

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

GRACIELA CRISTINA BERNARDES LIMA

***AdaptHA: Ambiente para Autoria e Ensino
Adaptativo***

Dissertação apresentada como requisito parcial
para a obtenção do grau de Mestre em Ciência
da Computação

Prof. Dr. Paulo Blauth Menezes
Orientador

Porto Alegre, maio de 2007.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Lima, Graciela Cristina Bernardes

AdaptHA: Ambiente para Autoria e Ensino Adaptativo / Graciela Cristina Bernardes Lima – Porto Alegre: Programa de Pós-Graduação em Computação, 2007.

108 f.:il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR – RS, 2007. Orientador: Paulo Blauth Menezes.

1.Educação a distância 2.Hipermídia adaptativa 3.Autoria 4.Autômatos. I. Menezes, Paulo Blauth. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. José Carlos Ferraz Hennemann

Vice-Reitor: Prof. Pedro Cezar Dutra Fonseca

Pró-Reitora de Pós-Graduação: Profa. Valquíria Linck Bassani

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenadora do PPGC: Profa. Luciana Porcher Nedel

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*“Nunca ande por caminhos já traçados, pois eles
sempre o levam a lugares onde os outros já foram.”*

(Alexander Graham Bell)

AGRADECIMENTOS

A Deus, sem Ele este trabalho não teria sido possível.

A meus queridos pais, que me educaram e sempre me apoiaram em todos os momentos da minha vida.

A meu esposo Joaquim, por seu amor, amizade, dedicação, apoio e compreensão incondicional.

A meus irmãos e sobrinhos.

A meu orientador, Prof. Paulo Blauth Menezes, por ter me dado a oportunidade de realizar este trabalho, além do apoio e atenção constante.

A minha grande amiga Renata Zanella, por sua indiscutível contribuição neste trabalho.

A meu grupo de pesquisa, em especial, Sandra Kniphoff e Sílvia Vargas.

A meu colega de trabalho e amigo Sérgio Farias, por seu incentivo e apoio na realização deste trabalho.

Ao BRDE, que gentilmente me dispensou parcialmente de minhas atividades durante um período, para que eu pudesse cumprir as atividades do curso de mestrado.

Aos funcionários do Instituto de Informática e do PPGC, sempre muito atenciosos.

Enfim, a todos aqueles que de alguma forma contribuíram para a realização deste trabalho.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	7
LISTA DE FIGURAS	9
LISTA DE TABELAS	10
RESUMO	11
ABSTRACT	12
1 INTRODUÇÃO	13
1.1 Proposta	14
1.2 Organização	16
2 HIPERMÍDIA ADAPTATIVA	17
2.1 Áreas de Aplicação da Hipermídia Adaptativa	19
2.1.1 Hipermídia Educacional.....	19
2.1.2 Sistemas de Informação <i>Online</i>	20
2.1.3 Sistemas Hipermídia de Recuperação de Informação.....	20
2.1.4 Sistemas de Ajuda <i>Online</i>	21
2.1.5 Sistemas de Informação Institucional.....	21
2.2 Aspectos do Usuário Levados em Consideração ao Adaptar	22
2.2.1 Dados do Usuário.....	22
2.2.2 Dados de Uso.....	24
2.2.3 Dados do Ambiente.....	25
2.3 Aspectos do Sistema que Podem ser Adaptados	27
2.3.1 Apresentação Adaptativa.....	28
2.3.2 Suporte à Navegação Adaptativa.....	30
2.4 Métodos da Hipermídia Adaptativa	33
2.4.1 Métodos da Apresentação Adaptativa.....	33
2.4.2 Métodos do Suporte à Navegação Adaptativa.....	34
2.5 Modelagem do Usuário	37
2.5.1 Características de Modelos do Usuário.....	38
2.5.2 Tipos de Modelos do Usuário.....	39
2.6 Sistemas Hipermídia Adaptativos Educacionais Baseados na <i>Web</i>	43
2.6.1 AHA.....	43
2.6.2 WHURLE.....	44
2.6.3 AdaptWeb.....	45

2.6.4	iWeaver.....	46
2.7	Considerações Finais.....	47
3	O MODELO PROPOSTO	49
3.1	Contexto do Trabalho	49
3.1.1	Primeira Geração de Pesquisa em Hiper m dia Educacional Adaptativa.....	49
3.1.2	Segunda Geração de Pesquisa em Hiper m dia Educacional Adaptativa.....	50
3.1.3	Terceira Geração de Pesquisa em Hiper m dia Educacional Adaptativa	50
3.2	Trabalhos Relacionados.....	51
3.3	Projeto Instrucional	53
3.3.1	Categorias de Capacidades Humanas.....	54
3.3.2	Eventos de Instrução	60
3.4	Modelo Hyper-Automaton.....	62
3.4.1	Autômatos Finitos	63
3.4.2	Autômatos Finitos com Saída	64
3.4.3	Cursos como Autômatos.....	65
3.4.4	Vantagens do Modelo Hyper-Automaton.....	66
3.5	Arquitetura do <i>AdaptHA</i>.....	66
3.5.1	Camada de Interface.....	66
3.5.2	Camada de Persistência	69
3.5.3	Camada de Negócio.....	75
3.6	Considerações Finais.....	82
4	O SISTEMA IMPLEMENTADO	83
4.1	Frameworks Utilizados pelo <i>AdaptHA</i>	83
4.1.1	Java2 Enterprise Edition (J2EE)	83
4.1.2	Hibernate.....	86
4.1.3	JavaServer Faces (JSF).....	88
4.2	O Ambiente de Autoria do <i>AdaptHA</i>.....	92
4.2.1	Transferindo Objetos de Aprendizagem para o Servidor	94
4.2.2	Editando Textos com o FCKeditor.....	95
4.2.3	Criando Textos.....	96
4.2.4	Criando Questões de Avaliação	97
4.2.5	Criando Avaliações	99
4.2.6	Criando Cursos.....	100
4.2.7	Criando Conceitos para um Curso	101
4.2.8	Anexando Recursos Didáticos a um Conceito.....	102
4.3	Considerações Finais.....	105
5	CONCLUSÃO.....	106
5.1	Trabalhos Futuros.....	107
5.2	Produção Científica.....	108
	REFERÊNCIAS.....	109

LISTA DE ABREVIATURAS E SIGLAS

AHA	Adaptive Hypermedia Architecture
API	Application Programming Interface
DAO	Data Access Object
EIS	Enterprise Information System
HA	Hipermídia Adaptativa
HCI	Human-Computer Interaction
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IDE	Integrated Development Environment
IU	Interface do Usuário
J2EE	Java2 Enterprise Edition
JDBC	Java DataBase Connectivity
JMS	Java Message Service
JSF	JavaServer Faces
JSP	JavaServer Pages
LMS	Learning Management System
ORM	Object / Relational Mapping
PDA	Personal Digital Assistant
RAD	Rapid Application Development
RDF	Resource Description Framework
SHA	Sistema Hipermídia Adaptativo
SQL	Structured Query Language
SVG	Scalable Vector Graphics
SWT	Standard Widget Toolkit
URL	Uniform Resource Locator
WCML	WHURLE Chunk Markup Language
WHURLE	Web-based Hierarchical Universal Reactive Learning Environment

WLPML	WHURLE Lesson Plan Markup Language
WML	Wireless Markup Language
WWW	World Wide Web
XHTML	EXtensible HyperText Markup Language
XML	EXtensible Markup Language

LISTA DE FIGURAS

Figura 2.1: Modelagem do usuário – adaptação em SHAs.....	17
Figura 2.2: Espectro de adaptação em sistemas.	18
Figura 2.3: Taxonomia das tecnologias da HA.	28
Figura 3.1: Habilidades Intelectuais	57
Figura 3.2: Representação da função programa como um grafo.....	64
Figura 3.3: Arquitetura do <i>AdaptHA</i>	67
Figura 4.1: Aplicações Multicamada	85
Figura 4.2: Servidor J2EE e <i>Containers</i>	86
Figura 4.3: Visão geral de uma aplicação JSF	89
Figura 4.4: Trecho do faces-config.xml do <i>AdaptHA</i>	91
Figura 4.5: Mensagens de validação localizadas no <i>AdaptHA</i>	91
Figura 4.6: Mecanismos de internacionalização e localização do <i>AdaptHA</i>	92
Figura 4.7: Interface de cadastramento de usuários do <i>AdaptHA</i>	92
Figura 4.8: Interface de gerenciamento do cadastro de autores do <i>AdaptHA</i>	93
Figura 4.9: Interface de <i>login</i> do <i>AdaptHA</i>	93
Figura 4.10: Interface de boas vindas do <i>AdaptHA</i>	93
Figura 4.11: Interface principal de autoria do <i>AdaptHA</i>	93
Figura 4.12: Interface de <i>upload</i> de objetos de aprendizagem do <i>AdaptHA</i>	94
Figura 4.13: Interface de visualização de objetos de aprendizagem do <i>AdaptHA</i>	94
Figura 4.14: FCKeditor, o editor de textos do <i>AdaptHA</i>	95
Figura 4.15: Inserindo objetos de aprendizagem no texto através do FCKeditor	96
Figura 4.16: Interface de criação de textos do <i>AdaptHA</i>	96
Figura 4.17: Interface de visualização de textos do <i>AdaptHA</i>	97
Figura 4.18: Navegando por um texto no <i>AdaptHA</i>	97
Figura 4.19: Interface de criação de questões do <i>AdaptHA</i>	98
Figura 4.20: Interface de visualização de questões do <i>AdaptHA</i>	99
Figura 4.21: Interface de criação de avaliações do <i>AdaptHA</i>	99
Figura 4.22: Interface de seleção de questões de avaliações do <i>AdaptHA</i>	100
Figura 4.23: Interface de visualização de avaliações do <i>AdaptHA</i>	100
Figura 4.24: Interface de criação de cursos do <i>AdaptHA</i>	101
Figura 4.25: Interface de criação de conceitos do <i>AdaptHA</i>	101
Figura 4.26: Estruturação do conteúdo dos conceitos no <i>AdaptHA</i>	102
Figura 4.27: Parte super. da interface de seleção de recursos didáticos do <i>AdaptHA</i> ..	103
Figura 4.28: Parte inferior da interface de seleção de recursos didáticos do <i>AdaptHA</i>	103
Figura 4.29: Interface de seleção de recursos didáticos do <i>AdaptHA</i>	103
Figura 4.30: Parte sup. da interface de seleção de recursos didáticos do <i>AdaptHA</i>	104
Figura 4.31: Parte inferior da interface de seleção de recursos didáticos do <i>AdaptHA</i>	104

LISTA DE TABELAS

Tabela 3.1: Comparação Entre o <i>AdaptHA</i> e Outros Trabalhos Relacionados.....	53
Tabela 3.2: Ações Relacionadas às Categorias de Capacidades Humanas.....	55
Tabela 3.3: Critérios para Escolha de uma Estratégia Pedagógica.....	77
Tabela 3.4: Capacidades Humanas X Domínios de Aprendizagem X Tipo da Tarefa ..	78
Tabela 3.5: Estrutura de Apresentação dos Conceitos.....	78
Tabela 3.6: Estados de <i>Links</i> Diferenciados e Anotados pelo <i>AdaptHA</i>	78
Tabela 3.7: Comportamento dos <i>Links</i> X Estratégias Pedagógicas do <i>AdaptHA</i>	79
Tabela 3.8: Exemplo de Um Curso no <i>AdaptHA</i>	79
Tabela 3.9: Exemplo de Cores para os <i>Links</i> Anotados.....	79
Tabela 3.10: Exemplo de Modelo do Aluno X Anotação e Comportamento dos <i>Links</i> .	80
Tabela 4.1: Elem. Estruturais X Eventos de Instrução X Dimensões de Adaptação....	102

RESUMO

A *Web* vem se apresentando um meio cada vez mais promissor para o desenvolvimento de sistemas de ensino. Um dos grandes desafios é fazer uso das vantagens da *Web*, que torna a aprendizagem disponível em qualquer lugar e em qualquer momento, e criar para o aluno uma experiência de aprendizagem *individualizada*, que represente um significativo melhoramento quando comparada à experiência de aprendizagem convencional de sala de aula.

Nesse contexto, este trabalho apresenta o *AdaptHA*, um ambiente para autoria e ensino adaptativo na *Web* baseado no modelo Hyper-Automaton. Com o *AdaptHA*, objetiva-se colaborar para a melhoria da qualidade da educação prática pela *Web*. Acredita-se que isto seja possível através de um sistema de ensino que reúna ferramentas que auxiliam tanto o professor quanto o aluno, mas com um grande diferencial no que se refere à capacidade deste sistema de se *adaptar* às características do aluno.

AdaptHA proporciona ao aluno uma experiência de aprendizagem individualizada ao apresentar o conteúdo adaptado e suportar a navegação adaptativa com base no modelo do aluno, utilizando para isto, técnicas da Hiper-mídia Adaptativa. Os mecanismos chave do *AdaptHA* que viabilizam tal adaptação são: sua estrutura do *modelo do domínio*, cujas principais características são flexibilidade e riqueza de metadados; e sua estrutura do *modelo do aluno*, que mantém o histórico detalhado de todas as interações do aluno com o ambiente.

A essência deste trabalho consiste em promover um tipo de ensino que seja centrado no aluno. Entretanto, é indispensável levar em consideração que o professor possui um papel fundamental dentro do cenário educacional, sendo, portanto, imprescindível prover mecanismos que colaborem para a melhoria do seu trabalho, como por exemplo: *ferramentas de autoria* que facilitem a criação, manutenção e reutilização de cursos e material instrucional; e ferramentas que possibilitem o *acompanhamento* contínuo do desempenho dos alunos.

A indisponibilidade de ferramentas que facilitam o trabalho do professor em um ambiente de ensino computadorizado, seja ele *Web* ou não, significa, em última instância, que a autoria de cursos e material instrucional irá exigir conhecimento de pessoal técnico, o que não é razoável. Evidentemente, este aspecto gera uma barreira quase que intransponível para a utilização desse tipo de ambiente por parte do professor. Neste sentido, o *AdaptHA* dispõe de um ambiente de autoria que reúne um rico conjunto de ferramentas que visa suprir essas necessidades.

Palavras-Chave: Educação a distância, hiper-mídia adaptativa, autoria, autômatos.

AdaptHA: Adaptive Authorship and Learning Environment

ABSTRACT

Web has becoming more and more a promising environment for the development of learning systems. One of the big challenges is making use of Web advantages, that makes learning available anywhere and anytime, and create to the student an *individualized* learning experience, that represents a significant improvement when compared to the conventional classroom learning experience.

In this context, this work presents *AdaptHA*, a Web adaptive authorship and learning environment based on Hyper-Automaton model. With *AdaptHA*, the goal is to collaborate to the improvement of Web practical education quality. This is possible through a learning system that groups tools that support teacher and student, but with a great plus that refers to the ability of this system to *adapt* itself to the student's characteristics.

AdaptHA provides to the student an individualized learning experience by presenting content adapted and supports adaptive navigation based on student model, making use of Adaptive Hypermedia techniques. *AdaptHA*'s key mechanisms that turn such adaptation possible are: its *domain model* structure, whose main features are flexibility and a rich metadata set; and its *student model* structure, that keeps a detailed history of all student interactions with the environment.

The essence of this work is to promote a student-centered education. However, it must be taken in consideration that the teacher performs a fundamental role in the educational setting, so, it is necessary supplying mechanisms that collaborate for the improvement of his / her work, as for instance: *authorship tools* that facilitate the creation, maintenance and reuse of courses and instructional material; and tools that enable the continuous *accompaniment* of students performance.

The unavailability of tools that facilitate teacher work in a computer-based education environment, Web or not, means that courses and instructional material authorship will require knowledge of technical staff, what is not reasonable. Obviously, this aspect creates a great barrier for the use of this kind of environment by the teacher. In this sense, *AdaptHA* provides an authorship environment that groups a rich set of tools that supplies these needs.

Keywords: Distance education, adaptive hypermedia, authorship, automata.

1 INTRODUÇÃO

Segundo Johnson (2003), na antiguidade a educação era transmitida por educadores profissionais a um grupo seletivo de pessoas, de forma individual ou em pequenos grupos. Foi o advento da educação universal e do sistema universitário moderno de larga-escala que levou ao atual sistema de ensino, em que um único professor é responsável pela educação simultânea de um grande número de alunos. Embora este sistema tenha tornado a educação mais disponível, ele também reduziu significativamente sua qualidade.

A tecnologia de educação baseada no computador tem buscado melhorar a qualidade da experiência de aprendizagem do aluno, tentando torná-la mais próxima do que é possível com a educação individual ou em pequenos grupos. Segundo Brusilovsky (2001), a escolha pela *Web* como plataforma de desenvolvimento de sistemas de ensino se tornou um padrão. Isto se justifica dado o fato de que na *Web* é possível distribuir o conhecimento em larga escala, potencialmente para todo o mundo, reduzir os custos de distribuição, corrigir e atualizar mais facilmente o conteúdo, diversificar as técnicas de ensino com a utilização de recursos multimídia, além de facilitar o trabalho colaborativo entre alunos e professores.

Um dos grandes desafios é fazer uso das vantagens da *Web*, que torna a educação disponível em qualquer lugar e em qualquer momento, e criar para o aluno uma experiência de aprendizagem *individualizada*, que represente um significativo melhoramento quando comparada à experiência de aprendizagem convencional de sala de aula. Neste sentido, a Hiperídia Adaptativa (HA), que tem a hiperídia educacional como uma de suas principais áreas de aplicação, apresenta alternativas para individualizar a experiência de aprendizagem do aluno. Esta individualização se torna possível quando os sistemas constroem um modelo dos objetivos, preferências e conhecimento de cada aluno e aplicam este modelo com o propósito de adaptar vários aspectos visíveis do sistema para o aluno (BRUSILOVSKY, 1996-a). Tais sistemas são denominados *sistemas hiperídia educacionais adaptativos*.

Apesar dos sistemas hiperídia educacionais adaptativos já terem atingido uma certa maturidade, Brusilovsky (2004) afirma que eles falharam ao tentar influenciar a educação prática baseada na *Web*. Poucos foram usados para ensino em cursos reais. Pelo contrário, a maioria absoluta de cursos na *Web* conta com os chamados *Learning Management Systems* (LMSs), como (BLACKBOARD, 2007; TOPCLASS, 2007). LMSs são poderosos sistemas integrados que suportam uma série de necessidades, tanto de professores quanto de alunos. Embora os LMSs, segundo Johnson (2003), ajudem a tornar a educação mais amplamente disponível, eles fazem pouco para melhorar a *qualidade* da educação. Para Brusilovsky (2004), parece óbvio que o problema com sistemas hiperídia educacionais adaptativos está relacionado à sua inabilidade de

atender às necessidades da educação prática pela *Web*. É justamente neste sentido, com o objetivo de ajudar a preencher esta lacuna, que este trabalho foi desenvolvido.

1.1 Proposta

A idéia central deste trabalho consiste em colaborar para a melhoria da qualidade da educação prática pela *Web*. Acredita-se que isto seja possível através de um sistema de ensino baseado na *Web* que reúna funcionalidades semelhantes às que são fornecidas por modernos LMSs, mas com um grande diferencial com relação a estes últimos, no que se refere à capacidade deste sistema de se *adaptar* às características do aluno.

Nesse contexto, propõe-se neste trabalho o *AdaptHA* (LIMA et al., 2005; LIMA et al., 2006-a; LIMA et al., 2006-b), um ambiente para autoria e ensino adaptativo na *Web* baseado no modelo Hyper-Automaton (MACHADO et al., 1999-a; MACHADO et al., 1999-b; LIMA et al., 2004). *AdaptHA* objetiva proporcionar ao aluno uma experiência de aprendizagem individualizada ao apresentar o conteúdo adaptado e suportar a navegação adaptativa com base no modelo do aluno. Os mecanismos chave do *AdaptHA* que viabilizam tal adaptação são: sua estrutura do *modelo do domínio*, cujas principais características são flexibilidade e riqueza de metadados; e sua estrutura do *modelo do aluno*, que mantém o histórico detalhado de todas as interações do aluno com o ambiente.

A subseção 3.5.3 apresenta em detalhes o mecanismo proposto pelo *AdaptHA* para preparar o conteúdo final que deve ser apresentado ao aluno. Tal mecanismo consiste na aplicação de três *níveis de adaptação* que são os seguintes:

- Escolha da estratégia pedagógica a ser aplicada (que pode ser diretiva, orientada por descoberta ou exploratória), com base no modelo do aluno;
- Aplicação da estratégia pedagógica selecionada no passo anterior, o que resulta em o sistema assumir comportamento diferenciado, de acordo com a estratégia pedagógica, oferecendo ao aluno menor ou maior liberdade de navegação pelo curso;
- Aplicação de técnicas da HA (que serão detalhadas na seção 2.3) para adaptação do conteúdo e suporte à navegação adaptativa.

O *AdaptHA* se baseia no Hyper-Automaton, cujo modelo suporta o gerenciamento e a apresentação de hiperdocumentos instrucionais como cursos na *Web*. O Hyper-Automaton utiliza o formalismo de *Autômatos Finitos com Saída* como um modelo estrutural para a organização de hiperdocumentos. Nesse modelo cada *autômato* define um *curso* e referencia um conjunto de hiperdocumentos independentes, tal que a estrutura navegacional do curso permanece separada do material instrucional.

Considera-se neste trabalho que o processo avaliativo em um ambiente de ensino é de substancial relevância. Isto se deve ao fato de que os resultados das avaliações realizadas pelos alunos constituem uma boa fonte de informação que pode ser utilizada tanto para oferecer *feedback* para o aluno quanto para fornecer a base para inferências futuras. O ambiente de autoria de questões de avaliações e avaliações do *AdaptHA* (que é parte do ambiente de autoria do *AdaptHA*) se inspirou fortemente no EASy (Sistema de Geração Automática de Avaliações via *Web* baseado no modelo Hyper-Automaton) (ZANELLA et al., 2005). EASy é um ambiente de autoria de questões e geração de avaliações a partir de metadados configuráveis.

Objetiva-se com o *AdaptHA* promover um ensino centrado no aluno. No entanto, embora a educação deva ser de fato centrada no aluno, é importante levar em consideração que o professor desempenha um papel fundamental dentro do cenário educacional, sendo, portanto, imprescindível prover mecanismos que colaborem para a melhoria do seu trabalho. Assim sendo, o *AdaptHA* também tem como objetivo fornecer ao professor: *ferramentas de autoria* que facilitem a criação, manutenção e reutilização de cursos e material instrucional; e ferramentas que possibilitem o *acompanhamento* contínuo do desempenho dos alunos. Em particular, as ferramentas de autoria que um sistema hipermídia educacional adaptativo oferece ao professor, constituem um fator muito forte de competitividade com os modernos LMSs. Aliás, segundo Brusilovsky (2001), o surgimento de ferramentas de autoria indica a maturidade da hipermídia educacional adaptativa e uma resposta à demanda impulsionada pela *Web* por cursos adaptativos de educação a distância.

A indisponibilidade de ferramentas de autoria adequadamente projetadas para serem utilizadas pelo professor em um ambiente de ensino computadorizado, seja ele *Web* ou não, significa, em última instância, que a autoria de cursos e material instrucional irá exigir conhecimento de pessoal técnico, o que não é razoável. Evidentemente, este aspecto gera uma barreira quase que intransponível para a utilização desse tipo de ambiente por parte do professor. Neste mesmo sentido, Cristea (2005) argumenta que para que os sistemas adaptativos se tornem *economicamente* viáveis e aceitáveis para potenciais clientes, o processo de autoria deve ser simples, porém eficiente, sistemático, orientado a papel, com clara semântica e transparente. Apenas desta forma, os professores, por exemplo, podem seguramente se tornar autores de hipermídia educacional adaptativa, sem a necessidade de habilidades de programação ou treinamento excessivo.

O desenvolvimento de ferramentas de autoria para sistemas de ensino não é uma tarefa trivial. Esta tarefa se torna bem mais complexa quando as ferramentas de autoria devem suportar a criação de cursos adaptativos. Por esta razão, a abordagem geral para o desenvolvimento de sistemas de ensino, adaptativos ou não, consiste em implementar uma porção do sistema que possibilita que o aluno realize um curso elaborado *manualmente* por pessoal técnico. As ferramentas de autoria, freqüentemente, não são implementadas. Este trabalho, por sua vez, tomou a direção contrária. Optou-se por iniciar o desenvolvimento a partir da construção de um conjunto de ferramentas de autoria, que se encontram reunidas em um ambiente de autoria. A segunda seção do capítulo 4 dissecou completamente o ambiente de autoria do *AdaptHA*.

A idéia geral que norteou o desenvolvimento do ambiente de autoria do *AdaptHA*, foi a de que as ferramentas construídas deveriam verdadeiramente facilitar as tarefas de autoria do professor, que neste ambiente é denominado *autor*. Neste aspecto, o ambiente de autoria do *AdaptHA*:

- Foi enriquecido com um poderoso editor de textos, o FCKeditor (FCKEDITOR, 2007), que permite desde a realização de operações elementares de edição de texto até as mais sofisticadas;
- Suporta a transferência de arquivos de diversos formatos para o servidor em que o *AdaptHA* é executado, sendo que tais arquivos podem ser inseridos posteriormente no texto editado no FCKeditor. Por exemplo, é possível inserir um arquivo de imagem no texto editado;

- Suporta a criação de diferentes tipos de recursos, como textos, questões de avaliações e avaliações;
- Suporta a reutilização dos recursos criados pelo autor, assim como dos recursos públicos criados por outros autores, pois os recursos são mantidos de maneira independente dos cursos. Quando um curso é criado, os recursos desejados devem ser anexados ao mesmo;
- Facilita a recuperação dos recursos criados, pois um rico conjunto de metadados é associado a eles;
- Suporta a criação de cursos adaptativos ou não. Cabe ao autor decidir o grau de adaptação que será oferecido ao aluno, sendo possível, inclusive, não ser oferecido grau algum de adaptação.

1.2 Organização

Este trabalho está organizado da seguinte forma: o capítulo 2, *Hipermídia Adaptativa*, traça um panorama bastante abrangente dos principais aspectos da HA, além de fornecer o referencial teórico que embasou o desenvolvimento do trabalho aqui apresentado; o capítulo 3, *O Modelo Proposto*, apresenta o modelo proposto neste trabalho, a teoria de aprendizagem na qual ele se inspira, uma comparação do *AdaptHA* com trabalhos relacionados, o modelo Hyper-Automaton e a arquitetura do *AdaptHA*, examinada detalhadamente em cada uma de suas camadas; o capítulo 4, *O Sistema Implementado*, apresenta os aspectos tecnológicos envolvidos no desenvolvimento do *AdaptHA*, assim como seu ambiente de autoria; o capítulo 5, *Conclusão*, apresenta a conclusão deste trabalho, incluindo suas contribuições, trabalhos futuros e produção científica ao longo do curso de mestrado.

2 HIPERMÍDIA ADAPTATIVA

De acordo com Brusilovsky (2001), a HA é uma linha de pesquisa relativamente recente que une outras duas linhas de pesquisa bastante próximas: a *hipermídia* e a *modelagem do usuário*. Uma limitação das aplicações hipermídia tradicionais (pode-se dizer também estáticas) é que elas fornecem o mesmo conteúdo e o mesmo conjunto de *links* para todos os usuários. Se o público-alvo é relativamente diverso, um sistema tradicional será incapaz de atender a todos os usuários. A HA é uma alternativa para a abordagem “*one-size-fits-all*” utilizada no desenvolvimento de sistemas hipermídia.

SHAs constroem um modelo dos objetivos, preferências e conhecimento de cada usuário individualmente e usam este modelo ao longo da interação com o usuário a fim de se adaptarem às necessidades desse usuário (BRUSILOVSKY, 1996-a). Assim sendo, Brusilovsky (1996-a) define SHAs como “*todos os sistemas hipertexto e hipermídia que refletem algumas características do usuário no modelo do usuário e aplicam esse modelo para adaptar vários aspectos visíveis do sistema para o usuário*”. A figura 2.1 ilustra esta definição.

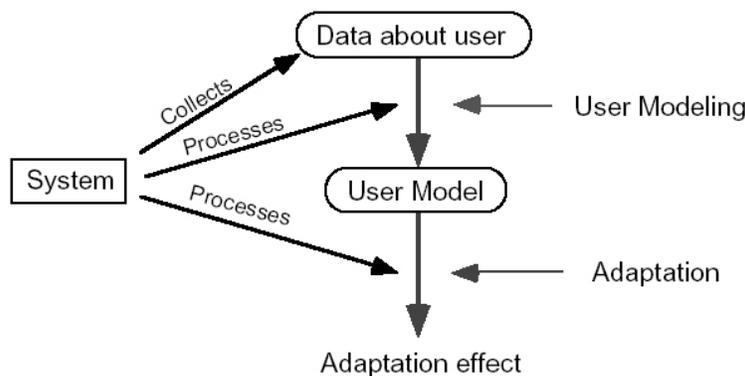


Figura 2.1: Modelagem do usuário – adaptação em SHAs (BRUSILOVSKY, 1996-a).

Vale ressaltar que Brusilovsky (2001) atribui em parte à *Web* a responsabilidade pelo grande crescimento e desenvolvimento da HA. Isso se explica devido ao fato da *Web* claramente lidar com um público altamente diverso e portanto apropriado para impulsionar a pesquisa em HA.

Segundo Oppermann (1994), um sistema é dito *adaptável* (adaptabilidade) se fornece ao usuário ferramentas que possibilitam modificações nas características do sistema. Este tipo de individualização torna o usuário responsável pelo controle da adaptação, ou seja, o usuário a inicia e a emprega. Um sistema é dito *adaptativo* (adaptatividade) se é capaz de modificar suas características automaticamente de acordo com as necessidades do usuário. Modificações na apresentação da interface ou no

comportamento do sistema dependem da maneira que o usuário interage com o mesmo. Desta forma, o sistema é quem inicia e executa as modificações apropriadas para o usuário. A figura 2.2 exibe um espectro do conceito de adaptação em sistemas computacionais (OPPERMANN et al., 1997).

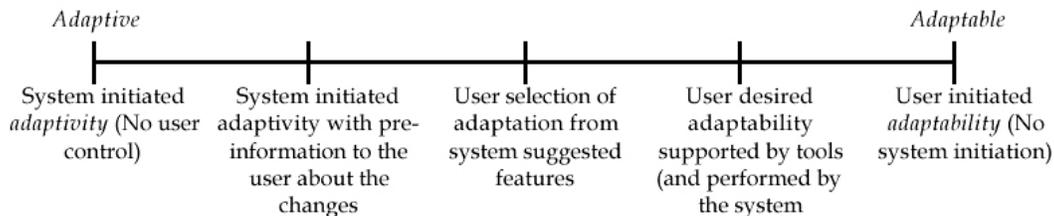


Figura 2.2: Espectro de adaptação em sistemas (OPPERMANN et al., 1997).

Klann et al. (2003) sustenta que ambas as abordagens, adaptativa e adaptável, mantêm o sistema flexível durante sua utilização. Tal sistema flexível, que se adapta a seus usuários e cujos usuários podem adaptá-lo de acordo com suas necessidades e preferências, deveria ser fácil de manipular e deveria aumentar a produtividade e satisfação do usuário, além de otimizar a carga de trabalho. O desafio é encontrar um equilíbrio entre a adaptatividade automática do sistema e a adaptabilidade controlada pelo usuário que resulte em um sistema flexível através do que Klann et al. (2003) denomina iniciativa compartilhada (*shared initiative*). A idéia por trás da iniciativa compartilhada é utilizar a adaptabilidade com o propósito de melhorar a adaptatividade e vice-versa. Uma melhor adaptatividade seria possível adicionando-se adaptabilidade que faça com que o usuário forneça ao sistema informações que de outra forma não poderiam ser obtidas, senão o questionando. Por outro lado, a adaptabilidade pode se beneficiar da adaptatividade, pois a última ajuda a reduzir a carga cognitiva associada à customização do sistema por parte do usuário final, ocultando funções de adaptação que não são pertinentes ao contexto atual.

Segundo Brusilovsky (1996-a), *técnicas de adaptação* são métodos de prover adaptação em um SHA e fazem parte do nível de implementação do mesmo. Cada técnica pode ser caracterizada por um tipo de representação do conhecimento e algoritmo de adaptação específicos. *Métodos de adaptação* são definidos como generalizações de técnicas de adaptação existentes. Cada método é baseado em uma clara idéia de adaptação que pode ser descrita em nível conceitual. Por exemplo:

- Inserir a comparação entre o conceito atual e outro conceito se o último é conhecido pelo usuário;
- Esconder os *links* que apontam para conceitos que não são conhecidos pelo usuário.

O mesmo método conceitual pode ser implementado por diferentes técnicas. Ao mesmo tempo, algumas técnicas são usadas para implementar diversos métodos utilizando a mesma representação do conhecimento.

Nas seções seguintes deste capítulo aspectos essenciais da HA são descritos em detalhes: a seção 2.1 descreve as áreas de aplicação da HA e de que maneira ela pode ajudar a solucionar problemas associados a cada uma dessas áreas; a seção 2.2 descreve aspectos do usuário que podem ser levados em consideração ao adaptar; a seção 2.3 descreve os aspectos visíveis do sistema que podem ser adaptados com base no modelo do usuário; a seção 2.4 apresenta os métodos da HA tanto no âmbito da apresentação adaptativa quanto no suporte à navegação adaptativa; a seção 2.5 apresenta questões

relacionadas à modelagem do usuário – que representa um aspecto fundamental para o desenvolvimento de SHAs – bem como características e tipos de modelos; a seção 2.6 apresenta alguns SHAs recentes baseados na *Web* e que serviram como referencial para o desenvolvimento deste trabalho; a seção 2.6 apresenta as considerações finais deste capítulo.

2.1 Áreas de Aplicação da Hipermedia Adaptativa

Brusilovsky (1996-a) identifica seis categorias de sistemas hipermedia utilizados como áreas de aplicação na maioria dos projetos de pesquisa em HA. Tais áreas de aplicação são: hipermedia educacional, sistemas de informação *online*, sistemas de ajuda *online*, sistemas hipermedia de recuperação de informação, sistemas de informação institucional e sistemas para gerenciamento de visões personalizadas.

Brusilovsky (2001) afirma que a hipermedia educacional e os sistemas de informação *online* são as áreas mais populares da HA e juntas são responsáveis por cerca de dois terços dos esforços de pesquisa. Sistemas hipermedia de recuperação de informação vêm em seguida. O escopo tradicional dos sistemas hipermedia de recuperação de informação foi estendido e inclui também os sistemas para gerenciamento de visões personalizadas. Já os sistemas de ajuda *online* e os sistemas de informação institucional quase não têm recebido atenção dos pesquisadores da HA nos últimos anos. Tais áreas de aplicação são descritas nas subseções abaixo.

2.1.1 Hipermedia Educacional

Sistemas de hipermedia educacional existentes têm hiperespaços relativamente pequenos que representam um curso ou seção de material instrucional de um assunto em particular. O objetivo do aluno é, em geral, aprender todo esse material ou parte razoável dele. O conhecimento do assunto ensinado é a característica mais importante do usuário a ser considerada. Técnicas da HA podem ser úteis para resolver alguns problemas associados à hipermedia educacional. Em primeiro lugar, o conhecimento de diferentes usuários pode variar enormemente e o conhecimento de um usuário em particular pode aumentar muito rapidamente. Uma dada página pode não ser evidente para um aluno iniciante enquanto que para um aluno de nível avançado pode ser trivial e entediante. Em segundo lugar, iniciantes entram no hiperespaço do material instrucional conhecendo quase nada a respeito do assunto. A maioria dos *links* oferecidos a partir de um nó leva a material que é completamente novo para eles. Eles precisam de assistência navegacional a fim de encontrar seu caminho pelo hiperespaço. Sem tal assistência eles podem ficar perdidos, mesmo em hiperespaços razoavelmente pequenos, ou usar estratégias de navegação muito ineficientes.

Brusilovsky (2001) afirma que o interesse crescente em prover educação a distância pela *Web* tem sido a grande força motriz por trás dos esforços de pesquisa e desenvolvimento em hipermedia educacional. A introdução da *Web* gerou impacto tanto na quantidade quanto nos tipos de sistemas desenvolvidos. Os sistemas mais antigos eram essencialmente experimentais, construídos para explorar alguns novos métodos que usavam adaptatividade em um contexto educacional. Por outro lado, vários sistemas mais recentes fornecem *frameworks* completos e até mesmo ferramentas de autoria para o desenvolvimento de cursos baseados na *Web*. O surgimento de ferramentas de autoria indica a maturidade da hipermedia educacional adaptativa e uma resposta à demanda

impulsionada pela *Web* por cursos adaptativos de educação a distância. A escolha pela *Web* como plataforma de desenvolvimento se tornou um padrão.

2.1.2 Sistemas de Informação *Online*

Sistemas de informação online “clássicos” têm como objetivo fornecer informação de referência (diferentemente da hipermídia educacional) a usuários com diferentes níveis de conhecimento a respeito de um dado assunto. Cada nó do hiperespaço, em geral, representa um conceito do assunto e contém várias páginas de informação. Dependendo do assunto, o tamanho do hiperespaço pode variar de razoavelmente pequeno a muito grande. Assim como na hipermídia educacional, sistemas de informação *online* enfrentam o problema de ter que satisfazer as necessidades de diferentes usuários. Aqueles com diferente conhecimento e *background* necessitam de informação diferente, com diferentes níveis de detalhes sobre um conceito. Os usuários normalmente não têm tempo de navegar por toda a informação sobre um conceito procurando pela porção de informação desejada. Usuários também têm diferentes objetivos quando acessam um sistema de informação. Em alguns casos eles sabem quais conceitos acessar para alcançar seu objetivo e não precisam de suporte navegacional. Entretanto, quando o objetivo não pode ser diretamente mapeado para a estrutura do hiperespaço ou quando o hiperespaço é muito grande, os usuários precisam de ajuda navegacional e também precisam de ajuda para encontrar pedaços relevantes de informação. Para fornecer tal ajuda, o sistema tem que saber o objetivo do usuário.

Segundo Brusilovsky (2001), sistemas de informação *online* formam um grupo heterogêneo de sistemas que pode ser dividido em subgrupos. Em conjunto com os sistemas de informação *online* “clássicos”, esta categoria inclui um número de sistemas especializados tais como enciclopédias eletrônicas, quiosques informativos, museus virtuais, guias *handheld*, sistemas de *e-commerce* e sistemas de apoio de desempenho (ESCOLANET, 2007).

2.1.3 Sistemas Hipermídia de Recuperação de Informação

De acordo com Brusilovsky (2001), a *Web* tem influenciado significativamente a hipermídia de recuperação de informação. Todos os novos sistemas interessantes se inspiraram na *Web* e foram implementados para a *Web*. O grande desafio nesta classe de sistemas tem sido suportar a atividade do usuário de recuperar informação em um hiperespaço irrestrito como a *Web*. É possível distinguir dois grupos de sistemas dentro da hipermídia adaptativa de recuperação de informação: sistemas *orientados a pesquisa* e sistemas *orientados a navegação*.

O objetivo dos sistemas *orientados a pesquisa* é criar uma lista de *links* para documentos que satisfazem à informação corrente requisitada pelo usuário. Sistemas adaptativos de recuperação de informação levam em consideração não apenas o conjunto de palavras informadas pelo usuário, mas também um modelo do usuário que inclui interesses e preferências. Brusilovsky (2001) distingue dois tipos de sistemas orientados a pesquisa: *sistemas de recuperação de informação clássicos*, ou seja, os que lidam com um espaço de informação fechado, e os *filtros de pesquisa* que tentam trabalhar com um espaço de informação irrestrito como a *Web*.

Sistemas *orientados a navegação* auxiliam seus usuários no processo de navegação orientada a pesquisa. Tal auxílio é provido aplicando-se tecnologias padrão de suporte à navegação adaptativa (discutidas na seção 2.3.2). Sistemas adaptativos de

direcionamento marcam um ou mais *links* na página corrente que são mais relevantes para o objetivo do usuário. Sistemas adaptativos de *anotação* anexam diversas pistas visuais aos *links* na página corrente a fim de auxiliar o usuário a selecionar o mais relevante. Sistemas adaptativos de *recomendação* tentam deduzir os objetivos e interesses do usuário a partir de sua atividade de navegação e constroem uma lista de *links* sugeridos que apontam para nós que normalmente não poderiam ser alcançados diretamente a partir da página corrente, mas que são mais relevantes para o usuário.

Brusilovsky (1996-a) sugere uma categoria de sistemas que gerencia visões personalizadas dentro de espaços de informação, no entanto, Brusilovsky (2001) considera que tais sistemas pertencem ao universo da recuperação de informação e podem ser considerados complementares aos sistemas clássicos de recuperação de informação. Enquanto os sistemas de recuperação de informação auxiliam o usuário a localizar nós com informações relevantes, os sistemas para gerenciamento de visões personalizadas têm como objetivo organizar essa informação de alguma maneira.

2.1.4 Sistemas de Ajuda Online

Sistemas de ajuda online são conceitualmente muito próximos a sistemas de informação *online*. Estes sistemas provêm informação *online* sobre aplicativos (como por exemplo, processadores de texto, ambientes de programação e planilhas eletrônicas) que é necessária para ajudar os usuários a utilizar o sistema. A diferença entre sistemas de informação *online* e sistemas de ajuda *online* é que estes não são independentes como os sistemas de informação *online* pelo fato de estarem vinculados a seu aplicativo. Outra diferença é que o hiperespaço em sistemas de ajuda *online* existentes é razoavelmente pequeno. Sistemas de ajuda *online* e sistemas de informação *online* compartilham o problema de fornecer informação diferente a diferentes usuários. No entanto, o problema de ajudar usuários a encontrar pedaços relevantes de informação é menos importante em sistemas de ajuda *online*, pois o hiperespaço não é muito grande e o sistema sabe o contexto a partir do qual o usuário invocou o sistema de ajuda. O contexto de trabalho em um aplicativo provê uma fonte confiável de informação para um sistema de ajuda *online* adaptativo. Com isso, é possível determinar o objetivo do usuário e dessa forma oferecer itens de ajuda mais relevantes.

2.1.5 Sistemas de Informação Institucional

Sistemas de informação institucional provêm toda a informação necessária para suportar o trabalho de uma instituição, por exemplo, um hospital. Originalmente, estes sistemas foram desenvolvidos como um conjunto de bases de dados independentes, que vêm sendo agrupadas em um único hiperespaço que pode ser consideravelmente grande. Uma característica destes sistemas é que eles consistem em um meio de trabalho de muitos funcionários de uma instituição. De acordo com os papéis que desempenham na instituição, os funcionários podem utilizar sempre uma área específica do hiperespaço. De acordo com o objetivo de trabalho atual eles podem precisar acessar um subconjunto muito pequeno do hiperespaço. Muitos dos usuários nunca precisam acessar porções do hiperespaço fora de sua área de trabalho, aliás, muitas oportunidades de navegação de alguma forma acabam os distraindo e os tirando de seu trabalho principal. Neste aspecto, sistemas de informação institucional *orientados a trabalho* diferem de sistemas de recuperação de informação *orientados a pesquisa* e de sistemas de informação *online*, cuja “área de trabalho” de um usuário é todo o hiperespaço. Usuários de sistemas de informação institucional podem precisar de assistência para organizar um

acesso personalizado mais conveniente a suas áreas de trabalho. Outro problema associado a esta área de aplicação é que (de forma similar à hipermídia educacional) novos funcionários não estão familiarizados com a estrutura do hiperespaço e podem ficar perdidos mesmo em sua pequena subárea.

2.2 Aspectos do Usuário Levados em Consideração ao Adaptar

Brusilovsky (1996-a) propõe os seguintes questionamentos:

- Que aspectos do usuário que está trabalhando com o sistema devem ser levados em consideração ao prover a adaptação?
- A quais características – que podem ser diferentes para diferentes usuários (e podem ser diferentes para o mesmo usuário em épocas diferentes) – o sistema pode se adaptar?

Com o propósito de responder a esses questionamentos, Kobsa et al. (2001) sugere a distinção entre os diferentes tipos de dados que SHAs podem precisar considerar ao adaptar. Os tipos de dados são: dados sobre o usuário, dados de uso e dados sobre o ambiente de hardware, software e físico do usuário.

2.2.1 Dados do Usuário

Kobsa et al. (2001) introduz o termo *dados do usuário* para denotar informações sobre características pessoais do usuário, enquanto os *dados de uso* (subseção 2.2.2) estão relacionados com o comportamento (interativo) do usuário. Embora tratados como categorias diferentes, há sobreposições potenciais entre os dois tipos de dados.

Os dados do usuário podem ser categorizados como: dados demográficos do usuário, conhecimento do usuário, objetivo do usuário, *background* do usuário, experiência do usuário, preferências do usuário, interesses do usuário e traços individuais do usuário. Encontram-se descritas abaixo as categorias mencionadas acima.

2.2.1.1 Dados Demográficos do Usuário

Os dados demográficos do usuário são fatos objetivos do usuário como por exemplo:

- Dados de registro (como por exemplo, nome, endereço e número de telefone);
- Dados geográficos (como por exemplo, cidade, estado e país);
- Características do usuário (como por exemplo, idade e sexo).

2.2.1.2 Conhecimento do Usuário

Segundo Kobsa et al. (2001) o conhecimento do usuário sobre os conceitos, relacionamentos entre conceitos e fatos e regras que dizem respeito ao domínio do sistema tem sempre estado entre as mais importantes fontes de personalização. Para Brusilovsky (1996-a), o conhecimento do usuário a respeito do assunto representado no hiperespaço parece ser a característica mais importante do usuário para SHAs existentes. Um SHA que utiliza o conhecimento do usuário como fonte de adaptação deve reconhecer as mudanças no estado do mesmo e atualizar o modelo do usuário adequadamente.

Segundo Brusilovsky (1996-a), o conhecimento do usuário com relação a um assunto é mais freqüentemente representado por um modelo *overlay* (discutido na

subseção 2.5.2), que se baseia no modelo estrutural do domínio. No entanto, alguns sistemas utilizam o modelo estereótipo (discutido na subseção 2.5.2). Em geral, o modelo do domínio é representado como uma rede de *conceitos* do domínio. Os conceitos são relacionados entre si, formando uma espécie de rede semântica, que por sua vez representa a estrutura do domínio do assunto. Os conceitos podem ser nomeados diferentemente nos diversos sistemas – tópicos, elementos de conhecimento, objetos, resultados de aprendizagem – mas em todos os casos eles representam pedaços elementares de conhecimento para um dado domínio. Alguns sistemas utilizam uma forma simplificada de modelo do domínio, sem quaisquer *links* entre conceitos, mas a maioria dos SHAs faz uso de modelos do domínio mais avançados, com vários tipos de conceitos que representam diferentes tipos de elementos de conhecimento ou objetos, além de vários tipos de *links* que representam diferentes tipos de relacionamentos entre conceitos.

2.2.1.3 *Objetivo do Usuário*

O objetivo do usuário (ou tarefa do usuário) é uma característica relacionada ao contexto de trabalho do usuário em um sistema. Dependendo do tipo de sistema, pode ser o objetivo do trabalho (em sistemas de informação institucional), um objetivo de pesquisa (em sistemas de recuperação de informação), o objetivo de resolver um problema ou aprender um assunto (em sistemas educacionais). Independentemente do contexto, a idéia é saber o porquê de o usuário estar usando o sistema hipermídia e o que ele deseja realmente atingir. O objetivo do usuário é a característica mais mutante do usuário: quase sempre se modifica entre sessões de uso do sistema, podendo, inclusive, modificar-se várias vezes dentro de uma única sessão de uso. Em alguns sistemas é razoável distinguir objetivos locais ou de baixo nível, que podem mudar com frequência, de objetivos e tarefas gerais ou de alto nível, que são mais estáveis. Por exemplo, em sistemas educacionais o objetivo de aprendizagem é um objetivo de alto nível, enquanto que o objetivo de resolução de problemas é um objetivo de baixo nível, que muda de um problema para outro, possivelmente várias vezes dentro de uma sessão de uso. O objetivo do usuário pode ser considerado uma característica muito importante do usuário em SHAs. O objetivo corrente do usuário é normalmente modelado de maneira similar à modelagem *overlay* do conhecimento.

2.2.1.4 *Background e Experiência do Usuário*

O *background* do usuário diz respeito a toda informação relacionada à experiência prévia do usuário, fora do assunto que o sistema hipermídia abrange, que é relevante o suficiente para ser considerada. A profissão do usuário, experiência de trabalho em áreas relacionadas, bem como seu ponto de vista e perspectiva são exemplos de informações que compõem o *background* do usuário.

A experiência do usuário em um dado hiperespaço diz respeito ao quão familiar o usuário se sente com relação à estrutura do hiperespaço e quão fácil é para o usuário navegar por ele. Isto é diferente do conhecimento do usuário sobre o assunto. Em alguns casos, o usuário se sente muito familiarizado com o assunto propriamente dito, mas não com a estrutura do hiperespaço. De modo contrário, o usuário pode se sentir muito familiarizado com a estrutura do hiperespaço mesmo sem possuir profundo conhecimento do assunto.

O *background* e *experiência* são freqüentemente modelados por um modelo estereótipo do usuário.

2.2.1.5 Preferências do Usuário

Por diferentes razões o usuário pode preferir alguns nós e *links* ao invés de outros, algumas porções de uma página ao invés de outras. Estas são preferências do usuário. Preferências são mais fortemente usadas em sistemas hipermídia de recuperação de informação. Na maioria destes sistemas, as preferências são a única informação armazenada sobre o usuário. As preferências do usuário diferem de outros componentes do modelo do usuário em diversos aspectos. Diferentemente de outros componentes, as preferências não podem ser deduzidas pelo sistema, o usuário tem que informá-las direta ou indiretamente. Por esta razão, as preferências parecem estar mais próximas da adaptabilidade do que da adaptatividade (OPPERMANN, 1994).

2.2.1.6 Interesses do Usuário

De acordo com Brusilovsky (2001), os interesses do usuário não costumavam fazer parte do modelo do usuário em SHAs mais antigos. Essa situação mudou dramaticamente com o surgimento dos sistemas hipermídia de recuperação de informação na *Web*. Estes sistemas tentam modelar os interesses de longo prazo do usuário e os utilizam em paralelo com o objetivo de busca de curto prazo do usuário a fim de melhorar o filtro de informações e recomendações.

2.2.1.7 Traços Individuais do Usuário

Segundo Brusilovsky (2001), os traços individuais do usuário agrupam características do usuário que juntas definem um usuário como um indivíduo. Exemplos são fatores de personalidade (introverso / extroverso), fatores cognitivos e estilos de aprendizagem. Assim como o *background* do usuário, os traços individuais são características estáveis de um usuário que não podem ser modificadas como um todo ou podem ser modificadas apenas após um longo período de tempo. Diferentemente do *background* do usuário, traços individuais são tradicionalmente extraídos não apenas através de uma simples entrevista, mas através de testes psicológicos especialmente projetados.

2.2.2 Dados de Uso

Segundo KOBASA et al. (2001), dados de uso podem ser diretamente observados e gravados ou então obtidos através da análise de dados observáveis. A extensão em que a observação é tecnicamente possível varia consideravelmente. Os dados de uso diretamente observados podem levar diretamente à adaptação. No entanto, eles são frequentemente apenas ou adicionalmente utilizados como base para inferências que objetivam descobrir regularidades mais gerais em dados de uso observáveis. Estes dados por sua vez podem ser utilizados com o propósito de fazer com que o sistema se adapte, por exemplo, às preferências ou hábitos do usuário que se manifestam nessas regularidades.

2.2.2.1 Uso Observável

Dados de uso diretamente observados são obtidos através da interação do usuário com o sistema. Kobasa et al. (2001) identifica os seguintes tipos de interação:

- *Ações seletivas*: o clique em um *link* e o *scrolling* em um objeto hipermídia são exemplos de ações seletivas. Ações seletivas podem indicar que o usuário possui

interesse ou não está familiarizado com um determinado conteúdo, bem como revelar suas preferências;

- *Tempo de permanência*: o tempo de permanência do usuário em uma dada página pode indicar o interesse do usuário pelo conteúdo da página. No entanto, a medição do tempo efetivo de permanência é difícil. É frequentemente impossível determinar se o usuário se manteve presente na frente da tela do computador e se deteve em um item específico em um dado intervalo de tempo. Assim sendo, o tempo de permanência pode ser visto como uma evidência negativa: se o tempo de permanência em uma dada página é inferior a um limiar, então é provável que a informação nessa página não interessa ao usuário;
- *Avaliações*: em vários sistemas os usuários são solicitados a explicitamente avaliar objetos (documentos, notícias, produtos). Essas avaliações indicam o quão relevante ou interessante o objeto é para o usuário ou o quão relevante ou interessante o usuário pensa que o objeto é para os outros usuários;
- *Compras ou ações relacionadas à compra*: compras efetuadas em *web sites* de *e-commerce* são, em geral, consideradas fortes indicadores de interesse do usuário. Vários *web sites* comerciais reagem adaptativamente à compra sugerindo bens similares ou relacionados.

2.2.2.2 Regularidades de Uso

Segundo Kobsa et al. (2001), em muitos casos, interações observadas do usuário não levam diretamente a adaptações. Nesses casos, o processamento adicional dos dados de uso é necessário para se encontrar regularidades. Abaixo são descritos exemplos típicos de informações de uso que são adquiridas a partir de dados de uso observáveis:

- *Frequência de uso*: a idéia é categorizar eventos e contar suas frequências de uso. Com isso é possível, por exemplo, ocultar botões ou *links* não utilizados e exibir os mais utilizados;
- *Seqüências de ações*: são, em sua maioria, analisadas com o propósito de recomendar a geração de macros para seqüências de ações frequentemente utilizadas, predizer ações futuras com base nas ações do passado e recomendar ações com base em seqüências de ações freqüentes de outros usuários.

2.2.3 Dados do Ambiente

Brusilovsky (2001) afirma que a adaptação ao ambiente do usuário é um novo tipo de adaptação que foi trazida pelos sistemas baseados na *Web*. Considerando-se que os usuários de uma mesma aplicação *Web* podem residir virtualmente em qualquer lugar e podem utilizar diferentes equipamentos, a adaptação ao ambiente do usuário se tornou uma questão importante.

Kobsa et al. (2001) afirma que o uso da *Web* pode ser influenciado tanto pelo *software* quanto pelo *hardware* do usuário, assim como pelas características da localização corrente do usuário. Todas essas restrições são consideradas abaixo.

2.2.3.1 Ambiente de Software

Em se tratando de aplicações *Web*, o ambiente de *software* do usuário, em geral um navegador *Web*, nem sempre possui todos os recursos necessários para executar

adequadamente uma dada aplicação. Essa situação é muito comum em virtude dos diferentes navegadores existentes no mercado, além da possibilidade do usuário utilizar versões muito antigas de navegadores, que freqüentemente possuem recursos muito limitados. A existência de dispositivos móveis *web-enabled* aumenta ainda mais essa diversidade. Abaixo são elencadas algumas informações importantes que devem ser observadas ao se desenvolver e disponibilizar aplicações para a *Web*:

- *Versão de navegador e plataforma*: navegadores *Web*, conforme mencionado acima, diferem na extensão de recursos que suportam. A variedade de versões em uso é substancial, especialmente se um público global for considerado. Determinar o conjunto mínimo de recursos necessários para a execução da aplicação e avaliar as diferentes versões de navegadores a fim de verificar quais delas possui o conjunto mínimo de recursos identificados, torna-se cada vez mais importante;
- *Disponibilidade de plugins*: aplicações *Web* que desejam disponibilizar conteúdo *rich media* (MULTIMEDIA, 2007) precisam levar em consideração se o usuário tem ou não os *plugins* apropriados à disposição. Se o usuário não possui os *plugins* necessários à disposição então não poderá instalá-los. O sistema deve fazer com que o *web site* ainda seja útil para o usuário, mesmo sem a possibilidade de receber o conteúdo *rich media*;
- *Java e JavaScript*: os navegadores modernos, em geral, vêm equipados com Java e JavaScript, no entanto, diferentes navegadores possuem diferentes versões desses recursos. Outro problema é que o usuário pode voluntariamente desabilitar esses recursos no navegador, por questões de segurança por exemplo.

2.2.3.2 Ambiente de Hardware

O ambiente de *hardware* dos usuários pode variar profundamente. As maiores diferenças entre um ambiente e outro são as seguintes:

- *Largura de banda*: os usuários podem estar conectados à Internet através de *modem* analógico, DSL, *cable modem* e etc. Dependendo do tipo de conexão, a largura de banda disponível varia consideravelmente. A largura de banda disponível influencia o tempo de *download* e tempos de *download* curtos são fatores significantes para a satisfação do usuário com o *web site*;
- *Velocidade de processamento*: diz respeito à máquina propriamente dita que o usuário tem disponível. Mesmo que o usuário possua uma conexão com a Internet que disponha de muita largura de banda, é possível que haja restrições do processador ou memória da máquina que impeçam, por exemplo, que uma página *web* com muitas imagens seja visualizada;
- *Dispositivos de visualização*: com o advento dos pequenos dispositivos móveis, como os computadores *handheld* e os telefones celulares, houve um enorme incremento na quantidade e tipos de dispositivos de visualização. O tipo de conteúdo que é adequado para um visor de um computador pessoal com tela de 17 polegadas e resolução de 1024 x 768 *pixels*, pode e em geral não é adequado para o visor de um telefone celular;
- *Dispositivos de entrada*: com a crescente utilização de modernos dispositivos pequenos, torna-se mais difícil prover a entrada de dados em *web sites* interativos. Exemplo disso é selecionar um elemento de navegação cuja área

alvo seja muito pequena. Outro problema é que dispositivos de entrada não tradicionais podem oferecer somente um conjunto limitado de teclas para interação.

2.2.3.3 Local de Uso

Informações a respeito do local de uso da aplicação pelo usuário podem ser utilizadas, por exemplo, com o propósito de filtrar conteúdo e adaptar o conteúdo ao idioma e moeda do local. Informações a respeito do local de uso incluem:

- *Localização corrente do usuário*: a granularidade requerida da informação de localização varia enormemente, podendo ser desde o país em que o usuário se encontra até metros ou centímetros. Informações adicionais como direção do olhar e direção do movimento também podem ser relevantes. Um guia *handheld* de museu é um exemplo de aplicação que necessita de uma informação bastante apurada da localização corrente do usuário;
- *Características do local de uso*: informações que podem ser levadas em consideração incluem o nível de poluição sonora e lugares ou objetos próximos ao local.

2.3 Aspectos do Sistema que Podem ser Adaptados

Brusilovsky (1996-a) propõe os seguintes questionamentos:

- Quais aspectos de um sistema adaptativo podem ser adaptados?
- Quais características do sistema podem ser diferentes para diferentes usuários?
- Qual o espaço das possíveis adaptações?

Brusilovsky (1996-a) responde aos questionamentos acima afirmando que o espaço de adaptação na HA é bastante limitado, pois não há muitas características do sistema que podem ser alteradas. A hipermídia consiste em um conjunto de nós ou hiperdocumentos (páginas) conectados por *links*. Cada página contém informação local e um conjunto de *links* que apontam para páginas relacionadas. Sistemas hipermídia podem também incluir um índice e um mapa global que fornece *links* para todas as páginas acessíveis. Assim sendo, na HA podem ser adaptados:

- O conteúdo das páginas (adaptação em nível de conteúdo);
- *Links* de páginas, de índices e mapas (adaptação em nível de *link*).

Brusilovsky (1996-a) distingue adaptação em nível de conteúdo e adaptação em nível de *link* como duas diferentes classes da HA e denomina a primeira como *apresentação adaptativa* e a segunda como *suporte à navegação adaptativa*. A apresentação adaptativa e o suporte à navegação adaptativa são em conjunto denominados *tecnologias da HA*. A figura 2.3 exhibe a taxonomia das tecnologias da HA. As subseções seguintes descrevem a apresentação adaptativa e o suporte à navegação adaptativa em mais detalhes.

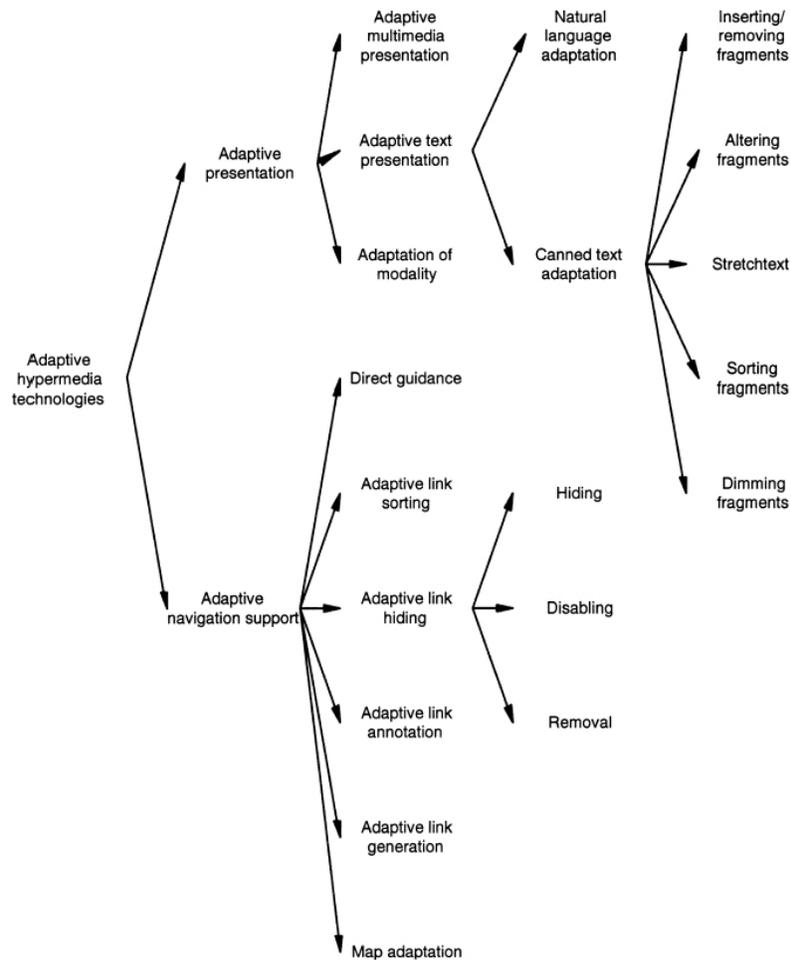


Figura 2.3: Taxonomia das tecnologias da HA (BRUSILOVSKY, 2001).

2.3.1 Apresentação Adaptativa

Brusilovsky (1996-a) afirma que a ideia por trás das diversas técnicas de apresentação adaptativa é adaptar o conteúdo de uma página acessada por um dado usuário a seu conhecimento atual, objetivos, dentre outras características. Por exemplo, um usuário qualificado pode receber informação mais detalhada e profunda ao passo que um iniciante pode receber explicações adicionais.

Brusilovsky (2001) subdivide a apresentação adaptativa em três grandes grupos: *apresentação adaptativa de texto*, *apresentação adaptativa de multimídia* e *adaptação de modalidade*.

2.3.1.1 Apresentação Adaptativa de Texto

De acordo com Brusilovsky (1997), a apresentação adaptativa de texto significa que diferentes usuários em diferentes épocas podem receber diferentes textos como conteúdo da mesma página.

Brusilovsky (2001) acredita ser apropriado dividir a apresentação adaptativa de texto em dois diferentes grupos: *adaptação de texto "enlatado"* (*canned text adaptation*) e

adaptação de língua natural. Evidentemente, muitos sistemas de geração de língua natural fazem uso de fragmentos (e até mesmo parágrafos) de texto “enlatado”. A distinção a que Brusilovsky (2001) se refere diz respeito aos sistemas que utilizam a tecnologia de língua natural como seu fundamento e aqueles que não o fazem.

As principais formas de adaptação de texto “enlatado” incluem: inserção / remoção de fragmentos, alteração de fragmentos, *stretchtext*, ordenamento de fragmentos e escurecimento (*dimming*) de fragmentos. Estas formas de adaptação são descritas abaixo:

- *Inserção / remoção de fragmentos*: segundo Cristea (2007), dentre as muitas formas de adaptação de texto, a tecnologia de inserção e remoção de fragmentos é a mais popular. Isto se deve ao fato de que esta tecnologia é fácil de implementar. Uma condição pode ser associada a um fragmento, ou seja, uma expressão *booleana* que avalia valores oriundos do modelo do usuário. Tal condição determina se um fragmento deverá ser exibido ou não;
- *Alteração de fragmentos*: segundo Cristea (2007), esta tecnologia pode ser usada para selecionar um fragmento dentre um conjunto de fragmentos contendo explicações alternativas sobre um dado conceito. A idéia é que os diferentes fragmentos foram projetados para atender diferentes tipos de usuários;
- *Stretchtext*: segundo Brusilovsky (1996-a), *stretchtext* é uma tecnologia que consiste em um tipo especial de hipertexto que possibilita apresentar ou não diferentes porções de conteúdo de acordo com o nível de conhecimento do usuário. Em hipertexto regular, ativar uma *hotword* resulta em apresentar outra página com texto relacionado. Em *stretchtext*, este texto relacionado pode simplesmente substituir a *hotword* ativada, estendendo o texto da página corrente. Se desejado, o texto estendido pode ser ocultado e a *hotword* voltar a ser apresentada. Brusilovsky (1996-a) destaca que uma importante característica desta técnica é que ela permite que tanto o usuário quanto o sistema adaptem o conteúdo de uma página em particular e que tal adaptação pode levar em consideração o conhecimento e as preferências do usuário. O sistema apresenta a página *stretchtext*, que pode ser adaptada pelo usuário, estendendo ou ocultando explicações e detalhes de acordo com suas preferências. O sistema então atualiza o modelo do usuário, de acordo com as preferências demonstradas pelo usuário, a fim de assegurar que ele sempre irá ver a combinação preferida de trechos ocultados e estendidos. Por exemplo, se o usuário ocultou as explicações adicionais de um dado conceito, o sistema irá sempre exibir as explicações adicionais deste conceito ocultadas, até que o usuário modifique suas preferências;
- *Ordenamento de fragmentos*: segundo Cristea (2007), esta tecnologia é mais útil quando um conjunto de fragmentos independentes pode ser apresentado em qualquer ordem. O ordenamento pode ser executado com o propósito de obter uma classificação dos fragmentos por relevância ou então para exibir um exemplo antes de uma explicação e vice-versa. Alunos com diferentes estilos cognitivos podem preferir uma ordem diferente de descrições ou explicações de um conceito;
- *Escurecimento de fragmentos*: a tecnologia de escurecimento de fragmentos faz com que um fragmento não relevante para o usuário seja exibido de forma não

ênfatizada. Algumas formas de aplicar esta tecnologia são: fazer com que o fragmento não relevante seja exibido em tom acinzentado ou com tamanho de fonte reduzida. Segundo Cristea (2007), a idéia por trás desta tecnologia é que o usuário seja estimulado a não ler o texto escurecido, embora possa lê-lo caso desejado.

2.3.1.2 Apresentação Adaptativa de Multimídia

Sistemas que suportam a apresentação adaptativa de multimídia alteram a apresentação de elementos multimídia com base no modelo do usuário. Carver et al. (1999) afirma que a adaptação de multimídia adiciona complexidade e requer um maior esforço de implementação. Elementos de mídia não textuais são mais difíceis de gerar e de recombinar automaticamente. É extremamente difícil, por exemplo, adaptar segmentos de vídeo automaticamente e apresentar os resultados para o usuário. Técnicas para adaptação e apresentação de elementos de mídia não textuais são questões de pesquisa em aberto.

2.3.1.3 Adaptação de Modalidade

Segundo Kobsa et al. (2001), um tipo especial de modificação na apresentação é a modificação de modalidade, por exemplo: de imagens para texto, de texto para áudio ou de vídeo para imagens. Brusilovsky (2001) denomina este tipo de adaptação de adaptação de modalidade e a considera uma tecnologia de adaptação de conteúdo de alto nível. SHAs modernos podem optar por diferentes tipos de mídia com as quais apresentar informação para o usuário, isto é, em adição ao texto tradicional, pode-se utilizar, por exemplo, música, vídeo, narração e animação. A idéia é ter o mesmo conteúdo representado em diferentes tipos de mídia. Dessa forma, o sistema pode escolher a mídia que é mais relevante para o usuário no nó em questão. Outra possibilidade é utilizar diferentes tipos de mídia em paralelo, de forma que o sistema possa escolher o subconjunto de mídias mais relevantes para o usuário em um dado nó.

2.3.2 Suporte à Navegação Adaptativa

Segundo Brusilovsky (1996-a), a idéia por trás das técnicas de suporte à navegação adaptativa é ajudar os usuários a encontrar caminhos no hiperespaço adaptando a forma de apresentação dos *links* a seus objetivos, conhecimento, dentre outras características. Brusilovsky (2001) classifica tais tecnologias em seis grupos: direcionamento direto, ordenamento adaptativo de *link*, ocultamento adaptativo de *link*, anotação adaptativa de *link*, geração adaptativa de *link* e adaptação de mapa.

Antes de apresentar as tecnologias de suporte à navegação adaptativa, é válido compreender como e em quais contextos os *links* são em geral apresentados. Com esse intuito, Brusilovsky (1996-a) distingue quatro tipos de apresentação de *links* que são diferentes, dada a perspectiva do que pode ser alterado e adaptado:

- *Links locais não contextuais*: esta categoria inclui todos os tipos de *links* em páginas hipermídia que são independentes do conteúdo da página. Podem aparecer como um conjunto de botões, uma lista ou um menu *pop-up*. São *links* facilmente manipuláveis, podem ser ordenados, ocultados ou anotados;
- *Links contextuais ou links “real hipertext”*: esta categoria compreende “*hotwords*” em textos, “*hot spots*” em figuras e outros tipos de *links* que são embutidos no contexto do conteúdo da página e não podem ser dela removidos.

Estes *links* podem ser anotados, mas não podem ser ordenados ou completamente ocultados;

- *Links de página de índice*: uma página de índice pode ser considerada um tipo especial de página que contém unicamente *links*. Estes *links* são normalmente apresentados em ordem fixa (por exemplo, alfabética). Frequentemente, *links* em páginas de índice são não contextuais;
- *Links em mapas locais e links em mapas globais do hiperespaço*: mapas geralmente representam graficamente um hiperespaço ou uma área local do hiperespaço como uma rede de nós conectados por setas. Usando mapas, o usuário pode navegar diretamente para todos os nós visíveis, bastando para isso um clique na representação do nó desejado.

Em seguida, são apresentados os seis grupos de tecnologias de suporte à navegação adaptativa.

2.3.2.1 Direcionamento Direto

Direcionamento direto é a mais simples das tecnologias aqui apresentadas. Pode ser aplicada em qualquer sistema que é capaz de decidir qual é o melhor nó para o usuário visitar de acordo com seu objetivo e outros parâmetros representados no modelo do usuário. Para prover direcionamento direto, o sistema pode contornar o *link* que aponta para o melhor nó ou então acrescentar um *link* dinamicamente gerado (frequentemente denominado “próximo”) que aponta para o melhor nó. Direcionamento direto é uma tecnologia clara e fácil de implementar, além de poder ser utilizada com todos os tipos de *links* apresentados acima. O problema é que o direcionamento direto provê suporte limitado: ou o usuário segue o caminho apontado pelo sistema ou não há ajuda. O direcionamento direto dificilmente pode ser a forma principal de suporte à navegação porque não oferece suporte a usuários que não gostariam de seguir a sugestão do sistema. O direcionamento direto é útil, mas deve ser usado com uma tecnologia que forneça maior suporte navegacional para o usuário.

2.3.2.2 Ordenamento Adaptativo de Link

O ordenamento adaptativo de *link* consiste em ordenar todos os *links* de uma página em particular, de acordo com o modelo do usuário e algum critério de valor para o usuário: quanto mais próximo do topo, mais relevante é o *link*. O ordenamento adaptativo tem aplicação limitada: pode ser utilizado com *links* não contextuais, dificilmente pode ser utilizado em páginas de índice e nunca pode ser utilizado com *links* contextuais e mapas. Outro problema associado a esta tecnologia é que ela torna a ordem dos *links* instável, podendo mudar a cada acesso do usuário à página.

2.3.2.3 Ocultamento Adaptativo de Link

O ocultamento adaptativo de *link* consiste em restringir o espaço de navegação ocultando *links* para páginas não relevantes. Uma página pode ser considerada não relevante por não estar relacionada ao objetivo corrente do usuário ou por apresentar conteúdo que o usuário ainda não está preparado para compreender. O ocultamento protege os usuários da complexidade do hiperespaço irrestrito e reduz sua sobrecarga cognitiva. O ocultamento tem larga aplicabilidade: pode ser utilizado com todos os tipos de *links* não contextuais, de índice e mapa, ocultando de fato botões ou itens de menu; e com *links* contextuais, fazendo com que “*hotwords*” se transformem em texto normal. O

ocultamento é também mais transparente e parece mais “estável” para o usuário do que o ordenamento, dado que os *links* são em geral adicionados incrementalmente, não sendo removidos ou reordenados.

De Bra e Calvi (1998) sugeriram e implementaram diferentes variações de ocultamento adaptativo de *link*:

- *Ocultamento*: os *links* considerados não relevantes permanecem visíveis e funcionais, porém com aparência igual à de texto convencional;
- *Remoção*: os *links* considerados não relevantes são removidos. Esta tecnologia funciona razoavelmente bem em menus, mas não em *links* que aparecem dentro de parágrafos de texto;
- *Desabilitação*: os *links* considerados não relevantes são desabilitados, ou seja, permanecem visíveis, porém não funcionais.

Brusilovsky (2001) acredita que as variações acima deveriam ser classificadas como tecnologias independentes dentro de uma tecnologia mais geral de ocultamento, conforme exibido na figura 2.3.

2.3.2.4 Anotação Adaptativa de Link

A idéia da tecnologia de anotação adaptativa de *link* é adornar os *links* com alguma forma de comentário que forneça ao usuário mais informação a respeito do estado corrente dos nós por trás dos *links* anotados. Essas anotações podem ser oferecidas em forma textual ou em forma de pistas visuais, usando, por exemplo, diferentes ícones, cores ou tamanhos de fontes.

De acordo com Brusilovsky (1996-a), a anotação de *link* é conhecida como uma tecnologia efetiva de suporte navegacional na hipermídia. Tipicamente a anotação utilizada na hipermídia tradicional é estática (independente do usuário). O suporte à navegação adaptativa pode ser provido pela anotação dinâmica dirigida pelo modelo do usuário. Mesmo em sua forma mais simples, em que a anotação adaptativa é capaz de distinguir apenas dois estados dos *links* (*links* para nós visitados e não visitados), parece ser bastante útil. SHAs são capazes de distinguir e anotar diferentemente mais do que dois estados com base no modelo do usuário.

Brusilovsky (1996-a) afirma que a anotação parece ser uma forma muito relevante de suporte à navegação adaptativa, podendo ser utilizada com todos os tipos de *links*. Esta tecnologia mantém o ordenamento estável dos *links* e evita problemas com mapas mentais incorretos. A anotação é em geral uma tecnologia mais poderosa que o ocultamento: o ocultamento é capaz de distinguir apenas dois estados para os nós – relevantes e não relevantes – enquanto que a anotação, conforme mencionado, é capaz de distinguir mais estados. As anotações não restringem a sobrecarga cognitiva tanto quanto o ocultamento, mas a tecnologia de ocultamento pode ser simulada pela tecnologia de anotação utilizando um tipo de “escurecimento”, que torna o *link* turvo ou menos perceptível, ao invés de ocultamento para *links* não relevantes. O escurecimento pode diminuir a sobrecarga cognitiva em alguma extensão (o usuário pode aprender a ignorar *links* escurecidos), mas ainda assim os *links* escurecidos são visíveis e funcionais, o que protege o usuário de formar mapas mentais incorretos.

2.3.2.5 Geração Adaptativa de Link

Segundo Brusilovsky (2001), com o surgimento dos sistemas de recomendação é necessário distinguir entre duas formas essencialmente diferentes de suporte à navegação adaptativa: adaptação a *links* que estavam presentes na página no momento em que o hiperespaço foi autorado e *geração* de novos *links* não autorados para a página. A geração de *link* inclui três casos: descobrir novos *links* úteis entre documentos e adicioná-los permanentemente ao conjunto de *links* existentes; gerar *links* entre itens para a navegação baseada em similaridade; e recomendação dinâmica de *links* relevantes.

2.3.2.6 Adaptação de Mapa

Segundo Cristea (2007), com o propósito de fornecer ao usuário uma idéia de todo o hiperespaço e prover algum suporte à orientação (orientação no sentido de auxiliar o usuário a entender sua posição relativa ou absoluta dentro do hiperespaço), muitas aplicações oferecem algum tipo de mapa.

Mapas podem ser textuais ou gráficos. *Web sites* freqüentemente oferecem um mapa textual, principalmente porque sua geração é mais fácil do que a de um mapa gráfico. Um mapa gráfico, preferivelmente baseado em relacionamentos entre conceitos ao invés de relacionamentos entre *links*, é uma excelente ferramenta para introduzir o usuário na estrutura do hiperespaço.

Mapas geralmente tendem a ser muito grandes e esse aspecto pode dificultar sua compreensão. Assim sendo, um mapa pode ser adaptativamente reduzido tal que o usuário possa capturar toda a estrutura do hiperespaço que ele representa. Nós em mapas podem ser anotados para indicar relevância, para indicar nós já visitados pelo usuário e indicar também nós já visitados por outros usuários.

2.4 Métodos da Hipermídia Adaptativa

2.4.1 Métodos da Apresentação Adaptativa

2.4.1.1 Explicações Adicionais

Segundo Brusilovsky (1996-a), o objetivo do método mais popular de adaptação de conteúdo, denominado explicações adicionais, é ocultar do usuário algumas porções de informação sobre um dado conceito que não são relevantes para o nível de conhecimento do usuário com relação ao conceito. Por exemplo, detalhes de baixo nível podem ser ocultados de usuários com baixo nível de conhecimento de um conceito, porque eles não são capazes de compreender estes detalhes. Por outro lado, explicações adicionais, freqüentemente necessárias para iniciantes, cujo propósito é facilitar a compreensão de um conceito, podem ser ocultadas de um usuário com um bom nível de conhecimento do conceito, porque eles não mais precisam destas explicações.

Em termos mais gerais, em adição à apresentação básica, algumas categorias de usuários podem receber alguma informação adicional que é preparada especialmente para esta categoria de usuários e que não será exibida para usuários de outras categorias. Uma variação deste método, baseada em objetivo, consiste em ocultar do usuário algumas porções de informação (relativas a um conceito) que não são relevantes, dado o objetivo corrente do usuário.

2.4.1.2 Explicações de Pré-requisitos e Explicações Comparativas

Explicações de pré-requisitos e explicações comparativas modificam a informação apresentada sobre um conceito com base no nível de conhecimento do usuário sobre os conceitos relacionados. O primeiro método se baseia nas relações de pré-requisitos entre conceitos. A idéia é inserir explicações de todos os conceitos pré-requisitos que não são suficientemente conhecidos do usuário antes de apresentar a explicação do conceito propriamente dito. O segundo método se baseia em relações de similaridades entre conceitos. Se um conceito similar ao conceito que está sendo apresentado é conhecido, o usuário recebe uma explicação comparativa que destaca similaridades e diferenças entre o conceito atual e o relacionado.

2.4.1.3 Variações de Explicação

O método variações de explicação, assim denominado por Brusilovsky (1996-a), assume que mostrar ou ocultar uma determinada porção de conteúdo nem sempre é suficiente para a adaptação, pois diferentes usuários podem precisar de informações diferentes. Neste método, o sistema armazena diversas variações de algumas porções de conteúdo da página e o usuário recebe a variação que corresponde a seu modelo do usuário.

2.4.1.4 Ordenamento

Ordenamento é um método que leva em consideração o *background* e nível de conhecimento do usuário. A idéia é ordenar fragmentos de informação sobre um conceito, posicionando a informação mais relevante (de acordo com o *background* e nível de conhecimento do usuário) no topo.

2.4.2 Métodos do Suporte à Navegação Adaptativa

Segundo Brusilovsky (1996-a), as técnicas de suporte à navegação adaptativa são usadas para atingir diversos objetivos de adaptação: fornecer direcionamento global, fornecer direcionamento local, suportar orientação local, suportar orientação global e auxiliar no gerenciamento de visões personalizadas em espaços de informação. Em geral, estes objetivos são diferentes, mas ao mesmo tempo cada par de objetivos vizinhos nesta lista tem algo em comum. Assim sendo, é mais uma seqüência contínua de objetivos, em que as fronteiras entre os vizinhos não são tão nítidas e alguns métodos e técnicas servem a mais de um objetivo.

2.4.2.1 Direcionamento Global

O direcionamento global pode ser oferecido em sistemas hipermídia nos quais os usuários têm algum objetivo “global” de informação (ou seja, precisam de informação que está contida em um ou vários nós do hiperespaço) e a navegação é o meio de encontrar a informação desejada. O objetivo dos métodos de direcionamento global é auxiliar o usuário a encontrar o caminho mais curto que o leve à informação objetivo, com o mínimo de dificuldade. Direcionamento global é o objetivo primário do suporte à navegação adaptativa em sistemas hipermídia de recuperação de informação e é também um importante objetivo de sistemas de ajuda *online* e de sistemas de informação *online* com hiperespaços razoavelmente grandes. A informação objetivo, que é, em geral, fornecida clara ou parcialmente pelo usuário, é a característica fundamental para o direcionamento adaptativo.

O método mais direto de fornecer direcionamento global é sugerir ao usuário, a cada passo de navegação, quais *links* seguir a partir do nó corrente. Um método que fornece maior suporte ao usuário, consiste em aplicar a tecnologia de ordenamento adaptativo e ordenar todos os *links* do nó corrente de acordo com sua relevância para o objetivo global (o *link* mais relevante deve ser o primeiro). Com este método, os usuários têm a possibilidade de navegar pelo *link* mais relevante, mas têm também alguma informação a mais (a relevância dos *links*) para fazerem uma livre escolha.

Para fornecer direcionamento global na hipermídia educacional, o sistema deve saber não apenas o objetivo global do usuário (que é um objetivo de aprendizagem), mas também o estado corrente de conhecimento do usuário com relação ao assunto que está sendo ensinado. Segundo Brusilovsky (1996-a), o método mais popular de fornecer direcionamento global na hipermídia educacional é o direcionamento direto através do botão dinâmico “próximo”. O botão “próximo”, na hipermídia educacional, ativa um mecanismo de sequenciamento que seleciona o nó com material educacional mais relevante, com base no conhecimento corrente do usuário, objetivo de aprendizagem e estratégia de tutoramento do sistema.

2.4.2.2 Direcionamento Local

O objetivo dos métodos de direcionamento local é o de auxiliar o usuário a executar um passo de navegação, sugerindo os *links* mais relevantes a seguir a partir do nó corrente. Este objetivo é similar ao objetivo do direcionamento global, porém mais modesto. Os métodos fazem a sugestão de acordo com as preferências, conhecimento e *background* do usuário, ou segundo o que for mais importante para a área de aplicação considerada.

2.4.2.3 Orientação Local

O objetivo dos métodos de suporte à orientação local é de auxiliar o usuário a compreender o que está em volta dele e qual sua posição relativa no hiperespaço. Segundo Brusilovsky (1996-a), SHAs existentes implementam o suporte à orientação local de duas formas: provendo informação adicional sobre os nós disponíveis a partir do nó corrente (utilizando a tecnologia de anotação) ou limitando o número de oportunidades de navegação com o propósito de reduzir a sobrecarga cognitiva e permitir que os usuários se concentrem em analisar os *links* mais relevantes (utilizando a tecnologia de ocultamento).

Métodos baseados na tecnologia de ocultamento, ocultam do usuário todos os *links* que não são relevantes para ele em um dado momento, ou seja, exibem somente os *links* relevantes. Os métodos diferem quanto aos princípios usados para decidir que *links* são relevantes ou não. Esta decisão pode ser feita de acordo com o conhecimento, objetivos, experiência ou preferências do usuário. Dois métodos baseados na tecnologia de ocultamento e que são específicos para a hipermídia educacional são: ocultar *links* para nós que ainda não estão prontos para serem aprendidos (isto geralmente significa que estes nós têm nós pré-requisitos ainda não aprendidos) e ocultar *links* para nós que pertencem a objetivos educacionais de conteúdos subseqüentes e não pertencem ao objetivo educacional corrente.

A idéia dos métodos baseados na tecnologia de anotação é informar o usuário sobre o estado corrente dos nós por trás dos *links* visíveis. A anotação pode ser usada para mostrar os vários graus de relevância dos *links* (muito relevante, relevante ou pouco

relevante) e para refletir os vários níveis de conhecimento do usuário dos nós por trás dos *links* anotados (não conhecido, em aprendizagem ou bem conhecido). Dois outros métodos usam a anotação em situações em que o ocultamento seria normalmente usado. O primeiro contorna os *links* relacionados ao objetivo corrente, já o segundo provê uma anotação especial para *links* que apontam para nós que não estão prontos para serem aprendidos. Estes dois últimos métodos mostram que em muitos casos a tecnologia de ocultamento pode ser implementada com a tecnologia de anotação, seja contornando os *links* relevantes ou escurecendo os *links* não relevantes. No entanto, a anotação dificilmente pode ser utilizada nos casos em que o número de *links* visíveis é muito grande (e o ocultamento ainda é aplicável).

Segundo Brusilovsky (1996-a), os métodos de suporte à orientação local não direcionam o usuário diretamente, mas auxiliam no entendimento de quais são os *links* próximos e em fazer escolhas de navegação bem fundamentadas. Sob outra perspectiva, os métodos de direcionamento global e local, baseados na tecnologia de ordenamento, podem suportar a orientação local até certa extensão, mas não tão efetivamente quanto os métodos específicos de suporte à orientação, descritos acima, pois a posição de um *link* em uma lista ordenada fornece ao usuário muito pouca informação adicional sobre este *link*.

2.4.2.4 Orientação Global

O objetivo dos métodos de suporte à orientação global é auxiliar o usuário a compreender a estrutura do hiperespaço como um todo e sua posição absoluta nele. Na hipermídia não adaptativa, este objetivo é normalmente atingido oferecendo-se marcos visuais e mapas globais que podem diretamente auxiliar o usuário. Uma outra maneira é oferecer *tours* guiados para ajudar o usuário a conhecer gradualmente o hiperespaço. A HA é capaz de oferecer maior suporte para o usuário nestas mesmas linhas aplicando as tecnologias de ocultamento e anotação. De fato, todos os métodos de ocultamento e anotação, considerados acima, suportam também a orientação global do usuário.

Anotações funcionam como marcos: considerando-se que um nó mantém a mesma anotação quando visto pelo usuário a partir de diferentes posições no hiperespaço, o usuário pode reconhecer mais facilmente os nós que ele encontrou previamente e compreender sua posição atual. O ocultamento reduz o tamanho do hiperespaço visível e pode simplificar tanto a orientação quanto a aprendizagem. Em áreas de aplicação como a hipermídia educacional ou institucional, nas quais o hiperespaço não é especialmente grande, o ocultamento pode efetivamente suportar a aprendizagem gradual do hiperespaço. Os métodos de ocultamento úteis para este propósito são aqueles que gradualmente exibem ao usuário partes incrementalmente maiores do hiperespaço. Exemplos destes métodos para a hipermídia educacional são: ocultar nós não prontos para serem aprendidos e ocultar nós que são objetivos educacionais de conteúdos subseqüentes. Um exemplo de método mais universal é o que exhibe gradualmente mais *links* de acordo com o crescimento da experiência do usuário em um dado hiperespaço.

2.4.2.5 Visão Personalizada

Uma visão personalizada é um meio de organizar um ambiente de trabalho eletrônico para os usuários que necessitam acessar uma porção relativamente pequena do hiperespaço em seu trabalho diário. A maneira usual de proteger esses usuários da complexidade de todo o hiperespaço é deixá-los organizar visões personalizadas

orientadas a objetivo. Cada visão é composta por um conjunto de *links* que apontam para todos os hiperdocumentos que são relevantes para um objetivo de trabalho em particular. Soluções baseadas em adaptabilidade permitem que os usuários criem e gerenciem suas visões personalizadas (através de *bookmarks*, por exemplo). Já as soluções adaptativas, ou seja, soluções em que o gerenciamento das visões personalizadas é suportado pelo sistema, os *links* podem aparecer, desaparecer ou evoluir de acordo com a relevância dos mesmos para o usuário.

2.5 Modelagem do Usuário

Segundo Eklund e Zeiliger (1996), a adaptação do conteúdo e o suporte à navegação adaptativa são desempenhados dentro do sistema com base nas informações armazenadas no modelo do usuário, ou seja, a representação do sistema de preferências, conhecimento, crenças ou objetivos do usuário. Em função disso, para Koch (2000), o modelo do usuário constitui parte relevante de um SHA.

Paiva et al. (1995) definem *modelos do aluno (learner models)* como representações de algumas características e atitudes dos alunos que são úteis para atingir a interação adequada e individualizada estabelecida entre ambientes computacionais e alunos. Modelos do aluno são constituídos de descrições do que é considerado relevante sobre o estado atual de conhecimento de um aluno e provêm informações para que o ambiente de aprendizagem se adapte ao aluno individual. Segundo Koch (2000), trocando-se o termo *aluno* por *usuário*, a definição acima também pode ser aplicada aos modelos do usuário.

Segundo Koch (2000):

- Um *modelo* é definido como uma representação abstrata de algo do mundo real. Esta representação é abstrata porque somente algumas propriedades relevantes para a aplicação são inclusas no modelo;
- Um *modelo do usuário*, o elemento real é o usuário, é representado como uma coleção de dados. É a representação explícita dos aspectos do usuário. Principalmente as crenças do sistema a respeito do usuário são retratadas;
- Uma *modelagem do usuário* é um processo que abrange todo o ciclo de vida de um modelo do usuário. A modelagem inclui a aquisição de conhecimento sobre o usuário, construção, atualização, manutenção e utilização do modelo do usuário.

Koch (2000) afirma que um modelo do usuário é necessário porque sem ele o sistema irá se comportar exatamente da mesma forma para todos os usuários. Mas usuários são diferentes: eles têm diferentes *backgrounds*, diferentes níveis de conhecimento sobre um assunto, diferentes preferências, objetivos e interesses. Para individualizar, personalizar ou customizar ações, um modelo do usuário é necessário de tal forma que permita a seleção de respostas individualizadas para o usuário.

De acordo com Koch (2000), um modelo do usuário não pode ser nunca completamente preciso, normalmente é uma aproximação bruta. Embora incompleto, o modelo do usuário pode ser útil. O esforço para melhorar sua precisão deve ser comparado aos benefícios de seu melhoramento. Desta forma, os modelos do usuário devem sempre ser considerados dentro do contexto em que serão utilizados e de acordo com os objetivos da aplicação que se baseia nesses modelos.

Em um contexto educacional, Ragnelmalm (1995) afirma que as informações que o modelo do aluno deveria ser projetado para conter, dependem do conjunto de tarefas instrucionais a serem suportadas bem como das características do conhecimento a ser ensinado. Por exemplo: se o sistema deve adaptar sua forma de ensino ao estilo de aprendizagem preferido do aluno, então ele precisa saber as preferências desse aluno; se o sistema pretende aumentar a motivação do aluno, então ele precisa saber o que motiva esse aluno.

2.5.1 Características de Modelos do Usuário

Segundo Koch (2000), modelos do usuário podem ser classificados como modelos:

- *Adaptativos ou adaptáveis*: modelos do usuário adaptativos são atualizados automaticamente pelo sistema de acordo com a informação obtida a partir do comportamento do usuário, como por exemplo: erros, utilização de sistemas de ajuda ou o diálogo entre usuário e computador. Modelos do usuário adaptáveis são modificados apenas pelo usuário, ou seja, o usuário é quem decide quando o sistema será adaptado para o mais novo estado do modelo do usuário;
- *Atualizados através de questionamento ou observação*: informações para inicializar ou atualizar modelos do usuário podem ser obtidas questionando ou observando o usuário. Comumente, opta-se pela combinação de ambas as técnicas com o objetivo de adquirir tanta informação quanto possível a respeito do usuário;
- *Dinâmicos ou estáticos*: modelos do usuário dinâmicos são constantemente atualizados com as informações adquiridas. Em modelos do usuário estáticos, as informações são coletadas (através de questionamento ou observação) e a atualização é feita apenas em fase inicial ou em intervalos regulares;
- *Agrupados ou individuais*: modelos para grupos representam características do usuário baseadas em estereótipo, enquanto que modelos individuais são especialmente gerados para cada usuário;
- *Internos ou externos*: modelos do usuário são considerados internos se estão embutidos no sistema, senão são considerados externos;
- *Explícitos ou implícitos*: modelos do usuário são explícitos se possuem representações separadas do modelo do domínio. Modelos do usuário são implícitos se estão inclusos na representação do modelo do domínio;
- *Visíveis ou opacos*: modelos do usuário são visíveis para o usuário quando as suposições que o sistema faz a respeito do usuário estão visíveis para o mesmo. Modelos do usuário são ditos visíveis e modificáveis se o usuário pode modificá-los explicitamente. Se o usuário não tem permissão para ver ou modificar os modelos do usuário, então esses modelos são ditos opacos;
- *De baixa ou alta granularidade*: modelos do usuário são descritos como de baixa ou alta granularidade dependendo do nível de detalhe dos assuntos representados nesses modelos;
- *De curto ou longo prazo*: modelos de curto prazo permanecem disponíveis até que a sessão de uso do sistema encerre. Modelos de longo prazo são mantidos pelo sistema entre uma sessão e outra.

2.5.2 Tipos de Modelos do Usuário

Koch (2000) afirma que os critérios mais freqüentemente utilizados para classificar modelos do usuário são: a natureza de seu conteúdo, o tipo de representação e os métodos usados para inicializar, construir e utilizar modelos do usuário. Abaixo tais critérios são descritos.

2.5.2.1 Natureza do Conteúdo dos Modelos do Usuário

Três tipos de modelos baseados em conteúdo podem ser caracterizados:

- *Modelo de conhecimento do domínio*: contém conhecimento que o sistema assume que o usuário tem sobre o domínio. Benyon e Murray (1993 apud KOCH, 2000) distinguem três níveis dentro deste modelo: nível de tarefa, lógico e físico. O nível de tarefa descreve os objetivos do usuário no domínio. O nível lógico registra o que o sistema acredita que o usuário compreende sobre os conceitos lógicos embutidos no domínio. O nível físico registra o conhecimento inferido do usuário;
- *Conhecimento independente do domínio*: contém conhecimento geral ou não específico do domínio bem como áreas de interesse e *background* do usuário. Estes dados não são de natureza psicológica, mas podem ser importantes para o modelo do usuário como um todo;
- *Modelo psicológico ou cognitivo*: preocupa-se com preferências, habilidades, deficiências e características de personalidade do usuário. Por exemplo, o sistema tem que saber as preferências do aluno Ragnemalm (1995) se ele pretende adaptar sua forma de ensino: ao estilo de aprendizagem ou estratégia preferida pelo usuário; à técnica de motivação que é mais efetiva para o usuário; ao tipo de raciocínio do usuário (indutivo ou dedutivo); ou ao nível de concentração do usuário. Usuários que diferem nestes estilos cognitivos requerem diferentes adaptações. Normalmente as características do modelo cognitivo são de longo prazo, pois são mais resistentes a modificações.

2.5.2.2 Estrutura Interna dos Modelos do Usuário

Brusilovsky (1996-b) considera que o coração de muitos SHAs é um conjunto de conceitos do domínio. Estes conceitos podem ser denominados diferentemente em diferentes sistemas – tópicos, elementos de conhecimento, objetos ou resultados de aprendizagem – mas em todos os casos eles são apenas pedaços elementares de conhecimento de um dado domínio. Dependendo do domínio e da área de aplicação, os conceitos podem representar pedaços maiores ou menores do conhecimento do domínio. O conjunto de conceitos do domínio é freqüentemente denominado *modelo do domínio*.

Brusilovsky (1996-b) caracteriza o modelo do domínio em três níveis:

- *Nível um*: é o nível mais simples no qual o modelo do domínio é composto de um conjunto de conceitos independentes;
- *Nível dois*: é um nível mais avançado do que o anterior em que o modelo do domínio consiste em um conjunto de conceitos relacionados um ao outro, formando, desta forma, uma espécie de rede semântica;
- *Nível três*: é o nível mais avançado, dentre os anteriores, em que o modelo do domínio utiliza uma representação do domínio baseada em *frames*, isto é,

representa a estrutura interna de cada conceito como um conjunto de atributos, sendo que diferentes tipos de tópicos normalmente são associados a diferentes conjuntos de atributos.

Segundo Brusilovsky (1996-b), uma das funções mais importantes do modelo do domínio é fornecer a estrutura para representação do domínio de conhecimento do usuário (modelo do usuário). Neste contexto, Koch (2000) afirma que o conhecimento que o sistema tem a respeito do usuário pode ser representado de diferentes formas.

No que diz respeito a modelos do aluno, Marietto (2000) destaca que na prática tais modelos são construídos a partir do modelo do especialista (modelo do domínio) adicionando-se informações que os diferenciam. Esta diferença se dá basicamente pela consideração das concepções ausentes e das concepções incorretas. Uma *concepção ausente (missing conception)* é um item do conhecimento que o especialista tem e o aluno não tem. Uma *concepção incorreta (misconception)* é um item que o aluno tem e que o especialista não tem. Assim sendo, uma concepção incorreta não é necessariamente um conhecimento incorreto.

Marietto (2000) afirma que há duas abordagens para inserir diferenças no modelo do aluno. A primeira se baseia na hipótese de que o aluno conhece o conteúdo a ser apresentado, sendo então adicionadas ao modelo do aluno suas concepções ausentes e / ou incorretas. A segunda abordagem considera que o aluno não domina o assunto e quando o aluno demonstra conhecimento em um determinado tópico, o modelo do aluno é atualizado considerando este aprendido.

Algumas formas de representação do modelo do usuário são apresentadas abaixo:

- *Modelo de estereótipo*: neste modelo as propriedades e conhecimento do usuário são representados como pares item – valor. As diferentes combinações entre os pares são atribuídas a estereótipos tais como: iniciante, intermediário e avançado. Desta forma, o usuário herda todas as propriedades definidas para o estereótipo que lhe foi atribuído. Modelos de estereótipo são simples de inicializar e manter, bem como suficientes para a modelagem de interfaces ou para a escolha do tipo de ensino (por exemplo, diretiva ou exploratória) mas são insuficientes quando a adaptação requer uma descrição mais detalhada do usuário (BRUSILOVSKY, 1996-a; KOCH, 2000);
- *Modelo overlay*: neste modelo o conhecimento do usuário é considerado um subconjunto do conhecimento do sistema. Apenas as concepções ausentes do usuário são representadas (MARIETTO, 2000). Por essa razão, Abdullah (2003) sustenta que este método é incompleto, dado que através dele apenas a ausência de conhecimento pode ser modelada. Em um contexto educacional, este modelo assume que o aluno não irá aprender nada que não esteja no modelo do domínio, pois as concepções incorretas que o aluno pode ter ou adquirir durante a utilização do sistema não são representadas. O modelo *overlay* é predominantemente representado como uma hierarquia ou rede semântica de nós diretamente relacionados a conceitos do domínio (KOCH, 2000). Brusilovsky (1996-a) e Koch (2000) afirmam que bons resultados podem ser obtidos ao combinar técnicas de estereotipagem e modelagem *overlay*. Com essa combinação, inicializa-se o modelo do usuário atribuindo um estereótipo ao aluno e em passos seguintes refina-se tal modelo através da implementação da modelagem *overlay*.

- *Modelo diferencial*: de acordo com Abdullah (2003) o modelo diferencial é visto como um melhoramento do modelo *overlay*. Ele divide o conhecimento do usuário em duas categorias: conhecimento para o qual o usuário foi exposto e conhecimento para o qual o usuário não foi exposto. Este modelo ainda é incompleto, assim como o *overlay*, pois assume que o modelo do usuário é essencialmente um subconjunto do conhecimento do sistema;
- *Modelo de perturbação*: segundo Marietto (2000) o modelo de perturbação, assim como o modelo diferencial, é um aperfeiçoamento do modelo *overlay*. Este modelo representa tanto as concepções ausentes quanto as concepções incorretas do usuário. A estratégia, quando se utiliza este modelo, é criar uma biblioteca que contenha as concepções incorretas. Os membros desta biblioteca são denominados erros ou *bugs*. Assim sendo, o modelo do usuário é composto pelo modelo do domínio (também chamado modelo do especialista em domínios educacionais) mais uma lista de *bugs*. Na prática, a biblioteca de *bugs* conterá ações que o usuário pode executar, mas que estão em desacordo com o modelo do domínio. Sistemas que utilizam o modelo de perturbação geralmente trabalham com concepções incorretas de duas formas: considerando-as como *bugs* e então direcionando os usuários a repará-las; ou as considerando como um conhecimento válido, inserindo-as, portanto, no modelo do domínio. Sistemas que adotam o modelo de perturbação são limitados na estruturação da biblioteca de *bugs*. A formação de tal biblioteca não pode ser completa, pois não há como catalogar explicitamente todos os possíveis *bugs*. Uma alternativa a esta limitação é a construção dinâmica da biblioteca de *bugs* durante o processo de diagnóstico. Esta variação é chamada *modelo generativo de perturbação*. Neste modelo a lista dos possíveis *bugs* não existirá *a priori*;
- *Redes bayesianas*: segundo Marietto (2000) as redes *bayesianas* raciocinam sobre um estado mental do usuário de forma probabilística. Cada nó nestas redes tem uma probabilidade indicando a possibilidade do usuário, por exemplo, conhecer, aquela parte do domínio.

2.5.2.3 Métodos para Inicializar e Atualizar Modelos do Usuário

Segundo Koch (2000), a primeira vez que o usuário executa uma aplicação adaptativa o modelo do usuário está vazio. Um modelo do usuário pode ser inicializado de duas formas:

- *Questionamento explícito*: o sistema pode adquirir conhecimento inicial sobre o usuário solicitando que ele responda a um questionário. O problema que surge é: quantas questões o usuário estaria disposto a responder? O fato é que a entrevista inicial com o usuário constitui uma fonte de informação primária e valiosa a respeito do usuário. Em alguns casos a entrevista é utilizada para atribuir o usuário a certas classes, também denominadas estereótipos, conforme já mencionado;
- *Suposições padrão*: se o questionamento explícito não é possível ou apenas um conjunto limitado de questões é tolerável, mas informação adicional é necessária, então algumas suposições padrão podem ser feitas. Por exemplo: supor que o usuário de um sistema educacional não tem conhecimento algum do assunto que será ensinado; auxiliar o usuário com informação de ajuda sem solicitação explícita até que ele solicite a redução desse auxílio. Estereótipos de

usuário são úteis no processo de inicialização de modelos do usuário, e em alguns casos durante todo o processo de modelagem do usuário. O problema é que eles não oferecem informação suficiente no caso de modelos do aluno. Estereótipos de usuário para modelos de aluno são raramente utilizados além do estágio inicial;

Vale lembrar que na maioria dos casos a combinação de ambas as formas de inicialização do modelo do usuário é implementada em sistemas.

Segundo Koch (2000), o processo de atualização do modelo do usuário requer *aquisição* de informação sobre o comportamento do usuário e *ajuste* do modelo do usuário se há evidência de que está incorreto. O modelo do usuário está incorreto se o usuário age de forma diferente da predita por ele.

A aquisição é o processo de coletar entrada do usuário através de quaisquer formas disponíveis – clique do *mouse*, entrada teclada ou de voz, toque de tela, tempo decorrido – correspondendo a interações do usuário em um processo, páginas visitadas em uma aplicação hipermídia, passos na resolução de um problema, etc. Com base nesta informação, objetiva-se inferir o que o usuário conhece ou não conhece, o que ele prefere ou tem como objetivo. O problema reside na interpretação dos dados (clique do *mouse*, entrada teclada) em ações ou proposições, o que freqüentemente não é trivial (RAGNEMALM, 1995).

As técnicas de aquisição podem ser caracterizadas da seguinte forma (CHIN, 1993 apud KOCH, 2000):

- *Ativas ou passivas*: técnicas ativas interagem diretamente com o usuário para questioná-lo (através de um formulário em uma página *Web*, por exemplo). Por outro lado, técnicas passivas constroem modelos do usuário baseados em inferências a partir de observações (páginas visitadas pelo usuário, por exemplo);
- *Automáticas ou iniciadas pelo usuário*: em se tratando de técnicas automáticas, o usuário não tem influência sobre quando ele é observado e quando seu modelo é atualizado. No caso de técnicas iniciadas pelo usuário, o usuário decide quando modificar o modelo do usuário. Técnicas automáticas são as mais freqüentemente utilizadas;
- *Diretas ou indiretas*: as técnicas de aquisição são diretas se o sistema deriva diretamente, a partir do *feedback* do usuário, informação que é utilizada para atualizar o modelo do usuário. Técnicas de aquisição indiretas são construídas a partir dos resultados das técnicas de aquisição diretas e tomam freqüentemente a forma de regras de inferência;
- *Explícitas ou implícitas*: técnicas explícitas são aquelas em que o usuário conscientemente provê informação para o sistema. Técnicas implícitas são baseadas na observação do comportamento do usuário;
- *Lógicas ou plausíveis*: técnicas lógicas e plausíveis diferem no grau de plausibilidade dos resultados. Técnicas plausíveis requerem a representação explícita da incerteza no modelo e precisam de mecanismos para manter a consistência do modelo do usuário. Modelos *overlay* são exemplos de resultados adquiridos através de técnicas lógicas;

- *Online ou offline*: a maioria das técnicas de aquisição é aplicada *online*. Excepcionalmente, usuários podem ser observados *offline*, para inferir estereótipos, extraindo, por exemplo, informações de bases de dados de clientes.

O processo de aquisição da informação é composto por três fases que são descritas abaixo:

- *Coleta*: é o processo de coletar dados do usuário. Os principais problemas associados à coleta de dados são: a confiabilidade dos dados, a quantidade de dados disponíveis e o nível de detalhe dos dados. A quantidade de dados que é necessária depende da granularidade do modelo. A granularidade ou o nível de detalhe dos dados varia de aplicação para aplicação;
- *Diagnóstico*: é o processo de encontrar falhas. Este processo pode ser subdividido em outros dois processos: *transformação* e *avaliação*. A transformação consiste na extração de informações relevantes dos dados coletados a fim de julgar as habilidades do usuário. A avaliação se refere ao processo de comparar o comportamento ou conhecimento do usuário a alguma concepção de comportamento ou conhecimento “correto”, que é representada explícita ou implicitamente no modelo do usuário;
- *Consistência*: este processo refere-se à manutenção da consistência do modelo do usuário. Ao incorporar novas proposições ao modelo do usuário, algumas inconsistências com proposições conceituais ou comportamentais já existentes podem aparecer.

2.6 Sistemas Hipermídia Adaptativos Educacionais Baseados na Web

As subseções abaixo apresentam alguns SHAs educacionais baseados na *Web* desenvolvidos recentemente e que serviram como referencial para o desenvolvimento deste trabalho.

2.6.1 AHA

AHA (DE BRA et al., 2003; DE BRA, 2006) é um SHA de propósito geral que objetiva incorporar capacidade de adaptação a quaisquer tipos de aplicações *Web*. AHA encontra-se atualmente em sua terceira versão e pode ser obtido livremente em <http://aha.win.tue.nl/>.

Uma aplicação AHA consiste em um conjunto de conceitos, cada qual associado a um conjunto de atributos e regras do tipo evento-condição-ação. O conteúdo propriamente dito fica armazenado em páginas que são associadas aos conceitos. A criação de uma aplicação AHA envolve quatro passos principais: escrever as páginas de conteúdo, criar a estrutura conceitual, definir as regras de adaptação (regras evento-condição-ação) e definir o *look and feel* (*layout*).

As páginas devem ser escritas em XHTML (PEMBERTON et al., 2007). Uma extensão ao XHTML foi desenvolvida a fim de incluir funcionalidades nas páginas através da utilização de um conjunto especial de *tags* do AHA. Assim sendo, as páginas devem necessariamente aderir ao padrão XHTML ou XHTML + AHA.

As páginas podem conter fragmentos ou objetos condicionais. Fragmentos condicionais são inclusos através da *tag* `<if>`. O conteúdo do fragmento deve atender à sintaxe XHTML + AHA. Objetos condicionais são inclusos através da *tag* `<object>` do

XHTML. Utilizar objetos condicionais ao invés de fragmentos condicionais é vantajoso, pois não exige nenhuma extensão do XHTML e o mesmo objeto pode ser incluso em diferentes páginas sem que haja necessidade de cópia do conteúdo.

Nas páginas é possível definir *links* condicionais, incondicionais e externos. *Links* condicionais apontam para um conceito ou página e são condicionalmente exibidos ou ocultados do usuário final dependendo da estrutura conceitual. *Links* incondicionais apontam para um conceito ou página e sua apresentação depende da situação do atributo “*visited*” do conceito para o qual aponta. *Links* que não são condicionais ou incondicionais são considerados externos e são apresentados de maneira diferenciada dos anteriores.

A estrutura conceitual de uma aplicação AHA pode ser criada de três maneiras: utilizando a ferramenta de alto nível *Graph Author*; utilizando a ferramenta de mais baixo nível *Concept Editor*, que exige maior conhecimento técnico do autor; ou então editando diretamente os arquivos XML que armazenam a estrutura conceitual da aplicação.

Os principais elementos que são associados a um conceito são os seguintes:

- Nome: o conceito deve ter um nome único na aplicação;
- Recurso: o conceito pode ser abstrato ou associado a uma página. Para o último, uma URL parcial (endereço da página) precisa ser especificada;
- Hierarquia: um conceito faz parte de uma hierarquia, cuja raiz é o conceito de mais alto nível da aplicação. Na hierarquia são armazenados os nomes dos nós pai, filho e irmão, se existirem;
- Atributos: cada atributo possui propriedades e um conjunto de regras evento-condição-ação. Algumas das propriedades dos atributos são: nome, tipo (booleano, inteiro ou *string*) e descrição. Exemplos de atributos são: acesso (a este atributo é atribuído o valor verdadeiro quando uma página é acessada), adequabilidade (*suitability*), conhecimento e interesse;
- Regras evento-condição-ação: a cada atributo de cada conceito pode ser associado um número irrestrito de regras evento-condição-ação que definem como o acesso a uma página resulta em atualizações no modelo do usuário. As regras são disparadas por um evento que é tipicamente o acesso a uma página.

AHA mantém um modelo do usuário para cada indivíduo. Tal modelo é composto por conceitos com atributos. As atualizações no modelo do usuário são feitas através das regras evento-condição-ação.

2.6.2 WHURLE

WHURLE (BRAILSFORD et al., 2002; ZAKARIA et al., 2003; MOORE et al., 2004) é um ambiente adaptativo de ensino voltado para a *Web* e pode ser obtido livremente em <http://whurle.sourceforge.net/>.

WHURLE armazena o material instrucional em construções atômicas denominadas *chunks*. Um *chunk* é um item conceitualmente autocontido que é especificado em um arquivo XML em WCML. A um *chunk* tipicamente são associados metadados, como por exemplo, versões (autor e data) e palavras-chave. O conjunto de todos os *chunks* em uma instância de WHURLE é denominado *melange*.

O mecanismo de armazenamento dos *chunks* é transparente para o usuário final. O que este último enxerga é uma lição. Uma lição referencia um número qualquer de *chunks*, em conjunto com *links* navegacionais. A lição é definida em um arquivo XML em WLPML e é denominada plano de lição. O plano de lição é criado por professores e especifica um caminho a ser percorrido no *melange*. Em sua forma conceitual mais simples, um plano de lição consiste em níveis hierárquicos, cada qual contendo uma ou mais páginas, que por sua vez referenciam os *chunks*. Quando a lição é adaptativa, o plano de lição se torna mais complexo, pois a inclusão de um *chunk*, página ou mesmo um nível hierárquico inteiro de uma lição é condicional.

A inclusão condicional de *chunks*, páginas ou níveis hierárquicos é especificada no plano de lição através de dependências. Dependências são definidas através da *tag* <dependancies>. Dependências podem envolver aspectos como experiência anterior, tempo (um determinado material pode estar disponível por um período limitado) ou nível de conhecimento do aluno em um dado domínio.

WHURLE contém três sistemas distintos de *links*: *intra-chunk links*; *links* sistêmicos; e *links* autorados. Encontram-se descritos abaixo cada um deles:

- *Intra-chunk links*: são *links* dentro de *chunks* criados pelo autor do mesmo. São ativados pela passagem do *mouse* sobre o *link* e exibidos como janelas *pop-up*. Este tipo de *link* é útil quando um pequeno pedaço de informação é conceitualmente parte do *chunk*, mas por razões pedagógicas ou de projeto de interface com o usuário é necessário removê-lo da visualização principal;
- *Links* sistêmicos: são automaticamente gerados pelo sistema em tempo de execução. A geração é feita com base no plano de lição e com base em um documento XML que define a IU ou *skin*. A partir do plano de lição é possível gerar *links* para nós ancestrais, irmãos e filhos;
- *Links* autorados: são manualmente criados por professores ou alunos e definidos em arquivos XML denominados *linkbases*.

O modelo do usuário que foi adaptado para o WHURLE consiste na fusão de duas técnicas amplamente utilizadas em sistemas hipermídia adaptativos: o modelo *overlay* e o modelo estereótipo.

2.6.3 AdaptWeb

AdaptWeb (BRUNETTO et al., 2003; OLIVEIRA et al., 2003) é um ambiente adaptativo de educação a distância via *Web* que tem como finalidade adaptar o conteúdo, a apresentação e a navegação de acordo com o perfil do aluno. O desenvolvimento do AdaptWeb foi motivado pela necessidade de facilitar a tarefa de autoria de material instrucional para a *Web* por parte dos professores. AdaptWeb pode ser obtido livremente em <http://adaptweb.sourceforge.net/>.

AdaptWeb é composto por um ambiente de autoria e por um ambiente de *run-time*. O ambiente de autoria auxilia o autor a desenvolver múltiplas apresentações dos conteúdos para um curso, com alternativas para diferentes programas e estilos de aprendizagem. O processo de criação do curso consiste em organizar os conteúdos educacionais através de uma estrutura hierárquica de conceitos. O ambiente de *run-time* provê a seqüência de apresentação dos conteúdos do curso de acordo com as características do aluno.

A etapa de autoria do material instrucional a ser disponibilizado envolve:

- Organizar o conteúdo programático estruturando-o através de conceitos;
- Associar a cada conceito um arquivo HTML (com o conteúdo propriamente dito do conceito), descrição, abreviatura, palavras-chave, lista de pré-requisitos e cursos (por exemplo, Ciência da Computação ou Matemática) para os quais o conceito deve ser disponibilizado. Exemplos, exercícios e material complementar também podem ser associados a um conceito. Para isso é necessário especificar um arquivo de conteúdo, descrição do conteúdo e os cursos para os quais o conteúdo deve ser exibido. Em se tratando de exemplos e exercícios, o nível de complexidade pode ser informado.

Finalizada a etapa de autoria, os dados referentes à estrutura do conteúdo se encontram em uma estrutura de dados matricial em memória. Tal matriz é utilizada como entrada para o processo que gera os arquivos XML que armazenarão a estrutura autorada. O armazenamento em XML é organizado de forma hierárquica. São criados um arquivo XML com a estrutura de conceitos e tantos arquivos XML quantos forem os conceitos definidos.

Antes de serem apresentados ao aluno, os documentos XML resultantes da etapa de autoria devem passar por um processo de adaptação. A adaptação ocorre em dois níveis: adaptação do conteúdo e da navegação. Quatro características do aluno são consideradas relevantes no processo de adaptação: o curso, o conhecimento já adquirido, as preferências e o ambiente de trabalho.

Para entrar no ambiente de *run-time* do AdaptWeb o aluno precisa estar devidamente cadastrado e fornecer um *e-mail* e senha de acesso. Efetuado o *login*, o sistema apresenta as disciplinas nas quais o aluno está cadastrado. Depois de selecionada uma disciplina e curso, o aluno deve informar o tipo de conexão de rede que tem disponível (ADSL ou *dial-up*) e sua preferência com relação ao modo de navegação (Tutorial ou Livre).

O curso selecionado faz com que o sistema apresente os exemplos, exercícios, materiais complementares e conceitos mais adequados para o aluno. Já o tipo de conexão de rede determina os formatos de mídia nos quais os conteúdos devem ser apresentados.

No modo tutorial, os pré-requisitos entre os conceitos determinam a navegação do aluno. A adaptação da navegação se baseia no registro dos componentes visitados. Conceitos cujos pré-requisitos foram visitados são liberados para acesso. Os *links* para os conceitos visitados, não visitados e o conceito atual são apresentados em cores diferentes. *Links* desabilitados indicam que o aluno ainda não tem o conhecimento necessário para visitá-los.

No modo livre, o aluno pode navegar livremente, acessando qualquer conceito disponível no menu de navegação.

2.6.4 iWeaver

iWeaver (WOLF, 2002; WOLF, 2003) é um ambiente adaptativo e interativo de aprendizagem baseado na *Web*. Baseia-se na hipótese de que ao oferecer ao aluno a oportunidade de aprender com material instrucional alinhado às suas preferências, um significativo melhoramento em seu desempenho, compreensão e motivação é esperado

quando comparado a um ambiente estático. A fim de obter tal melhoramento, iWeaver permite ao aluno modificar suas preferências.

iWeaver oferece ao aluno diferentes representações do mesmo material instrucional e também recomendações, ao invés de “encaixar” o aluno em um estilo prefixado ou modelo de aluno. Em sua versão atual o protótipo desenvolvido se destina ao ensino da linguagem de programação Java.

iWeaver se baseia no estilo de aprendizagem do aluno para adaptar o conteúdo, seja variando as diferentes representações de mídia ou usando texto condicional. As técnicas de ordenamento e ocultamento de *links* são implementadas a fim de suportar a navegação adaptativa.

Ao acessar pela primeira vez o ambiente, o aluno deve responder a um questionário de 118 questões de múltipla escolha. O resultado desse questionário determina o modelo do aluno inicial com o qual o iWeaver irá trabalhar. Após o questionário, são fornecidas ao aluno uma explicação de seu estilo de aprendizagem avaliado e recomendações de representação de mídia para o primeiro módulo de conteúdo. O aluno tem a opção de escolher uma representação de mídia diferente da que foi recomendada para seu estilo de aprendizagem.

O conteúdo instrucional é dividido em módulos que por sua vez são divididos em unidades instrucionais. Ao final de uma unidade instrucional, o aluno pode visitar a unidade utilizando uma representação de mídia diferente. Alternativamente, o aluno pode prosseguir para a próxima unidade. Entretanto, antes de prosseguir, o aluno deve classificar as representações de mídia experimentadas e fornecer sua impressão a respeito de seu progresso em termos de aprendizagem e nível de satisfação geral com o ambiente de aprendizagem.

Encontram-se descritos abaixo os quatro tipos de representação de mídia oferecidos pelo iWeaver:

- *Visual text*: é a representação de mídia mais simples. É adequada a alunos que memorizam melhor o conteúdo através da leitura;
- *Visual pictures*: é adequada a alunos que têm preferência por visualizar o conteúdo de maneira ilustrada. Nesta representação os textos são enriquecidos com ilustrações, diagramas, fluxogramas e animações não interativas;
- *Tactile kinaesthetic*: é adequada a alunos que têm preferência por interagir fisicamente com o que estão aprendendo. Animações interativas podem ser utilizadas neste tipo de representação;
- *Auditory*: é adequada a alunos que têm preferência por ouvir o conteúdo instrucional.

2.7 Considerações Finais

Ao longo deste capítulo, procurou-se traçar um panorama bastante abrangente dos principais aspectos que envolvem a HA. Tais aspectos englobam: as áreas de aplicação da HA; os aspectos do usuário que podem ser levados em consideração ao adaptar; os aspectos visíveis do sistema que podem ser adaptados (foram apresentadas as tecnologias da hipermídia adaptativa, tanto para a apresentação adaptativa quanto para o suporte à navegação adaptativa); os métodos da HA; e as principais questões

relacionadas à modelagem do usuário. Por fim, foram apresentados alguns SHAs educacionais baseados na *Web* (o AHA é um SHA de propósito geral, e portanto pode ser utilizado com fins educacionais) desenvolvidos recentemente e que serviram como referencial para o desenvolvimento deste trabalho.

Nos próximos capítulos, são mostrados os aspectos que envolvem a HA, porém aplicados no desenvolvimento de um SHA educacional baseado na *Web*, objeto deste trabalho, denominado *AdaptHA*: Ambiente para Autoria e Ensino Adaptativo. O modelo do usuário, a partir daqui, é denominado modelo do aluno. Como o próprio título deixa claro, o *AdaptHA*, além de se focar no ensino adaptativo, inclui ferramentas para a autoria de cursos adaptativos, e segundo Brusilovsky (2001), o surgimento de ferramentas de autoria indica a maturidade da hipermídia educacional adaptativa e uma resposta à demanda impulsionada pela *Web* por cursos adaptativos de educação a distância. Os próximos capítulos examinam em detalhes todas as questões envolvidas no desenvolvimento do *AdaptHA*.

3 O MODELO PROPOSTO

Este capítulo apresenta o *AdaptHA*: Ambiente para Autoria e Ensino Adaptativo. Nas seções seguintes este trabalho é contextualizado dentro da pesquisa em hipermídia educacional adaptativa, são apresentados os fundamentos teóricos nos quais este trabalho se apóia e é descrita em detalhes a arquitetura do *AdaptHA*: na seção 3.1 são apresentadas as gerações de pesquisa em hipermídia educacional adaptativa com o intuito de contextualizar este trabalho dentro dessa área de pesquisa; na seção 3.2 é traçado um quadro comparativo entre o *AdaptHA* e trabalhos relacionados; na seção 3.3 é apresentado o Projeto Instrucional, que é a teoria de aprendizagem na qual o *AdaptHA* se inspira; a seção 3.4 descreve o Hyper-Automaton, que fornece o modelo estrutural para a organização e gerenciamento dos cursos e material instrucional no *AdaptHA*; a seção 3.5 descreve em detalhes a arquitetura do *AdaptHA*; a seção 3.6 apresenta as considerações finais deste capítulo.

3.1 Contexto do Trabalho

Segundo Brusilovsky (2004), a hipermídia educacional foi uma das primeiras áreas de aplicação da HA e constitui atualmente uma das áreas mais populares e bem investigadas. É natural que a hipermídia educacional tenha sido uma das primeiras áreas de aplicação da HA. No contexto educacional, usuários com diferentes objetivos de aprendizagem e diferentes níveis de conhecimento requerem essencialmente diferentes tratamentos. É também na hipermídia educacional que o problema do “sentir-se perdido no hiperespaço” é especialmente crítico.

Brusilovsky (2004) apresenta uma revisão histórica e sumariza o estado da arte da pesquisa em *hipermídia educacional adaptativa*, que é o foco de pesquisa do trabalho aqui apresentado. Nessa revisão histórica, a pesquisa em hipermídia educacional adaptativa é dividida em três gerações, que são descritas sucintamente nas próximas subseções.

3.1.1 Primeira Geração de Pesquisa em Hipermídia Educacional Adaptativa

A primeira geração de pesquisa em hipermídia educacional adaptativa pode ser caracterizada entre os anos de 1990 e 1996. No início dos anos 90, vários grupos de pesquisa já haviam reconhecido os problemas do hipertexto estático em diferentes áreas de aplicação e já haviam começado a explorar diversas maneiras de adaptar o comportamento de sistemas hipertexto e hipermídia a usuários individuais. Alguns grupos de pesquisa estavam trabalhando com os problemas relacionados à *navegação* em hipermídia (problemas como a navegação ineficiente e a questão do usuário de se sentir perdido no hiperespaço) que foram descobertos quando a área de hipertexto alcançou sua relativa maturidade no final dos anos 80. Outros grupos estavam

trabalhando com os problemas relacionados à apresentação de conteúdo (como a necessidade de apresentar diferentes informações em uma página hipermídia a diferentes usuários). Dessas pesquisas resultaram os primeiros SHAs, bem como tecnologias para o *suporte à navegação adaptativa e apresentação adaptativa*.

3.1.2 Segunda Geração de Pesquisa em Hipermídia Educacional Adaptativa

A segunda geração de pesquisa em hipermídia educacional adaptativa pode ser caracterizada entre os anos de 1996 e 2002. Somente a partir de 1996 a pesquisa nessa área começou a atrair a atenção de uma grande comunidade de pesquisadores. Brusilovsky (2004) aponta dois principais fatores que contribuíram para o crescimento dessa atividade de pesquisa. O primeiro diz respeito à acumulação e consolidação de experiência de pesquisa na área. O segundo diz respeito ao rápido aumento do uso da *World Wide Web*. A *Web*, com sua clara demanda por adaptatividade, serviu para impulsionar a pesquisa em HA, fornecendo uma plataforma tanto desafiante quanto atrativa. A grande força motriz por trás da hipermídia educacional adaptativa de segunda geração foi a educação baseada na *Web*. A necessidade dos cursos baseados na *Web* atenderem individualmente um público heterogêneo estava clara para muitos pesquisadores. A maioria dos SHAs educacionais desenvolvidos desde 1996 são sistemas baseados na *Web*. A escolha pela *Web* como plataforma de desenvolvimento se tornou um padrão (BRUSILOVSKY, 2001).

O trabalho na hipermídia educacional adaptativa de segunda geração pode ser dividido em três segmentos com fronteiras não tão nítidas. O maior segmento de trabalho se focou na criação de SHAs educacionais baseados na *Web*. A principal motivação era produzir sistemas que pudessem ser utilizados no processo de ensino, não o desenvolvimento de novas tecnologias. Um pequeno segmento de trabalho se focou na produção de novas tecnologias para a HA. Por fim, outro segmento de trabalho, originalmente pequeno, mas que expandiu rapidamente, focou-se no desenvolvimento de *frameworks* e ferramentas de autoria para produzir SHAs. A maior parte do trabalho neste segmento conseguiu produzir *frameworks* para a educação adaptativa baseada na *Web*. Um *framework* tipicamente introduz uma arquitetura e abordagem genérica reutilizável que podem ser utilizadas para produzir uma série de sistemas adaptativos. Poucos dos grupos de pesquisa mais experientes, que estavam trabalhando com projetos de HA por vários anos, introduziram sistemas de autoria úteis e práticos que pudessem ser aplicados para uso do usuário final no desenvolvimento de SHAs e cursos.

3.1.3 Terceira Geração de Pesquisa em Hipermídia Educacional Adaptativa

SHAs educacionais de segunda geração falharam ao tentar influenciar a educação prática baseada na *Web*. Poucos desses sistemas foram usados para ensino em cursos reais. Pelo contrário, a maioria absoluta de cursos na *Web* conta com os chamados LMSs (BLACKBOARD, 2007; TOPCLASS, 2007). LMSs são poderosos sistemas integrados que suportam uma série de necessidades, tanto de professores quanto de alunos. Professores podem utilizar um LMS para desenvolver cursos e testes baseados na *Web*, comunicar-se com alunos e monitorar seu progresso. Alunos podem utilizá-lo para comunicação e colaboração. O completo domínio de LMSs sobre sistemas adaptativos pode parecer surpreendente. De fato, para cada função que um LMS típico desempenha, pode-se encontrar um SHA educacional baseado na *Web* que pode desempenhá-la muito melhor (BRUSILOVSKY, 2004). Parece óbvio que o problema com sistemas adaptativos modernos não é seu desempenho, mas sua inabilidade de

atender às necessidades da educação prática pela *Web*. O desafio de integrar tecnologias da HA em processos educacionais regulares definiu a *terceira geração* (atual) de pesquisa em hipermídia educacional adaptativa.

Diferentes grupos de pesquisa enfatizam diferentes razões para explicar o domínio dos LMSs sobre os SHAs educacionais e seguem por diferentes segmentos de pesquisa. Um dos segmentos se focou na versatilidade dos LMSs e tenta fornecer, em um único sistema, funcionalidades (como as que são fornecidas por um LMS moderno) tanto para o professor quanto para o aluno, mais a capacidade de se adaptar ao usuário. Um diferente segmento de pesquisa se preocupou com outra característica de LMSs: a capacidade de integrar conteúdo *Web open corpus* (BRUSILOVSKY, 2005). Os sistemas deste segmento exploraram diversas abordagens para integrar conteúdo *open corpus* em um SHA e ao mesmo tempo fornecer direcionamento adaptativo para esse conteúdo. Projetos mais recentes, entretanto, optaram por não competir com os LMSs atuais, mas sim focalizar em características adaptativas de uma próxima geração de sistemas educacionais baseados na *Web*. Essa nova geração, que está vindo para substituir os modernos LMSs, basear-se-á na interoperabilidade e reusabilidade de conteúdo que será suportado por padrões emergentes, incluindo o mais frequentemente citado padrão SCORM (ADL, 2007). Uma série de grupos de pesquisa está tentando integrar tecnologias adaptativas existentes com idéias de reusabilidade baseada em padrões, embora outros grupos argumentem que a geração atual de padrões não esteja preparada para suportar as idéias de aprendizagem adaptativa. Outros grupos estão focados em questões relacionadas às anteriores, mas que não são cobertas por padrões ainda, tais como arquiteturas de descoberta de recursos. Outro segmento de trabalho tenta explorar as idéias da *Web* semântica (SEMANTIC WEB, 2007) para representação de conteúdo e descoberta de recursos e utilizam padrões como RDF (RDF, 2007) e *Topic Maps* (TOPIC MAPS, 2007).

É justamente nesse contexto de terceira geração de pesquisa em hipermídia educacional adaptativa que está sendo desenvolvido o *AdaptHA*. O *AdaptHA* pode ser claramente considerado um SHA educacional de terceira geração pois:

- Foi projetado para promover a educação prática baseada na *Web*, podendo ser utilizado em processos regulares de ensino;
- Utiliza técnicas da HA tanto para a adaptação de conteúdo quanto para o suporte à navegação adaptativa.

Considerando-se os segmentos de pesquisa dentro da terceira geração, o *AdaptHA* se insere no segmento que compete diretamente com os LMSs, pois tem como objetivos:

- Proporcionar ao aluno uma experiência de aprendizagem individualizada, utilizando para isso as técnicas da HA;
- Fornecer para o professor ferramentas – de autoria e de acompanhamento do desempenho dos alunos – que venham a colaborar com a melhoria do seu trabalho.

3.2 Trabalhos Relacionados

Na seção 2.6 do capítulo 2 foram apresentados alguns SHAs educacionais baseados na *Web*, os quais foram utilizados como referencial para o desenvolvimento deste trabalho. São eles: AHA (DE BRA et al., 2003; DE BRA, 2006), WHURLE

(BRAILSFORD et al., 2002; ZAKARIA et al., 2003; MOORE et al., 2004), AdaptWeb (BRUNETTO et al., 2003; OLIVEIRA et al., 2003) e iWeaver (WOLF, 2002; WOLF, 2003).

A tabela 3.1, exibida abaixo, efetua uma comparação entre os trabalhos anteriormente citados e o *AdaptHA*. Para efetuar tal comparação foram considerados os seguintes critérios:

- *Independência de domínio*, ou seja, a capacidade do sistema de poder ser utilizado para o ensino de conteúdos de quaisquer domínios, por exemplo, matemática, história, português, etc., sem que seja necessário implementar alterações no mesmo;
- *Autoria de cursos*, ou seja, o sistema possibilitar a autoria de cursos, através de uma interface de usuário amigável, sem que seja requerido o conhecimento de pessoal técnico;
- *Avaliações*, ou seja, o sistema possibilitar a realização de avaliações por parte do aluno;
- *Localização e internacionalização*:
 - *localização*, ou seja, o sistema ter a capacidade de se adaptar a uma dada localidade, adaptando o idioma, formatos numéricos, formatos de data, etc. (este conceito é explicado com maior detalhamento na subseção 3.5.1);
 - *internacionalização*, ou seja, o sistema ter a capacidade de fácil localização, sem que haja a necessidade de implementar alterações no mesmo (este conceito é explicado com maior detalhamento na subseção 3.5.1).
- *Metadados associados ao material instrucional*, ou seja, o sistema permitir a associação de metadados ao material instrucional de forma que seja possível recuperar de forma mais eficiente o material instrucional desejado;
- *Material instrucional independente dos cursos*, ou seja, o sistema manter o material instrucional de forma independente dos cursos, de maneira que se possa reutilizar esse material em diferentes contextos e cursos;
- *Autoria de textos*, ou seja, o sistema fornecer um mecanismo que possibilite a autoria de textos, através de uma interface de usuário amigável, por meio de um editor de textos;
- *Autoria de questões de avaliações*, ou seja, o sistema possibilitar a autoria de questões de avaliações, através de uma interface de usuário amigável, sem que seja requerido o conhecimento de pessoal técnico;
- *Autoria de avaliações*, ou seja, o sistema possibilitar a autoria de avaliações, através de uma interface de usuário amigável, sem que seja requerido o conhecimento de pessoal técnico;
- *Livre estruturação hierárquica do modelo do domínio*, ou seja, o sistema possibilitar que o modelo do domínio seja livremente estruturado, de forma que seja possível construir uma estrutura hierárquica com níveis de aninhamento quaisquer entre os conteúdos do curso.

Tabela 3.1: Comparação Entre o *AdaptHA* e Outros Trabalhos Relacionados

Crítérios	<i>AdaptHA</i>	AHA	WHURLE	AdaptWeb	iWeaver
Independência de domínio	X	X	X	X	
Autoria de cursos	X	X		X	
Avaliações	X	X			
Localização / internacionalização	X		X	X	
Metadados associados ao material instrucional	X		X		
Material instrucional independente dos cursos	X	X			
Autoria de textos	X				
Autoria de questões de avaliações	X				
Autoria avaliações	X				
Livre estruturação hierárquica do modelo do domínio		X	X	X	

Conforme mostrado na tabela 3.1 acima, uma das grandes contribuições do *AdaptHA* reside na autoria. O sistema construído, que é detalhado no capítulo 4, possui ferramentas para a autoria de cursos, textos, avaliações e questões de avaliações. Para facilitar a autoria de textos, foi embutido no sistema um poderoso editor de textos, o FCKeditor (FCKEDITOR, 2007), que possui recursos para a edição de textos, com a inclusão de imagens, tabelas, *links* e etc.

No entanto, apesar de todas as facilidades de autoria que o *AdaptHA* oferece, há uma restrição quanto à estruturação do modelo do domínio, que será detalhado na subseção 3.5.2, pois não é possível construir uma estrutura hierárquica com níveis de aninhamento quaisquer entre os conteúdos (conceitos) do curso.

3.3 Projeto Instrucional

Para Gagné (1980), a aprendizagem ocorre como resultado da interação entre o aluno e seu ambiente. Sabe-se que a aprendizagem ocorreu quando se observa que houve modificação no desempenho do aluno. A instrução objetiva promover a aprendizagem (GAGNÉ, 1977) e pode ser definida como o *conjunto de eventos* planejados para iniciar, ativar e manter a aprendizagem no aluno. Tais eventos precisam ser primeiramente planejados, e, em segundo lugar, entregues, levados a produzir seus efeitos no aluno. A estruturação e seqüenciação desse conjunto de eventos, denominados *eventos de instrução*, são chamadas por Gagné et al. (1992 apud MARIETTO, 2000) de *projeto instrucional*.

Segundo Gagné (1977) a ocorrência da aprendizagem é inferida a partir da diferença entre a performance do ser humano antes e depois de ser colocado em uma situação de aprendizagem. Um conjunto de fatores que contribui para a aprendizagem já existe

antes que qualquer nova aprendizagem inicie. Habilidades previamente aprendidas constituem as *condições internas* necessárias para a aprendizagem. As *condições externas* correspondem a todas as ações intencionalmente realizadas que objetivam favorecer o processo de aprendizagem. Gagné (1977) denomina as condições internas e externas de *condições de aprendizagem*.

Segundo Marietto (2000), ambas as condições desempenham fator preponderante no desenvolvimento de projetos instrucionais, pois ajudam a definir como se dá o processo de aprendizagem (condições internas) e o processo de ensino (condições externas). A união delas resulta no processo de instrução. Por esta razão, pode-se dizer que os objetivos da instrução estão voltados a determinar uma estratégia instrucional para um objetivo instrucional em particular. Assim, faz-se necessária uma consonância entre estratégias e objetivos para que a aprendizagem ocorra de forma satisfatória.

3.3.1 Categorias de Capacidades Humanas

Gagné et al. (1992 apud MARIETTO, 2000) argumentam que a melhor maneira de planejar a instrução é trabalhar a partir de comportamentos que se espera que os alunos apresentem. Assim, a aprendizagem ocorre quando o aluno adquire a capacidade de fazer alguma coisa. A razão básica para o planejamento e desenvolvimento da instrução é tornar possível a realização de um conjunto de objetivos instrucionais. Objetivos instrucionais são as atividades humanas que contribuem para o funcionamento da sociedade e que podem ser adquiridas através de um processo de aprendizagem.

Gagné et al. (1992 apud MARIETTO, 2000) propõem que o planejamento instrucional pode ser simplificado pela classificação dos objetivos educacionais em cinco categorias de capacidades: informação verbal ou conhecimento declarativo, habilidades intelectuais ou conhecimento procedural, estratégias cognitivas, habilidades motoras e atitudes. Para cada uma destas categorias é possível analisar o desempenho por parte do aluno, bem como definir um conjunto de condições externas e internas que afetam a aprendizagem. A tabela 3.2 mostra exemplos de ações relacionadas às categorias de capacidades humanas.

De acordo com Marietto (2000), as cinco categorias de capacidades humanas são importantes para uma educação completa do ser humano. Entretanto, quando do desenvolvimento de um sistema educacional não é necessário considerá-las todas simultaneamente. O atual conhecimento na área de ciência cognitiva não permite trabalhar de forma adequada, em termos computacionais, com as atitudes e habilidades motoras dos alunos. Mas é possível modelar um sistema educacional levando em consideração o conhecimento declarativo, o conhecimento procedural e as estratégias cognitivas. Neste trabalho são consideradas duas destas categorias de capacidades humanas: o conhecimento declarativo e o conhecimento procedural. Ambos constituem possíveis objetivos instrucionais a serem atingidos.

Cada uma das categorias de capacidades humanas é descrita abaixo. As duas primeiras categorias são descritas em maiores detalhes, as demais são descritas de forma sucinta.

Tabela 3.2: Ações Relacionadas às Categorias de Capacidades Humanas

Capacidade	Ações
Informação Verbal	Contar um acontecimento passado
	Saber o nome dos dias do mês e da semana, em um determinado idioma
Habilidade Intelectual	Escolher uma fórmula matemática a ser aplicada em um problema
	Classificar uma planta
Estratégia Cognitiva	Deduzir uma lei física a partir de experimentos
	Raciocinar por analogias
Habilidade Motora	Digitar um texto
	Dirigir um carro
Atitude	Escolher assistir a um filme de ficção científica
	Escolher jogar tênis como uma atividade física regular

Fonte: MARIETTO, 2000. p. 55.

3.3.1.1 Informação Verbal

As informações verbais são os fatos e o conhecimento organizado do mundo. Corresponde ao saber *que* ou ao *conhecimento declarativo* (GAGNÉ et al., 1992 apud MARIETTO, 2000). Segundo Gagné (1980), a aprendizagem de informação verbal como uma capacidade, indica que o indivíduo é capaz de enunciar, em forma proposicional, o que ele aprendeu. No imaginário educacional, esse tipo de conhecimento está relacionado ao ato de decorar. Mas isto não corresponde à verdade. Por exemplo, uma pessoa apresenta conhecimento declarativo quando consegue contar a um amigo um filme que já assistiu.

Gagné et al. (1992 apud MARIETTO, 2000), consideram dois tipos de conhecimentos declarativos:

- *Fatos*: um fato é uma informação que expressa uma relação entre um ou mais objetos ou eventos previamente rotulados. Por exemplo, “O Brasil é um país da América Latina” é um fato que representa uma relação “faz-parte-de”. Os fatos podem ser analisados de forma isolada, ou fazendo parte de uma rede interconectada de conhecimentos. Em qualquer um dos casos ficam armazenados na memória e são recuperados quando necessários. O desempenho de um aluno, indicando o aprendizado de um fato, é a exposição das relações entre os rótulos relacionados ao mesmo. Por exemplo, o aluno deve estar apto a responder qual é a maior ilha do Brasil, ou qual o maior estado em extensão da federação. Para a aquisição deste tipo de conhecimento, a condição interna necessária é a existência de uma rede organizada de conhecimento declarativo, sendo que o fato recentemente estudado deve ser relacionado a esta rede. Com relação às condições externas, a rede de conhecimentos com a qual o novo fato estará relacionado deve ser lembrada. Isto pode ser feito através da apresentação de imagens, textos, dicas, etc. O importante é que a comunicação sugira a relação a ser estabelecida. Outra condição externa se refere a

reapresentar o fato no transcorrer do processo de aprendizagem objetivando estabelecer a relação por parte do aluno;

- *Conhecimento organizado*: o estabelecimento de redes proposicionais de fatos e a posterior interconexão entre as mesmas, de forma significativa, fazem com que o conhecimento fique organizado. Tal organização permite recuperar o conhecimento de forma mais eficiente. Por exemplo, o conhecimento de Geografia e de História do Brasil pode auxiliar no entendimento da atual situação político-sócio-econômica do país. Geralmente, esta organização é estabelecida dinamicamente durante o processo de aprendizagem. Espera-se, com tal organização, que o estudante aprenda e possa “relembrar” os principais temas ou generalizações de um material apresentado. O desempenho relacionado ao aprendizado de um conhecimento organizado é a exposição do significado do mesmo, e não necessariamente a exposição de detalhes a ele relacionados. As condições internas são as mesmas relacionadas ao aprendizado de fatos. As condições externas mais adequadas à aprendizagem de redes organizadas de conhecimento declarativo envolvem a apresentação de “pistas”. Estas pistas permitem ao estudante buscar a informação e recuperá-la. Tais pistas podem ser palavras, frases, imagens, sons, etc. Outra condição externa útil é a adoção de uma estratégia de atenção, utilizando, por exemplo, perguntas do tipo “com o que se parece?” e “o que lembra?”.

3.3.1.2 *Habilidade Intelectual*

Segundo Marietto (2000), quando se aborda habilidades intelectuais, geralmente procedimentos são descritos. Por isto, a expressão *conhecimento procedural* é utilizada para se referir às mesmas. Habilidades intelectuais permitem ao aluno interagir com o ambiente através de símbolos. As habilidades intelectuais constituem o “saber como” em comparação com o “saber o quê” da informação (GAGNÉ, 1980), assim sendo, aprender uma habilidade intelectual significa saber *como fazer* alguma coisa de forma intelectual. Este tipo de conhecimento contrasta com o conhecimento de que alguma coisa existe ou tem certas propriedades, denominado anteriormente informação verbal. Por exemplo, aprender a determinar a partitura de uma música a ouvindo é uma habilidade intelectual. Por outro lado, apenas conhecer a letra da música é uma informação verbal.

Segundo Marietto (2000), é possível, e até mesmo desejável, analisar as habilidades intelectuais de acordo com suas complexidades. Gagné et al. (1992 apud MARIETTO, 2000) apresentam uma classificação de habilidades intelectuais baseada em níveis de complexidade. Esta classificação está representada na figura 3.1. Nesta figura são considerados cinco níveis de habilidades intelectuais, apresentados em ordem decrescente de complexidade: resolução de problemas e regras complexas, regras simples, conceitos definidos, conceitos concretos e discriminação.

Ao contrário do conhecimento declarativo, que precisa principalmente de uma rede organizada de conhecimento para ser contextualizado, o aprendizado de habilidades intelectuais é influenciado pela recuperação de outras habilidades intelectuais consideradas pré-requisitos (GAGNÉ, 1980), conforme a representação hierárquica apresentada na figura 3.1. Abaixo são descritos os cinco níveis de habilidades intelectuais, do mais simples para o mais complexo:

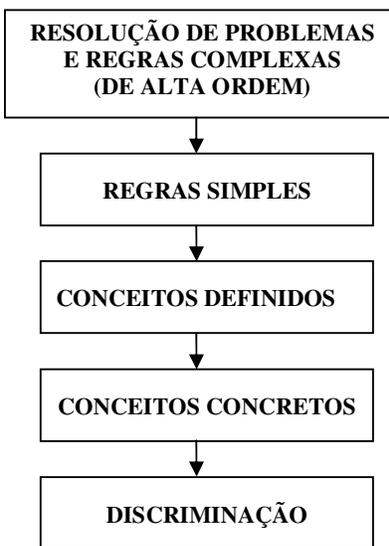


Figura 3.1: Habilidades Intelectuais (GAGNÉ et al., 1992 apud MARIETTO, 2000)

- Discriminação:* é a capacidade que permite ao indivíduo distinguir características distintas de estímulos, incluindo símbolos (GAGNÉ, 1980; GAGNÉ e MEDSKER, 1996 apud MARIETTO, 2000). Quando um aluno usa discriminação, ele consegue distinguir estímulos através de suas características. Na escala de complexidade adotada, a discriminação é considerada a habilidade intelectual mais simples. Com relação ao desempenho, deve haver uma resposta que indique que o aluno consegue distinguir estímulos que difiram ou se assemelhem em uma ou mais dimensões físicas. As condições internas são elementares, basicamente envolvendo raciocínio lógico. As condições externas estão relacionadas a alguns princípios da aprendizagem. Entre eles está o princípio da contigüidade, para o qual uma situação de estímulo deve ser apresentada simultaneamente com a resposta desejada. Outro princípio importante é o da repetição, para o qual a situação de estímulo e sua resposta devem ser repetidas e praticadas;
- Conceitos concretos:* de acordo com Gagné e Medsker (1996 apud MARIETTO, 2000), um conceito é a capacidade de classificar fenômenos através do uso de atributos críticos. Assim, com o uso de conceitos é possível colocar estímulos em classes e raciocinar com base nas idéias gerais da classe. Um conceito concreto identifica uma propriedade de um objeto. Tais conceitos são chamados concretos porque estão relacionados a objetos concretos, em oposição aos abstratos. A diferença que existe entre a discriminação e os conceitos concretos é a seguinte: o primeiro está relacionado a responder a uma diferença ou a uma semelhança, e o segundo está relacionado a classificar algo baseado em uma característica. Saber discriminar não é a mesma coisa que “ter o conceito de”. Entretanto, não é possível aprender um conceito (classificar estímulos) sem ter a capacidade da discriminação (distinguir atributos de estímulos). Alguns exemplos de conceitos concretos são: inflação, taxa de mercado, televisão, trânsito, etc. Com relação ao desempenho, um aluno identifica uma classe de propriedades de objetos (incluindo posição, forma, etc.) “apontando” dois ou mais membros desta classe. Para adquirir um conceito concreto, a condição interna necessária é o domínio da capacidade de discriminação. Como exemplo

de condição externa, é possível apresentar instâncias de um conceito, variando amplamente suas características, e então solicitar ao aluno que separe do grupo a instância que corresponde às características solicitadas. A apresentação de definições também auxilia no aprendizado de conceitos. Segundo Marietto (2000), é aconselhável que exemplos e definições sejam usados em conjunto;

- *Conceitos definidos*: diz-se que um indivíduo aprendeu um conceito definido quando ele pode “demonstrar” o significado de alguma classe particular de objetos, eventos ou relações (GAGNÉ et al., 1992 apud MARIETTO, 2000). Esta demonstração corresponde ao estabelecimento de uma seqüência de argumentos que leve à compreensão de tal conceito. Conceitos concretos auxiliam na compreensão de conceitos definidos. Por exemplo, é possível ensinar o conceito de triângulo com base em suas formas, o que caracteriza a apresentação de um conceito concreto. Posteriormente, uma definição formal pode ser colocada. Neste caso, o conceito concreto auxilia no entendimento do conceito definido. As abstrações constituem um tipo de conceito definido que não possui um conceito concreto correspondente. Exemplos de abstrações são: amor, alegria e raciocínio. A verificação do aprendizado de um conceito definido vai além da simples definição em palavras, devendo incluir questões que solicitem a identificação dos componentes do conceito, tais como relações e pré-requisitos. Com relação às condições internas, há necessidade que o aluno conheça todos os componentes do conceito, incluindo fatos, relações, etc. Uma condição externa adequada é a apresentação de uma definição. Outra condição externa recomendável é a apresentação de exemplos e contra-exemplos do conceito definido, objetivando estender o significado do mesmo;
- *Regras simples*: uma regra é uma capacidade interna que governa o comportamento de um indivíduo e permite que ele apresente uma resposta apropriada a uma classe de situações que possuam relação entre si (GAGNÉ et al., 1992 apud MARIETTO, 2000). Uma regra é freqüentemente nomeada e é considerada aprendida quando é possível dizer com certeza que o desempenho do aluno tem um tipo de regularidade. Ao contrário dos conceitos definidos, em algumas situações o indivíduo pode não conseguir verbalizar uma regra, mesmo sabendo aplicá-la. Este fato acontece, por exemplo, em questões de reconhecimento de padrão. Um cardiologista experiente consegue, através da análise de um eletrocardiograma, determinar as causas de uma determinada disfunção, mas pode não conseguir explicar com precisão as regras que usou para chegar ao diagnóstico. É possível notar que os conceitos constituem um tipo particular de regra que objetiva classificar objetos e eventos e, neste caso, os conceitos são considerados regras de classificação. As condições internas necessárias para o aprendizado de uma dada regra é o conhecimento de todos os conceitos relacionados à regra. Com relação às condições externas, regras geralmente são “comunicadas” aos alunos lembrando-se dos elementos que as compõem e os direcionando a ordenar estes elementos em uma seqüência adequada;
- *Resolução de problemas e regras complexas*: regras complexas são formadas por combinações de regras simples, com o objetivo de executar uma tarefa ou resolver um problema. Quando um aluno consegue chegar à solução de um problema, significa que adquiriu uma nova regra, ou talvez um conjunto delas. A condição essencial para que este tipo de aprendizagem ocorra é a ausência de

uma orientação detalhada (GAGNÉ et al., 1992 apud MARIETTO, 2000). A solução deve ocorrer por descoberta. A orientação deve se restringir a oferecer ou relembrar as regras básicas necessárias à descoberta da solução do problema. Isto não significa que a solução ocorra a partir do nada. Pelo contrário, de acordo com Gagné e Medsker (1996 apud MARIETTO, 2000), a solução de problemas é fortemente dependente do conhecimento do domínio específico. O desempenho esperado do aluno, que permite verificar o aprendizado deste tipo de conhecimento, requer a “invenção” e o uso de regras de alta ordem para encontrar a solução de um novo problema. As condições internas necessárias são o domínio de estratégias cognitivas, regras e conhecimentos relacionados à solução do problema. As condições externas se restringem à apresentação de um novo problema e de algumas dicas, caso se considere necessário.

3.3.1.3 *Estratégia Cognitiva*

Segundo Gagné (1980), as estratégias cognitivas são capacidades internamente organizadas das quais o aluno faz uso para guiar sua própria atenção, aprendizagem, memorização e pensamento. Considerando que uma estratégia cognitiva já foi aprendida por um aluno, ela pode ser utilizada como um modo de resolução de um novo problema. Assim, o raciocínio do encadeamento para frente (*forward chaining*) é uma estratégia cognitiva que pode ser usada por um aluno na resolução de vários problemas. Geralmente as estratégias cognitivas são dependentes de domínio, ou seja, algumas estratégias são mais adequadas para um tipo de domínio que outras. Por exemplo, existem estratégias para a fixação de palavras, para a aplicação de regras, para o treinamento de habilidades motoras, etc. Mas existem estratégias cognitivas genéricas, tais como os processos de dedução e indução. Estes tipos de estratégia demandam maior tempo para serem aprendidos, mas podem ser aplicados na solução de uma gama maior de problemas, bem como em situações de raciocínio e aprendizagem.

3.3.1.4 *Habilidade Motora*

As habilidades motoras correspondem aos movimentos físicos e motores do aluno. Segundo Gagné (1980), estas habilidades são aprendidas em conexão com atividades humanas comuns como: dirigir um carro, usar uma máquina e tocar um instrumento musical.

3.3.1.5 *Atitude*

Segundo Gagné (1977), a atitude é um estado interno que influencia (modera) as escolhas de ação pessoal feitas pelo indivíduo. A frequência com a qual uma pessoa toma uma atitude mostra a força que a mesma tem em sua personalidade. Assim, se uma pessoa tem uma tendência exagerada para falar, ela provavelmente falará quando surgir qualquer oportunidade. Segundo Marietto (2000), ao analisar as atitudes com relação ao processo instrucional, alguns alunos têm atitudes voltadas para a desatenção, para a preguiça, para o detalhamento, para a subjetividade, para a objetividade, etc. Estas atitudes podem ser detectadas após a análise do comportamento de um aluno durante um longo período de tempo. Tal observação às vezes é inviável e por isto, na prática, são realizadas inferências de atitudes baseadas em questionários, perguntando-se que tipo de ação o aluno tomaria em determinadas situações.

3.3.2 Eventos de Instrução

Segundo Marietto (2000), durante o processo de aprendizagem ocorrem mudanças no estado mental do aluno à medida que um conjunto de eventos age e envolve o mesmo. Este conjunto de eventos é chamado por Gagné (1985 apud MARIETTO, 2000) de *eventos de instrução*. Pode-se dizer que os eventos de instrução ordenam as condições externas de aprendizagem de tal forma a garantir que a mesma ocorra. Neste sentido, eventos de instrução são usados como sinônimo para condições externas de aprendizagem.

Em Gagné (1985 apud MARIETTO, 2000) são apresentados nove eventos de instrução relacionados com a aprendizagem: obtenção da atenção, informação dos objetivos, recuperação de conhecimentos prévios necessários à aprendizagem, apresentação do material instrucional, orientação da aprendizagem do aluno, obtenção gradual do desempenho do aluno, oferecimento de retorno sobre o desempenho, avaliação do desempenho, aumento da retenção e transferência. Tais eventos não precisam ser apresentados sequencialmente, nem todos precisam aparecer nas etapas de um projeto instrucional. Abaixo são descritos os eventos de instrução citados acima.

3.3.2.1 Evento 1: Obtenção da Atenção

Segundo Marietto (2000), vários tipos de ações podem ser realizados para obter a atenção do aluno, como por exemplo:

- Explorar o interesse e a curiosidade do aluno;
- Introduzir rápidas mudanças nos estímulos, objetivando deixar o aluno alerta. Isto pode ser feito alterando apresentações visuais, com sons, etc.;
- Fazer perguntas;
- Criar desafios mentais.

3.3.2.2 Evento 2: Informação dos Objetivos

O aluno deve estar informado sobre os objetivos da instrução, especificamente sobre o tipo de desempenho esperado para que ele possa ter um indicativo de como a instrução está se desenvolvendo, bem como estabelecer as expectativas apropriadas. Segundo Marietto (2000), deve-se tomar o cuidado de não apresentar muita informação, mas somente a informação relacionada aos objetivos da unidade ou do tópico. É possível mostrar tanto os objetivos quanto os pré-requisitos dos mesmos. Nesta apresentação a hierarquia de aprendizagem pode ser colocada através de estruturas organizacionais tais como uma lista, um grafo, uma tabela, ou então usar frases como “No final deste capítulo você será capaz de ...”.

De acordo com Marietto (2000), com relação ao conhecimento declarativo, recomenda-se que se informe ao aluno se o mesmo deverá declarar, descrever ou explicar algo, bem como qual informação e em que nível de detalhe ele será avaliado. Com relação às habilidades intelectuais, os objetivos podem ser transmitidos através de exemplos ou por demonstração.

3.3.2.3 *Evento 3: Recuperação de Conhecimentos Prévios Necessários à Aprendizagem*

Este evento prepara o aluno para a aprendizagem, assim como os dois anteriores, na medida em que se recupera conhecimentos já armazenados necessários à aprendizagem do material que está sendo ensinado. Em outras palavras, “relembra” o aluno de conteúdos já aprendidos. Este tipo de comunicação é muito importante, pois grande parte de novas aprendizagens se dá através da combinação de idéias já dominadas pelo aluno.

Com relação ao conhecimento declarativo, faz-se necessário relembrar conhecimentos declarativos já aprendidos e que estão estruturados de forma significativa. Isto pode ser feito através dos organizadores prévios de Ausubel (DAVID AUSUBEL, 2007), de *outlines* e sumários, através de perguntas, da apresentação explícita dos conhecimentos prévios, etc. Já com relação às habilidades intelectuais, faz-se necessário relembrar outros conhecimentos procedurais considerados como pré-requisitos.

3.3.2.4 *Evento 4: Apresentação do Material Instrucional*

Segundo Marietto (2000) o material deve ser apropriado ao assunto abordado. A própria seqüência de apresentação do material já é um tipo de orientação. Pode-se apresentar aos alunos todo o *menu* de aprendizagem.

Para a apresentação do conhecimento declarativo, sugere-se que sejam entregues, por exemplo, figuras, textos e sons organizados de forma significativa para o aluno. Também devem ser utilizadas estratégias que unam a nova informação ao conhecimento já existente.

Quando uma habilidade intelectual é ensinada, pode-se explicar ou demonstrar a habilidade utilizando uma variedade de exemplos em uma variedade de contextos. Características distintas do conceito devem ser enfatizadas. Estas ações objetivam levar o aluno à generalização.

3.3.2.5 *Evento 5: Orientação da Aprendizagem do Aluno*

Este evento tem por objetivo tornar o novo conteúdo o mais significativo possível (GAGNÉ e MEDSKER, 1996 apud MARIETTO, 2000). A orientação não tem por objetivo dizer a resposta, mas sim sugerir uma linha de raciocínio que presumidamente levará à desejada combinação de conceitos e regras subordinadas, que por sua vez, propiciarão a aprendizagem desejada.

Dependendo do assunto abordado e do aluno, a quantidade e o tipo de orientação podem variar. Uma solução prática para esta questão é fornecer orientação um pouco de cada vez, permitindo que o aluno solicite auxílio sempre que considerar necessário. Também em sistemas de ensino individualizado, uma grande parte da orientação está embutida no material, através de sugestões, ícones, dicas, etc. Por isto este evento está inserido, em parte, no evento apresentação do material instrucional.

Com relação ao conhecimento declarativo, pode-se levar o aluno a uma variedade de conexões entre a nova informação e as redes de conhecimento previamente organizadas. O uso de imagens, sons, histórias e gráficos são aconselháveis. Já para as habilidades intelectuais, os conceitos são melhor ensinados especificando seus atributos e características críticas, apresentando exemplos e contra-exemplos. Regras podem ser

apresentadas e logo após, demonstradas ou aplicadas em uma variedade de contextos. Em ambos os casos, é importante oferecer meios pelos quais o aluno possa unir novas informações com as já existentes.

3.3.2.6 *Evento 6: Obtenção Gradual do Desempenho do Aluno*

Segundo Marietto (2000), após um determinado tempo de estudo, é necessário verificar como está se processando a aprendizagem do aluno. Eventos com esse objetivo são comumente chamados de “prática” (exercícios).

Para o conhecimento declarativo pode ser solicitado ao aluno declarar a informação. Para as habilidades intelectuais, pode-se solicitar que o aluno as demonstre ou aplique.

3.3.2.7 *Evento 7: Oferecimento de Retorno sobre o Desempenho*

Este evento oferece retorno ao aluno sobre seu desempenho. Não é aconselhável esperar até o momento de um teste para fornecer retorno, mas sim após freqüentes intervalos. Isto pode ser feito, por exemplo, através de questões inseridas no meio do material instrucional. Alguns retornos podem ser entusiásticos como “Muito bem, você está correto!!!”. Outros retornos podem ser direcionadores como “Recomendo que você volte ao tópico anterior para rever o conceito de par ordenado.”. Quando o aluno cometer erros, ele deve ser informado sobre o motivo de suas respostas terem sido consideradas incorretas. Desta forma, o retorno será encarado como informação para seu desenvolvimento e não como crítica.

3.3.2.8 *Evento 8: Avaliação do Desempenho*

Este tipo de evento procura responder à seguinte questão: como verificar se a aprendizagem apresentada por um aluno, via seu desempenho, é *válida*? Também procura auxiliar no reforço e na recuperação de informação. Uma primeira avaliação é feita durante o evento “obtendo gradualmente o desempenho”, mas pode ser incompleta. Por isto, podem ser necessárias outras avaliações ou testes.

3.3.2.9 *Evento 9: Aumento da Retenção e Transferência*

Este evento é acionado após a avaliação de desempenho do aluno, quando o mesmo tiver obtido um resultado considerado satisfatório nesta avaliação. Entende-se por retenção a habilidade de reproduzir um comportamento aprendido, após um determinado espaço de tempo de sua aquisição. Transferência é a capacidade de usar os conhecimentos e habilidades aprendidos em situações diferentes das quais ocorreu a aprendizagem.

3.4 Modelo Hyper-Automaton

O *AdaptHA* se baseia no Hyper-Automaton, cujo modelo suporta o gerenciamento e a apresentação de hiperdocumentos instrucionais como cursos na *Web*. O Hyper-Automaton utiliza o formalismo de *Autômatos Finitos com Saída* como um modelo estrutural para a organização de hiperdocumentos.

Nesse modelo cada *autômato* define um *curso* e referencia um conjunto de hiperdocumentos independentes, tal que a estrutura navegacional do curso permanece separada do material instrucional. Esse mecanismo estimula o reuso do material

instrucional em diversos cursos além de possibilitar a construção de cursos com diferentes abordagens e objetivos utilizando os mesmos hiperdocumentos.

Nas subseções seguintes são apresentadas as definições de autômatos finitos e autômatos finitos com saída, em seguida é apresentada a perspectiva de cursos como autômatos e por fim são listadas as vantagens do modelo discutido nesta seção.

3.4.1 Autômatos Finitos

Segundo Menezes (2002), um *Autômato Finito Determinístico* ou simplesmente *Autômato Finito* pode ser visto como uma máquina composta, basicamente, de três partes:

- *Fita*: dispositivo de entrada que contém a informação a ser processada;
- *Unidade de controle*: reflete o estado corrente da máquina. Possui uma unidade de leitura (cabeça da fita), a qual acessa uma célula da fita de cada vez e se movimenta exclusivamente para a direita;
- *Programa* ou *função de transição*: função que comanda as leituras e define o estado da máquina.

A fita é finita (à esquerda e à direita), sendo dividida em células, em que cada uma armazena um símbolo. Os símbolos pertencem a um alfabeto de entrada. Não é possível gravar sobre a fita (e não existe memória auxiliar). Inicialmente, a palavra a ser processada (ou seja, a informação de entrada para a máquina) ocupa toda a fita.

A unidade de controle possui um número finito e predefinido de estados. A unidade de leitura lê o símbolo de uma célula de cada vez. Após a leitura, a *cabeça da fita* se move uma célula para a direita. Inicialmente, a cabeça está posicionada na célula mais à esquerda da fita.

O programa é uma função parcial que, dependendo do estado corrente e do símbolo lido, determina o novo estado do autômato. O autômato finito não possui memória de trabalho. Portanto, para armazenar as informações passadas, necessárias ao processamento, deve-se usar o conceito de estado.

Um autômato finito M é uma 5-upla:

$$M = (\Sigma, Q, \delta, q_0, F)$$

onde:

Σ : alfabeto de símbolos de entrada

Q : conjunto de estados possíveis do autômato, o qual é finito

δ : função programa ou função (parcial) de transição

$$\delta: Q \times \Sigma \rightarrow Q$$

q_0 : estado inicial tal que q_0 é elemento de Q

F : conjunto de estados finais tal que F está contido em Q

A função programa pode ser representada como um grafo finito direto como ilustrado na figura 3.2.

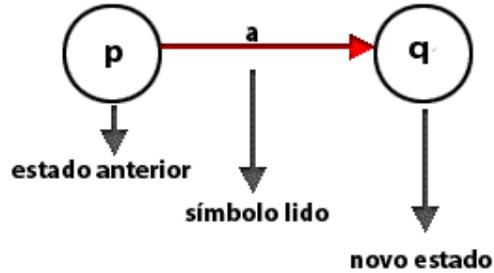


Figura 3.2: Representação da função programa como um grafo (MENEZES, 2002).

O processamento de um autômato finito M , para uma palavra de entrada w , consiste na sucessiva aplicação da função programa para cada símbolo de w (da esquerda para a direita) até ocorrer uma condição de parada.

3.4.2 Autômatos Finitos com Saída

De acordo com Menezes (2002), é possível estender a definição de autômato finito incluindo a geração de uma palavra de saída. As saídas podem ser associadas às transições (máquina de *Mealy*) ou aos estados (máquina de *Moore*). Em ambas as máquinas, a saída não pode ser lida, ou seja, não pode ser usada como memória auxiliar, e é como segue:

- É definida sobre um alfabeto especial, denominado alfabeto de saída (pode ser igual ao alfabeto de entrada);
- A saída é armazenada em uma fita independente da de entrada;
- A cabeça da fita de saída move uma célula para a direita a cada símbolo gravado;
- O resultado do processamento do autômato finito é o seu estado final (condição de aceita / rejeita) e a informação contida na fita de saída.

3.4.2.1 Máquina de Mealy

A máquina de *Mealy* é um autômato finito modificado de forma a gerar uma palavra de saída para cada transição. Formalmente, tem-se que uma máquina de *Mealy* M é uma 6-upla:

$$M = (\Sigma, Q, \S, q_0, F, \Delta)$$

Σ : alfabeto de símbolos de entrada

Q : conjunto de estados possíveis do autômato, o qual é finito

\S : função programa ou função (parcial) de transição

$$\S: Q \times \Sigma \rightarrow Q \times \Delta^*$$

q_0 : estado inicial tal que q_0 é elemento de Q

F : conjunto de estados finais tal que F está contido em Q

Δ : alfabeto de símbolos de saída

3.4.2.2 Máquina de Moore

A máquina de *Moore* possui uma segunda função, que gera uma palavra de saída (que pode ser vazia) para cada estado da máquina. Formalmente, tem-se que uma máquina de *Moore* M é uma 7-upla:

$$M = (\Sigma, Q, \xi, q_0, F, \Delta, \xi_s)$$

Σ : alfabeto de símbolos de entrada

Q : conjunto de estados possíveis do autômato, o qual é finito

ξ : função programa ou função (parcial) de transição

$$\xi: Q \times \Sigma \rightarrow Q$$

q_0 : estado inicial tal que q_0 é elemento de Q

F : conjunto de estados finais tal que F está contido em Q

Δ : alfabeto de símbolos de saída

ξ_s : função (total) de saída

$$\xi_s: Q \rightarrow \Delta^*$$

3.4.3 Cursos como Autômatos

Cada autômato define um curso que referencia um conjunto de páginas independentes (unidades de informação). A função programa ou função de transição funciona como ligação lógica entre os hiperdocumentos e a função de saída compõe as páginas. O resultado final é a estrutura básica de páginas e *links* de hipertexto em um *site* na *Web*.

As n -uplas do autômato, usadas para modelar os cursos, têm as seguintes correspondências na estrutura de hiperdocumentos na *Web*:

- O alfabeto de entrada Σ é um conjunto de nomes que identifica os *links* de navegação no curso hipertexto;
- O estado inicial q_0 corresponde ao ponto de entrada do curso, normalmente denominado *homepage*;
- O alfabeto de saída Δ é um conjunto de páginas *Web* relacionadas às unidades de informação;
- As palavras de saída Δ^* , presentes na função de transição ξ e na função de saída ξ_s , são unidades de informação concatenadas e apresentadas como uma única página para o usuário.

Neste modelo, um curso pode ser visto sob dois principais formalismos:

- *Máquina de Moore*: ao modelar o curso, a saída é relacionada às unidades de informação, que passam por um processo de concatenação para serem apresentadas ao aluno. Uma importante característica é que as páginas não contêm *links hard-coded*, as transições entre os estados do autômato determinam que *links* estarão disponíveis ao final de cada página;
- *Máquina de Mealy*: cada palavra de saída corresponde a unidades de informação do curso. As transições entre os estados funcionam como ligações lógicas que conectam os conteúdos do curso e serão vistos pelo aluno como *links*.

3.4.4 Vantagens do Modelo Hyper-Automaton

O modelo Hyper-Automaton apresenta as seguintes vantagens:

- Alto grau de modularização do material instrucional;
- Independência entre a base de hiperdocumentos e a estrutura de controle da aplicação hipermídia (autômato), cuja alteração não influencia nas páginas e vice-versa;
- Suporte à especificação de vários conjuntos de *links* sobre um mesmo corpo de hiperdocumentos mantendo os objetos separados da estrutura de navegação;
- Facilidade de reuso de páginas em diversos cursos, com eliminação da redundância;
- Suporte à criação de *links* de e para qualquer documento;
- Facilidade de implementação e manutenção;
- Suporte à elaboração de seqüências instrucionais com objetivos específicos e abordagens diferentes, com possibilidade de oferecer estudo individualizado;
- Possibilidade de composição de cursos por meio de Operações Categóricas, ou seja, a construção de novos cursos sobre cursos já existentes através de procedimentos de alto nível;
- Opção de limitação da excessiva liberdade do aluno ao explorar hipertextos, pois com o modelo de autômatos pode ser construído um material hipertexto fortemente hierarquizado, no qual os conteúdos são organizados em múltiplos níveis (definidos pela ordem dos estados do autômato);
- Possibilidade de gerenciamento da navegação dos alunos individualmente por meio da análise dos caminhos do autômato. Desta forma, os alunos podem continuar o curso a partir do ponto em que pararam na sessão de uso anterior. Para que isto seja possível é necessário salvar o estado de execução do autômato;
- Modificação da função de transição do autômato para possibilitar a inclusão de restrições temporais no modelo. Regras temporais controlam o sincronismo de diferentes atividades envolvidas no curso. Um exemplo de restrição é um limite temporal para a execução de uma atividade específica, por exemplo, completar um curso no período de quatro meses.

3.5 Arquitetura do *AdaptHA*

A arquitetura proposta para o *AdaptHA* é exibida na figura 3.3. Tal arquitetura é composta por três grandes camadas: *camada de interface*, *camada de negócio* e *camada de persistência*. Estas camadas são descritas em detalhes nas próximas subseções.

3.5.1 Camada de Interface

A camada de interface provê, de forma personalizada, o diálogo entre usuário e sistema. Cada tipo de usuário recebe uma interface adaptada com base no papel que desempenha no sistema. A camada de interface do *AdaptHA* é capaz de gerenciar quatro tipos de papéis desempenhados pelo usuário: *aluno*, *professor*, *autor* e *administrador*.

Um único usuário pode assumir mais de um papel no sistema, por exemplo, um autor pode atuar também como administrador ou aluno, sem a necessidade de manter cadastros diferenciados (separados) para isto.

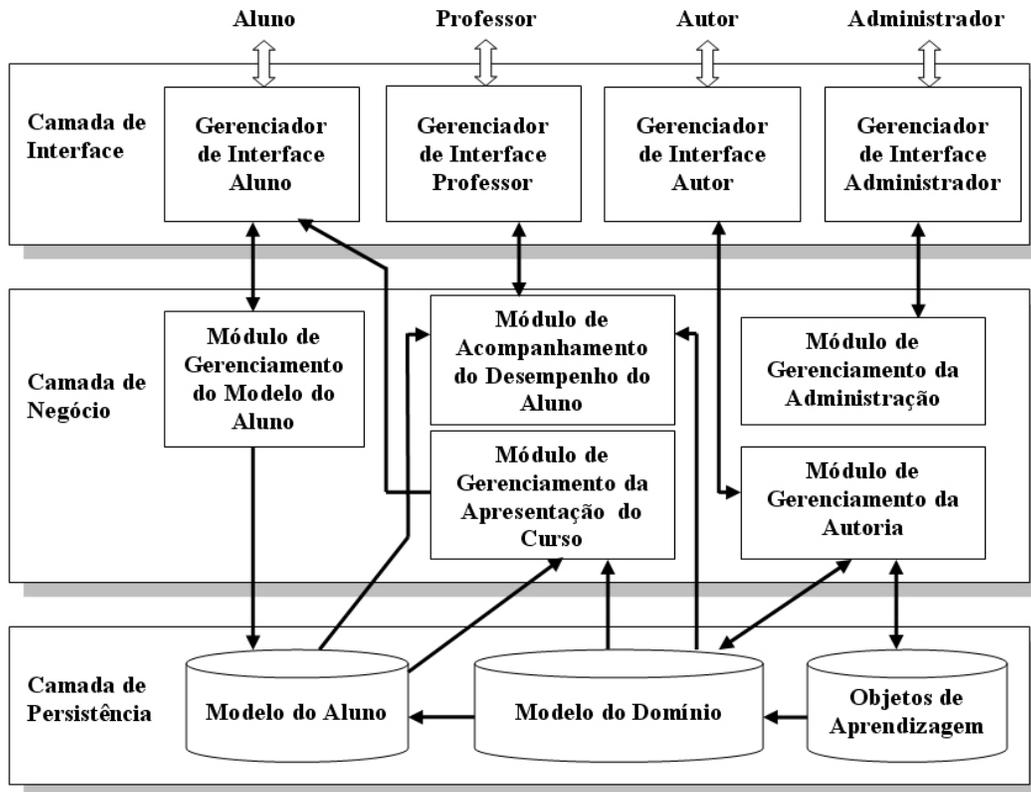


Figura 3.3: Arquitetura do *AdaptHA*.

A camada de interface do *AdaptHA* foi cuidadosamente projetada. Seu projeto levou em consideração os seguintes aspectos: *usabilidade*, recursos de *localização* e recursos de *internacionalização*. Tais aspectos são discutidos abaixo. Em seguida são examinados os gerenciadores de interface de cada um dos papéis de usuário que o sistema é capaz de gerenciar.

3.5.1.1 Usabilidade

De acordo com Nielsen (2007), a usabilidade é um atributo de qualidade que avalia o quão fácil de usar são as interfaces de usuário. A palavra usabilidade também se refere aos métodos para melhoria da facilidade de uso durante o processo de *design*. A usabilidade é definida através de cinco componentes de qualidade:

- *Facilidade de aprendizado*: quão fácil é para os usuários realizarem tarefas básicas na primeira vez em que eles se deparam com o sistema?
- *Eficiência de uso*: uma vez que os usuários já aprenderam a interagir com o sistema, quão rapidamente eles conseguem executar as tarefas?
- *Facilidade de memorização*: quando os usuários retornam ao sistema depois de um período sem usá-lo, quão facilmente eles conseguem restabelecer a proficiência?

- *Erros*: quantos erros os usuários cometem, quão severos são esses erros e quão facilmente eles conseguem se recuperar desses erros?
- *Satisfação*: quão agradável é para o usuário usar o sistema?

3.5.1.2 Localização

A localização se refere à adaptação de um produto, aplicação ou conteúdo de um documento com o propósito de adequá-los ao idioma, cultura ou outros requisitos de um mercado alvo específico (uma localidade (“*locale*”). A localização é algumas vezes denominada *l10n*, em que 10 é o número de letras entre 1 e n (considerando-se o termo em inglês). A localização é freqüentemente pensada como sinônimo de tradução de uma IU e documentação de um idioma para outro. No entanto, a localização, em geral, envolve questões substancialmente mais complexas, que podem abranger customizações relacionadas a (W3C, 2007):

- Formatos numéricos, de data e de hora;
- Moeda;
- Utilização do teclado;
- Critérios de comparação (*collation*) e ordenamento;
- Símbolos, ícones e cores;
- Textos e gráficos contendo referências a objetos, ações ou idéias que, em uma dada cultura, podem ser mal interpretados;
- Requisitos legais;

3.5.1.3 Internacionalização

O termo globalização é utilizado por alguns como sinônimo de internacionalização. A internacionalização compreende o projeto e desenvolvimento de um produto, aplicação ou conteúdo de um documento que permite *fácil* localização para públicos-alvo que variam em cultura, região ou idioma. A internacionalização é normalmente denominada *i18n*, em que 18 é o número de letras entre i e n (considerando-se o termo em inglês). A internacionalização envolve questões como (W3C, 2007):

- Projetar e desenvolver de forma a remover barreiras de localização ou *deployment* internacional. Isto inclui coisas como habilitar o uso de *Unicode* ou assegurar a manipulação apropriada de codificações de caracteres antigos, tomar cuidado com concatenação de *strings*, evitar dependências em *strings* apresentadas na IU, etc.;
- Prover suporte a características que podem não ser utilizadas até que a localização ocorra, como por exemplo: suporte a texto bidirecional, identificação de idioma, suporte a texto vertical ou outras características tipográficas não latinas;
- Habilitar o código para suportar preferências locais, regionais, de idioma ou culturais. Normalmente isto envolve incorporar dados e características predefinidas de localização derivadas de bibliotecas existentes ou de preferências do usuário. Exemplos incluem: formatos de data e hora, calendários

locais, formatos de números, sistemas numéricos, manipulação de nomes pessoais e formatos de endereço, etc.;

- Separar elementos localizáveis do código fonte ou conteúdo, tal que alternativas localizadas possam ser carregadas ou selecionadas com base nas preferências internacionais do usuário, conforme necessário.

Os itens acima não necessariamente incluem a localização do conteúdo, aplicação ou produto em outra língua. Eles constituem práticas de projeto e desenvolvimento que possibilitam que tal migração ocorra facilmente no futuro, mas que podem ter significativa utilidade, mesmo que nenhuma localização seja necessária (W3C, 2007).

O *AdaptHA* é capaz de localizar seu conteúdo no que diz respeito a formatos numéricos, formatos de data, formatos de hora e moeda. Sua versão atual suporta os idiomas *português* e *inglês*. No entanto, a extensão a outros idiomas é bastante facilitada devido aos mecanismos de internacionalização e localização utilizados. Tais mecanismos facilitam sobremaneira a localização porque dispensam alterações no código fonte do sistema. Os mecanismos de localização e internacionalização do *AdaptHA* serão mais detalhados na subseção 4.1.3.

3.5.1.4 Gerenciadores de Interface

Autores, de maneira geral, são responsáveis por criar, atualizar e disponibilizar conteúdo didático que será utilizado posteriormente por outros autores, professores e alunos. O *gerenciador de interface do autor* é responsável por permitir que os autores realizem um conjunto de funcionalidades relacionadas às tarefas de autoria.

Professores são responsáveis por acompanhar e fornecer *feedback* aos alunos a respeito de seu desempenho. O *gerenciador de interface do professor* é responsável por permitir que os professores efetuem um conjunto de funcionalidades relacionadas ao acompanhamento e orientação dos alunos.

Alunos são responsáveis por realizar os cursos disponibilizados por autores ou professores. O *gerenciador de interface do aluno* é responsável por permitir que os alunos se matriculem, realizem os cursos disponibilizados pelo sistema e forneçam *feedback* a respeito dos cursos realizados.

O *AdaptHA* trabalha com dois tipos de administradores: *administradores do sistema* e *administradores de instituição*. Essa diferenciação é necessária porque uma instalação do *AdaptHA* é capaz de gerenciar inúmeras *instituições* simultaneamente. O *gerenciador de interface do administrador* é responsável por permitir que os administradores do sistema gerenciem informações relativas ao sistema como um todo e também é responsável por permitir que os administradores de instituição gerenciem informações relativas à instituição que administram.

3.5.2 Camada de Persistência

A camada de persistência armazena os *objetos de aprendizagem*, o *modelo do domínio* e o *modelo do aluno*.

3.5.2.1 Objetos de Aprendizagem

Um objeto de aprendizagem é qualquer entidade, digital ou não digital, que pode ser utilizada para aprendizagem, educação ou treinamento (IEEE LTSC, 2007). Exemplos de objetos de aprendizagem são: um arquivo de imagem, um arquivo de áudio, uma

animação, uma página HTML, etc. O conceito de objeto de aprendizagem utilizado neste trabalho e implementado no *AdaptHA*, é o de arquivos aos quais são associados metadados quando da efetivação do *upload* dos mesmos para o sistema.

No *AdaptHA* os objetos de aprendizagem são mantidos de maneira *independente* do curso, o que conseqüentemente possibilita a *reutilização* desses objetos em diferentes contextos. Com o objetivo de facilitar a recuperação e reutilização dos objetos de aprendizagem, são associados a cada um deles os seguintes metadados:

- Nome;
- Descrição;
- Palavras-chave;
- Idioma em que o objeto de aprendizagem está disponível, quando se aplica;
- Nível de compartilhamento, que pode ser tanto público (disponível para uso para todos os autores cadastrados no sistema) quanto privado (disponível para uso somente para os autores do objeto de aprendizagem);
- Autores;
- Formato;
- Tamanho;
- Localização.

3.5.2.2 Modelo do Domínio

O modelo do domínio do *AdaptHA* se caracteriza como um modelo de nível três (BRUSILOVSKY, 1996-b), ou seja, representa a estrutura interna de cada conceito como um conjunto de atributos (metadados). O modelo do domínio tem uma função muito importante, que é a de fornecer a estrutura para representação do domínio de conhecimento do aluno, ou seja, o modelo do aluno, que é detalhado a seguir.

O modelo do domínio do *AdaptHA* contempla os *cursos, conceitos, recursos didáticos, bibliotecas de bugs e operadores de remediação*.

Entende-se como *curso* um conjunto de conceitos didaticamente estruturados que mantêm entre si relacionamentos de *pré-requisitos*. O *AdaptHA* se baseia no modelo Hyper-Automaton, detalhado na seção 3.4, de maneira que cada curso é representado como um autômato finito com saída.

As n-uplas do autômato têm as seguintes correspondências na estrutura dos cursos no *AdaptHA*:

- O alfabeto de entrada Σ é um conjunto de nomes que identifica os conceitos do curso;
- Pode haver um ou mais estados (conceitos) iniciais para o curso, de acordo com os relacionamentos de pré-requisitos estabelecidos entre os conceitos;
- O alfabeto de saída Δ é um conjunto de recursos didáticos, que são conteúdos instrucionais, conforme apresentado mais abaixo;
- As palavras de saída Δ^* , presentes na função de transição δ e na função de saída δ_s , são recursos didáticos concatenados que são apresentados para o aluno.

A um curso devem estar associados os seguintes metadados:

- Nome;
- Descrição;
- Disciplina, por exemplo, Linguagens Formais e Autômatos;
- Áreas, por exemplo, Ciência da Computação e Matemática. É importante destacar que um curso pode ser elaborado visando atender uma mesma disciplina oferecida a diferentes áreas. Considera-se que tal aspecto venha a reduzir sobremaneira as tarefas de autoria;
- Nível de abrangência. O curso pode ser abrangente ou profundo;
- Idioma em que o curso estará disponível;
- Desempenho exigido, ou seja, um percentual de acerto nas avaliações realizadas pelo aluno, que possibilite classificar seu nível de conhecimento acerca de um determinado conceito em básico, intermediário ou avançado;
- Autores;
- Nível de compartilhamento, que pode ser tanto público (o curso estará disponível para quaisquer alunos cadastrados no sistema) quanto privado (o curso estará disponível somente para alunos cadastrados no sistema e que foram aprovados pelo professor para participar do curso);
- Período, ou seja, uma data de início e término para o curso (caso a data de término não seja informada, o curso permanecerá sempre disponível a partir da data de início informada).

Depois de criar o curso, o autor deve criar os *conceitos* do curso e estabelecer os relacionamentos de pré-requisitos entre os conceitos criados. A cada conceito de um curso devem estar associados os seguintes metadados:

- Nome;
- Descrição;
- Assunto, por exemplo, Linguagens Regulares. O assunto deve estar relacionado à disciplina que o curso visa atender;
- Tópico, por exemplo, Autômatos Finitos. O tópico deve estar relacionado ao assunto que o conceito abrange;
- Pré-requisitos, ou seja, um conjunto de conceitos que são pré-requisitos do conceito que está sendo criado;
- Palavras-chave;
- Objetivo educacional do conceito. O objetivo educacional deve ser mensurável e pode ser expresso como uma sentença do tipo: “Após estudar este conceito você deverá ser capaz de ...”. A fim de qualificar melhor o objetivo educacional, são associados a ele os seguintes metadados:
 - Domínio de aprendizagem, que pode ser: conceito concreto, conceito definido e regra simples;
 - Verbo, que pode ser: identificar, demonstrar, classificar, definir e aplicar.

A cada conceito a ser ensinado no curso devem estar associados *recursos didáticos*. Recursos didáticos são conteúdos instrucionais utilizados com o objetivo de auxiliar na aprendizagem de um determinado conceito. Em um conceito, os recursos didáticos são seqüenciados de maneira a promover o ensino de forma adequada, conforme pode ser visto na tabela 3.5. A seqüenciação dos recursos didáticos, para cada conceito do curso, é feita tomando como base os eventos instrucionais do Projeto Instrucional (detalhado na seção 3.3). Na subseção 3.5.3 abaixo, detalha-se o comportamento do *AdaptHA*, quanto à apresentação dos recursos didáticos, para cada uma das estratégias pedagógicas possíveis (diretiva, orientada por descoberta e exploratória).

Os tipos de recursos didáticos utilizados pelo *AdaptHA* são:

- *Textos*: podem ser de três tipos:
 - Introdução;
 - Explicação;
 - Conclusão.
- *Exemplos*: podem ser de quatro tipos (MALTEMPI, 1995 apud MARIETTO, 2000):
 - *Iniciais*: apresentam o conceito abordado levando em consideração seus pré-requisitos, definições, aplicações, etc.;
 - *Referência*: estão relacionados a conceitos fundamentais de uma área;
 - *Modelo*: apresentam situações que ilustram a essência do conceito que está sendo trabalhado;
 - *Contra-exemplos*: apresentam situações nas quais uma afirmação relacionada a um conceito não é verdadeira.
- *Avaliações*: são compostas de questões objetivas de *múltipla escolha e verdadeiro / falso* (ZANELLA, 2005) e podem ser três tipos (BLAYA, 2007):
 - *Diagnóstica*: tem dois objetivos básicos: identificar as competências do aluno e adequar o aluno num grupo ou nível de aprendizagem. No entanto, os dados fornecidos pela avaliação diagnóstica não devem ser tomados como um “rótulo” que permanecerá colado para sempre ao aluno, mas sim como um conjunto de indicações a partir do qual o aluno possa conseguir um processo de aprendizagem. No *AdaptHA* a avaliação diagnóstica é aplicada antes do início do processo de ensino de cada conceito;
 - *Formativa*: é uma forma de avaliação em que a preocupação central reside em coletar dados para reorientação do processo de ensino-aprendizagem. Trata-se de uma “bússola orientadora” do processo de ensino-aprendizagem. O resultado de uma avaliação formativa não deve ser expresso através de uma nota, mas sim através de comentários. Neste texto o termo *exercício* também será utilizado para designar a avaliação formativa. No *AdaptHA* a avaliação formativa é aplicada durante o processo de ensino de cada conceito;
 - *Somativa*: tem como o objetivo representar um sumário, uma apresentação concentrada de resultados obtidos em uma situação

educacional. Com este tipo de avaliação, pretende-se traduzir, de forma *quantificada*, a distância entre o resultado obtido pelo aluno e uma meta que se arbitrou ser importante atingir. No *AdaptHA* a avaliação somativa é aplicada após a finalização do processo de ensino de cada conceito.

- *Definições*: para Gagné e Medsker (1996 apud MARIETTO, 2000) uma definição é uma afirmação que expressa regras de classificação. Uma definição deve conter características essenciais e funções do objeto ou relação que está sendo definido, bem como as superclasses das quais o conceito é um subconjunto;
- *Dicas*: lembretes e perguntas.

A cada curso é associada uma *biblioteca de bugs*. A biblioteca de *bugs* possui um catálogo de erros esperados e é construída manualmente pelo professor com base na análise de erros comumente cometidos pelos alunos. A idéia é que futuramente o sistema seja capaz de alimentar automaticamente a biblioteca de *bugs* com base no *feedback* corretivo dos erros cometidos pelos alunos durante a realização das avaliações. Para cada *bug* catalogado na biblioteca de *bugs*, deve haver um *operador de remediação*. Um operador de remediação é um conjunto de ações a serem executadas a fim de retirar do modelo do aluno uma concepção incorreta e inserir a concepção correta correspondente (MARIETTO, 2000).

3.5.2.3 Modelo do Aluno

AdaptHA mantém um modelo do aluno de maneira que torna possível a adaptação de conteúdo e o suporte à navegação adaptativa. Tal modelo pode ser caracterizado como:

- *Adaptativo e adaptável*: é adaptativo pelo fato de ser atualizado automaticamente pelo sistema com base na informação obtida a partir do comportamento do aluno. No entanto, tal modelo também é adaptável pelo fato de permitir que o aluno exponha ao sistema suas preferências;
- *Atualizado através de questionamento e de observação*: é atualizado através de questionamento ao permitir que o aluno exponha suas preferências ao sistema. No entanto, tal modelo é primordialmente de observação, porque a partir do comportamento do aluno, o modelo é atualizado;
- *Dinâmico*: pelo fato de haver atualização constante do modelo do aluno a partir de questionamento ou observação;
- *Agrupado e individual*: é agrupado porque classifica o aluno em um dos três possíveis níveis de conhecimento, ou seja, estereótipos: *básico*, *intermediário* e *avançado*. Ao mesmo tempo é individual porque utiliza o modelo de perturbação, que representa as concepções corretas e incorretas do aluno;
- *Interno*: pelo fato de estar embutido no sistema;
- *Explícito*: pelo fato de possuir uma representação separada do modelo do domínio, conforme mostrado na figura 3.3;
- *Visível e parcialmente modificável*: é visível porque as suposições que o sistema faz a respeito do aluno estão visíveis para o mesmo. No entanto, o aluno pode modificar somente parte do modelo, particularmente suas preferências;

- *De baixa granularidade*: por possuir um alto nível de detalhe do conteúdo representado;
- *De longo prazo*: pelo fato do modelo ser mantido pelo sistema entre uma sessão de uso e outra.

Quanto à natureza do conteúdo do modelo do aluno, pode-se afirmar que é primordialmente um modelo de conhecimento do domínio. Isso significa que o modelo do aluno utilizado pelo *AdaptHA*, contém conhecimento que o sistema assume que o aluno tem sobre o domínio. Complementarmente, tal modelo mantém as preferências do aluno, de maneira que é possível que o sistema se adapte, por exemplo, à estratégia pedagógica preferida pelo aluno.

AdaptHA utiliza duas formas de representação do modelo do aluno: *estereótipo* e *perturbação*. O modelo estereótipo classifica o aluno em um dos três possíveis níveis de conhecimento (estereótipos): *básico*, *intermediário* e *avançado*. O aluno é classificado em um estereótipo de acordo com o desempenho exigido, informado pelo autor quando da criação do curso, ou seja, um percentual de acerto nas avaliações realizadas. Com base no estereótipo do aluno, o sistema irá selecionar a estratégia pedagógica adequada. Já o modelo de perturbação é utilizado para representar as concepções corretas e incorretas do aluno.

O modelo do aluno contempla:

- *Dados pessoais do aluno*:
 - Nome;
 - E-mail;
 - Data de nascimento;
 - Sexo;
 - Nome de usuário;
 - Senha (mantida de forma criptografada);
 - Instituição à qual o aluno pertence. O aluno não precisa necessariamente estar vinculado a uma instituição para utilizar o *AdaptHA*, e neste caso deve declarar não possuir vínculo com instituição ao efetuar seu cadastro.
- *Dados sobre cada sessão de uso*:
 - Identificação da sessão, com datas e horários de *login* e *logout*;
 - Dados sobre a *localidade* a partir da qual o aluno está acessando o sistema. Este dado é obtido com base nos dados enviados pelo navegador ao servidor *Web* no momento em que o aluno acessa o sistema. Com isto é possível localizar (subseção 3.5.1) o idioma, formatos numéricos, formatos de data, formatos de hora e moeda para o aluno.
 - Para cada curso visitado pelo aluno ao longo da sessão, são armazenados:
 - Estratégia pedagógica, que pode ser *diretiva*, *orientada por descoberta* ou *exploratória* (detalhadas na subseção 3.5.3) selecionada pelo sistema;

- Conceitos desconhecidos: conceitos que não foram apresentados ao aluno;
 - Conceitos analisados: conceitos apresentados ao aluno, mas que não foram submetidos à avaliação somativa;
 - Conceitos satisfatoriamente aprendidos: conceitos que foram submetidos à avaliação somativa e para os quais o aluno obteve o desempenho exigido (o desempenho exigido é informado pelo autor na criação do curso, conforme mencionado anteriormente);
 - Conceitos supostamente alcançados: conceitos que são pré-requisitos de um conceito satisfatoriamente aprendido;
 - Conceitos não satisfatoriamente aprendidos: conceitos que foram submetidos à avaliação somativa e para os quais o aluno não obteve o desempenho exigido;
 - Concepções incorretas identificadas com base na biblioteca de *bugs*;
 - Exemplos, avaliações, definições, dicas e textos apresentados ao aluno.
- *Preferências:*
 - *Tipo de estratégia pedagógica:* no início de uma sessão de uso do sistema o aluno pode optar por um tipo de estratégia – que pode ser *diretiva*, *orientada por descoberta* ou *exploratória* (detalhadas na subseção 3.5.3) – ou delegar ao sistema a responsabilidade pela escolha da estratégia pedagógica mais adequada (com base no modelo do aluno);
 - *Presença ou não de recursos didáticos:* o aluno pode optar por receber ou não ao longo do curso, exemplos, exercícios e dicas.

3.5.3 Camada de Negócio

A camada de negócio do *AdaptHA* modela todos os *processos* envolvidos na manutenção do modelo do aluno, na apresentação do curso, na autoria, na geração de relatórios de acompanhamento do desempenho do aluno e modela também os processos administrativos. Cada um dos módulos da camada de negócio é descrito a seguir.

3.5.3.1 Módulo de Gerenciamento do Modelo do Aluno

O módulo de gerenciamento do modelo do aluno é responsável pela manutenção do modelo do aluno e modela os seguintes processos:

- Criação do modelo do aluno em seu primeiro acesso ao curso, registrando todos os conceitos do curso como desconhecidos;
- Inicialização de uma sessão de uso do sistema para o aluno, recuperando as informações da sessão imediatamente anterior, de maneira que o aluno possa prosseguir no curso exatamente a partir do ponto em que havia parado anteriormente. No início de uma sessão de uso do sistema o aluno pode optar por selecionar uma estratégia pedagógica, ou delegar ao sistema essa seleção, e pode optar por receber ou não ao longo do curso os recursos didáticos;

- Registro das preferências, concepções incorretas e alterações nos *status* dos conceitos do curso;
- Registro dos acertos e erros do aluno em cada questão das avaliações realizadas no curso;
- Registro de cada um dos recursos didáticos do curso apresentados ao aluno;
- Registro da estratégia pedagógica selecionada pelo sistema.

3.5.3.2 Módulo de Gerenciamento da Apresentação do Curso

O módulo de gerenciamento da apresentação do curso é responsável por preparar o conteúdo final que será apresentado ao aluno com base no modelo do aluno. As etapas de preparação do conteúdo final a ser entregue ao aluno são descritas abaixo:

- *Primeiro nível de adaptação*: envolve a escolha de uma estratégia pedagógica a ser aplicada. O *AdaptHA* trabalha com as estratégias pedagógicas *diretiva*, *orientada por descoberta* e *exploratória*. A adoção de uma ou outra estratégia depende primeiramente das *preferências* do aluno, que no início da sessão de uso do sistema pode optar por uma das estratégias pedagógicas disponíveis ou delegar ao sistema a responsabilidade por essa escolha. Caso o aluno delegue ao sistema a responsabilidade pela escolha da estratégia pedagógica a ser utilizada, a decisão do sistema é tomada com base em dois fatores: *nível de conhecimento* do aluno (básico, intermediário e avançado); e *tipo da tarefa* a ser executada pelo aluno, que pode ser de *desempenho de transferência próxima* (ocorre quando uma tarefa é realizada de forma padronizada, estabelecendo-se passo-a-passo uma seqüência bem definida de procedimentos) e *desempenho de transferência distante* (exige do aluno um certo grau de julgamento e abordagens diferentes quando da realização da tarefa). A tabela 3.3 exibe os critérios para a escolha de uma estratégia pedagógica. A tabela 3.4 exibe a relação entre as capacidades humanas, domínios de aprendizagem e os desempenhos de transferência próxima e distante;
- *Segundo nível de adaptação*: aplicar a estratégia pedagógica selecionada no passo anterior, o que resulta em o sistema assumir comportamento diferenciado de acordo com a estratégia pedagógica, oferecendo ao aluno menor ou maior liberdade de navegação pelo curso. O comportamento do sistema para cada uma das estratégias pedagógicas utilizadas pelo *AdaptHA* é o seguinte:
 - A estratégia pedagógica *diretiva* não permite ao aluno alterar a ordem instrucional, o que significa que os conceitos do curso são apresentados conforme a ordem especificada pelo autor (os pré-requisitos de cada conceito deverão ser respeitados) e os eventos de instrução relacionados à aprendizagem de um conceito são apresentados seqüencialmente conforme a estrutura exibida na tabela 3.5;
 - A estratégia pedagógica *orientada por descoberta*, assim como a *diretiva*, não permite ao aluno alterar a ordem instrucional, entretanto, oferece-se maior liberdade com relação à navegação na estrutura exibida na tabela 3.5. Essa liberdade tem como objetivo permitir ao aluno descobrir o conhecimento do sistema, direcionando seu próprio aprendizado. Um exemplo disso é o fato do aluno poder iniciar a

- navegação pela estrutura de ensino do conceito sem ter realizado uma avaliação diagnóstica;
- A estratégia pedagógica *exploratória* permite que o aluno examine todo o material instrucional disponibilizado no curso, sem qualquer comprometimento com a ordem de conceitos especificada pelo autor e a estrutura de apresentação dos eventos de instrução relacionados à aprendizagem de um conceito em particular.
 - *Terceiro nível de adaptação*: consiste em adaptar a apresentação do conteúdo, assim como promover o suporte à navegação adaptativa ao apresentar os conceitos do curso. Tal adaptação toma como base o modelo do aluno e utiliza algumas das técnicas da HA, detalhadas no capítulo 2. As técnicas da HA utilizadas pelo *AdaptHA*, para apresentação adaptativa e para o suporte à navegação adaptativa são, respectivamente:
 - *Inserção / remoção de fragmentos*: para inserir ou remover fragmentos o *AdaptHA* leva em consideração três aspectos:
 - *Áreas às quais o curso atende*: se o curso atende a x áreas (por exemplo Ciência da Computação e Matemática), então é possível fornecer diferentes conteúdos para cada uma das x áreas.
 - *Níveis de conhecimento do aluno*: é possível fornecer diferentes conteúdos para cada um dos níveis de conhecimento do aluno (básico, intermediário e avançado);
 - *Preferências do aluno*: no início de uma sessão de uso do sistema, o aluno pode optar por receber ou não ao longo do curso os seguintes recursos didáticos: exemplos, exercícios e dicas. Dependendo de suas preferências, tais recursos didáticos são apresentados ou não.
 - *Anotação de links*: o *AdaptHA* é capaz de diferenciar e anotar nove estados de *links*, que são sumarizados na tabela 3.6. A anotação utilizada pelo *AdaptHA* sempre apresenta todos os *links* (nenhum tipo de *dimming* é utilizado, por exemplo), mesmo que o conteúdo alvo do *link* não seja apropriado para o aluno em um dado momento. No entanto, dependendo da estratégia pedagógica adotada pelo sistema ou escolhida pelo aluno, o *link* pode se apresentar funcional ou não. Se o *link* se apresentar funcional, o clique sobre o mesmo conduzirá o aluno para o conteúdo apontado pelo *link*. Caso contrário, o clique sobre o *link* não apresentará efeito algum. A tabela 3.7 sumariza o comportamento dos *links* para cada uma das estratégias pedagógicas.

Tabela 3.3: Critérios para Escolha de uma Estratégia Pedagógica

Arquitetura	Usada Para	Nível de Conhecimento do Aluno
Diretiva	Transferência Próxima	Básico
Orientada por Descoberta	Transferência Distante	Básico
	Transferência Próxima ou Distante	Intermediário

Exploratória	Transferência Próxima ou Distante	Avançado
--------------	-----------------------------------	----------

Tabela 3.4: Capacidades Humanas X Domínios de Aprendizagem X Tipo da Tarefa

Capacidade Humana	Domínio de Aprendizagem	Tipo da Tarefa
Conhecimento Procedural	Conceitos concretos e conceitos definidos	Desempenho de Transferência Próxima
Conhecimento Procedural	Regras simples	Desempenho de Transferência Distante

Tabela 3.5: Estrutura de Apresentação dos Conceitos

Elemento Estrutural	Ação	Evento de Instrução
Visão Geral	Aplicar avaliação diagnóstica	8
	Apresentar objetivos	2
	Apresentar introdução e foco da instrução	3
	Apresentar pré-requisitos	3
Elaboração	Apresentar material instrucional	4
	Orientar a aprendizagem	5
	Avaliar, de forma gradual, o desempenho do aluno	6
	Oferecer feedback corretivo dos exercícios	7
Síntese	Apresentar sumário	4
	Aplicar avaliação somativa	8
	Oferecer feedback corretivo da avaliação somativa	7
Continuação	Apresentar atividades de retenção e transferência	9

Tabela 3.6: Estados de *Links* Diferenciados e Anotados pelo *AdaptHA*

Estado do <i>Link</i>	Sigla
<i>Links</i> para conceitos desconhecidos (não visitados) prontos para serem aprendidos, ou seja, cujos conceitos pré-requisitos já foram satisfatoriamente aprendidos	EL-1
<i>Links</i> para conceitos desconhecidos (não visitados) não prontos para serem aprendidos, ou seja, cujos conceitos pré-requisitos ainda não foram satisfatoriamente aprendidos	EL-2
<i>Links</i> para conceitos analisados (visitados) prontos para serem aprendidos, ou seja, cujos conceitos pré-requisitos já foram satisfatoriamente aprendidos	EL-3
<i>Links</i> para conceitos analisados (visitados) não prontos para serem aprendidos, ou seja, cujos conceitos pré-requisitos ainda não foram satisfatoriamente aprendidos	EL-4
<i>Links</i> para conceitos satisfatoriamente aprendidos	EL-5
<i>Links</i> para conceitos supostamente alcançados	EL-6

<i>Links</i> para conceitos não satisfatoriamente aprendidos	EL-7
<i>Links</i> para recursos didáticos já visitados	EL-8
<i>Links</i> para recursos didáticos não visitados	EL-9

Tabela 3.7: Comportamento dos *Links* X Estratégias Pedagógicas do *AdaptHA*

Estratégia Pedagógica	Estados dos <i>Link</i>	Comportamento do <i>Link</i>
Diretiva	EL-1, EL-3, EL-5, EL-6, EL-7, EL-8	Funcional
	EL-2, EL-4	Não funcional
	EL-9	Funcional caso o recurso didático seja o próximo a ser visitado, conforme a ordem estabelecida dentro do conceito. Caso contrário, não funcional
Orientada por descoberta	EL-1, EL-3, EL-5, EL-6, EL-7, EL-8, EL-9	Funcional
	EL-2, EL-4	Não funcional
Exploratória	EL-1 a EL-9	Funcional

Com o intuito de tornar mais claros os critérios que o *AdaptHA* utiliza para aplicar a anotação de *links*, tome-se o seguinte curso, conforme exibido na tabela 3.8 abaixo:

Tabela 3.8: Exemplo de Um Curso no *AdaptHA*

Curso	Conceito	Pré-requisito	Recursos Didáticos
Curso X	C1	-	R1.1, R1.2
	C2	C1	R2.1, R2.2
	C3	C1	R3.1, R3.2
	C4	C1	R4.1, R4.2
	C5	C2, C3, C4	R5.1, R5.2

Dado o curso exibido na tabela 3.8, foram estabelecidas as seguintes cores para os *links* anotados, conforme exibido na tabela 3.9 abaixo:

Tabela 3.9: Exemplo de Cores para os *Links* Anotados

Sigla	Cor
EL-1	Verde escuro
EL-2	Vermelho
EL-3	Verde claro

EL-4	Laranja
EL-5	Cinza claro
EL-6	Cinza escuro
EL-7	Roxo
EL-8	Azul claro
EL-9	Azul escuro

Dado o curso da tabela 3.8 e as cores utilizadas para anotar os *links*, exibidas na tabela 3.9, a tabela 3.10 mostra, para a estratégia pedagógica diretiva, um subconjunto do modelo do aluno, em uma determinada sessão de uso do sistema, as cores com as quais os *links* para conceitos e recursos didáticos são anotados e o comportamento de tais *links* (funcional ou não funcional):

Tabela 3.10: Exemplo de Modelo do Aluno X Anotação e Comportamento dos *Links*

Conceito	Recurso Didático	Situação no Modelo do Aluno	Sigla / Cor do Link	Comportamento do Link
C1		Conceito satisfatoriamente aprendido	EL-5 Cinza claro	Funcional
	R.1.1	Recurso didático visitado	EL-8 Azul claro	Funcional
	R1.2	Recurso didático visitado	EL-8 Azul claro	Funcional
C2		Conceito analisado	EL-3 Verde claro	Funcional
	R.2.1	Recurso didático visitado	EL-8 Azul claro	Funcional
	R2.2	Recurso didático não visitado	EL-9 Azul escuro	Funcional
C3		Conceito desconhecido	EL-1 Verde escuro	Funcional
	R.3.1	Recurso didático não visitado	EL-9 Azul escuro	Funcional
	R3.2	Recurso didático não visitado	EL-9 Azul escuro	Não funcional
C4		Conceito não satisfatoriamente aprendido	EL-7 Roxo	Funcional
	R.4.1	Recurso didático visitado	EL-8	Funcional

			Azul claro	
	R4.2	Recurso didático visitado	EL-8 Azul claro	Funcional
C5		Conceito desconhecido	EL-2 Vermelho	Não funcional
	R.5.1	Recurso didático não visitado	EL-9 Azul escuro	Não funcional
	R5.2	Recurso didático não visitado	EL-9 Azul escuro	Não funcional

3.5.3.3 Módulo de Gerenciamento da Autoria

O módulo de gerenciamento da autoria é responsável pela manutenção do modelo do domínio e dos objetos de aprendizagem. Os principais processos relacionados à autoria são:

- Transferência de objetos de aprendizagem para o sistema;
- Criação de questões de avaliação;
- Criação de avaliações;
- Criação de textos;
- Criação de cursos.

O módulo de autoria será descrito em detalhes no próximo capítulo.

3.5.3.4 Módulo de Acompanhamento do Desempenho do Aluno

O módulo de acompanhamento do desempenho do aluno é extremamente importante para o professor. É o módulo responsável pela extração de informações obtidas através do cruzamento de dados oriundos do modelo do domínio e do modelo do aluno. Estas informações possibilitam que o professor acompanhe o desempenho e oriente os alunos sob sua responsabilidade. Os principais processos relacionados ao acompanhamento do desempenho do aluno são:

- Geração de relatórios e /ou gráficos que apontem o progresso ou desempenho dos alunos;
- Orientação dos alunos, fornecendo-se *feedback* individual aos mesmos.

3.5.3.5 Módulo de Gerenciamento da Administração

O módulo de gerenciamento da administração é responsável pela manutenção de informações administrativas do sistema e das instituições mantidas pelo sistema.

Os principais processos relacionados à administração do sistema são:

- Cadastro das instituições;
- Aprovação / desaprovação do cadastro de administradores de instituição;
- Ativação / desativação de uma instituição.

Os principais processos relacionados à administração da instituição são:

- Aprovação / desaprovação do cadastro de autores, professores e alunos de uma instituição;
- Cadastro das áreas (por exemplo, Ciência da Computação) disponibilizadas pela instituição;
- Cadastro das disciplinas (por exemplo, Matemática Discreta) ofertadas pela instituição;
- Associação de disciplinas a áreas (associação, por exemplo, da disciplina de Matemática Discreta à área de Ciência da Computação e Engenharia Elétrica);
- Formação de turmas em uma disciplina, de uma determinada área, em um dado período e associação das turmas a um ou mais professores (formação por exemplo de uma turma da disciplina de Matemática Discreta, da área de Ciência da Computação, no primeiro semestre de 2007, cujo professor responsável é o professor João da Silva).

3.6 Considerações Finais

Neste capítulo, buscou-se apresentar, em nível conceitual, as principais idéias por trás do desenvolvimento do *AdaptHA*. Primeiramente, este trabalho foi contextualizado dentro da pesquisa em hiperídia educacional adaptativa. Neste contexto, o *AdaptHA* foi caracterizado como um SHA educacional de terceira geração, cujo segmento de pesquisa atua no sentido de promover a educação prática na *Web* e competir com os LMSs atuais. O Projeto Instrucional foi discutido em seguida. Foram apresentadas as categorias de capacidades humanas com as quais o *AdaptHA* trabalha e como se processa o planejamento da educação (através dos eventos de instrução). As categorias de capacidades humanas são importantes porque é a partir do que se espera que os alunos sejam *capazes* de fazer após o processo de aprendizagem que é possível definir os objetivos instrucionais e planejar a educação. Após o Projeto Instrucional foi discutido o modelo Hyper-Automaton, no qual este trabalho se baseia, e por fim foi apresentada em detalhes a arquitetura do *AdaptHA*.

O próximo capítulo também apresenta o *AdaptHA*, porém sob uma perspectiva de mais baixo nível do que a apresentada neste capítulo. Um novo desenho de arquitetura é examinado, destacando as tecnologias que foram utilizadas ao longo do desenvolvimento e o ambiente no qual o sistema é executado. Também são discutidos em detalhes os mecanismos de internacionalização e localização e o ambiente de autoria do sistema.

4 O SISTEMA IMPLEMENTADO

A seção 4.1 deste capítulo apresenta alguns aspectos tecnológicos envolvidos no desenvolvimento do *AdaptHA*. Em particular, são discutidos os *frameworks* utilizados, nos quais a arquitetura do *AdaptHA* se sustenta, e é apresentado o ambiente em que o sistema é executado. A seção 4.2 apresenta em detalhes o ambiente de autoria do *AdaptHA*. Por fim, a seção 4.3 apresenta as considerações finais do capítulo.

4.1 Frameworks Utilizados pelo *AdaptHA*

Um *framework* é uma aplicação reutilizável e semicompleta que pode ser especializada para produzir aplicações customizadas (JOHNSON e FOOTE, 1988). Customiza-se um *framework*, para uma aplicação específica, através da criação de classes específicas para a aplicação, sendo que tais classes específicas são subclasses de classes abstratas do *framework* (GAMMA et al., 2000).

Segundo Gamma et al. (2000), a utilização de um *framework* ao desenvolver uma aplicação, acaba por definir a arquitetura desta aplicação. Um *framework* predefine parâmetros de projeto, de maneira que o desenvolvedor pode se concentrar nos aspectos específicos de sua aplicação. *Frameworks* capturam decisões de projeto que são comuns a seu domínio de aplicação e, desta forma, enfatizam a *reutilização de projetos* ao invés da *reutilização de código*. A reutilização de projetos leva a uma inversão de controle entre a aplicação e o *software* no qual ela se baseia. Quando uma API (ou *toolkit*) é utilizada, é escrito o corpo principal da aplicação e invocado o código a ser reutilizado. Em se tratando de um *framework*, o corpo principal é reutilizado e é escrito o código que este corpo principal deve invocar. Como resultado, tem-se que é possível construir aplicações mais rapidamente, assim como as aplicações desenvolvidas utilizando um mesmo *framework* têm estruturas similares.

De acordo com Gamma et al. (2000), os *frameworks* estão se tornando cada vez mais comuns e importantes. Eles são a maneira pela qual os sistemas orientados a objetos conseguem a maior reutilização. Neste sentido, grande parte da arquitetura do *AdaptHA* é sustentada em *frameworks*. Os *frameworks* Java utilizados pelo *AdaptHA* são descritos nas subseções abaixo.

4.1.1 Java2 Enterprise Edition (J2EE)

Segundo Matena et al. (2007), atualmente as empresas precisam aumentar seu alcance, reduzir seus custos e diminuir os tempos de resposta de seus serviços a clientes, colaboradores e fornecedores. Em geral, aplicações que provêm estes serviços devem

combinar *Enterprise Information Systems* (EISs) com novas funções de negócio que entregam serviços a um número mais abrangente de usuários. Esses serviços precisam ser:

- *Altamente disponíveis*, a fim de atender as necessidades de um ambiente globalizado de negócio;
- *Seguros*, a fim de proteger a privacidade de usuários e a integridade da empresa;
- *Confiáveis e escaláveis*, para assegurar que as transações de negócio sejam acuradas e prontamente processadas.

Com o propósito de reduzir custos e o tempo de projeto e desenvolvimento da aplicação, a plataforma J2EE provê uma abordagem baseada em componentes para projetar, desenvolver, montar e distribuir aplicações. A plataforma J2EE oferece um modelo de aplicação multicamada distribuída, componentes reutilizáveis, um modelo de segurança unificado, controle de transação flexível, e o suporte a *Web services* através da troca integrada de dados utilizando padrões e protocolos abertos baseados em XML (J2EE 1.4 TUTORIAL, 2007).

Uma das grandes vantagens em se utilizar J2EE para o desenvolvimento e *deployment* de aplicações reside no fato de que J2EE é definida por uma especificação (MATENA et al., 2007) e desta forma, os fabricantes de *software*, caso queiram disponibilizar a plataforma J2EE, devem implementar a especificação J2EE. Isso significa que as aplicações desenvolvidas com J2EE são independentes de fabricante. Com isso, tem-se a liberdade de escolher os produtos e componentes que melhor atendem ao negócio e aos requisitos tecnológicos.

Em geral, os fabricantes de *software* denominam a implementação da especificação J2EE de *servidor de aplicação J2EE*. O servidor de aplicação J2EE utilizado pelo *AdaptHA* é o JBOSS Application Server (JBOSS APPLICATION SERVER, 2007).

Abaixo são apresentados os componentes e *containers* J2EE.

4.1.1.1 Componentes J2EE

A lógica de uma aplicação J2EE é dividida entre diferentes componentes de acordo com as funções que eles desempenham, e os diversos componentes que formam uma aplicação J2EE são instalados em diferentes máquinas. A máquina na qual um componente é instalado depende da camada, no ambiente multicamada J2EE, à qual o componente pertence. A figura 4.1 exibe dois modelos de aplicações J2EE multicamada. O *AdaptHA* utiliza um modelo de aplicação multicamada idêntico ao da Aplicação 2, exibida na figura 4.1.

A especificação J2EE define os seguintes componentes (J2EE 1.4 TUTORIAL, 2007):

- *Clientes*: um cliente J2EE pode ser:
 - *Cliente Web*: consiste em um conjunto de páginas *Web* dinâmicas geradas pelos componentes *Web* (na camada *Web*) e um navegador, que renderiza as páginas recebidas do servidor;
 - *Applet*: uma página *Web* recebida pela camada *Web* pode incluir um *applet* embutido. Sistemas cliente precisam possuir o *plugin* Java e possivelmente um arquivo de política de segurança para que o *applet*

execute com sucesso no navegador. Desta forma, componentes *Web* constituem a API preferida para a criação de um programa cliente *Web* porque não necessitam de *plugins* ou arquivos de política de segurança nos sistemas clientes;

- *Aplicação cliente*: uma aplicação cliente executa em uma máquina cliente e provê uma maneira dos usuários manipularem tarefas que requerem uma IU mais rica do que a que pode ser provida por uma linguagem de marcação. Aplicações clientes acessam diretamente *enterprise beans* rodando na camada de negócio.

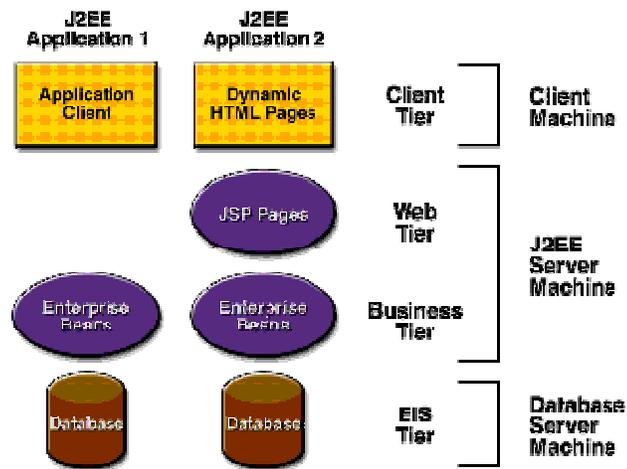


Figura 4.1: Aplicações Multicamada (J2EE 1.4 TUTORIAL, 2007)

- *Componentes Web*: são *servlets* e páginas criadas utilizando a tecnologia JSP. Páginas HTML estáticas e *applets* são empacotados com componentes *Web*, mas não são considerados componentes *Web* pela especificação J2EE;
- *Componentes de negócio*: código de negócio, que é a lógica que soluciona ou atende as necessidades de um domínio de negócio em particular, é manipulado por *enterprise beans* que rodam na camada de negócio. Há três tipos de *enterprise beans*:
 - *Session bean*: representa uma conversação transiente com um cliente. Quando o cliente termina de executar, o *session bean* e seus dados são descartados;
 - *Entity bean*: representa dados persistentes armazenados em uma linha de uma tabela do banco de dados. Se o cliente termina de executar ou se o servidor é desligado, os serviços base asseguram que os dados do *entity bean* serão salvos;
 - *Message-driven bean*: combina características de um *session bean* e de um *message listener* JMS, permitindo que um componente de negócio receba mensagens JMS assincronamente.

O *AdaptHA* encapsula sua lógica de negócio em *session beans* e não faz uso de *entity beans* e *message driven-beans*. No lugar de *entity beans*, o *AdaptHA* utiliza o *framework* de persistência Hibernate (discutido na subseção 4.1.2).

4.1.1.2 Containers J2EE

Em geral, as aplicações multicamada que utilizam um *thin client* são difíceis de escrever. Estas aplicações envolvem muitas linhas de código complexo para manipular transações, gerenciamento de estado, *multithreading*, *pooling* de recursos e outros detalhes complexos de baixo nível. A arquitetura J2EE baseada em componentes e independente de plataforma torna as aplicações J2EE fáceis de escrever porque a lógica de negócio é organizada em componentes reutilizáveis. Além disso, o servidor J2EE provê serviços de base, na forma de um *container*, para cada tipo de componente. Os *containers* constituem a interface entre um componente e uma funcionalidade de baixo nível que suporta o componente. A figura 4.2 ilustra esta idéia.

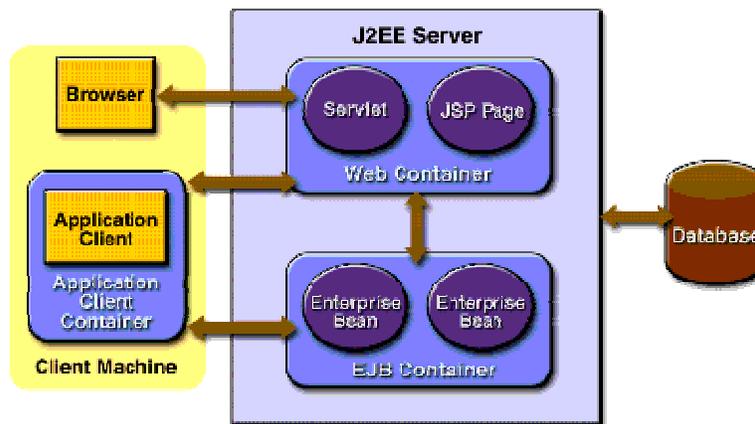


Figura 4.2: Servidor J2EE e Containers (J2EE 1.4 TUTORIAL, 2007)

4.1.2 Hibernate

Persistência é um dos conceitos fundamentais no desenvolvimento de aplicações. A persistência dos dados informados pelos usuários de uma dada aplicação faz com que estes dados sobrevivam, mesmo que a máquina em que a aplicação está rodando seja desligada. Em uma aplicação orientada a objetos, a persistência torna possível que um objeto sobreviva ao processo que o criou. Segundo Bauer e King (2005), a proposta do Hibernate é ser uma solução completa para o problema do gerenciamento de dados persistentes em Java. O Hibernate é responsável por intermediar a interação da aplicação com um banco de dados relacional, deixando o desenvolvedor livre para se concentrar nos problemas pertinentes ao negócio.

Hibernate é uma implementação *open source* de uma solução de persistência em Java denominada *Object / Relational Mapping* (ORM). O mapeamento objeto / relacional consiste na persistência automática (e transparente) de objetos em uma aplicação Java para tabelas em um banco de dados relacional, utilizando metadados que descrevem o mapeamento dos objetos para o banco de dados. ORM, em essência, trabalha transformando dados de uma representação para outra (BAUER e KING, 2005).

Segundo Bauer e King (2005), o Hibernate é útil para aplicações orientadas a objetos que trabalham com um *modelo do domínio*. Uma aplicação que possui um modelo do domínio não trabalha diretamente com a representação tabular de entidades de negócio. A aplicação tem seu próprio modelo orientado a objetos de entidades de negócio. Se o banco de dados possui as tabelas ITEM e LANCE, a aplicação Java

define as classes Item e Lance. Ou seja, ao invés de trabalhar diretamente com linhas e colunas de um conjunto de resultados de uma SQL, a lógica de negócio interage com o modelo do domínio orientado a objetos, cuja concretização em tempo de execução é um grafo de objetos interconectados. A lógica de negócio nunca é executada no banco de dados (como uma *stored procedure*, por exemplo), ela é implementada em Java. Isto torna possível que a lógica de negócio use conceitos sofisticados da orientação a objetos, como herança e polimorfismo.

O *AdaptHA* mantém um modelo do domínio de entidades de negócio e utiliza o Hibernate como *framework* de persistência.

A utilização do Hibernate apresenta muitas vantagens, algumas delas são listadas abaixo (BAUER e KING, 2005):

- *Produtividade*: código relacionado a persistência talvez seja o código mais entediante em uma aplicação Java. Hibernate elimina muito deste trabalho e permite que o desenvolvedor se concentre no problema do negócio;
- *Manutenibilidade*: pelo fato do Hibernate diminuir a quantidade de código a ser escrito em uma aplicação, ele torna a aplicação mais manutenível. No entanto, há outras razões que tornam a aplicação Hibernate mais manutenível. Em sistemas com persistência *hand-coded* (codificada a “mão”), uma inevitável tensão existe entre a representação relacional e o modelo de objetos que implementa o domínio. Mudanças na representação relacional quase sempre envolvem mudanças no modelo de objetos, e vice-versa. ORM provê uma separação entre os dois modelos, permitindo um uso mais elegante da orientação a objetos no lado Java e isolando cada modelo de pequenas mudanças um no outro;
- *Performance*: uma crítica comum à utilização da persistência automática é que o código de persistência *hand-coded* pode sempre ser tão rápido quanto a persistência automática, e pode frequentemente ser mais rápido do que a persistência automática. Dada uma tarefa de persistência, muitas otimizações são possíveis. Algumas, tais como *hints* de consulta, são muito mais fáceis de atingir com código SQL / JDBC *hand-coded*. No entanto, a maioria das otimizações é mais fácil de ser atingida com ORM automatizado. A persistência automática acaba por melhorar a produtividade do desenvolvedor tal que ele pode gastar algum tempo para otimizar manualmente alguns gargalos remanescentes que a persistência automática não conseguiu otimizar;
- *Independência de fabricante*: ORM abstrai a aplicação do banco de dados SQL e do dialeto SQL. O desenvolvedor não deve necessariamente esperar escrever uma vez e rodar em qualquer lugar (*write once / run anywhere*), pois as características e capacidades dos bancos de dados diferem e atingir total portabilidade requereria sacrificar recursos poderosos de poderosas plataformas de banco de dados. Todavia, é normalmente mais fácil desenvolver uma aplicação independente de plataforma usando ORM. Mesmo que a independência de plataforma não seja requerida, ORM pode ajudar a mitigar alguns dos riscos associados a ficar preso a um fabricante de banco de dados.

4.1.3 JavaServer Faces (JSF)

JSF é um *framework* de componentes de IU do lado do servidor, apropriado para o desenvolvimento de aplicações *Web* baseadas na tecnologia Java (J2EE 1.4 TUTORIAL, 2007). JSF é parte integrante da plataforma J2EE. JSF é uma especificação Java, e a implementação de JSF que o *AdaptHA* utiliza é a implementação da Apache, denominada MyFaces (MYFACES, 2007).

Segundo Mann (2005), um conceito bastante popular na era anterior à *Web* era o conceito de *Rapid Application Development* (RAD). O principal objetivo do RAD era possibilitar a construção de poderosas aplicações com um conjunto de componentes reutilizáveis. Mesmo um desenvolvedor não tão experiente era capaz de desenvolver IUs complexas e integrá-las com fontes de dados. Era possível arrastar componentes de uma paleta e soltá-los dentro de uma aplicação. Cada um destes componentes tinha propriedades que afetavam seu comportamento. Estes componentes gravavam um conjunto de eventos, e manipuladores de eventos definiam a interação entre a IU e o restante da aplicação. O desenvolvedor tinha acesso a tudo isto diretamente de um ambiente integrado de desenvolvimento (IDE), e neste ambiente era possível trocar facilmente entre a visão centrada no projeto e a visão centrada no código. Estas ferramentas tinham tipicamente quatro camadas:

- Uma arquitetura base de componentes;
- Um conjunto de componentes padrão;
- Uma infra-estrutura de aplicação;
- O ambiente propriamente dito.

Segundo Mann (2005), a adoção do conceito de RAD no mundo do desenvolvimento *Web* tem sido marcadamente lenta. A lentidão se deve em parte à complexidade de criar uma visão simples e coesa do desenvolvimento de aplicações *Web* em um mundo que não é simples e muito menos coeso. Aplicações *Web* são complexas se comparadas com aplicações de *desktop*. Há inúmeros diferentes recursos para gerenciar, como páginas, arquivos de configuração, imagens e código. Os usuários podem estar usando diferentes tipos de navegadores, que por sua vez podem estar rodando em diferentes sistemas operacionais. O desenvolvedor ainda tem que lidar com o HTTP, um protocolo que não foi projetado para suportar aplicações complexas.

Nos últimos anos começaram a surgir soluções RAD para o desenvolvimento de aplicações *Web*. Estas soluções trazem o poder do desenvolvimento visual e orientado a componentes para o complexo mundo do desenvolvimento *Web*. No mundo Java, muitos *frameworks* surgiram, muitos deles *open source*, como, por exemplo, o Struts (STRUTS, 2007). Alguns suportados por ferramentas, outros não. No entanto, ainda faltava um *framework Web* RAD padrão para Java e assim sendo, foi desenvolvido JSF com o propósito específico de preencher esta lacuna.

Considerando-se as quatro camadas de uma ferramenta RAD, JSF define três delas: uma arquitetura de componente, um conjunto padrão de componentes da IU e uma infra-estrutura de aplicação. A arquitetura de componente do JSF define uma maneira comum de construir componentes da IU. Os componentes são orientados a evento, e desta forma, JSF possibilita que eventos gerados no cliente sejam processados no servidor. Exemplos de eventos gerados no cliente são: a mudança do valor de um campo de texto ou o clique em um botão.

Há uma grande diferença entre o JSF e os *frameworks* de IU como Swing ou SWT. A diferença é que o JSF executa no *servidor*. Uma aplicação JSF executa em um container *Web* Java padrão como o Apache Tomcat (APACHE TOMCAT, 2007) e exibe HTML ou alguma outra marcação para o cliente. O *AdaptHA* utiliza o *container Web* distribuído com o servidor de aplicação JBoss, que coincidentemente é o Apache Tomcat.

Se um botão é clicado em uma aplicação Swing, um evento é disparado e o código que irá manipular este evento pode residir no próprio *desktop*. De maneira contrária, os navegadores *Web* nada sabem a respeito dos componentes JSF e de eventos, eles apenas sabem exibir HTML. Assim sendo, quando um botão é clicado em uma aplicação JSF, uma solicitação é enviada do navegador *Web* para o servidor. O JSF é responsável por traduzir a requisição em um evento que possa ser processado pela lógica da aplicação no servidor. O JSF também é responsável por garantir que cada componente de IU definido no servidor seja exibido apropriadamente no navegador. A figura 4.3 exibe uma visão de alto nível de uma aplicação JSF. É possível notar que a aplicação JSF executa no servidor e pode ser integrada com outros subsistemas, tais como EJBs e bancos de dados, como é o caso do *AdaptHA*.

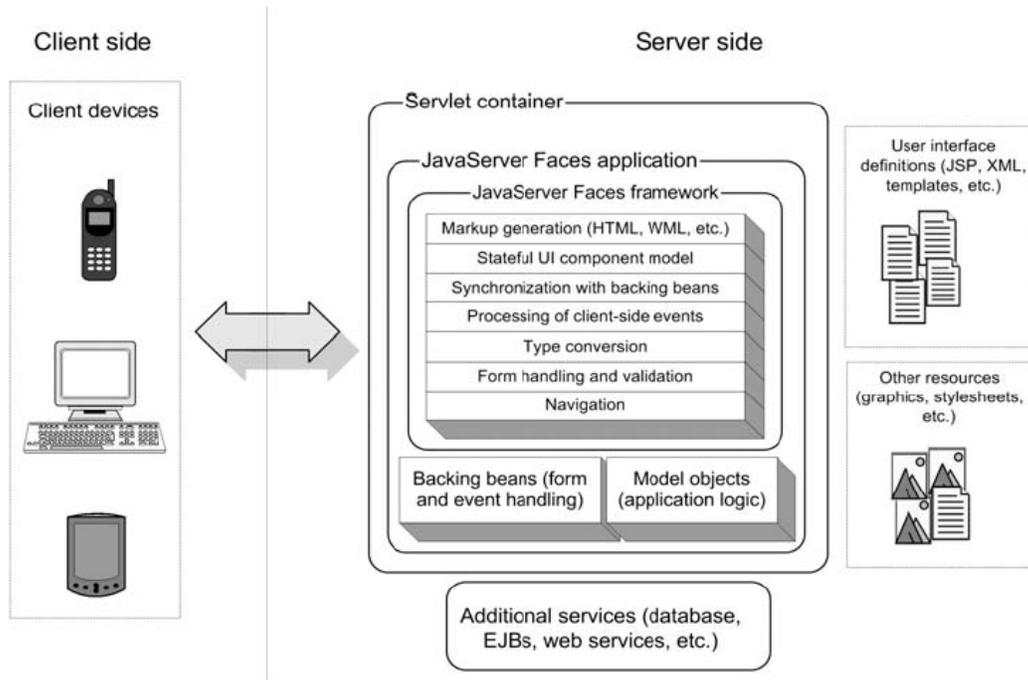


Figura 4.3: Visão geral de uma aplicação JSF (MANN, 2005)

Abaixo são apresentadas algumas das principais características do JSF.

4.1.3.1 Renderizadores

Pelo fato das aplicações *Web*, diferentemente das aplicações *desktop*, terem que freqüentemente atender a múltiplos clientes (como navegadores *Web*, telefones celulares e PDAs), JSF possui uma poderosa arquitetura para exibir componentes de IU de diferentes maneiras.

Antes que os componentes de IU possam ser exibidos para o cliente, eles devem ser renderizados. Os componentes de IU do JSF não são sempre responsáveis por sua

renderização. Quando os componentes são responsáveis por sua renderização eles utilizam um modelo denominado *implementação direta*, mas JSF também suporta o modelo denominado *implementação delegada*. Este modelo permite separar do componente de IU as classes que irão manipular o processo de renderização. Estas classes são denominadas *renderizadores* e são organizadas em *kits* de renderização (*render kits*), que normalmente se focalizam em um tipo específico de saída. JSF já vem equipado com um *kit* de renderização padrão para HTML 4.01, mas podem ser desenvolvidos *kits* de renderização para gerar, por exemplo, um *look and feel* (ou *skin*) diferente de HTML, ou gerar WML, SVG, etc.

4.1.3.2 Validadores

JSF tem facilidades para validar a entrada de dados dos usuários. Isto é feito através dos chamados *validadores*. JSF possui um conjunto de validadores já implementados que o desenvolvedor pode utilizar. Há validadores para verificar se um campo obrigatório foi informado, para verificar se um campo possui o comprimento mínimo exigido, para verificar se o comprimento máximo de um campo não foi ultrapassado, etc. JSF também possibilita que o desenvolvedor implemente seus próprios validadores.

4.1.3.3 Conversores

Elementos fundamentais na arquitetura JSF são os *conversores*. Os conversores são responsáveis por traduzir um objeto para uma *string* a ser exibida e fazer o caminho contrário, transformando uma *string* de entrada para um objeto. É necessário lembrar que toda a entrada de dados feita por um usuário em uma página *Web*, por exemplo, é transmitida para o servidor como *string*. No entanto, as classes no servidor, que irão manipular esses dados, irão trabalhar com *strings*, números, datas, e outros tipos de dados.

Sem os conversores, o desenvolvedor teria que necessariamente escrever o código de conversão das *strings* recebidas do cliente para os tipos de dados adequados e escrever também o código de conversão dos tipos de dados para as *strings* que devem ser apresentadas ao usuário. JSF possui conversores para tipos comuns como datas e números e para utilizá-los, basta associá-los a qualquer componente. O desenvolvedor também pode implementar seus próprios conversores.

Os conversores também são responsáveis pela localização de números, datas e outros objetos que devem ser apresentados ao usuário.

4.1.3.4 Backing Beans

JSF é capaz de manter automaticamente seus componentes de IU em sincronia com as classes Java que coletam os dados informados pelo usuário e respondem a eventos. Estas classes são denominadas *backing beans*. Declarativamente é possível associar um componente de IU a uma variável de instância de um *backing bean*. Feita esta associação, o JSF se encarrega de atualizar automaticamente a variável de instância de acordo com o valor que o usuário informa para o componente. Esta atualização automática poupa um grande esforço de programação por parte do desenvolvedor.

4.1.3.5 Navegação

JSF implementa um elegante sistema de navegação. O *gerenciador de navegação* é responsável por decidir que página carregar com base na saída lógica de métodos

especiais presentes em *backing beans*, denominados métodos de ação. Para cada página, uma *regra de navegação* (*navigation rule*) define que saídas são compreendidas e quais páginas carregar com base nessas saídas. Cada mapeamento específico entre uma saída e uma página é denominado *caso de navegação* (*navigation case*). As regras são definidas em um arquivo de configuração do JSF, normalmente nomeado *faces-config.xml*. A figura 4.4 exibe um trecho do *faces-config.xml* do *AdaptHA*.

```
<navigation-rule>
  <description>Regras de navegacao para /pages/logged/Autor/*</description>
  <from-view-id>/pages/logged/Autor/*</from-view-id>
  <navigation-case>
    <from-outcome>novoConceito</from-outcome>
    <to-view-id>/pages/logged/Autor/novoConceito.jsf</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>recurso</from-outcome>
    <to-view-id>/pages/logged/Autor/recurso.jsf</to-view-id>
  </navigation-case>
</navigation-rule>
```

Figura 4.4: Trecho do *faces-config.xml* do *AdaptHA*

4.1.3.6 Internacionalização e Localização

JSF possui um rico suporte à internacionalização e localização. O MyFaces, e também a implementação de referência do JSF (JAVASERVER FACES, 2007), provê mensagens localizadas para erros de validação e conversão. A figura 4.5 exibe uma pequena porção de uma tela do *AdaptHA* que apresenta para o usuário mensagens de validação (em vermelho) localizadas para português e inglês.



Figura 4.5: Mensagens de validação localizadas no *AdaptHA*

Caso o suporte à internacionalização e localização necessite ser estendido a outros tipos de *strings* apresentadas ao usuário (por exemplo, as *strings* apresentadas na figura 4.5 que não estão em vermelho), então os passos abaixo precisam ser executados:

- Informar ao JSF quais localidades a aplicação deve suportar. Esta configuração deve ser feita no *faces-config.xml*, conforme exibido na figura 4.6-a;
- Criar *resource bundles* para cada uma das localidades que a aplicação deve suportar. *Resource bundles* são arquivos de propriedade com pares chave / valor, como o exibido na figura 4.6-b;

- Referenciar as *strings* localizadas utilizando expressões especiais do JSF em *tags* JSF, conforme exibido na figura 4.6-c.

a) Configuração de localidades no faces-config.xml

```
<application>
  <view-handler>org.apache.myfaces.tomahawk.application..
  <locale-config>
    <default-locale>pt_BR</default-locale>
    <supported-locale>pt_BR</supported-locale>
    <supported-locale>en</supported-locale>
  </locale-config>
  <message-bundle>resources.application</message-bundle>
</application>
```

b) Resource bundles em português e inglês

```
qualquer=qualquer           qualquer=any
questoes=Questões           questoes=Questions
recursos=Recursos           recursos=Resources
repitaSenha=Repita a Senha  repitaSenha=Re-type Password
```

c) Referência a strings localizadas no código

```
<h:outputText value="#{bundle.repitaSenha}:"/>
```

Figura 4.6: Mecanismos de internacionalização e localização do *AdaptHA*

Segundo Mann (2005), a localidade corrente do usuário é determinada com base nas configurações de localidade da aplicação cliente e também com base nas localidades suportadas pela aplicação. Navegadores *Web* enviam um cabeçalho HTTP que especifica os idiomas que eles suportam. Por exemplo, se o idioma principal do navegador é inglês, e a aplicação JSF suporta português como padrão e inglês, então o texto aparecerá em inglês. Mas se a aplicação JSF não suporta inglês, então o texto aparecerá em português.

4.2 O Ambiente de Autoria do *AdaptHA*

Para ter acesso ao ambiente de autoria do *AdaptHA*, é necessário que o usuário tenha se cadastrado no sistema como *autor* (figura 4.7), seu cadastro tenha sido aprovado pelo administrador (figura 4.8) e que tenha efetuado procedimento de *login* (figura 4.9). Feito isto, o autor deve clicar na guia *Autor* (figura 4.10) e lhe é apresentada uma interface como a exibida na figura 4.11.

Cadastro	
Nome:	<input type="text" value="Graciela Lima"/>
E-mail:	<input type="text" value="gcblima@inf.ufrgs.br"/>
Data de Nascimento:	<input type="text" value="11/05/1977"/> dd/mm/aaaa
Sexo:	Feminino <input type="button" value="v"/>
Papel(éis):	<input type="button" value="Aluno"/> <input type="button" value="Autor"/>
Usuário	<input type="text" value="gracielaufgrs"/>
Senha:	<input type="password" value="....."/>
Repita a Senha:	<input type="password" value="....."/>
Instituição:	<input type="text" value="Sem vínculo com instituição"/> <input type="button" value="v"/>
	<input type="text" value="Sem vínculo com instituição"/> <input type="text" value="Universidade Federal do Rio Grande do Sul"/>

[Voltar](#)

Figura 4.7: Interface de cadastramento de usuários do *AdaptHA*

Administrador da Instituição Aluno Autor Sair				
Gerenciar Autores	Novo			
Gerenciar Alunos	Autores			
Gerenciar Áreas	Nome	Usuário	E-mail	Habilitado
Gerenciar Disciplinas	Graciela Lima	graciela.lima@ufrgs	gcblima@inf.ufrgs.br	<input checked="" type="checkbox"/>
	Renata Zanella	renataz	renataz@inf.ufrgs.br	<input type="checkbox"/>
	Habilitar/Desabilitar Autor			

Figura 4.8: Interface de gerenciamento do cadastro de autores do *AdaptHA*

Usuários Cadastrados

Usuário

Senha

[Esqueceu a Senha?](#)

[Cadastre-se](#)

[Pesquisar](#)

Figura 4.9: Interface de *login* do *AdaptHA*

Aluno Autor Sair

Seja Bem Vindo!!!

Para navegar nos cursos acesse a guia **Aluno**.

Para a autoria de cursos acesse a guia **Autor**.

Figura 4.10: Interface de boas vindas do *AdaptHA*

Cursos

[Novo Curso](#)

[Meus Cursos](#)

Linguagens Formais

[Novo Conceito](#)

Linguagens Regulares

[Visão Geral](#)

[Avaliação diagnóstica](#)

[Objetivos](#)

[Introdução e foco da instrução](#)

[Pré-requisitos](#)

[Elaboração](#)

[Material instrucional](#)

[Orientação da aprendizagem](#)

[Avaliação formativa](#)

[Síntese](#)

[Sumário](#)

[Avaliação somativa](#)

[Continuação](#)

[Atividades de retenção e transferência](#)

[Convidar Autor\(es\)](#)

[Visualizar Curso](#)

Recursos

[Arquivos](#)

[Upload](#)

[Visualizar](#)

Textos

[Novo Texto](#)

[Visualizar](#)

Questões

[Nova Questão](#)

[Visualizar](#)

Avaliações

[Nova Avaliação](#)

[Visualizar](#)

Figura 4.11: Interface principal de autoria do *AdaptHA*

A interface principal de autoria do *AdaptHA* é dividida em duas grandes porções:

- *Cursos*: nesta porção (localizada na parte superior da figura 4.11), o autor tem acesso a todas as funcionalidades relacionadas à criação e manutenção de cursos. Estas funcionalidades são detalhadas nas subseções 4.2.6 a 4.2.8;
- *Recursos*: nesta porção (localizada na parte inferior da figura 4.11), o autor tem acesso a todas as funcionalidades relacionadas à criação e manutenção de recursos que são utilizados quando da criação dos cursos. Estas funcionalidades são detalhadas nas subseções 4.2.1 a 4.2.5.

4.2.1 Transferindo Objetos de Aprendizagem para o Servidor

O *upload* (transferência) de objetos de aprendizagem para o servidor deve ser feito clicando-se na opção *Upload* (figura 4.11), quando então é apresentada ao autor a interface mostrada na figura 4.12. Vale destacar que o *AdaptHA* mantém os objetos de aprendizagem em local protegido no servidor, evitando desta forma o uso indiscriminado desse material. Os objetos de aprendizagem transferidos para o servidor devem ter no máximo 10 mega bytes e devem estar em um dos formatos de arquivo aceitos pelo sistema (os formatos de arquivos são exibidos na figura 4.12).

Upload

Nome: Hierarquia de Chomsky

Descrição: Esta figura exibe a Hierarquia de Chomsky, ou seja, a hierarquia de linguagens do tipo 0, 1, 2 e 3.

Palavra(s)-chave: Linguagens Formais; Hierarquia de Chomsky

Idioma: Português

Nível de Compartilhamento: Público

Arquivo: C:\Documents and Settings\ [Procurar...]

[Enviar]

Observações a respeito do upload de arquivos
 O tamanho do arquivo não deve exceder 10 mega bytes.
 Os formatos aceitos são:
 Word (DOC) PDF Rich Text Format (RTF) Power Point (PPT) Macromedia Flash (SWF) GIF JPEG PNG Texto (TXT) Vídeo (AVI)

Figura 4.12: Interface de *upload* de objetos de aprendizagem do *AdaptHA*

Para visualizar os objetos de aprendizagem do autor, bem como visualizar os objetos de aprendizagem públicos de outros autores, deve-se clicar na opção *Visualizar*, logo abaixo de *Arquivos*, conforme mostrado na figura 4.11, e é apresentada ao autor a interface exibida na figura 4.13. Os objetos podem ser pesquisados por seu formato (tipo de arquivo) e visualizados clicando-se no *link* em seu nome.

Meus Arquivos							
Nome	Descrição	Tamanho (bytes)	Data de Inclusão	Idioma	Formato	Nível de Compartilhamento	Palavra(s)-chave
Autômato Finito com Saída - Máquina de Mealy	Autômato Finito com Saída	362094.0	14/12/2006		JPEG	Público	Autômato Finito com Saída
Autômato Finito	Autômato Finito	200451.0	14/12/2006		JPEG	Público	Autômato Finito
Hierarquia de Chomsky	Hierarquia de Chomsky	200451.0	14/12/2006		JPEG	Público	Hierarquia de Chomsky

Arquivos Públicos de Outros Autores						
Nome	Descrição	Tamanho (bytes)	Data de Inclusão	Idioma	Formato	Palavra(s)-chave
Eu robô	Eu robô	12750.0	24/03/2007	Português	GIF	robô

Figura 4.13: Interface de visualização de objetos de aprendizagem do *AdaptHA*

4.2.2 Editando Textos com o FCKeditor

Conforme mencionado anteriormente, é fundamental oferecer ao autor ferramentas que facilitem as tarefas de autoria, e neste sentido, um ambiente como o *AdaptHA* deve fornecer um *editor de textos* que possibilite realizar operações elementares de edição de texto. Assim sendo, o *AdaptHA* incorporou a seu projeto o FCKeditor (FCKEDITOR, 2007) (figura 4.14), um poderoso editor de textos gratuito, facilmente configurável, independente de navegador e totalmente escrito em HTML e JavaScript. O fato de o FCKeditor ser escrito em HTML e JavaScript significa em última instância que para executá-lo não se faz necessária a instalação de nenhum componente no navegador do usuário. Outra grande vantagem do FCKeditor é que ele já vem com recursos de internacionalização e localização embutidos.

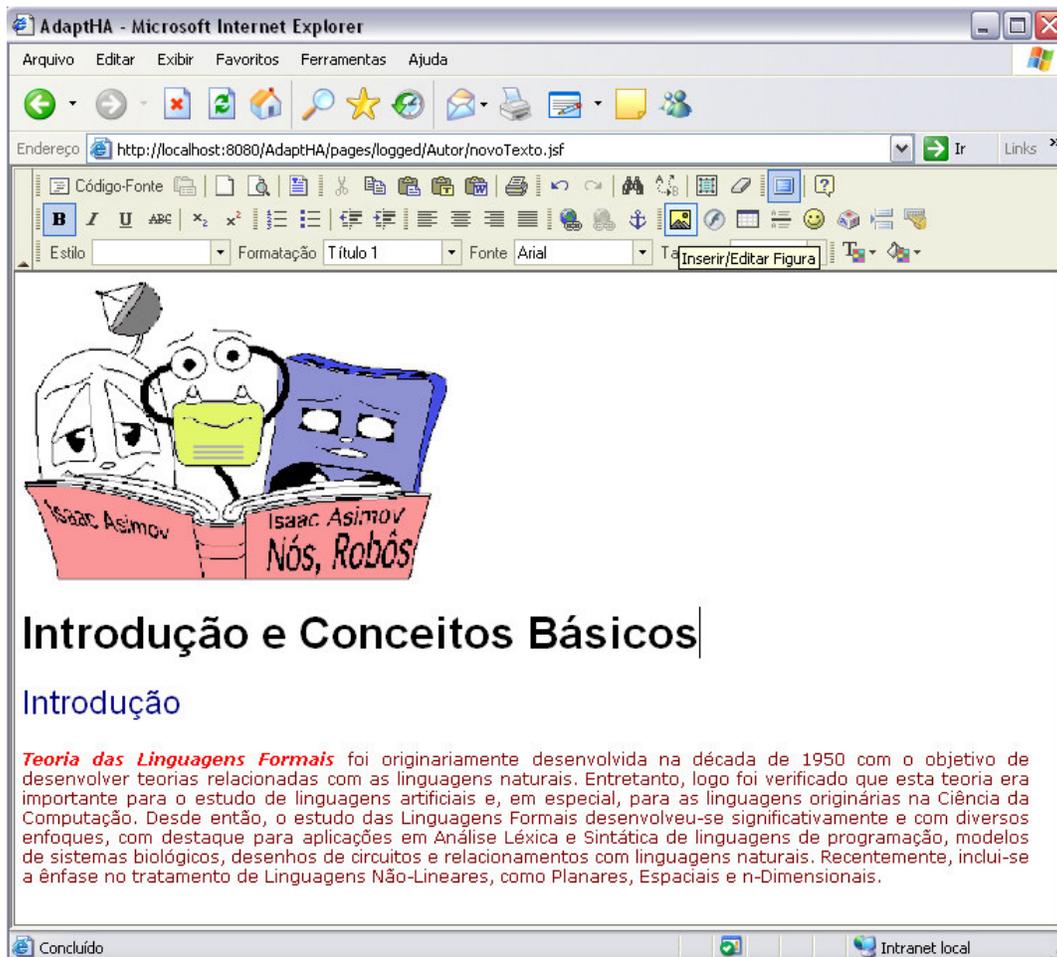


Figura 4.14: FCKeditor, o editor de textos do *AdaptHA*

Para inserir um objeto de aprendizagem no texto, é invocada a interface de visualização de objetos de aprendizagem (figura 4.13), ligeiramente modificada (figura 4.15) para permitir a seleção do objeto de aprendizagem desejado. Para inserir uma figura no texto, por exemplo, basta acessar a janela *Formatar Figura* (figura 4.15), acionada ao se clicar no botão *Inserir / Editar Figura* (figura 4.14), e em seguida clicar no botão *Localizar no Servidor* (figura 4.15) para ter acesso à interface de visualização e seleção de objetos de aprendizagem.

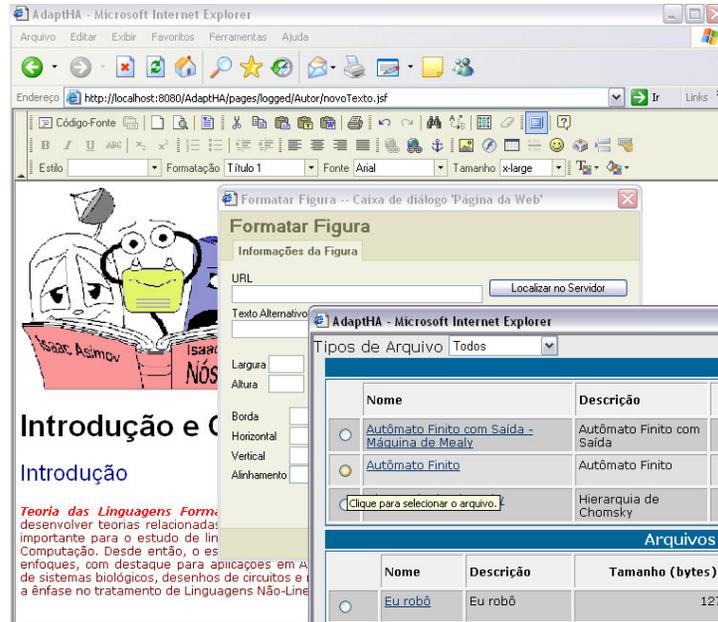


Figura 4.15: Inserindo objetos de aprendizagem no texto através do FCKeditor

4.2.3 Criando Textos

A criação de textos deve ser feita clicando-se na opção *Novo Texto* (figura 4.11), quando então é apresentada ao autor a interface mostrada na figura 4.16. O autor deve informar o título do texto, palavras-chave que devem ser associadas ao texto, idioma em que o texto está disponível e nível de compartilhamento do texto (público ou privado). Em seguida o autor deve inserir o conteúdo propriamente dito do texto. A inserção do conteúdo deve ser feita clicando-se no botão *Adicionar Página de Texto* (figura 4.16). Cada página de texto deve conter no máximo 40.000 caracteres e um número irrestrito de páginas de texto podem ser inclusas.

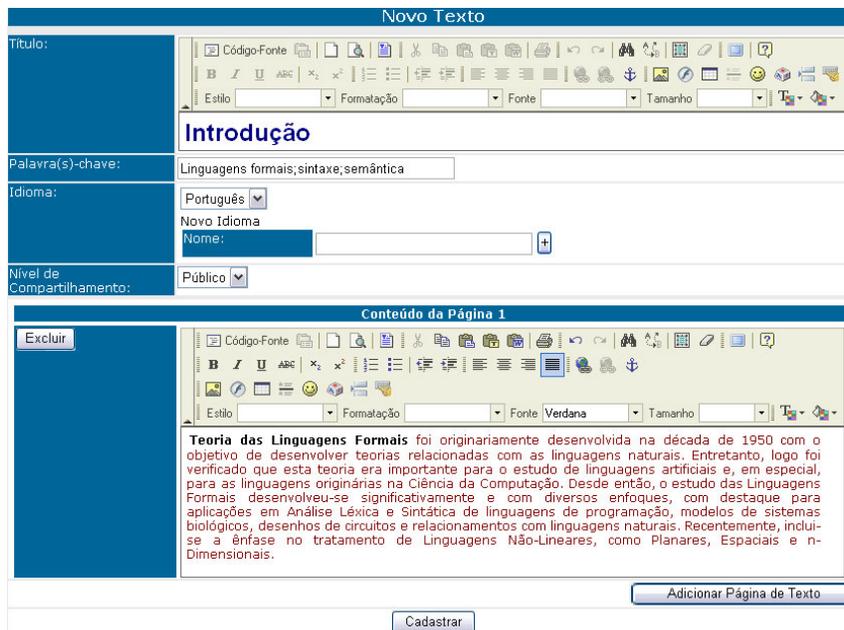


Figura 4.16: Interface de criação de textos do AdaptHA

Para visualizar os textos do autor, bem como visualizar os textos públicos de outros autores, deve-se clicar na opção *Visualizar*, logo abaixo de *Textos*, conforme mostrado na figura 4.11, e é apresentada ao autor a interface exibida na figura 4.17. Os textos podem ser editados clicando no *link Editar Texto* e visualizados clicando-se no *link* em seu título.

Meus Textos					
	Título	Data de Inclusão	Idioma	Nível de Compartilhamento	Palavra(s)-chave
Editar Texto	Introdução	25/03/2007	Português	Público	Linguagens formais semântica sintaxe
Editar Texto	O Ambiente de Autoria do AdapthA	14/12/2006	Português	Privado	Adaptação Ambiente de autoria Educação

Figura 4.17: Interface de visualização de textos do *AdapthA*

Clicando-se no título do texto (figura 4.17) é apresentada ao usuário a interface exibida na figura 4.18. É possível navegar pelas páginas do texto clicando-se nos botões verdes localizados no canto superior direito desta interface.

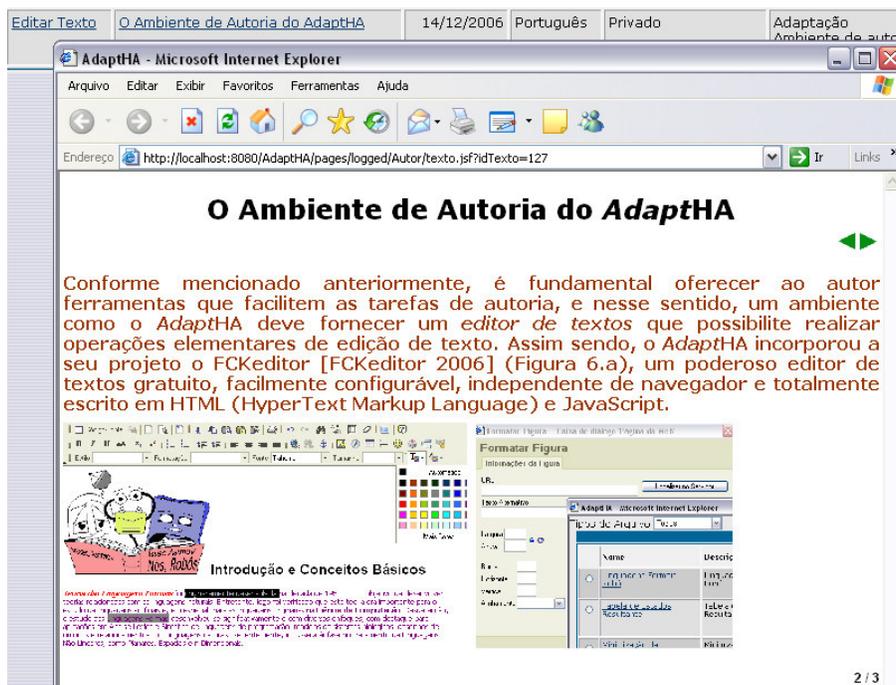


Figura 4.18: Navegando por um texto no *AdapthA*

4.2.4 Criando Questões de Avaliação

A criação de questões deve ser feita clicando-se na opção *Nova Questão* (figura 4.11), quando então é apresentada ao autor a interface mostrada na figura 4.19. O autor deve informar o enunciado da questão, o tipo da questão (que pode ser de múltipla escolha ou verdadeiro / falso), as palavras-chave que devem ser associadas à questão, o nível de dificuldade da questão (que pode ser muito fácil, fácil, dificuldade média, difícil ou muito difícil), o idioma em que a questão está disponível, o nível de compartilhamento da questão (público ou privado), o *feedback* para o aluno que responder corretamente à questão e o *feedback* para o aluno que não responder corretamente à questão.

Uma questão pode conter um número irrestrito de alternativas e para incluí-las basta clicar no botão *Adicionar Alternativa* (figura 4.19). Já para excluir uma alternativa é necessário clicar no botão *Excluir* (figura 4.19). Ao marcar o botão de *rádío* (figura 4.19) a alternativa se torna verdadeira. Quando a questão é de múltipla escolha, permite-se marcar somente uma das alternativas como verdadeira, quando a questão é do tipo verdadeiro / falso esta restrição não é imposta. Para cada alternativa da questão é possível informar a descrição da alternativa bem como o *feedback* a ser apresentado ao aluno caso ele marque a alternativa como verdadeira.

Nova Questão	
Enunciado:	<div style="border: 1px solid black; padding: 5px;"> <p>1B) Sobre AFD, AFN e AFE:</p> <p>I) Um AFE com uma função de transição tal que</p> $\delta(q, \epsilon) = \{q\}$ <p>pode ficar em loop infinito para a entrada vazia, caso o estado q seja não-final.</p> <p>II) Um movimento vazio sempre é um movimento não-determinista.</p> <p>III) O autômato sobre o alfabeto</p> $\{\epsilon, \eta, \mu\}$ <p>ilustrado abaixo é não determinista e aceita a linguagem $(\eta + \mu)^*$:</p> </div>
Tipo de Questão:	Múltipla escolha
Palavra(s)-chave:	Autômatos; Linguagens regulares
Nível de Dificuldade:	Dificuldade média
Idioma:	Português
	Novo Idioma Nome: <input type="text"/> +
Nível de Compartilhamento:	Privado
Feedback:	<div style="border: 1px solid black; padding: 5px;"> <p>Feedback para resposta correta</p> <div style="border: 1px solid black; padding: 2px;"> <p>Parabéns, agora você já pode estudar o conceito de autômatos finitos com saída!</p> </div> <p>Feedback para resposta incorreta</p> <div style="border: 1px solid black; padding: 2px;"> <p>Você deve estudar o conceito de autômatos finitos novamente.</p> </div> </div>
Alternativa 1	
<input type="radio"/> Excluir	<div style="border: 1px solid black; padding: 5px;"> <p>Descrição</p> <div style="border: 1px solid black; padding: 2px;"> <p>Nenhum dos itens.</p> </div> </div>
	<div style="border: 1px solid black; padding: 5px;"> <p>Feedback</p> <div style="border: 1px solid black; padding: 2px;"> <p></p> </div> </div>
<input type="button" value="Adicionar Alternativa"/>	
<input type="button" value="Cadastrar"/>	

Figura 4.19: Interface de criação de questões do *AdaptHA*

- *Exibição das alternativas*: as alternativas podem ser exibidas em seqüência pré-definida (seqüência na qual as alternativas são inclusas na autoria) ou em seqüência aleatória;
- *Feedback das questões*: o *feedback* pode ser exibido a cada questão respondida ou ao final da avaliação (quando as questões são exibidas em uma única página o sistema assume automaticamente a última opção).

Para selecionar as questões que irão compor a avaliação, o autor deve clicar no botão *Selecionar Questões* (figura 4.21). Clicando-se neste botão é apresentada ao autor a interface exibida na figura 4.22. As questões podem ser pesquisadas por tipo, palavras-chave, nível de dificuldade, idioma e nível de compartilhamento. Para que o autor possa incluí-las na avaliação elas devem ser marcadas e em seguida o botão *Selecionar Questões* (figura 4.22) deve ser clicado.

Seleção de Questões		
Tipo de Questão:	Múltipla escolha	
Palavra(s)-chave:	autômatos	
Nível de Dificuldade:	Muito fácil	
Idioma:	Português	
Nível(eis) de Compartilhamento:	Público	
Pesquisar Questões		
Questões		
	Enunciado	Data de Inclusão
<input checked="" type="checkbox"/>	Questão 3	14/12/2006
<input checked="" type="checkbox"/>	Questão 1	23/11/2006
<input type="checkbox"/>	Questão 1	23/11/2006
Selecionar Questões		

Figura 4.22: Interface de seleção de questões de avaliações do *AdaptHA*

Para visualizar as avaliações do autor, bem como visualizar as avaliações públicas de outros autores, deve-se clicar na opção *Visualizar*, logo abaixo de *Avaliações*, conforme mostrado na figura 4.11, e é apresentada ao autor a interface exibida na figura 4.23. Para visualizar a avaliação basta clicar em sua descrição.

Minhas Avaliações				
Descrição	Data de Inclusão	Idioma	Nível de Compartilhamento	Palavra(s)-chave
Esta avaliação tem como objetivo testar os conhecimentos adquiridos pelo aluno com relação ao conceito de Autômatos Finitos Determinísticos.	25/03/2007	Português	Público	autômatos linguagens regulares
Avaliação 1	23/11/2006	Espanhol	Público	avaliação 1

Figura 4.23: Interface de visualização de avaliações do *AdaptHA*

4.2.6 Criando Cursos

A criação de cursos deve ser feita clicando-se na opção *Novo Curso* (figura 4.11), quando então é apresentada ao autor a interface mostrada na figura 4.24.

Novo Curso

Nome do Curso: Linguagens Formais

Descrição: Este curso objetiva...

Disciplina: Linguagens Formais e Autômatos

Área(s): Ciência da Computação, Matemática

Nível de Abrangência: Profundo

Idioma: Português

Desempenho Exigido:

Nível de Conhecimento	Desempenho Exigido(%)
Básico	70
Intermediário	80
Avançado	90

Cadastrar

Figura 4.24: Interface de criação de cursos do AdaptHA

4.2.7 Criando Conceitos para um Curso

Após a criação do curso, o autor deve definir os conceitos que integrarão o curso e como tais conceitos se relacionam. Em essência, nesta etapa, o autor define o *autômato* que representa a estrutura navegacional do curso. Para criar um conceito, basta clicar no *link Novo Conceito* (figura 4.11) do curso desejado. A figura 4.25 exhibe a interface de criação de conceitos do AdaptHA.

Linguagens Formais - Novo Conceito

Nome do Conceito: Linguagens Livres do Contexto

Descrição: A Classe das Linguagens Livres do Contexto ou Tipo 2 contém propriamente a Classe das Linguagens Regulares. Seu estudo é de fundamental importância na informática pois:

Assunto: Linguagens formais

Tópico(s): Linguagens Livres do Contexto

Pré-requisito(s): Linguagens Regulares

Palavra(s)-chave: linguagem; linguagens formais; linguagens livres

Objetivo(s) Educacional (is):

Área(s)	Domínio Cognitivo:	Verbo:	Objetivo Educacional:
Ciência da Computação	Conceito Definido	Definir	Depois de finalizado o estudo deste conceito o aluno deve ser capaz de definir

Cadastrar

Figura 4.25: Interface de criação de conceitos do AdaptHA

4.2.8 Anexando Recursos Didáticos a um Conceito

Após o autor ter criado um conceito para um curso, a próxima etapa de autoria consiste em anexar ao conceito os recursos didáticos que encapsulam o conteúdo propriamente dito que o conceito deve abordar. A figura 4.26 mostra como o *AdaptHA* estrutura o conteúdo instrucional dos conceitos dos cursos. Imediatamente abaixo do conceito, encontra-se o *elemento estrutural*, cujo objetivo é agregar um conjunto de eventos de instrução a serem aplicados com um propósito específico. Por exemplo, o elemento estrutural *Visão Geral* (figura 4.26) agrega eventos de instrução que visam diagnosticar e contextualizar o aluno antes que a educação propriamente dita inicie. Assim sendo, a *Visão Geral* inclui os seguintes eventos de instrução: diagnóstico do conhecimento do aluno com relação ao conceito e seus pré-requisitos; informação dos objetivos a serem alcançados; introdução ao conteúdo que será trabalhado posteriormente; e apresentação dos pré-requisitos do conceito.

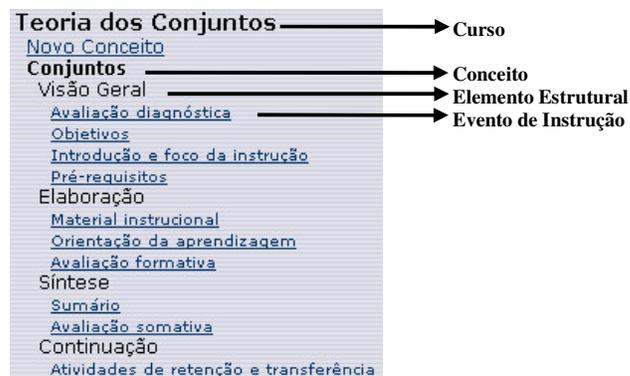


Figura 4.26: Estruturação do conteúdo dos conceitos no *AdaptHA*

Um dos aspectos mais importantes da etapa de entrada de conteúdo, é que nela se define o quão *adaptativo* o sistema será para o aluno. O *AdaptHA* fornece mecanismos que viabilizam a elaboração de cursos adaptativos, entretanto, cabe ao autor decidir o grau de adaptação que será oferecido ao aluno, sendo possível, inclusive, não ser oferecido grau algum de adaptação. Caso o autor queira elaborar um conteúdo adaptativo, é necessário que ele pense em duas *dimensões de adaptação*: áreas às quais o curso atende e nível de conhecimento do aluno. A tabela 4.1 mostra o relacionamento entre os elementos estruturais, eventos de instrução e capacidade de adaptação às dimensões área e nível de conhecimento do aluno.

Tabela 4.1: Elem. Estruturais X Eventos de Instrução X Dimensões de Adaptação

Elemento Estrutural	Evento de Instrução	Adaptável à Área	Adaptável ao Nível de Conhecimento do Aluno
<i>Visão Geral</i>	Avaliação diagnóstica	Sim	Não
	Objetivos	Sim	Não
	Introdução e foco da instrução	Sim	Sim
	Pré-requisitos	Não	Não
<i>Elaboração</i>	Material instrucional	Sim	Sim

<i>Síntese</i>	Orientação da aprendizagem	Sim	Sim
	Avaliação formativa	Sim	Sim
	Sumário	Sim	Sim
	Avaliação somativa	Sim	Sim
<i>Continuação</i>	Atividades de retenção e transferência	Sim	Sim

A fim de esclarecer o mecanismo que viabiliza a adaptação, tome-se como exemplo, um curso de Teoria dos Conjuntos (figura 4.26), que visa atender às áreas de Ciência da Computação e Matemática. Tendo o autor criado o conceito de Conjuntos (figura 4.26), tornar-se-ão disponíveis os elementos estruturais, com seus respectivos eventos de instrução. Supondo-se que o autor deseja elaborar uma avaliação diagnóstica, o evento de instrução *Avaliação diagnóstica* (figura 4.26) deve ser clicado. Com isso, é apresentada ao autor a interface exibida na figura 4.27.

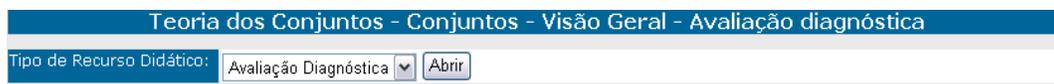


Figura 4.27: Parte super. da interface de seleção de recursos didáticos do *AdaptHA*

Clicando-se no botão *Abrir* (figura 4.27) o sistema apresenta ao autor a interface exibida na figura 4.28 (à esquerda). Com base na informação contida na tabela 4.1, o sistema possibilita que o autor selecione avaliações diagnósticas diferentes para cada área que o curso atende. O autor pode selecionar uma única avaliação para ambos os cursos marcando a opção *A avaliação é a mesma para todas as áreas do curso* (à direita da figura 4.28). A figura 4.29 exhibe a interface de seleção de recursos didáticos como um todo.



Figura 4.28: Parte inferior da interface de seleção de recursos didáticos do *AdaptHA*

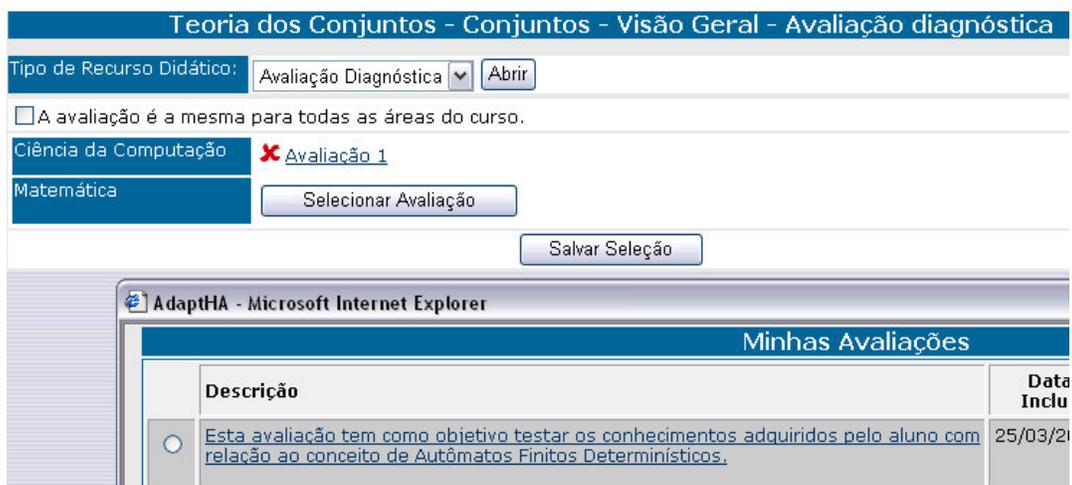


Figura 4.29: Interface de seleção de recursos didáticos do *AdaptHA*

Clicando-se no botão *Selecionar Avaliação* (figura 4.29), o sistema apresenta uma interface (parte inferior da figura 4.29) que permite a seleção de uma avaliação. As avaliações disponíveis são todas as avaliações criadas pelo autor mais as avaliações públicas de outros autores. O botão vermelho em X (figura 4.29) possibilita a exclusão da seleção feita anteriormente. O botão *Salvar Seleção* (figura 4.29) salva as seleções feitas nesta interface.

Supondo-se que o autor deseja elaborar um texto introdutório para focalizar o ensino, o evento de instrução *Introdução e foco da instrução* (figura 4.26) deve ser clicado. Com isto, é apresentada ao autor a interface exibida na figura 4.30. Clicando-se no botão *Abrir* (figura 4.30) é apresentada ao autor a interface exibida na parte superior da figura 4.31. Com base na informação contida na tabela 4.1, o sistema apresenta ao autor a possibilidade de criar textos diferentes para cada área e para cada nível de conhecimento do aluno. A figura 4.31 ilustra os quatro cenários possíveis de tipos de seleção que o sistema é capaz de disponibilizar para o autor.



Figura 4.30: Parte sup. da interface de seleção de recursos didáticos do *AdaptHA*

O(a) Texto de Introdução é (o)a mesmo(a) para todas as áreas do curso.
 O(a) Texto de Introdução é (o)a mesmo(a) para todos os níveis de conhecimento do aluno.

Ciência da Computação	Básico	Selecionar Texto
	Intermediário	Selecionar Texto
	Avançado	Selecionar Texto
Matemática	Básico	Selecionar Texto
	Intermediário	Selecionar Texto
	Avançado	Selecionar Texto

O(a) Texto de Introdução é (o)a mesmo(a) para todas as áreas do curso.
 O(a) Texto de Introdução é (o)a mesmo(a) para todos os níveis de conhecimento do aluno.

Básico	Selecionar Texto
Intermediário	Selecionar Texto
Avançado	Selecionar Texto

O(a) Texto de Introdução é (o)a mesmo(a) para todas as áreas do curso.
 O(a) Texto de Introdução é (o)a mesmo(a) para todos os níveis de conhecimento do aluno.

Ciência da Computação	Selecionar Texto
Matemática	Selecionar Texto

O(a) Texto de Introdução é (o)a mesmo(a) para todas as áreas do curso.
 O(a) Texto de Introdução é (o)a mesmo(a) para todos os níveis de conhecimento do aluno.

Selecionar Texto

Figura 4.31: Parte inferior da interface de seleção de recursos didáticos do *AdaptHA*

4.3 Considerações Finais

A primeira seção deste capítulo teve como foco os aspectos tecnológicos envolvidos no desenvolvimento do *AdaptHA*. Foram apresentados os *frameworks* Java utilizados no desenvolvimento e o ambiente de execução do sistema. A segunda seção apresentou em detalhes o ambiente de autoria do *AdaptHA*, desde o cadastramento do autor, aprovação de seu cadastro por parte do administrador da instituição, até as funcionalidades relativas à autoria de cursos e material instrucional.

O próximo capítulo apresenta a conclusão deste trabalho, o que inclui suas contribuições, trabalhos futuros e a produção científica ao longo do curso de mestrado.

5 CONCLUSÃO

É de fundamental importância que a educação via *Web* proporcione ao aluno uma experiência de aprendizagem individualizada que leve em consideração, por exemplo, seu nível de conhecimento e preferências. Para que isto seja possível, faz-se necessário manter um modelo do aluno que reflita tais características e as utilize para promover a adaptação dos aspectos visíveis do sistema. Conforme apresentado neste trabalho, o *AdaptHA* propõe um mecanismo para promover a referida individualização, através da aplicação de três *níveis de adaptação*, que são os seguintes:

- Definição da estratégia pedagógica (que pode ser diretiva, orientada por descoberta ou exploratória), com base no modelo do aluno;
- Aplicação da estratégia pedagógica selecionada no passo anterior;
- Aplicação de técnicas da HA, tanto para adaptação do conteúdo quanto para o suporte à navegação adaptativa.

No entanto, conforme discutido ao longo deste texto, para que seja possível alavancar de fato a educação prática via *Web*, é indispensável fornecer para o professor ferramentas que verdadeiramente viabilizem e facilitem a autoria de cursos e material instrucional. A *facilidade* a ser oferecida ao professor deve ser implementada, em um ambiente de autoria, pensando-se em três aspectos:

- Facilidade para *criar* material instrucional;
- Facilidade para *recuperar* eficientemente material instrucional;
- Facilidade para *reutilizar* material instrucional.

Considerando-se o ambiente de autoria do *AdaptHA*, apresentado neste trabalho, é possível observar que os aspectos de facilidade, mencionados acima, foram implementados neste ambiente, pois:

- A criação do material instrucional (que inclui textos, questões de avaliações e avaliações) é bastante facilitada devido ao editor de textos (FCKeditor) embutido no ambiente. O material instrucional que não pode ser criado dentro do ambiente de autoria, como imagens e animações, pode ser transferido para o servidor em que o *AdaptHA* é executado e inserido no material instrucional criado dentro do ambiente;

- A recuperação do material instrucional é facilitada devido ao rico conjunto de metadados associado tanto ao material instrucional criado dentro do ambiente quanto fora dele;
- A reutilização é facilitada porque o material instrucional autorado é mantido de maneira independente dos cursos, sendo apenas referenciado pelos que o utilizam. Este conceito de reutilização foi herdado do modelo Hyper-Automaton.

Um aspecto de acentuada relevância, em geral não priorizado na maioria dos sistemas de ensino via *Web*, é a aplicação sistemática de avaliações ao longo de todo o processo de ensino-aprendizagem. A fim de suprir tal necessidade, o *AdaptHA* dispõe de dois dispositivos distintos:

- Um ambiente de autoria de questões de avaliações e avaliações, que é parte integrante do ambiente de autoria. A criação de uma nova avaliação acaba sendo facilitada pelo fato de que o autor tem à sua disposição um *banco de questões* composto por todas as questões por ele autoradas mais as questões públicas criadas por outros autores. Outro agente facilitador neste processo é o eficiente mecanismo de recuperação de questões de avaliações a partir dos metadados a elas associados;
- A aplicação sistemática das avaliações ao longo do processo de ensino-aprendizagem acaba sendo estimulada porque a própria estrutura de apresentação dos conceitos (tabela 3.5) já sugere (não obriga) a aplicação das mesmas, tanto para diagnóstico (avaliação diagnóstica), quanto para formação (avaliação formativa) e quantificação do conhecimento do aluno (avaliação somativa).

Os resultados das avaliações realizadas pelos alunos constituem uma boa fonte de informação que pode ser utilizada tanto para oferecer *feedback* para o aluno quanto para fornecer a base para inferências futuras.

É necessário salientar que o grande diferencial deste trabalho, reside no fato de que o ambiente de autoria do *AdaptHA* é um ambiente que viabiliza a criação de cursos *adaptativos*. Compete ao autor decidir o grau de adaptação que será oferecido ao aluno, sendo possível, inclusive, não ser oferecido grau algum de adaptação. Caso o autor queira elaborar um curso adaptativo, é necessário que ele pense em duas *dimensões de adaptação*: áreas às quais o curso atende e nível de conhecimento do aluno.

5.1 Trabalhos Futuros

O trabalho desenvolvido até então, abre um leque de oportunidades que podem ser exploradas em trabalhos futuros, como por exemplo:

- Com a finalidade de facilitar ainda mais a criação de cursos, implementar um módulo de importação de cursos que permita transformar automaticamente arquivos de texto, de diversos formatos, em cursos do *AdaptHA*;
- Implementar um *applet* com interface gráfica sofisticada o suficiente que permita ao autor criar, editar e visualizar os conceitos do curso e seus relacionamentos, oferecendo-lhe, desta forma, uma visão de mais alto nível do curso, além de lhe poupar bastante trabalho;

- Permitir que o autor especifique relações de pré-requisitos entre conceitos de forma *disjunta*. Atualmente, as relações de pré-requisitos podem ser especificadas apenas de forma *conjunta*. Por exemplo: é possível especificar que o conceito C3 tem como pré-requisitos os conceitos C1 e C2, porém, não é possível especificar que o conceito C3 tem como pré-requisito C1 ou C2;
- Implementar um mecanismo para que sejam mantidas duas versões de material instrucional: uma versão corresponderia à versão atual (ativa), que pode ser recuperada e utilizada pelo próprio autor ou por outros autores; a outra versão corresponderia a uma versão de manutenção, utilizada somente pelo autor do material instrucional. Somente quando todas as manutenções necessárias tiverem sido feitas no material instrucional, o autor liberará a nova versão;
- Implementar mecanismos que possibilitem a reutilização de conceitos de cursos e até mesmo de cursos inteiros;
- Diminuir a *granularidade* em que a adaptação de conteúdo pode ser especificada;
- Implementar o ambiente de realização dos cursos, por parte do aluno, conforme proposto neste trabalho;
- Criar um mecanismo que permita exportar cursos em arquivos de texto, de diversos formatos, de maneira que o aluno possa ter acesso aos cursos, mesmo estando *offline*;
- Acoplar ao *AdaptHA* ferramentas colaborativas;
- Permitir que o modelo do domínio possa ser livremente estruturado, de forma que seja possível construir uma estrutura hierárquica com níveis de aninhamento quaisquer entre os conceitos do curso.

5.2 Produção Científica

O trabalho conduzido ao longo do curso de mestrado também teve como resultado a produção e publicação dos seguintes artigos e pôsteres:

- Ambiente para Disponibilização de Cursos na *Web* Utilizando Autômatos Finitos com Saída. Artigo completo. (LIMA et al., 2004);
- *AdaptHA*: Ambiente de Ensino Adaptativo na *Web* Baseado no Modelo Hyper-Automaton. Artigo completo. (LIMA et al., 2005);
- EASy – Recuperação de Questões através de Metadados e Geração Automática de Instrumentos de Avaliação via *Web*. Artigo completo. (ZANELLA et al., 2005);
- O Ambiente de Autoria do *AdaptHA*. Pôster. (LIMA et al., 2006-a);
- O Ambiente de Autoria do *AdaptHA*. Artigo completo. (LIMA et al., 2006-b).

REFERÊNCIAS

- ADVANCED Distributed Learning – SCORM. Disponível em: <<http://www.adlnet.gov/Scorm/index.cfm>>. Acesso em: mar. 2007.
- ABDULLAH, S. C. **Student Modelling By Adaptive Testing – A Knowledge-Based Approach**. 2003. 258 f. Thesis (PhD in Computer Science) – Computing Laboratory, University of Kent, Canterbury.
- APACHE TOMCAT. Disponível em: <<http://tomcat.apache.org/>>. Acesso em: mar. 2007.
- BAUER, C.; KING, G. **Hibernate in Action**. Greenwich: Manning Publications, 2005.
- BLACKBOARD. Disponível em: <<http://www.blackboard.com/>>. Acesso em: mar. 2007.
- BLAYA, C. **Processo de Avaliação**. Disponível em: <http://www.ufrgs.br/tramse/med/textos/2004_07_20_tex.htm>. Acesso em: mar. 2007.
- BRAILSFORD, T. J. et al. Autonavagation, Links and Narrative in an Adaptive Web-Based Integrated Learning Environment. In: INTERNATIONAL WORLD WIDE WEB CONFERENCE, 11., 2002, Honolulu. **Proceedings...** [S.l.:s.n.], 2002.
- BRUNETTO, M. A. C.; OLIVEIRA, J. P. M.; PROENÇA JR, M. L.; PIMENTA, M. S.; RIBEIRO, C. H. F. P.; LIMA, J. V.; FREITAS, V.; MARÇAL, V. P.; GASPARINI, I.; AMARAL, M. A. AdaptWeb: um Ambiente para Ensino-aprendizagem Adaptativo na Web. **Educar em Revista**, Curitiba, n. 107, p. 175-198, 2003.
- BRUSILOVSKY, P. Methods and Techniques of Adaptive Hypermedia. **User Modeling and User-Adapted Interaction**, Dordrecht, v. 6, n. 2-3, p. 87-129, 1996-a.
- BRUSILOVSKY, P. Adaptive Hypermedia: an attempt to analyze and generalize. In: INTERNATIONAL CONFERENCE ON MULTIMEDIA, HYPERMEDIA AND VIRTUAL REALITY MODELS, SYSTEMS AND APPLICATIONS, MHVR, 1., 1994, Moscow. **Selected Papers**. Berlin: Springer, 1996-b. p. 288-304. (Lecture Notes in Computer Science, v. 1077).
- BRUSILOVSKY, P. Efficient Techniques for Adaptive Hypermedia. In: **Intelligent Hypertext: advanced techniques for the World Wide Web**. Berlin: Springer, 1997. p. 12-30. (Lecture Notes in Computer Science, v. 1326).
- BRUSILOVSKY, P. Adaptive Hypermedia. **User Modeling and User-Adapted Interaction**, Dordrecht, p. 87-110, 2001.
- BRUSILOVSKY, P. Adaptive Educational Hypermedia: From Generation to Generation. In: HELLENIC CONFERENCE ON INFORMATION AND

COMMUNICATION TECHNOLOGIES IN EDUCATION, 4., 2004. **Proceedings...** [S.l.:s.n.], 2004. p. 13-33.

BRUSILOVSKY, P.; NEJDL, W. Adaptive Hypermedia and Adaptive Web. In: SINGH, M. P. **Practical Handbook of Internet Computing**. Boca Raton: Chapman & Hall / CRC, 2005. p. 1-14.

CARVER, C. A.; HILL, J. M. D.; POOCH, U. W. Third Generation Adaptive Hypermedia Systems. In: WORLD CONFERENCE ON THE WWW AND INTERNET, 1999, Honolulu. **Proceedings...** [S.l.:s.n.], 1999.

CRISTEA, A. Authoring of Adaptive Hypermedia. **Educational Technology & Society**, [S.l.], v. 8, n. 3, p. 6-8, 2005.

CRISTEA, A. **Adaptive Hypermedia**. Disponível em: <www.wis.win.tue.nl/~acristea/AH/CoursePowerpoints&Demos/AH1.ppt>. Acesso em: fev. 2007.

DAVID AUSUBEL. Disponível em: <http://pt.wikipedia.org/wiki/David_Ausubel/>. Acesso em: mar. 2007.

DE BRA, P.; CALVI, L. AHA! An Open Adaptive Hypermedia Architecture. **The New Review of Hypermedia and Multimedia**, [S.l.], v. 4, p. 115-139, 1998.

DE BRA, P. et al. AHA! The Adaptive Hypermedia Architecture. In: ACM HYPERTEXT HYPERMEDIA CONFERENCE, HYPERTEXT, 14., 2003. **Proceedings...**New York: ACM, 2003. p. 81-84.

DE BRA, P. **AHA! 3.0 Tutorial**. Disponível em: <<http://aha.win.tue.nl:18080/aha/tutorial/>>. Acesso em: dez. 2006.

EKLUND, J. H.; ZEILIGER, R. Navigating the Web: Possibilities and Practicalities for Adaptive Navigation Support. In: AUSTRALIAN WORLD WIDE WEB CONFERENCE, 2., 1996, Gold Coast. **Proceedings...** [S.l.:s.n.], 1996. p. 73-78.

ESCOLANET – Educação Continuada – Dicionário. Disponível em: <http://www.escolanet.com.br/dicionario/dicionario_p.html>. Acesso em: fev. 2007.

FCKEDITOR. Disponível em: <<http://www.fckeditor.net/>>. Acesso em: mar. 2007.

GAGNÉ, R. M. **Princípios Essenciais da Aprendizagem para o Ensino**. Porto Alegre: Globo, 1980.

GAGNÉ, R. M. **The Conditions of Learning**. 3rd ed. New York: Holt, Rinehart and Winston, 1977.

GAMMA, E. et al. **Padrões de Projeto – Soluções Reutilizáveis de Software Orientado a Objetos**. Porto Alegre: Bookman, 2000.

IEEE LTSC. **Draft Standard for Learning Object Metadata**. Disponível em: <http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf>. Acesso em: mar. 2007.

J2EE 1.4 TUTORIAL. Disponível em: <<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>>. Acesso em: mar. 2007.

JAVASERVER Faces. Disponível em: <<http://java.sun.com/javaee/javaserverfaces/>>. Acesso em: mar. 2007.

- JBOSS Application Server. Disponível em: <<http://labs.jboss.com/portal/jbossas/>>. Acesso em: mar. 2007.
- JOHNSON, R.; FOOTE, B. Designing Reusable Classes. **Journal of Object-Oriented Programming**, [S.l.], v. 1, n. 2, p. 22-35, June/July 1988.
- JOHNSON, W. L. Using Agent Technology to Improve the Quality of Web-Based Education. In: ZHONG, N.; LIU, J.; YAO, Y. **Web Intelligence**. Berlin: Springer, 2003. p. 77-101.
- KLANN, M. et al. Shared Initiative: Cross-fertilisation Between System Adaptivity and Adaptability. In: INTERNATIONAL CONFERENCE ON HUMAN-COMPUTER INTERACTION, 10., 2003, Creta. **Proceedings...** [S.l.:s.n.], 2003.
- KOBSA, A.; KOENEMANN, J.; POHL, W. Personalised Hypermedia Presentation Techniques for Improving Online Customer Relationships. **The Knowledge Engineering Review**, New York, v. 16, p. 111-155, Mar. 2001.
- KOCH, N. P. **Software Engineering for Adaptive Hypermedia Systems**. 2000. 371 f. Thesis (PhD in Computer Science) – Computing Institute, University of Munich, Munich.
- LIMA, G. C. B.; MENEZES, P. B.; ZANELLA, R.; MACHADO, J. P. Ambiente para Disponibilização de Cursos na *Web* Utilizando Autômatos Finitos com Saída. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, SBIE, 15., 2004, Manaus. **Anais...** Manaus: EDUA, 2004.
- LIMA, G. C. B.; ZANELLA, R.; MENEZES, P. B. *AdaptHA*: Ambiente de Ensino Adaptativo na *Web* Baseado no Modelo Hyper-Automaton. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2005, Juiz de Fora. **Anais...** Juiz de Fora: SBC, 2005. p. 424-434.
- LIMA, G. C. B.; ZANELLA, R.; MENEZES, P. B. O Ambiente de Autoria do *AdaptHA*. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 17., 2006. **Anais...** Brasília: SBC, 2006-a. 1 CD-ROM.
- LIMA, G. C. B.; ZANELLA, R.; MENEZES, P. F. B. O Ambiente de Autoria do *AdaptHA*. **RENTE: Revista Novas Tecnologias na Educação**, Porto Alegre, v. 4, n. 2, dez. 2006-b. Disponível em: <<http://www.cinted.ufrgs.br/rente/dez2006/artigosrente/25162.pdf>>. Acesso em: fev. 2007.
- MACHADO, J. P.; PENCZEK, L.; MORAIS, C. T. Q.; MENEZES, P. B. Autômatos Finitos: um Formalismo para Cursos na *Web*. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, SBES, 13., 1999, Florianópolis. **Anais...** Florianópolis: EDUA, 1999-a.
- MACHADO, J. P.; MENEZES, P. B.; PENCZEK, L. A System for Web Based Instruction Using Sequential Automata. In: INTERNATIONAL CONFERENCE ON ENGINEERING AND COMPUTER EDUCATION, 1999, Rio de Janeiro. **Proceedings...** [S.l.:s.n.], 1999-b.
- MANN, K. D. **JavaServer Faces in Action**. Greenwich: Manning Publications, 2005.
- MARIETTO, M. G. B. **Definição Dinâmica de Estratégias Instrucionais em Sistemas de Tutoria Inteligentes: uma Abordagem Multiagentes na WWW**. 2000. 192

f. Tese (Doutorado em Engenharia Eletrônica e Computação) – Instituto Tecnológico de Aeronáutica, São José dos Campos.

MATENA, V. et al. Java™ 2 Platform Enterprise Edition Specification, v1.4. Disponível em: < http://java.sun.com/j2ee/j2ee-1_4-fr-spec.pdf>. Acesso em: mar. 2007.

MENEZES, P. B. **Linguagens Formais e Autômatos**. 4. ed. Porto Alegre: Sagra Luzzatto, 2002.

MOORE, A. et al. Links for Learning: Linking for an Adaptive Learning Environment. In: INTERNATIONAL CONFERENCE ON WEB-BASED EDUCATION, WBE, 2004, Innsbruck. **Proceedings...**[S.l.:s.n.], 2004.

MULTIMEDIA. Disponível em: < <http://en.wikipedia.org/wiki/Multimedia/>>. Acesso em: fev. 2007.

MYFACES. Disponível em: <<http://myfaces.apache.org/>>. Acesso em: mar. 2007.

NIELSEN, J. **Usability 101: Introduction to Usability**. Disponível em: <<http://www.useit.com/alertbox/20030825.html>>. Acesso em: mar. 2007.

OLIVEIRA, J. P. M.; SILVA, L.; FREITAS, V.; MARÇAL, V. P.; GASPARINI, I.; AMARAL, M. A. AdaptWeb: an Adaptive Web-based Courseware. In: ANNUAL ARIADNE CONFERENCE, 3., 2003, Leuven. **Proceedings...** [S.l.:s.n.], 2003.

OPPERMANN, R. Adaptively Supported Adaptability. **International Journal of Human-Computer Studies**, [S.l.], v. 40, n. 3, p. 455-472, Mar. 1994.

OPPERMANN, R.; RASHEV, R.; KINSHUK. Adaptability and Adaptivity in Learning Systems. In: KNOWLEDGE TRANSFER, 1997, London. **Proceedings...** [S.l.:s.n.], 1997.

PAIVA, A.; SELF, J.; HARTLEY, R. Externalising Learner Models. In: ARTIFICIAL INTELLIGENCE IN EDUCATION, 1995, Washington. **Proceedings...** [S.l.:s.n.], 1995.

PEMBERTON, S. et al. **XHTML™ 1.0 The Extensible HyperText Markup Language**. 2nd ed. Disponível em: <<http://www.w3.org/TR/xhtml1/>>. Acesso em: jan. 2007.

RAGNEMALM, E. L. Student Diagnosis in Practice; Bridging a Gap. **User Modeling and User-Adapted Interaction**, Dordrecht, v. 5, n. 2, p. 93-116, 1995.

RDF – Resource Description Framework. Disponível em: < <http://www.w3.org/RDF/#specs/>>. Acesso em: mar. 2007.

SEMANTIC Web. Disponível em: <<http://www.w3.org/2001/sw/>>. Acesso em: mar. 2007.

STRUTS. Disponível em: <<http://struts.apache.org/>>. Acesso em: mar. 2007.

TOPIC Maps. Disponível em: <<http://www.topicmaps.org/>>. Acesso em: mar. 2007.

TOPCLASS. Disponível em: <<http://www.topclass.nl/>>. Acesso em: mar. 2007.

WOLF, C. iWeaver: Towards an Interactive Web-Based Adaptive Learning Environment to Address Individual Learning Styles. In: INTERACTIVE COMPUTER AIDED LEARNING WORKSHOP, 2002, Villach. **Proceedings...** [S.l.:s.n.], 2002.

WOLF, C. iWeaver: Towards ‘Learning Style’-based e-Learning in Computer Science Education. In: AUSTRALASIAN COMPUTING EDUCATION CONFERENCE AS PART OF THE AUSTRALASIAN COMPUTER SCIENCE WEEK, 5., 2003, Adelaide. **Proceedings...** [S.l.:s.n.], 2003.

ZAKARIA, M. R. et al. “Pluggable” User Models for Adaptive Hypermedia in Education. In: ACM CONFERENCE ON HYPERTEXT AND HYPERMEDIA, 14., 2003, Nottingham. **Proceedings...** [S.l.:s.n.], 2003.

ZANELLA, R.; LIMA, G. C. B., ANTUNES, B., MENEZES, P. B. EASy – recuperação de questões através de metadados e geração automática de instrumentos de avaliação via web. **RENOTE: revista novas tecnologias na Educação**, Porto Alegre, v. 3, n. 2, nov. 2005. Trabalho apresentado no VI Ciclo de Palestras Novas Tecnologias na Educação. Disponível em: <<http://www.cinted.ufrgs.br/renote/nov2005/index.html>>. Acesso em: mar. 2007.

W3C. **FAQ:** Localization vs. Internationalization. Disponível em: <<http://www.w3.org/International/questions/qa-i18n/>>. Acesso em: mar. 2007.