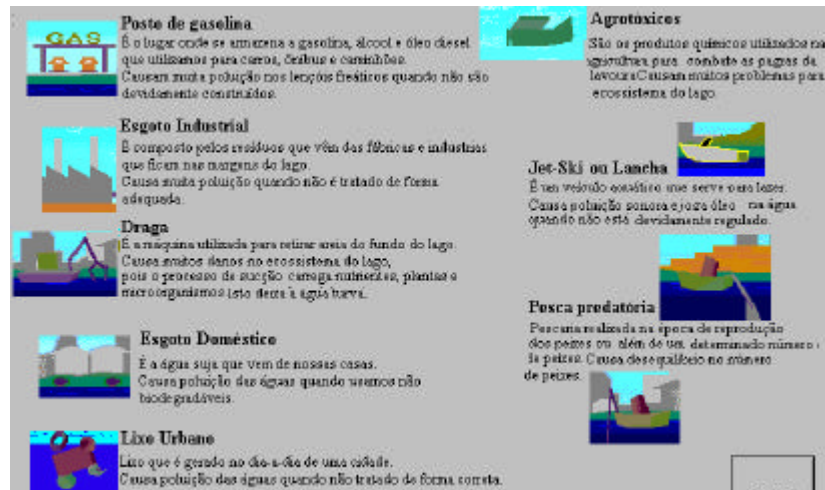


FIGURA 3: A tela de escolha dos elementos interventores.

Cada elemento afete de forma diferente o ecossistema. Esta característica está representada no item Fator de poluição ilustrado na TABELA abaixo.

Cada elemento interventor possui uma característica particular que o identifica como sendo de pouco ou muito risco. Essas características foram definidas junto aos especialistas em Ecologia Ambiental da Universidade Federal do Rio Grande do Sul, e seus pesos foram definidos em virtude da nossa realidade local (estuário do Guaíba) tendo em vista que muitos fatores que não são representados no jogo influenciariam nesses pesos (como os componentes químicos dos poluentes) e que não existe um consenso geral entre os especialistas sobre a exatidão desses pesos.

Alguns elementos incluídos no jogo não são efetivamente poluentes perigosos, mas sua presença se faz importante com o objetivo de ilustrar que mesmo utensílios de lazer como o Jet-Ski e a lancha podem trazer prejuízos ao ambiente, mesmo que pequenos.

Os objetivos específicos de cada elemento são detalhados a seguir:

- **Draga** - Mostrar que remexer a areia do fundo do lago traz prejuízos aos peixes que habitam as proximidades.
- **Lixo** - Mostrar que qualquer lixo, mesmo um insignificante papel de bala que se joga no chão acabará chegando no lago e ocasionando desequilíbrio.
- **Clube Náutico** - O Jet-Ski e a lancha tem o mesmo objetivo de mostrar que trazem leves prejuízos ao ambiente já que deixam combustível na superfície do

lago, além de ocasionarem estresse auditivo nos peixes o que dificulta a reprodução dos mesmos.

- Pesca Predatória

- Pesca em áreas proibidas - mostrar que esta pode trazer a extinção de espécies e este fato representa um grande risco pois impede o restabelecimento do ecossistema mesmo com o estancamento da poluição.

- Rede de arrastão - mostrar que a pesca de arrastão é uma prática predatória e incompatível com qualquer ecossistema equilibrado.

- Esgoto Pluvial - mostrar que o esgoto que sai das casas e sarjetas é um forte contribuinte para a poluição no lago.

- Esgoto Industrial - mostrar que os dejetos proveniente de indústrias sem tratamento trazem um forte desequilíbrio ao ambiente, e que os dejetos quentes além de poluírem alteram a temperatura da água o que também é negativo pois altera o habitat dos peixes que vivem nas proximidades do local que recebe o dejetos.

- Agrotóxicos - mostrar que a utilização de agrotóxicos é mais um fator que dificulta a obtenção do equilíbrio pois afeta não só as pragas da lavoura, mas também todo ecossistema da região.

- Postos de Gasolina - mostrar que o combustível armazenado e utilizado em postos de gasolina que não possuem caixa de areia acaba por infiltrar nos lençóis freáticos e termina indo até o lago, tornando um forte fator de desequilíbrio.

6. O esquema da energia no sistema

Todo o esquema é baseado no nível de energia dos elementos do cenário. São sorteados elementos interventores no cenário e estes começam a atuar com intervalos de tempo definidos pelo usuário no arquivo de dados.

6.1 Esquema da reprodução

O peixe A come o peixe C e isto acontece quando ele passa perto de C e este está sozinho. Seu nível de energia deve estar: 80% a 100%.

O peixe C come microorganismos e isto corre quando ele passa por um microorganismo.

Os peixes andam em grupos reunidos por espécie. Vai se colocar uma aleatoriedade de movimentos de maneira a permitir que de tempo em tempo uma certa quantidade (1/30 se espalhe permitindo que os predadores comam e a cadeia alimentar ocorra.

Os microorganismos também se movimentam, em movimentos aleatórios.

PEIXE		PLANTAS	
Nível de energia %	ação	Nível de energia %	ação
80-100	reproduz, alimenta, movimentada	70-100	reproduz, movimentada
60-79	alimenta, movimentada	50-69	movimentada
40-59	alimenta	10-49	muda de cor e imóvel
10-30	muda de cor	menos de 10	Morre
menos de 10	morre		

MICROORGANISMOS	
Nível de energia %	ação
70-100	reproduz, movimentada
50-69	movimentada
0-49	morre

Obs.:

Para a planta continuar a viver ela depende das condições da água e do fundo do lago.

Para o peixe B viver ele depende dos microorganismos. E estes dependem das condições da água.

Para o peixe A viver ele depende do peixe C, que depende das plantas.

6.2 Retirada e reposição de energia

As tabelas 1 e 2 mostram como a energia é retirada pelos poluentes e como as ferramentas agem para recompor as perdas causadas pela poluição. Estas regras foram utilizadas para criar o mecanismo de funcionamento do programa e além destas existem os mecanismos criados para manter a cadeia alimentar e todo ciclo de vida dos elementos do cenário.

TABELA 1: Retirada de energia

Elemento do Cenário Elemento Interventor	Peixe 1	Peixe 2	Peixe 3	Plantas	Microorganismos	Água	Fundo do lago
Draga				-20	-30		-30
Lixo Urbano	-5	-5	-5	-10	-5	-15	-10
Jet-Ski ou lancha			-10		-10		
Pesca Predatória	-30	-20	-10		-10		
Esgoto domestico	-10	-10	-10	-10	-10	-10	-10
Esgoto Industrial				-20	-20	-20	-20
Produtos tóxicos	-20	-20	-20	-20	-20	-20	-20
Posto de Gasol.	-20	-20	-20	-20	-20	-20	-20

TABELA 2: Reposição de energia

Elemento do Cenário	Peixe 1	Peixe 2	Peixe 3	Planta A	Planta B	Micros	Água	Fundo do lago
Ferramentas								
Acelerar degradação	+10	+10	+10	+20	+10	+20	+10	+10
Intensificar a luz do sol	+15	+15	+15	+15	+15	+15	+15	+15
Adicionar mais plantas	+5	+5	+5	+10	+5	+15	+15	+5
Trocar fundo	+5	+5	+5	+10	+5	+15	+15	+5
Colocar lixeiras	+5	+5	+5	+5	+10	+5	+15	+10
Proibição (genérica)	+5	+10	+10	+10	+10	+10	+10	+10
Multar	+10	+20	+20	+15	+20	+10	+10	+10
Retirar licença	+15	+20	+20	+15	+20	+20	+10	+10
Pescar em área permitida	+10	+10	+10	+5	+5	+5	+5	+5
Utilizar equipa. adequado	+20	+20	+15	+5	+5	+5	+5	+5
Retirar lixo da água	+15	+20	+15	+10	+5	+10	+20	+10
Devolver peixes em reprodução	+20	+20	+15	+5	+5	+5	+5	+5
Não jogar lixo no chão	+5	+5	+5	+15	+5	+15	+20	+15
Não levar recordações do lago	+5	+5	+5	+10	+5	+15	+10	+10
Nadar no lago	+10	+10	+10	+5	+10	+10	+10	+5
Tirar fotografias	+10	+10	+10	+5	+10	+10	+10	+5

7. Como configurar o MCOE para jogar

Após ativar MCOE aparecerá a tela de apresentação do jogo. Na barra de programas selecione JOGO para ter acesso as principais funções ativadas nesta etapa.

A figura 4 apresenta as opções disponíveis.

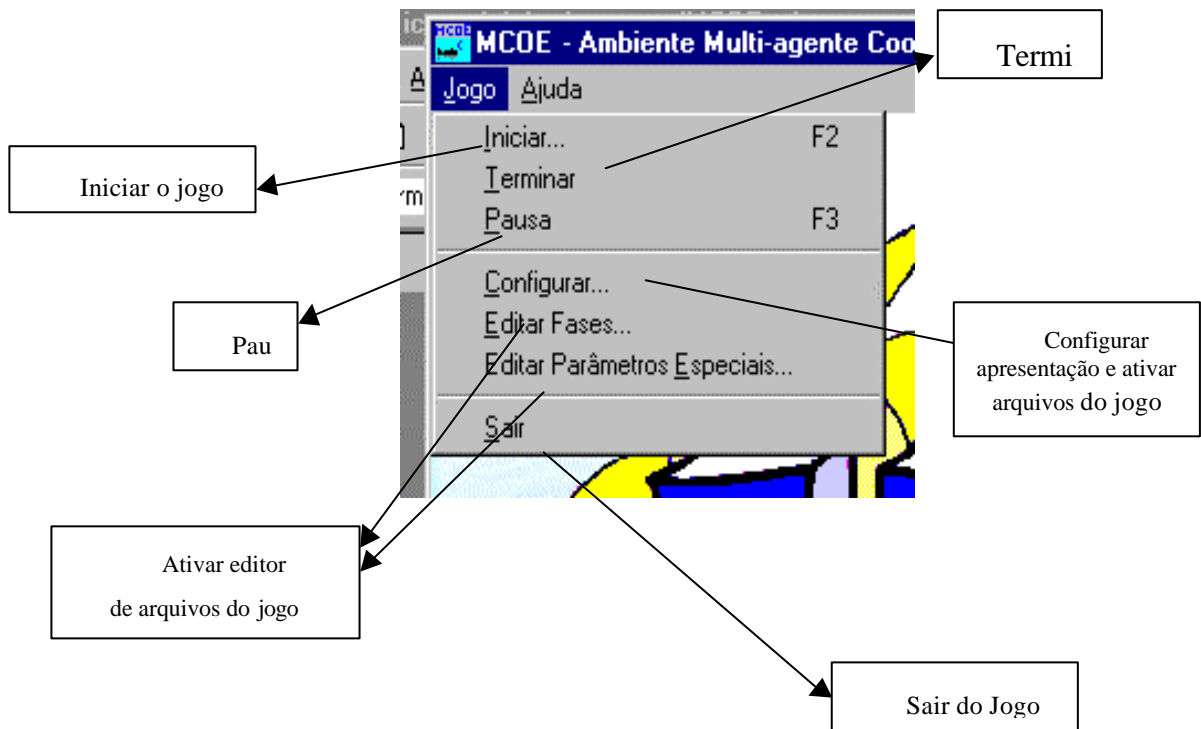


FIGURA 4: Opções do Jogo

A opção CONFIGURAR permite seleccionar o tamanho da janela de jogo do MCOE e o arquivo que será usado no jogo (poluidores e duração da partida). A figura 5 apresenta a janela de CONFIGURAR.



FIGURA 5: Janela de configuração

A opção EDITOR permite que o usuário crie o arquivo que será usado no jogo. A figura 6 apresenta a janela do EDITOR.

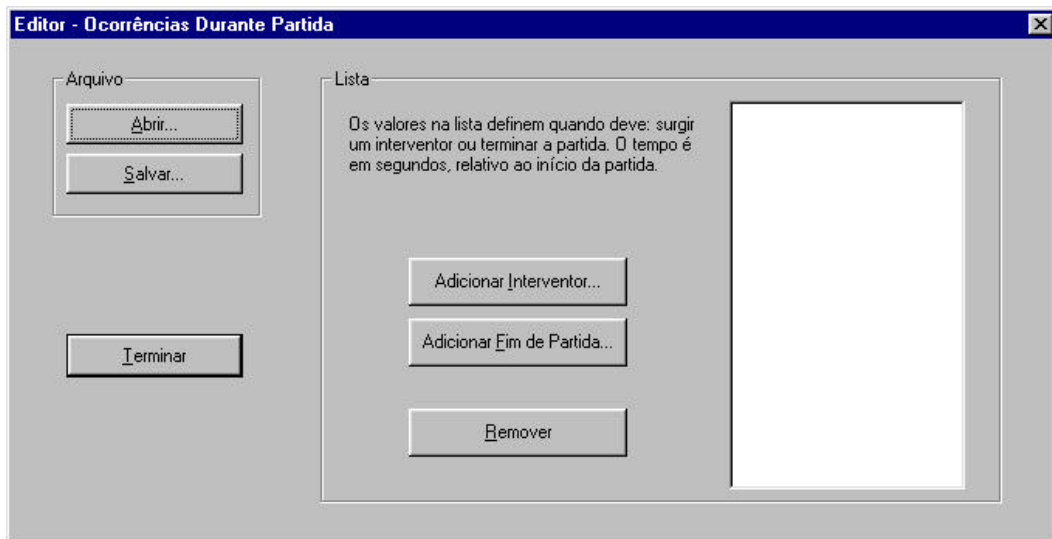


FIGURA 6: Janela do Editor

Para criar um novo arquivo de partida basta apertar o botão ADICIONAR INTERVENTOR e escolher o tempo do jogo onde tu desejas que o interventor (a ser escolhido pelo programa) vai aparecer. Veja um exemplo deste procedimentos na figura 7.

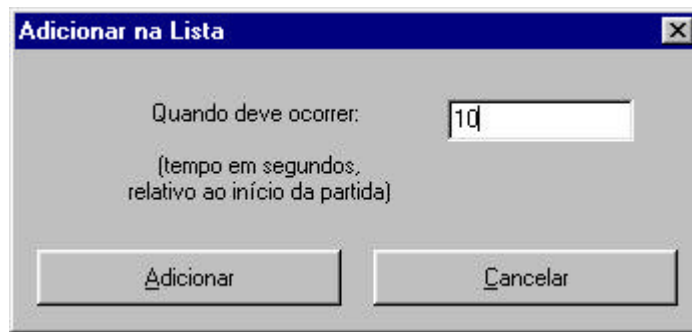


FIGURA 7: Adicionando um interventor

O usuário poderá colocar quantos interventores desejar . desta maneira a complexidade do Jogo é definida (em termos de grau de poluição e conseqüente grau de dificuldade para os jogadores combaterem a poluição causada no lago por estes interventores.

O botão ADICIONAR FIM DE PARTIDA serve para o programa saber em que tempo o jogo deve terminar.

O botão REMOVE serve para o usuário remover algum item colocado no arquivo e que não deseja mais que seja ativado. Para remover basta selecionar o item A figura 8 apresenta um exemplo de arquivo utilizando o editor.

Para salvar o novo arquivo criado pelo usuário use a opção SALVAR. Todos os arquivos criados são armazenados na pasta DP (dados do programa) que é criada automaticamente quando da instalação do MCOE. Os arquivos possuem a extensão **.com**

Sempre que um arquivo for salvo o sistema perguntará se ele deve fazer parte da lista de opções do MCOE. Responda sim para que o arquivo seja salvo ou não caso ele já faça parte da lista .

Eventualmente o usuário pode desejar editar um arquivo existente e poderá fazer isto através do botão ABRIR.

A figura 9 apresenta a janela associada à opção ABRIR e SALVAR.

FIGURA 8: Exemplo de arquivo criado para partida



FIGURA 9: janelas das opções SALVAR COMO e ABRIR

O arquivo criado deve ser colocado na janela de configuração . Após a digitação do nome do arquivo que o programa vai utilizar, o usuário deve fechar a janela e depois sair do programa a fim de que as configurações sejam assumidas pelo programa.

A opção EDITAR PARÂMETROS ESPECIAIS permite que o usuário altere o número de vezes que uma ferramenta pode ser utilizada. Ou seja, o poder de combate de um personagem pode aumentar ou diminuir através da quantidade de vezes que ele pode utilizar uma determinada ferramenta. A figura 10 apresenta a janela associada à opção EDITAR PARÂMETROS ESPECIAIS.

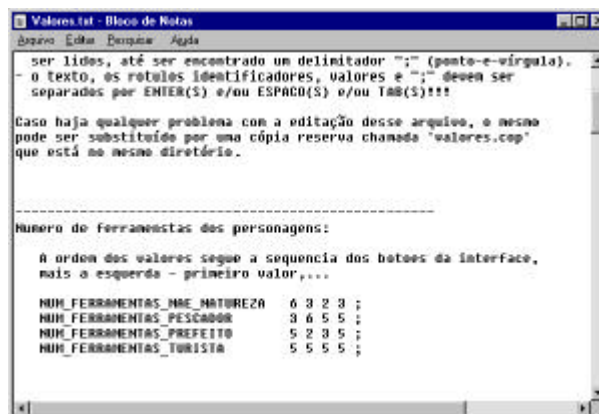


FIGURA 10: Edição de parâmetros especiais**8. A escolha dos personagens**

Ao selecionar a opção INICIAR partida aparecerá na tela a janela apresentada na figura 11, onde o primeiro jogador escolhe o seu personagem. Após a seleção o jogador deve apertar o botão INICIAR PARTIDA.

**FIGURA 11: Escolha do primeiro personagem**

A seguir, o segundo jogador escolhe entre os três personagens restantes o que vai jogar (figura12).

FIGURA 12: Escolha do segundo personagem

9. O ambiente do jogo

O jogo inicia com a ativação do arquivo selecionado pelo usuário (*nome_de_arquivo.mco*), a inicialização do jogo com todos os seus elementos, o Ecômetro apresentando cor relativa ao estado atual da energia do ecossistema, os bitmaps que representam os personagens escolhidos pelos dois jogadores, os botões das ferramentas associadas a cada personagem, os botões de ajuda e informações sobre o *kernel* (núcleo) cognitivo do sistema onde são processados os estados mentais. A figura 13 apresenta a tela do jogo no início de uma partida, onde foram selecionados os personagens Mãe natureza e Prefeito.

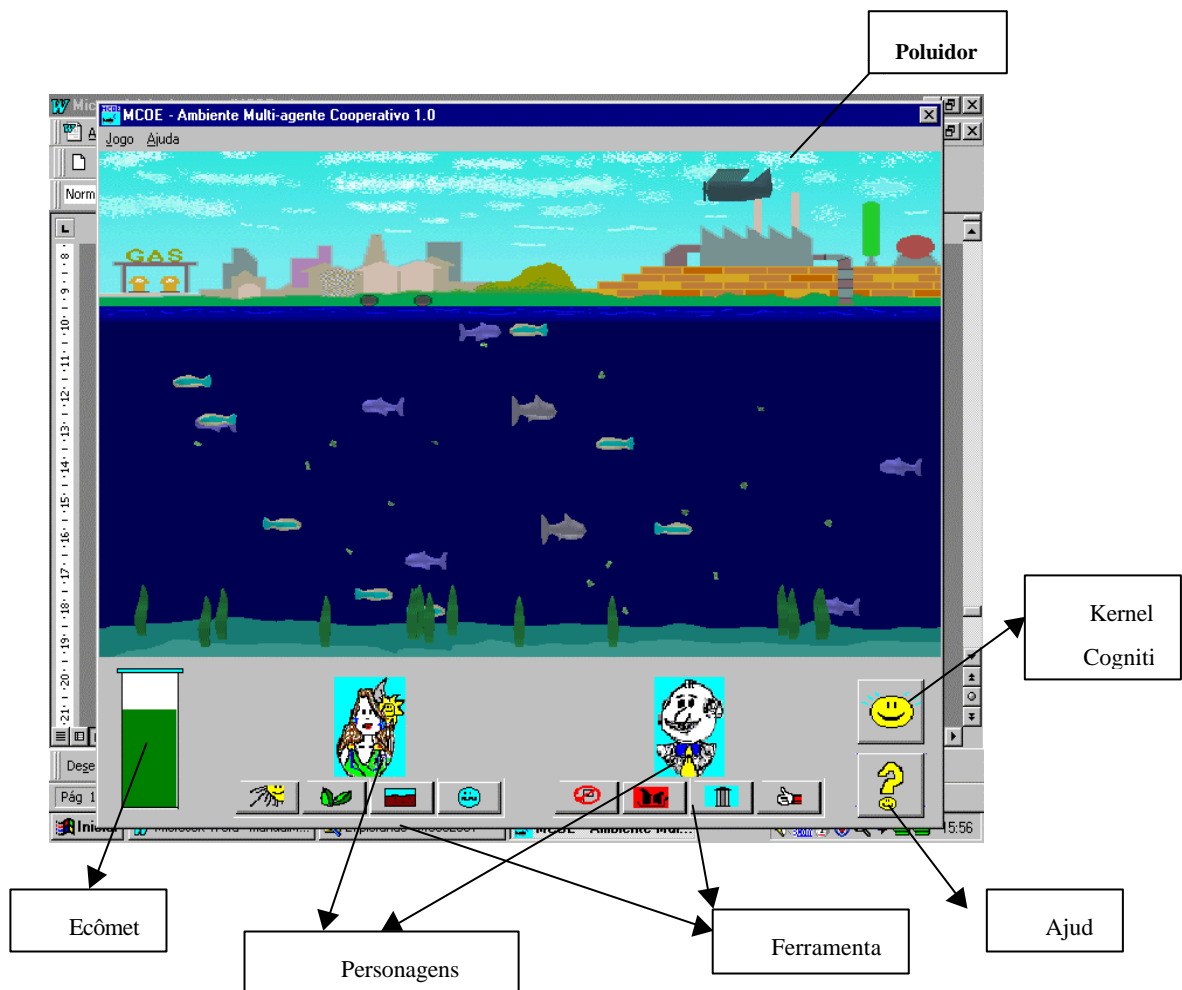


FIGURA 13: Interface do MCOE

A figura 14 apresenta a tela do jogo após o encerramento da partida. Isto pode acontecer de duas maneiras:

- Um dos jogadores decide terminar a partida (através do menu de comandos);
- O tempo configurado no arquivo do jogo se esgota.



FIGURA 14 : Tela ao final do jogo

9.1 Cálculo do Nível de Energia do Ecômetro

O ecômetro é um recurso da interface do MCOE que mostra ao usuário o nível geral de energia do cenário.

Ao clicar com o botão direito sobre o ecômetro, o usuário recebe textualmente a informação do nível de energia geral numa escala de 0 a 100.

Tanto a altura da barra de nível de energia do ecômetro quanto o valor do texto são calculados a partir da quantidade de energia de cada elemento do cenário: peixes, algas, microorganismos, fundo do lago e água. Cada um desses elementos possui um nível de energia individual que varia conforme a sua interação com o meio. Por exemplo, se um peixe se alimenta de outro isso aumenta o seu nível de energia, mas mantém inalterado o nível de energia dos demais peixes.

O nível de energia do ecômetro é determinado em duas etapas.

Primeiramente é calculada a soma do nível de energia individual de cada elemento do cenário. Isso fornece o nível total de energia do cenário naquele instante de tempo.

O segundo passo consiste em transformar esse valor para uma escala de 0 a 100. Como cada elemento do cenário tem um limite máximo de energia e o cenário possui uma limitação quanto ao número máximo de seres, é possível determinar o nível máximo de energia. Ocorreria se o cenário estivesse com o máximo de seres, todos com o máximo de energia. Assim basta fazer um cálculo de proporções:

$$\text{nível_total_de_energia} / \text{nível_máximo_de_energia} * 100$$

10. O sistema de ajuda



O sistema de ajuda pode ser ativado diretamente do botão

ou na barra de comandos na janela principal do jogo. A FIGURA 14 apresenta a janela



FIGURA 14: Sistema de ajuda

Cada imagem que aparece na janela do sistema de ajuda é um botão que ativa outra janela onde o usuário acessa informação a respeito do jogo e do seu conteúdo (domínio da aplicação). A figura 15 apresenta um exemplo de conteúdo que pode ser encontrado na ajuda.

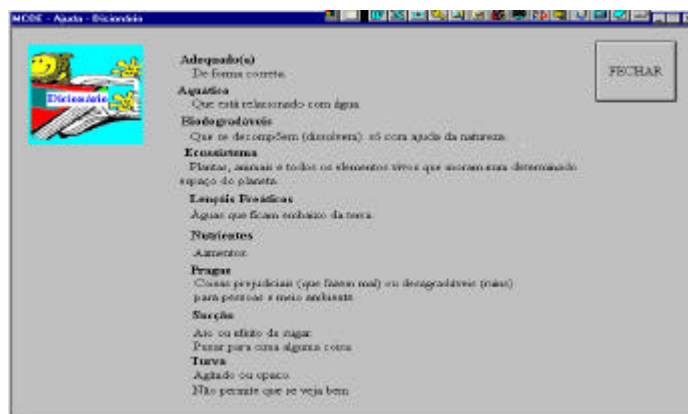


FIGURA 15: Exemplo de conteúdo disponível na ajuda do MCOE

11. Requisitos de hardware e Software para utilizar o MCOE

11.1 Requisitos de hardware

- Microcomputador com processador Pentium 166 ou superior
- Memória RAM de no mínimo 20Mb
- Disco rígido com espaço disponível de no mínimo 20Mb
- Teclado
- Mouse
- Kit Multimídia
- Monitor colorido SVGA com resolução 800x600 e configurado para 256 cores

11.2 Requisitos de software

- Sistema operacional Windows 95 ou superior
- Ambiente de execução do DirectX

12. O tutor do MCOE (apenas na versão 2.0)

No início do jogo aparece uma janela com a opção de jogar com ou sem o auxílio do tutor.

Quando o tutor é ativado os jogadores passam a ter suas ações monitoradas pelo tutor que fará intervenções de acordo com as ações realizadas por cada aluno.

As mensagens do tutor aparecem em janelas disponibilizadas apenas na versão 2.0.

O tutor utiliza os estados mentais associados à cada ação dos alunos para fazer um perfil individualizado do aluno e poder tomar a decisão de como ele deve se comportar face às necessidades de cada aluno. O tutor pode ser mais diretivo ou menos diretivo dependendo do tipo de aluno e a situação em que se encontra o ambiente.

Os estados mentais associados às ações dos alunos podem ser vistos pressionando apresentado na FIGURA 16.



FIGURA 16: Botão dos estados mentais.

13. Informações sobre o grupo de pesquisa em IA-ED

O grupo de pesquisa em Inteligência Artificial aplicada á Educação (IA-ED) é coordenado por:

Prof.a. Dra. Rosa Maria Viccari

CPGCC/UFRGS

rosa@inf.ufrgs.br

Programa de Pós-Graduação em Computação – UFRGS

Av. Bento Gonçalves 9500 – Bloco IV – Porto Alegre – RS – Brasil

Fazem parte deste grupo todos os alunos de Iniciação Científica, mestrados e doutorandos que trabalham em projetos diretamente ligados à pesquisa envolvendo ensino e/ou aprendizagem suportado por ambientes computacionais e que utilizam técnicas de Inteligência Artificial no seu projeto.

Este manual é parte integrante do projeto de doutoramento da aluna Lucia Maria Martins Giraffa, professora da FACIN/PUCRS.

Mais detalhes a cerca deste projeto podem ser obtidos em:

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

giraffa@inf.ufrgs.br

14. Bibliografia do manual do usuário

- [GIR97] Giraffa, L.M.M.; Nunes, M. A.; Viccari, R.M. Multi-Ecological: an Intelligent Learning Environment using Multi-Agent architecture. MASTA'97: Multi-Agent System: Theory and Applications. **Proceedings...** Coimbra: DE-Universidade de Coimbra, 1997.
- [GIR98] Giraffa, L.M.M; Viccari, R.M.; Self,J. Improving tutoring activities using a Multi-Agents system Architecture. Twenty-ninth SIGCSE Technical Symposium on Computer Science Education, 29,1998. **Proceedings...** Atlanta: Georgia, February 25-March 2,1998.
- [GIR98a]Giraffa, L.M.M; Viccari, R.M.; Self. J. Multi-Agent based pedagogical games. ITS'98- Fourth International Conference on Intelligent Tutoring Systems. **Proceedings...** San Antonio: Texas, 1998.
- [GIR98b] Giraffa, L.M.M; Móra, M.;Viccari, R.M Modelling the MCOE Tutor using a Computational Model. In: **Lectures Notes on Artificial Intelligence - SBIA'98**. Oliveira,F.(Ed.). Berlin: Springer Verlag, 1998.
- [GIR98c] Giraffa, L.M.M; Móra, M.;Viccari, R.M Pedagogical game using ITS architecture. IBERAMIA: Workshop on Artificial Intelligence. Toledo: 1998.

- [MOR97] Móra, M.C.; Lopes, J.G.; Coelho, J.G.; Viccari, R. Modelling dynamic aspects of intentions. In E.Costa, editor, 8th Portuguese Conference on Artificial Intelligence. Springer-Verlag, **Proceedings...**, 1997.
- [MOR98a] Móra, M.C.; Lopes, J.G.; Coelho, J.G.; Viccari, R. Modelling agents with extended logic programa. In **International Workshop on Engineering of Intelligent Systems** . ICSC co., 1998.
- [RAA96] Raabe, A.L.A; Javimczik, A .M.; Giraffa, L.M.M. Eco-Lógico: Ambiente interativo para suporte ao ensino de educação ambiental. In: VII Simpósio Brasileiro de Informática na Educação, 7., 1996. **Anais ...** Belo Horizonte: DCC/UFMG, 1996.

Anexo 3 – Questionário

DADOS DE IDENTIFICAÇÃO

Localização da escola: Porto Alegre () Grande Porto Alegre ()
 Tipo de escola: Particular () Escola Pública ()
 Iniciais do nome do (a) professor (a) : _____
 Sexo: Feminino () Masculino ()
 Idade: _____
 Escolaridade: Magistério () Pedagogia () Licenciatura em Ciências ()
 Outro ()
 Série que leciona no 1º grau : 3ª () 4ª () 5ª ()

QUESTIONÁRIO

Que tipo de exercícios você utiliza para introduzir o estudo de poluição com seus alunos?

- () vídeos
- () trabalho com colagem e desenhos feitos pelos alunos
- () passeios a lugares poluídos e locais saudios
- () programas de computador
- () outros (*favor especificar*)

Que tipo de exercícios você utiliza para complementar o estudo de poluição com seus alunos?

- () vídeos
- () trabalho com colagem e desenhos feitos pelos alunos
- () passeios a lugares poluídos e locais saudios
- () programas de computador
- () outros (*favor especificar*)

-
-
-
1. Qual a frequência que você utiliza programas de computador (software educacional) como tarefa complementar às atividades de sala de aula?
 - () Até 5 vezes ao ano
 - () Mais de 5 vezes ao ano
 - () Não conto quantas vezes utilizo

 2. Os programas de computador selecionados por você para seu ensino junto aos seus alunos, estão escritos em português?
 - () sim/todos estão escritos em português
 - () geralmente/ a maioria das vezes estão escritos em português
 - () às vezes estão escritos em português
 - () quase nunca estão escritos em português
 - () não/nenhum/nunca estão escritos em português

 - () isto não é importante/significativo

 3. Os programas de computador selecionados por você para seu ensino junto aos seus alunos, possuem características semelhantes a jogos?
 - () sim, todos possuem
 - () geralmente/a maioria das vezes possuem
 - () às vezes possuem
 - () quase nunca possuem
 - () não, não possuem

 - () isto não é importante/significativo

 4. Os programas de computador selecionados por você para seu ensino junto aos seus alunos, usam recursos sonoros?
 - () sim, todos usam
 - () geralmente/ a maioria das vezes usam
 - () às vezes usam
 - () quase nunca usam
 - () não, nunca usam

- isto não é importante/significativo
5. Os programas de computador selecionados por você para seu ensino junto aos seus alunos, usam imagens em movimento (animações) ?
- sim, todos usam
- geralmente/ a maioria das vezes usam
- às vezes usam
- quase nunca usam
- não, nunca usam
- isto não é importante/significativo
6. Os programas de computador selecionados por você para seu ensino junto aos seus alunos, possibilitam aos seus alunos interagirem com o programa?
- sim, todos possibilitam
- geralmente/ a maioria das vezes possibilitam
- às vezes possibilitam
- quase nunca possibilitam
- não, nunca possibilitam
- isto não é importante/significativo
7. Os programas de computador selecionados por você para seu ensino junto aos seus alunos, registram o trabalho dos seus alunos durante sua utilização ?
- sim, todos registram
- geralmente/maioria das vezes registram
- às vezes registram
- quase nunca registram
- não, nunca registram
- isto não é importante/significativo
8. Os programas de computador selecionados por você para seu ensino junto aos seus alunos, permitem ao professor poder adequar o tempo do programa aos seus objetivos ?
- sim, todos permitem

- geralmente/a maioria das vezes permitem
- às vezes permitem
- quase nunca permitem
- não, nunca permitem
- isto não é importante/significativo
9. Os programas de computador selecionados por você para seu ensino junto aos seus alunos, funcionam de maneira diferente a cada vez que os seus alunos os utilizam?
- sim, todos funcionam
- geralmente/a maioria das vezes funcionam
- às vezes funcionam
- quase nunca funcionam
- não, nunca funcionam
- isto não é importante/significativo
10. Os programas de computador selecionados por você para seu ensino junto aos seus alunos, possuem um sistema de ajuda para entender como funciona o programa?
- sim, todos possuem
- geralmente/a maioria das vezes possuem
- às vezes possuem
- quase nunca possuem
- não, nunca possuem
- isto não é importante/significativo
11. Os programas de computador selecionados por você para seu ensino junto aos seus alunos, possuem um sistema de ajuda para auxiliar o aluno a entender o conteúdo trabalhado pelo programa?
- sim, todos possuem
- geralmente/a maioria das vezes possuem
- às vezes possuem
- quase nunca possuem
- não, nunca possuem

- isto não é importante/significativo
12. Os programas de computador selecionados por você para seu ensino junto aos seus alunos, devem possuir um sistema de ajuda para auxiliar o aluno a entender o conteúdo trabalhado pelo programa?
- sim, todos devem possuir
- geralmente/ a maioria das vezes devem possuir
- às vezes devem possuir
- quase nunca devem possuir
- não, nunca devem possuir
- isto não é importante/significativo
13. Os programa de computador Eco-Lógico e MCOE possibilitam aos seus alunos interagirem entre si?
- sim, ambos possibilitam a interação entre os alunos
- só programa MCOE possibilita a interação entre os alunos
- só programa Eco-Lógico possibilita a interação entre os alunos
- não, nenhum deles possibilita a interação entre os alunos
14. Os programa de computador Eco-Lógico e MCOE, permitem ao professor poder adequar o tempo do programa aos seus objetivos ?
- sim , ambos possibilitam a adequação do tempo
- só programa MCOE possibilita a adequação do tempo
- só programa Eco-Lógico possibilita a adequação do tempo
- não, nenhum deles possibilita a adequação do tempo
15. Os programa de computador Eco-Lógico e MCOE, se apresentam de maneira diferente a cada vez que os seus alunos os utilizam?
- sim, ambos se apresentam de maneira diferente
- só programa MCOE se apresenta de maneira diferente
- só programa Eco-Lógico se apresenta de maneira diferente
- não, nenhum deles se apresenta de maneira diferente
16. O objetivo do jogo Eco-Lógico é claro para o usuário na sua apresentação?
- sim, o objetivo é claro e não necessita de explicações adicionais
- o objetivo não está claro e necessita de explicações adicionais

17. O objetivo do jogo MCOE é claro para o usuário na sua apresentação?
- sim, o objetivo é claro e não necessita de explicações adicionais
 - o objetivo não está claro e necessita de explicações adicionais
18. O jogo Eco-Lógico utiliza uma representação adequada do ecossistema de um lago?
- sim, a representação está adequada
 - a representação está parcialmente adequada
 - a representação não está adequada
19. O jogo MCOE utiliza uma representação adequada do ecossistema de um lago?
- sim, a representação está adequada
 - a representação está parcialmente adequada
 - a representação não está adequada
20. Comprando os dois jogos (Eco-Lógico e MCOE) qual o mais próximo de representar a vida num lago?
- O jogo Eco-Lógico
 - O jogo MCOE
 - ambos são igualmente representativos
 - nenhum é representativo
21. Qual dos dois ambientes possui mais conteúdo relevante para ser explorado por parte do professor numa atividade de ensino junto aos seus alunos?
- O ambiente do Eco-Lógico
 - O ambiente do MCOE
 - ambos possuem conteúdos equivalentes
 - nenhum possui conteúdo relevante a ser explorado
22. Qual dos ambientes na sua opinião é mais dinâmico em termos de movimento dos elementos do jogo?
- O ambiente do Eco-Lógico
 - O ambiente do MCOE
 - ambos são igualmente dinâmicos
 - nenhum dos dois é dinâmico

23. Dos dois programas que você analisou qual o mais fácil de ser entendido num primeiro momento?

- O ambiente do Eco-Lógico
- O ambiente do MCOE
- ambos são igualmente fáceis de serem entendidos
- ambos são igualmente difíceis de serem entendidos

24. Qual dos dois programas é mais adequado ao tipo de trabalho que você desenvolve?

- O programa Eco-Lógico
- O programa MCOE
- ambos são igualmente adequados
- nenhum dos dois é adequado

Se você quiser fazer um comentário, use este espaço para alguma consideração que acha importante ou coloque sugestões que você julga importantes para melhoria do programa **Eco-Lógico**(utilize o verso da folha se necessário).

Se você quiser fazer um comentário, use este espaço para alguma consideração que acha importante ou coloque sugestões que você julga importantes para melhoria do programa **MCOE** (utilize o verso da folha se necessário).

Obrigada pela sua colaboração

Anexo 5 – Exemplo do conteúdo do arquivo do tutor

```

bel(tutor, tempo_restante(Tr)) if
    bel(tutor, tempo_partida(Tp),
        bel(tutor, tempo(T),
            Tr is Tp-T.

bel(tutor, tempo_folga_1(Tx1))if
    bel(tutor, tempo_partida(Tp),
        Tx1 is Tp*0.8.

bel(tutor, tempo_folga_2(Tx2))if
    bel(tutor, tempo_partida(Tp),
        Tx2 is Tp*0.2.

bel(tutor, tempo_folga_3(Tx3))if
    bel(tutor, tempo_partida(Tp),
        Tx3 is Tp*0.5.

/* alerta do tempo */

des(tutor, alertar_tempo) if
    bel(tutor, tempo_restante(Tr)),
    Tr <= Tx2.

/* envio dos identificadores de mensagens */

des(tutor, enviar_mensagem(X)).

bel(tutor, enviar_mensagem(1)) if
    bel(tutor,INT_THAT(Aluno, receber_auxilio),
        bel(tutor, energia(ecometro, Ee)),
        Ee < 70,
        bel(tutor, tempo_folga_1(Tx1)).

bel(tutor, enviar_mensagem(2))if
    bel(tutor, energia(peixeA, PA)),
    PA < 80,

```

```
bel(tutor, tempo_folga_3(Tx3)).
```

```
bel(tutor, enviar_mensagem(2)) if  
    bel(tutor, energia(peixeB, PB)),  
    PB < 80,  
    bel(tutor, tempo_folga_1(Tx3)).
```

```
bel(tutor, enviar_mensagem(2)) if  
    bel(tutor, energia(peixeC, PC)),  
    PC < 80,  
    bel(tutor, tempo_folga_3(Tx3)).
```

```
bel(tutor, enviar_mensagem(3) if  
    bel(tutor, energia(plantaA, Pa)),  
    Pa < 70,  
    bel(tutor, tempo_folga_3(Tx3)).
```

```
bel(tutor, enviar_mensagem(3)) if  
    bel(tutor, energia(plantaB, Pb)),  
    Pb < 70,  
    bel(tutor, tempo_folga_3(Tx3)).
```

```
bel(tutor, enviar_mensagem(4)) if  
    bel(tutor, energia(micro, M)),  
    M < 70,  
    bel(tutor, tempo_folga_3(Tx3)).
```

```
bel(tutor, enviar_mensagem(5)) if  
    bel(tutor, energia(fundo, F)),  
    F < 80,  
    bel(tutor, tempo_folga_3(Tx3)).
```

```
bel(tutor, enviar_mensagem(6)) if  
    bel(tutor, energia(agua, A)),  
    A < 80,  
    bel(tutor, tempo_folga_3(Tx3)).
```

```
bel(tutor, enviar_mensagem(7)) if  
    bel(tutor, e_um_personagem(prefeito)),
```

```

bel(tutor, numero_ferramenta(lixeiras, N)),
N <= 2,
bel(tutor, tempo_folga_2(Tx2)).

```

```

bel(tutor, enviar_mensagem(7)) if
bel(tutor, e_um_personagem(prefeito)),
bel(tutor, numero_ferramenta(proibicao, N)),
N <=2,
bel(tutor, tempo_folga_2(Tx2)).

```

```

bel(tutor, enviar_mensagem(7)) if
    bel(tutor, e_um_personagem(prefeito)),
    bel(tutor, numero_ferramenta(multa, N)),
    N <= 1,
bel(tutor, tempo_folga_2(Tx2)).

```

```

bel(tutor, enviar_mensagem(7)) if
    bel(tutor, e_um_personagem(prefeito)),
    bel(tutor, numero_ferramenta(licenca, N)),
    N<= 1,
bel(tutor, tempo_folga_2(Tx2)).

```

```

bel(tutor, enviar_mensagem(7)) if
    bel(tutor, e_um_personagem(maenatureza)),
    bel(tutor, numero_ferramenta(accelera, N)),
    N <= 1,
bel(tutor, tempo_folga_2(Tx2)).

```

```

bel(tutor, enviar_mensagem(7)) if
    bel(tutor, e_um_personagem(maenatureza)),
    bel(tutor, numero_ferramenta(intensifica, N)),
    N <= 1,
bel(tutor, tempo_folga_2(Tx2)).

```

```

bel(tutor, enviar_mensagem(7)) if
    bel(tutor, e_um_personagem(maenatureza)),
    bel(tutor, numero_ferramenta(trocafuno, N)),
    N <= 1,
bel(tutor, tempo_folga_2(Tx2)).

```

```

bel(tutor, enviar_mensagem(7)) if
    bel(tutor, e_um_personagem(maenatureza)),
    bel(tutor, numero_ferramenta(adicionaplanta, N)),
    N <= 2,
bel(tutor, tempo_folga_2(Tx2)).

/* comportamento do tutor com aluno sem acao */

bel(tutor, aluno_sem_acao) if
    bel(tutor, aluno_e_um_personagem(Aluno, P1)),
    P1= "prefeito",
    before(bel(tutor, numero_ferramenta(lixeira, Nan))),
    bel(tutor, numero_ferramenta(lixeira, Nat)),
    Nan=Nat,
    before(bel(tutor, numero_ferramenta(proibicao, Nan))),
    bel(tutor, numero_ferramenta(proibicao, Nat)),
    Nan=Nat,
    before(bel(tutor, numero_ferramenta(multa, Nan))),
    bel(tutor, numero_ferramenta(multa, Nat)),
    Nan=Nat,
    before(    bel(tutor, numero_ferramenta(licenca, Nan))),
    bel(tutor, numero_ferramenta(licenca, Nat)),
    Nan=Nat.

/* envio de exemplos */

des(tutor, enviar_exemplo(X)).

bel(tutor, enviar_exemplo(1)) if
    des(Aluno, receber_auxilio),
    bel(tutor, energia(ecometro, Ee)),
    Ee >=50,
    Ee < 70,
    bel(tutor, aluno_sem_acao).

bel(tutor, enviar_exemplo(2)) if
    des(Aluno, receber_auxilio),
    bel(tutor, energia(ecometro, Ee)),
    Ee < 50,
    Ee >= 40,

```

```
bel(tutor, aluno_sem_acao).

bel(tutor, enviar_exemplo(3)) if
    des(Aluno, receber_auxilio),
    bel(tutor, energia(ecometro, Ee)),
    Ee < 40,
    Ee >= 35,
    bel(tutor, aluno_sem_acao).

bel(tutor, enviar_exemplo(4)) if
    des(Aluno, receber_auxilio),
    bel(tutor, energia(ecometro, Ee)),
    Ee < 35,
    Ee >= 30,
    bel(tutor, aluno_sem_acao).

bel(tutor, enviar_exemplo(5)) if
    des(Aluno, receber_auxilio),
    bel(tutor, energia(ecometro, Ee)),
    Ee < 30,
    Ee >= 25,
    bel(tutor, aluno_sem_acao).

bel(tutor, enviar_exemplo(6)) if
    des(Aluno, receber_auxilio),
    bel(tutor, energia(ecometro, Ee)),
    Ee < 25,
    bel(tutor, aluno_sem_acao).

/* envio de alerta */

des(tutor,enviar_alerta(X)).

bel(tutor, enviar_alerta (1)) if
    bel(tutor, energia(ecometro, Ee)),
    Ee < 20,
    bel(tutor, aluno_sem_acao).

bel(tutor, menor(Elemento, 20)) if
    bel(tutor, energia(Elemento, Nivel)),
```


Nivel < 20.

```
bel(tutor, energia_cenario_menor(Z)) if
  bel(tutor, menor(peixeA, Za)),
  bel(tutor, menor(peixeB, Zb)),
  bel(tutor, menor(peixeC, Zc)),
  bel(tutor, menor(plantaA, Zd)),
  bel(tutor, menor(plantaB, Ze)),
  bel(tutor, menor(micro, Zf)),
  bel(tutor, menor(fundo, Zg)),
  bel(tutor, menor(agua, Zh)),
  Z is Za + Zb + Zc + Zd + Ze + Zf + Zg + Zh.
```

```
bel(tutor, enviar_alerta(2)) if
  bel(tutor, energia_cenario_menor(Y),
  Y >2,
bel(tutor, tempo_folga_1(Tx1)).
```

```
bel(tutor, enviar_alerta(3)) if
  bel(tutor, elem_cenario_sumiu) if
  bel(tutor, peixe(Peixe)),
  bel(tutor, peixe(peixeA),
  bel(tutor, energia(peixeA, PA),
  PA= 0.
```

```
bel(tutor, enviar_alerta(3)) if
  bel(tutor, elem_cenario_sumiu) if
  bel(tutor, peixe(Peixe)),
  bel(tutor, peixe(peixeB),
  bel(tutor, energia(peixeB, PB),
  PB= 0.
```

```
bel(tutor, enviar_alerta(3)) if
  bel(tutor, elem_cenario_sumiu) if
  bel(tutor, peixe(Peixe)),
  bel(tutor, peixe(peixeC),
  bel(tutor, energia(peixeC, PC),
  PC= 0.
```

```
bel(tutor, enviar_alerta(3)) if
  bel(tutor, elem_cenario_sumiu) if
    bel(tutor, planta(Planta)),
    bel(tutor, planta(plantaA),
    bel(tutor, energia(plantaA, Pa),
    Pa= 0.
```

```
bel(tutor, enviar_alerta(3)) if
  bel(tutor, elem_cenario_sumiu) if
    bel(tutor, planta(Planta)),
    bel(tutor, planta(plantaB),
    bel(tutor, energia(plantaB, Pb),
    Pb= 0.
```

```
bel(tutor, enviar_alerta(3)) if
  bel(tutor, elem_cenario_sumiu) if
    bel(tutor, energia(micro, M),
    M= 0.
```

```
bel(tutor, enviar_alerta(3)) if
  bel(tutor, elem_cenario_sumiu) if
    bel(tutor, energia(fundo, F),
    F= 0.
```

```
bel(tutor, enviar_alerta(3)) if
  bel(tutor, elem_cenario_sumiu) if
    bel(tutor, energia(agua, A),
    A= 0.
```

Anexo 6 - Exemplo de arquivo do aluno

```
/* crencas do aluno sobre seu personagem prefeito */
```

```
bel(aluno, e_um_personagem(prefeito)).
```

```
bel(aluno, prefeito(lixadeiras)).
```

```
bel(aluno, prefeito(proibicao)).
```

```
bel(aluno, prefeito(multa)).
```

```
bel(aluno, prefeito(licenca)).
```

```
/* crencas sobre o ambiente */
```

```
bel(aluno, e_um_elementocenario(Peixe)) if
```

```
bel(aluno, peixe(Peixe)).
```

```
bel(aluno, e_um_elementocenario(Planta)) if
```

```
bel(aluno, planta(Planta)).
```

```
bel(aluno, e_um_elementocenario(micro)).
```

```
bel(aluno, e_um_elementocenario(fundo)).
```

```
bel(aluno, e_um_elementocenario(agua)).
```

```
bel(aluno, e_um_elementocenario(ecometro)).
```

```
bel(aluno, reproduz(Peixe)) if
```

```
bel(aluno, peixe(Peixe)).
```

```
bel(aluno, alimenta(Peixe)) if
```

```
bel(aluno, peixe(Peixe)).
```

```
bel(aluno, movimenta(Peixe)) if
```

```
bel(aluno, peixe(Peixe)).
```

```
bel(aluno, reproduz(Planta)) if
```

```
bel(aluno, planta(Planta)).
```

```
bel(aluno, alimenta(Planta)) if
```

```
bel(aluno, planta(Planta)).
```

```
bel(aluno, reproduz(micro)).
```

```
bel(aluno, movimenta(micro)).
```

```
bel(aluno, alimenta(fundoLago)).
```

```
bel(aluno, alimenta(agua)).
```

```
bel(aluno, peixe(peixeA)).
bel(aluno, peixe(peixeB)).
bel(aluno, peixe(peixeC)).

bel(aluno, planta(plantaA)).
bel(aluno, planta(plantaB)).

/* crencas sobre poluentes */

bel(aluno, retira_energia(draga)).
bel(aluno, retira_energia(lixourbano)).
bel(aluno, retira_energia(jetski)).
bel(aluno, retira_energia(pescapredatoria)).
bel(aluno, retira_energia (esgotoindustrial)).
bel(aluno, retira_energia (esgotodomestico)).
bel(aluno, retira_energia (produtostoxicos)).
bel(aluno, retira_energia (postogasolina)).

/* crencas sobre ferramentas */

bel(aluno, coloca_energia(lixadeiras)).
bel(aluno, coloca_energia(proibicao)).
bel(aluno, coloca_energia(multa)).
bel(aluno, coloca_energia(licenca)).
bel(aluno, coloca_energia(accelera)).
bel(aluno, coloca_energia(intensifica)).
bel(aluno, coloca_energia(trocafuno)).
bel(aluno, coloca_energia(adicionaplanta)).
bel(aluno, coloca_energia(pescapermitida)).
bel(aluno, coloca_energia(equipamento)).
bel(aluno, coloca_energia(retiralixo)).
bel(aluno, coloca_energia(devolvepeixe)).

bel(aluno, combate(proibicao, draga)).
bel(aluno, combate(multa, draga)).
bel(aluno, combate(licenca, draga)).

bel(aluno, combate(lixadeiras,lixourbano)).
```

```

bel(aluno, combate(proibicao, jetski)).
bel(aluno, combate(multa, jetski)).
bel(aluno, combate(licenca, jetski)).

bel(aluno, combate(proibicao, pescapredatoria)).
bel(aluno, combate(multa, pescapredatoria)).
bel(aluno, combate(licenca, pescapredatoria)).

bel(aluno, combate(lixeiras, esgotodomestico)).

bel(aluno, combate(proibicao, esgotoindustrial)).
bel(aluno, combate(multa, esgotoindustrial)).
bel(aluno, combate(licenca, esgotoindustrial)).
bel(aluno, combate(retiralixo, esgotoindustrial)).

bel(aluno, combate(proibicao, produtostoxicos)).
bel(aluno, combate(multa, produtostoxicos)).

bel(aluno, combate(proibicao, postogasolina)).
bel(aluno, combate(multa, postogasolina)).
bel(aluno, combate(licenca, postogasolina)).

/* calculo da energia do ecometro */

bel(aluno, energia(ecometro,Ee)) if
bel(aluno, energia(peixeA, PA)),
bel(aluno, energia(peixeB, PB)),
bel(aluno, energia(peixeC, PC)),
bel(aluno, energia(plantaA, Pa)),
bel(aluno, energia(plantaB, Pb)),
bel(aluno, energia(micro,M)),
bel(aluno, energia(fundo,F)),
bel(aluno, energia(agua, A)),
Ee= (PA+PB+PC+Pa+Pb+M+F+A)/8.

/* ciclo de vida dos elementos do cenario */

bel(aluno, ~reproduz(peixeA)) if
    bel(aluno, energia(peixeA, PA)),
    PA < 80.

```

```
bel(aluno, ~movimenta(peixeA)) if
    bel(aluno, energia(peixeA, PA)),
    PA < 60.
bel(aluno, ~alimenta(peixeA)) if
    bel(aluno, energia(peixeA, PA)),
    PA < 40.
bel(aluno, morre(peixeA)) if
    bel(aluno, energia(peixeA, PA)),
    PA < 10.
bel(aluno, ~reproduz(peixeB)) if
    bel(aluno, energia(peixeB, PB)),
    PB < 80.
bel(aluno, ~movimenta(peixeB)) if
    bel(aluno, energia(peixeB, PB)),
    PB < 60.
bel(aluno, ~alimenta(peixeB)) if
    bel(aluno, energia(peixeB, PB)),
    PB < 40.
bel(aluno, morre(peixeB)) if
    bel(aluno, energia(peixeB, PB)),
    PB < 10,

bel(aluno, ~reproduz(peixeC)) if
    bel(aluno, energia(peixeC, PC)),
    PC < 80.
bel(aluno, ~movimenta(peixeC)) if
    bel(aluno, energia(peixeC, PC)),
    PC < 60.
bel(aluno, ~alimenta(peixeC)) if
    bel(aluno, energia(peixeC, PC)),
    PC < 40.
bel(aluno, morre(peixeC)) if
    bel(aluno, energia(peixeC, PC)),
    PC < 10.

bel(aluno, ~reproduz(plantaA)) if
    bel(aluno, energia(plantaA, Pa)),
    Pa < 70.
bel(aluno, ~movimenta(plantaA)) if
    bel(aluno, energia(plantaA, Pa)),
```

```

Pa < 50.
bel(aluno, morre(plantaA)) if
  bel(aluno, energia(plantaA, Pa)),
  Pa < 10.

bel(aluno, ~reproduz(plantaB)) if
  bel(aluno, energia(plantaB, Pb)),
  Pb < 70.
bel(aluno, ~movimenta(plantaB)) if
  bel(aluno, energia(plantaB, Pb)),
  Pb < 50.
bel(aluno, morre(plantaB)) if
  bel(aluno, energia(plantaB, Pb)),
  Pb < 10.

bel(aluno, ~reproduz(micro)) if
  bel(aluno, energia(micro, M)),
  M < 70.
bel(aluno, ~movimenta(micro)) if
  bel(aluno, energia(micro, M)),
  M < 50.
bel(aluno, morre(micro)) if
  bel(aluno, energia(micro, M)),
  M < 40.

des(aluno, observar_cenario) if
  bel(aluno, energia_ecometro(Ee),
  Ee>70.

des(aluno, verificar_ecom) if
  bel(aluno, energia(ecometro, Ee)),
  Ee<100.

des(aluno, receber_auxilio)if
  bel(aluno, energia(ecometro, Ee)),
  Ee < 50.

des(aluno, parar_jogar)if
  bel(aluno, zerada(F, 0)) if
  bel(aluno, numero_ferramenta (F, No)),

```

No= 0.

```
bel(aluno, numero_ferramenta_zerada_prefeito(No)) if
  bel(aluno, zerada(lixeira, L)),
  bel(aluno, zerada(proibicao, P)),
  bel(aluno, zerada(multa, Mu)),
  bel(aluno, zerada(licenca, Li)),
  No is L+ P + Mu + Li.
```

/* suspender partida para discutir estrategia com colega ou
consultar a ajuda*/

```
des(aluno, suspender_partida)if
  bel(aluno,energia_ecometro(Ee),
  Ee <30,
  bel(aluno, numero_ferramenta(lixeira, L)),
  L=0.
```

```
des(aluno, suspender_partida)if
  bel(aluno,energia_ecometro(Ee),
  Ee <30,
  bel(aluno, numero_ferramenta(proibicao, P)),
  P=0.
```

```
des(aluno, suspender_partida)if
  bel(aluno,energia_ecometro(Ee),
  Ee <30,
  bel(aluno, numero_ferramenta(multa, Mu)),
  Mu=0.
```

```
des(aluno, suspender_partida)if
  bel(aluno,energia_ecometro(Ee),
  Ee <30,
  bel(aluno, numero_ferramenta(licenca, Li)),
  Li=0.
```

```
des(aluno, não_jogar)if
  bel(aluno, nao_tem_ferramenta_adequada).
```

```
des(aluno, aumentar_energia) if
```



```
    bel(aluno, prefeito(lixeiras)).
des(aluno, aumentar_energia) if
    bel(aluno, prefeito(proibicao)).

des(aluno, aumentar_energia) if
    bel(aluno, prefeito(multa)).

des(aluno, aumentar_energia) if
    bel(aluno, prefeito(licenca)).

/*crencas sobre os elementos do cenario e a energia */
bel(aluno, energia_ecometro_baixa) if
bel(tutor, reproduz(Peixe)).

bel(aluno, energia_ecometro_baixa) if
bel(tutor, reproduz(Planta)).

bel(aluno, energia_ecometro_baixa) if
bel(tutor, reproduz(Micro)).
```

Anexo 7 - Publicações relacionadas com a tese

- Relatórios técnicos:

Giraffa, Lucia M. M. **Fundamentos de teorias de Ensino-Aprendizagem e sua aplicação em Sistemas Tutores Inteligentes**. Porto Alegre: CPGCC/UFRGS, 1995. (Trabalho Individual N° 487)

Este trabalho apresenta uma revisão das teorias de ensino-aprendizagem sob o ponto de vista psicológico e pedagógico, discute a questão da aprendizagem sob o ponto de vista da IA e analisa sua utilização em STI.

Giraffa, L.M.M. **Seleção e adoção de Estratégias de Ensino em Sistemas Tutores Inteligentes**. Porto Alegre: CPGCC/UFRGS, 1997. (EQ-10)

Este trabalho fez uma ampla revisão bibliográfica dos STI desde sua criação até os projetos publicados no início do ano de 1997. São aprofundados os aspectos referentes às estratégias e táticas de ensino utilizadas nos STI da literatura. Na sua conclusão apresentamos a modelagem inicial do protótipo.

Giraffa, L.M.M. **Selecting teaching strategies using pedagogical agents**. Porto Alegre: CPGCC/UFRGS, 1998. (Thesis proposal)

Apresenta em detalhes a proposta do trabalho de tese e o estado do protótipo na época. Apresentada em 23/3/1998.

- Comunicações e posters:

Giraffa, L.M.M.; Nunes, M.A; Viccari, R.M. Multi-Ecológico: proposta de Ambiente de Ensino Inteligente utilizando arquitetura de Sistemas Multiagentes. I ENIA: Encontro Nacional de Inteligência Artificial. **Anais...**Brasília: SBC, 1997.

Este trabalho apresentou a idéia geral da sociedade de agentes reativos e cognitivos da arquitetura geral do STI.

Giraffa, L.M.M; Viccari, R.M.; Self, J. Improving tutoring activities using Multi-Agents system Architecture. Twenty-ninth SIGCSE Technical Symposium on Computer Science Education, 29,1998. **Proceedings...** Atlanta: Georgia, 1998. (PhD Students Contest)

Este trabalho foi enviado para o concurso de teses em andamento da ACM, onde foi classificado entre os dez finalistas. No resultado final recebeu o quarto lugar. O texto mostrava a proposta da arquitetura geral do sistema e a arquitetura do tutor modelado na abordagem mentalística.

Giraffa, L.M.M; Viccari, R.M.; Self, J. Multi-Agent based pedagogical games. ITS'98- Fourth International Conference on Intelligent Tutoring Systems. **Proceedings...** San Antonio, Texas, 1998.

Este poster apresenta a arquitetura geral do tutor e foi apresentado durante o ITS'98.

Giraffa, L.M.M; Móra, M.; Viccari, R.M. Towards a New a Computational Model to build a Tutor. IX International Conference on Artificial Intelligence in Education – AIED'99. **Proceedings ...** France, Le Mans, July 1999.

Este artigo apresenta a descrição geral de todo o trabalho de tese, a descrição da implementação do kernel cognitivo, os avanços obtidos no ambiente de Móra et al. em função do trabalho conjunto desenvolvido nas duas teses de doutorado com a supervisão da mesma orientadora.

- Artigos em congresso nacionais e internacionais:

Raabe, A.L.A; Javimczik, A .M.; Giraffa, L.M.M. Eco-Lógico: Ambiente interativo para suporte ao ensino de educação ambiental. In: VII Simpósio Brasileiro de Informática na Educação, 7., 1996. **Anais** Belo Horizonte: DCC/UFMG, 1996.

O artigo descreve em detalhes a modelagem do ambiente Eco-Lógico

Giraffa, L.M.M.; Nunes, M.A.; Viccari, R.M. Multi-Ecological: an Intelligent Learning Environment using Multi-Agent architecture. In: MASTA'97: Multi-Agent System: Theory and Applications, 1., 1997. **Proceedings...** Coimbra: DE - Universidade de Coimbra, 1997.

Este artigo descreve em detalhes o ambiente multiagente proposto a partir do Eco-Lógico que serviu de protótipo para testar a arquitetura multiagente proposta para o STI. Apresenta em detalhes as idéias da comunicação enviada para o ENIA. Foram enfatizados os aspectos de IA

Giraffa, L.M.M.; Nunes, M.A; Viccari, R.M. Multi-Ecológico: um Ambiente de Ensino Inteligente para suporte ao ensino de Educação Ambiental. In: VIII SBIE: Simpósio Brasileiro de Informática na Educação, 8., 1997. **Anais...** São Carlos: ITA, 1997

Este artigo descreve em detalhes o ambiente multiagente proposto a partir do Eco-Lógico que serviu de protótipo para testar a arquitetura multiagente proposta para o STI, só que explorando os aspectos pedagógicos do sistema.

Giraffa, L.M.M; Viccari, R.M. Tutor behaviour in multi-agent ITS guided through mental states activities. Workshop: Pedagogical Agents. In: ITS'98- Fourth International Conference on Intelligent Tutoring Systems. Proceedings... San Antonio, Texas, 1998.

Este artigo descreve em detalhes a arquitetura do tutor e como é tratado o modelo do aluno.

Devido às suas contribuições no aspecto pedagógico ele foi selecionado para o Workshop sobre estratégias de ensino, que foi realizado simultaneamente ao Workshop de agentes.

Giraffa, L.M.M; Móra, M.; Viccari, R.M Modelling the MCOE Tutor using Computational Model. In: **Lectures Notes on Artificial Intelligence - SBIA'98**. Oliveira, F. (Ed.). Berlin: Springer Verlag, 1998.

Este artigo descreve em detalhes o kernel cognitivo. Apresenta a arquitetura do tutor e a coreografia dos estados mentais utilizando o modelo computacional dos agentes utilizando o formalismo de Móra et al.

Giraffa, L.M.M; Móra, M.; Viccari, R.M. Pedagogical game using ITS architecture. Workshop Multi-Agents Architectures -IBERAMIA'98. **Proceedings...**Toledo: Spain, 1998.

Este artigo descreve a forma como a arquitetura do sistema e do tutor foram utilizadas no protótipo. São exploradas os aspectos de construção de sistema multiagentes.

Giraffa, L.M.M.; Viccari, R.M. The Use of Agents Techniques on Intelligent Tutoring Systems. 18th International Conference of the Chilean Computer Science Society (SCCC'98), 18, 1998. Proceedings... Antofagasta, Chile: IEEE, 1998.

Este artigo apresenta a revisão bibliográfica realizada sobre agentes pedagógicos e a proposta por Viccari e Giraffa para uma taxinomia de agentes pedagógicos, indicando as propriedades básicas de um agente pedagógico.

Giraffa, L.M.M.; Viccari, R.M. Estratégias de Ensino em Sistemas Tutores Inteligentes modelados através da tecnologia de agentes. IX SBIE'98 –Simpósio Brasileiro de Informática na Educação. **Anais...**Fortaleza: SBC/IE, 1998.

Este artigo apresenta os aspectos pedagógicos da arquitetura do tutor. Explora a forma como foram associadas as táticas às estratégias de ensino o comportamento do tutor.

Este artigo recebeu a premiação de melhor artigo do SBIE'98 e será publicado na Revista de IE da SBC no ano de 1999.

Giraffa, L.M.M; Móra, M.; Viccari, R.M. Modelling an interactive ITS using a MAS approach: from design to pedagogical evaluation. 3rd International Conference on

Computational Intelligence and Multimedia Applications - ICCIMA'99. **Proceedings...** India, New Delhi, September 1999. (Submitted paper)

Este artigo apresenta os resultados obtidos com o experimento realizado na tese e algumas considerações sobre o projeto de STI utilizando agentes pedagógicos.

Tutorial:

VICCARI, R.M.;GIRAFFA,L.M.M. Sistemas Tutores Inteligentes: abordagem tradicional x abordagem de agentes Brazilian Symposium on Artificial Intelligence, 13., 1996. **Proceedings...** Curitiba: Springer Verlag, 1996.

Este trabalho foi desenvolvido em paralelo com o exame de qualificação. Reflete a parte de abrangência do exame, onde foi realizada a revisão geral sobre STI. Apresentado num minicurso durante o SBIA'96.

Artigo em revista:

•

Giraffa, L.M.M.; Móra, M.; Viccari, R. M. Towards a New Perspective for ITS: Using a Mentalistic Approach to Model Cognitive Pedagogical Agents. Special Issue on Intelligent Systems/Tools in Training and Life-Long Learning of the **International Journal of Continuing Engineering Education and Life-Long Learning**, Osaka, Japão, 1999.

- Este artigo foi selecionado para compor a edição especial da revista que tratou apenas de sistemas que tragam abordagens novas e visões diferenciadas de sistemas educacionais

Bibliografia

- [AIM96] AIMUER, E.; FRASSON, C. Analysing a new learning strategy according to different knowledge levels. **Computers Education**, London, v.27,n.2,1996.
- [AKH96] AKHRAS, F; SELF, J. From the Process of Instruction to the Process of Learning: Constructivist Implications for the Design of Intelligent Learning Environment. In: AIED: EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 1., 1996. **Proceedings...** Lisbon: Calouste Gulbenkian, 1996.
- [AKH97] AKHRAS, F; SELF, J. Modelling Learning as a Process. In: AIED: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 8.,1997. **Proceedings...** Kobe, Japan: [s.n.],1997.
- [ALF96] ALFERES, J.J.; PEREIRA, L.M. **Reasoning with Logic Programming**. Berlin: Springer-Verlag, 1996. Lecture Notes In Artificial Intelligence Series
- [ALV96] ALVES,I.;KOLBERG,L.;GIRAFFA,L.M.M.HEI:assistente inteligente para o suporte ao ensino de Lingua Inglesa. In: SBIE: SIMPOSIO BRASILEIRO DE INFORMATICA NA EDUCACAO,8.,1997, São José dos Campos, SP.**Anais...**São José dos Campos: ITA,1997.
- [ALV97] ALVARES,L.O. SICHMAM, J.S. Introdução aos Sistemas Multiagentes. In: JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA, 16., 1997, Brasília, DF. **Anais ...** Brasília: UnB, 1997.
- [AND97] ANDRE, E.; MULLER, J.; RIST, T. Life-Like Presentations Agents: A New perspective for Computer Based Technical Documentation. In: AIED: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 8.,1997. **Proceedings...** Kobe, Japan: [s.n.],1997.
- [ASA91] ASANOME, C. **Sistemas Tutoriais Inteligentes: um estudo**. Rio de Janeiro: COPPE/UFRJ, 1991.
- [AUS80] AUSUBELD.P; NOVAK,J.D; HANESIAN,H. **Psicologia Educacional**. Rio de Janeiro:Interamericana,1980.
- [BAL97] BALACHEFF N; KAPUT J. Computer-Based Learning Environments In Mathematics. In: BISHOP, A. et al. (Eds.) **International Handbook In Mathematics Education** Dordrecht : Kluwer Academic, 1997.
- [BAR83] BARWISE, J.; PERRY, J. **Situations and Attitudes**. Cambridge: MIT Press, 1983.
- [BEC98] BECK, J.E.; WOOLF, P. Using a Learning Agent with a Student Model. In: INTELLIGENT TUTORING SYSTEMS-ITS'98, 4., 1998, San Antonio. **Proceedings...** Berlin: Springer-Verlag, 1998. (Lecture Notes In Computer Science, v.1452)
- [BER99] BERCHT,M; VICCARI, R.M; CECCIM, R. Afetividade em Máquina: uma possibilidade? Non-sense complete? Pós-modernidade em IA?. In:

- WORKSHOP EM INFORMÁTICA NA EDUCAÇÃO, 1., 1999, Porto Alegre, RS. **Anais ...** Porto Alegre: CPGIE/UFRGS, 1999.
- [BLO83] BLOOM, B.S. **Taxionomia de objetivos educacionais.** Porto Alegre: Globo, 1983.
- [BON88] BOND, A. H ; GASSER, L.. An Analysis of Problems and Research in DAI. In: **Distributed Artificial Intelligence.** San Mateo: Morgan Kaufmann, 1988.
- [BOY92] BOYD, G. MITCHELL, P.D. How can Intelligent CAL better adapt to learners? **Computers Education**, London, v.18, n.1-3, p.23-28, 1992.
- [BRA84] BRATMAN, M. Two Faces of Intention. **The Philosophical Review**, [S.l.], v.93, n.3, p.275-405, 1984.
- [BRA87] BRATMAN, M.; ISRAEL, D.; POLLACK, M. **Toward an architecture for resource-bounded agents.** [S.l.], Stanford University, 1987.
- [BRA89] BRATMAN, M. What is Intention. In: COHEN, P; MORGAN, J; POLLACK, M. (Eds.). **Intentions In Communication** [S.l.]: MIT Press, 1989.
- [BRO78] BROW, J.S.; COLLINS, A.; HARRIS, G. Artificial Intelligence and Learning Strategies In: O'NEIL, H.(Ed.). **Learning Strategies.** New York: Academic Press, 1978.
- [BRO82] BROW, J.S.; BURTON, R.B. Pedagogical natural language and knowledge engineering techniques in SOPHIE I, II, and III. In: SLEEMAN, D.; BROWN, J.S (Eds.). **Intelligent Tutoring Systems.** New York: Academic Press, 1982.
- [BUR79] BURTON, R.R.; BROWN, J.S. An investigation of computer coaching for informal learning activities. In: SLEEMAN, D.H.; BROWN, J.S (Eds.). **Intelligent Tutoring Systems.** London: Academic Press, 1979.
- [CAL99] CALLEGARI, D.C. **Uma proposta de extensão ao ambiente MCOE utilizando aprendizagem por reforço.** Porto Alegre: FACIN/PUCRS, 1999.
- [CAR70] CARBONELL, J.R. AI and CAI: an artificial intelligence approach to computer-assited instruction. **IEEE Trans, on Man-Machine System**, New York, v.11, n.2, p.190-202, 1970.
- [CAS95] CASTELFRANCHI, C. Guarantees for autonomy in cognitive agent architecture. In: WOOLDRIDGE, M.; JENNINGS, N.R. (Eds.). **Intelligent Agents: Theories, Architectures and Languages.** Heilderberg: Springer-Verlag, 1995.
- [CAS99] CASAS, L.A.A. **Contribuições para a modelagem de um ambiente inteligente de educação baseado em realidade virtual.** Florianópolis: PGPP/UFSC, 1999.
- [CHA91] CHAVES, E. **Multimídia: conceituação, aplicações e tecnologia.** Campinas: People Computação, 1991.
- [CLA82] CLANCEY, W.J. Tutoring rules for guiding a case method dialogue. In: SLEEMAN, D.; BROWN, J.S (Eds.). **Intelligent Tutoring Systems.** Orlando: Academic Press, 1982.

- [CLA89] CLANCEY, W.J. Situated Cognition and Intelligent Tutoring Systems. In: MAUER, H (Ed.). INTERNATIONAL CONFERENCE ON COMPUTER-ASSISTED LEARNING,2.,1989, Dallas. **Proceedings...** Berlin: Springer-Verlag, 1989.
- [CLA90] CLANCEY, W.J ; SOLOWAY, E. Artificial Intelligence and Learning Environments. **Artificial Intelligence**, Amsterdam, v.42, n.1,p.1-6, 1990.
- [COH87] COHEN, P. R., LEVESQUE, H. J. Intention = Choice + Commitment. In: NATIONAL CONFERENCE ON AI, 6.,1987. **Proceedings...** Berlin: Springer-Verlag, 1987.
- [COH89] COHEN, P. et al. Trial by Fire: Understanding the Design Requirements for Agents In Complex Environment. **AI Magazine**, [S.l.], Fall, 1989.
- [COL97] COLAZZO, L.; SILVESTRI, L. The pragmatics of the Tutor: A proposal of modelling. In:: AIED: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 8.,1997. **Proceedings...** Kobe, Japan: [s.n.],1997.
- [CON92] CONTE,R; CASTELFRANCHI, C. Mins is not enough: precognitive bases of social interactions. In: SYMPOSIUM ON SIMULATING SOCIETIES, 1., 1992. **Proceedings....** UK: Guilford, 1992.
- [COO97] COOK J. An Empirically Based mentoring Agent for Supporting Music Composition Learning. In: AIED: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 8.,1997. **Proceedings...** Kobe, Japan: [s.n.], 1997.
- [COR94] CORRÊA,M. **A Arquitetura de Diálogos entre Agentes Cognitivos Distribuídos**. Rio de Janeiro: UFRJ, 1994.
- [COR95] CORRÊA, M.; MENDES, S. A Computational Approach to Situation Theory Based on Logic Programming to design Cognitive Agents. In: BRASILIAN SYMPOSIUM ON ARTIFICIAL INTELLIGENCE, 12., 1995, Campinas. **Proceedings...** Campinas: Springer-Verlag, 1995.
- [COR98] CORRÊA, M.; VICCARI, R.M.; COELHO, H. Dynamics In Transition Mental Activity. In: ICMAS –INTERNATIONAL CONFERENCE ON MULTI-AGENT SYSTEMS, 1., 1998, Paris. **Proceedings...** Paris: ICMAS,1998.
- [COR98a] CORBETT, A.T. et al. A Formative Evaluation of the PACT Algebra II Tutor: Support for Simple Hierarchical Reasoning. In: INTELLIGENT TUTORING SYSTEMS-ITS'98, 4., 1998, San Antonio. **Proceedings...** Berlin: Springer-Verlag, 1998. (Lecture Notes In Computer Science, v.1452)
- [COS92] COSTA, E. GASPAR, G. COELHO, H. A formal approach to ILES: a first tentative. ADVANCED WORKSHOP ON STUDENT MODELLING, 1992. **Proceedings...**Montreal: Springer-Verlag, 1992.
- [COS96] COSTA, E. B. **Um modelo de Ambiente Interativo de Ensino-Aprendizagem baseado numa Arquitetura Multi-Agentes**. Campina Grande: CPGEE/UFPA, 1996.
- [COS97] COSTA, E.B.; PERKUSICH, A. A Multi-Agent Interactive Learning Environment Model. In: AIED: WORLD CONFERENCE ON

- ARTIFICIAL INTELLIGENCE IN EDUCATION, 8.,1997. **Proceedings...** Kobe, Japan: [s.n.], 1997.
- [DAM95] D'AMICO O,C. **Modelo de usuário para Sistemas Tutores Inteligentes.** Porto Alegre: CPGCC/UFRGS, 1995.
- [DAM99] D'AMICO, C.B. **Aprendizagem estática e dinâmica em sistemas multi-agentes de ensino-aprendizagem** PortoAlegre:PGCC/UFRGS:1999.
- [DAV80] DAVIDSON, D. **Essays on actions and events.** New York: Oxford University Press, New York, NY, 1980.
- [DEM90] DEMAZEAU, Y. ; MÜLLER, J. Decentralised Artificial Intelligence. In: EUROPEAN WORKSHOP ON MODELLING AUTONOMOUS AGENTES IN A MULTI-AGENT WORD,1.,1990. **Proceedings...** Cambridge: North-Holland ,1990.
- [DEV91] DEVLIN, K. **Logic and Information.** Cambridge: Cambridge Press, 1991.
- [DIL95] DILLENBOURG, P. From Mutual Diagnosis to Collaboration Engines: Some technical Aspects of Distributed Cognition. In: AIED: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 7.,1995. **Proceedings...** [S.l.; s.n.], 1995.
- [DIL96] DILLENBOURG, P. Some technical Implications of Distributed Cognition on the Design on Interactive learning Environments.**Journal of Artificial Intelligence In Education** London: Elsevier, v.7, n.2, 1996.
- [DIL97] DILLENBOURG, P; et al. The Design of MOO Agents: Implications from an Empirical CSCW Study. In: AIED: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 8.,1997. **Proceedings...** Kobe, Japan: [s.n.], 1997.
- [EDG95] EDGAR, R. PC is to Piaget as WWW is to Vygotsky. In: SIGGRAPH COMPUETR GRAPHICS CONFERENCE, 5., 1995. **Proceedings....** Los Angeles: SIGGRAPH, 1995.
- [ELL97] ELLIOTT, C. Affective Reasoner Personality Models for Automated Tutoring Systems. In: AIED: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 8.,1997. **Proceedings...** Kobe, Japan: [s.n.],1997.
- [EUS95] EUSÉBIO, A.M.J. **Ambiente de Aprendizagem em Matemática apoiado em Agentes Autônomos Inteligentes.** Lisboa: Universidade Técnica de Lisboa, 1995.
- [FER91] FERBER, J; JACOBIN, E. The Framework of Eco-Problme Solving. IN: Demazeau, Y; Muller, J.P. (Eds.). **Descentralized AI 2.** Amsterdam: North-Holland, 1991.
- [FIS90] FISCHETTI, E.; GISOLFI, A. From Computer Aided Instruction to Intelligent Tutoring System. **Educational Technology**, Cambridge, v.7,n. 17, 1990.
- [FRA97] FRASSON, C.; MENGELLE, T.; AIMEUR, E. Using Pedagogical Agents In a Multi-Strategic Intelligent Tutoring System. In: AIED: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 8.,1997. **Proceedings...** Kobe, Japan: [s.n.],1997.

- [FUC94] FUCHS, D. et al. The nature of student interactions during peer tutoring with and without peer training experience. **American Educational Research Journal**, [S.l.], v.31, n.1, p.75-103, 1994.
- [GAG96] GAGNÉ, D.; TRUDEL, A . A Highly Flexible Student Driven Architecture for Computer Based Instruction. In: INTERNATIONAL CONFERENCE ON INTELLIGENT TUTORING SYSTEMS-ITS, 3.,1996, Montreal. **Proceedings...Berlin: Springer-Verlag, 1996.**
- [GAL88] GALLIERS, J.R. **A Theoretical Framework for Computer Models of Co-operative Dialogue, Acknowledging Multi-Agent Conflict.** UK: Open University, 1988.
- [GEN77] GENTNER, D.R. The FLOW tutor: a schema-based tutorial system. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 5., 1977. **Proceedings...** Los Altos: Morgan Kaufmann, 1977.
- [GIR93] GIRAFFA, L.M.M. Abracadabra- Ambiente de ensino-aprendizagem computadorizado. SBIE - SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 4., 1993, Recife, PE. **Anais ...Recife: UFPE,1993.**
- [GIR96] GIRAFFA,L.M.M;OLIVEIRA, J.; BORGES,L. Túnel do Tempo - Ferramenta de auxílio ao ensino de História. In: CONGRESSO IBERO-AMERICANO DE INFORMÁTICA NA EDUCAÇÃO,3.,1996, Barranquilla. **Anais...** Barranquilla: RIBIE,1996.
- [GIR97] GIRAFFA, L.M.M.; NUNES,M.A; VICCARI,R.M. Multi-Ecological: proposta de Ambiente de Ensino Inteligente utilizando arquitetura de Sistemas Multiagentes. In: ENIA: ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL, 1. , 1997, Brasília. **Anais...** Brasilia:UnB,1997.
- [GIR97] GIRAFFA, L.M.M.; NUNES, M.A.; VICCARI, R.M. Multi-Ecological: an Intelligent Learning Environment using Multi-Agent architecture. In: MASTA:MULTI-AGENT SYSTEM: THEORY AND APPLICATIONS, 1.,1997, Coimbra. **Proceedings...** Coimbra: DE-Universidade de Coimbra, 1997.
- [GIR97] GIRAFFA, L.M.M.; NUNES, M.A; VICCARI, R.M. Multi-Ecologico: um Ambiente de Ensino Inteligente para suporte ao ensino de Educação Ambiental. In: SBIE: SIMPOSIO BRASILEIRO DE INFORMATICA NA EDUCACAO,8.,1997, São José dos Campos, SP. **Anais...**São José dos Campos: ITA,1997.
- [GIR98a] GIRAFFA, L.M.M. **Selecting teaching strategies using pedagogical agents.** Porto Alegre: CPGCC/UFRGS, 1998.
- [GIR98b] GIRAFFA, L.M.M; VICCARI, R.M.; SELF, J. Improving tutoring activities using a Multi-Agents system Architecture. In: SIGCSE TECHNICAL SYMPOSIUM ON COMPUTER SCIENCE EDUCATION, 29.,1998, Atlanta. **Proceedings...** Atlanta: Georgia, 1998.
- [GIR98c] GIRAFFA, L.M.M; VICCARI, R.M.; SELF, J. Multi-Agent based pedagogical games. In: ITS - INTERNATIONAL CONFERENCE ON

- INTELLIGENT TUTORING SYSTEMS, 4., 1998, San Antonio. **Proceedings...** Berlin: Springer-Verlag,1998
- [GIR98d] GIRAFFA, L.M.M; VICCARI, R.M. Tutor behaviour in a multi-agent ITS guided through mental states activities. Workshop: Pedagogical Agents. In: INTELLIGENT TUTORING SYSTEMS-ITS, 4., 1998, San Antonio. **Proceedings...** San Antonio: AIED, 1998.
- [GIR98e] GIRAFFA, L.M.M; MÓRA, M.; VICCARI, R.M. Modelling the MCOE Tutor using a Computational Model. In: SBIA – SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICIAL, 14., Porto Alegre, RS 1999. Berlin: Springer-Verlag, 1998.
- [GIR98f] GIRAFFA, L.M.M; MÓRA, M.; VICCARI, R.M. Pedagogical game using ITS architecture. In: WORKSHOP MULTI-AGENTS ARCHITECTURES -IBERAMIA, 1., 1998. **Proceedings...** Toledo: IBERAMIA, 1998.
- [GIR98g] GIRAFFA, L.M.M.; VICCARI, R.M. Estratégias de Ensino em Sistemas Tutores Inteligentes modelados através da tecnologia de agentes. In: SBIE: SIMPOSIO BRASILEIRO DE INFORMATICA NA EDUCACAO,19.,1998, Fortaleza, CE. **Anais...**Fortaleza: UFCE,1998.
- [GIR98h] GIRAFFA, L.M.M.; VICCARI, R.M. The Use of Agents Techniques on Intelligent Tutoring Systems. In: INTERNATIONAL CONFERENCE OF THE CHILEAN COMPUTER SCIENCE SOCIETY (SCCC), 18., 1998, Antofagasta **Proceedings...** New York: IEEE, 1998.
- [GOL82] GOLDSTEIN, I. The Genetic graph: a representation for the evolution of procedural knowledge, In: SLEEMAN,D. ; BROWN,J.S. (Eds.). **Intelligent Tutoring Systems..** Great Britain: Academic Press, 1982.
- [GRA94] GRAESSER, A. C; PERSON, N. K. Question asking during tutoring. **American Educational Research Journal**, [S.l.] , v.31, n.1, p.104-137, 1994.
- [HAL92] HALPERN, J.Y; MOSES, Y. A guide to completeness and complexity for modal logics of knowledge and belief. **Artificial Intelligence**, Amsterdam, v.54, p.319-379, 1992.
- [HAY97] HAYES-ROTH, B.; VAN GENT, R.; HUBER,D. Acting In: TRAPPL, R.; PETTA, P. (Eds.). Character. **Creating Personalities for Synthetic Actors:** towards autonomous personality agents. Berlin: Springer-Verlag, 1997.
- [HEN92] HENDLEY, R. JURASCHECK, N. Cascade: introducing AI into CBT. **Computers Education**, London, v.18, n.1-3, p.71-76, 1992.
- [HIE97] HIETALA, P.; NIEMIREPO,T. Collaboration with Software Agents: What if the Learning Companion Agent Make Errors? In: AIED: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 8.,1997. **Proceedings...** Kobe, Japan: [s.n.],1997.
- [HOP95] HOPPE, U. The Use of Multiple Student Modelling to parametrize group learning. In: GREER, J. (ed.). **Artificial Intelligence in Education**, [S.l.]:AACE, 1995.

- [IKE97] IKEDA, M; GO, S. ;MIZOGUCHI, R. Opportunistic Group Formation - a Theory for Intelligent Support In Collaborative Learning. In: AIED: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 8.,1997. **Proceedings...** Kobe, Japan: [s.n.], 1997.
- [JEN95] JENNINGS, N.R. Controlling co-operative problem solving In industrial multi-agent systems using joint intentions. **Artificial Intelligence**, London, Elsevier, v.74, n.2, 1995.
- [KAT96] KATZ, S. et al. Towards the Design of more Effective Advisors for learning-by-Doing Systems. In: INTERNATIONAL CONFERENCE ON INTELLIGENT TUTORING SYSTEMS-ITS, 3., 1996, Montreal. **Proceedings...** Berlin: Springer-Verlag, 1996.
- [KEA87] KEARSLEY, G. **Artificial Intelligence, and Instruction: applications and methods.** Massachusetts: Addison-Wesley, 1987.
- [KEL87] KELLER, A. **When machines teach: designing computer courseware.** New York: Harper&Row, 1987.
- [KNE88] KNELLER, G. **Arte e Ciência da Criatividade.** São Paulo: Ibrasa, 1988.
- [KHU96] KHUWAJA.R.; DESMARAIS,M.;CHENG,R. Intelligent Guide: Combining User Knowledge Assessment with pedagogical Guidance. In: INTERNATIONAL CONFERENCE ON INTELLIGENT TUTORING SYSTEMS-ITS, 3., 1996, Montreal. **Proceedings...** Berlin: Springer-Verlag, 1996.
- [LAU92] LAURILLARD, D. Learning through collaborative computer simulations. **British Journal of Educational Technology**, [S.l], v.23, p.164-171, 1992.
- [LEM96] LEMAN, S.; MARCENAC, P.; GIROUX, S.A. Generic Architecture for ITS Based on a Multi-Agent Approach. In: INTERNATIONAL CONFERENCE ON INTELLIGENT TUTORING SYSTEMS-ITS, 3., 1996, Montreal. **Proceedings...** Berlin: Springer-Verlag, 1996.
- [LER97] LEROUX ,P. Co-operation between a Pedagogical Assistant, a Group of Learners and a Teacher. In: EURO AIED - EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 1., 1996. **Proceedings...**Lisboa: Colobri,1996.
- [LES96] LESTER, J.C. et al. Focusing Problem Solving In Design-Centred Learning Environments. In: INTERNATIONAL CONFERENCE ON INTELLIGENT TUTORING SYSTEMS-ITS'96, 3., 1996, Montreal. **Proceedings...** Berlin: Springer-Verlag, 1996.
- [LES97] LESTER, J. et. al. Mixed Initiative Problem Solving with Animated Pedagogical Agents. IN: AIED: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 8.,1997. **Proceedings...** Kobe, Japan: [s.n.], 1997.
- [LEW97] LEWIS, W J.; SHAW, E. Using Agents to Overcome Deficiencies In Web-Based Courseware. IN: AIED: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 8.,1997. **Proceedings...** Kobe, Japan: [s.n.],1997.

- [LUC94] LUCENA, M.W.F.P. **O uso das tecnologias da informática para o desenvolvimento da educação.** Rio de Janeiro: COPPE, 1994. (Publicações Técnicas ES-301/94).
- [LUC98] LUCKIN, R. Ecolab: Exploring the Construction of a learning Assistant. In: INTELLIGENT TUTORING SYSTEMS-ITS, 4., 1998, San Antonio. **Proceedings...** Berlin: Springer-Verlag, 1998. (Lecture Notes in Computer Science, v.1452).
- [LUZ97] LUZZI, F. et al. WOHLER: assistente inteligente para suporte ao ensino de Química. VIII In: SBIE: SIMPOSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 8., 1997, São José dos Campos, SP. **Anais...** São José dos Campos: ITA, 1997.
- [MAR98] MARSELLA, S.C.; JOHNSON, W.L. An Instructor's Assistant for Team-Training In Dynamic Multi-Agent. In: INTELLIGENT TUTORING SYSTEMS-ITS, 4., 1998, San Antonio. **Proceedings...** Berlin: Springer-Verlag, 1998. (Lecture Notes in Computer Science, v.1452).
- [MEN98] MENGELLE, T.; de LÉAN, C.; FRASSON, C. Teaching and Learning with Intelligent Agents; actors. In: INTELLIGENT TUTORING SYSTEMS-ITS, 4., 1998, San Antonio. **Proceedings...** Berlin: Springer-Verlag, 1998. (Lecture Notes in Computer Science, v.1452)
- [MIL82] MILLER, M.L. A Structured Planning and Debugging Environment for Elementary Programming. In: SLEEMAN, D.H. ; BROWN, J.S. (Eds.) **Intelligent Tutoring Systems.** New York, Academic press, 1982.
- [MIN87] MINSKY, M. **A Sociedade da Mente.** Lisboa: Francisco Editores, 1987.
- [MIT97] Mitsuru, I. et al. Opportunistic Group Formation. In: AIED: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 8., 1997. **Proceedings...** Kobe, Japan: [s.n.], 1997.
- [MOR97a] MORIN, J.F.; LELOUCHE, R. Tutoring Knowledge Modelling as Pedagogical Agents in an ITS. In: AIED: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 8., 1997. **Proceedings...** Kobe, Japan: [s.n.], 1997.
- [MOR97] MÓRA, M.C. et al. Modelling dynamic aspects of intentions. In: PORTUGUESE CONFERENCE ON ARTIFICIAL INTELLIGENCE, 1997. **Proceedings...** Berlin: Springer-Verlag, 1997.
- [MOR98] MÓRA, M.C.; LOPES, J.G.; COELHO, J.G.; VICCARI, R. BDI models and systems: Reducing the gap. In: Agents Theory, Architecture and Languages Workshop, 8., 1998, Canarias. **Proceedings...** Berlin: Springer-Verlag, 1998. (Lecture Notes In Computer Science)
- [MOR99] MÓRA, M.C. **Um modelo de agente executável.** Porto Alegre: CPGCC/UFRGS, 1999.
- [MOU96] MOUSSALE, N.M.; VICCARI, R.M.; CORRÊA, M. Intelligent Tutoring Systems Modelled Through the Mental States. In: SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICIAL - SBIA, 13., 1996. **Proceedings...** Berlin: Springer-Verlag, 1996.

- [MUR97] MURRAY, W.R. Knowledge-based Guidance In the CAETI Center Associate. In: AIED: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 8.,1997. **Proceedings...** Kobe, Japan: [s.n.], 1997.
- [NIE95] NIELSEN, J. **Heuristic Evaluation.** Disponível em <http://www.useit.com/papers/heuristic> (ago.1995.).
- [NIL87] NILSSON, N.; Genesereth, M. **Logical Foundations of Artificial Intelligence.** [S.l.]:Morgan Kaufmann, 1987.
- [NOR85] NORMAN, D. **El aprendizaje y la memoria.** Madrid: Alianza Editorial, 1985.
- [NUN93] NUNES, M.G.V. et al. **Uso de Hipertexto/hipermídia em Sistemas Tutores Inteligentes.** São Carlos: ICM, 1993.
- [OHL87] OHLSSON, S. Some principles of Intelligent Tutoring. In: LAWLER,R.;YAZDANI,M.(Eds.). **Intelligent Tutoring Systems.** USA: Ablex Publishing Corporation, 1987.
- [OLI92] OLIVEIRA, F; VICCARI, R. **Sistemas Tutores Inteligentes.** Porto Alegre: CPGCC/UFRGS ,1992.
- [OLI94] OLIVEIRA, F.M. **Crerios de equilibrao para sistemas tutores Inteligentes.** Porto Alegre: CPGCC/UFRGS, 1994.
- [OLI95] OLIVEIRA, M.;GIRAFFA,L.M.M.. LEONARDO - mquina de auxlio à resoluo de problemas. In: SBIE: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO,5.,1997, Florianópolis, SC. **Anais...** Florianópolis: UFSC, 1995.
- [PAI96] PAIVA, A. Learner Modelling Agents. In: EUROAIED: EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 1., 1996. **Proceedings...** Lisbon: Calouste Gulbenkian, 1996.
- [PAI97] PAIVA, A. Learner Modelling for Collaborative Learning Environments. IN: AIED: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 8.,1997. **Proceedings...** Kobe, Japan: [s.n.], 1997.
- [PAP80] PAPERT, S. **Mindstorms**: children, computers and powerful ideas. New York: Basic Books, 1980.
- [PIA 70] PIAGET, J. **Psicologia e pedagogia.** São Paulo:Summus,1970.
- [PIA 74] PIAGET,J. et al. **Educar para o futuro.** Rio de Janeiro: Fundação Getúlio Vargas, 1974.
- [PIA 76] PIAGET,J.A **equilibrão das estruturas cognitivas.** Rio de Janeiro:Zahar,1976.
- [PIA 78] PIAGET, J. **Epistemologia genética.** São Paulo:Abril,1978.
- [PIA 80] PIAGET,J.**O nascimento da inteligência na criança.** Rio de Janeiro: Zahar,1980.
- [PIN95] PINTO,S.C. **M-Asserte**: Um Meta-Assistente Adpatativo para Suporte à Navegação em documentos Hipermídias. Rio de Janeiro: COPPE/Sistemas-UFRJ, 1995.

- [PIL96] PILKINGTON, R. M.; PARKER-JONES, C. H. Interacting with computer-based simulation: the role of dialogue. **Computers and Education**, London, v. 27, n.1, 1996.
- [QUA97] QUARESMA, P. **Inferência de atitudes em diálogos**. Lisboa: FCT/UNL, 1997.
- [RAA96] RAABE, A.L.A; JAVIMCZIK, A .M.; GIRAFFA, L.M.M. Eco-Lógico: Ambiente interativo para suporte ao ensino de educação ambiental. In: SBIE - SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 7.,1996. **Anais...** Belo Horizonte: DCC/UFGM, 1996.
- [RAO92] RAO, A.S; GEORGEFF, M. P. An abstract architecture for rational agents. In: INTERNATIONAL CONFERENCE ON PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING – KR, 3., 1992. **Proceedings ...** [S.l.]:Morgan Kaufman, 1992.
- [RAO95] RAO, A.S; GEORGEFF, M. P. BDI-agents: from theory to practice. In: INTERNATIONAL CONFERENCE ON MULTIAGENT SYSTEMS – ICMAS, 1., 1995. **Proceedings...** San Francisco: [s.n.], 1995.
- [ROS92] ROSHELLE, J.Learning by collaborating: convergent conceptual change. **Journal of the Learning Sciences**, [S.l], v.2, p.235-276, 1992.
- [ROS96] ROSSLER,F.;ZANESCO,J.;GIRAFFA,L.M.M. XPS/E- Plus : uma ferramenta educacional para construção de Sistemas Especialistas baseados em Regras. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 7., 1996. **Anais...** Belo Horizonte: DCC/UFGM, 1996.
- [RUS96] RUSSEL, S.; NORVIG, P. **Artificial Intelligence: A modern Approach**. New Jersey: Prentice Hall, 1996.
- [SAL91] SALOMON, G.; PERKINS, D.N; GLOBERSON, T. Partners in cognition: Extending human intelligence with intelligent technologies. **Educational Research**, [S.l], v.20, n.3, 1991.
- [SAV90] SAVIANI, D. **Pedagogia Histórico Crítica: primeiras aproximações**. São Paulo: Ed. Cortez, 1990.
- [SEA84] SEARLE, J.. What is an intentional state? In: DREYFUSS, D., HALL, S. (Eds.) **Intentionality and Cognitive Science**, [S.l], v.42, 1984.
- [SEL88] SELF, J. **Artificial Intelligence and Human Learning**. London: Chapman Hall, 1988.
- [SEL95] SELF,J. Dormobile: a Vehicle for Metacognition. In: EMERGING COMPUTER TECHNOLOGIES IN EDUCATION, 1., 1995. **Proceedings ...**Charlottesville: ACE, 1995.
- [SHA95] SHAPIRO, B.; VAN DEN BROEK, P.; LETCHER, C. Using story-based causal diagrams to analyse disagreements about complex events. **Discourse Processes**, [S.l], v.20, n.1, 51-77, 1995.
- [SHO90] SHOHAM,Y. **Agent-oriented programming**. Stanford: CSD-Stanford University, 1990.
- [SHO93] SHOHAM,Y. Agent-oriented programming. **Artificial Intelligence**, Amsterdam, v.60, n.1,1993.

- [SIC92] SICHMAN, J.S.; DEMAZEAU, Y.; BOISSIER, O. When can knowledge-bases systems be called agents? In: SBIA – SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICIAL, 9., 1992. **Anais...** Rio de Janeiro: SBC, 1992.
- [SIL92] SILVEIRA, R. A. **Inteligência Artificial em Educação: um modelo de sistema tutorial inteligente para microcomputadores.** Porto Alegre, FAGED/PUCRS, 1992.
- [SIL97] SILVEIRA, R.A.; VICCARI, R.M.. Projeto Eletrotutor: Desenvolvimento e Avaliação de Ambientes Inteligentes de Ensino-Aprendizagem. In: CLEI-PANEL: CONFERÊNCIA LATINOAMERICANA DE INFORMÁTICA, 23., 1997. **Proceedings...** Valparaíso: CLEI, 1997.
- [SIM69] SIMON, H. **The Sciences of the Artificial.** Cambridge: MIT Press, 1969.
- [SIN94] SINGH, M.P. **Multi-agent Systems: A theoretical Framework for Intentions, Know-how, and Communications.** Heidelberg: Springer-Verlag, 1994. (Lecture Notes in Artificial Intelligence, v. 478)
- [SLE82] SLEEMAN, D. Assessing aspects of competence in basic algebra. In: SLEEMAN, D. ; BROWN, J.S. (Eds.). **Intelligent Tutoring Systems.** New York: Academic Press, 1982.
- [SOL84] SOLOWAY, E.M.; JOHNSON, W.L. Remembrance of blunders past: a retrospective on the development of PROUST. In: COGNITIVE SCIENCE SOCIETY CONFERENCE, 6., 1984. **Proceedings...** New Jersey: Lawrence Erlbaum, 1984.
- [SOU95] SOUZA, L.; GARCIA, L.; GIRAFFA, L.M.M. GUTEMBEG- ferramenta de hipermídia para fins educacionais. In: SBIE: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 5., 1997, Florianópolis, SC. **Anais...** Florianópolis: UFSC, 1995.
- [SOU97] SOUZA, Patrícia C. **Um Sistema de Autoria para Adventures Educacionais.** Florianópolis : CPGCC/UFSC, 1997.
- [STE77] STEVENS, A.L.; COLLINS, A. The goal structure of a Socratic tutor. In: NATIONAL ACM CONFERENCE, 1., 1977. **Proceedings...** New York: ACM, 1977.
- [STR85] STREIBEL, M. Analisis critico de tres enfoques del uso de la informatica en La Educacion In: [S.n]. **School uses of microcomputers reports from a National Survey.** Baltimore:[s.n.], 1985.
- [TAY80] TAYLOR, R.P. **The computer in school: tutor, tools, tutee.** New York: Teachers College Press, 1980.
- [TEI98] TEIXEIRA, J.F. **Mentes e máquinas: uma introdução à ciência cognitiva.** Porto Alegre: Artes Médicas, 1998.
- [VAS97] VASSILEVA, J. Goal-Based Pedagogical Agents. In: AIED: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 8., 1997. **Proceedings...** Kobe, Japan: [s.n.], 1997.
- [VIC90] VICCARI, R. **Um Tutor Inteligente para a programação em Lógica: Idealização, Projeto e Desenvolvimento.** Coimbra: Universidade de Coimbra, 1990.

- [VYG84] VYGOTSKY, L.S. **A formação social da mente**. São Paulo: Martins Fontes, 1984.
- [YIN98] YIN, J. et al. Incorporating Personality into a Multi-agent Intelligent System for Training Teachers. In: GOETTL, B.P. et al. (Ed.). **INTELLIGENT TUTORING SYSTEMS INTERNATIONAL CONFERENCE, ITS. Proceedings ...** Berlin: Springer-Verlag, 1998.
- [WAX93] WAXMAN, H. BRIGHT, G. Approaches to Research on Teacher Education and Technology. Virginia, USA: AACE, 1993.
- [WER83] WERNER, E. Co-operating agents: A unified theory of communication and social structure. In: GASSER, L.; HUHNS, M. (Eds.). **Distributed Artificial Intelligence**. London: Pitman, 1983.
- [WER95] WERNECK, V.M.B. **Ambiente de Desenvolvimento de Sistemas baseados em Conhecimento**. Rio de Janeiro: COPPE/UFRJ, 1995. Tese de Doutorado.
- [WHI86] WHITE, B.Y.; FREDERIKSEN, J.R. Intelligent Tutoring Systems based upon qualitative model evolutions. In: NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, 5., 1986. **Proceedings...** Los Altos: Morgan Kaufmann, 1986.
- [WIN84] WINOGRAD, T.; FLORES, F. **Understanding Computers and Cognition: a new foundation for design**. New Jersey: Ablex Publishing, 1984.
- [WOO95] WOOLDRIDGE, M. **On the logical Modelling of Computational Multiagent Systems**. Manchester: UMIST, Department of Computation, 1992.
- [WOO95a] WOOLDRIDGE, M.; JENNINGS, N.R. **Intelligent Agents: Theory and Practice**. London: QM&Wc, 1995.