

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA DA COMPUTAÇÃO**

Marcel Alexandre Cabral Ribas

**UMA REPRESENTAÇÃO MULTI-AGENTES
PARA O DOMÍNIO EM UM
SISTEMA EDUCACIONAL VIA WEB**

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação

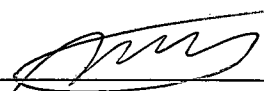
**Prof. Fernando Alvaro Ostuni Gauthier, Dr.
Orientador**

Florianópolis, 19 de dezembro de 2000.

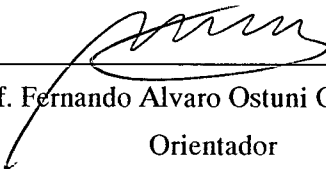
UM REPRESENTAÇÃO MULTI-AGENTES PARA O DOMÍNIO EM SISTEMAS EDUCACIONAIS VIA WEB

Marcel Alexandre Cabral Ribas

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação Área de Concentração Sistemas de Conhecimento e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.




Prof. Fernando Alvaro Ostuni Gauthier, Dr.
Coordenador da Pós-Graduação em Ciências da Computação

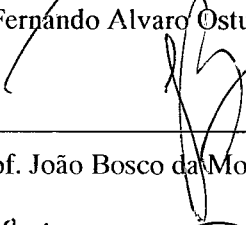


Prof. Fernando Alvaro Ostuni Gauthier, Dr.
Orientador

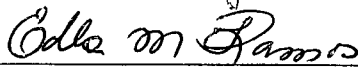
Banca Examinadora



Prof. Fernando Alvaro Ostuni Gauthier, Dr.



Prof. João Bosco da Mota Alves, Dr.



Profa. Edla Maria Faust Ramos, Dra.

Agradecimentos

A Maíra, que foi o porto seguro durante as tempestades de falta de confiança, pelo seu infinito amor e compreensão.

A Salomão, Chola e Ricardo, minha família querida.

Às Comunidades de Vida Cristã, que me ensinaram a sempre confiar no Pai e a “em tudo amar e servir”. Em especial à Comunidade Magis, que partilhou comigo cada alegria e tristeza, e à Pré-Comunidade Santo Estanislau Kostka, que perseverou na caminhada mesmo sem seu assessor, mostrando o quanto vale a pena estar com eles.

A Fernando Gauthier, meu orientador, que nunca desistiu de mim.

Aos amigos Bosco, Dulce e Simone, que sempre acreditaram no que eu era capaz.

Aos colegas da Dualline, que seguraram a barra e as responsabilidades enquanto eu estava trabalhando neste Mestrado.

Aos músicos da banda CODA, meus irmãos, que entenderam a missão que eu teria que cumprir e esperaram pacientemente pelo término dela.

Aos alunos de Engenharia de Telemática da UNISUL, e ao seu Coordenador Youssef, que tanto me ensinaram a ensinar, por sua compreensão e amizade.

Sumário

Agradecimentos	i
Sumário	ii
Lista de Abreviaturas.....	vi
Lista de Figuras	vii
Lista de Tabelas	ix
Resumo	x
Abstract	xi
1 Introdução	1
1.1 Considerações Iniciais	1
1.2 Objetivos.....	2
1.2.1 Objetivo geral	2
1.2.2 Objetivos específicos.....	2
1.3 Justificativa.....	2
1.4 Limitações	6
1.5 Estrutura do Trabalho	6
2 Sistemas Tutores Inteligentes	7
2.1 Definição	7
2.2 Componentes de um sistema tutor inteligente.....	7
2.2.1 Módulo do Aprendiz.....	8
2.2.2 Módulo do Domínio	9
2.2.3 Módulo de Interface	9
2.2.4 Módulo Tutor.....	10
2.3 Modelos de Estudante.....	10
2.3.1 Modelo Overlay	11
2.3.2 Modelos de erro (Buggy).....	12
2.3.3 Modelos aprendizes	13
2.3.4 Modelos baseados em restrições	13
2.3.5 Modelos interativos	14
2.4 Características de sistemas tutores inteligentes	14
2.4.1 Sequência adaptativa de currículo	15
2.4.2 Suporte à resolução de problemas	15
2.5 Sistemas Tutores Hipermissão Inteligentes.....	16
3 Hipermissão Adaptativa	17
3.1 Introdução	17
3.2 Conceitos	17
3.2.1 Hipermissão	17
3.2.2 Hipermissão adaptativa	18
3.3 Aplicações	18

3.4	Tipos de hipermídia adaptativa	19
3.4.1	Apresentação adaptativa	20
3.4.2	Suporte à navegação adaptativa.....	20
3.4.3	Projetos de hipermídia adaptativa	21
3.4.4	Eficácia da hipermídia adaptativa	24
3.5	Modelos de Usuário.....	25
3.5.1	Modelo Superposição (Overlay).....	25
3.5.2	Modelo Estereótipo	26
3.6	Hipermídia Adaptativa na Educação via Web.....	26
4	Sistemas Educacionais na Web	28
4.1	Introdução.....	28
4.2	Aspectos Pedagógicos	28
4.2.1	Teorias de aprendizado.....	29
4.2.1.1	Comportamentalismo	29
4.2.1.2	Cognitivismo	29
4.2.1.3	Construtivismo	30
4.2.2	Estilos de Aprendizado.....	30
4.2.2.1	MBTI (Myers-Brigs Type Indicator).....	31
4.2.2.2	Modelo de aprendizado de Kolb.....	32
4.2.2.3	HBDI (Herrmann Brain Dominance Instrument).....	32
4.2.2.4	Modelo de aprendizado de Felder-Silverman.....	33
4.3	Design Instrucional.....	34
4.4	Aprendizado pela Web	35
5	Sistemas Multi-Agentes.....	37
5.1	Agentes Inteligentes	37
5.1.1	Definições e conceitos gerais	37
5.1.2	Principais Propriedades de um Agente Inteligente.....	37
5.1.2.1	Autonomia	38
5.1.2.2	Comunicabilidade.....	38
5.1.2.3	Cooperação	38
5.1.2.4	Raciocínio.....	39
5.1.2.5	Comportamento Adaptativo	39
5.1.2.6	Confiabilidade	39
5.1.2.7	Antropomorfismo	40
5.1.3	Classificação de Agentes	40
5.1.3.1	Agentes Inteligentes	40
5.1.3.2	Agentes Móveis	41
5.1.3.3	Agentes Autônomos	41
5.1.3.4	Agentes Coordenados.....	41
5.1.3.5	Agentes Aprendizes e Adaptativos.....	41
5.1.3.6	Agentes Reativos	41
5.1.3.7	Agentes Cognitivos	42
5.2	Sistemas Multi-Agentes.....	42
5.2.1	Conceito.....	42
5.2.2	Características de sistemas multi-agentes	42
5.2.2.1	Cooperação	42
5.2.2.2	Coordenação	42
5.2.2.3	Comunicação	43

5.2.3	Arquiteturas de Sistemas Multi-Agentes.....	44
6	Um Modelo de Representação do Domínio de Conhecimento	45
6.1	Introdução.....	45
6.2	A Composição de um Domínio de Conhecimento.....	46
6.3	Organização do Conteúdo	48
6.4	Definição da Plataforma de Testes	50
6.5	Modelagem dos Agentes	51
6.5.1	Características dos agentes	51
6.6	A Linguagem UML	52
6.6.1	Modelo Use-Case	53
6.6.2	Diagramas de interação	54
6.6.2.1	Diagrama de seqüência.....	55
6.6.2.2	Diagrama de colaboração	55
6.6.3	Diagramas de atividade	56
6.6.4	Diagramas de classe	57
6.7	A Extensão AgentUML.....	58
6.7.1	Descrevendo um sistema multi-agentes	58
6.7.1.1	Packages	58
6.7.1.2	Templates	59
6.7.2	Descrevendo as interações entre os agentes	59
6.7.2.1	Diagramas de Seqüência.....	59
6.7.2.2	Diagramas de Colaboração.....	60
6.7.2.3	Diagramas de Atividade	61
6.7.2.4	Gráficos de Estado.....	62
6.7.3	Descrevendo o funcionamento interno dos agentes	63
6.8	Modelo dos Agentes	63
6.9	Inferência e raciocínio dos agentes.....	64
6.9.1	Resultado de testes dinâmicos.....	65
6.9.2	Tempo estudado.....	66
6.9.3	Revisões.....	67
6.9.4	Valor aprendido	68
6.9.5	Cálculo do valor adequação.....	68
6.10	Comunicação entre os agentes.....	69
6.11	Modelagem do Banco de Dados.....	73
7	Implementação do Modelo.....	76
7.1	Introdução.....	76
7.1.1	Fase de Visão.....	76
7.1.2	Fase de Design.....	77
7.1.3	Fase de Desenvolvimento.....	77
7.1.4	Fase de Instalação e Testes.....	78
7.2	Plataforma.....	78
7.3	Ferramentas de Desenvolvimento	79
7.4	Recursos do Sistema.....	80
7.4.1	Funcionalidades.....	80
7.4.2	Limitações	81
7.5	Funcionamento	81
7.6	Desenvolvimento.....	86
7.7	Modelo Físico das Classes.....	86

7.8	Descrição das classes.....	87
7.8.1.1	Classe CAgente	87
7.8.1.2	Classe CALuno	88
7.8.1.3	Classe CTeste	88
7.8.1.4	Classe CCurso	88
7.8.1.5	Classe CXMLParser	88
7.9	Modelo Físico do Banco de Dados.....	88
8	Conclusões.....	90
9	Referências Bibliográficas	95
	ANEXO 1 – Regras para Atualização do Valor Revisão	103
	ANEXO 2 – Regras para Cálculo do Valor Ajustado	104

Lista de Abreviaturas

AH	Adaptive Hypermedia
AUML	Agent Unified Modeling Language
DLL	Dynamic Link Library
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
KQML	Knowledge Query and Manipulation Language
STI	Sistemas Tutores Inteligentes
UML	Unified Modeling Language
WBE	Web-based Education
WWW	World Wide Web
XML	eXtended Markup Language

Lista de Figuras

Figura 1 - Estrutura de um Sistema Tutor Inteligente	8
Figura 2 – Uma representação do modelo de estudante overlay (Fonte: SMITH (2000)).....	12
Figura 3 - Uma representação do modelo perturbação (buggy) (Fonte: SMITH (2000)).....	13
Figura 4 - Tela do ELM-ART	22
Figura 5 - Tela do InterBook	23
Figura 6 - Tela do curso Estruturas e Sistemas Hipermídia	24
Figura 7 - Exemplo de Organização de Conteúdo.....	50
Figura 8 - Exemplo de use case UML (Fonte: STURM (1999)).....	54
Figura 9 - Exemplo de Diagrama de Seqüência (Fonte: STURM (1999)).....	55
Figura 10 - Exemplo de diagrama de colaboração UML (Fonte: STURM (1999)).....	56
Figura 11 - Exemplo de diagrama de atividade UML (Fonte: STURM (1999)).....	57
Figura 12 - Exemplo de diagrama de classe UML	57
Figura 13 - Diagrama de Seqüência AUML (Fonte: ODELL et al. (2000)).....	60
Figura 14 - Mensagens Simultâneas em AUML (Fonte: ODELL et al. (2000)).....	60
Figura 15 - Exemplo de diagrama de colaboração AUML (Fonte: ODELL et al. (2000)).....	61
Figura 16 - Exemplo de diagrama de atividade AUML (Fonte: ODELL et al. (2000))	62
Figura 17 - Exemplo de Gráfico de Estado AUML (Fonte: ODELL et al. (2000)).....	63
Figura 18 – Diagrama Lógico de Classes.....	64
Figura 19 - Curva para cálculo do Escore de Tempo Estudado	67
Figura 20 - Diagrama AUML de seqüência da seleção do próximo tópico	70
Figura 21 - Diagrama Gráfico de Estado do Agente Teste	71
Figura 22 - Processo de ajuste do valor adequação	71
Figura 23 - Processo de seleção do melhor tópico	72
Figura 24 - Modelo lógico do banco de dados	74
Figura 25 - Tela de login	82
Figura 26 - Exemplo de página de conteúdo	83
Figura 27 - Teste dinâmico.....	84
Figura 28 - Tela de respostas de teste.....	85
Figura 29 - Exemplo de tela de feedback	86

Figura 30 - Diagrama físico das classes do protótipo.....87
Figura 31 - Modelo físico do banco de dados89

Lista de Tabelas

Tabela 1 - Valores necessários por tipo de link.....	69
--	----

Resumo

O uso da Internet como meio de transferência de conhecimento vem crescendo a cada dia, e há um número de grupos que estudam formas de tornar o ensino pela Web cada vez mais atraente e eficaz.

Muitos educadores esbarram na inadequação das tecnologias em relação a teorias pedagógicas avançadas, pois hoje ainda são poucas as iniciativas de instrução pela Web que promovem um verdadeiro aprendizado.

Na busca por tecnologias que dêem maior flexibilidade aos educadores e permitam o emprego de estratégias modernas de ensino, o presente trabalho tem por objetivo propor uma arquitetura multi-agentes para a representação do domínio de sistemas educacionais via Web.

Faz-se uma análise das pesquisas recentes sobre sistemas tutores inteligentes, sistemas de hipermídia adaptativa e sistemas multi-agentes, para depois integrar estas pesquisas na arquitetura proposta. O modelo proposto, que difere um pouco das abordagens tradicionais de sistemas tutores inteligentes, e sua implementação através de um protótipo são descritos, e são feitas anotações sobre os resultados obtidos.

Por fim, se apresentam conclusões sobre a aplicabilidade do modelo e possibilidades de futuros desenvolvimento no sentido da criação de ambientes favoráveis ao aprendizado.

Palavras-chave:

Sistemas tutores inteligentes, hipermídia adaptativa, sistemas multi-agentes

Abstract

The use of the Internet as a way of knowledge transfer is growing day after day, and there are a number of groups studying ways to make the process of learning through the Web more attractive and efficient.

Many educators get stuck in the inadequacy of the technologies regarding pedagogical theories, because there aren't many Web instructional ventures that really promote learning.

In the quest for technologies that give more flexibility to the educators and allow for the employment of modern teaching strategies, the present work aims to propose a multi-agent architecture for the representation of the domain in Web educational systems.

We perform an analysis of the recent research efforts on intelligent tutoring systems, adaptive hypermedia systems and multi-agent systems, to integrate them in the proposed architecture.

The proposed model, that differs a little from the traditional approaches of intelligent tutoring systems, and its implementation in a prototype are described, and we make some notes on the results.

Then, we present conclusions on the applicability of the model and possibilities of future development towards the creation of learning environments.

Keywords

Adaptive hypermedia, intelligent tutoring systems, multi-agent systems, web-based education

1 Introdução

1.1 Considerações Iniciais

Faz algum tempo que o ensino à distância pela Internet se tornou uma realidade. Várias empresas e universidades estão disponibilizando disciplinas e até mesmo cursos inteiros pela Internet. É claro que estes cursos não têm como objetivo substituir as salas de aula em todo o seu poder. São apenas alternativas para que as pessoas que não têm tempo ou que estão distantes geograficamente dos centros de ensino possam aprender.

A Internet, ou melhor, a World Wide Web, se constituiu numa ótima forma de disponibilizar conteúdo para esses alunos. Munidos de um computador e um simples browser, eles podem ter acesso a cursos em qualquer lugar do mundo, 24 horas por dia, 7 dias por semana.

Os educadores modernos já consideram a Internet como uma forma concreta de obtenção de conhecimento, e já são bastante utilizadas suas facilidades de disponibilização de conteúdo multimídia, trabalho colaborativo e workflow no ensino.

A primeira forma de ensino pela Internet a ser utilizada, e até hoje a mais utilizada, foi o aprendizado assíncrono, que consiste em conteúdo disponibilizado em servidores Web, para que alunos possam acessá-lo individualmente. Este tipo de ensino, mesmo não sendo completo, pôde ultrapassar barreiras e alcançar pessoas que antes não poderiam angariar tal conhecimento.

Pessoas que vivem distantes dos centros de conhecimento podem acessar o material criado pelos educadores destes centros. Assim também pessoas que trabalham

durante todo o dia, que tinham dificuldades em assistir aulas durante o dia, podem acessar este conteúdo nos horários mais adequados para elas.

É cada vez mais importante o papel da Internet no ensino e no aprendizado.

1.2 Objetivos

1.2.1 Objetivo geral

Estudar e sugerir um mecanismo de organização de um curso pela Web, utilizando as tecnologias de sistemas multi-agentes, visando o estabelecimento de uma plataforma para um sistema educacional adaptativo via Web.

1.2.2 Objetivos específicos

Fazer um levantamento do estado da arte em sistemas multi-agentes e sua utilização em sistemas educacionais.

Verificar a factibilidade de uma representação do conteúdo de um curso via Web através de um sistema multi-agentes.

Propor um mecanismo que a partir de um conteúdo organizado, possa dar suporte a um tutor que guie o aluno em um sistema educacional adaptativo via Web.

Implementar um protótipo de um curso via Web que utilize a arquitetura proposta.

1.3 Justificativa

Uma das grandes virtudes dos cursos eletrônicos é justamente a capacidade do aluno em navegar em seu próprio ritmo, o que não se consegue na sala de aula, onde os alunos têm que acompanhar o professor. Num curso eletrônico, individual, o aluno pode

voltar atrás e reler algum trecho que não ficou claro. A grande desvantagem desse tipo de curso é que, ao contrário da sala de aula, o aluno não tem ninguém ali para quem ele possa perguntar, tirar alguma dúvida.

O que aconteceu com a Web é que muitos professores simplesmente transformaram seus textos em HTML, a linguagem usada para as páginas Web, e colocaram estes arquivos num servidor Web. Tem-se uma reedição do que aconteceu quando o CD-ROM tornou-se popular. Dizia-se que era uma revolução, que a indústria literária iria morrer, e que o CD-ROM iria substituir o livro como ele é conhecido hoje. O tempo passou, e aquele afã tecnológico também. Logo viu-se que ler numa tela é muito mais desagradável do que ler no papel.

Colocar conteúdo didático na Internet tinha como objetivo ativar o ensino, animar os alunos com elementos gráficos e sonoros do computador, recursos que num livro não era possível se ter. Mas muitos sites ficaram só nisso: verdadeiras “apostilas eletrônicas”, que apenas substituem a chatice de se ler um livro pela chatice de ler páginas numa tela de computador. Alguns sites pelo menos fornecem conteúdo interativo, que espera respostas do usuário, e utilizam animações e filmes.

Só que o que poucos perceberam foi que o conteúdo de cada curso pela Internet é exatamente o mesmo e apresentado da mesma forma para todos os alunos. Várias teorias mostram que cada pessoa aprende de forma diferente. Uns entendem melhor quando lêem, outros quando fazem exercícios, outros através de gráficos, outros com exemplos, apenas para citar algumas formas de apresentação de conteúdo.

Além disso há o fato que os alunos dificilmente entram numa disciplina nivelados. O ponto de partida de cada um depende de experiências passadas, de conhecimento prévio, de uma série de coisas. Numa sala de aula, isso sempre causa um problema para o professor. Se ele ensina a parte básica, os alunos mais adiantados ficam aborrecidos de

estar vendo aquilo tudo de novo. Se ele pula uma parte básica, os alunos menos avançados não irão entender muito bem e podem ter dificuldades no futuro. Cabe ao professor encontrar uma média, um ritmo no qual ele consiga atender às necessidades de todos.

É possível aproveitar a individualidade do ambiente Web para abordar estes dois problemas. É necessário que um curso à distância tenha essa capacidade, como um professor faz na sala de aula, de perceber um ritmo que o aluno acompanhe, e até mesmo de mudar a ordem dos assuntos ou o tipo de apresentação de acordo com cada aluno.

Aí entra em cena a inteligência do sistema. Um sistema inteligente de ensino à distância deve dar suporte à personalização. Deve acompanhar o aluno em sua navegação e de alguma forma perceber seu entendimento das coisas para então guiá-lo em seu aprendizado.

Nesse sentido, percebe-se que é muito importante a organização do conteúdo de um curso, ou seja, o domínio de ensino/aprendizado. Esse conteúdo é frequentemente apenas uma seqüência de páginas Web, ou seja, linear, organizado a partir de um índice. Isto é muito pouco para dar o suporte àquela personalização comentada anteriormente. Cada assunto, ou elemento de conteúdo, pode ser relacionado com os demais. Pode ser que para um bom entendimento de um assunto, um ou mais assuntos anteriores tenham que estar também bem entendidos. Um sistema inteligente deveria informar isso para o aluno, e sugerir que ele volte atrás e entenda os pré-requisitos antes de partir para o próximo assunto.

Outro fator fundamental é a capacidade de acompanhar individualmente cada aluno, manter um histórico de seus acessos, e poder avaliar o grau de conhecimento

sobre cada elemento de conteúdo, para poder sugerir os passos seguintes ao aluno, ou seja, guiá-lo pelo domínio.

São necessários estudos e pesquisas no sentido de desenvolver sistemas que apresentem de forma personalizada as páginas Web que aparecem para o usuário. Em sistemas que levem isso em consideração, cada aprendiz poderá ver o conteúdo de um jeito diferente, sem precisar perder tempo com assuntos que já domina, nem ficar sem entender algum assunto por falta de conhecimento prévio. Assim, espera-se sistemas de ensino mais agradáveis, que possam fazer com que o entusiasmo do aluno seja maior, e conseqüentemente, seu aprendizado seja mais eficiente.

Existem esforços nesse sentido, que empregam técnicas de inteligência artificial como agentes inteligentes, sistemas especialistas, raciocínio baseado em casos, redes Bayesianas, redes neurais, entre outras. A maioria deles se refere aos Sistemas Tutores Inteligentes, que vieram antes da popularização da Internet, e tenta adaptar, a eles e sua teoria, para a Web. Por isso, mesmo os trabalhos que envolvem agentes inteligentes utilizam o modelo de estudante definido na teoria de Sistemas Tutores Inteligentes.

Acontece que dentro de um domínio, existem diferenças entre os conceitos envolvidos. Dependendo do contexto, uns tópicos são mais importantes que os outros. As representações que consideram o módulo de domínio de forma monolítica não dão suporte a estas particularidades. Por isso, este trabalho propõe uma nova representação do estudante e do domínio de aprendizado utilizando um sistema de agentes inteligentes, que possa aproveitar melhor as características intrínsecas de cada tipo de domínio, e possa prover uma plataforma flexível para desenvolvimento de técnicas de ensino e tutoria pela Web.

1.4 Limitações

Não é o objetivo deste trabalho desenvolver um sistema “inteligente” no sentido acadêmico da palavra, mas sim apresentar uma proposta de plataforma para a construção de um sistema educacional adaptativo via Web. Por isso o raciocínio e aprendizado do sistema é limitado a algumas regras, para somente demonstrar a viabilidade do modelo.

Também não foi empregado o uso de protocolos de comunicação entre os diversos agentes, pois eles foram todos implementados numa mesma linguagem seguindo um mesmo padrão.

Para se inserir um curso novo nesta plataforma, é necessário entrar no banco de dados manualmente, ou seja, em primeira instância não será desenvolvida uma interface amigável para os educadores colocarem seu conteúdo no sistema.

1.5 Estrutura do Trabalho

O trabalho que se segue está apresentado em nove capítulos. O segundo capítulo faz uma revisão bibliográfica da área de sistemas tutores inteligentes. O terceiro capítulo apresenta os sistemas de hipermídia adaptativa. O quarto capítulo discorre sobre os sistemas educacionais via Web. O quinto capítulo relata as pesquisas na área de sistemas multi-agentes. O sexto capítulo propõe um modelo para a representação do domínio em um sistema educacional via Web. O sétimo capítulo mostra a implementação do modelo e os resultados obtidos. O oitavo capítulo traz as conclusões da pesquisa e sugestões para pesquisas futuras. E por fim, o nono capítulo lista as referências bibliográficas utilizadas.

2 Sistemas Tutores Inteligentes

2.1 Definição

Um Sistema Tutor Inteligente (STI) é um sistema computacional complementar para ensino, que procura modelar os aspectos envolvidos na tutoria humana. (MARIETTO et al., 1997)

Alguns autores referem-se a tais tipos de sistemas como ICAI (Intelligent Computer Aided Instruction) ou ILS (Intelligent Learning Systems).

De acordo com ZHOU (2000), este campo envolve áreas de pesquisa multidisciplinares, como inteligência artificial, teoria da educação, psicologia, ciência cognitiva, e teorias de interação homem-computador.

Parte-se do princípio que a tutoria humana, do tipo um-para-um, de acordo com o demonstrado por (BLOOM, 1984 apud BECK, 1997) é muito mais eficiente do que o ensino tradicional em sala de aula. Os sistemas tutores inteligentes procuram simular tal experiência através do computador ou da Internet.

A pesquisa no campo de STI começou nos anos 70. Desde então vários foram criados, mas poucos estão em uso real. (KHUWAJA, 1994)

2.2 Componentes de um sistema tutor inteligente

Um STI é um programa, um software, e sua finalidade é engajar o aluno em uma atividade instrucional. KHUWAJA (1994) afirma que um STI pode ser projetado através de duas visões: a visão conceitual, que se preocupa com a capacidade de tutoria,

relacionando-a com experiências naturais de tutoria, e a visão de sistema, que se preocupa com a organização como um sistema computacional.

Utilizando a visão de sistema, pode-se estruturar um STI em módulos, como qualquer sistema computacional. Embora alguns autores diverjam no número de módulos, a estrutura básica de um STI adotada na literatura é a seguinte:

- Módulo do Aprendiz
- Módulo do Domínio
- Módulo de Interface
- Módulo Tutor

A Figura 1 ilustra a visão de sistema de um STI.

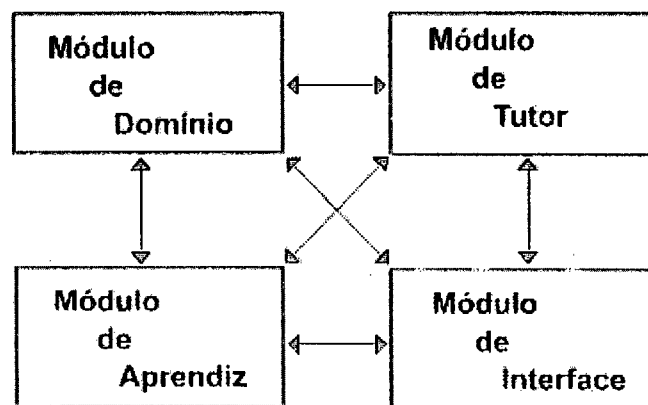


Figura 1 - Estrutura de um Sistema Tutor Inteligente

2.2.1 Módulo do Aprendiz

Mantém as informações sobre o usuário-aluno que utiliza o sistema, para que seja possível oferecer as funções de tutor. É o que o sistema percebe do conhecimento adquirido pelo aluno sobre o domínio.

O módulo de aprendiz, que BURNS & CAPPS (1988) apud BRANDLE (1998) chamam de módulo do estudante, é responsável por construir um modelo do que o aluno sabe e é capaz de fazer.

COSTA (1999) afirma que no mínimo o módulo de aprendiz deve manter um histórico sobre como o estudante está trabalhando no material em questão. E afirma ainda que é interessante que ele mantenha também registro sobre os erros do estudante, para que alimentar com a maior quantidade de informações possível o módulo do especialista.

2.2.2 Módulo do Domínio

Mantém o conteúdo que será transmitido, a estrutura deste conteúdo, e as ligações e conexões entre os assuntos componentes deste conteúdo.

BURNS & CAPPS (1988) apud BRANDLE (1998) chama este módulo de módulo do especialista, e diz que ele contém informações sobre o domínio do problema (a base de conhecimento) e pode realizar várias operações sobre este domínio, tais como resolver problemas.

Alguns autores, como COSTA (1999) separam este módulo em módulo do domínio e módulo do especialista.

2.2.3 Módulo de Interface

Gerencia a forma de apresentação do conteúdo ao aluno. Tem como objetivo tornar o sistema atrativo, através de componentes visuais que facilitem a compreensão e a integração entre o aluno e o computador.

Segundo BURNS & CAPPS (1988) apud BRANDLE (1998) é o ambiente de trabalho que mantém a comunicação real com o usuário.

2.2.4 Módulo Tutor

É a parte inteligente, que pode utilizar-se dos outros módulos, principalmente o de aprendiz e o do domínio, para guiar o estudante pelo domínio e tornar o aprendizado mais eficaz e agradável.

É um sistema especialista no domínio pedagógico, que planeja as sessões em vários níveis de detalhe e cuida de outras funções de ensino/aprendizado. (BURNS & CAPPS (1988) apud BRANDLE (1998).

Tem a responsabilidade de adequar as interações entre o sistema e o aluno de acordo com o modelo que o módulo do aprendiz vem construindo e com as técnicas pedagógicas que nele foram alimentadas.

2.3 Modelos de Estudante

Os Sistemas Tutores Inteligentes precisam manter informações dinâmicas sobre o usuário, o aluno, para que possam se adaptar ou fornecer apoio ao seu aprendizado. Chama-se a este conjunto de informações *modelo do estudante*.

PAIVA et al. (1995) definem o modelo do estudante como “algumas características e atitudes dos estudantes, que são úteis para alcançar a interação adequada e individualizada estabelecida entre os ambientes computacionais e os estudantes”.

MARIETTO et al. (1997) destacam os modelos de estudante do tipo Overlay (sobreposição) e Buggy (perturbação). Citam ainda o modelo Estereótipo, que é um modelo de usuário descrito no Capítulo 3.

Segundo MARIETTO et al. (1997), a maioria dos STIs desenvolvidos utiliza o modelo de Superposição ou o de Perturbação. Ambos são construídos em cima do

modelo do especialista, adicionando-se informações que os diferenciam. Isto se dá ou pela consideração das concepções ausentes, ou das concepções incorretas.

KING (1998) realizou um estudo, tentando unificar a classificação de modelos de estudante. Depois de avaliar as metodologias de classificação propostas por Brusilovsky, Elsom-Cook e Djordjevi-Kajan et al., King conseguiu classificar os modelos de estudante em cinco categorias:

- modelos overlay
- modelos de erro
- modelos aprendizes
- modelos baseados em restrições
- modelos interativos

2.3.1 Modelo Overlay

O modelo overlay é a abordagem mais popular. É quando o modelo do estudante representa um subconjunto do conhecimento de um especialista. Não leva em conta os erros ou concepções erradas do estudante. KING (1998) diz que neste modelo, considera-se que o estudante não pode desenvolver unidades de conhecimento além das existentes no conhecimento do especialista.

A Figura 2 mostra como pode-se imaginar um modelo overlay.

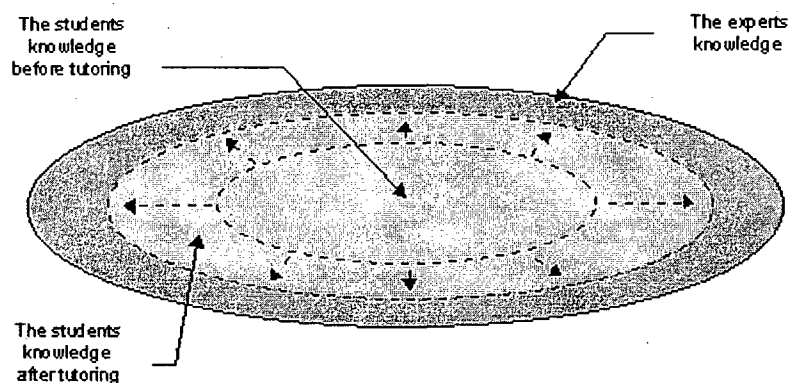


Figura 2 – Uma representação do modelo de estudante overlay (Fonte: SMITH (2000))

Sua implementação pode ser simples, por exemplo, atribuindo um nível de conhecimento a cada unidade de conhecimento, indicando se o estudante sabe ou não aquele elemento, ou complexo, se tenta determinar a extensão do conhecimento adquirido sobre uma unidade ou lição em especial. A avaliação deste conhecimento pode ser através de uma medida escalar, tal como uma probabilidade, ou através de níveis de corte (thresholds) como o sugerido por BRUSILOVSKY (1994b).

2.3.2 Modelos de erro (Buggy)

Quando o modelo representa tanto as concepções ausentes quanto as incorretas, ele é chamado de buggy, ou perturbação.

KING (1998) ainda divide os modelos de erro em duas sub-categorias: os modelos enumerativos e os modelos reconstrutivos. Os modelos enumerativos modelam o conhecimento correto assim como concepções falsas e erros comuns. Os modelos reconstrutivos determinam as concepções falsas quando um estudante aplica operadores impropriamente durante uma tarefa de uma lição.

Os modelos enumerativos dependem de uma grande base de conhecimento, de uma biblioteca de erros, para funcionar corretamente. Já os modelos reconstrutivos,

determinando os erros ou concepções falsas em tarefas procedurais, evitam a necessidade dessa biblioteca.

A Figura 3 mostra um exemplo de representação gráfica do modelo de perturbação.

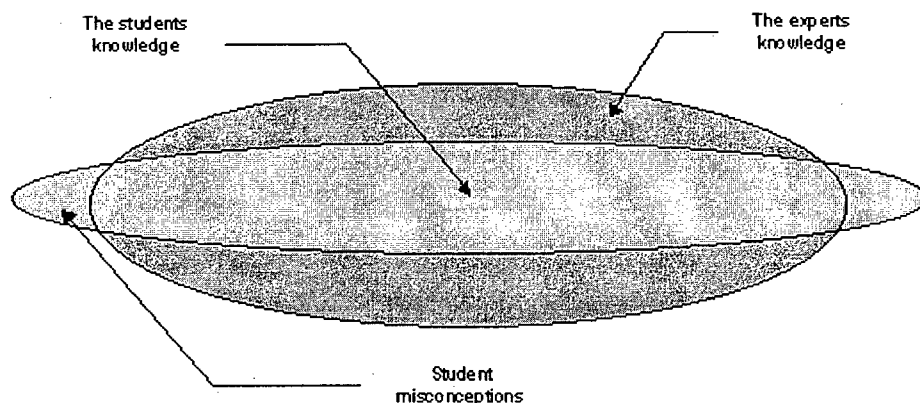


Figura 3 - Uma representação do modelo perturbação (buggy) (Fonte: SMITH (2000))

2.3.3 Modelos aprendizes

Os modelos aprendizes tentam explicar os erros no conhecimento do estudante em termos de seu processo gerador. São baseados na idéia de que como os estudantes desenvolvem conhecimento ao longo do tempo, eles podem começar gradualmente a compor concepções falsas.

Podem ser implementados utilizando diversas abordagens, mas a maioria das implementações utilizam técnicas de inteligência artificial e aprendizado de máquina como redes neurais e algoritmos genéticos. (KING, 1998)

2.3.4 Modelos baseados em restrições

É uma abordagem especial proposta por OHLSSON (1996) apud KING (1998) “que explora classes equivalentes de estados de problemas como o recurso primário

para a modelagem do conhecimento do estudante”. As explicações para concepções falsas são determinadas pelo estado do problema no qual o estudante chegou.

2.3.5 Modelos interativos

Este tipo de modelos permite a negociação do modelo de estudante entre o estudante e o sistema tutor inteligente. Um modelo interativo pode ser construído usando alguma das técnicas descritas acima, mas deve considerar diversas outras características de implementação: externalizar o modelo do estudante e permitir a interação do estudante com o modelo do estudante. (KING, 1998)

O ELM-ART , descrito no Capítulo 3, é um exemplo de sistema que permite esta interação. O estudante pode inspecionar o modelo do estudante, que são as concepções que o sistema tem desse estudante, e alterá-lo até um certo ponto.

2.4 Características de sistemas tutores inteligentes

Os sistemas tutores inteligentes podem ter as seguintes funcionalidades:

- Análise inteligente de soluções do estudante
- Suporte interativo à resolução de problemas
- Resolução de problemas baseada em exemplos
- Apresentação adaptativa
- Suporte à colaboração adaptativa
- Seqüência de currículo (ou planejamento instrucional)
- Suporte à navegação adaptativa

As duas características principais de sistemas tutores inteligentes são a seqüência adaptativa de currículo e o suporte interativo à resolução de problemas (WEBER &

SPECHT, 1997). Estas duas diferenciam os sistemas inteligentes de aprendizado da instrução assistida por computador tradicional, no sentido que eles incorporam técnicas inteligentes que professores humanos capacitados usam em aulas ou acompanhando aprendizes individualmente.

2.4.1 Seqüência adaptativa de currículo

A seqüência de currículo descreve a ordem na qual novas unidades de conhecimento e habilidades serão aprendidas e as operações de ensino correspondentes (ex.: apresentar exemplos e demonstrações, fazer perguntas, fornecer exercícios e testes, resolver problemas) são apresentadas a um aprendiz em particular. Em livros, o meio tradicional de aprendizado, o currículo é pré-definido pelo autor. O mesmo acontece com a maioria dos textos na Web. Ou seja, os autores fornecem um caminho bom para aprendizado para um aprendiz genérico.

Num sistema tutor inteligente esta seqüência de currículo pode ser gerada dinamicamente, de acordo com o conhecimento que o aluno demonstra ter sobre o assunto, a seqüência prévia de assuntos visitados, o tempo gasto em cada assunto, etc. Então, de posse destas informações do aprendiz, o chamado “modelo do estudante”, o sistema pode adaptar a apresentação das páginas ao usuário.

2.4.2 Suporte à resolução de problemas

Sistemas de aprendizado baseados em conhecimento ajudam os aprendizes enquanto eles estão trabalhando em exercícios e resolvendo problemas. Há duas técnicas usadas principalmente. Uma delas é o diagnóstico inteligente de soluções completas de exercícios e tarefas de resolução de problemas. A outra é a abordagem chamada de *model tracing*, ou acompanhamento do modelo. Tal abordagem acompanha

o usuário na resolução de um problema, e dá conselhos quando o caminho escolhido por ele irá resultar em um erro.

2.5 Sistemas Tutores Hipermedia Inteligentes

Com a popularização da Web, também os sistemas tutores inteligentes começaram a se adaptar para este meio, e então surgiu uma nova categoria de STI chamada Sistemas Tutores Hipermedia Inteligentes (COSTA, 1997). Este tipo de sistema integra a tecnologia de sistemas tutores inteligentes com as técnicas de hipermedia adaptativa, descritas no Capítulo 3.

3 Hipermídia Adaptativa

3.1 Introdução

Sistemas de hipermídia em geral e, em especial, sistemas baseados na Web estão se tornando cada vez mais populares como ferramentas para aquisição de informações. O mecanismo de links oferece aos usuários um alto grau de liberdade para a navegação.

Entretanto, esta liberdade navegacional em aplicações convencionais hipermídia leva a problemas de compreensão e orientação. (NIELSEN, 1990 apud DE BRA et al., 1999)

A carga cognitiva e a desorientação limitam a eficiência do hipertexto para o aprendizado. Adaptando cognitivamente um sistema de hipermídia ao usuário e dando dicas instrucionais, os efeitos destes problemas podem ser reduzidos (CLIBBON, 1995). Este campo de estudo chama-se **hipermídia adaptativa**.

3.2 Conceitos

3.2.1 Hipermídia

Entende-se por *hipermídia* conteúdo de diversas formas (texto, áudio, imagem, etc.) dividido e organizado em uma forma de rede, com ligações entre os elementos, os chamados *links*.

HENZE (2000) compila as definições da literatura e resume:

Hipertexto: Um conjunto de nós de texto que estão conectados por links. Cada nó contém uma certa quantidade de informação (texto) e um número de links para outros nós.

Hipermídia: Extensão do hipertexto que faz uso de múltiplas formas de mídia, tais como texto, vídeo, áudio, gráficos, etc.

3.2.2 Hipermídia adaptativa

Sistemas de hipermídia adaptativa, são sistemas que aplicam diferentes formas de modelos de usuário para adaptar o conteúdo e os links de páginas hipermídia para o usuário (BRUSILOVSKY, 1997). Tais sistemas precisam conhecer o usuário para poder se adaptar ao seu conhecimento, preferências e objetivos atuais.

HENZE (2000) diz que o objetivo de tais sistemas é aumentar a funcionalidade dos sistemas hipermídia, personalizando-os para cada indivíduo. Cita também BRUSILOVSKY (1996): “Por sistemas de hipermídia adaptativa nos referimos a todos os sistemas de hipertexto e hipermídia que refletem algumas características do usuário no modelo do usuário e aplicam este modelo para adaptar ao usuário vários aspectos visíveis do sistema.”

3.3 Aplicações

Segundo BRUSILOVSKY (1994) existem três principais campos para a aplicação de sistemas de hipermídia adaptativa: sistemas de documentação online, aplicações com ajuda avançada e facilidades explicativas e sistemas educacionais.

Sistemas de documentação online: As tecnologias de hipermídia adaptativa são usadas aqui para fornecer informações diferentes para usuários diferentes, e fornecer suporte à navegação personalizada em uma grande quantidade de informações.

Aplicações com ajuda avançada e facilidades explicativas: Aplicações como pacotes estatísticos, sistemas CAD (Computer Aided Design) e sistemas de suporte à decisão podem usar a hipermídia para fornecer explicações sobre detalhes e conceitos específicos da aplicação ou do sistema. A razão para usar hipermídia adaptativa seria a necessidade de adaptar as explicações para classes diferentes de usuários.

Sistemas educacionais: Os componentes hipermídia são utilizados em sistemas educacionais para dar suporte à exploração do material educacional por parte do estudante. A hipermídia adaptativa é necessária aqui para adaptar a informação apresentada ao nível atual de conhecimento do estudante, e fornecer suporte navegacional comentando links existentes ou sugerindo o melhor link para seguir.

3.4 Tipos de hipermídia adaptativa

A hipermídia adaptativa, segundo BRUSILOVSKY (1996) é dividida em duas principais áreas: apresentação adaptativa e suporte à navegação adaptativa.

DE BRA (1999) discorda da separação feita por BRUSILOVSKY, e evita estes termos por julgar que eles podem causar confusão, uma vez que frequentemente o suporte à navegação adaptativa é feito através de anotação de links. Isto altera a apresentação do link, mas mesmo assim não é considerado apresentação adaptativa.

3.4.1 Apresentação adaptativa

Apresentação adaptativa é a idéia de adaptar o conteúdo de uma página acessada por um usuário em particular, ao conhecimento, objetivos e outras características atuais deste usuário. (BRUSILOVSKY, 1996)

Segundo BRUSILOVSKY (1996), embora exista um bom número de boas técnicas para apresentação adaptativa de multimídia, poucos sistemas têm implementado completamente a apresentação adaptativa. Um dos exemplos de sistemas que o fazem é o C-Book (KAY & KUMMERFIELD, 1994), um curso adaptativo da linguagem C.

3.4.2 Suporte à navegação adaptativa

Suporte à navegação adaptativa é a idéia de ajudar os usuários a encontrar seus caminhos no hiperespaço, através da adaptação da maneira de apresentar links, ao conhecimento atual, objetivos e outras características deste usuário. (BRUSILOVSKY, 1996).

O suporte à navegação adaptativa pode ser feito de diversas formas, listadas por DE BRA et al. (1999):

Condução direta: São mostrados botões do tipo “Próximo” ou “Continue”, que levam o usuário para o nó que o sistema determina ser o mais apropriado.

Ordenação de links: Uma lista de links é ordenada e apresentada do mais relevante ao menos relevante.

Anotação de links: Os links aparecem de forma diferente (cores e formas) dependendo da relevância do seu destino.

Ocultamento de links: Links que levam a locais inapropriados naquele momento são ocultos.

Desabilitação de links: Links que levam a locais inapropriados naquele momento são desabilitados. Aparecem como texto normal.

Remoção de links: Links apropriados são simplesmente removidos.

Adaptação de mapas: Há sistemas que apresentam uma representação gráfica da estrutura de links, chamada mapa. Este mapa gráfico pode ser adaptado.

3.4.3 Projetos de hipermídia adaptativa

Uma forma de hipermídia adaptativa, a anotação de links, pode ser feita através da coloração dos links, dando a cada cor um significado, como por exemplo no ELM-ART (WEBER & SPECHT, 1997). O ELM-ART, que significa ELM Adaptive Remote Tutor (Tutor Remoto Adaptativo), difere-se de “hiperlivros” na Web em dois aspectos principais. Primeiro, o ELM-ART “conhece” o material que apresenta aos alunos e dá suporte no aprendizado e navegação pelo material do curso. Segundo, todos os exemplos e problemas (que são componentes importantes de qualquer livro) no ELM-ART não são somente texto, mas “experiência ao vivo”. Usando o ELM-ART, o estudante pode investigar todos os exemplos e resolver todos os problemas online.

A forma de adaptabilidade do ELM-ART chama-se *anotação adaptativa*, e significa que o sistema usa dicas visuais (ícones, fontes e cores) para mostrar o tipo e estado educacional de cada link. Ele se utiliza da metáfora de um sinal de trânsito, e coloca uma bolinha verde ao lado do link que o aluno já está pronto para visitar e aprender, uma bolinha amarela para aquele link que o aluno está pronto mas que ainda não é recomendado, e uma bolinha vermelha para indicar que o link ainda não deve ser visitado. Além disso, tem barrinhas indicadoras da quantidade de conhecimento já adquirido sobre cada assunto. A Figura 4 mostra uma tela do ELM-ART.

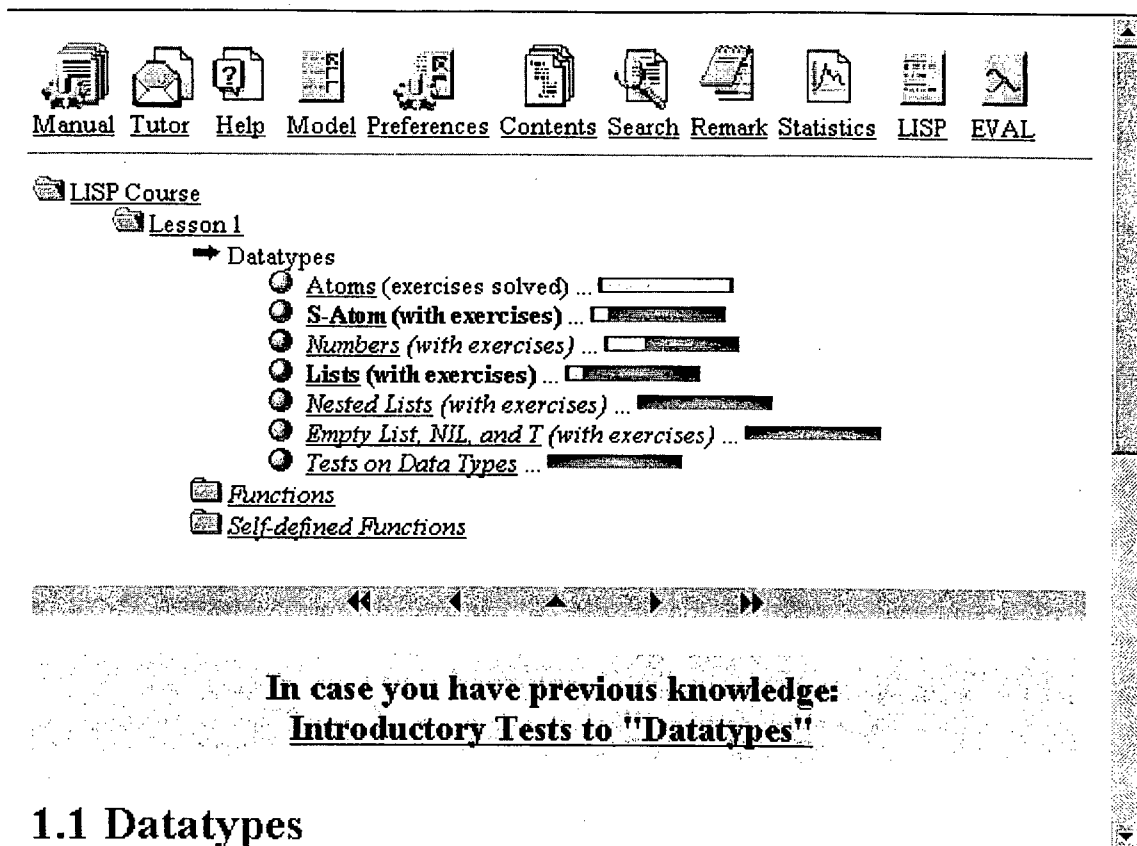


Figura 4 - Tela do ELM-ART

Já o INTERBOOK (BRUSILOVSKY, 2000), tem uma interessante interface, que simula um livro, ou caderno, para causar um ambiente familiar ao usuário. A Figura 5 traz uma tela do InterBook.

InterBook User and Author Manual
 ♦3. Interbooks: InterBook-served electronic textbooks

Teach this Page

3. Interbooks: InterBook-served electronic textbooks

InterBook supports an author in producing full-featured WWW ET. All InterBook-served ET have generated table of content and a glossary integrated with index and a search interface. In addition, InterBook provides an author with an easy way to produce adaptive ET with adaptive navigation support and adaptive help. This section uncovers the structure of knowledge that stands behind each InterBook-served ET and presents some specific features of InterBook-served ET which we call an *interbook*.

Background:

[adaptive ET](#)

[ET](#)

Outcome:

[interbook](#)

Figura 5 - Tela do InterBook

O curso em Estruturas e Sistemas Hiperímia, organizado por DE BRA (2000) e baseado no AHA (DE BRA, 1998) funciona numa estrutura mais livre. A pessoa vai lendo as páginas sem ser guiado por nenhuma ordem. Ao chegar ao fim de um dado conceito, ela simplesmente não encontra links para continuar, e então deve recorrer à lista de páginas ainda não visitadas. Na Figura 6 observa-se uma tela do curso.

tu Eindhoven University of Technology **change configuration** **message board**
 Marcel Ribas has read 22/155 pages. (these read, these still to do.)

Instructions for studying this course off-line

This course consists of three kinds of items:

- There are information pages about the course subject. These pages can be read in a different order by every student. In order to ensure that this personal choice does not lead to information you cannot understand because of missing foreknowledge, and to avoid leading you to the same pages over and over again without you knowing it, the course uses *adaptive hypermedia* techniques. This implies that the text and the links you see are *not static*.
- There are a series of multiple choice tests, one after each "section" and one at the end. These tests are evaluated by a server-side script.
- There is an assignment, which consists of a large linear document which you must transform into a hyperdocument.

Figura 6 - Tela do curso Estruturas e Sistemas Hiper mia

Percebe-se na tela que o usu rio pode a qualquer momento saber quantas e quais p ginas j  visitou, e acessar todas as p ginas ainda n o visitadas.

3.4.4 Efic cia da hiper mia adaptativa

Por causa dos problemas da desorienta o, da sobrecarga cognitiva, do desvio das informa es relevantes, t m-se tentando provar a efic cia da hiper mia adaptativa no aprendizado.

Um experimento bastante interessante foi o feito por Clibbon, e descrito em (CLIBBON, 1995). Ele tomou 84 estudantes, e apresentou um mesmo conte do usando tr s m todos diferentes: hipertexto est tico, hipertexto adaptativo, e palestra em sala de aula. Cada estudante, atrav s de um pr -teste, foi classificado em um grupo, de acordo com o seu estilo de aprendizado: ativista, reflexivo, te rico ou pragm tico. Depois estes foram distribu dos entre as tr s formas de apresenta o. Os alunos tinham uma hora

para adquirir o máximo de conhecimento possível, e depois realizaram um teste objetivo de 14 questões. CLIBBON (1995) pretendia responder três perguntas:

O estilo de aprendizado afeta o resultado do aprendizado?

O método de apresentação afeta o resultado do aprendizado?

Há efeitos de interação entre estilos de aprendizado e método de apresentação que afetam o resultado do aprendizado?

De acordo com os resultados dos testes e usando métodos estatísticos, ele chegou à conclusão que, para aquele domínio, o método mais eficaz de ensino foi a palestra em sala de aula, seguido de perto pelo hipertexto adaptativo. Por último ficou o hipertexto estático, bem abaixo dos outros dois. Destacou que é necessário realizar este teste com outros domínios de ensino, mas serve como prova que o hipertexto ou hipermídia adaptativos são mais eficazes do que suas contrapartes estáticas no ensino.

3.5 Modelos de Usuário

A tecnologia de hipermídia adaptativa é centrada no que se chama de “modelos de usuário”, o que é o conhecimento que o sistema tem sobre aquele usuário que o está acessando. O modelo de usuário é algo dinâmico, que vai se alterando e adaptando conforme o sistema vai descobrindo mais informações do usuário.

HENZE (2000) distingue dois tipos principais do modelos de usuário: o *modelo superposição* e o *modelo estereótipo*.

3.5.1 Modelo Superposição (Overlay)

No modelo superposição, o estado ou conhecimento do usuário é descrito como um subconjunto do conhecimento de um especialista sobre o domínio, por isso o termo

“superposição” (overlay). A falta de conhecimento do usuário é derivada através da comparação do conhecimento deste usuário com o conhecimento do especialista.

Pode-se dizer que um modelo de usuário baseado em superposição pretende ser uma intersecção entre conhecimento do usuário e o conhecimento do especialista.

Segundo HENZE (2000), a parte crítica deste tipo de modelo de usuário é encontrar uma boa estimativa inicial de conhecimento.

3.5.2 Modelo Estereótipo

O modelo estereótipo classifica usuários em estereótipos. Considera-se que os usuários que pertencem a uma certa classe têm as mesmas características. Pode acontecer uma classificação de estereótipo diferente para cada característica adaptativa do sistema.

HENZE (2000) considera que um problema com essa abordagem é que os estereótipos podem se tornar tão restritos que se tornam obsoletos (um usuário por estereótipo) ou que um usuário não possa ser classificado em nenhum deles.

3.6 Hipermídia Adaptativa na Educação via Web

A educação é uma das principais áreas de aplicação da hipermídia adaptativa e um número de sistemas educacionais de hipermídia adaptativa foi criado antes mesmo da “corrida da Internet” (BRUSILOVSKY, 1997).

A adaptação na educação via Web é especialmente importante por pelo menos duas razões. Primeiro, a maioria das aplicações baseadas em Web são utilizadas por uma variedade muito maior de usuários do que uma aplicação standalone. Uma aplicação Web que seja desenhada tendo em mente uma particular classe de usuários

não servirá para outros usuários. Segundo, em muitos casos, o usuário está “sozinho” trabalhando com um “tutor” ou “curso” Web (provavelmente de casa). Por isso a ajuda que um colega ou o professor tipicamente fornecem adaptativamente em uma situação normal de sala de aula não está disponível. (BRUSILOVSKY, 1997)

Além disso, há várias semelhanças técnicas entre as metodologias de Sistemas Tutores Inteligentes e Sistemas de Hiperídia Adaptativa. A maior delas é no que diz respeito ao Modelo do Usuário, ou Modelo do Estudante. Ambos os sistemas precisam manter informações sobre o seu usuário para que possam se adaptar ou fornecer auxílio.

Portanto, a junção dos sistemas tutores inteligentes com os sistemas de hiperídia adaptativa promete ser uma área de muitos avanços no aprendizado via Web.

4 Sistemas Educacionais na Web

4.1 Introdução

Os benefícios da educação pela Web são claros: independência de salas de aula e independência de plataforma (BRUSILOVSKY, 1997). Uma aplicação instalada e assistida em um lugar pode ser utilizada por milhares de aprendizes por todo o mundo que estejam equipados com algum tipo de computador conectado à Internet. Milhares de cursos baseados em Web e outras aplicações educacionais foram disponibilizadas na Web nos últimos cinco anos.

O problema é que a maioria do material educacional e instrucional hoje existente na Internet não é nada mais do que redes de páginas estáticas de hipertexto. Alguns grupos vêm estudando formas de aproveitar as teorias e técnicas instrucionais já existentes, para uso na instrução via Web.

Foi preciso rever alguns conceitos e teorias de aprendizado, para depois analisar o estado atual do ensino pela Web e como estas teorias vêm sendo empregadas.

4.2 Aspectos Pedagógicos

Existem diversas teorias sobre como o ser humano adquire conhecimento. Mas não se pode ter certeza de todos os processos que acontecem na mente humana que fazem com que o ser humano passe a saber algo que não sabia antes. Nenhuma teoria pode ser provada 100% correta.

4.2.1 Teorias de aprendizado

Dentre as várias teorias existentes, são descritas as três mais significativas pela sua importância histórica e aceitabilidade: o comportamentalismo, o cognitivismo e o construtivismo.

4.2.1.1 Comportamentalismo

A teoria do comportamentalismo concentra-se no estudo de comportamentos externos que podem ser observados e medidos (GOOD & BROPHY, 1990 apud MERGEL, 1998). Ela vê a mente como uma “caixa preta”, no sentido que a resposta a estímulos pode ser observada quantitativamente, totalmente ignorando a possibilidade de processos de pensamento ocorrendo dentro da mente. Alguns dos principais pesquisadores foram Pavlov, Watson, Thorndike e Skinner.

Há também uma tendência, capitaneada por Vygotsky, que mistura o comportamentalismo com o cognitivismo. Ou seja, o desenvolvimento humano se dá por uma mistura do interior com o exterior, uma integração dos aspectos biológicos e sociais do indivíduo.

4.2.1.2 Cognitivismo

“Teóricos cognitivistas reconhecem que muito do aprendizado envolve associações estabelecidas entre a contigüidade e a repetição. Eles também reconhecem a importância do reforço, embora ressaltem o seu papel em fornecer realimentação sobre a correção de respostas sobre o papel como um motivador. Entretanto, mesmo ao aceitar tais conceitos comportamentalistas, teóricos cognitivistas vêem o aprendizado como envolvendo a aquisição ou reorganização das estruturas cognitivas através das quais os humanos processam e guardam informação.” (GOOD & BROPHY, 1990 apud MERGEL, 1998).

Um dos principais nomes no desenvolvimento do cognitivismo é Jean Piaget, que desenvolveu os principais aspectos de sua teoria na década de 20. Os estudos de Piaget foram continuados e acabaram gerando uma nova corrente, o *construtivismo*.

4.2.1.3 Construtivismo

Há um certo consenso no meio dos educadores que uma das teorias que melhor explica como o ser humano aprende é o construtivismo. Os construtivistas acreditam que “os aprendizes constroem sua própria realidade ou pelo menos a interpretam baseados em suas percepções de experiências, tal que o conhecimento de um indivíduo é uma função das experiências anteriores, estruturas mentais, e crenças do indivíduo que são utilizadas para interpretar objetos e eventos.” (JONASSON, 1991 apud MERGEL, 1998).

No construtivismo, o conhecimento é construído a partir da experiência. Aprendizado é uma interpretação pessoal do mundo, um processo ativo no qual o significado é desenvolvido com base na experiência. O conhecimento conceitual vem da negociação do significado, o compartilhamento de múltiplas perspectivas e a mudança de nossas representações internas através do aprendizado colaborativo. O aprendizado deveria estar situado em condições realistas, os testes deveriam ser integrados com a tarefa e não uma atividade em separado (MERRIL, 1991 apud MERGEL, 1998).

4.2.2 Estilos de Aprendizado

É de senso comum a afirmação de que pessoas diferentes aprendem de forma diferente. Cada pessoa em seu processo educativo vai descobrindo a forma de estudo pela qual sente que vai aprendendo mais, retendo mais conhecimento.

Alguns estudantes tendem a focalizar em fatos, dados e algoritmos; outros se sentem mais confortáveis com teorias e modelos matemáticos. Uns respondem melhor a figuras, diagramas, esquemas; outros aproveitam melhor formas verbais, escritas ou orais. Alguns preferem aprender ativamente e interativamente; outros são mais introspectivos e individualistas.

Alguns estudiosos conseguiram identificar os chamados estilos de aprendizado, ou estilos de aprendizes. FELDER (1996) reuniu os quatro modelos de estilos de aprendizado mais referenciados: o MBTI (Myers-Brigs Type Indicator), o modelo de aprendizado de Kolb, o HBDI (Herrmann Brain Dominance Instrument) e o modelo de aprendizado Felder-Silverman.

4.2.2.1 MBTI (Myers-Brigs Type Indicator)

O modelo MBTI, que significa indicador de tipo Myers-Brigs, classifica os aprendizes de acordo com suas preferências em escalas derivadas da teoria de tipos psicológicos de Carl Jung. Os aprendizes podem ser:

- *Extrovertidos* (tentam as coisas, focalizam no mundo exterior das pessoas) ou *introvertidos* (pensam nas coisas, focalizam no mundo interior das idéias);
- *Sensores* (práticos, orientados a detalhes, focalizam em fatos ou procedimentos) ou *intuitivos* (imaginativos, orientados a conceitos, focalizam em significados e possibilidades);
- *Pensadores* (céticos, tendem a tomar decisões baseadas em lógica ou em regras) ou *sensíveis* (apreciativos, tendem a tomar decisões baseados em considerações pessoais ou humanísticas);
- *Julgadores* (preparam e seguem agendas, procuram o fechamento mesmo com dados incompletos) ou *perceptores* (adaptam-se a circunstâncias variáveis, resistem ao fechamento para obter mais dados).

Classificando os aprendizes nestes grupos, e combinando-os, tem-se 16 tipos diferentes de estilos de aprendizado.

4.2.2.2 Modelo de aprendizado de Kolb

Este modelo classifica os aprendizes como tendo uma preferência por experiência concreta ou conceitualização abstrata, e experimentação ativa ou observação reflexiva.

Ele define quatro tipos de aprendizes:

- Tipo 1 (concreto, reflexivo): Respondem bem a explicações sobre como o material aprendido se relaciona a suas experiências, seus interesses, e suas carreiras;
- Tipo 2 (abstrato, reflexivo): Respondem bem a informações apresentadas de uma forma lógica e organizada, e tiram mais proveito se tiverem tempo para reflexão;
- Tipo 3 (abstrato, ativo): Respondem bem quando têm oportunidades para trabalhar ativamente em tarefas bem definidas e aprender por tentativa e erro em um ambiente que permita que eles errem com segurança.
- Tipo 4 (concreto, ativo): Gostam de aplicar o material do curso em novas situações para resolver problemas reais.

4.2.2.3 HBDI (Herrmann Brain Dominance Instrument)

O HBDI, que significa Instrumento de Dominação do Cérebro de Herrmann, classifica os aprendizes em quatro classes diferentes baseado no funcionamento do cérebro humano, onde cada quadrante tem uma tarefa específica. Os quatro modos ou quadrantes são:

- Quadrante A: Lógico, analítico, quantitativo, factual, crítico;
- Quadrante B: Seqüencial, organizado, planejado, detalhado, estruturado;

- Quadrante C: Emocional, inter-pessoal, sensitivo, cinético, simbólico;
- Quadrante D: Visual, holístico, inovador.

4.2.2.4 Modelo de aprendizado de Felder-Silverman

Este modelo classifica os aprendizes como:

- *Sensores* (concretos, práticos, orientados a fatos e procedimentos) ou *intuitivos* (conceituais, inovadores, orientados a teorias e significados);
- *Visuais* (preferem representações visuais – figuras, diagramas) ou *verbais* (preferem explicações escritas ou faladas);
- *Indutivos* (preferem apresentações que procedem do específico para o genérico) ou *dedutivos* (preferem apresentações que partem do genérico para o específico);
- *Ativos* (aprendem tentando, trabalhando com outros) ou *reflexivos* (aprendem pensando nas coisas, trabalhando sozinhos);
- *Seqüenciais* (lineares, ordenados, aprendem em pequenos passos incrementais) ou *globais* (holísticos, aprendem sem uma ordem definida).

Existem também alguns testes, desenvolvidos por estes estudiosos, que são aplicados aos estudantes para tentar classificar o aprendiz em algum dos tipos descritos.

É importante salientar que em todos estes modelos, é praticamente impossível situar uma pessoa permanentemente em um único estilo de aprendizado ou tipo de aprendiz. Dependendo do assunto estudado, do ambiente, e de uma série de fatores, uma pessoa pode preferir um certo estilo de aprendizado ao invés do outro. A preferência pode ser forte, moderada, ou suave, mas nunca permanente.

FELDER (1996) ainda comenta que para funcionar bem em qualquer capacidade profissional, a pessoa deve trabalhar bem em todos os modos de estilo de aprendizado. Um objetivo da educação deveria ser ajudar os alunos a construir suas habilidades tanto

nos modos de aprendizados preferidos quanto nos que eles não se identificam tanto, pois a informação vem de todas as formas.

4.3 Design Instrucional

Design instrucional é a forma pela qual um conteúdo será ensinado aos estudantes. A eficácia do aprendizado depende diretamente de como foi desenhada a instrução. Esta forma deverá estar apoiada em uma ou mais teorias de aprendizado.

O profissional que realiza esta tarefa, geralmente um professor ou um especialista no domínio, é chamado de *designer instrucional*.

Dependendo dos aprendizes e do contexto, podem ser aplicadas diferentes teorias de aprendizado. O designer instrucional deve entender os pontos fortes e os pontos fracos de cada teoria de aprendizado para otimizar o uso delas numa estratégia apropriada de instrução.

Para que a Web, com sua natureza assíncrona, seja utilizada para instrução com possibilidades de adaptação e tutoria ao usuário, o design instrucional deve ser feito com bastante cuidado, com conhecimento das teorias de aprendizado e com vistas a uma boa organização do conteúdo a ser ensinado.

O design instrucional para cursos pela Web não pode ser o mesmo de sala de aula. Não se pode simplesmente transformar cursos de sala de aula ou apostilas em conteúdo HTML seqüencial. Assim, apenas se transfere o tédio da leitura no papel para a leitura numa tela de computador, que diga-se de passagem, é bem menos agradável.

O maior desafio hoje é encontrar técnicas de design instrucional que possam aplicar na Web e pela Web métodos modernos e libertadores de ensino.

4.4 Aprendizado pela Web

A Internet, em especial a Web, fascina os educadores. A possibilidade de alcançar distâncias maiores e proporcionar experiências mais ricas de aprendizado fez com que vários educadores tradicionais, e também pessoas da área técnica da computação, viessem a estudar formas de transmitir conhecimento pela rede.

De acordo com REEVES & REEVES (1997), há pelo menos dez dimensões em jogo para um aprendizado interativo: filosofia pedagógica, teoria de aprendizado, orientação de objetivos, orientação de tarefas, fonte de motivação, papel do professor, suporte meta-cognitivo, aprendizado colaborativo, sensibilidade cultural e flexibilidade estrutural. A Web é apenas um veículo para estas dimensões.

Muito se fala que hoje existe aprendizado pela Web em larga escala. Há uma confusão entre aprendizado e instrução. Grande parte das iniciativas explícitas de transmissão de conhecimento que se tem na Web hoje é uma mera roupagem nova da antiga educação à distância. Como defende RAMOS (1996), não há espaço para o diálogo, o fluxo das informações é unidirecional, tem-se um caráter autoritário. Se se assimila um conteúdo pré-definido, com atividades pré-planejadas, exatamente como previa o designer instrucional, houve instrução, treinamento, e não aprendizado. Conclui RAMOS (1996), “Se a perspectiva pedagógica adotada for opressora, então não haverá aprendizado. Não importa quão maravilhosa seja a ferramenta.”

A hipermídia pode ser uma valorosa ferramenta para uma instrução construtivista. A hipermídia permite um design ramificado ao invés do que um formato linear de instrução (MERGEL, 1998). Os hiperlinks permitem o controle por parte do aprendiz, o que é crucial para o aprendizado na teoria construtivista. Logo, a Web é um meio que

pode ser usado para o ensino construtivista, desde que o conteúdo instrucional seja projetado desta forma.

E se à Web forem adicionadas as outras tecnologias de colaboração da Internet, como salas de chat, e-mail, videoconferência, capacidades de trabalho em grupo (workflow), permitindo o diálogo entre estudantes e professores, dando aos aprendizes o protagonismo no aprendizado e o poder de expressão, tem-se aí o caminho para o desenvolvimento de uma consciência crítica no aprendiz e um efetivo aprendizado pela Web.

Não são poucas as iniciativas de se construir um ambiente definitivo para o aprendizado via Web. É um campo de estudo que está em alta nos dias de hoje. Percebe-se que infelizmente as plataformas propostas não conseguem uma distinção muito clara entre o conteúdo e a filosofia pedagógica. As plataformas Web de ensino tendem para uma massificação e centralização do processo de criação da instrução. Este trabalho busca uma forma de representar o conteúdo que se quer ensinar, de uma maneira tal que consiga acolher qualquer que seja a estratégia ou filosofia de ensino desejada. Inclusive permitindo o uso de diversas abordagens no mesmo processo, podendo alternar entre elas conforme o estudante ou o contexto. Para se chegar em tal nível, é preciso primeiro validar a estrutura de representação do domínio, para depois construir uma plataforma Web de ensino em cima dessa estrutura, ou integrá-la com ambientes já existentes.

5 Sistemas Multi-Agentes

5.1 Agentes Inteligentes

5.1.1 Definições e conceitos gerais

É possível definir um agente como alguém ou algo que age como representante de uma outra parte, para a finalidade específica de realizar atos específicos que parecem ser benéficos à parte representada. (HEILMANN et al., 1995)

Um agente de software é um programa que roda em background num ambiente computacional, e que realiza tarefas para o seu usuário (RIVER, 1996 apud FLEISCHHAUER, 1996). Agentes podem ser simples como sub-rotinas, mas tipicamente são entidades maiores com alguma forma de controle persistente (GENESERETH & KETCHPEL, 1994). Hoje em dia muitos programas já fazem isso. Só que existem características distintas que constituem um agente de software. Os agentes de software diferenciam-se de outras aplicações por terem mobilidade, autonomia, e a habilidade de interagir independente da presença do usuário. E, se for adicionada a inteligência a um agente, também terá que ser incluída a habilidade de tomar decisões adaptativamente. Isto implica na capacidade de processar informações de ambientes externos, como redes, bancos de dados, e a Internet, dado um conjunto de atitudes e conhecimento do usuário, que são entendidas pelo agente.

5.1.2 Principais Propriedades de um Agente Inteligente

São notadas algumas características chave de agentes inteligentes que os diferenciam de outras aplicações:

5.1.2.1 Autonomia

É a capacidade de tomar ações para realizar algumas tarefas e objetivos, sem a interferência do usuário final. Deve haver um elemento de independência no agente. Como se espera de humanos, os agentes tomam nossos interesses e desejos como entrada, e saem por si só para fazer as tarefas esperadas (HEILMANN et al., 1995).

Um agente autônomo deve manter uma agenda independentemente de seu usuário. Isto requer aspectos de ação periódica, execução espontânea, e iniciativa, na qual o agente deve ser capaz de tomar ações preemptivas ou independentes que irão eventualmente beneficiar o usuário. (FONER, 1993)

5.1.2.2 Comunicabilidade

Os agentes inteligentes devem acessar informação de outras fontes, sobre o atual estado do ambiente, durante o curso de suas tarefas. Isto requer a habilidade de comunicar-se com os repositórios desta informação. Estes podem ser outros agentes ou bases de dados. (HEILMANN et al., 1995)

Obviamente, quando se fala em comunicação, considera-se que ambas as partes devem falar a mesma língua. Ou seja, é necessário definir um protocolo e interfaces de comunicação, de uma forma bem detalhada e precisa.

5.1.2.3 Cooperação

Como extensão natural da comunicabilidade, os agentes inteligentes devem ter um espírito cooperativo. A idéia é que eles trabalhem juntos para realizar tarefas complexas, aproveitando-se das particularidades de cada um para cumprir seus objetivos. (HEILMANN et al., 1995)

5.1.2.4 Raciocínio

A capacidade de raciocínio é talvez o aspecto mais importante que distingue um agente inteligente dos outros agentes. Dizer que um agente inteligente tem raciocínio significa dizer que ele tem a habilidade de inferir e extrapolar, baseado no conhecimento atual e experiências, numa maneira racional e reprodutível. (BELGRAVE, 1995)

Esse raciocínio pode ser de três tipos:

Baseado em regras, onde eles usam um conjunto de condições prévias para avaliar as condições no ambiente externo;

Baseado em conhecimento, onde eles têm à disposição grandes conjuntos de dados sobre cenários anteriores e ações resultantes, dos quais eles deduzem seus movimentos futuros ou;

Baseado em evolução artificial, onde eles dão à luz novas gerações de agentes que carregam capacidades de raciocínio cada vez maiores.

O raciocínio é altamente dependente do antropomorfismo, que falaremos em seguida.

5.1.2.5 Comportamento Adaptativo

Para manter as capacidades de autonomia e raciocínio, o agente deve ter alguma forma de analisar o estado de seu domínio externo, ou seja, a porção do ambiente ao alcance do agente, e incorporar isto em suas decisões sobre ações futuras. Os agentes devem ser capazes de examinar o seu ambiente externo, no nosso caso a Web, e o sucesso de ações anteriores tomadas sob circunstâncias similares, e adaptar suas ações para aumentar a probabilidade de alcançar com sucesso seus objetivos.

5.1.2.6 Confiabilidade

Deve existir uma forte confiança que o agente vai representar o seu usuário com precisão. Os agentes inteligentes devem demonstrar veracidade, que é a suposição que

este não comunicará informações falsas, e benevolência, que é a suposição que ele não terá objetivos conflitantes e não atuará de modo contrário ao seu objetivo.

5.1.2.7 Antropomorfismo

É a habilidade de demonstrar propriedades humanas, como por exemplo, emoções, crenças e intenções.

Além destas, existem também outras características, como mobilidade, reatividade, iniciativa, sociabilidade e percepção, que também ajudam a identificar os agentes inteligentes e classificá-lo. Além disso, é importante salientar que nenhuma destas características sozinha estabelece que o software é um agente inteligente.

5.1.3 Classificação de Agentes

É interessante conhecer os diversos tipos de agentes, para que se possa entender as diferenças entre eles, e escolher a ferramenta adequada para nossa aplicação. Além dos agentes inteligentes, já comentados, existem ainda os agentes móveis, os autônomos, os coordenados, os aprendizes ou evolutivos, os reativos e os cognitivos.

5.1.3.1 Agentes Inteligentes

Agentes inteligentes são entidades que realizam algum conjunto de operações em favor de um usuário ou outro programa, com algum grau de independência ou autonomia, e assim empregam algum conhecimento ou representação dos objetivos ou aspirações do usuário.

5.1.3.2 Agentes Móveis

Mobilidade é a capacidade de um agente de mover-se pela rede, transportar-se entre máquinas, levando consigo seus dados e código. A tecnologia de agentes móveis é relativamente nova, e por isso somente agora vê-se exemplos destes já implementados. Com o advento da linguagem Java, tem-se um meio fácil de transportar um programa pela rede. Os desenvolvedores podem se aproveitar disso para mover seus agentes.

5.1.3.3 Agentes Autônomos

Sistemas capazes de apresentarem autonomia, ou seja, situados dentro de um ambiente, que sentem esse ambiente e agem sobre ele, de acordo com sua própria agenda e determinação.

5.1.3.4 Agentes Coordenados

Coordenação é o ato de gerar interdependências entre atividades. Para alcançar um objetivo maior, podemos arranjar diversos agentes que realizam tarefas específicas, coordenando-os de forma que suas atividades se completem.

5.1.3.5 Agentes Aprendizes e Adaptativos

Um agente aprendiz é aquele que observa o comportamento do usuário ou de outro programa, para poder depois otimizar estes serviços, automatizando-os da melhor forma possível. É aquele agente que pode, ou precisa, ser treinado, para depois executar suas tarefas sem a assistência do usuário. Ele vai criando um conjunto de conceitos, sobre os quais pode basear melhor suas decisões.

5.1.3.6 Agentes Reativos

Um agente reativo responde a estímulos, sem ter memória das ações realizadas no passado nem previsão da ação a ser tomada no futuro. Geralmente atuam em sociedades, baseando-se em modelos de organização biológica, como uma colônia de

formigas, por exemplo. Um formiga não aparenta ser muito inteligente, mas o grupo comporta-se como uma entidade inteligente, quando constrói seu formigueiro.

5.1.3.7 Agentes Cognitivos

Os agentes cognitivos podem raciocinar sobre as ações tomadas no passado e planejar as ações a serem tomadas no futuro.

5.2 Sistemas Multi-Agentes

5.2.1 Conceito

Sistemas multi-agentes são uma coleção de agentes autônomos que, normalmente, cooperam em direção a um objetivo global. (GARCIA, 1998)

Os agentes em sociedade, representados principalmente pelos sistemas multi-agentes, propõem um incremento substancial na funcionalidade da emergente tecnologia dos agentes inteligentes.(FARACO, 1998) É o mesmo que afirmar que uma comunidade realiza uma determinada tarefa melhor e mais rápido do que um único ente.

5.2.2 Características de sistemas multi-agentes

5.2.2.1 Cooperação

Uma política de cooperação se faz necessária num sistema multi-agentes, já que é ela quem dita a maneira com que estes demonstram suas necessidades a outros agentes a fim de realizar uma determinada tarefa. (FARACO, 1998)

5.2.2.2 Coordenação

A coordenação diz respeito à maneira com que os agentes estão organizados para que o esforço individual de cada um se reflita no objetivo maior do sistema.

Entre os mecanismos de coordenação de um sistema multi-agentes, destacamos dois, definidos em (PARAISO, 1997), o mestre-escravo e o mecanismo de mercado:

Mestre-escravo: Neste mecanismo existem duas classes de agentes, os mestres e os escravos. Os escravos são coordenados por um mestre que distribui as tarefas e fica esperando o resultado.

Mecanismo de Mercado: Os agentes estão num mesmo nível, e sabem quais as tarefas que cada um pode realizar. Este tipo de estrutura tem a vantagem de diminuir a quantidade de mensagens trocadas, pois os agentes conversam direto entre si, sem precisar passar por um mestre.

5.2.2.3 Comunicação

É necessário que eles se entendam, ou seja, falem a mesma língua, ou protocolo de comunicação. Um recurso que vem sendo bastante utilizado é a linguagem KQML (Knowledge Query and Manipulation Language).

Um protocolo define o formato e o tipo de conteúdo de todas as mensagens que podem ser enviadas ou reconhecidas por um agente.

KQML é uma linguagem e protocolo para troca de informações e conhecimento. KQML é tanto um formato de mensagens e um protocolo de tratamento de mensagens to suportar o compartilhamento de informações entre agentes. KQML pode ser usada como uma linguagem para uma aplicação interagir com um sistema inteligente ou para dois ou mais sistemas inteligentes compartilharem conhecimento para suportar a solução cooperativa de problemas.

O uso de KQML para a comunicação entre um sistema multi-agentes, descrito em (FININ et al., 1997) é bastante disseminado. Mas nada impede que se desenvolva toda uma linguagem e protocolo especial para isso. Principalmente hoje depois da popularização do XML (eXtended Markup Language), linguagem de marcação

genérica, que permite a livre definição de semântica de mensagens, e suportada por diversas plataformas computacionais.

5.2.3 Arquiteturas de Sistemas Multi-Agentes

GENESERETH & KETCHPEL (1994) distinguem as arquiteturas de sistemas multi-agentes em dois grandes grupos: comunicação direta e coordenação assistida.

Na comunicação direta, os agentes trocam mensagens independentemente. Na coordenação assistida, eles dependem de um processo ou outro agente coordenador, para determinar o destino das mensagens e organizar o fluxo da comunicação.

Esta concepção é bastante semelhante à dos mecanismos de coordenação, apresentada anteriormente no item 5.2.2.2.

6 Um Modelo de Representação do Domínio de Conhecimento

6.1 Introdução

Tanto os tradicionais Sistemas Tutores Inteligentes, antes da popularização da Internet, quanto as suas versões Web, assim como qualquer sistema computacional que tenha objetivos educacionais, necessitam de uma representação lógica do domínio de aprendizado.

Existem várias iniciativas na busca de sistemas educacionais adaptativos via Web, e algumas utilizando inclusive sistemas multi-agentes, como os trabalhos de GIRAFFA & VICCARI (1998), COSTA (1999) e BICA (2000).

GIRAFFA & VICCARI implementaram o MCOE (Multi-agent CO-operative Environment), um sistema tutor inteligente via Internet. No sistema MCOE existem três agentes, um tutor e dois estudantes, pois se busca incentivar o aprendizado através da cooperação.

BICA demonstrou com o Eletrotutor (BICA, 2000) a utilização de uma sociedade de agentes, composta pelos agentes: Gerenciador do Domínio, Gerenciador de Exercício, Gerenciador de Exemplo, Gerenciador de Atividades, Gerenciador de Modelo do Aluno, Gerenciador de Interfacê e Gerenciador de Comunicação.

COSTA (1999) propõe uma arquitetura de federação de agentes organizada através de três classes de agentes: agentes de interface, agentes de informação e agentes consultivos.

Estas três arquiteturas descritas acima representam o domínio de aprendizado de uma forma genérica, com um único agente gerenciando e mantendo informações sobre todo o conteúdo.

Este trabalho tem por objetivo propor um modelo de representação do domínio de ensino/aprendizado, e uma diferente representação do conhecimento do estudante sobre este domínio.

6.2 A Composição de um Domínio de Conhecimento

A proposta é fugir um pouco dos modelos tradicionais de sistemas tutores inteligentes, e distribuir o conhecimento que o sistema tem sobre o aluno, o chamado Modelo do Aprendiz, por uma comunidade de agentes inteligentes. Espera-se com isso permitir maior flexibilidade no comportamento individual de cada agente e aproveitar melhor as diferenças entre tipos e níveis de importância de cada unidade de conhecimento.

Nas abordagens multi-agentes descritas anteriormente, cada seção do domínio, ou unidade de conteúdo, é tratada rigorosamente da mesma forma. Mas o conteúdo pode ser encontrado das mais diversas formas (vídeo, áudio, exemplos, texto). Às vezes, existem unidades de conteúdo que descrevem o mesmo conceito, mas de formatos diferentes. Em algumas abordagens, como a de COSTA (1999), existe até esse tipo de informação sobre uma determinada unidade de conteúdo armazenada numa base de dados, e lida pelo agente gerenciador quando necessita. Mas é um simples atributo e não significa necessariamente um comportamento diferenciado do agente, dependendo do tipo de conteúdo.

Além disso, cada unidade de conteúdo tem importância diferenciada, dependendo do contexto onde está inserida. O domínio de conhecimento, ou domínio de aprendizado, em um sistema educacional é um conjunto de unidades de conhecimento, selecionadas por um especialista. Este domínio muitas vezes é chamado de *currículo*.

Este currículo é montado a partir de unidades de conhecimento já criadas anteriormente, permitindo a reutilização de material didático, ou novas unidades recém desenvolvidas especialmente para este novo domínio. Mas torna-se possível que uma unidade de conteúdo esteja presente em mais de um domínio de aprendizado, a partir do momento em que a equipe de designers instrucionais começa a intercambiar material produzido.

Portanto, fica claro que a importância de uma unidade de conteúdo não é algo estático, mas depende do contexto ou domínio de conhecimento no qual ela está inserida. E esta importância é atribuída pelo designer instrucional, o especialista no domínio.

Dependendo do grau de importância atribuído a uma unidade de conteúdo, pode-se ou não exigir que o aluno domine aquela parte do assunto, para que se considere que ele já alcançou um nível bom de conhecimento do todo.

O comportamento do sistema em relação às unidades mais importantes deve ser diferente do comportamento em relação às unidades menos relevantes.

Pode-se também buscar outras características de unidades de conhecimento, como por exemplo, o grau de dificuldade, e se desenhar diversos tipos de tratamento entre o sistema tutor e o aluno.

Neste modelo, procurou-se uma representação do domínio através de um sistema multi-agentes, de uma forma que essas individualidades das unidades de conhecimento pudessem ser respeitadas. Busca-se uma representação flexível, que se adapte melhor

aos diversos tipos de conteúdo, assim como também às diversas técnicas de ensino, táticas instrucionais e estilos de aprendizado.

Para construir este sistema multi-agentes, será preciso primeiramente definir uma organização do conteúdo, para em cima disto, modelar os agentes, estabelecer a comunicação entre eles, o banco de dados para a persistência das informações e o formato da apresentação do conteúdo para o aluno.

Também, para que fosse possível demonstrar o funcionamento do sistema multi-agentes, foi necessário definir uma plataforma limitada de um sistema tutor inteligente via Web. Essa plataforma de testes não se preocupa tanto em implementar técnicas de ensino ou observar estilos de aprendizado. A pesquisa se concentra apenas em uma das possíveis características dos sistemas tutores inteligentes via Web, que é a seqüência de currículo.

6.3 Organização do Conteúdo

Nesta abordagem, tudo começa pela organização do conteúdo. Não basta simplesmente transformar as páginas de uma apostila ou livro já existente para HTML e fazer links entre os capítulos.

É necessário que se segmente o domínio de ensino em pequenas unidades de conhecimento, e que se crie uma rede de ligações entre as unidades de conhecimento que compõem um curso.

Para organizar as unidades de conhecimento, será utilizado o modelo sugerido por STERN & WOOLF (1998), similar ao Intelligent Guide (KHUWAJA et al., 1996). É importante salientar que embora este modelo tenha sido o escolhido para fins de validação da arquitetura multi-agentes, não se considera este modelo como definitivo. A

arquitetura foi desenhada de forma a poder adaptar-se a outras formas de segmentação do domínio, e permitir a integração com plataformas de ensino à distância pela Web.

STERN & WOOLF propõem a segmentação do conteúdo a ser ministrado em uma rede semântica de unidades de conhecimento. Cada unidade de conhecimento representa um conceito, ou tópico, que se quer ensinar.

As unidades de conhecimento se relacionam através de links definidos por um especialista no conteúdo. Estes links podem ser de quatro tipos:

Pré-Requisito: diz-se que uma unidade de conhecimento é pré-requisito de outra quando o conhecimento que se tem sobre essa unidade é necessário para que se aprenda a seguinte.

Co-Requisito: diz-se que uma unidade de conhecimento é co-requisito de outra quando o conhecimento das duas unidades é necessário para que se aprenda uma outra seguinte.

Relacionado: diz-se que uma unidade de conhecimento é relacionada a outra quando o conhecimento que se tem sobre essa unidade não é necessário, mas pode auxiliar no aprendizado da seguinte.

Reforço: diz-se que uma unidade de conhecimento é reforço de outra quando o estudo dessa unidade pode auxiliar a compreensão da outra unidade, quando esta já foi estudada e ainda não assimilada por completo.

A Figura 7 apresenta um gráfico exemplificando um domínio de conhecimento organizado conforme a proposta.

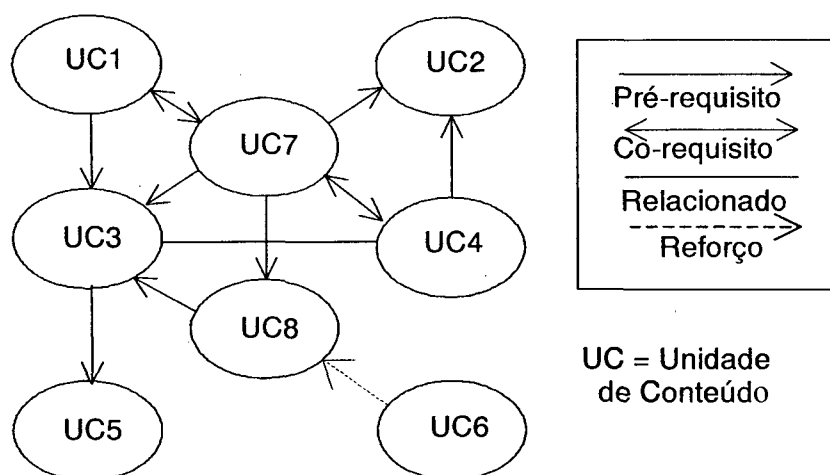


Figura 7 - Exemplo de Organização de Conteúdo

Para se ter um bom desempenho dos agentes do protótipo em sua tarefa de guiar o aluno pelo melhor caminho possível, é importante que o domínio seja bem segmentado e as relações entre as unidades de conhecimento sejam bem definidas. Logicamente se faz imprescindível a figura do especialista no domínio para esta tarefa.

6.4 Definição da Plataforma de Testes

A plataforma de testes, doravante chamada *protótipo*, fez-se necessária para que se observasse o sistema multi-agentes em funcionamento. Não se tinha o objetivo de construir todo um sistema educacional via Web, ou um sistema tutor inteligente completo, pois isso foge do escopo deste projeto. Todavia, também não se desejava já no início integrar esse sistema a uma plataforma já consolidada. Embora a intenção seja mesmo poder, no futuro, utilizar esta representação de domínio em outros sistemas educacionais via Web, a sua concepção teve que ser completamente independente, para que se concentrasse nas particularidades e dificuldades que ela apresenta em si própria.

Definiu-se então que o sistema seria testado com um curso em Técnicas de Programação, onde cada unidade de conteúdo seria uma página HTML, apresentando texto ou figuras. O conteúdo foi segmentado seguindo as orientações anteriores e as relações entre as unidades de conteúdo definidas.

No Capítulo 7 é descrita toda a estrutura física da implementação.

6.5 Modelagem dos Agentes

Existem quatro tipos de agentes: Aluno, Curso, Objeto de Conteúdo e Teste, numa arquitetura descentralizada. É uma arquitetura descentralizada pois não é em forma de estrela. Nenhum agente funciona como um “hub” ou roteador, centralizando o envio e recepção de mensagens.

Cada objeto de conteúdo definido no item anterior é representado por um agente. Cada instância do curso para um dado estudante consistirá numa comunidade de agentes, um sistema multi-agentes.

O agente Aluno tem a missão de validar a entrada do usuário no sistema e junto com o agente Curso acordar a comunidade de agentes objeto de conteúdo responsáveis pelo aprendizado daquele aluno específico naquele curso. Depois que a comunidade de agentes estiver ativa, não há mais necessidade dos agentes Aluno e Curso. Os agentes objeto de conteúdo conversam entre si.

6.5.1 Características dos agentes

De acordo com a teoria de agentes colocada no Capítulo 5, os agentes deste sistema tem as seguintes características: autonomia, pois funcionam independentes, comunicabilidade, pois precisam se comunicar com os outros agentes e o meio,

cooperação, pois trabalham juntos, raciocínio, para poder inferir no conhecimento e apresentar o melhor caminho ao usuário, comportamento adaptativo, para atualizar seu conhecimento de acordo com o progresso do aluno, confiabilidade, já que estão realizando as ações para o fim que foram criados, e antropomorfismo, pois seu objetivo é ser eficaz na ajuda ao estudante como um tutor humano.

Pelas mesmas razões, podem ser classificados nas seguintes classes listadas no Capítulo 5: agentes inteligentes, autônomos, coordenados, aprendizes e adaptativos, reativos e cognitivos. De todas as classes mencionadas anteriormente, a única na qual estes agentes não se encaixam é na classe de agentes móveis, pois eles permanecem no mesmo servidor durante toda a sua vida.

6.6 A Linguagem UML

Os agentes estão sendo projetados usando a linguagem UML (Unified Modeling Language), originalmente desenvolvida para representar designs de aplicações orientadas a objeto. A UML vem ganhando grande aceitação comercial em projetos orientados a objeto.

UML (Unified Modeling Language) é um conjunto de modelos padrão usados para fazer o design de projetos de programação orientados a objeto. (STURM, 1999)

Um modelo é uma descrição do problema que deve ser resolvido. Ele simplifica a realidade capturando um subconjunto de entidades e relacionamentos no domínio do problema. (STURM, 1999)

Um domínio do problema descreve não somente um problema particular mas também as condições sob as quais o problema ocorre. É portanto uma descrição de um problema e o contexto relevante daquele problema. (STURM, 1999)

No caso, o problema consiste em dar um acompanhamento personalizado a cada um dos alunos que participa do curso. Por isso o contexto do aprendizado assíncrono é bastante importante e deve ser considerado na modelagem.

Uma linguagem de modelagem, portanto, é uma linguagem para descrever modelos. Linguagens de modelagem geralmente utilizam diagramas para representar várias entidades e seus relacionamentos dentro do modelo. (STURM, 1999)

A UML foi criada para realizar as seguintes tarefas: (STURM, 1999)

- Representar todas as partes de um projeto que está sendo construído com técnicas orientadas a objeto;
- Estabelecer uma maneira de conectar idéias, conceitos e técnicas de design genéricas com a criação de código orientado a objeto;
- Criar um modelo que possa ser compreendido por humanos e também por computadores – para que um computador possa gerar uma boa parte da aplicação automaticamente.

A linguagem UML tem diversos diagramas e modelos usados para descrever o problema e a solução orientados a objeto. Descrevemos alguns deles, de maior importância:

- Modelos Use-Case
- Diagramas de interação
- Diagramas de atividade
- Diagramas de classe

6.6.1 Modelo Use-Case

O modelo *use case* traduz a necessidade do usuário em um modelo fácil de compreender. O usuário em questão pode ser um indivíduo ou um sistema externo (ou

outro agente, neste caso) e é conhecido como ator. Ou seja, o modelo use case é uma representação de como o sistema, ou parte dele, funciona do ponto de vista do ator. (STURM, 1999)

A Figura 8 mostra um exemplo de um use case.

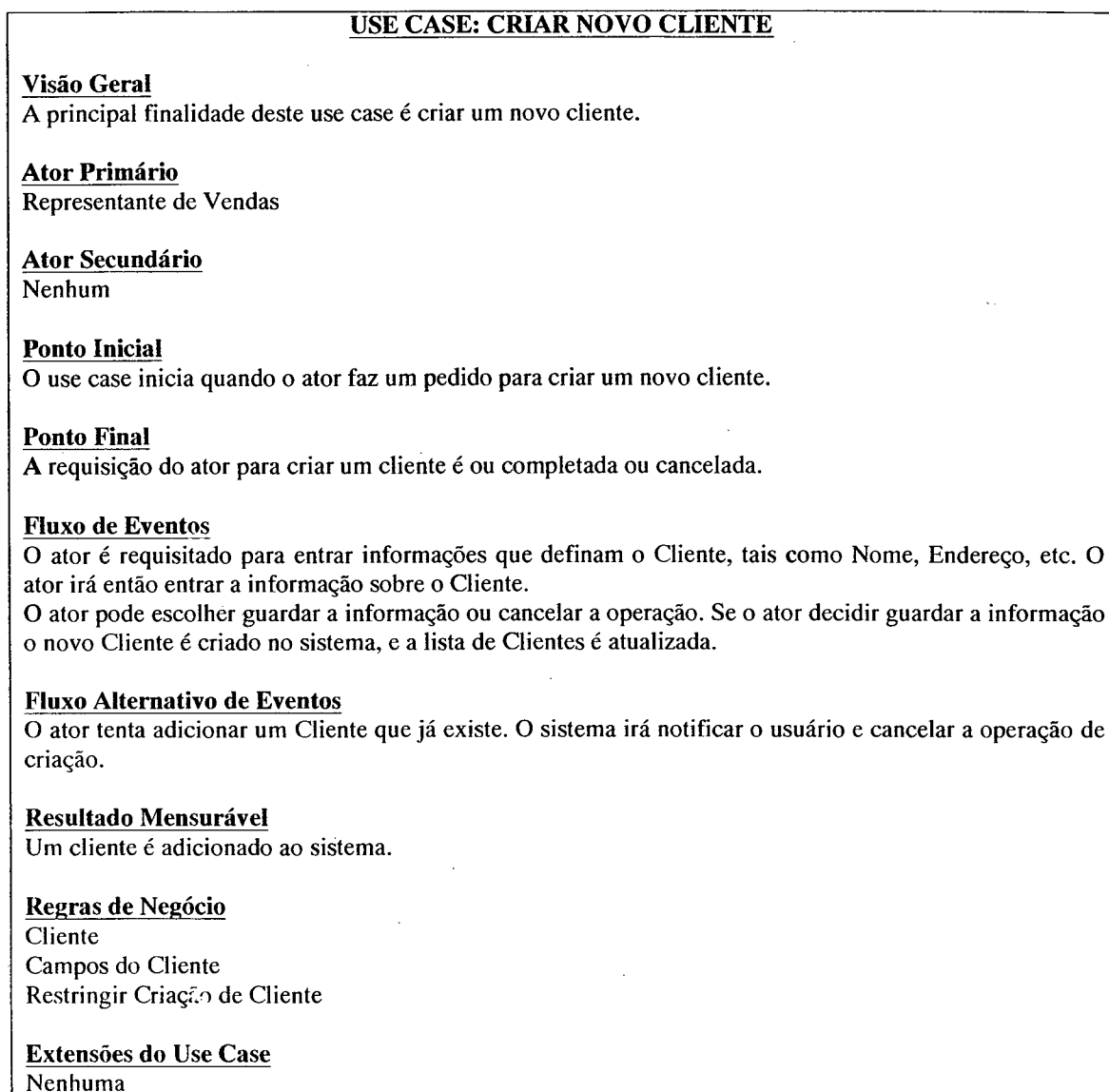


Figura 8 - Exemplo de use case UML (Fonte: STURM (1999))

6.6.2 Diagramas de interação

Os diagramas de interação são o próximo passo no processo de design UML. Eles se concentram em mostrar como os objetos ou coisas no sistema interagem um com o

outro para dar uma visão dinâmica do sistema. Há dois tipos de diagramas de interação: os diagramas de *seqüência* e de *colaboração*.

6.6.2.1 Diagrama de seqüência

Serve para converter os modelos use case em modelos visuais mais claros. Ele mostra como os objetos associados a um use case específico podem se comunicar entre eles e com os usuários ao longo do tempo. A Figura 9 mostra um exemplo de diagrama de seqüência.

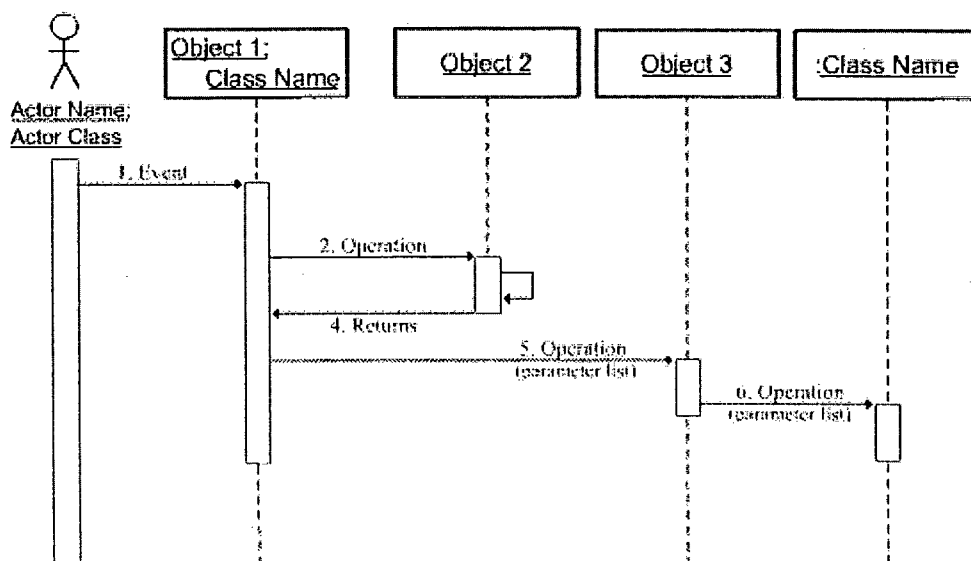


Figura 9 - Exemplo de Diagrama de Seqüência (Fonte: STURM (1999))

6.6.2.2 Diagrama de colaboração

Os diagramas de colaboração também são construídos a partir de use cases, mas eles enfatizam a distribuição espacial dos objetos envolvidos. Também mostra a seqüência de eventos utilizando números para identificar as mensagens. A Figura 10 mostra um exemplo de um diagrama de colaboração.

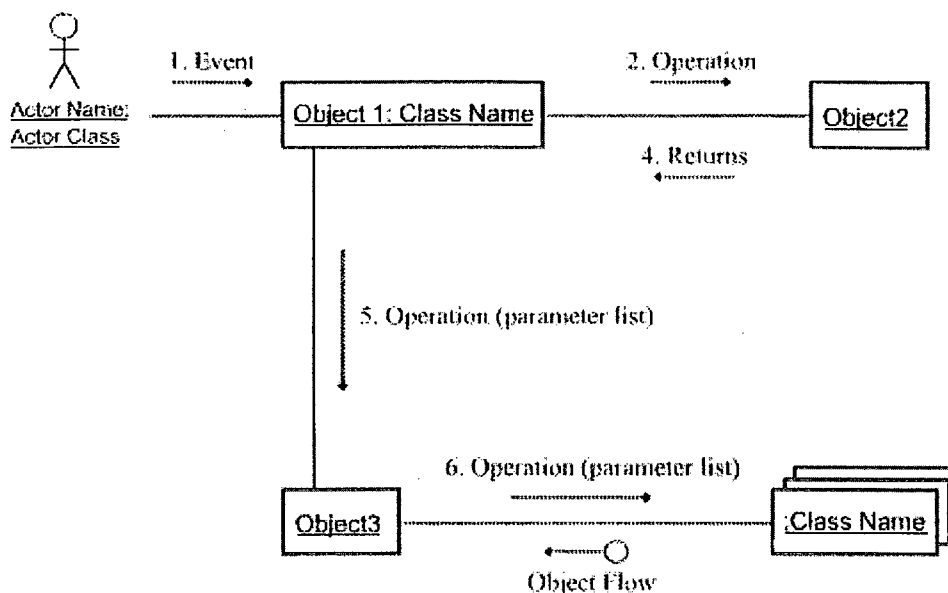


Figura 10 - Exemplo de diagrama de colaboração UML (Fonte: STURM (1999))

6.6.3 Diagramas de atividade

Os diagramas de atividade tomam a informação disponível dos diagramas de interação e a apresentam de uma forma mais detalhada. A finalidade dos diagramas de atividade é mostrar o funcionamento interno de um objeto em particular. A Figura 11 mostra um exemplo de um diagrama de atividade.

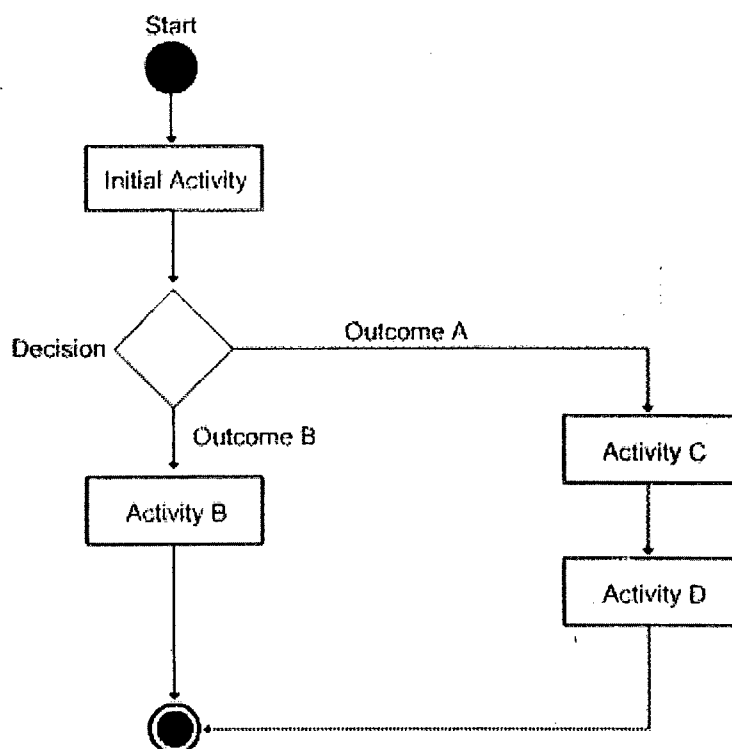


Figura 11 - Exemplo de diagrama de atividade UML (Fonte: STURM (1999))

6.6.4 Diagramas de classe

Os diagramas de classe, por último, representam as classes de objetos (ou agentes) que serão construídas. São a parte mais detalhada da especificação UML e freqüentemente pode-se fazer uma relação direta entre eles e o código que será escrito. A Figura 12 mostra um exemplo de como se parece um diagrama de classe UML.

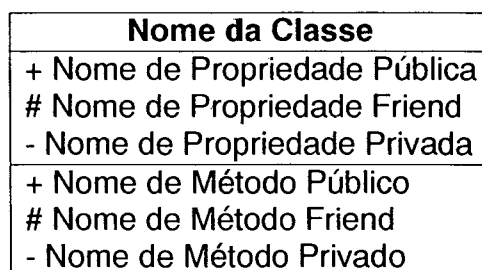


Figura 12 - Exemplo de diagrama de classe UML

6.7 A Extensão AgentUML

Era necessário que se utilizasse uma metodologia para análise e projeto dos agentes e da interação entre eles. ODELL et al. (2000) sugerem uma extensão da UML para representar agentes, o que eles chamaram de AgentUML (AUML).

Alguns grupos já desenvolveram metodologias para design de sistemas multi-agentes. Destacam-se os trabalhos de BRYSON & MCGONIGLE (1997) e a metodologia GAIA, proposta por WOOLRIDGE et al. (2000). Todos eles compartilham semelhanças com os mecanismos de design de objetos.

A UML é, algumas vezes, insuficiente para modelar agentes e sistemas multi-agentes (ODELL et al., 2000). Mas, por se tratar de um padrão já bastante adotado na indústria, que representa com bastante clareza as características de sistemas orientados a objeto, e por permitir propostas de extensões, torna-se uma plataforma bastante favorável para a construção de um padrão para a descrição de sistemas multi-agentes.

6.7.1 Descrevendo um sistema multi-agentes

Para melhor descrever sistemas multi-agentes como um todo, ODELL et al. (2000) sugerem a utilização das seguintes estruturas:

- Packages
- Templates

6.7.1.1 Packages

Já que os protocolos de interação entre agentes obedecem padrões, eles podem ser tratados como objetos agregados reutilizáveis de processamento (ODELL et al., 2000).

Em UML existem duas maneiras de expressar essa agregação, através de *componentes*

ou de *packages*. Os componentes descrevem uma agregação física para fins de implementação. Os *packages* agregam elementos de modelagem.

6.7.1.2 Templates

Um template é um elemento parametrizado de modelagem cujos parâmetros são instanciados em tempo de modelagem, ou seja, quando o novo modelo customizado é produzido. Pode-se definir um diagrama de seqüência genérico, com parâmetros, e depois reutilizá-lo em situações diferentes, com novos valores.

6.7.2 Descrevendo as interações entre os agentes

Para descrever as interações entre os agentes, a AUML se utiliza dos seguintes diagramas:

- Diagramas de Seqüência
- Diagramas de Colaboração
- Diagramas de Atividade
- Gráficos de Estado

6.7.2.1 Diagramas de Seqüência

Os diagramas de seqüência no AUML são semelhantes aos diagramas de seqüência em UML. Entretanto agora aparece a capacidade do agente de assumir papéis (role change). Um mesmo agente pode em períodos diferentes do tempo estar em papéis diferentes, e os diagramas AUML de seqüência permitem isso. A Figura 13 mostra um exemplo de um diagrama de seqüência AUML:

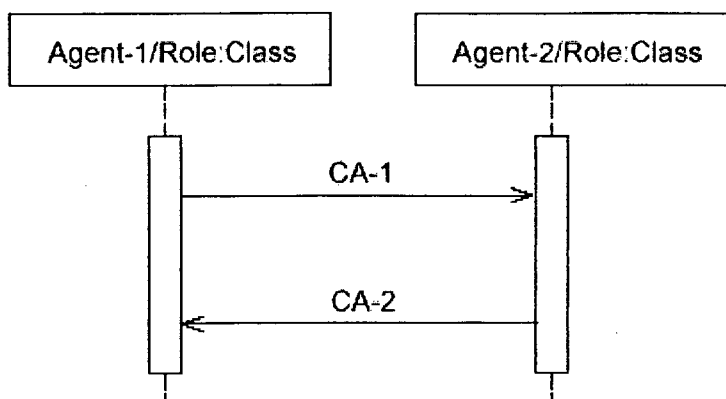


Figura 13 - Diagrama de Seqüência AUML (Fonte: ODELL et al. (2000))

Nota-se que a representação dá suporte à característica que os agentes têm de representar papéis (roles). Além disso, muitas vezes os agentes enviam mensagens simultâneas, e a Figura 14 mostra como representar isso em AUML.

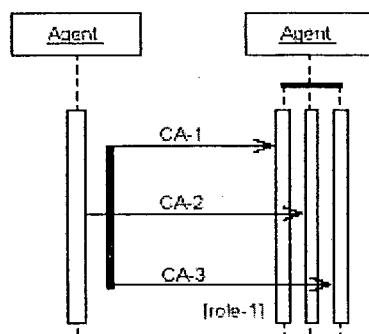


Figura 14 - Mensagens Simultâneas em AUML (Fonte: ODELL et al. (2000))

6.7.2.2 Diagramas de Colaboração

Os diagramas de colaboração mostram quase a mesma informação que os diagramas de seqüência, mas não enfatizam o tempo, e sim a distribuição espacial.

Segundo ODELL et al. (2000), os diagramas de colaboração complementam os diagramas de seqüência, pois dependendo do agente e do protocolo de interação que está sendo descrito, um tipo de diagrama pode fornecer uma visão mais nítida e uma representação mais compreensível do que o outro.

A Figura 15 mostra um exemplo de um diagrama AUML de colaboração.

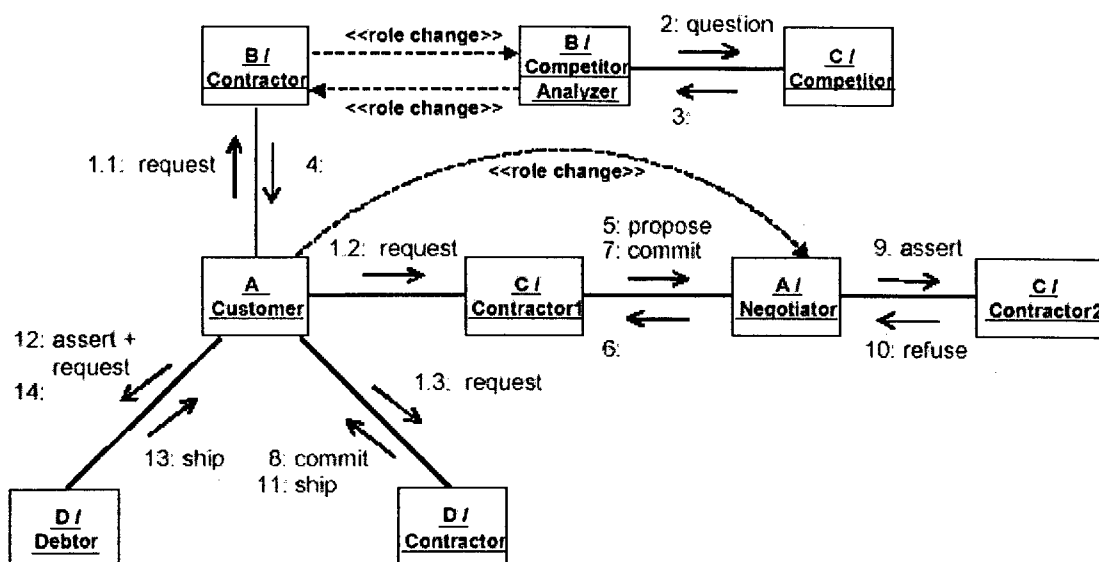


Figura 15 - Exemplo de diagrama de colaboração AUML (Fonte: ODELL et al. (2000))

Nota-se, na Figura 15, o suporte da representação às mudanças de papéis que os agentes podem realizar (role change).

6.7.2.3 Diagramas de Atividade

Um diagrama de atividade expressa operações e os eventos que as dispararam. A Figura 16 mostra um exemplo de um diagrama de atividade UML.

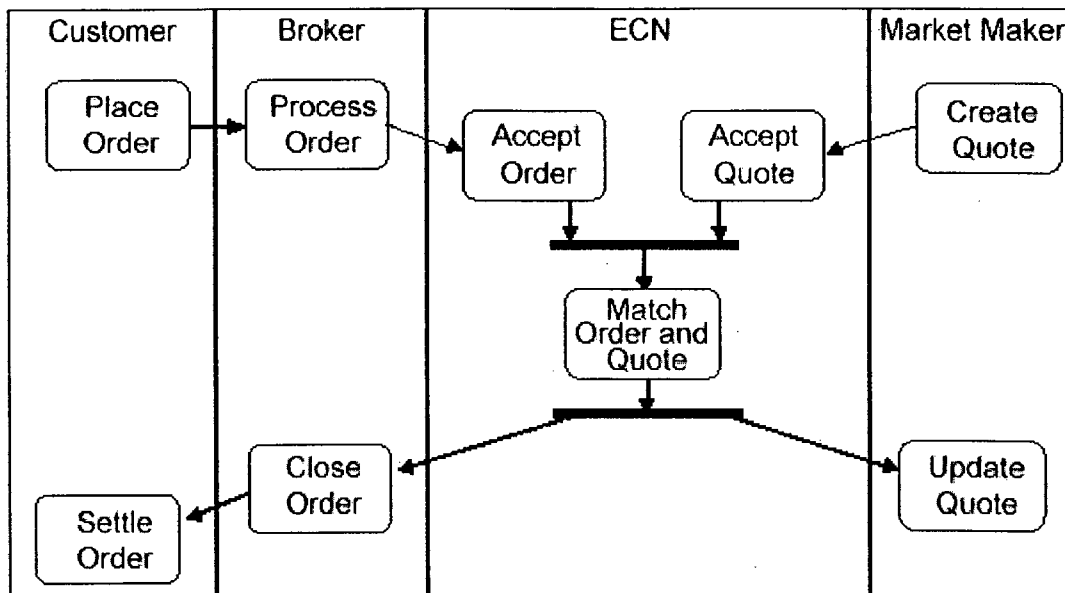


Figura 16 - Exemplo de diagrama de atividade AUML (Fonte: ODELL et al. (2000))

6.7.2.4 Gráficos de Estado

Um outro diagrama AUML relacionado a processos é o gráfico de estado (statechart). Um gráfico de estado representa uma máquina de estado. Estados são representados como retângulos com os cantos arredondados, e as transições entre eles são geralmente representadas por arcos direcionados interconectando os estados. A Figura 17 mostra um exemplo de um gráfico de estado AUML.

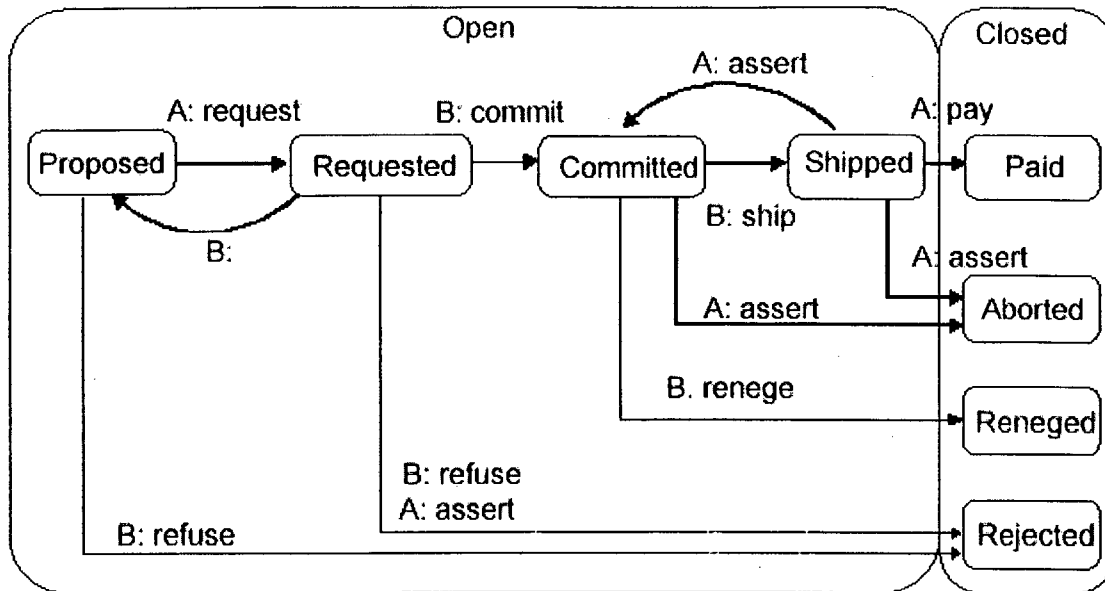


Figura 17 - Exemplo de Gráfico de Estado AUML (Fonte: ODELL et al. (2000))

6.7.3 Descrevendo o funcionamento interno dos agentes

Para descrever o funcionamento interno dos agentes, a AUML permite o uso dos seguintes diagramas, já descritos no item anterior:

- Diagramas de atividade
- Gráficos de estado

6.8 Modelo dos Agentes

Os agentes foram modelados usando AUML. A Figura 18 mostra o diagrama de classes.

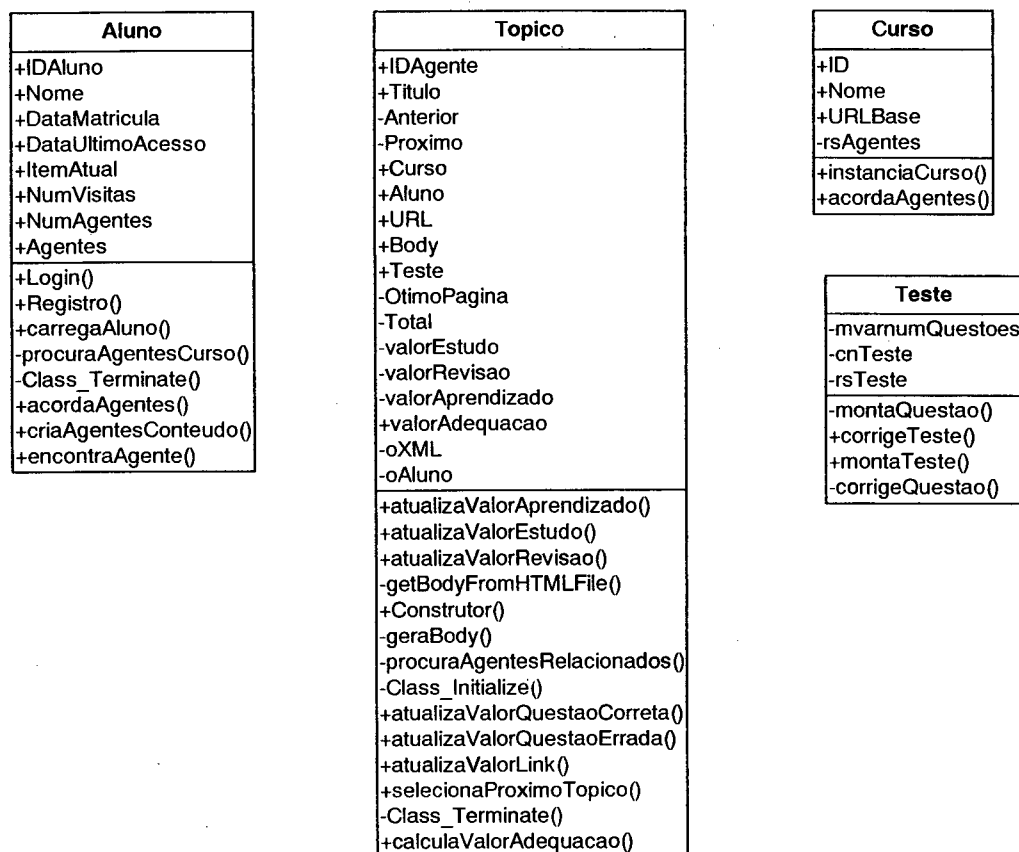


Figura 18 – Diagrama Lógico de Classes

6.9 Inferência e raciocínio dos agentes

Todo o raciocínio empregado pelos agentes é do tipo baseado em regras, fazendo com que cada agente seja um pequeno sistema especialista. É claro que podem ser aplicadas técnicas mais avançadas de inteligência artificial, mas isso não era o objetivo deste trabalho. Assim também o trabalho se limitou a uma funcionalidade dos sistemas tutores inteligentes que é o planejamento instrucional.

O planejamento instrucional, ou seqüência de currículo, pode ser visto como um processo de duas partes: decidir os tópicos relevantes baseando-se no estado atual do modelo do estudante, e então selecionar o mais adequado.

Como o modelo do estudante nesta abordagem está espalhado entre os agentes, para saber se um aluno está pronto para aprender um tópico, o agente daquele tópico terá que determinar o “quão bem” o aluno está nos tópicos anteriores (pré-requisitos).

Para isto, será usado um coeficiente chamado de *valor adequação* do agente, que é calculado conforme o nível de aprendizado do aluno nos tópicos que precedem este agente. Este valor adequação é que indicará aos outros agentes se o tópico que este agente representa é adequado para ser aprendido naquele momento. O intervalo de valores válidos para esta variável, assim como também para todos os outros coeficientes utilizados no raciocínio, é o dos valores entre 0 e 1.

Cada agente também, por sua vez, mantém um valor que pretende indicar o nível de aprendizado do aluno no tópico, chamado de *valor aprendizado*. É a partir dos valores aprendizado dos agentes precedentes que um agente pode calcular o seu valor adequação.

O valor aprendizado pode levar em conta diversos fatores, como o desempenho do aluno em exercícios ou testes, velocidade com a qual ele navega pelo conteúdo e o número de acessos, informações fornecidas pelo aluno em um questionário prévio, entre outros.

Este valor aprendizado, nesta abordagem, é alcançado através de três variáveis: o resultado do aluno nos testes dinâmicos, o número de visitas aos tópicos e o tempo total gasto num tópico, derivado do proposto em (STERN & WOOLF, 1998).

6.9.1 Resultado de testes dinâmicos

O resultado nos testes, como já foi comentado anteriormente, é o indicador mais direto sobre o conhecimento do aluno sobre um tópico. Por isto ele terá um peso maior no cálculo do coeficiente.

Cada vez que o aluno termina de estudar um tópico, o agente deste tópico lhe redireciona ao agente Teste, para que lhe aplique um pequeno teste com questões objetivas sobre o assunto recém estudado.

6.9.2 Tempo estudado

O tempo gasto num tópico pode ser um indicador valioso. Se um aluno gasta muito pouco tempo ou tempo demais numa página, isso pode implicar que a compreensão total não foi alcançada. Mas esta presunção pode estar completamente equivocada, pois o aluno pode simplesmente esquecer a página aberta. E não há como distinguir o estudante que ficou lendo o conteúdo por muito tempo do que esqueceu o computador parado na mesma página.

Para nossa pesquisa, consideraremos que todo o tempo gasto em um tópico é tempo que o aluno passou lendo e estudando o material. Desconsideraremos os tempos de latência da rede ou abandono da página.

Portanto, para calcular um escore de tempo, o agente utiliza uma regra de tempo ótimo. O *tempo ótimo* é um tempo determinado pelo especialista no conteúdo, que significa a quantidade de tempo de estudo adequada para perfeita compreensão do tópico. Para cada tópico, o especialista determina este tempo ótimo.

O escore de tempo estudado é calculado fazendo-se uma comparação entre o tempo efetivamente utilizado pelo aluno no estudo do tópico e o tempo ótimo determinado pelo especialista, de acordo com a curva também proposta por STERN & WOOLF (1998) e mostrada na Figura 19.

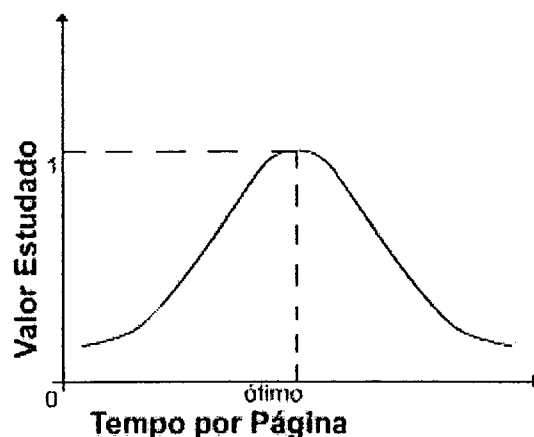


Figura 19 - Curva para cálculo do Escore de Tempo Estudado

Pela curva da Figura 19, nota-se que quanto mais o tempo estudado ficar perto do tempo ótimo, mais perto o valor estudado chegará de 1. Se o tempo estudado for muito curto ou muito longo, consideraremos que a compreensão não foi alcançada na totalidade, e por isso este escore, o valor estudado, será baixo.

6.9.3 Revisões

Será interessante guardar registros da frequência com a qual um aluno sente a necessidade de revisar um determinado material. Em geral, se ele retorna com frequência, ele não tem retido informação suficientemente, e então ainda não aprendeu o material. É claro que isso pode variar de acordo com o estudante, mas atualmente isso não está sendo levado em consideração.

O indicador de revisão, ou *valor revisão*, registra o número de vezes que o estudante retornou para visitar o mesmo tópico novamente, assim como quanto do material foi revisto a cada vez.

Assim como o modelo proposto por STERN & WOOLF (1998), o valor revisão inicia sempre com um valor igual a 0,1. A cada vez que o aluno visita o tópico, o valor revisão é atualizado, com base no escore de tempo estudado descrito anteriormente, e utilizando as Regras de Atualização do Valor Revisão descritas no Apêndice 1.

6.9.4 Valor aprendido

Combinando os três escores (resultado do teste, tempo estudado e revisão) é possível chegar num coeficiente de aprendizado para aquele estudante no objeto de conteúdo em questão. Usa-se uma média ponderada. Este coeficiente é chamado de *valor aprendido*. Para este modelo, foram utilizadas as regras propostas por STERN & WOOLF (1998):

valor aprendido \leftarrow 0,6 * resultado teste + 0,25 * valor estudado + 0,15 * valor revisão

Há um problema com esta equação. Se um aluno estudou um tópico e foi bem nos testes, mas não precisou revisar o tópico, ele é penalizado. Então, para o caso quando o valor estudado é maior do que 0,7, e o resultado no teste também é maior do que 0,7, esta equação é substituída pela seguinte:

valor aprendido \leftarrow 0,7 * resultado teste + 0,2 * valor estudado + 0,1 * (1 - valor revisão)

6.9.5 Cálculo do valor adequação

A partir daí, pode-se escolher qual o próximo tópico a ser visitado pelo estudante, analisando todos os tópicos relacionados, levando em conta o coeficiente de aprendizado em cada um deles e o tipo de relação (pré-requisito, relacionado, co-requisito e reforço) definida na organização do conteúdo em rede semântica descrita anteriormente. Por exemplo, um aluno deveria mostrar maior conhecimento em um pré-requisito direto do que em um tópico relacionado.

Para cada tipo de link, definiu-se um limite a partir do qual considera-se que o aluno domina o assunto suficientemente naquele contexto. Se o tipo de link é pré-

requisito, este limite ou valor necessário é maior do que se o tipo de link fosse relacionado. Os valores necessários estão definidos na Tabela 1.

Tipo de link	Valor necessário
Pré-requisito	0,8
Co-requisito	0,65
Relacionado	0,5
Reforço	0,8

Tabela 1 - Valores necessários por tipo de link

Se o valor aprendido é superior que o valor necessário estipulado na Tabela 1, não é necessário nenhum ajuste e ele mesmo é utilizado na escolha do próximo tópico. Mas quando o valor aprendido fica abaixo do valor necessário, na tentativa de não se ter limites tão estritos, STERN & WOOLF (1998) propõem um ajuste, a partir das Regras para Cálculo do Valor Ajustado descritas no Anexo 2.

No final os links que tiverem os valores atualizados mais altos, aparecerão com maior prioridade para o próximo passo do estudante, e basta ao agente selecionar o de valor mais alto.

Vale lembrar que os valores utilizados são empíricos, derivados do sugerido por STERN & WOOLF (1998).

6.10 Comunicação entre os agentes

No raciocínio do sistema demonstrado, é fundamental a troca de informações entre os agentes objeto de conteúdo. Já que a inteligência está distribuída pela comunidade de agentes, cada vez que o aluno interage com o sistema implica em uma ou mais trocas de mensagens entre os agentes.

Não foi necessário desenvolver ou mesmo utilizar uma linguagem ou protocolo de comunicação entre os agentes, já que no modelo os agentes não são móveis nem funcionam em processos distintos. Não existe comunicação inter-processo, mas a arquitetura pode ser implementada de forma distribuída, e as mensagens podem ser transformadas para se adequarem a algum padrão de comunicação de agentes, como o KQML. Este modelo se preocupou mais com o tipo de mensagens e o conteúdo transmitido do que com a forma.

Para ilustrar as trocas de mensagens que foram definidas entre os agentes será feito o uso de dois tipos de diagramas AUMML, que são os diagramas de colaboração e de seqüência.

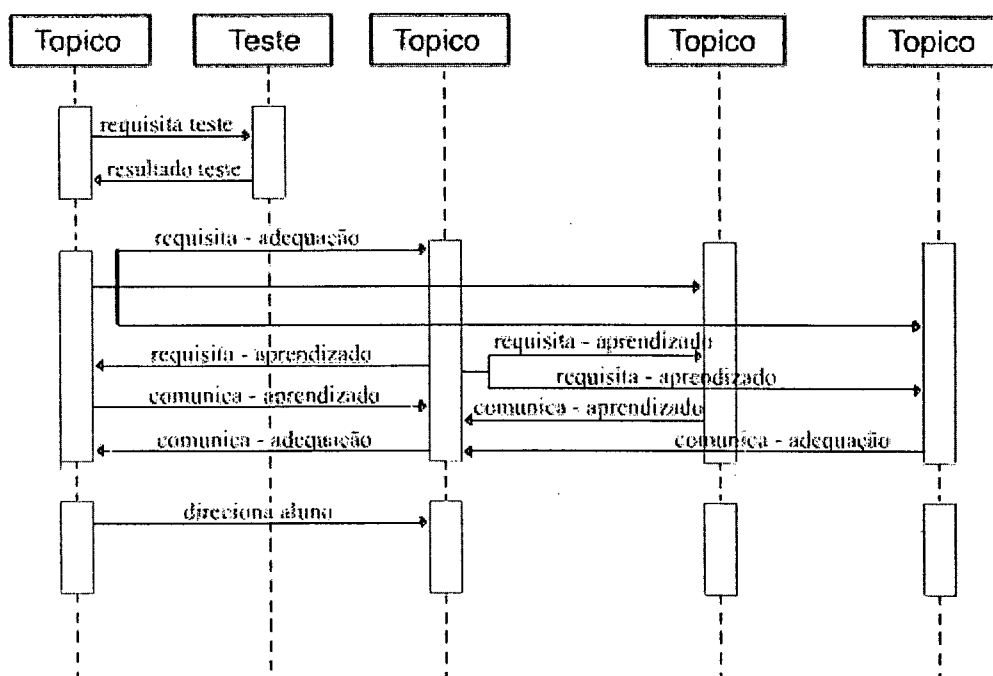


Figura 20 - Diagrama AUMML de seqüência da seleção do próximo tópico

A Figura 20 é um diagrama AUMML de seqüência, e mostra toda a troca de mensagens que acontece entre os agentes para a seleção do próximo tópico, o tópico que seria o mais adequado para ser estudado pelo aluno naquele momento.

O diagrama de seqüência obedece uma ordem temporal, e deve ser lido de cima para baixo. O processo de escolha do próximo passo inicia quando o aluno termina de estudar um determinado conteúdo e é direcionado para o agente Teste, que lhe mostra um teste com questões objetivas. Cada barra vertical representa uma *ativação* do agente. Para se descrever as ativações, utiliza-se geralmente diagramas Gráfico de Estado. A Figura 21 mostra um diagrama Gráfico de Estado que descreve o funcionamento interno do agente Teste quando recebe esta mensagem.

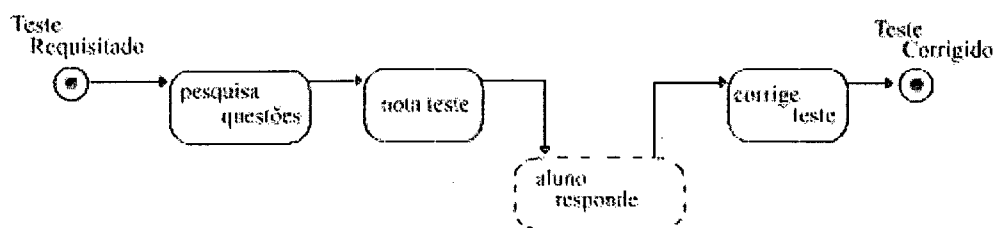


Figura 21 - Diagrama Gráfico de Estado do Agente Teste

Quando este teste é respondido, de posse dos resultados, o agente Tópico é enviada uma mensagem para todos os agentes Tópico que estão relacionados a ele, ou seja, para os quais o tópico que ele representa é um pré-requisito, co-requisito ou relacionado, requisitando qual o seu valor adequação. A Figura 20 ilustra um exemplo onde ele se comunica com três agentes.

Cada um destes agentes Tópico que recebe a mensagem, determina e ajusta o seu valor adequação de acordo com o diagrama Gráfico de Estado mostrado na Figura 22, e com as regras definidas na Seção 6.9.

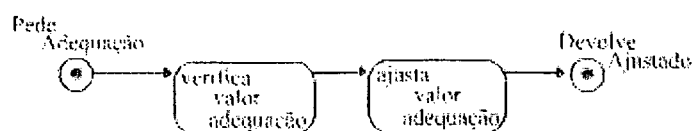


Figura 22 - Processo de ajuste do valor adequação

Pode-se continuar especificando cada vez mais os diagramas para se detalhar mais o funcionamento dos agentes. Na Figura 22, aparece um bloco **verifica valor adequação**, que por si só é um outro processo de troca de mensagens.

Continuando no diagrama de seqüência da Figura 20, podemos verificar o que cada agente faz para verificar o seu valor adequação. Vê-se que cada agente precisa enviar mensagens para todos os agentes que representam tópicos anteriores ao seu, requisitando destes agentes os seus respectivos valores aprendizado. O valor aprendido representa o nível de aprendizado do aluno naquele assunto e foi definido na seção 6.9.4.

Depois que cada agente determina o seu próprio valor adequação, ele retorna a mensagem de solicitação, e o agente solicitante pode agora selecionar o próximo tópico a apresentar ao aluno, como mostra o processo ilustrado na Figura 23.

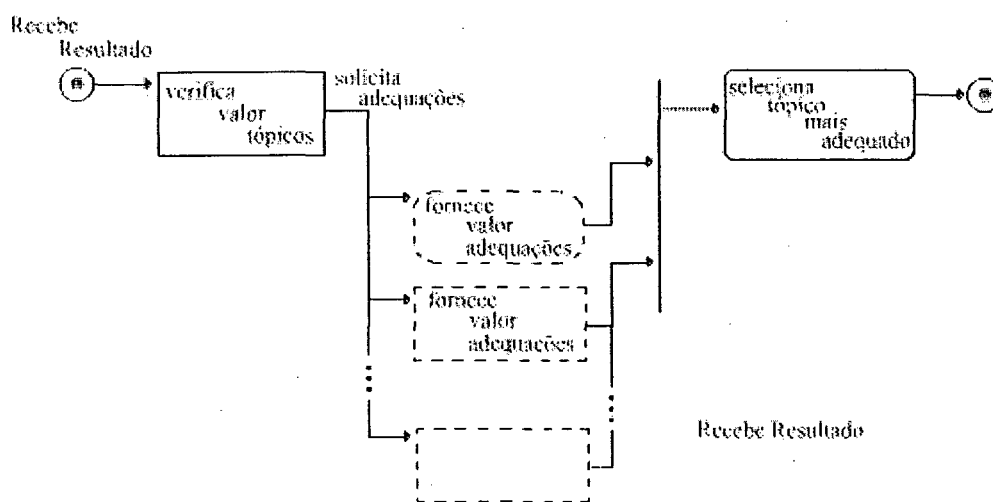


Figura 23 - Processo de seleção do melhor tópico

6.11 Modelagem do Banco de Dados

Para a persistência dos dados, foi utilizado um banco de dados relacional. Este foi modelado de acordo com o modelo definido para os agentes. Existe uma tabela para cada tipo de agente: Aluno, Curso, Tópicos, Testes, e tabelas relacionando-os: AlunoCurso e Links. Além disso há uma tabela para guardar os Feedbacks, retornos dos alunos sobre o curso. A Figura 24 mostra um diagrama entidade-relacionamento da estrutura lógica do banco de dados.

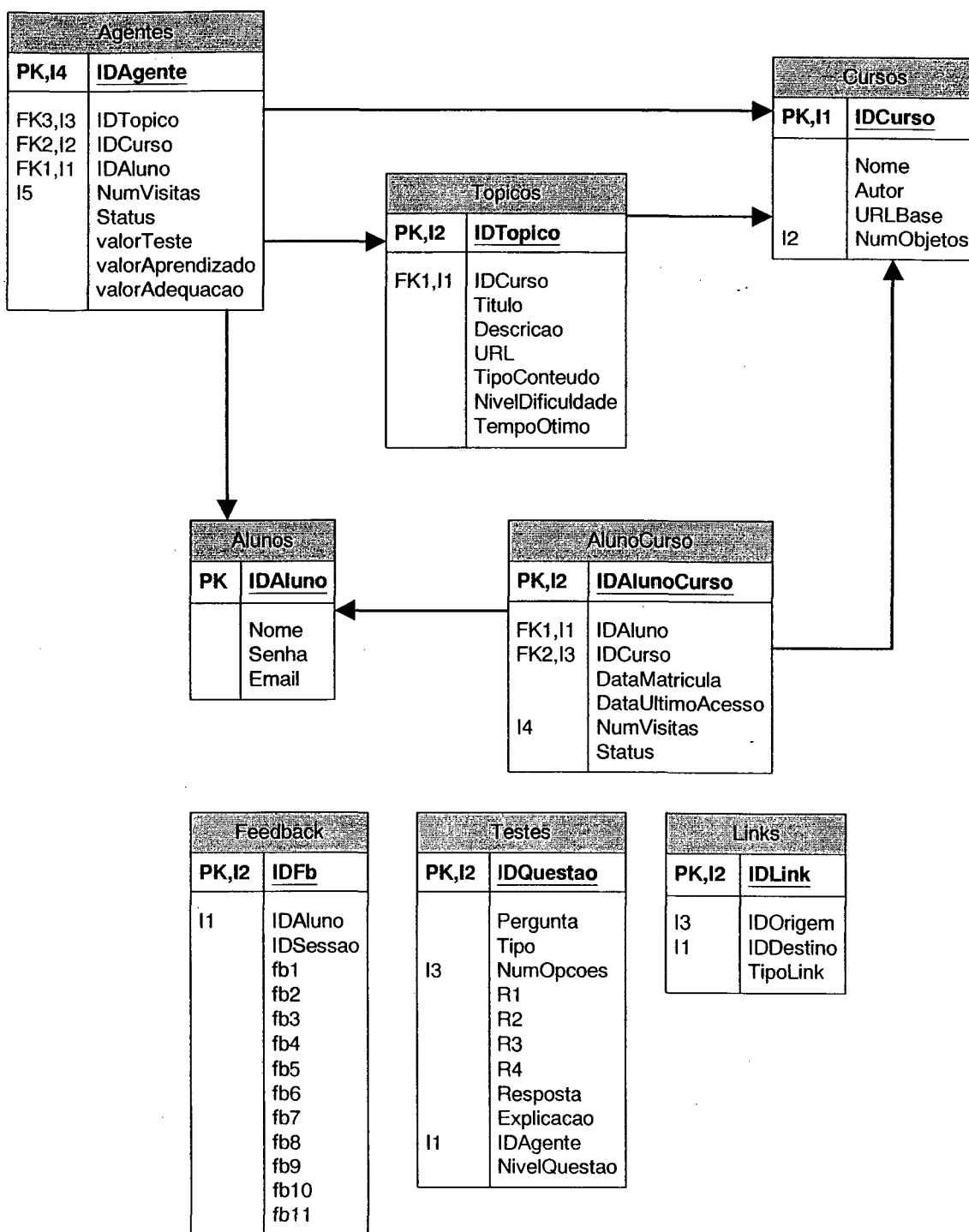


Figura 24 - Modelo lógico do banco de dados

Segue-se uma pequena descrição das tabelas:

Aluno: Guarda as informações relativas ao agente Aluno, como por exemplo seu nome, e-mail e senha.

Curso: Guarda as informações relativas ao agente Curso, como por exemplo o nome do autor, o título e a quantidade de tópicos.

Tópico: Guarda as informações relativas a cada objeto de conteúdo, como por exemplo a localização da mídia (arquivo), a descrição do conteúdo, etc.

Agente: Guarda as informações relativas aos agentes Tópico, como por exemplo, qual o tópico relacionado, e o número de visitas recebidas.

AlunoCurso: Guarda as relações de alunos e cursos.

Feedback: Guarda as informações relativas aos retornos fornecidos pelos alunos ao terminarem o estudo do conteúdo.

Testes: Guarda as informações relativas aos testes, como questões, respostas, explicações. Utilizada pelo agente Teste.

Links: Guarda todas as relações entre os objetos de conteúdo, utilizada pelos agentes Tópico.

7 Implementação do Modelo

7.1 Introdução

A implementação deste modelo foi gerenciada como um projeto de software, e isso implica algumas regras que foram seguidas. Adotou-se a Metodologia Cíclica sugerida por (STURM, 1999). Segundo STURM, um projeto de software deve ter quatro fases:

- Visão
- Design
- Desenvolvimento
- Instalação e testes

Juntas, elas formam um ciclo do projeto – um ciclo que pode ser repetido infinitamente enquanto o sistema evolui para adicionar novas funcionalidades.

7.1.1 Fase de Visão

Durante esta fase, o sistema é todo definido, através de dois aspectos: a visão e o escopo. A visão determina os objetivos de todo o sistema e fornece uma direção para todos os produtos que são necessários para criar o sistema. O escopo define os limites de cada projeto individual. Juntos, estes dois aspectos garantem que o problema está sendo resolvido, o que permite seguir para as próximas fases do projeto. Sem saber direito o que o sistema irá fazer, é impossível desenhar ou implementar o projeto.

7.1.2 Fase de Design

Consiste em criar uma série de documentos que definem os requisitos dos usuários, os componentes e serviços necessários para atender estes requisitos, e documentar um design detalhado destes componentes. Um projeto moderno de software na maioria dos casos terá um design orientado a objeto. Existem três estágios na fase de design: os estágios conceitual, lógico e físico. Durante o estágio de design conceitual, as necessidades dos usuários são analisadas e definidas. Uma descrição detalhada dos objetivos dos usuários é usada como base para a geração de uma solução detalhada. No estágio de design lógico, é criada uma descrição mais detalhada dos objetivos dos desenvolvedores, ou seja, uma transformação das necessidades dos usuários em um conjunto de serviços que o sistema deve ter para atender os requisitos deles. O design detalhado dos componentes que vão realizar estes serviços acontece no estágio de design físico. Isto converte os objetivos em um conjunto detalhado de documentos de design para os componentes do sistema.

7.1.3 Fase de Desenvolvimento

A fase de desenvolvimento consiste na construção e testes iniciais dos componentes e montagem destes componentes em uma aplicação final. É a construção dos componentes baseada no design feito na fase de design. É essencial que ao partir para esta fase, os documentos que definem a base de dados, os requisitos do sistema e as interfaces de usuário sejam “congelados”. Não se pode incluir novas funcionalidades nesta fase, pois se corre o risco de ter que reescrever o código, fazer adaptações desordenadas e não documentadas.

7.1.4 Fase de Instalação e Testes

Esta é a fase de estabilização, onde os testes finais são completados. O foco primário desta fase é encontrar e consertar bugs no software. Se possível, testes já devem ser realizados com a ajuda do usuário final da aplicação. O produto desta fase é o projeto completo.

As fases de visão e design foram documentadas no capítulo 5.

7.2 Plataforma

O modelo proposto pode ser implementado em qualquer plataforma de sistema operacional com servidor Web que dê suporte a aplicações.

Foi escolhida a plataforma do sistema operacional Windows 2000 e seu serviços Web, de nome Internet Information Services 5.0, pela ótima integração entre as aplicações Web e o sistema operacional.

O modelo foi todo implementado em forma de Dynamic Link Libraries (DLLs), que são arquivos binários, compilados, contendo classes e objetos. Estas DLLs integram-se ao processo servidor Web quando são requisitadas.

O sistema de banco de dados usado no protótipo é o Microsoft Access, mas o objetivo é utilizar qualquer banco de dados relacional, como o Microsoft SQL Server ou o Oracle, que têm maior capacidade para múltiplas requisições.

7.3 Ferramentas de Desenvolvimento

O conjunto de ferramentas escolhido, também levando em conta a grande integração com o sistema operacional, além da alta produtividade de desenvolvimento, foi o Microsoft Visual Studio 6.0.

O Visual Studio é composto das seguintes ferramentas de desenvolvimento:

- Visual Basic 6.0
- Visual C++ 6.0
- Visual J++ 6.0
- Visual FoxPro 6.0
- Visual InterDev 6.0

e algumas outras ferramentas de auxílio, como o sistema de controle de versão de código Visual SourceSafe.

Destas foram utilizadas o Visual Basic e o Visual InterDev.

O Visual Basic é a linguagem mais utilizada no mundo, por sua facilidade de desenvolvimento rápido de aplicações (RAD – Rapid Application Development), e vem constantemente evoluindo para acompanhar as tecnologias de desenvolvimento de software e tornar-se uma linguagem madura para desenvolvedores.

O Visual InterDev é a ferramenta mais completa para os desenvolvedores Web na plataforma Microsoft, pois ela permite, num mesmo ambiente, editar código VBScript, JavaScript, de cliente e de servidor, páginas HTML e ASP. Traz componentes Web, de navegação e acesso a dados, já prontos para serem utilizados. Dá ao desenvolvedor uma visão geral da aplicação, pois mostra no mesmo ambiente todos os arquivos que a compõem. Além de outras facilidades de trabalho em equipe e com banco de dados.

Quase todo o desenvolvimento foi feito em Visual Basic, pois toda a funcionalidade do sistema está nos agentes, que são compostos por classes criadas em Visual Basic.

O Visual InterDev foi usado ocasionalmente, para a montagem e manutenção da solução Web como um todo, e eventual confecção de scripts ou páginas HTML.

7.4 Recursos do Sistema

A função inicial do sistema é validar o modelo de representação do domínio através de agentes, visando um ambiente educacional adaptativo genérico no futuro. Não é um sistema completo, ou uma plataforma de ensino à distância.

Então, as funcionalidades do sistema foram limitadas para torná-lo factível e para que fosse possível cumprir o objetivo da pesquisa com ele. Não se desejava correr o risco de assumir um projeto faraônico de software, e ficar sem terminá-lo e sem tirar conclusões dali.

7.4.1 Funcionalidades

Inicialmente foi implementado um único curso, o de Técnicas de Programação I, com o mesmo conteúdo que é ministrado aos alunos de Engenharia de Telemática da UNISUL. Mas a plataforma foi desenhada para se adaptar a cursos de diversas naturezas.

O sistema tem as seguintes funcionalidades:

- Login e validação do usuário
- Montagem dinâmica das páginas de conteúdo
- Oferecimento de caminho personalizado para cada aluno

- Registro dos passos do aluno

7.4.2 Limitações

Como já foi comentado, o raciocínio dos agentes para guiar o aluno pelo melhor caminho foi simples, baseado em regras. Para validação do modelo, estamos usando as regras sugeridas por (STERN & WOOLF, 1998), descritas no Capítulo 5.

Não foi implementada uma interface para carregamento dos cursos no sistema, ou transformação de cursos já existentes para o padrão desta plataforma. O conteúdo tem que ser copiado manualmente para o diretório, e os links entre os assuntos estabelecidos diretamente no banco de dados que guarda as informações dos agentes.

Também não temos a interface de gerenciamento, utilizada pelo professor, para obter informações sobre o progresso dos alunos. A idéia é que esta representação possa ser utilizada em outras plataformas de ensino pela Web, então não vale a pena desenvolver tal funcionalidade, pois se estaria fugindo do objetivo.

7.5 Funcionamento

No momento que um aluno entra no sistema, a comunidade de agentes é acordada, e os dados de histórico do aluno são trazidos do banco de dados pelos agentes.

Na tela de login, o aluno se identifica, e se for um aluno novo, tem a opção de se registrar e receber uma senha de acesso. A Figura 25 mostra a tela de login.

<h1>Protótipo</h1>	
<p>Um Modelo Multi-Agentes para Representação de um Sistema Educacional via Web</p>	
<p>Marcel Alexandre Cabral Ribas</p>	
<p>Prof. Fernando Alvaro Ostuni Gauthier, Dr. Orientador</p>	
<p>Entre com seu login e senha, então clique em "Login"</p>	
<p>Este é um site seguro. Você deve identificar-se antes de prosseguir.</p>	
Login:	<input type="text" value="marcel"/>
Senha:	<input type="password" value="rodotox"/>
<input type="button" value="Login"/>	<input type="button" value="Registro"/>

Figura 25 - Tela de login

A montagem das páginas é realizada pelos agentes, mas não se tem ainda um desenho mais agradável para a interface. O agente extrai o conteúdo de um arquivo HTML através de um parser HTML/XML, e gera uma página para aquele usuário específico, montando os links e layout.

O conteúdo da página pode ser qualquer um dos tipos de conteúdo que um browser suporta: texto, imagens, GIFs animados, applets Java, Flash, etc.

A estrutura básica das páginas é mostrada na figura 2. O ambiente é mostrado em tela cheia, e os links para navegação ficam na parte inferior da página. Para definir o link os agentes fazem uso das regras definidas no Capítulo 6.

Sempre ao fim de cada tópico, o aluno é convidado a verificar seus conhecimentos sobre o assunto, através de um pequeno teste dinâmico.

Programação de Computadores

Se você está aqui iniciando este curso é porque se interessa pelo assunto. A esta altura já deve saber que o computador não é uma máquina inteligente. Ele pode fazer coisas maravilhosas, desde que seja bem instruído por um humano inteligente.

Como um humano pode instruir o computador? Como podemos fazer com que o computador execute uma seqüência de ações, ou operações matemáticas? Através da programação.

Programar um computador consiste em definir todas as operações que você deseja que ele realize. Um programa é um conjunto de instruções que faz com que o computador manipule dados.

Infelizmente o computador não entende comandos como:
"Computador, calcule a média de todos os alunos da turma",

nem, se pedirmos:
"Por favor, querido computador, calcule a média de todos."

Quem sabe um dia chegaremos a este nível.

Por mais evoluídas que sejam as linguagens de programação atuais, o computador entende apenas um conjunto limitado de comandos básicos. Conhecendo os comandos que ele entende, podemos usá-los para resolver nossos problemas.

Vamos verificar seu aprendizado: [Teste sobre Programação de Computadores](#)

Figura 26 - Exemplo de página de conteúdo

A Figura 27 mostra um exemplo de um teste dinâmico. As perguntas são sorteadas pelo agente encarregado dos testes ao ser requisitado pelo agente representando o tópico em questão.

Para efeito deste protótipo, os testes são compostos apenas de questões de múltipla escolha, com uma alternativa correta. Não se investiu em programar toda uma estrutura de testes, pois já existem outras pesquisas nesta área, e a intenção é buscar uma integração com um componente de testes já desenvolvido.

Vamos ver como você está em Identificadores da Informação:

Assinale o identificador inválido:

- a. U2
- b. NOVE
- c. KM/L
- d. X

Assinale o identificador inválido:

- a. (X)
- b. B52
- c. L
- d. TESTE

Assinale o identificador válido:

- a. AB°C
- b. NOVO-TESTE
- c. K7
- d. NOTA/2

Figura 27 - Teste dinâmico

As respostas do teste são dadas logo em seguida, e o escore de teste mantido pelo agente automaticamente atualizado. Se o desempenho do aluno no teste for baixo, a ele é oferecida a opção de retornar ao conteúdo e estudá-lo mais uma vez. Se ele foi bem no teste, ele vai receber dos agentes a sugestão do próximo passo, de acordo com o modelo. A Figura 28 mostra um exemplo de uma tela de respostas de teste.

Respostas do teste

Pergunta: 1

Resposta errada.
A resposta correta é KM/L. A barra é o operador de divisão, não pode ser utilizado no nome de uma variável.

Pergunta: 2

Resposta correta!

Pergunta: 3

Resposta errada.
A resposta correta é K7. Nenhum operador matemático pode fazer parte do nome de uma variável.

Você não parece seguro neste conceito. Veja de novo: [Identificadores da Informação](#)

Figura 28 - Tela de respostas de teste

Nesta estrutura, o aluno vai navegando, sendo guiado pelos agentes que sugerem os links para os seus próximos passos.

Ao final do conteúdo, é pedido ao aluno que preencha um formulário de *feedback* (retorno), para que ele possa expressar seus sentimentos, sugestões e críticas em relação à experiência vivida. A Figura 29 mostra um exemplo do questionário de feedback.

A partir das informações deixadas pelos alunos, será possível desenvolver um estudo da eficácia do sistema no aprendizado ou instrução.

Obrigado por ter experimentado o nosso protótipo!

Feedback

Esperamos que tenha sido possível aprender o assunto através de nosso sistema. Mas para que cada vez mais possamos aumentar a eficácia do sistema para o ensino, precisamos saber a sua opinião. Então, por favor, responda estas perguntas sobre o nosso sistema.

1. Você se sente capaz de criar algoritmos agora que utilizou nosso curso? Qual a sua impressão?

2. A orientação fornecida nas páginas foi boa? Os elementos nas páginas são intuitivos? O que deve ser melhorado?

Figura 29 - Exemplo de tela de feedback

7.6 Desenvolvimento

Foram realizadas as seguintes atividades no desenvolvimento do protótipo:

- Implementação das classes dos agentes, com todos os atributos definidos na fase anterior;
- comunicação de informações entre os agentes;
- implementação das regras de raciocínio dos agentes;
- funções para apresentação dos links mais adequados para cada aluno.

7.7 Modelo Físico das Classes

O diagrama da apresenta o modelo físico das classes de agentes implementadas.

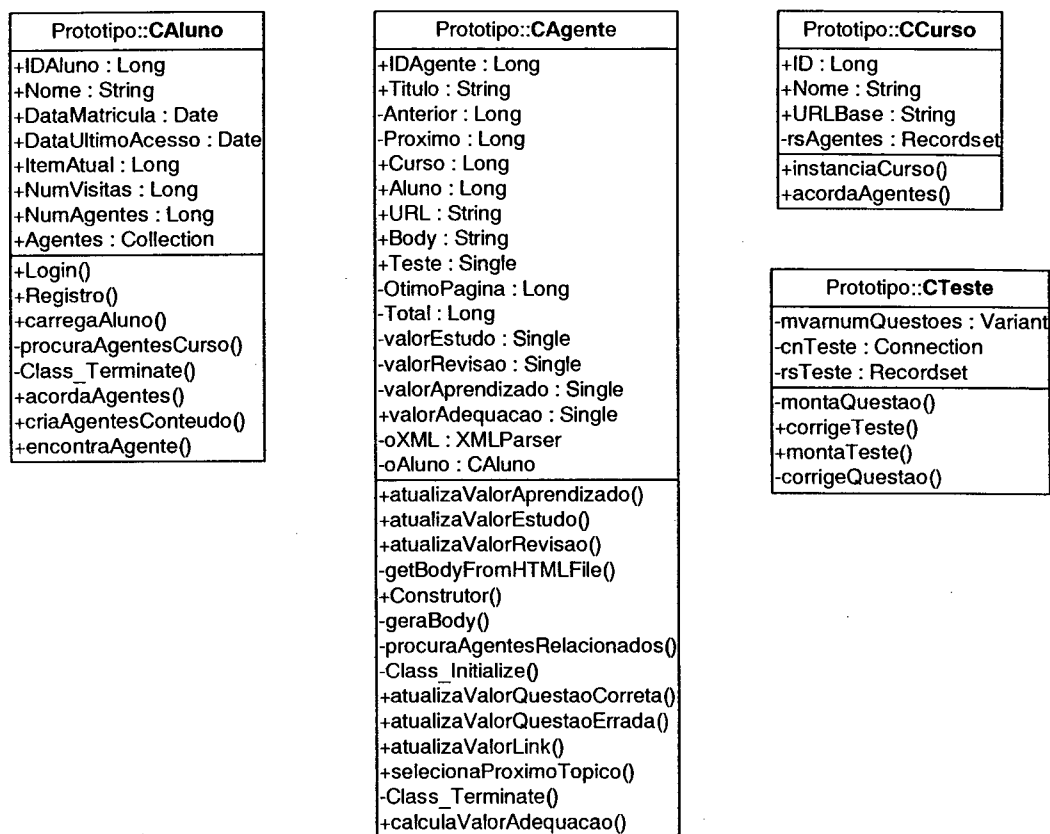


Figura 30 - Diagrama físico das classes do protótipo

7.8 Descrição das classes

A partir de agora, as classes implementadas e suas funcionalidades serão descritas brevemente.

7.8.1.1 Classe CAgente

Esta é a classe que mais é instanciada, pois para cada aluno, um número de agentes igual ao número de tópicos do curso é instanciado. Ela mantém como propriedades as principais informações, como o valor aprendido, o valor adequação e

o tempo estudado, e contém as principais funções do sistema, como o cálculo do valor aprendido e o cálculo e informação do valor adequação.

7.8.1.2 Classe CAluno

Corresponde ao agente responsável pela inicialização da comunidade de agentes, buscando dados na base para isso.

7.8.1.3 Classe CTeste

Define o agente responsável pela geração dos testes dinâmicos, quando requisitado pelos outros agentes.

7.8.1.4 Classe CCurso

Descreve o agente responsável pelas informações referentes ao curso. Futuramente poderá ser incrementada com funções de colaboração entre alunos.

7.8.1.5 Classe CXMLParser

Classe auxiliar, não está no diagrama da Figura 30, mas é utilizada pela classe CAgente para extrair dos documentos HTML que compõem cada unidade de conteúdo, apenas o que interessa, que é a parte <BODY> da página.

7.9 Modelo Físico do Banco de Dados

A Figura 31 apresenta o modelo físico do banco de dados.

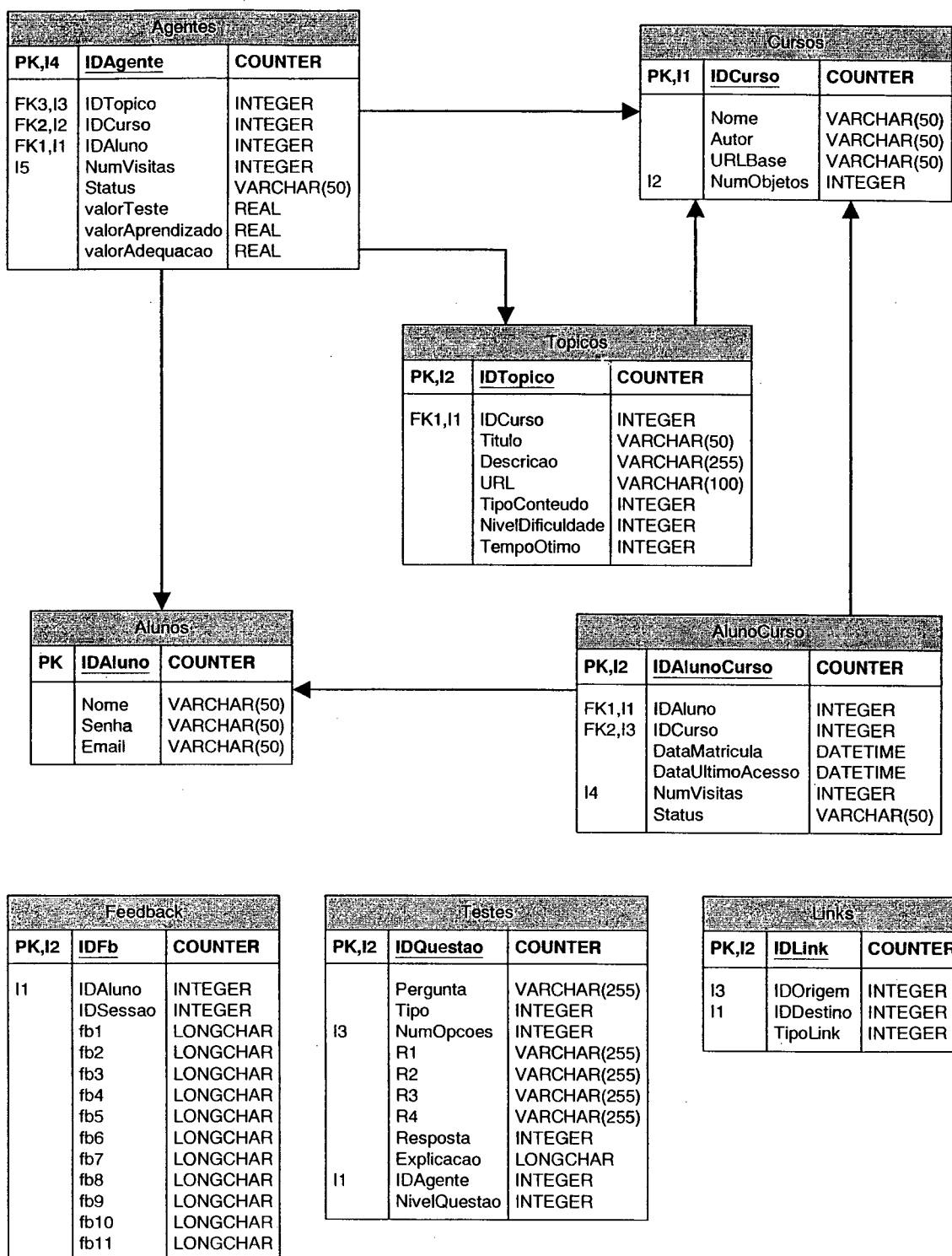


Figura 31 - Modelo físico do banco de dados

8 Conclusões

Ensinar e aprender pela Internet é realmente algo completamente diferente do que os professores e alunos desta geração estão acostumados. Provavelmente as crianças que nascem nos dias de hoje irão utilizar a Internet para aprender de uma forma muito mais eficaz do que hoje o fazemos. É que hoje, nem os professores, os designers instrucionais, ou os provedores de conteúdo, sabem quais recursos didáticos utilizar ao tentar transmitir o seu conhecimento a outras pessoas pela Web, e nem os alunos em geral se sentem confortáveis com este meio de comunicação.

Todas as teorias de aprendizado, técnicas instrucionais, modelos de ensino, que foram moldados durante séculos, agora têm que ser revistas, adaptadas, e às vezes até esquecidas, para a utilização da Internet como meio de aquisição de conhecimento. Ainda não é grande o número de profissionais do ensino que têm fluência na utilização dos computadores.

Os computadores, por sua vez, ainda são bastante limitados em sua “inteligência”, e pouco podem ajudar o aluno se este se vê em apuros e está sozinho. A natureza distribuída e independente da Internet, que impede que os alunos tenham sempre à disposição um tutor ou professor para tirar suas dúvidas, fez com que um número de pessoas e grupos passasse a estudar técnicas que pudessem dar um certo grau de suporte nos cursos pela Internet.

Viu-se a evolução dos Sistemas Tutores Inteligentes, que antes mesmo da Internet já buscavam auxiliar o aluno em seus estudos por conta própria, e simular o apoio de um tutor, dando dicas, mostrando o caminho ou corrigindo exercícios. Quando a Web

tornou-se popular, houveram muitas tentativas de adaptação da teoria de Sistemas Tutores Inteligentes para os cursos Web.

Por outro lado, também desenvolviam-se fortes estudos no campo da Hipermídia Adaptativa, que buscava reduzir os problemas que ocorrem quando o usuário se perde num ambiente hipermídia por causa da quantidade de informações, ou por causa de falsas direções e estrutura mal-formada. A Hipermídia Adaptativa utiliza um modelo de usuário, para adaptar o ambiente hipermídia a cada usuário, assim como os Sistemas Tutores Inteligentes utilizam o modelo de estudante. Percebeu-se que os dois campos tinham muito em comum.

E desta união entre duas linhas de pesquisa nasceram os Sistemas Educacionais Adaptativos, agora já completamente orientados para a Internet. Vários grupos de pesquisa pelo mundo estão desenvolvendo sistemas que empregam técnicas de inteligência artificial, teorias de aprendizado e métodos de ensino para tentar incentivar o desenvolvimento do conhecimento do aluno.

Infelizmente, quase a totalidade destes empreendimentos está restrita ao meio acadêmico. Alguns poucos projetos foram testados com alunos, mas também quase na totalidade, com alunos de universidades, e de ciências exatas, que já têm facilidade com o computador e só por isso já têm o incentivo necessário para utilizar tais sistemas.

O desenvolvimento de currículos para estes sistemas é uma tarefa ainda bastante complicada, exige não só que o designer instrucional seja um especialista no domínio de ensino/aprendizado, mas também que tenha conhecimentos avançados de computação. Ainda falta muito para que este tipo de sistema invada os ambientes de ensino não técnico.

O emprego da teoria de Agentes Inteligentes neste tipo de sistema provou ser bastante adequado e promissor. Este trabalho mostrou que é possível e eficiente se

representar um domínio de aprendizado através de um sistema multi-agentes. Foi demonstrado que a independência entre os agentes de conteúdo pode abrir uma arquitetura para a implementação de técnicas avançadas de inteligência artificial.

Os pontos mais importantes que se desejava verificar eram: saber se tal representação como a apresentada era possível de ser implementada, já que divergia um pouco das arquiteturas hoje utilizadas em Sistemas Tutores Inteligentes, e mais, se ela permitiria a utilização das mais diversas abordagens pedagógicas, sendo flexível para adaptar-se às técnicas de ensino desejadas.

Mesmo com um raciocínio simples baseado em regras, o ambiente proporcionado pelo protótipo do sistema multi-agentes já aparenta tornar o estudo um pouco mais agradável, pois tinha intuito de incentivar o aprendiz a ver o conteúdo, acompanhando-o e avaliando o seu desempenho.

Uma representação multi-agentes de um domínio de ensino/aprendizado abre um mundo novo no desenvolvimento de Sistemas Tutores Inteligentes ou Sistemas Educacionais de Hipermedia Adaptativa. Trata-se de uma plataforma que pretende permitir a utilização de, dentro de um mesmo curso ou currículo, diversas abordagens pedagógicas, suporte a vários tipos de aprendizes, definição de mais de um objetivo de ensino, e a adaptação dinâmica entre todas estas opções, dependendo do contexto, do aluno, do desempenho deste aluno e do ambiente em questão.

Como a tecnologia de agentes inteligentes permite mudanças no comportamento dos agentes, as chamadas mudanças de papéis, será possível criar diversos comportamentos ou papéis para os agentes, e fazer com que eles assumam este ou aquele papel dependendo do momento. Será possível também criar diferentes classes de agentes, para representar diferentes classes de alunos ou de objetivos instrucionais.

O mecanismo de inteligência é distribuído. Não se tem um módulo tutor, ou um módulo especialista. O agente aluno só é responsável por iniciar o sistema multi-agentes e acordar os agentes que representam o domínio. Ele não é responsável pelo raciocínio do sistema. Esta abordagem permite que os agentes possam também assumir diferentes mecanismos de raciocínio dependendo do caso. De acordo com a estratégia escolhida pelo designer instrucional, com o tipo de conteúdo (vídeo, áudio, animação, etc.), com o nível de dificuldade do item em questão, o agente pode usar regras diferentes para seu raciocínio, ou até mesmo técnicas de inteligência artificial distintas. Um agente pode aprender através de redes neurais, um outro através de redes Bayesianas, um terceiro pode ter o raciocínio baseado em casos e utilizar uma base de conhecimento, e assim por diante.

A colaboração entre agentes que representam alunos diferentes também pode ser explorada. Se um agente que representa determinado conteúdo e determinado aluno puder conversar com um agente de outro aluno que representa o mesmo conteúdo, eles poderão determinar padrões, ou detectar concepções falsas em comum, e assim alimentar uma base de conhecimento ou buscar desta base de conhecimento providências de tutoria que já provaram ter funcionado com outros alunos que tiveram as mesmas concepções errôneas.

Como o objetivo do trabalho era validar a representação do domínio, o protótipo oferecia um tutor que fornecia uma seqüência adequada de ensino, de acordo com as regras embutidas nos agentes e com a negociação entre eles. Pode-se dizer que este protótipo segue uma estratégia pedagógica comportamentalista, numa perspectiva de treinamento, não permitindo muito controle por parte do aluno. Como a atividade do aluno é toda planejada pelo tutor, pode-se dizer que não há a participação ativa do estudante, o que é combatido pelos educadores modernos.

Como uma pesquisa futura, sugere-se a validação do modelo utilizando técnicas de ensino diferentes, permitindo maior controle por parte do aluno. Não só a validação técnica do modelo, mas também a pedagógica, colocando um ambiente destes em utilização por alunos verdadeiros.

Outra sugestão para desenvolvimento seria a utilização de sistemas derivados deste com alunos de áreas não exatas, com conteúdo produzido por educadores destas áreas. Poderá se avaliar o comportamento do aluno que não tem a aptidão nata para o uso do computador para o aprendizado.

Enfim, este foi mais um passo na busca da utilização efetiva da Web para o aprendizado. Hoje existem muitos sistemas e iniciativas que promovem a instrução, mas poucos se preocupam com o desenvolvimento de uma consciência crítica através da exploração por parte do estudante, fatores dos quais depende o verdadeiro aprendizado. Este pode ser o início da construção de plataformas que permitam aos educadores aplicarem as teorias mais modernas e realmente promover o aprendizado via Web.

9 Referências Bibliográficas

- BECK, Joseph E., STERN, Mia K., HAUGSJAA, Erik. Applications of AI in Education, ACM Crossroads, September 1996. Disponível em: <<http://www.acm.org/crossroads/xrds3-1/aied.html>> Acesso em: 7 dezembro 2000.
- BECK, Joseph E., STERN, Mia K., WOOLF, Beverly Park. Using the student model to control problem difficulty. In Anthony Jameson, Cécile Paris, and Carlo Tasso (Eds.) User Modeling: Proceedings of the Sixth International Conference, UM97 (pp. 277-288). Vienna, New York: Springer Wien New York, 1997.
- BECK, Joseph E., STERN, Mia K. Bringing back the AI to AI & ED. In: Proceedings of the Ninth International Conference on Artificial Intelligence in Education. pp. 233-240. 1999. Disponível em: <<ftp://queequeg.cs.umass.edu/public/papers/aied99.ps>> Acesso em: 7 dezembro 2000.
- BELGRAVE, M. The Unified Agent Architecture: A White Paper, 1995. Disponível em: <http://www.ece.mcgill.ca/~belmarc/uaa_paper.html> Acesso em: 7 dezembro 2000.
- BERRYMAN, Sue E. Designing effective learning environments: cognitive apprenticeship models. Institute on Education and the Economy, New York. 1993. Disponível em: <<http://www.ilt.columbia.edu/k12/livetext/docs/berry1.html>> Acesso em: 7 dezembro 2000.
- BICA, F. Eletrotutor III: uma abordagem Multiagente para o ensino à distância. Dissertação de Mestrado. Orientadora: Rosa Maria Vicari. Universidade Federal do Rio Grande do Sul. Janeiro, 2000. Disponível em: <<http://www.inf.ufrgs.br/~francine/dissertacao/francine.zip>> Acesso em: 4 dezembro 2000.
- BOLZAN, R. O conhecimento tecnológico e o paradigma educacional. Dissertação de mestrado em Engenharia de Produção, Universidade Federal de Santa Catarina, Março, 1998. Cap. II. Disponível em: <<http://www.eps.ufsc.br/disserta98/regina/index.htm>> Acesso em: 7 dezembro 2000.
- BOSTOCK, S. J. Designing web-based instruction for active learning. In: Badrul Khan (org.) Web-based Instruction. New Jersey, 1998. c.26. p.225-230
- BRANDLE, S. Using Joint Actions to Explain Acknowledgments in Tutorial Discourse: Application to Intelligent Tutoring Systems. Ph.D., Thesis Illinois Institute of

Technology, 1998. Disponível em: <<http://www.csam.iit.edu/~circsim/index.html>>
Acesso em: 10 dezembro 2000.

BRUSILOVSKY, P. (1994a) Adaptive hypermedia: an attempt to analyse and generalise. Proceedings of the Workshop Adaptive hypertext and hypermedia, held in conjunction with UM'94 (Fourth International Conference on User Modeling). Hyannis, Cape Cod, Massachusetts, U.S.A. Aug 1994. Disponível em: <<http://www.wis.win.tue.nl/ah94/Brusilovsky.html>> Acesso em: 7 dezembro 2000.

BRUSILOVSKY, P. (1994b) Student model centered architecture for intelligent learning environments. In Proc. of Fourth international conference on User Modeling, 15-19 August, Hyannis, MA, USA. User Modeling Inc, 1994. 31-36

BRUSILOVSKY, P., SCHWARZ, E., WEBER, G. (1996a) A tool for developing hypermedia-based ITS on WWW. Position paper for the ITS'96 Workshop on Architecture and Methods for Designing Cost-Effective and Reusable ITSs. Montreal, Junho de 1996.

BRUSILOVSKY, P. Methods and techniques of adaptive hypermedia. User Modeling and User-Adapted Interaction Journal Vol. 6, pp. 87-129. Kluwer Academic Publishers. Holanda, 1996.

BRUSILOVSKY, P., SCHWARZ, E., WEBER, G. (1996b) ELM-ART: An intelligent tutoring system on World Wide Web. In Frasson, C., Gauthier, G., & Lesgold, A. (Ed.), Intelligent Tutoring Systems (Lecture Notes in Computer Science, Vol. 1086). Berlin: Springer Verlag, 1996. p.261-269.

BRUSILOVSKY, P. Adaptive educational systems on the world-wide web: a review of available technologies. In: Proceedings of Workshop "WWW-Based Tutoring" at 4th International Conference on Intelligent Tutoring Systems (ITS'98), San Antonio, TX, August 16-19, 1998.

BRUSILOVSKY, P, SCHWARZ, E, WEBER, G. Electronic textbooks on the World Wide Web: from static hypertext to interactivity and adaptivity. In Badrul Khan (org.) Web-based Instruction. New Jersey, 1998. c.30. p.255-261

BRUSILOVSKY, P, EKLUND, J. A study of user model based link annotation in educational hypermedia. Journal of Universal Computer Science, [S.L.], apr. 1998. v.4, n.4, p.429-448.

BRUSILOVSKY, P. Adaptive and Intelligent Technologies for Web-based Education. In: C. Rollinger and C. Peylo (eds.) Künstliche Intelligenz, Special Issue on Intelligent Systems and Teleteaching, 1999, 4, 19-25. Disponível em: <<http://www2.sis.pitt.edu/~peterb/papers/KI-review.html>> Acesso em: 7 dezembro 2000.

BRUSILOVSKY, P. InterBook Home Page: Adaptive educational hypermedia on the WWW. Disponível em: <<http://www2.sis.pitt.edu/~peterb/InterBook.html>> Acesso em: 12 dezembro 2000.

- BRYSON, J, MCGONIGLE, B. Agent Architecture as Object-Oriented Design. In: Proceedings of Intelligent Agents IV: Agent Theories, Architectures and Languages (ATAL '97), 1997.
- CAGLAYAN, Alper, HARRISON, Colin. Agent Sourcebook. John Wiley & Sons, Inc. 1997.
- CHAN, Tak-Wai. Learning companion systems, social learning systems, and the global social learning club. Based on an edited transcript of an invited talk at the World Conference on Artificial Intelligence on Education, Washington, DC, Agosto, 1995.
- CLIBBON, K. Conceptually adapted hypertext for learning. In: Proceedings of the 1995 Conference on Human Factors in Computing Systems (CHI95), Denver, Colorado, USA. May 7-11, 1995. Disponível em: <http://www.acm.org/sigchi/chi95/proceedings/shortppr/kc_bdy.htm> Acesso em: 7 dezembro 2000.
- COELHO, M. Definição de um módulo hiperídia adaptativo para integrar sistemas tutores inteligentes. Curso de Sistemas Tutores Inteligentes, Departamento de Engenharia Eletrônica do ITA – Instituto Tecnológico de Aeronáutica. 1996.
- COSTA, Marcelo Thiry Comicholi da. Uma arquitetura baseada em agentes para suporte ao ensino à distância. Tese submetida à Universidade Federal de Santa Catarina para a obtenção do título de Doutor em Engenharia de Produção, Florianópolis, SC, abril de 1999.
- COSTA, Rosa Maria E. Moreira da, ROCHA, Ana Regina C. da, SANTOS, Neide dos et al. Desenvolvimento de sistemas tutores inteligentes: questões e perspectivas. In Workshop em Sistemas de Tutoria Inteligente Aplicados a Educação e Treinamento, VIII Simpósio Brasileiro de Informática e Educação – SBIE, São José dos Campos, SP, 1997.
- COSTA, Rosa Maria E. Moreira da, WERNECK, Vera Maria B. Sistemas Tutoriais: Aplicação das Tecnologias de Hiperídia e de Inteligência Artificial em Educação. Relatório Técnico do Programa de Engenharia de Sistemas e Computação, COPPE – UFRJ, ES-427/97, março 1997.
- COSTA, Rosa Maria E. Moreira da. Desenvolvimento de Sistemas Tutores Hiperídia Inteligentes para Cardiologia. Relatório Técnico do Programa de Engenharia de Sistemas e Computação. COPPE – UFRJ. ES-440/97. Julho, 1997.
- DE BRA, P., CALVI, L. AHA! An open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia, vol. 4, pp. 115-139, Taylor Graham Publishers, 1998. Disponível em: <<http://wwwis.win.tue.nl/~debra/review/paper.html>> Acesso em: 8 dezembro 2000.
- DE BRA, P., BRUSILOVSKY, P., HOUBEN, G. Adaptive Hypermedia: From Systems to Framework. ACM Computing Surveys 31(4). Dezembro 1999. Disponível em: <<http://wwwis.win.tue.nl/~debra/survey/debra.ps>> Acesso em: 7 dezembro 2000.

- DE BRA, P. Design Issues in Adaptive Hypermedia Application Development. In: Proceedings of the Second Workshop on Adaptive Systems and User Modeling on the World Wide Web, pp. 29-39, Toronto and Banff, Canada, 1999. Disponível em: <<http://wwwis.win.tue.nl/asum99/>> Acesso em: 7 dezembro 2000.
- DE BRA, P. Curso 2L690:Hypermedia Structures and Systems. Exemplo de curso utilizando hipermídia adaptativa. Disponível em: <<http://wwwis.win.tue.nl/2L690/>> Acesso em: 9 dezembro 2000.
- DRESCHER, Gary L. Made-up minds: a constructivist approach to artificial intelligence. MIT Press, Cambridge, MA, EUA, 1996.
- FARACO, Rafael Avila. Uma arquitetura de agentes para negociação dentro do domínio do comércio eletrônico. Florianópolis, 1998. Dissertação para obtenção do grau de Mestre em Engenharia de Produção, Universidade Federal de Santa Catarina. Florianópolis, SC. 1998.
- FELDER, R. Matters Of Style. ASEE Prism, 6(4), 18-23. American Society for Engineering Education. Washington, DC, EUA. Dezembro, 1996. Disponível em: <<http://www2.ncsu.edu/unity/lockers/users/f/felder/public/Papers/LS-Prism.htm>> Acesso em: 10 dezembro 2000.
- FININ, T., LABROU, Y., MAYFIELD, J. KQML as a Communication Language. In: Jeff Bradshaw (Ed.), "Software Agents", MIT Press, Cambridge, Mass. EUA, 1997.
- FONER, Leonard N. What's an agent, anyway? A sociological case study. In Agents Memo 93-01, Agents Group, MIT Media Lab, Cambridge, MA, EUA, 1993. Disponível em: <<http://foner.www.media.mit.edu/people/foner/Julia>> Acesso em 7 dezembro 2000.
- FLEISCHHAUER, Luciana Irene Amaral. O uso da tecnologia de agentes na integração da programação da produção. Dissertação para obtenção do grau de Mestre em Engenharia de Produção, Universidade Federal de Santa Catarina. Orientador: Fernando Alvaro Ostuni Gauthier. Florianópolis, SC. 1996.
- GARCIA, Leticia Silva. Aplicações de sistemas multi-agentes a sistemas de hipermídia adaptativa – uma proposta de ampliação à ferramenta Gutemberg. Dissertação apresentada como requisito parcial à obtenção do grau em mestre em Informática. Pontifícia Universidade Católica do Rio Grande do Sul. Orientador: Prof. Dr. Flávio Moreira de Oliveira. Porto Alegre, RS. 1998.
- GIRAFFA, Lucia Maria Martins, VICCARI, Rosa Maria. Estratégias de Ensino em Sistemas Tutores Inteligentes modelados através da tecnologia de agentes. In: IX SBIE SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 1998, Fortaleza. IX SBIE Simpósio Brasileiro de Informática na Educação. Fortaleza: UFCE/SBC, 1998. v.1. p.260-269.
- GIRAFFA, Lucia Maria Martins, VICCARI, Rosa Maria. Uma arquitetura de tutor utilizando sistemas multiagentes: da modelagem a validação pedagógica. In: X SBIE

SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 1999, Curitiba.
X SBIE Simpósio Brasileiro de Informática na Educação. Curitiba: UFPR/SBC,
1999. v.1. p.169-178.

GENESERETH, Michael R., KETCHPEL, Steven P. Software Agents. Computer
Science Department, Stanford University. In Communications of the ACM, July
1994, pp. 48-53. Disponível em:
<<http://www.cs.nccu.edu.tw/~jong/reports/ps/sagents.ps.gz>> Acesso em: 7
dezembro 2000.

GUTIERREZ, J., PEREZ, T., LOPISTÉGUY, P. et al. Sistemas Tutores Inteligentes:
una Forma de conseguir Sistemas Hipermedia Educativos. In: Sistemas Hipermedia
Educativos: Conferencia de la Asociación Española para la Inteligencia Artificial
(CAEPIA '95). Asociación Española para la Inteligencia Artificial, AEPIA.
Alicante, España. 1995.

HALL, Lynne, GORDON, Adrian, Synergy on the Net: integrating the web and
intelligent learning environments. In: Proceedings of the Workshop on Intelligent
Tutoring Systems on the Web, "Web-based ITS" at ITS'98, 4th International
Conference in Intelligent Tutoring Systems, 16-19, San Antonio, Texas. August,
1998. Disponível em: <<http://www-aml.cs.umass.edu/~stern/webits/itsworkshop/>>
Acesso em: 7 dezembro 2000.

HEILMANN, Kathryn, KIHANYA, Dan et al. Intelligent Agents: a technology and
business application analysis. Novembro, 1995. Disponível em:
<<http://www.mines.u-nancy.fr/~gueniffe/CoursEMN/I31/heimann/heimann.html>>
Acesso em: 7 dezembro 2000.

HENZE, N. Adaptive Hyperbooks: Adaptation for Project Based Learning Resources.
PhD Thesis. Univerität Hannover, Alemanha. 2000. Disponível em:
<<http://www.kbs.uni-hannover.de/~henze/>> Acesso em: 8 dezembro 2000.

JOHNSON, W. Lewis, SHAW, Erin, GANESHAN, Rajaram. Pedagogical Agents on
the Web. In: Proceedings of the Third Int'l Conf. on Autonomous Agents (Agents
'99), pp. 283-290, Seattle, Washington. Maio, 1999. Disponível em:
<<http://www.isi.edu/isd/ADE/papers/agents99/agents99.htm>> Acesso em: 7
dezembro 2000.

KAY, J. and KUMMERFELD, R. J. (1994) An individualised course for the C
programming language. In: Proceedings of Second International WWW
Conference, Chicago, IL, 17-20 October, 1994. Disponível em:
<[http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Educ/kummerfeld/kummerfeld.h
tml](http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Educ/kummerfeld/kummerfeld.html)> Acesso em: 7 dezembro 2000.

KEARSLEY, Greg. A web site for theories of learning and instruction. In Badrul Khan
(org.) Web-based Instruction. Educational Technology Publications. Engewood
Cliffs, New Jersey, 1998. c.32. p.271-274

KOSLOSKY, Marco Antonio Neiva. Aprendizagem baseada em casos: um ambiente
para ensino de lógica de programação. Dissertação de mestrado em Engenharia de

Produção. Orientador: Fernando Alvaro Ostuni Gauthier. Universidade Federal de Santa Catarina, 1999.

KHUWAJA, Razman, DESMARAIS, Michel, CHENG, Richard. Intelligent Guide: combining user knowledge assessment with pedagogical guidance. In Proceedings of Intelligent Tutoring Systems, 1996. p.225-233.

KHUWAJA, Razman. A Model of Tutoring: Facilitating Knowledge Integration Using Multiple Models of the Domain. Ph.D. Thesis, 354 pp. Illinois Institute of Technology, EUA. 1994. Disponível em: <<http://www.csam.iit.edu/~circsim/index.html>> Acesso em: 10 dezembro 2000.

KING, R. Classification of Student Modeling Approaches for Intelligent Tutoring. Technical Report No. MSSU-COE-ERC-98-4. Engineering Research Center for Computational Field Simulation, Mississippi State University, EUA. Janeiro, 1998. Disponível em: <http://www.erc.msstate.edu/news_and_publications/tech_reports1998/ERC_98_4/> Acesso em: 11 dezembro 2000.

LABIDI, Sofiane, FERREIRA, Jeane. Technology-assisted instruction applied to cooperative learning: the SHIECC project. To appear at the Proceedings of the IEEE International Conference Frontiers in Education (FIE'98). Tempe, Arizona. November 4-7, 1998.

LIMA, Cynthia M., LABIDI, Sofiane. Modelagem da cooperação e das comunicações entre os agentes do sistema SHIECC. Nos Anais do Simpósio Brasileiro de Informática na Educação -SBBD'2000. Workshop Abordagem Multi-Agentes para os Sistemas Tutoriais Inteligentes, 1999.

MARIETTO, Maria das Graças B, OMAR, Nizam, FERNANDES, Clovis T. Tendências nas áreas de sistemas de tutoria inteligente e modelagem do aprendiz. In Workshop em Sistemas de Tutoria Inteligente Aplicados a Educação e Treinamento, VIII Simpósio Brasileiro de Informática e Educação – SBIE, São José dos Campos, SP, 1997.

MERGEL, Brenda. Instructional design & learning theory. In Professional and Theoretical Issues in Educational Technology: Occasional Papers, College of Education, University of Saskatchewan, Canada. Maio, 1998. Disponível em: <<http://www.usask.ca/education/coursework/802papers/mergel/brenda.htm>> Acesso em: 24 maio 2000.

MULLIER, D. J., MOORE, D. J. A Web based Intelligent Tutoring System. In: Proceedings of the 4th EATA International Conference on Networking Entities NETIES98: Networking for the Millennium. Leeds Metropolitan University. Leeds, West Workshire, Reino Unido. Outubro 1998. Disponível em: <<http://www.lmu.ac.uk/ies/conferences/mullier.html>> Acesso em: 11 dezembro 2000.

ODELL, James, PARUNAK, H. Van Dyke, BAUER, Bernhard. Extending UML for agents. Invited paper to be presented at the Second International Bi-Conference

Workshop on AGENT-ORIENTED INFORMATION SYSTEMS (AOIS-2000), 5-6 June 2000, Stockholm, Sweden. Disponível na Internet em: <<http://www.jamesodell.com>> Acesso em: 7 dezembro 2000.

PIMENTEL, Maria da Graça Campos, SANTOS JR., João Benedito dos, FORTES, Renata Pontin de Mattos. Tools for Authoring and Presenting Structured Teaching Material in the WWW. Accepted in WebNet'98, Top-paper Award. Extended version invited to be published at Journal of Universal Computer Science in 1998. Disponível em: <<http://www.icmsc.sc.usp.br/~mgp/webnet98>> Acesso em: 7 dezembro 2000.

PAIVA, A., SELF, J. A learner model reason maintenance system. In: Proceedings of ECAI-94, European Conference on Artificial Intelligence, Amsterdã, John Wiley & Sons, 1994. p. 193-196.

PAIVA, A. SELF, J. HARTLEY, R. Externalising Learner Models. In: Proceedings of "International Conference on Artificial Intelligence in Education", Washington, DC. Ed. J. Greer, AACE, 1995. Disponível em: <<http://www.cbl.leeds.ac.uk/amp/personal.html>> Acesso em: 7 dezembro 2000.

RAMOS, Edla M. F. Análise ergonômica do sistema hiperNet buscando o aprendizado da cooperação e da autonomia. Florianópolis, 1996. Tese (Doutorado em Engenharia de Produção) Universidade Federal de Santa Catarina. Disponível em: <<http://www.inf.ufsc.br/~edla>> Acesso em: 10 dezembro 2000.

REEVES, T., REEVES, P. Effective Dimensions of Interactive Learning on the World Wide Web. In: KHAN, B. (org.). Web-Based Instruction. New Jersey, EUA: Educational Technology Publications, 1997.

RUSSELL, Stuart J. NORVING, Peter. Artificial intelligence: a modern approach. Prentice Hall, New Jersey, 1995.

REGO, Teresa Cristina. Vygotsky, uma perspectiva histórico-cultural da educação. 4^a ed., Vozes, Petrópolis, 1997. pgs. 41-43.

SMITH, Roger W. Representing and manipulating user-specific knowledge in a semantic net to aid in the treatment of aphasia victims. Presented as part of the requirements for the award of the Degree of Bachelor of Science. Curtin University of Technology, USA, 1994. Supervisor: David Calder.

SMITH, S. Tutorial on Intelligent Tutoring. Disponível em: <<http://www.cs.mdx.ac.uk/staffpages/serengul/table.of.contents.htm>> Acesso em: 11 dezembro 2000.

SPECHT, Marcus, OPPERMANN, Reinhard. ATS – Adaptive Teaching System a WWW-Based ITS, In U. Timm (Eds.), Proceedings of Workshop Adaptivität und Benutzermodellierung in Interaktiven Softwaresystemen: ABIS 98, 1998.

- STERN, Mia K., WOOLF, Beverly Park, KUROSE, James F. Intelligence on the Web?, In the Proceedings of the 8th World Conference of the AIED Society, Kobe, Japan, 18-22 August, 1997.
- STERN, Mia K. The difficulties in web-based tutoring. In Proceedings of the workshop "Intelligent Educational Systems on the World Wide Web", 8th World Conference of the AIED Society, Kobe, Japan, 18-22 August 1997. Disponível em: <<http://www.cs.umass.edu/~stern/>> Acesso em: 9 dezembro 2000.
- STERN, Mia K. Web-based intelligent tutors derived from lecture-based courses, Unpublished Dissertation Proposal, Computer Science Department, University of Massachusetts, EUA, 1997. Disponível em: <<http://www.cs.umass.edu/~stern/>> Acesso em: 9 dezembro 2000.
- STERN, Mia K., WOOLF, Beverly Park. Curriculum Sequencing in a Web-based Tutor. In: Proceedings of Intelligent Tutoring Systems 1998 (ITS'98), San Antonio, Texas, August 1998. Disponível em: <<http://www.cs.umass.edu/~stern/>> Acesso em: 9 dezembro 2000.
- STURM, Jake. Professional VB Project Management. Wrox Press, USA, 1999.
- URBAN-LURAIN, Mark. Intelligent Tutoring Systems: an historic review in the context of the development of artificial intelligence and educational psychology. Paper presented in the Department of Computer Science and Engineering, Michigan State University. Michigan, EUA. Spring, 1996. Disponível em: <<http://aral.cse.msu.edu/Publications/ITS/its.htm>> Acesso em: 9 dezembro 2000.
- WEBER, G. SPECHT, M. User modeling and adaptive navigation support in WWW-based tutoring systems. Proceedings of User Modeling '97, Cagliari, Italy, June 2-5, 1997. p. 289-300. Disponível em: <<http://www.psychologie.uni-trier.de:8000/projects/ELM/Papers/UM97-WEBER.html>> Acesso em: 7 dezembro 2000.
- WOODS, Pamela, WARREN, James. Adapting teaching strategies in intelligent tutoring systems. Position Paper for ITS'96 Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs, Montreal, June 10th 1996.
- WOOLDRIDGE, Michael, JENNINGS, Nicholas R. Intelligent Agents: Theory and Practice, Reino Unido, Knowledge Engineering Review, v10n2, Junho 1995. Disponível em: <<http://www.elec.qmw.ac.uk/dai/pubs/KER95/>> Acesso em: 7 dezembro 2000.
- WOOLDRIDGE, M., JENNINGS, N. R., KINNY, D. The Gaia Methodology for Agent-Oriented Analysis and Design. Journal of Autonomous Agents and Multi-Agent Systems 3 (3), 2000, p. 285-312.
- ZHOU, Y. Building a New Student Model to Support Adaptive Tutoring in a Natural Language Dialogue System. Ph.D.Thesis, 140pp. Illinois Institute of Technology, EUA. 2000. Disponível em: <<http://www.csam.iit.edu/~circsim/index.html>> Acesso em: 10 dezembro 2000.

ANEXO 1 – Regras para Atualização do Valor Revisão

SE score de tempo estudado $\geq 0,8$

ENTÃO valor revisão \leftarrow valor revisão * 1,5

SE score de tempo estudado $\geq 0,5$

E score de tempo estudado $< 0,8$

ENTÃO valor revisão \leftarrow valor revisão * 1,35

SE score de tempo estudado $\geq 0,3$

E score de tempo estudado $< 0,5$

ENTÃO valor revisão \leftarrow valor revisão * 1,2

SE score de tempo estudado $< 0,3$

ENTÃO valor revisão \leftarrow valor revisão * 1,1

ANEXO 2 – Regras para Cálculo do Valor Ajustado

SE tipo link = pré-requisito

E valor necessário – valor aprendido $\leq 0,1$

ENTÃO valor ajustado $\leftarrow 0,8$

SE tipo link = pré-requisito

E valor necessário – valor aprendido $> 0,1$

E valor necessário – valor aprendido $\leq 0,3$

ENTÃO valor ajustado $\leftarrow 0,6$

SE tipo link = pré-requisito

E valor necessário – valor aprendido $> 0,3$

E valor necessário – valor aprendido $\leq 0,5$

ENTÃO valor ajustado $\leftarrow 0,4$

SE tipo link = pré-requisito

E valor necessário – valor aprendido $> 0,5$

ENTÃO valor ajustado $\leftarrow 0,2$

SE tipo link = co-requisito

E valor necessário – valor aprendido $\leq 0,1$

ENTÃO valor ajustado $\leftarrow 0,7$

SE tipo link = co-requisito

E valor necessário – valor aprendido $> 0,1$

E valor necessário – valor aprendido $\leq 0,25$

ENTÃO valor ajustado $\leftarrow 0,4$

SE tipo link = co-requisito

E valor necessário – valor aprendido $> 0,25$

ENTÃO valor ajustado $\leftarrow 0,2$

SE tipo link = relacionado

E valor necessário – valor aprendido $\leq 0,1$

ENTÃO valor ajustado $\leftarrow 0,8$

SE tipo link = relacionado

E valor necessário – valor aprendido $> 0,1$

E valor necessário – valor aprendido $\leq 0,3$

ENTÃO valor ajustado $\leftarrow 0,5$

SE tipo link = relacionado

E valor necessário – valor aprendido $> 0,3$

ENTÃO valor ajustado $\leftarrow 0,2$

SE tipo link = reforço

E valor necessário – valor aprendido $\leq 0,1$

ENTÃO valor ajustado $\leftarrow 0,8$

SE tipo link = reforço

E valor necessário – valor aprendido $> 0,1$

E valor necessário – valor aprendido $\leq 0,35$

ENTÃO valor ajustado $\leftarrow 0,5$

SE tipo link = co-requisito

E valor necessário – valor aprendido $> 0,35$

ENTÃO valor ajustado $\leftarrow 0,2$