

FACULDADE DE INFORMÁTICA

PUCRS – Brazil

<http://www.inf.pucrs.br>

ARQUITETURAS DE SISTEMAS TUTORES INTELIGENTES

Rodrigo Rafael Villarreal Goulart and Lucia Maria Martins Giraffa

Technical Report Series

Number 011

June, 2001

Contact:

rgoulart@inf.pucrs.br

<http://www.inf.pucrs.br/~rgoulart>

giraffa@inf.pucrs.br

<http://www.inf.pucrs.br/~giraffa>

Rodrigo Rafael Villarreal Goulart is a graduate student of PUCRS – PUC/RS – Brazil. He is member of the GIE research group (Computer Science apply to Education Research Group) since 2000. He receives a federal graduate research grant from CAPES (Brazil) to support his research.

Lucia Maria Martins Giraffa works at PUCRS/Brazil since 1986. She is an titular professor and leader of the GIE Gr. She develops research in Multi-agent systems, Artificial Intelligence apply to Education, and design of educational software. She got her Ph.D. in 1999 at UFRGS (Brazil).

Copyright © Faculdade de Informática – PUCRS

Published by the Campus Global – FACIN – PUCRS

Av. Ipiranga, 6681

90619-900 Porto Alegre – RS – Brazil

ARQUITETURAS DE SISTEMAS TUTORES INTELIGENTES

Relatório Técnico N° 011/2001

Rodrigo Rafael Villarreal Goulart (mestrando)¹

Lucia Maria Martins Giraffa²

1 Introdução

A modelagem de Sistemas Tutores Inteligentes (STI) é uma tarefa complexa, pois implica em considerar os três módulos fundamentais da arquitetura tri-partida, proposta por Carbonel [CAR70] e revista por Self [SEL99], de forma integrada e com tarefas concomitantes. Ao se modelar um STI devemos considerar as características do domínio (conteúdo), o comportamento observável e mensurável do aluno (modelo do aluno) e, o conjunto de estratégias a serem adotadas pelo módulo tutor na busca de um ensino personalizado.

A sistematização do conhecimento pedagógico a ser colocado no tutor consiste na tarefa mais complexa de toda a arquitetura, no que concerne à modelagem e implementação. No entanto, desde o início da pesquisa em Inteligência Artificial e Educação (IA-ED), a necessidade de inserir mecanismos de apoio à aprendizagem, tais como estratégias e táticas de ensino, baseadas naquelas utilizadas pelos professores em sala de aula, é um dos grandes desafios na pesquisa desta área. As limitações de hardware e software são fatores que contribuem para que esta tarefa não seja atingida ainda na sua plenitude.

Novas tecnologias surgiram e permitiram aos pesquisadores resgatar muitas questões em aberto na pesquisa de STI. A tecnologia de agentes tem se mostrado muito promissora para modelagem e implementação de STI [GIR99].

Neste trabalho realizamos uma síntese desta abordagem multiagente e relacionamos alguns sistemas que exemplificam a pesquisa recente em STI.

¹ E-mail: rgoulart@inf.pucrs.br

² E-mail: giraffa@inf.pucrs.br

Este trabalho está dividido em 4 seções. Na seção 2 são abordados os papéis desenvolvidos pelo módulo tutor nos STI, e as mudanças que ocorreram em função da evolução das tecnologias de modelagem. Na seção 3, são descritas as duas formas para modelagem de STI e alguns com estas abordagens. Por fim, na seção 4, são apresentadas algumas considerações a respeito da modelagem em STI e a sua relação com o trabalho proposto no PEP [GOU00].

2 O módulo tutor em STI

A idéia de se utilizar o computador como uma “máquina de ensino” buscando automatizar o processo de aprendizagem, motivou os primeiros esforços da década de 60 na utilização de computadores para suporte às atividades docentes. O paradigma vigente, na época, era eminentemente comportamentalista (Psicologia) e tecnicista (Ciência da Educação).

As tentativas iniciais baseavam-se em apresentar o material instrucional selecionado pelo professor ou especialista, de forma estruturada obedecendo a uma determinada seqüência. Não havia previsão de escolhas por parte do aluno. Os sistemas basicamente ofereciam duas opções para que o aluno escolhesse a solução do problema. A modelagem apoiada em textos e exercícios associados, onde os alunos adquiriam conhecimento e habilidades em um determinado domínio (conteúdo).

Os softwares reproduziam o sistema expositivo de ensino, que permeava o sistema educacional da época. Uma porção do conteúdo era apresentada em uma ou mais telas (no início interfaces orientadas a caracteres, sem gráficos ou desenhos), e a interação limitava-se a apertar a tecla <ENTER> para mudar de tela. Após a seqüência de telas onde o conteúdo era disponibilizado, era apresentado o bloco de exercícios. Estes eram de escolha simples ou escolha múltipla.

Esta forma de ensino computadorizado recebeu o nome Computer Assisted Instruction (CAI). Eles eram impessoais, sob o ponto de vista instrucional, e não objetivavam nenhum tipo de personalização levando em conta o perfil do usuário (aluno).

Na década de 70, Carbonell [CAR70] apresentou uma nova proposta sobre como tratar a questão da “aprendizagem-computadorizada”. Ele propôs, com um sistema denominado SCHOLAR, com uma nova maneira de conceber sistemas educacionais, levando em

consideração a forma como o professor estrutura e desenvolve o conteúdo em sala de aula. Ou seja, considerando a dinamicidade que existe na relação aluno-professor. Ele se baseou no fato de que numa situação de sala de aula o professor observa os alunos e faz uma verificação constante do que está acontecendo durante o andamento das atividades na sala de aula. E isto deveria ser possível de ser modelado em um software educacional. No entanto, apesar de ser uma idéia aparentemente “simples”, trata-se de uma tarefa de alta complexidade, tanto em nível de modelagem, como de implementação.

O professor recebe “*feedback*” dos alunos tanto verbal, como na forma de atitudes observadoras (movimentos dos olhos, expressão do corpo, etc.). Quando ele aplica um exercício ou teste, é possível medir ou inferir o estado cognitivo corrente do aluno (indicador do seu aproveitamento). A partir da avaliação destes dados ele muda seu comportamento (estratégias) e táticas para atuar junto ao aluno.

Esta dinâmica de sala de aula é que se pretende “transferir” ao sistema, i.é, a idéia de se buscar uma instrução mais personalizada, ou menos impessoal. Ao contrário dos CAIs, estes novos sistemas não realizariam a tarefa de ensinar de uma forma única, mas como uma estrutura orientada à informação, onde o conteúdo que o sistema possui é representado computacionalmente através de diversas formas de representação do conhecimento. No sistema proposto por Carbonel, o conteúdo era representado por redes semânticas. Desta forma, o sistema mantém um diálogo com o aluno, através do formato textual, utilizando um subconjunto da língua natural em que o sistema foi desenvolvido. O sistema se baseava nos avanços da área de Inteligência Artificial (IA), na época (década de 70).

O sistema SCHOLAR e outros trabalhos como o WHY e SOPHIE de Collins e Brown apud [WEN87], constituem os primeiros Intelligent CAIs (ICAI) ou Sistemas Tutores Inteligentes (STI), lembrando que ICAI e STI eram considerados sinônimos. Atualmente isto não é consenso na área de IA-ED. Muitas taxonomias tratam desta classificação, sem apresentar convergência. Maiores detalhes podem ser encontrados em [GIR97; GIR99].

Os STI tem como principal objetivo realizar a tarefa de ensino de um dado conteúdo (domínio) na forma mais adaptada às necessidades individuais do aluno. Estes sistemas se baseiam em uma arquitetura (representada na Figura 2-1) composta basicamente por:

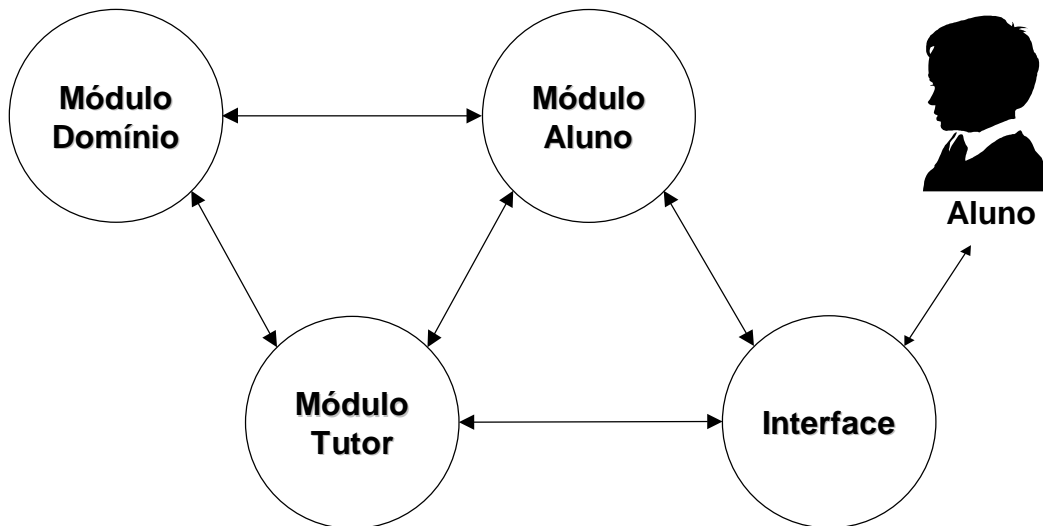


Figura 2-1: Arquitetura clássica de um STI

Módulo aluno: neste módulo estão armazenadas/modeladas as características individuais do aluno (conhecimento individual sobre o domínio, por exemplo);

Módulo tutor: possui o conhecimento sobre as estratégias e táticas para selecioná-las em função das características do aluno (representadas no módulo aluno);

Módulo domínio: detêm o conhecimento sobre a matéria no formato de regras de produção, estereótipos, etc.;

Interface: intermedia a interação entre o tutor e o aluno.

Esta arquitetura é denominada de clássica e também conhecida como funcional tripartida ou tradicional de STI. O termo tripartida se refere às funções associadas aos módulos tutor, do aluno e domínio. Esta proposta trouxe grandes avanços à modelagem de ambientes educacionais, pois separou o domínio da sua forma de manipulação (no sentido de utilização). Permitindo, assim, que estratégias de ensino fossem associadas em função das informações oriundas da modelagem do aluno.

Os STI originalmente foram implementados utilizando paradigmas procedurais e apresentavam certas limitações em função das linguagens utilizadas e do hardware disponível. A impossibilidade de representar domínios complexos, e falta de uma visão (ou controle) global sobre a aprendizagem do aluno (que permitisse a compreensão das concepções errôneas e má interpretações por parte do aluno), não permitiram a criação de sistemas que atendessem às pretensões iniciais dos STI propostas por Carbonell e seu grupo.

No entanto, as pesquisas em Inteligência Artificial Distribuída (IAD), especialmente os Sistemas Multiagentes (SMA), estão permitindo que novos níveis de abstração possam ser incorporados ao projeto de STI, resgatando assim questões em aberto pendentes na área.

Dentre as várias questões que poderiam ser elencadas, destacamos:

- Modelagem de múltiplas estratégias do aluno;
- Seleção de múltiplas estratégias pelo módulo tutor;
- Múltiplas representações do conhecimento associadas a diversas formas de exploração do mesmo.

Começaram a surgir STIs modelados através de sociedades de agentes, os quais substituem os módulos tradicionais, permite, por exemplo, que o módulo tutor seja composto não por uma mas por várias entidades, dividindo a tarefa de tutoria. Cada entidade pode desempenhar tarefas distintas ou de mesma categoria.

Por exemplo, no projeto MATHEMA de [COS95] cada tutor é um especialista em um subconjunto do domínio, permitindo, assim, uma maior especialização do módulo tutor e a adição de novas possibilidades, sob o ponto de vista de uso de múltiplas estratégia de ensino.

Self [SEL99] associou à arquitetura clássica um modelo de interação (Figura 2-2), onde o módulo de domínio não é mais uma forma de tornar as informações inter-relacionadas, mas sim um modelo dos aspectos do conhecimento sobre o domínio que o aluno pode acessar durante as interações com o STI (Modelo da situação).

O módulo do estudante não mais relaciona somente as informações sobre a análise das interações do aluno com o domínio, mas busca uma contextualização maior destas interações em função das ações do aluno, o contexto em que elas ocorrem e a estrutura cognitiva do aluno naquele momento (Modelo de Interação).

O módulo tutor deixou de ser o responsável pela seleção do conteúdo e estratégias para se tornar, de uma forma mais ampla, aquele que conduz o aluno de acordo com objetivos e desafios educacionais que o ambiente proporciona ao aluno (Modelo de permissões).

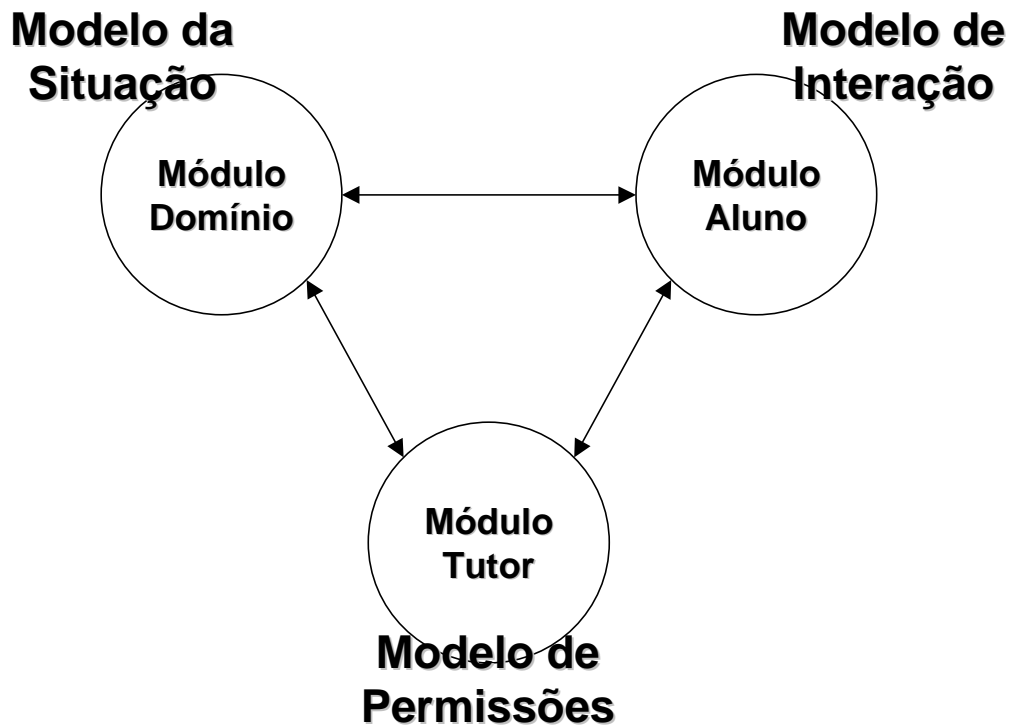


Figura 2-2: Arquitetura do STI ampliada por [SEL99]

No entanto, assim como a arquitetura clássica possuía limitações, ainda existem questões que não puderam ser solucionadas como por exemplo:

o hardware ainda não tem a capacidade de processar satisfatoriamente imagem, voz, cheiro, etc.;

o processamento da linguagem natural ainda é uma questão de pesquisa em aberto;

a comunicação e o tratamento de informações oriundas da interface são restritos, caracterizando uma interação considerada “pobre”.

Todas estas restrições acabam tornando o tutor incapaz de tratar, em sua totalidade, o conjunto de informações que o professor possui em sala de aula.

Contribuições de outras áreas, como a de redes de computadores e sistemas distribuídos, trouxeram novas ferramentas para o ensino, como por exemplo *e-mail*, fórum, *chat*, vídeo conferências, informações on-line, etc. ([BOF00], [CAB00], [FER00], [NET00], e [ZEV00]). Estas contribuições colocam a relevância de um ensino-computadorizado, uma vez que estas

alternativas vêm ao encontro das necessidades que os tutores se propõem a resolver: ensino personalizado e com a possibilidade de ser realizado em grande escala.

Observa-se, no entanto, que a utilização de um tutor, com uma estratégia menos diretiva, a qual não determina totalmente os caminhos da interação entre o aluno e o conhecimento, permite que o STI atue de forma a complementar o ensino em sala de aula, auxiliando o professor em sua tarefa.

Esta assistência desenvolve um processo contínuo, ao qual Giraffa [GIR99] chama de coreografia (Figura 2-3).



Figura 2-3: Coreografia da assistência do STI ao aluno [GIR99]

Esta coreografia representa assistência personalizada do tutor, em função do modelo cognitivo atual do aluno. Esta assistência tem com finalidade organizar o domínio de acordo com os objetivos educacionais modelados no STI (como por exemplo o paradigma construtivista). Esta organização do domínio busca alterar o estado cognitivo do aluno (modelo do aluno), finalizando um ciclo na coreografia desenvolvida pelo tutor.

Desta forma, o tutor realiza suas funções com a finalidade de assistir o aluno em função de suas características individuais, auxiliando-o a organizar o conteúdo apresentado. Esta

assistência segue então os objetivos educacionais desempenhados pelo tutor, e que são explicitados através das estratégias e táticas selecionadas.

Desta forma os STI podem desempenhar suas tarefas, auxiliar o aluno e principalmente coletar informações e complementar as tarefas do professor.

3 Modelagem de tutores em STI

Na literatura encontramos os STI modelados em duas abordagens: a clássica (funcional, em módulos) e utilizando agentes (os módulos tradicionais são substituídos por uma sociedade de agentes). Nesta seção analisaremos a modelagem do tutor nestas duas abordagens.

3.1 Modelagem Clássica

Segundo Wenger [WEN87], na abordagem clássica o papel do módulo tutor é tomar as decisões pedagógicas em função das interações com o aluno. Estas decisões derivam de regras ou estruturas de conhecimento as quais representam o conhecimento do tutor a respeito do domínio, e estas estão representadas de forma definida no sistema (Módulo Tutor).

O tutor utiliza o modelo do aluno e do modelo de domínio para tomada das decisões didáticas e pedagógicas. A nível global, estas decisões afetam a seqüência de apresentação do conteúdo, intervenções e questionamentos a serem feitos ao aluno. Através da forma com que os assuntos estão relacionados, o módulo tutor adapta sua apresentação dos tópicos às necessidades do aluno. A nível local, no sistema (módulo tutor), o tutor decide quando uma intervenção é desejável, se o aluno deve ser interrompido em sua atividade ou não e, o que deve ser apresentado em função das informações avaliadas em um dado momento.

A ordem e a maneira com que cada assunto é trabalhado pode produzir diferentes resultados na aprendizagem.

Nos tutores clássicos o controle varia conforme algum critério de otimização, e costuma ser uma decisão de projeto, ou seja, não existe mudança na forma com que é controlada a interação. Alguns sistemas podem monitorar as atividades dos alunos de perto, adaptando suas ações as respostas dos alunos mas nunca cedendo o controle da interação.

Em diálogos com iniciativas mistas, o controle é compartilhado pelo estudante e o sistema, assim como ocorre a troca de perguntas e respostas. Neste caso, o sistema deve então ser capaz de responder as perguntas do estudante. Em atividades orientadas ou *guided-discovery learning*, o estudante possui o controle total da atividade, e a única maneira do sistema conduzir o aluno é realizando mudanças no ambiente. Esta forma de controle torna difícil a análise do processo, que poderia ser realizada posteriormente pelo professor. Porém a possibilidade do aluno agir de forma independente impede que o sistema interfira inadequadamente em caso de dúvida.

O tutor analisa as respostas e perguntas do aluno, assim como outras interações (como por exemplo a forma como o aluno desenvolve a solução das tarefas) afins de atualizar o modelo de aluno do sistema. Esta análise busca adaptar os objetivos do tutor em função do aluno e do conteúdo que deve ser apresentado.

Em comparação com o professor, os tutores clássicos estão limitados a um único método de ensino e aprendizagem, em função das limitações na quantidade de regras que podem ser descritas. A falta de habilidade para ensinar de maneira flexível, adotando diferentes métodos de ensino de maneira apropriada, e permitindo que os estudantes utilizem diferentes estilos aprendizagem restringem as capacidades de um STI clássico.

Wenger [WEN87] ressalta que no clássico SCHOLAR as estratégias eram restritas e consistiam principalmente na seleção de tópicos locais. Estes tópicos locais estavam localizados nos nodos das redes semânticas³, que continham uma representação estruturada do conteúdo. O aluno podia fazer perguntas ao sistema e este deveria identificar as informações relevantes contidas na pergunta de acordo com o domínio. Isto era feito em função da distância entre o nodo em que o dado relevante da pergunta está localizado, e o nodo atual (tópico atual). A cada nodo foram adicionados pesos (na forma de etiquetas numéricas) que permitiam o tutor identificar se o próximo nodo era ou não mais relevante para então gerar perguntas quando a iniciativa era a sua. No entanto, se estas etiquetas não serviam para a tomada de decisão, eram então realizadas escolhas aleatórias.

³ “Uma Rede Semântica é uma estrutura para a representação do conhecimento definida como padrão de nodos interconectados por arcos rotulados. As redes deste tipo não só captam as definições dos conceitos mas também, inerentemente, proporcionam ligações com outros conceitos” [CHA96]

A partir dos resultados obtidos com o SCHOLAR, Collins apud [WEN87], apresentou o projeto WHY onde seu principal objetivo era fazer um tutor, fruto de sua análise da aplicação do método socrático em tutores e da pesquisa realizada por Carbonell.

Segundo [GIR97], um tutor socrático ensina através de uma abordagem de exposição indireta, que consiste na oferta, ao aluno, de questões sucessivas que visam formular princípios gerais baseados em casos particulares para ele poder analisar e avaliar hipóteses, descobrir contradições e finalmente fazer inferências corretas.

Collins gerou aproximadamente sessenta regras que procuravam montar uma estratégia tutorial, que levava em consideração o tipo de aluno que usava o tutor.

Já na metade da década de 80, era crescente o interesse em modelos didáticos computacionais diversificados. No entanto, isso permaneceu como uma limitação aos tutores, já que novas estratégias eram abordagens baseadas no conhecimento do tutor humano, cuja habilidade é complexa e, até aquele momento, difícil de ser tratada computacionalmente.

Alguns trabalhos tentaram buscar alternativas na tentativa de tornar um tutor flexível na condução das interações com o aluno, alterando suas estratégias. O tutor QUADRATIC apud [WEN87], podia modificar suas próprias estratégias, mas estas modificações eram basicamente ajustes de parâmetros e não uma mudança efetiva de estratégia.

Outro exemplo é o Tutor Espertinho de Konzen [KOZ00], onde o primeiro passo na utilização do sistema é a determinação do perfil do aluno. Este perfil influenciará na forma de exposição do conteúdo é determinado através de um questionário que determinam o nome, preferência de cores e características de interação (se o aluno possui um perfil visual ou auditivo). Estas informações restringem a adaptação do tutor à determinação de fatores sobre a interface, e que por sua vez são estáticos, já que não se alteram durante a sessão de trabalho.

Observa-se que os STI clássicos, de uma forma geral, se caracterizam por utilizar uma única estratégia para o ensino do aluno. Ou seja, estes sistemas possuem pouca versatilidade em seu comportamento pedagógico, sem proporcionar uma adaptação dinâmica (em nível de estratégias) as características individuais de aprendizagem de cada aluno.

No trabalho de [KAW00] é proposto um modelo de STI para educação de adultos. Para tal, o modelo abrange alguns princípios pedagógicos das teorias de Knowles, Cross, Bruner e Vygostky apud [KAW00] aplicados para adultos. Estes princípios foram aplicados a cada um

dos módulos da arquitetura clássica de STI. O modelo foi implementado utilizando mapas conceituais para representar interna e graficamente o conteúdo. Um editor gráfico permite que tanto o aluno quanto o autor da base de conhecimento (domínio) possam criar mapas conceituais para descrever o conhecimento no sistema. O módulo tutor tem a tarefa de transformar os relacionamentos expressos no mapa em asserções, para então apresentá-las ao aluno. Através da análise gráfica do mapa conceitual do aluno, o módulo tutor pode propor ao aluno que novos relacionamentos podem ser incluídos. Esta análise se dá em função da comparação entre o mapa conceitual do aluno e o do professor.

3.2 Modelagem com agentes

Em função do desenvolvimento da IA distribuída, e a crescente pesquisa em Sistemas Multiagentes (SMA), a tecnologia de agentes passou a ser adotada como uma metodologia alternativa para projetar STI.

Segundo [GIR99], a razão fundamental para modelar STI com uma arquitetura multiagente são suas capacidades de comunicação e interação. Agentes podem se adaptar e aprender durante uma sessão. [SIL00] considera que a utilização de sociedades baseadas em agentes está se consolidando como uma alternativa apropriada para o projeto de STI.

Existem na literatura diferentes abordagens da utilização de agentes em STI. Segundo [GIR98], os agentes desenvolvidos para ambientes de ensino recebem o nome de agentes pedagógicos. Seus objetivos podem ser descritos em função do seu comportamento (estratégia). Os comportamentos possíveis para um tutor, segundo [GIR98] são:

- Guia: o agente é diretivo em suas intervenções e monitora o aluno todo tempo, conduzindo o aluno na resolução do problema por todo processo de interação;
- Assistente: o agente é menos diretivo e monitora o aluno todo o tempo, intervindo baseado em heurísticas sobre a resolução do problema naquele domínio;
- Facilitador: o agente monitora o aluno todo o tempo porém não é diretivo, ele apenas dá dicas sobre a resolução do problema e só intervém quando solicitado.

O tutor pode utilizar um ou vários comportamentos, sendo assim os objetivos podem variar em função das informações recebidas do usuário (ações do aluno) e da situação que se deseja atingir (plano).

Os agentes pedagógicos podem atuar como tutores virtuais, alunos virtuais ou colegas virtuais que auxiliam no processo de aprendizagem. Estas diferentes formas de atuação podem compor grupos de agentes que distribuem entre si as tarefas. Estas tarefas podem ser de maior abrangência (modelagem do aluno ou seleção da estratégia e táticas) ou de menor abrangência (cada agente é responsável por uma estratégia).

A distribuição de tarefas permite um melhor desempenho sistema, subdividido a complexidade da tutoria em tarefas menores, a tarefa de tutoria passa a ser executada em um nível mais alto de abstração. Desta forma, Gouarderes [GOU99] sugere uma caracterização do processo de aprendizagem desenvolvido em sistemas multiagentes em três níveis de abstração:

- Aprendizagem por replicação, onde os agentes disponibilizam o conteúdo, a representação da estratégia pedagógica selecionada, e dentro desta arquitetura um dos agentes representa as ações do professor, e processo de aprendizagem esta representado de forma reativa pelos agentes;
- Aprendizagem por tautologia, onde a apresentação do conteúdo é desenvolvida de forma a guiar o aluno através de agentes especializados que auxiliam o aluno, nesta arquitetura o processo de aprendizagem se caracteriza pela capacidade de adaptação dos agentes na condução do aluno ao longo das interações;
- Aprendizagem por interações dinâmicas e compartilhadas, onde o computador é mais ativo e a informação não é fornecida somente pelo sistema, mas pode ser modificada e/ou gerada pelo aluno, desta forma o processo de aprendizagem caracteriza um modelo cognitivo.

A partir destes conceitos, passaremos a analisar alguns STIs que utilizam agentes em sua arquitetura e como se desenvolve a tarefa de tutoria nestes sistemas. Estes sistemas foram intencionalmente escolhidos por serem atuais e utilizarem a abordagem de agentes de forma a exemplificar os conceitos até aqui apresentados.

3.2.1 LeCS [ROS99]

O sistema LeCS [ROS99] tem como objetivo dar suporte à aprendizagem colaborativa através da Internet utilizando o método de Estudos de Casos. Este método tem como objetivo ensinar aos alunos conteúdos baseados em situações complexas e reais. São apresentadas situações problema onde o aluno deverá buscar uma solução. A falta de uma estrutura definida no domínio permite que o aluno desenvolva uma flexibilidade cognitiva para que possa lidar com tais situações.

O sistema LeCS é composto por uma sociedade de agentes, composta por: agentes de interface, agente de informação, agente de aconselhamento e um agente facilitador (Figura 3-1). O papel do tutor é desempenhado de forma distribuída entre estes agentes.

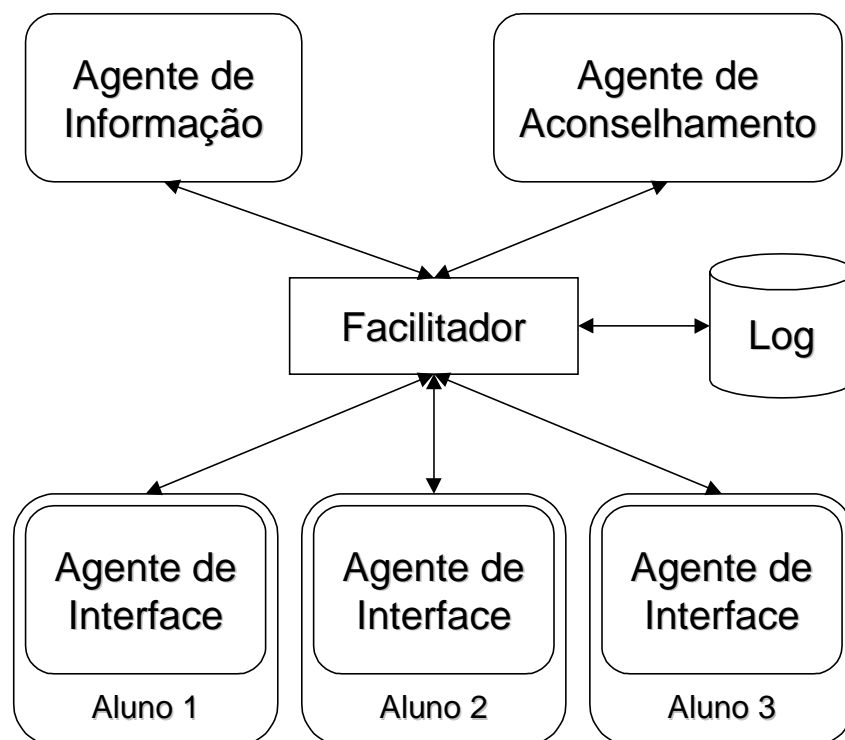


Figura 3-1: LeCS [ROS99]

O Agente de Informação lida com o domínio e com o conhecimento pedagógico. Ele armazena o conteúdo a ser apresentado ao aluno bem como o conhecimento para transmiti-lo. Este conhecimento está armazenado sob a forma de casos e regras.

O Agente de Aconselhamento dá assistência aos participantes através da monitoração do processo de aprendizagem e realizando intervenções quando necessário. Estas intervenções

dizem respeito ao tempo, à participação, aos erros de compreensão dos alunos que são específicos do estudo de casos e à coordenação no uso das ferramentas da interface.

O Agente de Interface além de se comunicar com o aluno realiza a tarefa de modelagem do aluno. A base de crenças do agente inclui as informações que o agente armazena sobre usuários individuais: o passo da metodologia em que cada participante está trabalhando, a resposta conjunta do grupo para caso (esta informação é armazenada em apenas um agente de um grupo de alunos). O agente prepara e ordena as requisições do aluno em uma estrutura de dados interna monitorando suas atividades.

A comunicação entre os agentes é intermediada pelo agente facilitador.

Com base nas informações (modelo do aluno) armazenadas pelo Agente de Interface, este agente e os Agentes de Informação e Aconselhamento desenvolvem o conteúdo e inferem o auxílio necessário para cada aluno. O conhecimento sobre a estratégia esta distribuída. O Agente de informação contém o conhecimento sobre a estratégia e a apresentação do conteúdo, já o Agente de Aconselhamento possui o conhecimento específico de como auxiliar o aluno por meio da estratégia de estudo de casos.

3.2.2 ITStrategic [MAR00]

O sistema ITStrategic de Marietto tem como objetivo definir as estratégias instrucionais do STI de forma dinâmica, visando aumentar a adaptabilidade do sistema ao perfil do aluno.

O ITStrategic é composto por cinco módulos (Figura 3-2): o módulo percepção, módulo estudante, o módulo tutoria e o módulo entrega.

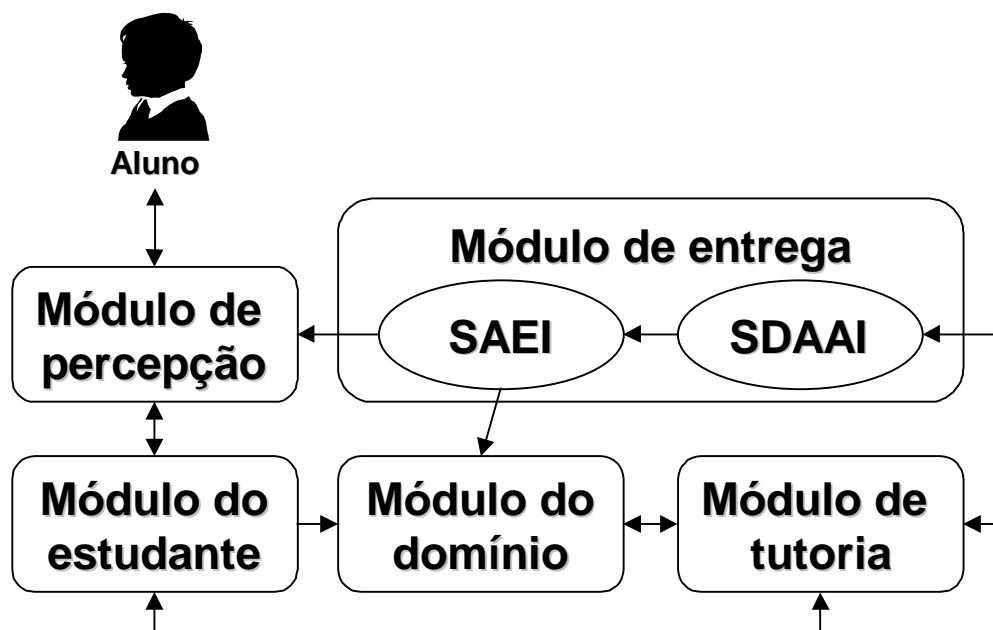


Figura 3-2: Arquitetura do ITStrategic [MAR00]

No módulo percepção é formado por receptores (administrador, educador e estudante), que personalizam os diálogos (interface) com os diferentes usuários do sistema (administrador, professor e aluno). Cada usuário tem uma interação relacionada a suas tarefas para com o sistema (como por exemplo o professor, que tem o papel de atualizar a base de domínio com novos conteúdos).

O módulo estudante é formado pelo agente Guia, que é responsável pela ativação ou criação de um modelo de aluno, bem como atualizar e disponibilizar este modelo à sociedade de agentes.

O módulo tutoria é composto pelo agente Tutor, que é responsável pela seleção da estratégia de acordo com o modelo cognitivo atual do aluno. Ele utiliza as informações do aluno para criar um plano instrucional (chamado plano instrucional linear). Estas estratégias são descritas por meio de Arquiteturas Instrucionais, propostas por Ruth Clark apud [MAR00].

Ruth Clark apresenta quatro arquiteturas instrucionais para sistemas de tutoria. São elas: receptiva, diretiva, orientada a descoberta e exploratória. A partir destas arquiteturas é proposto o momento instrucional adequado para aplicação de cada uma delas. Para isso são definidos os conceitos de desempenho de transferência próxima e distante. Desempenho de transferência próxima define uma tarefa para o aluno com formato padronizado, com uma estrutura que permite a execução passo a passo, numa seqüência bem definida de procedimentos. Já o

desempenho de transferência distante exige do estudante um certo grau de julgamento e abordagens diferentes para realizar suas tarefas.

As arquiteturas propostas por Clark foram adaptadas para o protótipo ITStrategic, a fim de demonstrar a dinâmica a sua seleção. É possível fazer uma analogia destas arquiteturas e sua correspondência com os comportamentos do tutor descritos anteriormente por [GIR98]. Na Tabela 3-1 estão relacionadas as linhas gerais para seleção de uma arquitetura instrucional no ITStrategic e os comportamentos do tutor.

Usada para	Arquitetura	Nível do estudante	Comportamento do tutor
Transferência próxima	Diretiva	Básico	Guia
Transferência distante	Orientada por	Básico	Assistente
Transferência próxima ou distante	Descoberta	Intermediário	Assistente
Transferência próxima ou distante	Exploratória	Avançado	Facilitador

Tabela 3-1: Seleção da Arquitetura Instrucional x Comportamentos do tutor

O plano criado pelo módulo de tutoria é então executado pelo módulo de entrega que é composto por duas sociedades de agentes: SAEI (Sociedade dos agentes de eventos instrucionais) e a SDAAI (Sociedade didática dos agentes com arquitetura instrucional). Os SDAAI fazem a exposição do conteúdo em função das arquiteturas instrucionais determinadas pelo Módulo Tutorial. Nesta sociedade cada agente é responsável por uma arquitetura, retirando do tutor a tarefa de sequenciamento do conteúdo.

Marietto considera que a utilização de sociedades de agentes permite analisar a tutoria sob aspectos sociais e políticos, dando uma nova abordagem à implementação da instrução por computador.

3.2.3 *Troublemaker* [AIM96] e [FRA97]

Aïmuer e Frasson desenvolveram um estudo [AIM96] para verificar a utilização de múltiplas estratégias em função de diferentes níveis de conhecimento. Neste trabalho foram analisadas as interações com os seguintes modelos: um a um, com um colega (*companion*), através do ensino (*by teaching*) e aprendizagem por perturbação (*disturbing*). Esta última, proposta por Aïmuer e Frasson [AIM96].

Na aprendizagem um a um o conteúdo é desenvolvido pelo tutor, o qual analisa as ações do aluno e provém um ensino adaptado às necessidades individuais do aluno.

Na aprendizagem com um colega, o aluno aprende em companhia de um colega artificial (agente) que interage com o tutor, em conjunto com o aluno. O aluno artificial também busca respostas às questões propostas pelo tutor, como também as discute com o aluno real. Desta forma procura-se que o aluno acompanhe o raciocínio do colega.

A aprendizagem por ensino, coloca o aprendiz humano na posição de professor. Desta forma, o aluno tem de ensinar o seu companheiro e pode utilizar exemplos, explicações e soluções criadas por ele para auxiliar o colega artificial.

Na aprendizagem por perturbação, participam da interação o tutor, o aluno e o aluno artificial polemizador (Encrenqueiro - *Troublemaker*). Esta proposta difere da aprendizagem com um colega, uma vez que o “Encrenqueiro” possui comportamentos diferenciados. Ele pode dar respostas para um dado problema com a finalidade de forçar o aluno a reagir e propor a solução correta, ou esperar pela solução do aluno e então dar sugestões, soluções erradas ou um contra-exemplo. Desta forma o aluno apresenta sua explicação para o aluno artificial sob a supervisão do tutor. Se o aluno não é capaz de dar a resposta correta, então o tutor auxilia, dando indicações para a resposta, ou até mesmo respondendo a questão.

Aïmuer e Frasson realizaram um experimento em [AIM96], utilizando a linguagem HTML onde eram apresentadas questões sobre normas de trânsito para validar o uso de múltiplas estratégias em uma forma progressiva (Figura 3-3). O experimento trouxe os seguintes resultados:

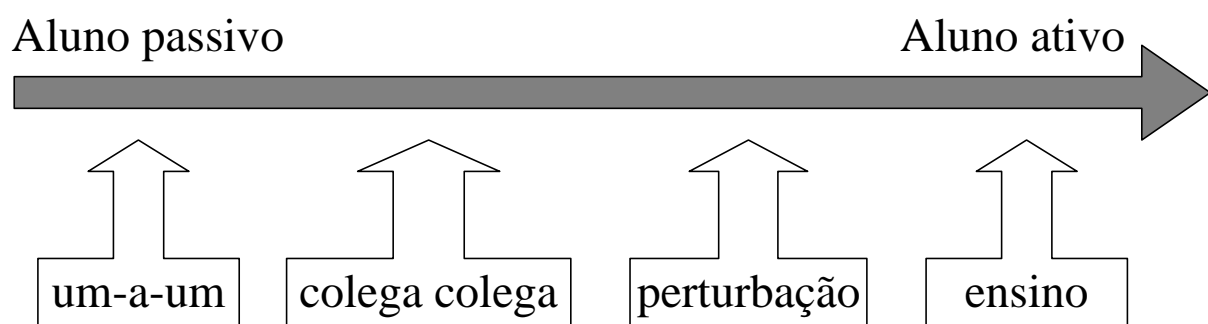


Figura 3-3: Evolução das estratégias de aprendizagem [AIM96]

A aprendizagem por perturbação traz melhores resultados para pessoas com bons conhecimentos sobre o conteúdo, e parece ser perigosa para pessoas com pouco conhecimento;

O colega artificial se mostrou eficiente com alunos fracos;

Frasson sugere que o motivo para a estratégia de perturbação não ser eficiente com alunos fracos seja:

O processo de memorização é mais eficaz através dos erros cometidos ao invés dos acertos, especialmente com alunos com bons conhecimentos.

A aprendizagem por perturbação reforça a atenção e aumenta a percepção de pontos importantes, produzindo significados para memorização e argumentação.

Alunos com bom conhecimento são mais motivados pela competição enquanto os com pouco conhecimento, preferem ajuda ao invés de novos problemas.

Em consequência dos experimentos realizados em [AIM96], Frasson e seu grupo desenvolveram um protótipo utilizando agentes [FRA97], onde se observou que a aprendizagem aparece como um processo distribuído entre vários componentes ativos. Considerando que o conhecimento pedagógico é complexo, muitas vezes impreciso e difícil de avaliar, Frasson utilizou o conceito de Atores para criar um ambiente onde os comportamentos individuais e sociais de diferentes alunos (o real e o artificial) e suas interações com o tutor pudessem ser controlados.

O conceito de atores baseia-se na idéia de agentes inteligentes com capacidades específicas. Como os outros agentes, atores são entidades autônomas e podem operar sem o controle humano, interagindo com os outros atores através de uma linguagem de comunicação. Frasson define atores como um agente inteligente que é reativo, orientado, adaptativo e cognitivo [FRA97].

Desta forma a estratégia de aprendizagem por perturbação é representada da seguinte maneira:

um ambiente com dois atores, o tutor e o agente artificial (*troublemaker*);

o aluno e o aluno artificial trabalham juntos sob a supervisão do tutor;

Frasson conclui que a utilização de uma abordagem multiagente permitiu que o sistema abordasse múltiplas estratégias separadamente e as selecionasse de forma dinâmica.

3.2.4 Lahystotrin [ARC00]

O sistema LAHYSTOTRAIN tem como objetivo o treinamento cirurgiões por meio de um Ambiente com Realidade Virtual (utilizando personagens), Treinamento Teórico e tutoria para resolução de exercícios.

O sistema é composto dois sistemas, um simulador VR (Realidade Virtual) e o ATS (*Advanced Training System*). O ATS é dividido em seis subsistemas: Assistente, Tutor, Cirurgião assistente, Enfermeira, Anestesista, Estudante e a Interface de usuário. Há também uma base de dados associada ao log de uma sessão. Entre estes sistemas, dois são agentes pedagógicos que desenvolvem as seguintes tarefas: o Assistente, que busca ensinar o conteúdo, e o Tutor cuja tarefa é conduzir o processo de treinamento.

Outros três agentes (Assistente de cirurgião, Enfermeira e o Anestesista) são reativos e desenvolvem comportamentos relacionados com ambiente. O simulador de realidade virtual envia informações para o ATS sobre o estado do paciente ou tarefas para treinamento.

Cabe salientar que o subsistema Estudante não é considerado um agente mas uma interface com usuário.

O modelo de usuário é dividido em estático e dinâmico. O estático contém informações sobre preferências, informações pessoais, nível de experiência e o conhecimento teórico prévio do aluno. O modelo dinâmico inclui o desempenho do aluno, exercícios realizados, erros cometidos e o conhecimento sobre patologias e instrumentos.

3.2.5 SmartEgg [MIT00]

No tutor SmartEgg o agente pedagógico é utilizado para auxiliar e ampliar as capacidades de interação do tutor SQLT-Web, cuja aplicação é o ensino de SQL.

O sistema é composto por um servidor (SQLT-Web server) e interação com o usuário se dá por meio de um navegador web através da Internet. O agente pedagógico está personificado na interface por meio de desenho animado, que se comunica com o servidor através da rede utilizando *sockets*. O agente explica as funções do tutor, viabiliza um *feedback* sobre as ações do aluno e expõe outras formas de obter ajuda e informações, auxiliando o aluno no uso do tutor SQL-Web.

Seu comportamento está restrito a três tipos: introdutório, explicativo e de congratulações. Cada comportamento é descrito por um conjunto predefinido de regras, e a seleção delas ocorre por meio da observação das interações do aluno com o SQL-Web.

Neste artigo é apresentada uma avaliação do uso de agentes pedagógicos em um sistema tutor pré-existente. Foram realizados pré e pós-testes, que identificaram na avaliação final um aumento significativo da motivação, resultando em períodos de interação mais prolongados, assim como um aumento da qualidade da aprendizagem. Por outro lado, os autores observaram a dificuldade (tanto técnica como humana) para desenvolver um assistente personificado na interface, que interaja de forma satisfatória.

3.2.6 Explanation Agent [ZOU00]

Zouaq propõe o Agente de Esclarecimentos (Explanation Agent) que tem como principal objetivo prover repostas ou explicações sobre o conteúdo com maior qualidade, identificando problemas que possam ocorrer durante o processo de explicação ou resolução de problemas.

O agente possui as características de reatividade, autonomia, aprendizagem, mobilidade, dedução e capacidade de compreensão. A mobilidade permite que o agente no início de uma sessão se mova do *servidor de cursos* para máquina do aluno, o que torna o código móvel do agente o mais atualizado possível. A capacidade de aprendizagem e dedução permite que ele identifique interpretações errôneas, padrões de comportamento relacionados ao perfil do aluno que são relativos a determinados tipos de erros. O agente também avalia a qualidade suas explicações.

A arquitetura do agente é dividida em camadas. Uma de cognição, outra de explicação e outra de percepção. Cada uma sintetizando uma ou várias das características do agente. Os conteúdos, exercícios e explicações (Domínio) são descritos por meio de um editor, e que permite a inclusão, exclusão ou alteração de informações. Estas informações ficam disponíveis em um servidor onde são armazenados o Domínio, o modelo de estudante (que neste caso representa a estratégia de aprendizagem por perturbação) e o servidor do Agente.

3.2.7 Framework JADE [SIL00]

Em função do estado da arte em STI indicar que a utilização de sociedades baseadas em agentes esta se consolidando como uma alternativa apropriada para o seu projeto, Silveira

propõe uma metodologia e um *framework* numa arquitetura multiagente para criação de ambientes educacionais distribuídos, JADE (*Java Agents for Distance Education Framework*).

O *framework* Jade disponibiliza os recursos necessários para o projeto e construção de STI. Sua arquitetura é baseada em uma sociedade de agentes, segundo a seguinte classificação:

- Agente de domínio: responsável pela apresentação do domínio;
- Gerenciador de exercícios: responsável pelo controle do envio de exercícios ao aluno e o controle de sua performance;
- Gerenciador de exemplos: responsável pela elaboração e controle da apresentação de exemplos;
- Gerenciador de atividades: responsável pelas demais atividades pedagógicas, como simulações, demonstrações, etc.;

Estes agentes podem compor um ambiente projetado para um determinado domínio e compartilham uma mesma base de conhecimento.

Além destes, o *framework* possui três agentes responsáveis por tarefas relacionadas ao ambiente de rede, são eles:

- Gerenciador de modelos de aluno: responsável pela construção e manutenção do conhecimento sobre cada aluno conectado ao sistema;
- Agente gerenciador de comunicação: responsável pela administração da sociedade de agentes e sua comunicação;
- Gerenciador da interface de apresentação e comunicação: responsável pela comunicação ponto-a-ponto entre cada aluno, ambiente de sistema e controle da navegação no ambiente do aluno (ambiente *Web*).

A arquitetura interna de cada agente do *framework* é composta por uma estrutura cognitiva (baseada em estados mentais e compõe a base de conhecimento do agente) e habilidades (particulares do agente e de comunicação).

Como estudo de caso, Silveira desenvolveu o ambiente ELETROTUTOR cujo conteúdo é a Eletrodinâmica para alunos do segundo grau.

3.2.8 AME-A [PER00]

Neste trabalho é proposta a especificação de um agente pedagógico que tem por objetivo a seleção automática de estratégias de ensino adaptadas às características individuais do aluno, seguindo o modelo pedagógico apresentado por Brightman apud [PER00]. Este agente faz parte de uma sociedade de agentes denominada AME-A, Ambiente Multiagente de Ensino-Aprendizagem, proposto por D'Amico apud [PER00].

O modelo do aluno é representado através de 16 perfis definidos por Myers apud [PER00], onde um agente identifica (no início do curso através de um questionário, e durante analisando as interações do aluno). Desta forma o perfil do aluno pode ser alterado dinamicamente. O agente *Seleciona_estratégia* analisa as mudanças de perfil e seleciona a estratégia mais adequada.

3.2.9 MarCo [TED00]

O sistema MarCo (Mediator Artificial de Conflitos) é um ambiente colaborativo para resolução de problemas baseado em um modelo computacional de resolução de conflitos metacognitivo. Este modelo permite que o ambiente infira a respeito dos conflitos do grupo de alunos e interfira na busca de melhores interações.

O sistema é composto por: uma interface distribuída, pela qual os alunos interagem uns com os outros, elaboram uma solução para um dado problema e recebem o auxílio do sistema; Servidor Prolog, que processa as informações oriundas da interface.

O Servidor Prolog é composto por três processos se comunicando: o Processador de diálogo, que recebe as expressões utilizadas pelos alunos para discutir e elaborar a solução e as mapeia em termos de atitudes BDI⁴; o Mantenedor, que atualiza os modelos individuais e de grupo dos alunos; o Agente Mediador, que observa as atitudes BDI dos alunos em busca de conflitos.

⁴ BDI – Belief, Desires and Intentions; crenças, desejos e intenções.

A partir destas informações o Mediador utiliza três níveis de estratégia de mediação de conflitos: primeiro nível, apenas informando que um conflito ocorreu; segundo nível, onde ele além de informar os grupos a respeito do conflito identificado ele busca identificar qual (is) aluno(s) estão dizendo algo que esteja inconsistente com o seu modelo individual, estimulando uma nova reflexão por parte do(s) mesmo(s); terceiro nível, o Mediador sugere ações que conduzam a soluções mais refinadas, para isso o agente gera um modelo de tarefas e de mudanças de estratégias, baseado nas características do grupo.

4 Considerações finais

Modelar um STI, através de uma arquitetura multiagente, permite dividir as tarefas do tutor e tornar sua modelagem mais sistemática. A utilização de múltiplas estratégias viabiliza STIs mais robustos e com maior potencial de adaptação as situações dinâmicas encontradas no processo de aprendizagem que se pretende ao promover a interação entre o aluno e o sistema.

Observa-se, no entanto, que a modelagem de STI com agentes está adquirindo novas perspectivas e gerando novas questões na área de IA-ED. Ao mesmo tempo em que possuímos recursos para ampliar a modelagem de domínios complexos, implementar diferentes estratégias de ensino, monitorar diferentes alunos, etc., novas questões se apresentam. Como por exemplo, a utilização de novas estratégias associadas a recursos de interface, a quantidade de informações a serem tratadas pelo sistema cresce proporcionalmente a exploração e representação do domínio. A tarefa de monitorar o ambiente (sistema) e o(s) aluno(s) durante as interações, podem trazer uma sobrecarga ao sistema como um todo, e comprometer sua performance (desempenho). Porém este conjunto de informações é fundamental para o auxílio personalizado do(s) aluno(s).

Neste contexto, o trabalho de pesquisa realizado no curso de mestrado do PPGCC, tem como objetivo propor uma solução para este problema através da introdução de um agente para auxiliar o tutor nas suas tarefas de gerenciamento das informações do ambiente e atividades dos alunos.

Este trabalho se constitui no PEP (Plano de Estudos e Pesquisa) apresentado e defendido no decorrer do 3º trimestre de 2000, no Programa de Pós-graduação em Ciência da Computação da Faculdade de Informática da PUCRS.

A proposta deste trabalho é utilizar um agente como auxiliar do tutor na tarefa de monitoração das informações referentes a um ambiente dinâmico (jogo educacional) e as interações dos alunos com o ambiente, e assim, reduzir a complexidade do sistema. Também serão definidas quais as funções específicas deste agente dentro de uma arquitetura de STI.

Com base nas colocações de [LIE97], [ETZ94] e [LAS94], onde o foco do auxílio prestado pelo agente está centrado no usuário, empregaremos as características de um agente de interface com o foco no tutor, direcionando seus esforços para o auxílio através da terceirização de tarefas. Desta forma o agente será capaz de automatizar tarefas para o tutor e mediá-las com o aluno.

As propriedades de agentes de interface, e do ambiente, permitem decompor o agente em função das tarefas que podem ser desempenhadas. São elas:

- Agente mediador: que desempenha a tarefa de mediação entre o módulo tutor e a interface, observando o ambiente e executando tarefas para o tutor.
- Agente interface: o agente assume algumas tarefas do tutor.
- Agente facilitador: que atua como um elo entre os demais agentes do sistema, cujas tarefas podem ser distintas e/ou distribuídas, viabilizando a interação entre os mesmos.

Desta forma, um ou mais agentes podem ser adicionados à arquitetura do STI. No entanto, a necessidade de cada um destes agentes e suas funções específicas é uma das hipóteses que serão confirmadas em nossa pesquisa.

A partir das considerações de [WEX97], [BON91] e [GIR99], estenderemos a arquitetura proposta por [GIR99], o MCOE, com a finalidade de validar nossa proposta.

No MCOE o modelo do aluno é composto por estados mentais, crenças e desejos, que são enviados para um *kernel* cognitivo (parte deliberativa do agente tutor) utilizando arquitetura BDI e a ferramenta X-BDI [MOR00].

A cada ação dos alunos, por exemplo apertar um botão (ferramenta), é enviado o conjunto de estados mentais associados e as informações do ambiente. No entanto, se o aluno

não realiza nenhuma ação, o sistema não percebe o que está acontecendo no ambiente. Esta situação caracteriza-se até o momento como uma limitação do sistema.

No decorrer de uma partida de jogo estas ações do aluno podem ter múltiplos significados e implicações. O aluno pode estar deliberadamente testando os limites dos personagens, desejando o colapso do sistema, não cooperar com o colega, etc.. O tutor deve então selecionar a tática mais adequada para o aluno e executar as táticas selecionadas por meio da interface.

Wexelblat, em [WEX97], postula que um agente de interface pode ser um intermediador entre o aluno e o tutor, e tratar os assuntos relacionados à interface de forma autônoma, manipulando os elementos da interface sem instruções explícitas do aluno ou do tutor.

Desta forma podem ser incluídos nesta arquitetura um ou mais agentes para auxiliar o tutor na tarefa de monitoração das informações do ambiente e das atividades do aluno (Figura 4-1), intermediando parte da interação com o aluno.

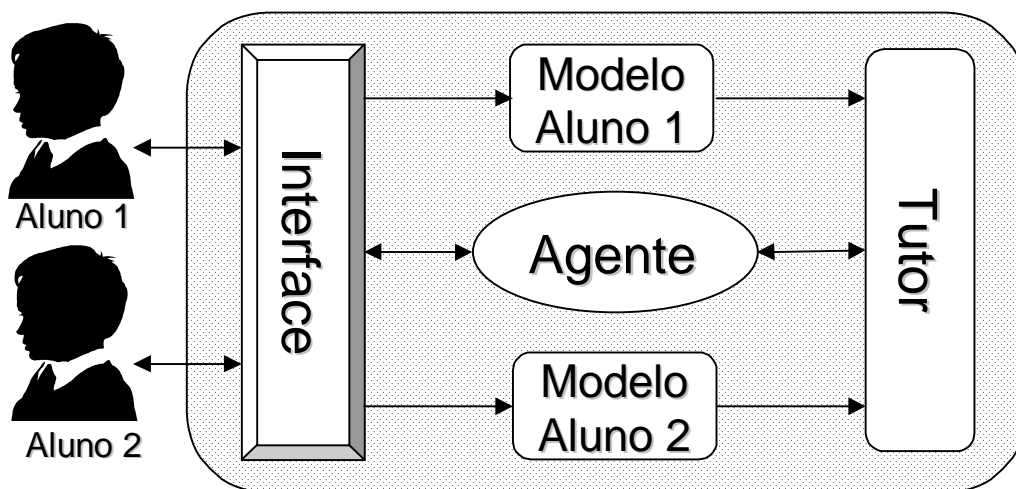


Figura 4-1: Arquitetura proposta

Sendo assim, de forma geral as propriedades deste agente podem ser definidas como:

- Autonomia: os agentes escolhem a ação a tomar baseados na própria experiência;
- Habilidade social: os agentes possuem um elo de comunicação com o tutor e/ou outros agentes;

- Reatividade: os agentes reagem aos estímulos recebidos dos outros agentes (tutor e/ou aluno) e do ambiente.
- Pró-atividade: os agentes monitoram o ambiente e continuamente passa as informações para o tutor sobre o estado do ambiente e do aluno.

O agente é então capaz de responder a estímulos do ambiente e exibir um comportamento orientado a objetivos, pensando no futuro, antecipando e agindo em função de suas previsões. Com a ajuda deste agente, transformações que ocorrem no ambiente e não são percebidas pelo aluno (ou não devidamente compreendidas) são repassadas ao tutor, que então irá selecionar as estratégias e táticas adequadas.

Um exemplo da monitoração do ambiente que pode ser realizada pelo agente é a identificação de que os níveis de energia ou elementos do cenário estão em situação crítica. Estas informações são atualmente observadas somente quando o aluno interage com o sistema.

Uma vez feita à deliberação por parte do tutor, ele passa a executar um plano, e este plano está associado com a escolha de determinadas táticas (enviar mensagem, sinal sonoro, etc.). O tutor envia ao agente o tipo de tática escolhida e cabe ao agente viabilizá-la junto ao aluno.

Em resumo, o tutor fica com as atividades que lhes são mais pertinentes, tais como seleção da estratégia e o conjunto de táticas associadas. Esta seleção ocorre em função das informações enviadas pelo agente mediador. Cabe ao agente monitorar as ações do aluno, os elementos do jogo e organizá-las para enviar ao tutor.

Ao tutor são atribuídas as tarefas de:

- modelar o aluno (caso não seja delegada a um agente);
- deliberar sobre as estratégias e táticas a serem utilizadas com o(s) alunos(s), ou seja, como ajudar;
- enviar as táticas a serem aplicadas pelo agente mediador.

Ao agente são delegadas as tarefas de:

- capturar (coletar) informações do ambiente resultantes das ações do aluno;

- realizar ajustes na interface;
- viabilização no nível de apresentação das decisões do tutor (lembrar que os alunos podem atuar em máquinas diferentes e até mesmo em espaços diferentes).

Esta divisão de tarefas aumenta o potencial exploratório do sistema (em nível pedagógico) e converge na busca de uma arquitetura multiagente de STI para proporcionar auxílio individualizado e mais personalizado ao aluno.

5 Referências Bibliográficas

- [AIM96] Aimuer, E.; Frasson, C. Analysing a new learning strategy according to different knowledge levels. *Computer Education*, Londres, v.27, n.2, 1996.
- [BOF00] Boff, Elisa; Giraffa, Lucia M. M. Construindo um ambiente de ensino-aprendizagem cooperativo: uma experiência interdisciplinar. Em: SBIE 2000 - SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2000, Maceió, Alagoas. Anais. . . 2000. p.112–119.
- [BON91] Bonar, J. G. Interface architectures for intelligent tutoring systems, 1991, p.35–68.
- [CAB00] Cabral, Anderson R. Y. Giraffa, Lucia M. M. Wbt-builder - um ambiente para autoria de WBT. Em: SBIE 2000 - SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2000, Maceió, Alagoas. Anais. . . 2000. p.154–159.
- [CAR70] Carbonell, J. R. AI in CAI: an artificial intelligence approach to computer assisted instruction. *IEEE Transactions on Man Machine Systems*, v.11, n.4, p.190–202, 1970.
- [COS95] Costa, E. B.; Lopes, M. A.; Ferneda, E. Mathema: A learning environment based on a multi-agent architecture. In: SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICIAL, 1995, Berlin. Anais. . . Springer Verlag, 1995.

v.991, p.141–150.

- [ETZ94] Etzioni, O.; Weld, D. A softbot-based interface to the internet. *Communications of the ACM*, v.37, n.7, p.72–76, 1994.
- [FER00] Fernandes, C. T.; Lima, M. D. De Alecar; Zuanásbar, D. H. M. Adeche - ambiente de apoio ao desenvolvimento de cursos hiperídia na web. Em: SBIE 2000 - SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2000, Maceió, Alagoas. Anais. . . 2000. p.24–31.
- [FRA97] Frasson, C.; Mengelle, T.; Aimeur, E. Using pedagogical agents. In: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 1997, kobe, Japão. Anais. . . 1997.
- [GIR97] Giraffa, L. M. M. Seleção e adoção de estratégias de ensino em sistemas tutores inteligentes. Porto Alegre: CPGCC/UFRGS, 1997. Exame de qualificação.
- [GIR98] Giraffa, Lucia M. M.; Viccari, Rosa M. Estratégias de ensino em sistemas tutores inteligentes modelados através de agentes. In: SBIE 1998 - SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 1998, Fortaleza ,Ceará. Anais. . . 1998.
- [GIR99] Giraffa, L. M. M.; Viccari, R. M. Intelligent tutoring systems built using agents techniques. *Revista de Educação, Ciência e Cultura*, Canoas: La Salle, p.23–40, 1999.
- [GIR99] Giraffa, L. M. M. Uma arquitetura de tutor utilizando estados mentais. Porto Alegre: CPGCC/UFRGS, 1999. Tese de Doutorado.
- [GOU00] Goulart, Rodrigo R. V.; Giraffa, Lucia M. M. Uma proposta de agente de interface para ambientes de ensino-aprendizagem. Porto Alegre:, 2000. Plano de Estudos e Pesquisa.
- [GOU99] Gouarderes, G.; Francoise Canut, C. M.; Sanchis, E. The SYS- TEMION: a new agent model to design intelligent tutoring system. IOS Press, 1999. p.54–63.
- [KAW00] Kawasaki, Evelise I.; Omar, Nizam; Fernandes, Clovis T. Um modelo de sistema de tutoria inteligente baseado em princípios pedagógicos para educação de

adultos. Em: SBIE 2000 - SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2000, Maceió, Alagoas. Anais. . . 2000. p.154–159.

- [KON95] Konzen, A. A. Uma estratégia de ensino híbrida para sistemas tutores inteligentes. Em: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 1995, Florianópolis - Brasil. Anais. . . Imprensa Universitária da Universidade Federal de Santa Catarina, 1995. p.327–338.
- [LAS94] LASHKARI, Y.; METRAL, M.; MAES, P. Collaborative interface agents. Em: AAI, 1994. Anais. . . 1994.
- [LIE97] Lieberman, H. Autonomous interface agents. Em: CHI, 1997. Anais. . . 1997. v.1, p.67–74.
- [LOS00] Los Arcos, J. L.; Muller, W.; Fuente, O.; Orúe, L.; Arroyo, E.; Leaznibarrutia, I.; Santander, J. LAHYSTOTRAIN – Integration of Virtual Environments and ITS to Surgery Training. Em: ITS 2000 – Intelligence Tutoring Systems, 2000, Montreal, Canada. Anais. . . 2000. p.43–52.
- [MAR00] Marietto, Maria G. B.; Ornar, Nizam. Definição dinâmica de estratégias instrucionais em sistemas de tutoria inteligentes: uma abordagem multiagentes na WWW. In: SBIE 2000 - SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2000, Maceió, Alagoas. Anais. . . 2000. p.154–159. 22
- [MIT00] Mitrovic, A.; Suraweera, P. Evaluating an Animated Pedagogical Agent. Em: ITS 2000 – Intelligence Tutoring Systems, 2000, Montreal, Canada. Anais. . . 2000. p.73–82.
- [NET00] Neto, Hermínio B.; Filho, Raimir H.; Silva, Wilker B.; Braga, Oscar S. G. Especificando o tele-ambiente no contexto da educação a distância. Em: SBIE 2000 - SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2000, Maceió, Alagoas. Anais. . . 2000. p.154–159.
- [PER00] Pereira, Adriana S.; Geyer, Claudio F. R. Um Agente para Seleção de Estratégias de Ensino em Ambientes Educacionais na Internet. Em: IBERAMIA – SBIA 2000, 2000, Atibaia, São Paulo. Anais. . . 2000. p.362-369.

- [ROS99] Rosatelli, M. C. Um ambiente inteligente para aprendizado colaborativo no ensino a distância utilizando o método de casos. Florianópolis: PPEP/UFSC, 1999. Tese de Doutorado.
- [SEL99] Self, John. The defining characteristics of intelligent tutoring systems research: itss care, precisely. International Journal of Artificial of Artificial Intelligence in Education, Leeds, England, 1999.
- [SIL00] Silveira, R. A.; Bica, F.; Viccari, R. M. JADE – Java Agents for Distance Education framework. Em: Iberoamerican Workshop on Distributed Artificial Intelligence and Multi-Agent Systems, 2000, Atibaia, São Paulo. Anais. . . 2000. p.112–122.
- [TED00] Tedesco, P. A.; Self, J. Using Meta-cognitive Conflicts to support Group Problem Solving. Em: ITS 2000 – Intelligence Tutoring Systems, 2000, Montreal, Canada. Anais. . . 2000. p.112–119.
- [WEN87] WENGER, E. Artificial intelligence and tutoring systems. [S.l.]: Morgan Kaufmann Publishers, INC, 1987. p.486.
- [WEX97] Wexelblat, A.; Maes, P. Issues for software agent ui. Disponível por WWW em <http://wex.www.media.mit.edu/people/wex/agent-ui-paper/agentui.htm> (1997).
- [ZEV00] Zeve, Carlos M. D.; Sloczinski, Helena; Polonia, Eunice; Nitz- Ke, Julio A.; Lima, José V. Aprendizagem colaborativa: a utilização de cd-rom e internet em um sistema integrado. In: SBIE 2000 - SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2000, Maceió, Alagoas. Anais. . . 2000. p.66–72. 23
- [ZOU00] Zouaq, A.; Frasson, C.; Rouane, K. The Explanation Agent. Em: ITS 2000 – Intelligence Tutoring Systems, 2000, Montreal, Canada. Anais. . . 2000. p.112–119.