

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

**Interface Adaptativa no ambiente AdaptWeb:
navegação e apresentação adaptativa
baseada no modelo do usuário**

por

ISABELA GASPARINI

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

Prof. Marcelo Soares Pimenta
Orientador

Porto Alegre, junho de 2003.

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Gasparini, Isabela

Interface Adaptativa no Ambiente AdaptWeb: navegação e apresentação adaptativa baseada no modelo do usuário / por Isabela Gasparini. – Porto Alegre: PPGC da UFRGS, 2003.

97p.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2003. Orientador: Pimenta, Marcelo Soares.

1. Educação a distância baseada na Web. 2. Sistemas Hiperídia Adaptativos. 3. AdaptWeb. I. Pimenta, Marcelo Soares. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Prof^a. Wrana Maria Panizzi

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

Pró-Reitor Adjunto de Pós-Graduação: Prof. Jaime Evaldo Fensterseifer

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

Coordenador do PPGC: Prof. Carlos Alberto Heuser

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

Dedicatória

Dedico esta dissertação à minha mãe Maria Helena, ao meu irmão Vinícius, ao meu pai José Gasparini, a meu esposo Marcos Hideshi Kimura, e a meu tio Antônio Valdemir Zago.

Agradecimentos

Primeiramente, agradeço a DEUS, que sempre me mostrou o caminho certo a seguir!

Agradeço a todos que colaboraram com esta dissertação e em especial a Marcos Hideshi Kimura por me ajudar sempre que necessário.

A minha família que sempre me apoiou.

Ao Prof. Marcelo S. Pimenta, pela sua orientação, confiança e por sua amizade.

Aos integrantes do projeto AdaptWeb, José Palazzo de Oliveira, José Valdeni de Lima, Mário L. Proença Jr, Viviane Prado Marçal, Marília A Amaral, Veronice de Freitas e em especial a Maria Angélica C. Brunetto por sua colaboração, ajuda, paciência e atenção disponibilizada para a conclusão desta dissertação.

A todos os integrantes do departamento de computação da UEL.

Ao professor Rodolfo M. de Barros pela sua compreensão e ajuda na hora do aperto.

Agradeço a todas as outras pessoas que sempre torceram por mim.

Sumário

Lista de Abreviaturas.....	7
Lista de Figuras	8
Lista de Tabelas	10
Resumo	11
Abstract	12
1 Introdução.....	13
2 Sistemas Hipermedia Adaptativos	15
2.1 Introdução a SHA.....	15
2.2 Onde e porque SHA podem ser úteis.....	18
2.2.1 SHA para EAD baseada na <i>Web</i>	19
2.3 Arquitetura básica de um SHA	20
2.4 Modelo de Usuário em SHA	21
2.4.1 Modelos de descrição	22
2.4.2 Quais aspectos podem ser adaptados?	24
2.4.3 Classificação do Modelo do Usuário.....	27
2.5 O que pode ser adaptado em SHA	29
3 Métodos e Técnicas Adaptativas para SHA	31
3.1 Métodos e Técnicas para Apresentação Adaptativa	31
3.1.1 Métodos de Adaptação de Conteúdo	31
3.1.2 Técnicas de Adaptação de Conteúdo.....	33
3.1.3 Métodos e Técnicas de Adaptação da Apresentação da Interface.....	34
3.2 Métodos e Técnicas para Suporte a Navegação Adaptativa.....	35
3.2.1 Classificação dos <i>links</i>	36
3.2.2 Métodos para Suporte a Navegação Adaptativa.....	36
3.2.3 Técnicas para Suporte a Navegação Adaptativa.....	39
4 Ambiente Adaptweb.....	48
4.1 Ambiente de desenvolvimento	48
4.2 Arquitetura	49
4.2.1 Módulo 1: Autoria	50
4.2.2 Módulo 2: Armazenamento em XML	52
4.2.3 Módulo 3: Adaptação de conteúdo baseado no modelo do aluno	54
4.2.4 Módulo 4: Interface Adaptativa.....	55
5 Interface Adaptativa	56
5.1 Tecnologias utilizadas para o desenvolvimento do módulo de Interface Adaptativa	57
5.2 Aspectos que podem ser adaptados no Ambiente AdaptWeb	58
5.2.1 Formação do Usuário	58

5.2.2 Ambiente de Trabalho	59
5.2.3 Conhecimento.....	59
5.2.4 Preferência pelo modo de navegação	60
5.3 Modelagem do Ambiente	61
5.3.1 Modelagem do Banco de Dados Administrativo.....	61
5.3.2 Modelagem do Usuário e do Domínio	61
5.4 Técnicas Adaptativas utilizadas para prover adaptação baseada nas características do Usuário.....	62
5.5 Interface do ambiente de ensino.....	66
5.6 Monitoramento	69
5.7 Filtros.....	70
6 Conclusão	86
Anexo XML de Estrutura de Conceito da Disciplina Computação Algébrica e Numérica.....	88
Referências	92

Lista de Abreviaturas

AdaptWeb	Ambiente de Ensino-Aprendizagem Adaptativo na Web
ADSL	Asymmetrical Digital Subscriber Line
AHS	Adaptive Hypermedia System
API	Application Programming Interface
CAI	Computer Aided Instruction
DOM	Document Object Model
DTD	Document Data Definition
EAD	Educação a Distância
ECA	Estilo Cognitivo de Aprendizagem
HTML	HyperText Markup Language
IHC	Interface Humano-Computador
PHP	Hypertext Preprocessor
SHA	Sistema Hipermídia Adaptativo
SQL	Structured Query Language
STI	Sistemas Tutores Inteligentes
XML	Extensible Markup Language
UEL	Universidade Estadual de Londrina
UFRGS	Universidade Federal do Rio Grande do Sul
W3C	World Wide Web Consortium

Lista de Figuras

FIGURA 2.1 – Diferentes estruturas para um sistema hipermídia [LYN 2002].....	16
FIGURA 2.2 – Processo de adaptação nos SHA [BRU 96], [PAL 2002].....	17
FIGURA 2.3 – Arquitetura de um SHA.....	21
FIGURA 2.4 – Tecnologias para SHA.....	30
FIGURA 3.1 – Orientação Direta no ISIS-Tutor [PES 96].....	40
FIGURA 3.2 – Sistema Adaptive Hyperman [MYK 92].....	41
FIGURA 3.3 – Exemplo do Sistema AST e suas variações das técnicas de adaptação..	42
FIGURA 3.4 – ISIS-Tutor sem (a) e com (b) a utilização da técnica de remoção de <i>links</i>	43
FIGURA 3.5 – A utilização da técnica de anotação no sistema Interbook.....	45
FIGURA 3.6 – Sistema ELM-ART.....	45
FIGURA 3.7 – O sistema ELM-ART II.....	46
FIGURA 4.1 – Arquitetura do Ambiente AdaptWeb.....	50
FIGURA 4.2 – Esquema de geração de arquivos XML.....	53
FIGURA 4.3 – Organização de um conteúdo institucional em lista de conceitos e em árvore de conceito.....	53
FIGURA 5.1 – Arquitetura do ambiente AdaptWeb com o módulo de Interface Adaptativa em destaque.....	56
FIGURA 5.2 – Relacionamento entre as tabelas do banco de dados administrativo.....	61
FIGURA 5.3 – Exemplificação da técnica de remoção de <i>links</i>	63
FIGURA 5.4 – Exemplificação da técnica de anotação unida com a desabilitação de <i>links</i>	64
FIGURA 5.5 – Interface do ambiente de ensino no modo livre dividida por áreas de visualização	66
FIGURA 5.6 – Autenticação do usuário	72
FIGURA 5.7 – Disciplinas que o aluno pode assistir.....	72
FIGURA 5.8 – Seleção da conexão de rede e do modo de navegação	73
FIGURA 5.9 – Esquema para obtenção do nome da disciplina dinamicamente.....	73
FIGURA 5.10 – Geração do nome da disciplina e sua localização	73
FIGURA 5.11 – Esquema para achar tópicos conceitos por curso	74
FIGURA 5.12 – Menus de navegação para os cursos de Ciência da Computação e Matemática	74
FIGURA 5.13 – Esquema para habilitação das guias	75
FIGURA 5.14 – Tela da interface com o tópico Método de Gauss visualizado e as guias de <i>exemplos e material complementar</i> habilitadas.....	76
FIGURA 5.15 – Geração do nome do tópico que está sendo visualizado	76
FIGURA 5.16 – Esquema para achar o caminho hierárquico do conceito	77
FIGURA 5.17– Caminho hierárquico	77

FIGURA 5.18 – Esquema para caminho em exemplo, exercício ou material complementar	77
FIGURA 5.19 – Caminho hierárquico quando navegando pelo material complementar .	78
FIGURA 5.20 – Esquema para apresentação da navegação direta	78
FIGURA 5.21 – Navegação direta	78
FIGURA 5.22 – Esquema para a apresentação da Interface	79
FIGURA 5.23 – Interface inicial da disciplina no modo TUTORIAL	79
FIGURA 5.24 – Interface inicial da disciplina no modo LIVRE	80
FIGURA 5.25 – Esquema da identificação do menu	80
FIGURA 5.26 – Menu de navegação para tópicos conceitos no modo livre identificando conceitos visitados/aprendidos, não visitados/não aprendidos e atual. ..	81
FIGURA 5.27 – Menus dos tópicos para o curso de ciência da computação (a) Conceito (b) Exemplos (c) Exercícios (d) Material Complementar	82
FIGURA 5.28 – Esquema para geração do menu de navegação	83
FIGURA 5.29 – Menu de navegação no modo tutorial	83
FIGURA 5.30 – Menu de navegação de Material Complementar no modo tutorial	84
FIGURA 5.31 – Esquema para pesquisa.....	84
FIGURA 5.32 – Resultado da pesquisa	85
FIGURA 5.33 – Mapa da disciplina Computação Algébrica e Numérica.....	85

Lista de Tabelas

TABELA 3.1 – Métodos e Técnicas para Adaptação do Conteúdo.....	34
TABELA 3.2 – Métodos e Técnicas para Adaptar a apresentação da Interface	35
TABELA 3.3 – Exemplo de Sistemas que utilizam Métodos e Técnicas de Apresentação Adaptativa.....	35
TABELA 3.4 – Métodos e Técnicas para Adaptação Navegacional	47
TABELA 3.5 – Exemplos de Sistemas que utilizam Adaptação Navegacional	47
TABELA 5.1 – Técnicas adaptativas utilizadas no Ambiente AdaptWeb.....	66
TABELA 5.2 – Estrutura Hierárquica de Conceitos da Disciplina Computação Algébrica e Numérica.....	71
TABELA 5.3 – Estrutura dos Pré-requisitos e da Adaptabilidade por Cursos	71
TABELA 5.4 – Exemplos, exercícios e material complementar para o curso de Ciência da Computação	75

Resumo

A área de pesquisa em Informática na Educação tem evoluído com a introdução da Internet e milhares de cursos baseados na *Web* que surgiram nos últimos anos. Entretanto, estas novas aplicações educacionais são, em muitos casos, desenvolvidas como um repositório estático de conteúdo, sem aproveitar adequadamente as características dinâmicas e de adaptabilidade que o ambiente *Web* propicia. Este fato ocasiona sérios problemas de usabilidade:

- Problemas de orientação: uma página apresenta constantemente o mesmo conteúdo e os mesmos *links*, não levando em conta quais páginas são mais importantes para o aluno, dificultando a busca de informações de seu interesse. Nossa premissa é que alunos com diferentes metas e conhecimentos podem estar interessados em diferentes porções da informação apresentada em uma página hipermídia e podem utilizar diferentes *links* para a navegação entre as páginas.
- Problemas de caminho: a navegação por caminhos totalmente livres pode ocasionar problemas de compreensão de localização. Nossa premissa é que muitas vezes é aconselhável uma determinada ordenação para percorrer um conjunto de conhecimentos, o qual é construído aos poucos, tornando-se pré-requisito para posteriores conhecimentos. No entanto, reconhecemos que a navegação arbitrária e livre não pode ser de todo abolida.

Estes problemas são críticos para ambientes de educação a distância (EAD) baseados na *Web*. Sistemas Hipermídia Adaptativos (SHA), são sistemas que refletem algumas características dos alunos no modelo do usuário e aplicam este modelo para adaptar vários aspectos visíveis do sistema. SHA tentam superar estes problemas adaptando o conteúdo (informação) e a navegação (*links*) a serem apresentados para o usuário.

O AdaptWeb (Ambiente de Ensino-aprendizagem Adaptativo na *Web*) é um sistema hipermídia adaptativo de educação a distância baseado na *Web* que tem como finalidade adaptar o conteúdo, a apresentação e a navegação de acordo com o modelo do usuário.

O módulo de interface adaptativa descrito neste trabalho é parte integrante deste projeto e têm como finalidade adaptar a navegação do aluno de acordo com suas características descritas no modelo do usuário, além de adaptar a apresentação da interface juntamente com o módulo de adaptação de conteúdo.

Palavras-chave: educação a distância baseada na *Web*, sistemas hipermídia adaptativos, AdaptWeb.

TITLE: “ADAPTIVE INTERFACE IN THE ADAPTWEB ENVIRONMENT: ADAPTIVE NAVIGATION AND ADAPTIVE PRESENTATION BASED ON USER’S MODEL”

Abstract

The research in Computer Science and Education has evolved with the introduction of the Internet and thousand of courses based on the Web that had appeared in recent years. However, these new educational applications are, in many cases, developed as a static repository of content, without using to advantage the dynamic characteristics adequately and of adaptability that the Web environment propitiates. This fact causes serious usability problems:

- Orientation problems: a page constantly presents the same content and set of links, don't taking in account which pages are more important for the student, making the search of information of his interest a very difficult task. Our premise is that students with different goals and knowledge can be interested in different portions of the information presented in a hypermedia page and they can use different links for the navigation between the pages.
- Navigation problems: the navigation for totally open ways can cause problems of localization understanding. Our premise is that many times is advisable determined ordinance to cover a set of knowledge, which is constructed gradually, becoming prerequisite for posterior knowledge. However, we recognize that arbitrary and open navigation cannot of all be abolished.

These problems are critical for e-learning system. Adaptive Hypermedia System (AHS) are systems that reflect some characteristics of the students in the user's model and apply this model to adapt some visible aspects of the system. AHS try to surpass these problems adapting their content (information) and their navigation (links) to be presented for the user.

The AdaptWeb Environment (Adaptive Web-based learning Environment) is an adaptive e-learning system that has as purpose to adapt the content, the presentation and the navigation in accordance with the user's model.

The adaptive interface module described in this work is integrant part of the AdaptWeb project and has as purpose to adapt the student's navigation in accordance with his described characteristics in the user's model, besides together adapting the presentation of the interface with the content adaptation module.

Keywords: e-learning, adaptive hypermedia system, AdaptWeb Environment.

1 Introdução

Sistemas Hiperfídia, e particularmente sistemas hiperfídia baseados na Web, tm se tornado muito populares nos ltimos anos como ferramentas versteis e poderosas para organizaço e acesso a informaço. Uma das mais promissoras estratgias de sistemas hiperfídia baseados na Web sã os sistemas que se relacionam com a educaço a distncia (EAD). Isto porque cada vez mais pessoas precisam aprender ou se especializar em um tempo relativamente pequeno, e a Web permite ao usuáriu-aluno a chance de poder participar até mesmo de cursos nos quais seria impossível estar presente fisicamente.

Um dos aspectos mais importantes nos sistemas de EAD é encontrar a melhor maneira em que a informaço possa ser apresentada aos alunos. Porém, em um mesmo ambiente, pode-se encontrar diferentes classes de alunos com características e objetivos bem distintos. Sendo assim começaram as investigaçoes por novas maneiras de apresentar os conteúdos em diferentes formas para os diferentes usuárius, criando os denominados sistemas hiperfídia adaptativos [BOT 2000].

Brusilovsky [BRU 96] explica que Sistemas Hiperfídia Adaptativos (doravante abreviados SHA) podem ser úteis em qualquer área de aplicaço onde o sistema pode ser utilizado por pessoas com diferentes metas e conhecimentos e onde o hiperespaço é razoavelmente grande. Usuárius com diferentes metas e diferentes conhecimentos podem estar interessados em diferentes porçoes da informaço apresentada em uma página hiperfídia e podem utilizar diferentes *links* para a navegaço entre as páginas. SHA tentam superar estes problemas utilizando representaço do conhecimento a partir do modelo do usuáriu para adaptar a informaço e os *links* a serem apresentados para este usuáriu. A adaptaço pode também ajudar o usuáriu no sentido navegacional, o qual é particularmente relevante para um hiperespaço extenso. Conhecendo as metas e os conhecimentos dos usuárius, SHA podem apoiar os usuárius em sua navegaço limitando o espaço de navegaço, sugerindo os *links* mais relevantes para seguir, ou provendo comentários/observaçoes adaptados aos *links* visíveis.

Este trabalho faz parte do Projeto AdaptWeb (Ambiente de Ensino-Aprendizagem Adaptativo na Web), uma parceria da UFRGS (Universidade Federal do Rio Grande do Sul) com a UEL (Universidade Estadual de Londrina), gerando como produto final um ambiente hiperfídia adaptativo de EAD baseado na Web. O módulo especificado e implementado neste trabalho é o módulo de interface adaptativa que tem como objetivo principal adaptar a navegaço do usuáriu de acordo com suas características armazenadas em seu modelo, além de adaptar a apresentaço juntamente com o módulo de adaptaço de conteúdo. As principais características do usuáriu utilizadas pelo ambiente para adaptaço sã o (1) conhecimento do usuáriu, (2) sua preferência navegacional, (3) seu curso – formaço e (4) seu ambiente de trabalho. O módulo de interface adaptativa leva em conta as três primeiras características para adaptar a navegaço do aluno. A quarta é utilizada em outro módulo do projeto [MAR 2003]. Os detalhes do processo de adaptaço, de como estas informaçoes sã coletadas e usadas sã descritas no decorrer do trabalho, mais precisamente no capítulo 5.

O objetivo geral desta dissertaço é desenvolver o módulo de Interface Adaptativa do ambiente AdaptWeb. Os objetivos específicos sã estudar os motivos para a utilizaço desta nova tecnologia, investigar sua utilizaço dentro do contexto da educaço a distncia, estudar como projetar o modelo do usuáriu em conjunto com o módulo de adaptaço de conteúdo, fazer o estudo de como projetar a interface do ambiente de ensino com usabilidade e consistência, e verificar quais métodos e técnicas

podem proporcionar uma adaptação coerente e de fácil entendimento para o usuário dentro do escopo do projeto.

Este trabalho está estruturado como segue. O capítulo 2 discute sobre os Sistemas Hipermídia Adaptativos (SHA), apresentando uma breve introdução, explicando onde e porque SHA são úteis, especificando a arquitetura básica de um SHA, bem como o modelo de usuário em um SHA.

O terceiro capítulo explica os métodos e as técnicas adaptativas para Sistemas Hipermídia Adaptativos, tanto para a apresentação adaptativa (seção 3.1), quanto para a navegação adaptativa (seção 3.2).

O capítulo 4 apresenta o ambiente AdaptWeb, descrevendo seu ambiente de desenvolvimento, sua arquitetura básica e explicando cada módulo.

O capítulo 5 explica o módulo de Interface Adaptativa e a implementação do módulo. Os exemplos utilizados nos capítulos 4 e 5 foram extraídos do estudo de caso real de um curso EAD na *Web* desenvolvido para a UEL: um módulo da disciplina de Computação Algébrica e Numérica para 3 diferentes cursos desta universidade.

Finalmente o sexto capítulo apresenta a conclusão desta dissertação.

2 Sistemas Hipermídia Adaptativos

Esta seção é dedicada a introduzir o conceito de sistemas hipermídia adaptativos. A seção 2.1 faz a introdução de SHA; a seção 2.2 explica onde e porquê SHA podem ser úteis; a seção 2.3 mostra a arquitetura básica de um SHA; a seção 2.4 descreve sobre o modelo do usuário em SHA; e a seção 2.5 esclarece o que pode ser adaptado em SHA.

2.1 Introdução a SHA

Segundo [KOC 2001] existe uma distinção sobre os tipos de sistemas hipermídia (incluindo *Websites*) capazes de executar algum tipo de personalização. Eles podem ser customizáveis, também chamados de adaptáveis, ou podem ser adaptativos, e em ambos os casos o usuário possui um papel fundamental para que o sistema possa oferecer a personalização. A diferença entre eles é a maneira que a adaptação ocorre:

- **Hipermídia Adaptáveis/Ajustáveis ou Customizáveis (*Adaptable Hypermedia*)**: o usuário pode fornecer algum perfil (através de um diálogo ou de um questionário). Baseado neste perfil o sistema fornece uma versão da aplicação do hipermídia que corresponde ao perfil selecionado. Os ajustes podem incluir determinadas preferências da apresentação (cores, tipo de mídia, estilo da aprendizagem, etc.) e características do usuário (qualificações, conhecimento sobre conceitos, etc.) [DEB 99]. São exemplos comuns *sites* de comércio-eletrônico e de cadastro de *e-mail* gratuito, onde o usuário deve fazer um cadastro com suas preferências, seu perfil econômico, social, etc., e pode mudá-lo quando achar necessário.
- **Hipermídia Adaptativos/Adaptáveis (*Adaptive Hypermedia*)**: o sistema monitora o comportamento do usuário e adapta a apresentação adequadamente. A evolução das preferências do usuário e de seu conhecimento pode ser deduzida dos acessos da página. Às vezes o sistema necessita utilizar questionários ou testes para ter uma impressão mais exata do estado do usuário. A maioria da adaptação, entretanto é baseada nas ações de navegação do usuário, e possivelmente também no comportamento de outros usuários [DEB 99]. A adaptação pode ser feita modificando-se apresentações e formas de acesso preestabelecidas ou construindo-as com partes da informação. Neste caso, onde existe uma geração dinâmica de páginas, esses sistemas também são conhecidos como hipermídia dinâmicos [KOC 2001]. Alguns exemplos de sistemas hipermídia adaptativos podem ser vistos no capítulo 3.

Este trabalho trata de Sistemas Hipermídia Adaptativos (*Adaptive Hypermedia System*), sendo este definido em [BRU 96] como: “o estudo de sistemas, arquiteturas, métodos e técnicas capazes de promover a adaptação de sistema hipermídia e hiperdocumento em geral aos objetivos, necessidades, preferências e desejos de seus usuários”.

Nora Koch [KOC 2001] explica que “sistemas hipermídia adaptativos são tanto sistemas hipermídia como sistemas adaptativos. Eles combinam o hipermídia com os Sistemas Tutores Inteligentes (STI) guiados pela adaptação da informação, da interface e do *layout* apresentados, ou pelo modo de como as unidades de informação são visualizadas, isto é, como a navegação é proposta”.

Hiperdocumento pode ser definido como o conjunto de elementos de textos interconectados que permite múltiplos caminhos de navegação, com suporte multimídia

(som, imagem e vídeo), podendo ter estruturas mais complexas como documentos estruturados e semi-estruturados. A Web é um hipermídia com maiores limitações de espaço, em que cada página pode ser caracterizada por ser um hiperdocumento.

O hiperespaço definido pela aplicação pode possuir diferentes estruturas: a seqüencial, a hierárquica e a em rede. A estrutura seqüencial é caracterizada por mostrar a informação de maneira linear, a exemplo de glossários e enciclopédias. Na área de educação, essa estrutura foi bastante utilizada pelos sistemas do tipo Instrução Assistida por Computador (*Computer Aided Instruction – CAI*), onde apresentavam restrições quanto aos aspectos cognitivos, pois induziam o aluno a atuar de forma passiva, resumindo a participação do aluno em assimilar respostas ou seguir a técnica de instrução programada [COS 9?]. A estrutura hierárquica é bastante utilizada quando existe uma única página inicial e as outras páginas devem ser organizadas remetendo-se a página inicial, o que pode ser bastante útil para sistemas educacionais, pois ainda existe uma ordem sobre os subtópicos, mas essa ordem desaparece para tópicos vizinhos. Na estrutura em rede, várias páginas podem ser acessadas a partir de uma mesma página, o que pode acarretar uma desorientação por parte do usuário, se o sistema não estiver bem projetado. A figura 2.1 exemplifica as estruturas em questão.

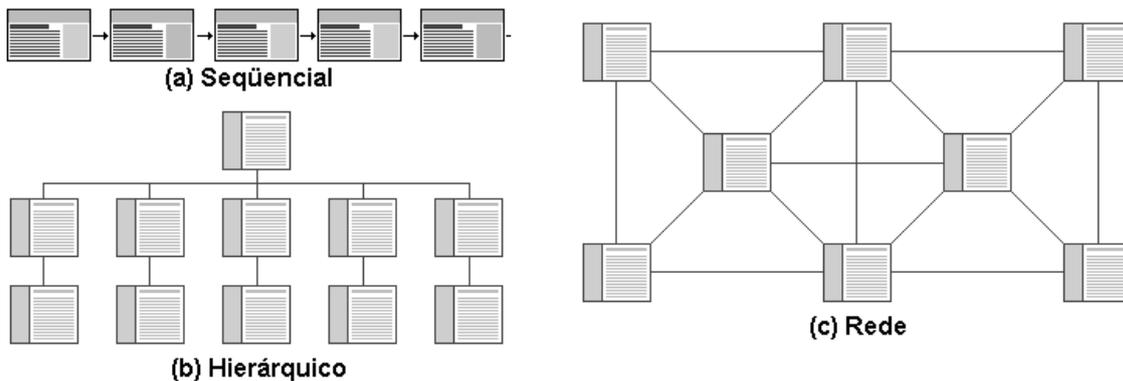


FIGURA 2.1 – Diferentes estruturas para um sistema hipermídia [LYN 2002]

O objetivo fundamental dos STI é proporcionar uma instrução adaptada ao aluno, tanto em conteúdo como na forma, tentando se comportar de forma mais próxima a um professor humano ou um comportamento mais próximo possível disto. Segundo [VIC 2000] os STI são programas que, interagindo com o aluno modificam suas bases de conhecimento, percebem as intervenções do aluno, possuem a capacidade de aprender e adaptar as estratégias de ensino de acordo com o desenrolar do diálogo com o aluno. Caracterizam-se principalmente por construir um Modelo Cognitivo do Aluno, através da interação, e, através da formulação e comprovação de hipótese sobre o estilo cognitivo do aluno, sobre o seu procedimento, o seu nível de conhecimento do assunto e suas estratégias de aprendizagem e na capacidade de formular uma estratégia de ensino-aprendizagem adequada ao aluno e à situação do momento. Para ser inteligente, um tutor deve ser flexível, isto é, ter capacidade para aprender com o meio ambiente e atualizar seu conhecimento [VIC 2000].

De acordo com [BER 96], a arquitetura de um STI pode ser composta por:

- módulo de controle;
- base de conhecimento ou domínio;
- modelo do aluno;
- interface com o usuário;

- estratégias de ensino ou módulo pedagógico, sendo este não utilizado em vários STI.

Desta maneira, um SHA educacional é a união de sistemas hipermídia com os STI.

Os SHA devem satisfazer três critérios: (i) deve ser um sistema hipermídia ou de hipertextos, (ii) deve possuir e manter um modelo do usuário, e (iii) deve ser capaz de se adaptar ao usuário utilizando esse modelo [BRU 96]. A visão clássica do laço de adaptação do sistema hipermídia ao modelo do usuário é mostrada na figura 2.2. O objetivo de um SHA é tornar o sistema hipermídia personalizado de acordo com as características dos usuários. Para isso, os SHA coletam dados do usuário, construindo um modelo do usuário a ser utilizado para adaptar as características dos usuários às suas necessidades.

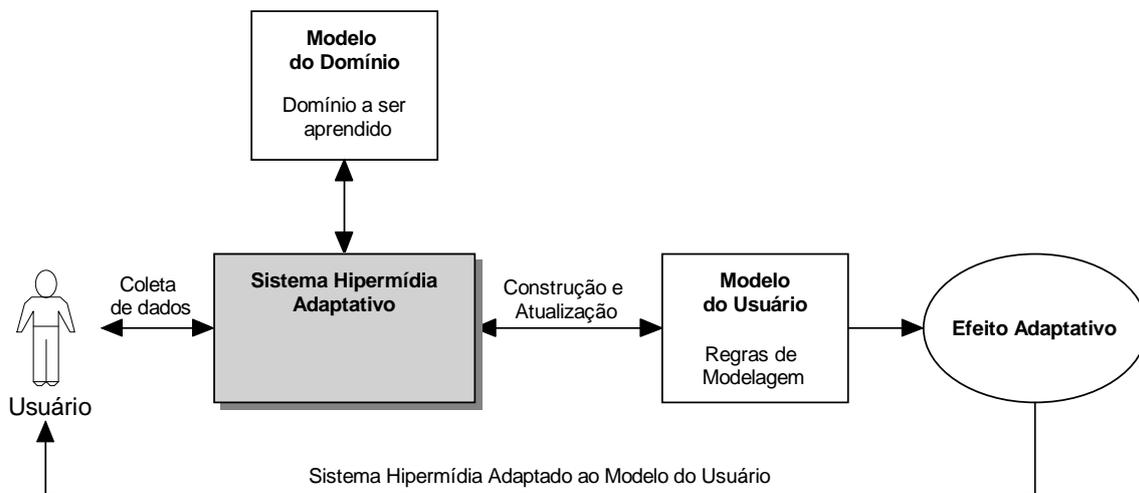


FIGURA 2.2 – Processo de adaptação nos SHA [BRU 96], [PAL 2002]

A idéia por trás dos SHA é a expectativa de oferecer a cada usuário uma interface, uma estrutura, e um conteúdo modelado de acordo com suas características específicas. Em outras palavras, em SHA os usuários acessam interfaces cujos estilo, conteúdo, recursos e *links* serão dinamicamente selecionados entre diversas possibilidades, reunidos e apresentados a eles conforme seus objetivos, necessidades, preferências, desejos, etc [PAL 2002]. Para atingir esse objetivo os SHA se relacionam com diversas tecnologias existentes, como os Sistemas Tutores Inteligentes, que são a base para a investigativa sobre o modelo do usuário (a seção 2.4 dedica-se a explicar este assunto), o estudo de banco de dados para armazenar os dados coletados e atualizados sobre o usuário, e a programação para a Web, cujas ferramentas proporcionam a implementação do SHA.

Sistemas hipermídia adaptativos tentam antecipar as expectativas dos usuários a partir de modelos representando seu perfil. O objetivo geral de um SHA é prover a seus usuários conteúdo atualizado, de seu interesse, com as mídias pertinentes, e sempre observando o modelo do usuário, sendo este uma referência para o sistema, o qual adapta seu ambiente a ele.

A base para classificar um SHA é identificar suas dimensões que serão discutidas nas seções seguintes:

- Onde sistemas hipermídia adaptativos podem ser úteis (seção 2.2);
- Quais características do usuário estão sendo utilizadas como fonte de adaptação (seção 2.4.1);

- O que pode ser adaptado (seção 2.5);
- Como adaptar (seção 3).

2.2 Onde e porque SHA podem ser úteis

Sistemas Hipermídia Adaptativos são especialmente úteis quando há a necessidade de disponibilizar informação seletiva e contextual a usuários com diferentes objetivos e níveis de conhecimento e em situações onde o hiperespaço (rede de nós) é razoavelmente grande. SHA é uma alternativa para a teoria “*one-size-fits-all*” de sistemas hipermídia [BRU 2001a], significando que sempre aparecerá o mesmo conjunto de páginas, a mesma interface e o mesmo conteúdo para os usuários, independente do perfil do usuário, de suas preferências, de seus conhecimentos prévios, etc.

Segundo Brusilovsky [BRU 96], existem seis áreas de aplicações principais onde sistemas adaptativos podem ser utilizados. Estas áreas são:

- Hipermídia Educacionais: Alguns exemplos são INTERBOOK, AHA!, ISIS-Tutor, ELM-ART, TANGOW [KOC 2001], e o AdaptWeb que é o escopo desta dissertação e será explicado nas próximas seções;
- Sistemas de Informação *On-Line*: provêem acesso a informação no hiperespaço. Esse grupo inclui aplicações de *e-commerce*, bibliotecas digitais, catálogos eletrônicos, e todas as outras aplicações de documentação *on-line* [KOC 2001]. São exemplos deste tipo de sistemas PUSH, AVANTI [KOC 2001];
- Sistemas de Ajuda *On-Line*: Sistemas de ajuda *online* são sempre ligados a alguma ferramenta ou sistema, isto é, eles não são sistemas independentes. De certo modo eles podem ser considerados também como sistemas de informação *on-line*, mas o seu objetivo é ajudar o usuário quando este encontra dificuldades com a utilização da ferramenta. Essa ajuda consiste em apresentar a informação necessária, quando requisitada e automaticamente reconhecer quando o usuário estiver precisando de ajuda. Um exemplo deste tipo de sistema é o ORIMUHS [KOC 2001];
- Sistemas de Recuperação e Busca de Informação: São sistemas que combinam as tradicionais técnicas de recuperação de informação com o acesso a hipertextos através de índices e possibilitam a navegação pelo hiperespaço utilizando a similaridade dos *links* entre os documentos. São exemplos: CID, DHS, Adaptive HyperMan, HYPERFLEX, WebWatcher [BRU 96];
- Sistemas de Informação Institucional: São sistemas que suportam o trabalho *on-line* de alguma instituição, como por exemplo um hospital, em que vários dados podem ser acessados e utilizados via Internet [BRU 96]. Um exemplo deste tipo de sistema é o Hynecosum [BRU 96];
- Sistemas para Gerenciar Visões Personalizadas: A Internet oferece uma enorme variedade de informação, e de serviços. Muitos usuários necessitam possuir acesso a alguns conjuntos de sites todos os dias de trabalho. Para eles se protegerem da complexidade de todo o hiperespaço global, eles podem requerer uma visão personalizada do hiperespaço. Um exemplo é o Basar, Information Islands [BRU 96].

Outras novas aplicações começam a ser delineadas na área de serviços personalizados, comércio eletrônico, pesquisa de opinião, agendas coletivas adaptativas,

gestão do conhecimento, comunicação pessoal, etc. [PAL 2002]. Billsus et al [BIL 2002] cita várias tendências para a personalização adaptativa, como para aplicações móveis, que exploram a localização e o horário do usuário, a plataforma utilizada e a largura de banda.

Em cada uma dessas áreas, técnicas hipermídia adaptativas podem ser úteis porque podem resolver o problema de identificação. Especialmente em sistemas educacionais SHA são necessários por duas razões: adaptar a informação apresentada ao nível de conhecimento dos alunos, e prover suporte de navegação nos vários níveis desde comentar os *links* existentes a sugerir o melhor *link* a seguir. Os SHA educacionais devem possuir regras para adaptar o domínio da aplicação de forma que o usuário não se sinta perdido, além de somente apresentar o conteúdo que lhe é indicado no momento. Dentro do SHA, a etapa encarregada de fazer essas avaliações são os módulos que utilizam o modelo do usuário, e adaptam o conteúdo, a apresentação e a navegação deste usuário de acordo com as características verificadas em seu modelo.

2.2.1 SHA para EAD baseada na Web

O contexto de ensino à distância é aquele onde o sistema não é o centro da interação, pois existem os professores e os alunos que também fazem parte dessa interação, e o sistema é visto como uma ferramenta. O sistema deve ser transparente, não podendo assumir o controle absoluto das atividades. O importante para as necessidades do usuário é um direcionamento de como utilizar o *feedback* dado a ele para que possa atingir seus objetivos de aprendizado. Isso pode evitar que o aluno fique desorientado no meio do conteúdo e o ajudar a aproveitar o conteúdo exposto da melhor maneira possível.

Uma das características mais importantes do usuário em sistemas educacionais é o conhecimento que este tem sobre um assunto em questão. Sistemas adaptativos educacionais podem resolver problemas como:

- os níveis de conhecimento dos usuários que utilizam este sistema podem ser diferentes e em alguns usuários este nível de conhecimento pode aumentar mais rapidamente que em outros;
- um mesmo assunto pode não estar muito claro para um aluno-novato e ao mesmo tempo ser trivial para um outro aluno num estágio mais avançado;
- os alunos podem possuir formações diferentes, e não precisar ver os mesmos conceitos para obter o conhecimento;
- o leitor pode estar desorientado, não sabendo onde está, como chegou até ali e como pode fazer para chegar a um outro nó (página).

Usuários de sistemas educacionais têm um objetivo prioritário a atingir (aprender algo). Deve poder navegar dentro dos conteúdos de forma eficiente, tanto no sentido navegacional quanto em termos de alcançar seus objetivos educacionais. Sabendo que navegar dentro do conteúdo envolve os movimentos do aluno visando seus objetivos e informações, uma maneira de tornar essa navegação fácil é estruturar cuidadosamente o conteúdo instrucional a ser aplicado. Vários estudos mostram que o suporte a navegação adaptativa pode aumentar a velocidade de navegação pelo conteúdo e também de aprendizado enquanto que a apresentação adaptativa pode melhorar o entendimento do conteúdo instrucional [MAY 2002].

A principal desvantagem de um SHA aparece quando o sistema não define corretamente suas estratégias para adaptar o conteúdo e a navegação para o aluno, fazendo muitas vezes com que a interface do sistema se modifique com frequência para

atender o modelo do usuário que se modifica a cada nova descoberta sobre este, não obtendo consistência. Assim, quando o usuário se acostuma com a interface e se torna hábil em sua utilização, a interface muda e todo o processo de aprendizado recomeça novamente. Todavia, interfaces adaptativas podem servir para manter homogeneidade para tarefas similares e aumentar a consistência entre as aplicações.

O ambiente AdaptWeb se preocupou em deixar a interface do ambiente das aulas consistente, mesmo que o domínio seja apresentado de forma adaptativa, o aluno ainda assim entende a interface pois as áreas estipuladas para cada função dentro do ambiente nunca serão trocadas de disposição.

2.3 Arquitetura básica de um SHA

Sistemas adaptativos possuem três componentes básicos [WU 2001]: (i) Modelo do usuário; (ii) modelo do domínio e (iii) mecanismo de adaptação.

Modelo do Usuário

O modelo do usuário descreve o usuário para o sistema, ou seja, representa as preferências, os conhecimentos, os objetivos, o histórico navegacional e o nível de conhecimento do usuário (a seção 2.4 dedica-se a explicar este tópico).

Modelo do Domínio

O modelo do domínio descreve como a informação da aplicação é estruturada conceitualmente, ou seja, descreve o domínio da aplicação, utilizando páginas e conceitos definidos pelo autor [WU 2001]. Também pode ser chamado de base de conhecimento do domínio, e é ali que é representado o material instrucional, ou seja, o conteúdo que será ministrado pelo SHA [VIC 2000].

No ambiente AdaptWeb, toda a organização e o armazenamento da base de conhecimento do domínio utiliza documentos XML (*Extensible Markup Language*). Nas páginas referentes ao modelo do domínio, as informações sobre o modelo do usuário, o processo de organização e padronização dos dados provenientes da autoria, bem como para prover mecanismos para a disponibilização de diferentes apresentações de um mesmo conteúdo no AdaptWeb, são implementadas através de documentos XML. XML é uma linguagem extensível, possuindo uma alta flexibilidade para descrição e representação de dados que necessitam de informações agregadas, com mais flexibilidade que HTML (*Hyper Text Markup Language*) por não possuir um conjunto de marcações fixo. Oferece ainda a estruturação de documentos de forma independente da apresentação. Assim, documentos XML possuem uma descrição própria, permitindo que o autor defina sua própria estrutura de informação [AMA 2002].

Mecanismo de Adaptação

O mecanismo de adaptação utiliza o modelo do usuário e o modelo do domínio para prover a adaptação do sistema dinamicamente. A figura 2.3 mostra a arquitetura básica de SHA.

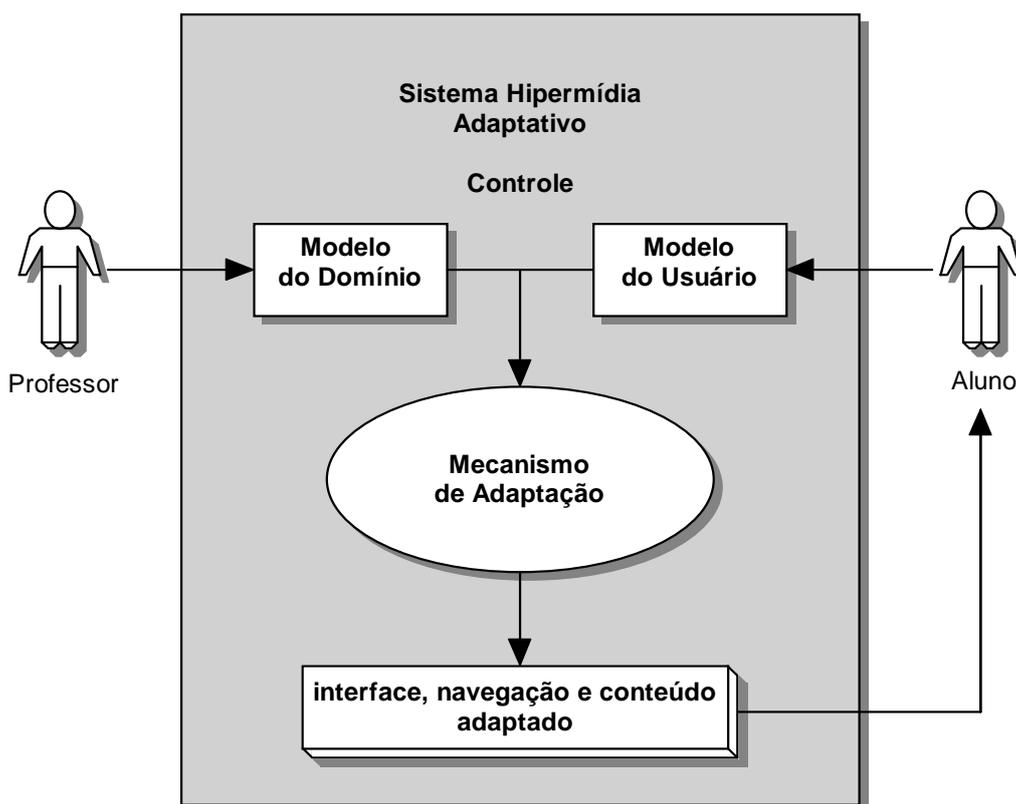


FIGURA 2.3 – Arquitetura de um SHA

Para sistemas adaptativos educacionais ainda pode existir um outro modelo chamado de modelo pedagógico ou modelo de aprendizagem que possui regras que indicam sobre quais circunstâncias pode ser desejável ou não que os usuários sejam guiados para certas partes do domínio da aplicação [WU 98]. O modelo pedagógico ou de estratégia de ensino-aprendizagem também vem dos STI. As estratégias constituem conhecimento sobre como ensinar, ou seja, sobre como gerar, a partir das informações de diagnóstico, monitoração e análise, uma seqüência de táticas de ensino capazes de apresentar com sucesso um determinado tópico a um determinado estudante [VIC 2000].

Existem vários modelos para as táticas de ensino. O modelo de hipertexto é aquele em que o estudante navega numa estrutura de hipertexto e explora o conteúdo a partir de seus interesses e pré-requisitos. Este documento é organizado de tal maneira que cada subdivisão lógica do assunto está ligada com o documento através de diversos tipos de *links*, possibilitando que o aluno navegue por diferentes alternativas para explorar o domínio. O modelo de hipertexto abre excelentes perspectivas para a construção de STI, bem como de SHA, uma vez que pode abrigar, no mesmo documento, diferentes formas de representação de conhecimento. Além disto, o aluno pode trabalhar de forma mais participativa e dinâmica, podendo ser orientado no aspecto pedagógico de forma diversificada.

2.4 Modelo de Usuário em SHA

O modelo do usuário representa toda informação que o sistema conhece sobre o usuário. Este modelo tem sua manutenção feita pelo sistema, apesar do usuário poder realizar modificações em seu perfil, tais como revisar e editar seus dados. Padrões de

ações do usuário e eventos realizados nos vários níveis conceituais, como tarefas completadas e requisições de ajuda, são adicionadas ao perfil do usuário.

O modelo do usuário também é chamado de modelo do aluno pelos SHA educacionais, e também vêm dos STI. Pode-se dizer que o modelo do usuário representa o conhecimento e as habilidades cognitivas do aluno em um dado momento. É constituído por dados estáticos e dados dinâmicos. Contém uma representação do estado do conhecimento do aluno no momento que interage com o sistema [VIC 2000].

No AdaptWeb o dado estático sobre o usuário é a formação (o curso) do aluno. Os dados dinâmicos são: (i) o conhecimento, que pode ser modificado a cada interação com o ambiente, (ii) as preferências navegacionais do aluno, que somente pode ser modificada de uma sessão a outra (o AdaptWeb possui a navegação pelo modo tutorial e pelo modo livre), e (iii) o ambiente de trabalho do aluno, que é a forma de conexão do mesmo.

A partir desse modelo e do conteúdo a ser ensinado, o sistema deve ser capaz de inferir a melhor estratégia de ensino a ser utilizada em seguida. Um modelo realista do aluno implica numa atualização dinâmica à medida que o sistema avalia o desempenho do estudante [VIC 2000].

No AdaptWeb, o módulo de Interface Adaptativa avalia o aluno em seus conhecimentos a cada interação com o mesmo, assim, quando o aluno está navegando pelo modo tutorial, o sistema somente libera os conceitos cujos pré-requisitos foram satisfeitos.

A modelagem do usuário e a interface com o usuário focalizam as necessidades do usuário, geralmente envolvendo uma análise detalhada do domínio de tarefas. Modelagens de usuário para SHA determinam um conjunto de regras para gerar a interface em tempo de execução. No ambiente AdaptWeb, o módulo de Interface Adaptativa verifica quais conceitos os alunos já visitaram, atingindo assim o conhecimento, quais eles devem ser capazes de assistir, e quais ainda não devem ser mostrados ao estudante.

2.4.1 Modelos de descrição

O modelo do aluno, segundo [COS 96 apud VIC 2000], pode ser representado apoiando-se em alguns modelos de descrição, sendo eles: modelo diferencial, modelo de *overlay*, modelo de perturbação, modelo de simulação, modelo de crenças e modelo de estereótipo, que são vistos a seguir.

Modelo Diferencial

No modelo Diferencial “a resposta do aluno é comparada com a base de conhecimento” [VIC 2000]. Esta modelagem compara à performance do especialista/professor com a do estudante e não o conhecimento deles. A modelagem por diferenciação divide o conhecimento em duas classes: a do conhecimento que se espera que o estudante possua e a outra que não se espera que ele tenha. Nesta modelagem o conhecimento do aluno é somente um subconjunto do conhecimento do especialista [VIC 2000].

Modelo de *Overlay* ou superposição

Neste modelo o conhecimento do aluno é representado como um subconjunto da base de conhecimento do sistema tutor [COS 9?]. Isso implica em que a representação

de conhecimento utilizada no modelo do aluno e na base do domínio seja a mesma [VIC 2000]. O modelo é construído por comparação do desempenho do estudante com a ação adotada pelo computador baseada no conhecimento do especialista para o mesmo problema. Os erros neste modelo são creditados à ausência de alguma informação presente na base de domínio [COS 9?].

O modelo de *overlay* é baseado no modelo estrutural do domínio. Geralmente o modelo estrutural do domínio é representado através de uma rede de conceitos do domínio. Os conceitos são relacionados entre si e formam uma rede semântica que representa a estrutura deste domínio. Estes conceitos podem receber diferentes nomes em diferentes sistemas, tais como: tópicos, elementos do conhecimento e objetos, porém todos são considerados como porções elementares do domínio total [BRU 96].

A idéia do modelo de *overlay* é representar o conhecimento dos usuários de maneira individual. Para cada conceito do modelo do domínio, um modelo de *overlay* armazena alguns valores com uma estimativa sobre o nível de conhecimento do usuário sobre este conceito [BRU 96]. Este valor pode ser binário (possui conhecimento / não possui conhecimento), um valor qualitativo (bom / mediano ou regular / pouco ou baixo), ou um valor quantitativo, como a probabilidade do usuário saber o conceito.

O modelo de *overlay* é poderoso e flexível, ele pode de maneira independente medir o conhecimento do usuário em diferentes conceitos [COS 9?]. Foram originariamente desenvolvidos na área de STI e modelagem do aluno [BRU 96]. Em muitos STI o modelo do aluno é apenas um modelo de *overlay* do conhecimento do estudante. Como resultado, na área de interfaces adaptativas, um modelo de *overlay* do conhecimento do aluno é em alguns casos chamado de modelo do aluno.

Modelo de Perturbação ou Buggy

Também relaciona o modelo do aluno com a base de conhecimento do domínio. O modelo de perturbação assume que os erros do aluno são decorrentes da concepção errônea de algum conceito ou ausência dele [VIC 2000]. No modelo de perturbação, existe a base do domínio e uma biblioteca de erros típicos, o modelo do aluno inclui elementos da base do domínio e da biblioteca de erros. Estes modelos permitem um tratamento mais inteligente dos comportamentos incorretos do estudante, mas no entanto, ainda são limitados, principalmente pela estrutura da biblioteca de erros [VIC 2000].

Modelo de Simulação

Este modelo permite prever o comportamento futuro do comportamento modelado, ou seja, prevê a resposta do estudante baseado em seu comportamento [COS 9?].

Modelo de Crenças

Consiste em um conjunto de crenças refletindo o grau que é pensado que o estudante entende sobre um conceito em particular [VIC 2000].

Modelo de Estereótipo

Em geral, classifica o usuário como novato, intermediário ou avançado, em uma área de conhecimento específica. O nível de conhecimento do usuário pode ser representado como um conjunto de pares (tópico,valor), sendo que o usuário pode ter

um ou mais estereótipos ao longo da realização da tarefa [COS 9?]. Por exemplo, o domínio da aplicação sendo o conhecimento em lógica de programação, e na linguagem Pascal. Um usuário pode ser intermediário na lógica e iniciante na linguagem Pascal.

Segundo [BRU 96] a combinação do modelo de *overlay* com o modelo de estereótipo tem gerado bons resultados. O modelo de estereótipo se mostra confiável e é mais simples que o modelo de *overlay*, entretanto menos flexível e poderoso.

Ambiente AdaptWeb

O ambiente AdaptWeb utiliza o modelo de *overlay* e o modelo de estereótipo para representar o aluno. O modelo de *overlay* é utilizado pelo módulo de Interface Adaptativa para representar o conhecimento do aluno. O módulo possui um atributo para cada conceito do domínio da aplicação. Esse atributo pode ser: (i) visitado, que quer dizer que o aluno já possui conhecimento sobre este conceito; (ii) atual, que significa que o estudante está adquirindo o conhecimento no momento; e (iii) não-visitado, que implica que o aluno ainda não possui conhecimento sobre o conceito. Já o modelo de estereótipo é utilizado pelos módulos de Interface Adaptativa e o módulo de adaptação de conteúdo, pois o ambiente foi projetado para que uma mesma disciplina seja disponibilizada por vários cursos, ou seja, um aluno pertence a algum curso, e para cada curso, o professor determina quais conceitos vão existir (ou seja, o professor determina para cada curso, o nível de conhecimento necessário). Para cada conceito pode existir uma lista de exemplos, exercícios e material complementar. Esses exemplos e exercícios são dispostos como fácil, intermediário, difícil ou sem classificação (não possuem classificação).

2.4.2 Quais aspectos podem ser adaptados?

Brusilovsky [BRU 96] identificou cinco características que são utilizadas pelos sistemas hipermídia adaptativos existentes para tratar o usuário como uma pessoa individual ou em grupo de usuários: os conhecimentos do usuário, seus objetivos/metapas, seus fundamentos/base (*background*), suas experiências no hiperespaço, e suas preferências. Outras características foram identificadas por [KOB 2001], como o ambiente de trabalho do usuário e seu estilo cognitivo.

Conhecimento

O conhecimento do usuário é a característica mais importante a ser considerada para adaptação dos sistemas e é largamente utilizado pelas técnicas adaptativas [BRU 96]. Um sistema adaptativo baseado no conhecimento do usuário deve reconhecer as mudanças no estado do conhecimento do usuário e atualizar o modelo do usuário de acordo com as informações obtidas. O conhecimento do usuário é armazenado em seu modelo, e para reconhecer as mudanças do usuário, o sistema deve sempre processar e atualizar este modelo.

Segundo [VIC 2000], muitas técnicas são utilizadas para construir o modelo do aluno, como por exemplo:

- incluir um reconhecimento de padrões aplicados à história das respostas fornecidas por ele;
- comparar a conduta do aluno com a de um especialista e verificar os pontos em comum;
- colocar as preferências do aluno;

- seus objetivos particulares;
- atividades que ele sempre costuma esquecer quando interage com o tutor;
- indicação dos seus objetivos particulares.

O conhecimento do usuário é freqüentemente representado pelo modelo de *overlay*, como nos sistemas Hypadapter, EPIAIM, KN-AHS, ITEM/PG, ISIS-Tutor, ELM-ART, SHIVA, HyperTutor, modelo no qual é baseado no modelo da estrutura do domínio [BRU 96].

Algumas vezes, somente o modelo de estereótipos é utilizado para representar o conhecimento do usuário [BRU 96]. Um modelo de estereótipos pode distinguir vários estereótipos, utilizando para cada dimensão do modelo do usuário um conjunto possível de estereótipos. Por exemplo, o sistema MetaDoc utiliza duas dimensões de classificação e dois conjuntos de estereótipos (principliante, intermediário e avançado). As dimensões são: (1) para representar o conhecimento geral do usuário sobre os conceitos de computação e (2) para representar o conhecimento do usuário sobre UNIX, que é um domínio específico do sistema. Isso pode gerar modelo do tipo intermediário para os conceitos gerais de computação e principliante para UNIX [BRU 96].

Objetivos

Os objetivos e as tarefas dos usuários são características relacionadas com o contexto do trabalho do usuário no hipermídia. Para sistemas educacionais, essa característica significa seus objetivos e metas de aprendizagem, ou seja, o objetivo maior é adquirir o conhecimento. Em qualquer sistema, educacional ou não, uma questão deve ser levada em conta: Porquê o usuário está utilizando o sistema e o que o usuário realmente deseja alcançar [BRU 96].

O objetivo do usuário é a característica considerada mais mutável: pode mudar de sessão para sessão, e freqüentemente se modifica em uma mesma sessão de trabalho. Os objetivos do usuário são normalmente modelados de modo parecido com o modelo de *overlay* para modelar o conhecimento [BRU 96]. Como regra, cada sistema suporta um conjunto de possíveis objetivos e tarefas do usuário que podem ser reconhecidas pelo sistema. Vários sistemas utilizam este procedimento, como os sistemas HyPLAN, ORIMUHS, PUSH, HYPERCASE, Hynecosum, HYPERFLEX [BRU 96].

Background

Por *background* entende-se toda a informação relacionada com a experiência prévia do usuário fora do contexto do sistema hipermídia. Inclui a profissão do usuário, sua experiência de trabalho com as áreas relacionadas, seus pontos de vista e suas perspectivas. Os sistemas EPIAIM, C-Book, e Anatom-Tutor utilizam o *background* no modelo do usuário para aplicar na apresentação adaptativa, já o sistema HyperMan utiliza essa característica para aplicar na navegação adaptativa [BRU 96].

Experiências

Por experiência traduz-se quão familiar está o usuário com a estrutura do hiperespaço e com qual facilidade o usuário pode navegar pelo sistema. A diferença entre experiência e conhecimento é que o usuário pode estar bem familiar com a estrutura do hiperespaço sem conhecer profundamente o assunto a ser tratado. Tanto o *background* quanto as experiências são características utilizadas pelo modelo de estereótipo como nos exemplos MetaDoc, Anatom-Tutor, EPIAIM, C-Book [BRU 96].

Preferências

Por diferentes razões, usuários podem preferir alguns nós e *links* a outros, e algumas partes das páginas em relação a outras. Essas preferências podem ser absolutas (como nos sistemas Hypadapter e Information Islands), ou relativas (sistemas PUSH, e HYPERFLEX), isto é, dependendo do nó e do contexto corrente ou não [BRU 96].

As preferências do usuário diferem das outras características utilizadas pelo modelo do usuário. O usuário precisa informar ao sistema direta ou indiretamente sobre suas preferências. Diretamente por questionários, ou indiretamente através de ações que se repetem durante a utilização do ambiente. Essa característica pode ser considerada mais adaptável do que realmente adaptativa. A diferença é que os sistemas hipermídia adaptativos podem generalizar e adaptá-las em novos contextos. As preferências também podem ser utilizadas para grupo de usuários [BRU 96].

Ambiente de Trabalho

Para sistemas hipermídia adaptativos baseados na Web, a taxa de utilização de *hardware* e *software* é bastante grande. Deste modo, a chance de se construir uma aplicação baseada na Web que sirva para a grande maioria dos usuários tem ficado cada vez menor. A utilização da Web pode ser influenciada por ambos, o *software* e o *hardware* do usuário individual, e pelas características do local atual do usuário [KOB 2001]:

- *Software*: hoje em dia existem vários navegadores, com diferenças cruciais, que podem modificar a apresentação do ambiente para o sistema; várias plataformas; vários *plug-in* que podem ser necessários para a visualização de determinado conceito.
- *Hardware*: a largura de banda é um fator importante, pois existe a diferença de usuários que estão navegando por acesso discado *dial-up* (modem), e outros que possuem conexão direta com o ADSL, além da localização do usuário, pois a largura de banda em certas regiões é bastante diferente de outras. Isso influencia a taxa de *download*, além da satisfação do usuário.

Estilos Cognitivos

Entende-se por Estilo Cognitivo de Aprendizagem (ECA) como sendo um padrão de ações cognitivas que se traduzem em trajetórias que podem ser observadas concretamente, pelo desempenho comportamental observável, o qual é gerenciado pelos processos cognitivos subjacentes. O estudo destas trajetórias é realizado através dos caminhos percorridos pelos aprendizes durante um processo de aprendizagem livre, isto é, sem a interferência de um tutor artificial ou humano [SOU 2000].

Atualmente, existe uma grande preocupação em identificar quais características individuais do aprendiz efetivamente influenciam o seu processo de aprendizagem em ambientes de ensino hipermídia ou via Web [SOU 2002a].

Um exemplo da utilização das características cognitivas do usuário pode ser visto em [CHE 2002], onde o autor utiliza essa característica para planejar as estratégias navegacionais. Quando foram examinadas as ferramentas selecionadas apresentadas aos usuários por pesquisas anteriores, eles se mostraram interessados em encontrar consistência com suas características de diversos estilos cognitivos. Com isso, foi descoberto que o estilo cognitivo do estudante influencia em suas estratégias navegacionais. Estudantes considerados independentes (*field-Independent students*) tendem a navegar livremente pelo ambiente, utilizando um índice, enquanto estudantes

dependentes (*field-Dependent student*) tendem a seguir a seqüência delimitada pelo sistema do começo até o final [CHE 2002].

Ambiente AdaptWeb

O ambiente AdaptWeb utiliza quatro características para serem utilizadas no modelo do usuário: (1) o conhecimento do usuário; (2) suas preferências; (3) seu *background* e (4) seu ambiente de trabalho.

O conhecimento é utilizado para adaptar a navegação do usuário, e é feita pelo módulo de Interface Adaptativa, descrito neste trabalho. Sua preferência também é em relação ao seu modo de navegação. O ambiente possui dois modos de navegação: o modo tutorial e o modo livre. No modo tutorial, o aluno deve navegar pelo ambiente observando a seqüência de conceitos feitas na fase de autoria. Ou seja, o aluno somente pode assistir um conceito, se os conceitos que são pré-requisitos a ele já são conhecidos pelo aluno. No modo livre o aluno pode navegar livremente pelo ambiente. Essa característica também é utilizada pelo módulo de Interface Adaptativa.

O *background* do usuário é observado tanto para adaptar a navegação do usuário, quanto o conteúdo do usuário. No ambiente AdaptWeb, utiliza-se a denominação formação do usuário, já que essa adaptação é feita observando o curso pertencente do usuário. Por exemplo, um aluno do curso de matemática não precisa ver todos os conceitos sobre a linguagem de programação Pascal, ao passo que estudantes do curso de computação precisam entender todos os conceitos sobre essa linguagem. Para adaptar a navegação do usuário, o módulo de Interface Adaptativa utiliza essa característica.

O ambiente de trabalho do aluno é verificado em um outro módulo deste projeto [MAR 2003] para adaptar o conteúdo do material instrucional. O módulo verifica qual é o perfil do usuário, que pode ser conexão lenta (por exemplo *dial-up*) ou conexão rápida (padrão ADSL), e decide se mostra ou não determinadas figuras, vídeos, etc, para o aluno.

2.4.3 Classificação do Modelo do Usuário

Os modelos de usuário (ou modelo do aluno como também são chamados em SHA na EAD) podem ser classificados de acordo com cinco critérios [FRA 91]: grau de especialização, modificabilidade, extensão temporal, método de uso, e número de modelos. Eles serão detalhados a seguir.

Grau de Especialização

Segundo este critério os modelos podem ser divididos em genéricos ou individuais ou ainda respectivamente macro e micro perfil de usuário [LIM 2002], [LIM 2002a]. Modelos classificados como genéricos, também chamados normativos, pressupõem um conjunto homogêneo de usuários. Sendo assim, só devem ser utilizados quando os usuários tenham tanta similaridade que possam ser tratados da mesma forma. Já os modelos individuais possuem informações que correspondem a um dado usuário. Este tipo de modelo possui um problema que é o grande requerimento de memória, ainda mais se a comunidade de usuários que utilizam o sistema for muito grande.

Pode-se também criar um modelo misto, através da utilização conjunta das duas abordagens anteriores. O modelo genérico seria responsável pelas características gerais do grupo geral de usuários, e por sua vez o modelo individual seria responsável pelas

características específicas ou até mesmo as diferenças de um determinado usuário no grupo ao qual pertence.

No AdaptWeb o modelo do usuário é misto pois observa características genéricas como curso que o aluno faz, e individuais, como o conhecimento, preferência de navegação e ambiente de trabalho do aluno.

Modificabilidade

A possibilidade de modificar um determinado modelo de usuário é o que se pode chamar de modificabilidade, o que determina se um modelo é estático ou dinâmico. Modelos dinâmicos são aqueles que são atualizados de acordo com a obtenção de novos conhecimentos, enquanto modelos estáticos ficam inalterados durante toda interação.

O tipo ideal de modelo é aquele que consegue mesclar as duas características descritas, pois de acordo com o tipo da informação a ser armazenada ela deve ser estática, não havendo mudanças no decorrer das interações, ou dinâmicas podendo mudar de sessão para sessão, ou dentro de uma própria sessão, sendo sessão definida como toda a interação do usuário com o sistema, desde seu processo de *login* (autenticação) até seu momento de *logout* (saída do sistema).

Outra denominação para modelos estáticos e dinâmicos são modelos explícitos e implícitos. Modelos estáticos são denominados explícitos, pois o usuário define o seu conteúdo no início de uma sessão e ele permanece do mesmo jeito durante todo tempo. Já os modelos dinâmicos são denominados implícitos, sendo definido pelo sistema à medida que as interações são realizadas durante as sessões.

Muitos dos SHA modelam o usuário de maneira colaborativa, ou seja, o usuário participa do processo da coleta de informações (explícitos – maneira adaptável) e o sistema captura informações referentes a sua navegação pelo ambiente (implícito – forma adaptativa) [BRU 96].

O AdaptWeb pode ser classificado como colaborativo, ou seja, ele é tanto explícito quanto implícito. O módulo de interface adaptativa (tema deste trabalho) realiza a adaptação da forma explícita e implícita, pois o usuário escolhe sua preferência navegacional (modo de navegação) no começo da sessão, logo após sua identificação no sistema (modo explícito), mas o módulo de interface adaptativa é quem adapta o menu de navegação de acordo com o conhecimento do usuário, além de adaptar a navegação pela formação do usuário (modo implícito).

Extensão Temporal

Este critério está relacionado com a durabilidade de um modelo. Modelos a curto termo são criados e armazenados durante toda sessão, e quando chega ao seu final, eles são descartados. Modelos a longo termo são preservados de uma sessão para outra.

O modelo do usuário no ambiente AdaptWeb pode ser classificado como um modelo a longo termo pois o módulo de interface adaptativa monitora o aluno. A cada mudança do aluno, o módulo atualiza o banco de dados, na tabela de *log* do usuário, e o estado no modelo do usuário, assim, quando o usuário utilizar uma outra vez o ambiente, os dados anteriores são observados para mostrar a navegação e a apresentação da interface adaptada.

Método de Uso

A utilização dos modelos de usuário pode ser prescritiva ou descritiva, sendo este último mais tradicional. O modelo prescritivo realiza a simulação do comportamento de um usuário, com a finalidade de prever ou entender suas ações. O modelo descritivo, como o nome já indica, descreve o usuário, podendo ser considerado como uma base de informações sobre o mesmo.

O AdaptWeb utiliza o modelo descritivo, pois observa os dados do usuário para a adaptação, e não tenta simular seu comportamento.

Número de modelos

Múltiplos modelos podem ser necessários para representar um único usuário em algumas situações. “Por exemplo, quando se trata de modelar a familiaridade de um usuário com um sistema operacional. Normalmente os conhecimentos de um sistema operacional são muitos e não homogêneos, podendo ser quebrados em subdomínios. Para cada subdomínio é criado um modelo do usuário pois um só modelo não retrataria tão bem o nível de conhecimento dele nos vários subdomínios. Os sub-modelos são subordinados a um modelo central, usado na troca de subdomínio, que pode ser feita de um momento para o outro, e também para armazenar características do usuário que não mudam de um subdomínio para outro (por exemplo, nome, idade, etc.)” [FRA 91].

No ambiente AdaptWeb existe um modelo de usuário que observa quatro características principais sobre o usuário: seu conhecimento, suas preferências navegacionais, seu ambiente de trabalho e seu curso, e ele é compartilhado entre os módulos para atualizações. O módulo de interface adaptativa observa três destas características para prover a adaptação, sendo elas o conhecimento do usuário, suas preferências e seu curso.

O processo de modelagem do usuário pode ser realizado através de questionário, interação deste com o ambiente e também observando o modelo do domínio. O AdaptWeb começa construindo o modelo do usuário pela sua formação, observando o modelo do domínio, onde o professor fez a autoria dos conceitos relacionados com a formação do usuário.

Posteriormente, o módulo de adaptação de conteúdo verifica qual é essa formação, e faz o questionamento para o usuário quanto ao seu ambiente de trabalho e sua escolha para navegar no ambiente (modo tutorial ou livre). Esses dados são passados para o módulo de Interface Adaptativa, que inicialmente observa se este é um usuário que já entrou alguma vez na disciplina. Se este usuário está entrando no ambiente da disciplina pela primeira vez, o módulo assume seu conhecimento como “vazio”, ou seja, ele ainda não possui nenhum conhecimento. O módulo observa as outras características passadas para adaptar a interface (*layout*) e a navegação deste usuário. Se o usuário já utilizou o ambiente da disciplina, o módulo verifica qual é conhecimento que ele possui até o momento, observando a base de dados, e a cada novo conhecimento adquirido o módulo grava essa atualização na base de dados.

2.5 O que pode ser adaptado em SHA

Generalizando, hipermídia consiste de conjuntos de nós ou hiperdocumentos (chamados aqui por páginas) conectados por *links*. Cada página contém algumas informações locais e um número de *links* para páginas relacionadas. Sistemas hipermídia podem também incluir um índice e um mapa global que provê *links* para todas as páginas acessíveis. Sistemas adaptativos podem se beneficiar da diferenciação

entre adaptação de conteúdo, interface e navegação [KOC 2001]. O conteúdo consiste em trechos de informação inclusos nas aplicações hipermídia como textos, imagens, vídeos, animações. A estrutura representa a organização do conteúdo com a especificação de quais itens poderão ser vistos e como eles serão visitados pela navegação. A apresentação da interface é a visualização do conteúdo, dos elementos interativos e da navegação, que suportam as funcionalidades do hipermídia.

Duas tecnologias diferentes de adaptabilidade são apresentadas por Brusilovsky [BRU 96]: a adaptação do conteúdo das páginas (*content-level adaptation*) e a adaptação de seus *links*, páginas de índices e mapas (*link-level adaptation*). A adaptação ao nível de conteúdo (*content-level adaptation*) e adaptação ao nível de ligamentos (*link-level adaptation*) são mostradas como duas classes de adaptação chamadas respectivamente de **apresentação adaptativa** (*adaptive presentation*) e **suporte de navegação adaptativa** (*adaptive navigation support*).

Outra adaptação possível quanto a apresentação é mudar o *layout* da aplicação (não afetando o conteúdo propriamente dito), como as cores, os tipos e tamanhos das fontes. Essa adaptação pode ser definida como uma outra tecnologia para prover adaptabilidade como adaptação da apresentação da Interface como definido em [KOC 2001]. Este trabalho trata como apresentação adaptativa os dois aspectos: o conteúdo, e a interface. Um esquema das técnicas de adaptabilidade pode ser visualizado na figura 2.4.

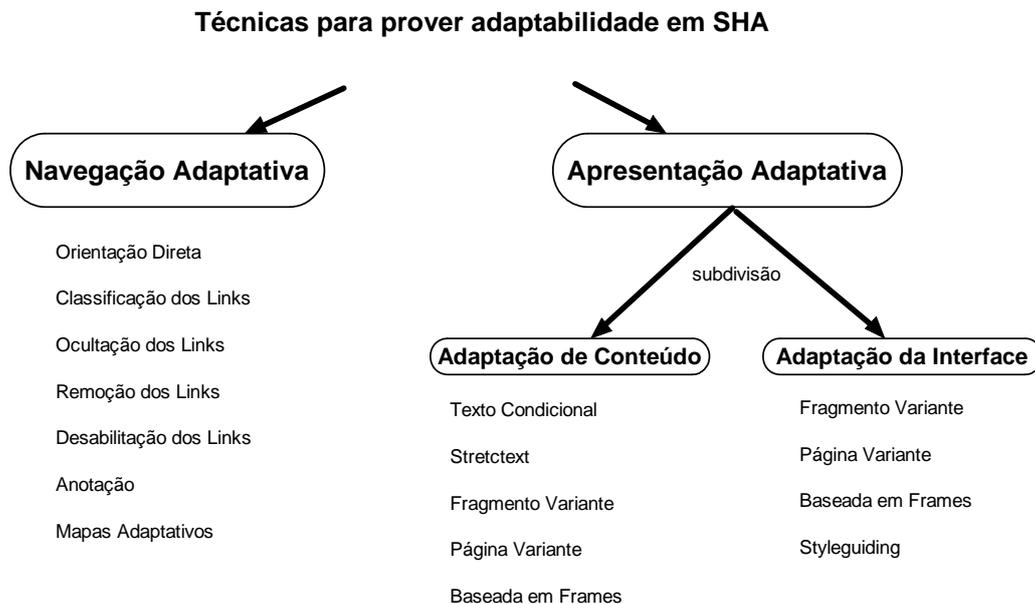


FIGURA 2.4 – Técnicas de adaptabilidade para SHA

3 Métodos e Técnicas Adaptativas para SHA

Nesta seção, considera-se os métodos pelos quais sistemas hipermídia adaptativos podem resolver alguns problemas e descreve-se a maior parte das técnicas empregadas nos sistemas hipermídia adaptativos existentes para implementar esses métodos.

A seguir serão apresentados os métodos e as técnicas para prover apresentação e navegação adaptativa. Este capítulo está fortemente baseado na revisão da literatura, em especial [BRU 96], [BRU 2001], [KOB 2001] e [KOC 2001].

3.1 Métodos e Técnicas para Apresentação Adaptativa

A idéia básica da técnica de apresentação adaptativa é criar várias apresentações adaptando tanto o conteúdo apresentado (*content level*) e a interface do ambiente (*presentation level*) ao modelo dos usuários. Por exemplo, para alunos com maior qualificação (experiente) é apresentado um conteúdo mais detalhado sobre o domínio, que se aprofunde mais no assunto estudado, pois ele já sabe o conteúdo, e uma interface com mais recursos navegacionais; já para alunos iniciantes um conteúdo mais básico e restrições de navegação seria mais apropriado, pois ele poderia ficar perdido com tanta informação, não adquirindo o conhecimento.

3.1.1 Métodos de Adaptação de Conteúdo

O objetivo dos métodos de adaptação de conteúdo é aumentar a usabilidade da aplicação para um amplo grupo de usuários que podem possuir diferentes conhecimentos, *background* e objetivos. Em sistemas hipermídia o conteúdo de um nodo (página) pode não ser somente um texto, mas também som, imagem, música, vídeo, etc. Assim, pode-se encontrar tanto a apresentação adaptativa de textos como de objetos multimídia.

A adaptação de conteúdo consiste essencialmente em prover conteúdos adicionais, comparativos ou alternativos, bem como ocultar partes desse conteúdo para algum grupo de usuário [GAS 2001], [GAS 2002a].

Os métodos de adaptação de conteúdo das páginas podem ser divididos em: conteúdo adicional, explicação adicional, explicação requerida, explicação comparativa, explicação variante e classificação de fragmentos, que são resumidos a seguir.

Conteúdo Adicional (*Additional content*)

Esse método é o mais utilizado para adaptar o conteúdo dos sistemas hipermídia. Ele consiste em mostrar somente as partes relevantes da informação, escondendo as partes irrelevantes, de acordo com os níveis de conhecimento do usuário, seus objetivos, seus interesses ou suas preferências. Esse método pode ser especializado em outros três métodos para serem aplicados aos conceitos (em ambientes educacionais) [KOC 2001]: Explicação Adicional, Explicação Requerida e Explicação Comparativa, que serão detalhadas a seguir.

- Explicação Adicional (*Additional Explanations*): Este é um dos métodos mais populares de adaptação de conteúdo e seu objetivo é ocultar do usuário alguma parte da informação sobre um determinado conceito que não é relevante para o nível de conhecimento ou interesse do usuário. Por exemplo, detalhes de baixo nível podem ser escondidos para usuários com nível de

conhecimento insuficiente para entendê-los. Já para usuários novatos, a explicação adicional é requerida para entender um conceito e para usuários experientes essas explicações podem ficar escondidas (ou ocultas) [BRU 96]. Generalizando, um grupo de usuários pode precisar de informações adicionais, especialmente preparadas para elas, enquanto para outros grupos de usuários essas informações serão ocultadas [PAL 2002]. Esse método é utilizado em sistemas como MetaDoc, KN-AHS, ITEM/IP, EPIAIM e Anatom-Tutor, e sua meta é esconder do usuário algumas partes da informação sobre um conceito particular que não é relevante para os objetivos do usuário (observados no modelo do usuário) [BRU 96].

- Explicação Requerida (*Prerequisite Explanations*): Tanto o método de explicação requerida quanto o de explicação comparativa modificam a informação apresentada ao usuário dependendo do seu nível de conhecimento sobre os conceitos relacionados. O método de explicação requerida é baseado em *links* de pré-requisitos entre conceitos. A idéia consiste em apresentar uma explicação do conceito selecionado somente depois que o sistema apresentar as explicações de todos os conceitos que são pré-requisitos para o conceito em questão, e que ainda não são conhecidos pelo usuário. Esse método é utilizado nos sistemas Lisp-Critic e C-book [BRU 96].
- Explicação Comparativa (*Comparative Explanations*): Este método é baseado na familiaridade de *links* entre conceitos. Se um conceito é similar ao conceito que está sendo apresentado, o usuário recebe uma explicação comparativa (similaridades e diferenças) entre o conceito que está sendo apresentado e seus conceitos relacionados. Esse método é particularmente eficiente para o domínio de linguagens de programação. São exemplos de sistemas que aplicam esse método: ITEM/IP, Lisp-Critic e C-book [BRU 96].

Explicação Variante (*Explanation Variants*)

Também conhecido como Conteúdo Variante (*Content Variant*), esse método assume que somente mostrar ou ocultar certas partes da informação não é suficiente, uma vez que usuários podem precisar de informação essencialmente diferente [KOB 2001]. Com esse método, o sistema armazena várias variantes para um mesmo conceito disponível em uma página, e o usuário recebe a página de acordo com seu modelo de usuário. Por exemplo, quando uma disciplina de programação é ministrada para alunos dos cursos de computação, engenharia e matemática. Apesar de todos aprenderem vários conceitos semelhantes sobre programação, essencialmente os exemplos e programas a serem realizados por cada curso serão totalmente diferenciados, devido à especificidade de cada curso. Esse método é utilizado em sistemas como Anatom-Tutor, Lisp-Critic, Hypadapter, ORIMUHS, SYPROS e WING-MIT [BRU 96].

Classificação de Fragmentos (*Sorting*)

Esse é um método que considera o conhecimento e o *background* do usuário. Ele ordena fragmentos da informação sobre o conceito de modo que as informações sobre o conceito mais importantes para o usuário sejam apresentadas em primeiro plano (de acordo com o modelo do usuário), fazendo deste modo uma hierarquia destes fragmentos. Esse método é utilizado nos sistemas Hypadapter e EPIAIM [BRU 96].

3.1.2 Técnicas de Adaptação de Conteúdo

As principais técnicas para adaptação do conteúdo são: texto condicional, *stretchtext*, fragmento variante, página variante e baseada em frames descritas a seguir.

Texto Condicional (Conditional Text)

Com essa técnica, toda informação possível sobre um conceito é dividida em várias porções (*chunks*) de textos. Cada porção é associada a uma ou mais condições representadas no modelo do usuário. Quando o sistema está apresentando a informação, o sistema mostra somente as porções do texto onde as condições são verdadeiras no modelo do usuário. Essa técnica é bastante flexível, e pode implementar todos os métodos descritos anteriormente, com exceção do método de classificação de fragmentos. Os sistemas ITEM/IP, Lisp-Critic e C-book utilizam essa técnica [BRU 96] e [BRU 2001].

Stretchtext

Essa técnica também permite apresentar ou ocultar porções do texto de acordo com o modelo do usuário. *Stretchtext* é um tipo especial de hipertexto onde os *links* podem ser expandidos para seus conteúdos ou concentrados novamente em uma palavra-chave [BRU 96] e [PAL 2002]. A idéia é apresentar para o usuário uma página em que todas as informações relevantes sejam visualizadas e todas as informações irrelevantes estejam representadas por apenas uma palavra ou frase. Exemplos da utilização dessa técnica são encontrados nos sistemas MetaDoc e KN-AHS [BRU 96].

Fragmento Variante (Fragment Variants)

O método de explicação variante pode ser implementado com as técnicas de fragmento variante e página variante (descrita a seguir). Essa técnica permite que uma mesma página apresente vários conceitos. Cada conceito pode assumir diversos fragmentos variantes e a página é apresentada com a combinação desses fragmentos que melhor satisfaça ao modelo do usuário [KOB 2001]. Um exemplo da utilização dessa técnica está em Anatom-Tutor [BRU 96]. Nesse sistema, uma página não é igual a um conceito como em outros sistemas, e o sistema pode conter explicações de vários conceitos. O sistema armazena as variações de explicações de cada conceito e o usuário recebe os fragmentos que seu modelo de usuário permitir.

Página Variante (Page Variants)

Com essa técnica o sistema mantém duas ou mais variantes de uma mesma página com diferentes apresentações sobre o mesmo conceito. Como regra, cada alternativa é preparada para um possível grupo de usuários (por exemplo pelo seu estereótipo). Essa técnica é utilizada nos sistemas Anatom-Tutor (com diferentes estereótipos), ORIMUHS, WING-MIT e C-Book [BRU 96] e [PAL 2002].

Baseada em Frames (Frame-based approach)

Essa técnica permite a inclusão de toda informação relacionada em um *frame*. Frames podem ser vistos, ocultados, e apresentados de forma alternativa ou de forma ordenada. Essa técnica inclui regras para decidir quais serão os frames apresentados para o usuário. Essa técnica foi empregada nos sistemas Hypadapter e EPIAIM [BRU 96].

Uma outra técnica desenvolvida no projeto PUSH [HOO 96] pode ser considerada a combinação da técnica de *stretchtext* e a técnica baseada em frames. Uma página hipermídia contém entidades ordenadas, e cada objeto se refere a uma coleção de entidades. Cada uma dessas entidades contém uma grande quantidade de hipertexto. Sendo assim, o sistema só deve apresentar as entidades relevantes para o atual objetivo do usuário, a fim de protegê-lo de uma possível sobrecarga de informação. Pode ocorrer que o usuário não se satisfaça com o conteúdo que o sistema decidiu apresentar, então o usuário precisa apenas encapsular e estender o conteúdo de uma entidade clicando no ícone próximo a seu título.

A tabela 3.1 mostra quais as técnicas podem ser utilizadas na implementação de cada método de adaptação de conteúdo [KOC 2001]. O “X” na tabela indica que as técnicas podem ser utilizadas na implementação do método.

TABELA 3.1 – Métodos e Técnicas para Adaptação do Conteúdo

Técnica/ Método	Texto Condicional	<i>Stretchtext</i>	Fragmento Variante	Página Variante	Baseada em Frame
Explicações Adicionais, Requeridas e Comparativas	X	X	X		X
Explicação Variante	X		X	X	X

3.1.3 Métodos e Técnicas de Adaptação da Apresentação da Interface

O objetivo da adaptação da apresentação da interface é adaptar o *layout* de acordo com as preferências ou necessidades dos usuários. Métodos para adaptar a apresentação auxiliam o usuário com o *layout* ou linguagem apropriados. A adaptação consiste em mudanças na apresentação da interface. Algumas dessas mudanças acontecem simultaneamente com a adaptação de conteúdo e de navegação. Métodos e técnicas para adaptação da apresentação da interface são frequentemente agrupados com os métodos e técnicas de adaptação de conteúdo [KOC 2001].

Os métodos mais comuns para adaptar a interface são os métodos de linguagens múltiplas e *layout* variante [KOC 2001] descritos a seguir.

Linguagens Múltiplas (*multi-languages*)

O objetivo do método de linguagens múltiplas é adaptar a linguagem da interface de acordo com as preferências do usuário [KOC 2001].

Layout Variante

Este método inclui todas as possíveis alternativas requeridas em uma apresentação, isto é, cores, tamanho e tipo das fontes, tamanho máximo para imagens, orientação textual, ordenação dos fragmentos do conteúdo, etc.

As mesmas técnicas para adaptação de conteúdo, com exceção da técnica *stretchtext*, podem ser utilizadas também para adaptar a apresentação da interface. Essas técnicas são página variante, fragmento variante e a técnica baseada em frame [KOC 2001].

Além dessas técnicas, a técnica de *styleguiding* é utilizada para implementar os métodos mencionados de adaptação de apresentação da interface. Essa técnica consiste na definição de diferentes guias de estilos que são utilizados alternativamente para o *layout* variante.

A tabela 3.2 mostra quais as técnicas podem ser utilizadas na implementação de cada método de adaptação da apresentação da interface [KOC 2001]. O “X” indica a relação de quais técnicas implementam os respectivos métodos.

TABELA 3.2 – Métodos e Técnicas para Adaptar a apresentação da Interface

Método	Técnica/ Fragmento Variante	Página Variante	Baseada em Frame	em <i>Styleguiding</i>
Linguagens Múltiplas	X	X	X	
<i>Layout</i> Variante	X	X	X	X

Na tabela 3.3 são apresentados diferentes sistemas que possuem a apresentação adaptativa (tanto de conteúdo quanto de interface) e que implementam alguma técnica associada a algum método. Uma descrição detalhada de cada um pode ser obtida em [BRU 96].

TABELA 3.3 – Exemplo de Sistemas que utilizam Métodos e Técnicas de Apresentação Adaptativa

Técnica/ Método	Texto Condicional	<i>Stretchtext</i>	Fragmentos Variantes	Páginas Variantes	Frames
Explicações Adicionais, Requeridas e Comparativas	C-book ITEM/IP Lisp-Critic	MetaDoc KN-AHS PUSH			EPIAIM PUSH
Explicações Variantes	C-book		Anatom-Tutor Lisp-Critic WING-MIT	Anatom-Tutor C-book EPIAIM ORIMUHS SYPROS	Hypadapter
Classificação de fragmento					EPIAIM Hypadapter

3.2 Métodos e Técnicas para Suporte a Navegação Adaptativa

O objetivo da adaptação navegacional é apoiar a navegação prevenindo os usuários de seguir caminhos irrelevantes para suas tarefas e metas [BRU 97].

Antes de comparar as tecnologias para adaptar a navegação de um SHA, deve-se entender como e em qual contexto os *links* são usualmente apresentados. A partir dessa classificação dos *links* em uma rede hipermídia, é possível comparar as técnicas existentes e verificar quais são melhores para cada contexto [PAL 2002].

3.2.1 Classificação dos *links*

São distinguidas quatro formas de apresentação dos *links* do ponto de vista do usuário (isto é, a representação visível e “clacável”¹ das páginas relacionadas por onde o usuário pode navegar) [BRU 96].

Links locais e não-contextuais (Local non-contextual links)

Correspondem aos *links* independentes do conteúdo da página em que se encontram, normalmente apresentados como um conjunto de botões, uma lista ou um menu *pop-up*. Tais *links* são fáceis de manipular e podem ser classificados, ocultos ou anotados [BRU 96] e [PAL 2002].

Links contextuais (Contextual Links)

Compreendem os *links* vinculados ao contexto, tais como os representados por “*hotwords*” em textos, ou “*hotspot*” em imagens, ou seja, *links* que estão embutidos em um contexto e não podem ser removidos. Estes *links* podem ser anotados, mas nunca poderão ser classificados nem totalmente ocultos.

Links para índices e conteúdo de páginas (Links from index and content pages)

Uma página com índices ou tabelas de conteúdos pode ser vista como um tipo especial de página que só contém *links*. Esses *links* são geralmente apresentados em uma ordem fixa, como por exemplo uma ordem alfabética para índices. Tais *links* são assumidos como não-contextuais, a menos que sejam implementados sob a forma de uma imagem.

Links para mapas locais e globais (Links on local maps and links on global hyperspace maps)

Mapas são representações gráficas de um hiperespaço ou área local de um hiperespaço como uma rede de páginas conectadas por setas. O usuário pode navegar diretamente sobre todas as páginas visíveis no mapa.

3.2.2 Métodos para Suporte a Navegação Adaptativa

A navegação adaptativa auxilia usuários a encontrar seus caminhos no hiperespaço através da adaptação da forma de apresentar os *links* aos objetivos, conhecimento e outras características de seus usuários. Essa adaptação pode ser aplicada tanto para *links* contextuais, como para *links* não-contextuais. Suporte de navegação adicional como mapas, tabelas de conteúdos, índices e listas de histórico navegacionais podem ser aproveitados para as técnicas de navegação [GAS 2001], [GAS 2002a].

Os métodos de suporte à navegação podem ser resumidos como: condução global, condução local, suporte à orientação local, suporte à orientação global e visões personalizadas, que serão resumidos a seguir.

¹ Âncoras com especificação de destino para uma outra página.

Condução Global (*Global guidance*)

O objetivo da condução global é ajudar o usuário a encontrar o menor caminho para a informação que ele está procurando ou quer aprender.

A forma mais clara de oferecer condução global é, para cada passo da navegação, sugerir ao usuário os *links* mais apropriados a atingir, a partir da página corrente. Este método é empregado no sistema WebWatcher [BRU 96]. Uma outra forma é a classificação dos *links* que serão sugeridos ao usuário em sua ordem de relevância (observando o modelo do usuário). Os sistemas Adaptive HyperMan e HYPERFLEX [BRU 96] implementam este método.

Sistemas educacionais são um caso especial para a condução global. Em sistemas hipermídia educacionais os usuários geralmente possuem um objetivo global, que é o conhecimento que o estudante deve obter. Portanto, o objetivo não é simplesmente o da informação e sim o do conhecimento. Outra característica específica de sistemas educacionais é que para prover condução global o sistema deve saber não somente o objetivo de aprendizado propriamente dito, mas também o estado corrente de conhecimento do usuário no momento [BRU 96]. SHA educacionais devem capturar a dinâmica do aprendizado em cada um de seus usuários e proporcionar a condução global tendo em vista essa dinâmica, retardando ou acelerando tópicos de acordo com o *feedback* do usuário [PAL 2002].

Condução Local (*Local guidance*)

O objetivo do método de condução local é ajudar o usuário em seu próximo passo de navegação, isto é, encontrar o(s) melhor(es) *link(s)* a seguir a partir da página corrente [KOC 2001]. A condução local sugere os melhores *links* de acordo com as preferências dos usuários, seus conhecimentos e seu *background*, tudo que seja importante para a aplicação. Uma forma de obter a condução local é classificar os *links* de acordo com as preferências do usuário (como nos sistemas Adaptive HyperMan e HYPERFLEX [BRU 96]), ou/e de acordo com seu *background* (como o sistema Adaptive HyperMan).

Suporte à Orientação Local (*Local orientation*)

O objetivo deste método é apoiar o usuário para que este entenda as diferentes possibilidades de navegação de uma determinada posição significativa e ajudá-lo a seguir o *link* apropriado.

Os SHA existentes implementam esse método de duas maneiras distintas: provendo informação adicional sobre as páginas disponíveis de uma página corrente (utilizando a tecnologia de anotação) e limitando o número de oportunidades navegacionais para diminuir a sobrecarga cognitiva deixando o usuário livre para analisar os *links* mais relevantes (utilizando a tecnologia de ocultação que esconde para o usuário toda a informação que não é relevante para seus objetivos em um dado momento).

Uma forma considerada simples sobre este método é mostrar somente os *links* relevantes para as preferências estabelecidas dos usuários. Uma forma mais popularmente empregada é a que considera o objetivo corrente do usuário. Esta maneira foi empregada nos sistemas HyPLAN, Hynecosum, CID e PUSH [BRU 96]. Uma outra forma, adotada nos sistemas HyperTutor e Hynecosum, consiste em mostrar mais *links* para os usuários com maior experiência no hiperespaço. Os usuários novatos vêm

somente um número menor de *links*, que vai aumentando a medida em que sua experiência aumenta [BRU 96] e [PAL 2002].

Dois métodos de suporte a orientação local baseados na tecnologia de ocultação são específicos para hipermídia educacionais. Um deles, muito popular, consiste na ocultação de página em que o usuário ainda não está preparado para aprender (geralmente isso significa que a página corrente possui páginas que são pré-requisitos para ela e que ainda não foram aprendidas pelo usuário). Sistemas que utilizam esse método são: ISIS-Tutor, HyperTutor e Hypadapter. O outro método consiste em ocultar os *links* para página que pertencem a outros objetivos educacionais e estão fora dos objetivos correntes. Isto é feito nos sistemas ISIS-Tutor e SYPROS [BRU 96].

A idéia de métodos de adaptação baseados em anotação é informar ao usuário seu estado corrente sobre as páginas atrás dos *links* visíveis. Quatro métodos foram sugeridos para essa adaptação. Por exemplo, o sistema Hypadapter utiliza três tamanhos de fontes para anotar os *links* como muito relevante, relevante e pouco relevante. A anotação pode refletir vários níveis de conhecimento do usuário sobre as páginas por trás dos *links* anotados. Muitos sistemas utilizam técnicas para implementar esse método distinguindo três níveis de conhecimento sobre uma página: “não-conhecido”, “em-trabalho” e “aprendido”. O segundo método consiste em refletir os diversos níveis de conhecimento do usuário acerca dos nodos. Outros dois métodos utilizam anotação onde a ocultação também é utilizada. Um deles consiste em sublinhar os *links* que estão diretamente relacionados com o objetivo corrente. A outra forma é provendo anotação especial para os *links* que o usuário ainda não está preparado para aprender. São exemplos desse método o ITEM/PG e o ISIS-Tutor, e o sistema ELM-ART que utiliza ícones vermelhos para os itens que o usuário ainda não está preparado [BRU 96].

Suporte à Orientação Global (*Global orientation*)

O objetivo deste método é apoiar o usuário a entender a estrutura de todo o hiperespaço e sua posição dentro dele. Em sistemas não-adaptativos isto é obtido por marcas visuais e mapas globais que auxiliam o usuário a se localizar em relação ao contexto global. SHA podem oferecer um suporte maior para o usuário utilizando as tecnologias de ocultação e anotação. A anotação funciona como uma marca pois os usuários podem perceber mais facilmente as páginas que ele já estudou e assim reconhecer onde está no hiperespaço. A ocultação reduz o número de *links* visíveis no hiperespaço e pode simplificar o aprendizado e auxiliar na orientação.

Exemplos de sistemas educacionais que utilizam a ocultação para conceitos que ainda não estão prontos para serem aprendidos pelo usuário são ISIS-Tutor, HyperTutor e Hypadapter. Um outro método para suporte à orientação global é aumentar gradualmente o número de *links* visíveis na medida em que vai crescendo a experiência do usuário no hiperespaço considerado. Isto ocorre nos sistemas HyperTutor e Hynecosum.

Uma direção interessante para o método de suporte a orientação global é a adaptação de mapas globais e locais, porém essa pesquisa ainda está sob investigação. Genericamente, todos os métodos de ocultações e anotações podem ser aplicados para adaptar mapas, e um exemplo desse método pode ser observado no sistema HYPERCASE.

Visões Personalizadas (*Personalized views*)

Esse método consiste em gerar e atualizar o hiperespaço com a visão personalizada do usuário, ou seja, é uma forma de construir interfaces de trabalho personalizadas por meio de adaptação. Estas visões são necessárias em ambientes dinâmicos na *Web*, onde itens podem aparecer, desaparecer e evoluir. Um sistema que utiliza esse método é o BASAR, que emprega agentes inteligentes para coletar e manter um conjunto de *links* relevantes para cada um dos seus usuários observando o objetivo dos mesmos. Os agentes executam pesquisas regularmente por novos itens relevantes para cada usuário e verificam também itens que não existem mais ou atualizam os mesmos baseando-se no modelo do usuário.

3.2.3 Técnicas para Suporte a Navegação Adaptativa

As técnicas para a navegação adaptativa manipulam âncoras e *links* com o propósito de adaptar a navegação dinamicamente sobre as características dos usuários encontradas no estado corrente no modelo do usuário, isto é, as técnicas são utilizadas para implementar os métodos relacionados anteriormente.

O suporte à navegação adaptativa oferece algumas técnicas que podem ser classificadas de acordo com o modo que elas adaptam a apresentação dos *links*. Elas podem ser: orientação direta, classificação de *links*, ocultação de *links*, remoção de *links*, desabilitação de *links*, anotação e mapas adaptativos.

Orientação Direta (*Direct guidance*)

O usuário possui somente uma opção para continuar a navegação, isto é, somente uma âncora ou botão para navegar para a “próxima” página. O destino (conceito) é escolhido pelo sistema de acordo com características do modelo do aluno.

O sistema decide qual a próxima “melhor” página para o usuário visitar de acordo com seus objetivos e outros parâmetros representados no modelo do usuário. Porém, a orientação direta possui o problema de ser muito direcionado e quase não fornecer suporte ao usuário que gostaria de fazer suas próprias escolhas ao invés de apenas seguir as sugestões do sistema. Além disso, essa técnica não permite ao usuário tomar decisões sobre seu caminho através do material, não permitindo a este usuário ter uma figura clara da estrutura hipermídia, o que mostra que essa técnica deve ser aplicada junto a outras técnicas [BRU 96].

Para oferecer orientação direta o sistema pode destacar visualmente o *link* para a melhor página, como é feito no WebWatcher ou apresentar um *link* dinâmico adicional (geralmente denominado *next* ou *continue*) que é conectado a melhor página selecionada, como no sistema ISIS-Tutor [PAL 2002] e [WU 98]. A figura 3.1 mostra o sistema ISIS-Tutor. A seta indica a orientação direta, que mostra a “melhor” próxima página para o aluno (*page down*).

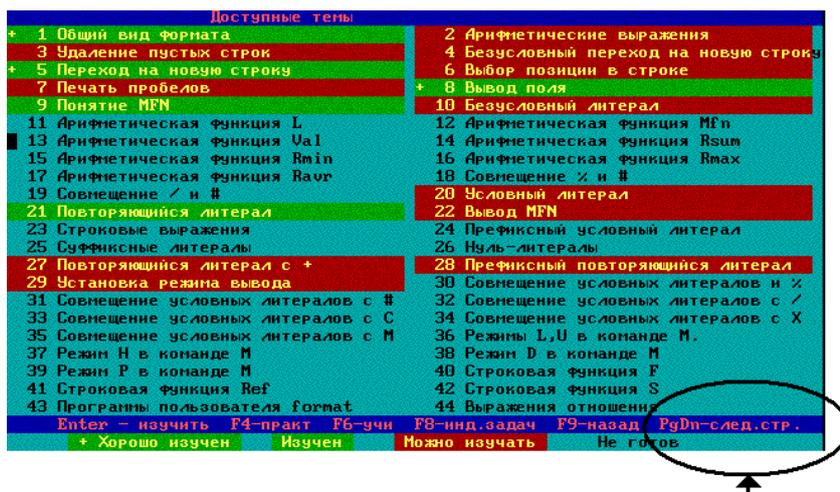


FIGURA 3.1 – Orientação Direta no ISIS-Tutor [PES 96]

Classificação de Links (Sorting of Links)

Também chamada de classificação adaptativa ou ainda ordenação adaptativa, essa técnica consiste em ordenar o conjunto de âncoras para os *links* que são apresentados em ordem decrescente de relevância para o usuário. A desvantagem dessa ordenação adaptativa é que para cada vez que o usuário entre na mesma página (mesmo conceito), a ordem dos *links* pode estar diferente.

É uma técnica tipicamente usada em aplicações onde se deve gerar uma lista de *links* para páginas. Cada um dos *links* pode representar um item de uma lista, assim sendo considerado como um fragmento. A apresentação da lista de *links* torna-se então um problema de selecionar fragmentos de uma lista maior e agrupá-los por sua classe (ou tipo). De maneira geral, duas áreas de aplicações podem utilizar essa técnica: sistemas de recuperação de informações classificam *links* para páginas de acordo com um critério relevante; e aplicações educacionais onde o usuário seleciona um objetivo de aprendizagem e o sistema gera uma lista de páginas para estudar, ordenada de acordo com o relacionamento com os pré-requisitos.

A idéia da tecnologia da ordenação adaptativa é organizar todos os *links* de uma página de acordo com o modelo do usuário usando alguma maneira fácil de transmitir isto ao usuário. Ordenação adaptativa tem uma aplicabilidade limitada: pode ser usada com *links* não contextuais, mas não pode ser facilmente usada para índices e conteúdos de página (os quais geralmente já têm uma ordem estabelecida de *links*), e nunca pode ser usado com *links* contextuais e mapas. Ordenação adaptativa pode não fornecer uma interface estável para os usuários trabalharem nela.

O sistema Adaptive Hyperman implementa a técnica de ordenação de acordo com algumas características de entrada do usuário como seu perfil, sua profissão e experiências de trabalho, seus objetivos, nó corrente de interesse, etc, e retorna para o usuário como saída, um conjunto de documentos relevantes pela entrada provida. Para calcular a relevância dos documentos, o sistema utiliza uma rede de relevância que armazena as preferências individuais dos usuários. Para prover *feedback*, o usuário avalia o conjunto de nós selecionados pelo sistema como “relevantes” ou “não-relevantes” para o conjunto de entrada, pelo seu ponto de vista. Esse *feedback* é utilizado pelo sistema para atualizar a rede individual de relevância, e como consequência, o sistema gradualmente aprende as preferências do usuário [MYK 92]. A

figura 3.2 mostra o sistema, na fase em que o usuário avalia os *links* apresentados pelo sistema.

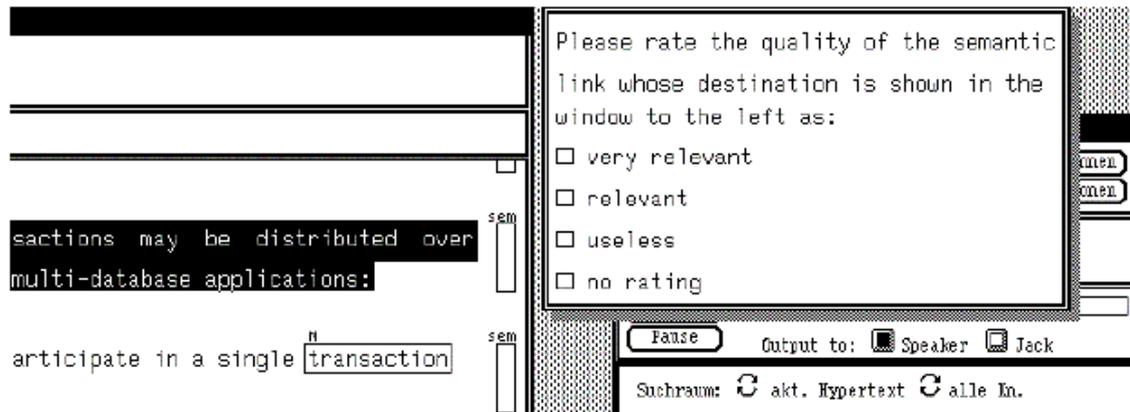


FIGURA 3.2 – Sistema Adaptive Hyperman [MYK 92]

Ocultação de Links (Link Hidding)

A tecnologia de suporte de navegação através da técnica de ocultação de *links* (ou simplesmente ocultação) consiste em restringir o espaço de navegação ocultando os *links* de páginas consideradas irrelevantes (de acordo com o modelo do usuário). Uma página pode ser considerada irrelevante por diversos motivos. Talvez ela não esteja relacionada ao objetivo do usuário atual ou ela pode apresentar material que o usuário ainda não esteja preparado para entender. Ocultar possui uma ampla aplicabilidade, protege os usuários da complexidade de um hiperespaço irrestrito e reduz sua carga cognitiva na navegação. Os *links* podem ser apresentados como textos “normais” [DEB 2000]. A cor do âncora padrão utilizada pelo sistema AHA é a preta para *links* indesejáveis, correspondendo a ocultação de *links* [DEB 2000].

Para efeito de exemplificação, [SPE 99] mostra um experimento onde o sistema primeiro não fez adaptação alguma, ou seja, onde tinha *link* para páginas, este aparecia. Depois o sistema testou a adaptação com a anotação de *links*, a ocultação de *links* (chamada no experimento de *links* incrementais), e por último testou as duas técnicas em conjunto. A figura 3.3 mostra os experimentos realizados no ambiente AST [SPE 99].

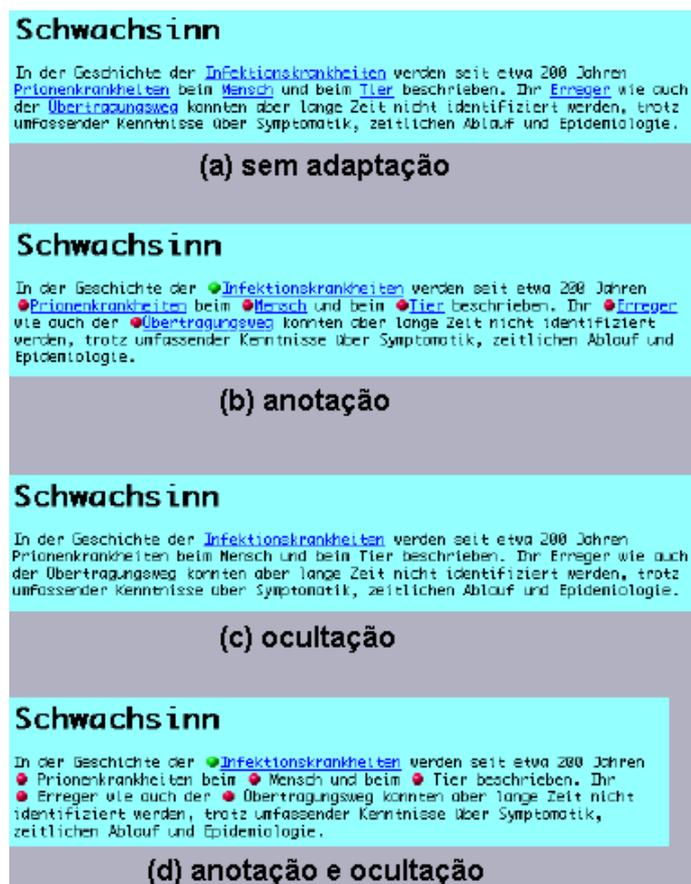


FIGURA 3.3 – Exemplo do Sistema AST e suas variações das técnicas de adaptação

Os resultados dos testes com usuários mostraram a imprevisibilidade de uma interface com a adaptação da ocultação isoladamente. Essa técnica pode ter um impacto negativo no suporte global a navegação. Já a utilização da técnica de ocultação com a de anotação mostrou que as marcas da técnica de anotação onde os *links* podem aparecer ou não, reconcilia a violação do requerimento de previsibilidade requerida em Interfaces Humano-Computador (IHC) [SPE 99].

Remoção de Links (Link Removal)

A remoção de *links* consiste em simplesmente excluir *links* que não sejam relevantes aos interesses do usuário ou que o usuário ainda não esteja preparado para visualizá-lo (pré-requisitos). Esta técnica é freqüentemente combinada com a de classificação de *links*, sendo que apenas os primeiros *links* (mais relevantes) serão mostrados, os demais sendo removidos da página.

O sistema ISIS-Tutor utiliza essa técnica [PES 96]. A figura 3.4 mostra o sistema sem a utilização (a) e com a utilização (b) da técnica de remoção de *links*

Доступные темы	
+ 1 Общий вид формата	2 Арифметические выражения
3 Удаление пустых строк	4 Безусловный переход на новую строку
+ 5 Переход на новую строку	6 Выбор позиции в строке
7 Печать пробелов	+ 8 Вывод поля
9 Понятие MFN	10 Безусловный литерал
11 Арифметическая функция L	12 Арифметическая функция Mfn
13 Арифметическая функция Val	14 Арифметическая функция Rsum
15 Арифметическая функция Rmin	16 Арифметическая функция Rmax
17 Арифметическая функция Ravr	18 Совмещение % и #
19 Совмещение / и #	20 Условный литерал
21 Повторяющийся литерал	22 Вывод MFN
23 Строковые выражения	24 Префиксный условный литерал
25 Суффиксные литералы	26 Нуль-литералы
27 Повторяющийся литерал с +	28 Префиксный повторяющийся литерал
29 Установка режима вывода	30 Совмещение условных литералов и %
31 Совмещение условных литералов с #	32 Совмещение условных литералов с /
33 Совмещение условных литералов с C	34 Совмещение условных литералов с X
35 Совмещение условных литералов с M	36 Режимы L,U в команде M.
37 Режим H в команде M	38 Режим D в команде M
39 Режим P в команде M	40 Строковая функция F
41 Строковая функция Ref	42 Строковая функция S
43 Программы пользователя format	44 Выражения отношения
Enter - изучить F4-практ F6-учи F8-инд.задач F9-назад PgDn-след.стр.	
+ Хорошо изучен Изучен Можно изучать Не готов	

(a) ISIS-Tutor sem a utilização da técnica de remoção de links

Доступные темы	
+ 1 Общий вид формата	2 Арифметические выражения
3 Удаление пустых строк	4 Безусловный переход на новую строку
+ 5 Переход на новую строку	6 Выбор позиции в строке
7 Печать пробелов	+ 8 Вывод поля
9 Понятие MFN	10 Безусловный литерал
13 Арифметическая функция Val	20 Условный литерал
21 Повторяющийся литерал	22 Вывод MFN
27 Повторяющийся литерал с +	28 Префиксный повторяющийся литерал
29 Установка режима вывода	52 Размещение первой строки поля
53 Выбор длины фрагмента поля	54 Выбор смещения фрагмента поля
55 Вывод подполя	56 Повторяющиеся группы
Enter - изучить F4-практ F6-учи F8-инд.задач F9-назад	
+ Хорошо изучен Изучен Можно изучать	

(b) ISIS-Tutor com a utilização da técnica de remoção de links

FIGURA 3.4 – ISIS-Tutor sem (a) e com (b) a utilização da técnica de remoção de links

Desabilitação de Links (Link disabling)

Essa técnica é baseada na idéia de que a funcionalidade de um *link* pode ser removida. Essa tecnologia possui maior usabilidade quando utilizada em combinação com a técnica de ocultação de *links* ou a técnica de anotação que será explicada a seguir [WU 98].

Quando se utiliza a técnica de desabilitação de *links* remove-se a âncora de *link*, no caso de ambientes hipermídia para Web baseados em HTML (*HyperText Markup Language*), a tag <A> que torna os *links* acessíveis. Assim, o texto é apresentado como *link* visível ao usuário mas se o mesmo tentar clicar no *link* não causará efeito algum.

Anotação (Link Annotation)

Ela incrementa *links* com um comentário que provê ao usuário informações sobre o estado corrente dos nós atrás dos *links* comentados. Essa tecnologia tem se mostrado especialmente eficiente em ambientes hipermídia educacionais. O comentário dos *links* pode ser fornecido na forma textual ou na forma de dicas visuais, por exemplo, usando ícones diferentes, ou cores, tamanho ou tipos de letras diferentes entre outros.

A anotação pode ser usada com todos os quatro tipos de *links* considerados. Seu uso mantém uma ordenação estável dos *links* e em geral é uma técnica mais poderosa do que a ocultação, na medida em que esta última oferece somente dois estados possíveis para os *links* (relevante/visível e não-relevante/oculto), enquanto que a anotação adaptativa pode estabelecer diversos níveis de relevância. Além disso, a anotação pode simular a ocultação simplesmente obscurecendo os itens considerados não-relevantes no contexto considerado. O obscurecimento pode reduzir em alguma extensão a sobrecarga cognitiva (o usuário pode aprender a ignorar os *links* obscurecidos) mas estes se manteriam ainda visíveis (e operacionais, se for necessário) [PAL 2002] e [WU 98]. São encontrados três diferentes métodos para decidir pela anotação [WU 98]:

1. A anotação pode indicar relevância entre os *links*. Cores podem ser utilizadas para distinguir, por exemplo os *links* “mais relevantes”, “relevantes” e “não-relevantes”.
2. Uma anotação pode indicar se o usuário já possui conhecimento sobre um conceito descrito em uma página em que é demonstrado por um *link*. Diversos níveis de conhecimento podem existir, como por exemplo *links* “não-preparado-para-ser-aprendido” pelo usuário (isto significa que os conceitos pré-requisitos a este ainda não foram aprendidos), “preparado-para-ser-aprendido”, “em aprendizagem”, “aprendidos” pelo usuário e “muito-bem-aprendidos” pelo usuário.
3. Uma anotação também pode indicar se o usuário está capacitado a entender a informação contida na página destino (como por exemplo “pronto para ler” ou “não-pronto”).

Todas as três formas de representação descritas podem ser utilizadas juntas, fazendo com que o sistema seja capaz de discernir entre diferentes tipos de *links* visuais. São exemplos do uso dessa técnica os sistemas ITEM/PG, ISIS-Tutor e o ELM-ART.

O sistema Interbook apresenta a técnica de anotação mostrando as marcações verde para conceitos “recomendados”, vermelho significa “não-prontos para serem aprendidos”, branco é “nada de novo”, e um *check* (sinal de conferido) significa “visitado” [BRU 98a]. A figura 3.5 apresenta uma tela do Interbook.

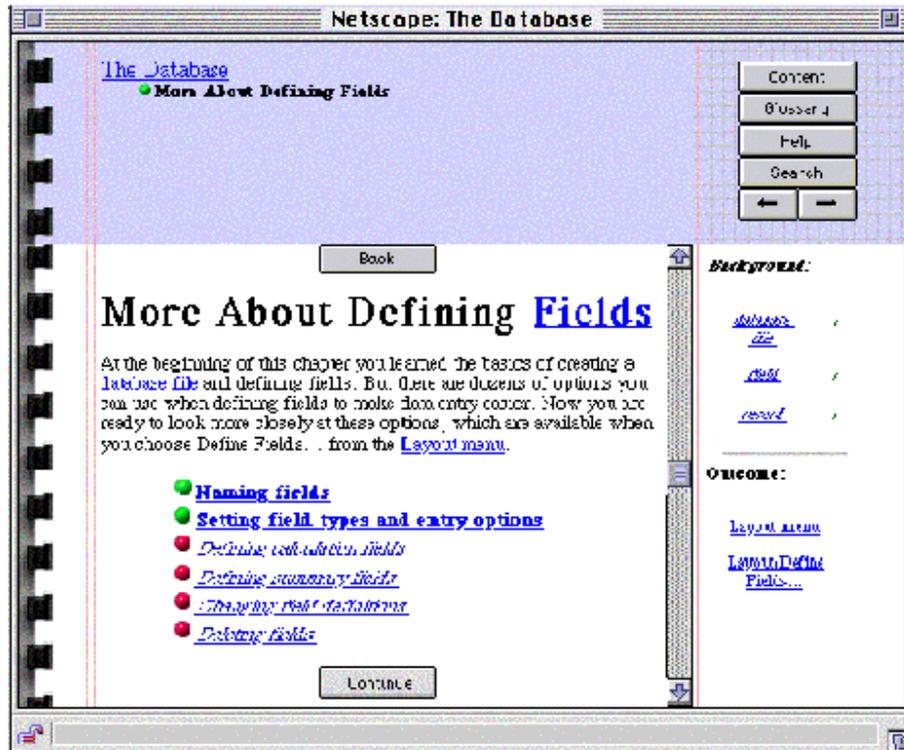


FIGURA 3.5 – A utilização da técnica de anotação no sistema Interbook

O sistema ELM-ART também utiliza a técnica de anotação. A figura 3.6 mostra uma tela do sistema. Marcação vermelha e itálico significam “não-pronto” para ser aprendido, marcação verde e negrito quer dizer “pronto” e “recomendável” do ponto de vista de atingir os objetivos do usuário, e marcação amarela quer dizer “pronto” mas “não-recomendável” [BRU 97].

3.1 Prädikate

In der letzten Lektion wurden die Datentypen von LISP vorgestellt. Um kompliziertere Funktionen in LISP schreiben zu können, muß oft bekannt sein, welchen Datentyp ein Ausdruck besitzt. LISP stellt hierfür eine Anzahl von Funktionen zur Verfügung, die Prädikate genannt werden. Prädikate sind also Testfunktionen, die testen, ob ein Ausdruck eine bestimmte Eigenschaft besitzt oder nicht, z.B. ob ein Ausdruck ein Atom ist oder nicht. Als Wert liefert ein Prädikat T für 'wahr', wenn der Ausdruck die Eigenschaft besitzt und NIL für 'falsch', wenn er sie nicht besitzt.

- *LCONS und ATOM*
- LIST
- *MEMBER und EQ*
- EQUAL
- KALT-P (Beispiel)
- ENDF
- ZEROP
- MEMBER
- *Aritmetische Prädikate*
- GROESSER 15 P (Beispiel)
- Zusammenfassung Prädikate
- PALINDROMP (Aufgabe)
- *HAUT (Aufgabe)*
- MEHR-ALS-ZWEI-P (Aufgabe)
- NREST (Aufgabe)

FIGURA 3.6 – Sistema ELM-ART

O sistema ELM-ART II também utiliza marcações coloridas. A marcação verde significa que a página está “pronta” e é “recomendada” para ser visitada, e os conceitos estão “prontos” para serem aprendidos. Isso significa que os pré-requisitos para este conceito já foram aprendidos. A marcação vermelha indica que a página “não está pronta” para ser visitada. Neste caso, quer dizer que pelo menos um pré-requisito ainda não é conhecido pelo usuário. A marcação amarela pode assumir diferentes significados, dependendo dos tipos das páginas que os *links* indicam. Por exemplo, no caso de páginas terminais com um teste ou problema a ser resolvido a marcação indica que o teste ou o problema foi resolvido corretamente. No caso de páginas conceito, a marcação amarela significa que a página já foi visitada. A marcação laranja quer dizer que o sistema inferiu que aquela página contém conteúdo já conhecido pelo usuário pois o usuário já tem o conhecimento de outras páginas com o mesmo conteúdo [SPE 97]. Na figura 3.7 são apresentadas as marcações no sistema ELM-ART-II em uma determinada tela da aplicação.



FIGURA 3.7 – O sistema ELM-ART II

Mapas Adaptativos (Map Adaptation)

A técnica de mapa adaptativo consiste da combinação de outras técnicas, com a diferença de ser aplicada para uma visualização gráfica da estrutura navegacional. As técnicas de orientação direta, anotação e ocultação podem ser utilizadas para adaptar mapas em SHA, mas essas técnicas não modificam a forma ou a estrutura dos mapas. Um exemplo da utilização de mapas adaptativos é o sistema HYPERCASE.

As técnicas de adaptação navegacional reduzem o espaço de navegação tanto eliminando âncoras (orientação direta, ocultação, desabilitação e remoção de *links*), ou guiando o usuário para um número reduzido de âncoras (anotação e classificação de *links*). Uma limitação específica para cada usuário (observando seu modelo) previne o usuário de ficar perdido no hiperespaço [KOC 2001].

A tabela 3.4 mostra algumas técnicas combinadas com cada método de adaptação navegacional [KOC 2001]. O “X” indica a relação entre eles.

TABELA 3.4 – Métodos e Técnicas para Adaptação Navegacional

Técnica Método	Orientação Direta	Anotação	Ocultação, Remoção, e Desabilitação	Classificação
Condução global	X		X	X
Condução local	X	X	X	X
Suporte a orientação global		X		
Suporte a orientação local		X	X	
Visões personalizadas	X	X	X	X

Na tabela 3.5 são apresentados diferentes sistemas que possuem a navegação adaptativa e que implementam alguma técnica associada a algum método [BRU 96].

TABELA 3.5 – Exemplos de Sistemas que utilizam Adaptação Navegacional

Técnicas Métodos	Orientação Direta	Classificação	Ocultação	Anotação	Mapas Adaptativos
Condução Global	WebWatcher ITEM/IP HYPERFLEX	HyperMan HYPERFLEX			
Condução Local	HyperTutor	HyperMan Hypadapter HYPERFLEX	Hypadapter PUSH	ISIS-Tutor ELM-ART	Hypercase
Orientação Local		Hypadapter	HyperTutor Hypadapter ISIS-Tutor Hynecosum HyPlan ISIS-Tutor PUSH SYPROS	ISIS-Tutor ELM-ART ITEM/PG ISIS-Tutor ELM-ART ITEM/PG	Hypercase
Orientação Global			Hynecosum HyperTutor ISIS-Tutor SYPROS	ISIS-Tutor ELM-ART ITEM/PG	Hypercase

A seguir, no capítulo 4, o ambiente proposto AdaptWeb será apresentado, desde sua arquitetura, até as técnicas de adaptação utilizadas.

4 Ambiente AdaptWeb

O AdaptWeb² (Ambiente de Ensino-aprendizagem Adaptativo na *Web*) é um sistema hipermídia adaptativo de educação a distância baseado na *Web* que tem como finalidade adaptar o conteúdo, a apresentação e a navegação de acordo com o modelo do usuário. O foco deste projeto é o desenvolvimento de um ambiente para a autoria e apresentação de cursos na *Web*, com condições de adaptabilidade. O projeto é desenvolvido por um consórcio de pesquisa envolvendo a Universidade Federal do Rio Grande do Sul (UFRGS) e a Universidade Estadual de Londrina (UEL) [GAS 2002].

O AdaptWeb é um ambiente hipermídia adaptativo para *Web* com a finalidade de disponibilizar um mesmo conteúdo para aprendizes de grupos distintos. Este ambiente propõe adaptabilidade nos dois níveis: apresentação adaptativa e suporte de navegação adaptativa, permitindo a adaptação no conteúdo a ser apresentado, na interface do ambiente da disciplina e na navegação, apresentando ou não informações ao usuário com base em seu modelo.

Boa parte dos sistemas disponíveis para o ensino na *Web* continua utilizando-se do paradigma de cursos baseados em material impresso transformado em páginas HTML, onde recursos de adaptabilidade são muito pouco utilizados. Muitos trabalhos de pesquisa buscam a adaptação dos materiais didáticos aos diferentes estilos de aprendizagem dos alunos [BRU 98], [BRU 99], [ROU 99], [SOU 2002]. A diversidade de preferências e capacitação dos alunos demanda técnicas adaptativas para suportar ambientes EAD na *Web* [BRU 98]. No AdaptWeb, um mesmo programa de disciplina pode ser adaptado a diferentes cursos e a diferentes características de alunos.

Neste projeto, a diversidade dos estudantes é suportada pela criação de um modelo flexível do estudante (modelo do usuário), onde, para cada aluno, são armazenadas informações sobre o curso, conhecimento, ambiente de trabalho, preferências e histórico navegacional.

Os conteúdos educacionais do ambiente são modelados através de uma estrutura hierárquica de conceitos onde são estabelecidos critérios de pré-requisitos. Esta estrutura é definida durante a fase de autoria e posteriormente armazenada no formato XML (*Extensible Markup Language*). A partir do(s) documento(s) XML, são geradas diferentes apresentações de uma disciplina, baseado nas características do aluno representadas em seu modelo. A divisão do material instrucional pode ser organizada em quatro categorias: conceito, exemplo, exercício e material complementar.

Os documentos XML resultantes da etapa de autoria devem passar por um processo de filtragem, antes de serem apresentados ao aluno. Os filtros acontecem dinamicamente durante a interação do aluno no ambiente e obedecem aos critérios de adaptação representados no modelo deste aluno. O curso (formação do aluno), o conhecimento, as preferências navegacionais e o ambiente de trabalho do aluno são as principais características representadas neste modelo.

4.1 Ambiente de desenvolvimento

O ambiente AdaptWeb é baseado em um conjunto de ferramentas que suportam as fases de pré-autoria, adaptação de conteúdo e de navegação. A implementação utiliza primordialmente a linguagem de programação PHP (*Hypertext Preprocessor*), o

² Parcialmente financiado pelo CNPq, projetos: Plano Sul de Pós-graduação, Electra e Edital para Reequipamento de Laboratórios, AdaptWeb.

gerenciador de banco de dados MySQL e a linguagem XML para a organizar e disponibilizar as disciplinas.

O PHP é uma linguagem de elaboração de *scripts* embutida que opera do lado do servidor. Isso significa que ela funciona dentro de um documento HTML (*HyperText Markup Language*) para conferir-lhe a capacidade de gerar instruções específicas [CAS 2001]. Seu principal objetivo é permitir o desenvolvimento de páginas *Web* geradas dinamicamente.

Optou-se pelo uso da linguagem PHP pois além desta possuir simplicidade, uma forma quase natural de se usar banco de dados e ter independência de plataforma, ela possui código-fonte aberto, projetada para trabalhar na *Web*, possuindo um sistema de processamento muito bem otimizado para os tempos de respostas necessários nos aplicativos *Web* [CAS 2001].

O MySQL é um sistema de gerenciamento de banco de dados relacional de código-aberto. Foi originalmente desenvolvido para manipular bases de dados muito grandes com maior rapidez que as soluções existentes e tem sido utilizado com sucesso em ambientes de produção com alta demanda. Apesar de estar em constante desenvolvimento, o MySQL oferece um rico conjunto de funções, como conectividade, velocidade e segurança, fazendo com que ele seja altamente apropriado para acesso a banco de dados na Internet [DEM 2001].

Por ser um sistema de alta performance, o MySQL foi utilizado como a base de dados administrativa e contém informações do modelo do usuário, como histórico navegacional e seus conhecimentos adquiridos.

Nos documentos XML são armazenadas todas as informações sobre uma determinada disciplina. O XML possui uma estrutura hierárquica, além de manter o conteúdo instrucional separado do *layout* da apresentação.

O Servidor Web que está sendo utilizado é o Apache por ser um servidor que pode ser aplicado em diversos sistemas operacionais e possui código aberto.

4.2 Arquitetura

O AdaptWeb é composto por quatro módulos distintos denominados: (1) módulo de autoria [FRE 2003]; (2) módulo de armazenamento em XML [AMA 2002] (3) módulo de adaptação de conteúdo baseado no modelo do aluno [MAR 2003] e; (4) módulo de interface adaptativa descrito nesta dissertação. A arquitetura do AdaptWeb é apresentada na figura 4.1.

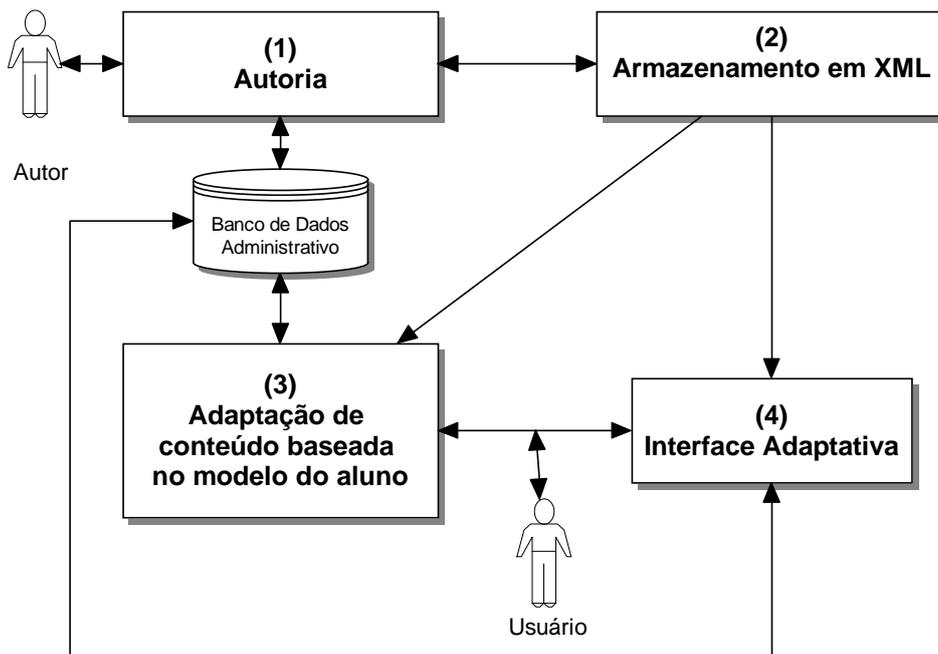


FIGURA 4.1 – Arquitetura do Ambiente AdaptWeb

4.2.1 Módulo 1: Autoria

O módulo de autoria consiste na organização do conteúdo instrucional a ser disponibilizado para o aluno. A estruturação de cada disciplina é baseada em conceitos. Para estruturação do conteúdo o autor tem como base uma sistemática para pré-autoria e uma ferramenta de autoria. A sistemática de pré-autoria auxilia na organização da estrutura geral da disciplina e na identificação dos arquivos relacionados a esta estrutura. A ferramenta de autoria possibilita criar a estrutura de conceitos através da sistemática de pré-autoria e classificar e associar os arquivos. Os arquivos associados a cada conceito podem ser classificados em quatro categorias: conceito propriamente dito, exemplos, exercícios e material complementar que por sua vez estão em uma única estrutura adaptada para os diferentes cursos (diferentes grupos podem assistir a uma mesma disciplina).

Sistemática para pré-autoria

Nesta fase o autor organiza e estrutura o conteúdo programático a ser disponibilizado para o aluno. Para essa organização do conteúdo, o autor necessita identificar os dados de entrada, e para cada conceito possui um arquivo HTML relacionado e outros arquivos associados, que são classificados como exemplos, exercícios e material complementar. Os arquivos associados podem estar em diferentes mídias, como imagens, vídeos, áudio, HTML.

Nessa sistemática o autor define os objetivos da disciplina; organiza os conceitos de forma hierárquica através de um editor de textos, independente do público-alvo; organiza o material instrucional e identifica os arquivos existentes para cada conceito, classificando-os de acordo com as categorias existentes (conceito, exemplo, exercício ou material complementar); edita os arquivos referentes a cada conceito para serem renomeados de acordo com critérios da sistemática de autoria, facilitando sua identificação dentro do ambiente e; posteriormente o autor relaciona os arquivos pertencentes a cada conceito a cada curso (grupo de usuários).

Ambiente de Autoria

Em primeiro lugar, para o autor ter acesso a ferramenta de autoria, ele deve solicitar acesso ao administrador do ambiente AdaptWeb. Após a liberação, ele tem acesso a ferramenta e pode criar a estrutura de conceito. Primeiramente o autor deve cadastrar os cursos, ou grupos de usuários que podem utilizar o sistema; depois deve cadastrar as disciplinas que serão apresentadas dentro do ambiente e finalmente o autor deve especificar para cada disciplina, quais são os cursos que podem acessá-la. Feito isso, o autor pode criar a estrutura de conceito, e associar a esta estrutura os arquivos relacionados.

Para cada conceito o autor deve informar um arquivo de conceito (obrigatoriamente em formato HTML), descrição do conceito, abreviação, lista de palavras-chave, lista de pré-requisitos e também selecionar para quais cursos deseja disponibilizar este conceito. Para as outras categorias, (exemplos, exercícios e material complementar), o autor pode informar uma lista de arquivos, podendo especificar para cada arquivo também o público alvo. Para exemplos e exercícios, o autor deve informar a descrição, o nível de complexidade (Sem Classificação, Fácil, Médio, Complexo), para quais cursos deseja disponibilizar o material e o arquivo referente a este. Para a categoria material complementar deve informar a descrição, para quais cursos deseja disponibilizar o material complementar e o arquivo referente ao material complementar.

Ao término do processo de autoria, os dados referentes a estrutura do conteúdo da disciplina estão armazenados em uma estrutura de dados matricial em memória, onde cada linha da matriz contém informações referentes a um conceito.

Estrutura de Dados resultante da fase de autoria

Ao finalizar a etapa de autoria todos os dados definidos pelo autor são armazenados em uma estrutura de dados em memória do tipo matriz. A seguir é feita uma descrição da estrutura de dados resultantes da fase de autoria, contendo as informações inseridas pelo autor:

São dados descritores do conceito: identificador do conceito (cada conceito é identificado por um número que mostra seu nível na estrutura, como por exemplo: 1, 1.1, 1.2, 1.3, 2. 2.1); nome do conceito, que armazena o nome do tópico de forma completa; abreviação do conceito que armazena o nome do conceito de forma resumida; palavra-chave que armazena uma relação de palavras-chave; pré-requisitos: armazena uma lista dos pré-requisitos referente ao conceito.

A estrutura do conceito possui: um arquivo principal, que é a matriz que armazena dados sobre o arquivo principal de conceito; a matriz curso, que armazena os identificadores dos cursos e seus respectivos *status*, ou seja, se o curso está ou não relacionado a este conceito; matriz exemplos, exercícios e materiais complementares que armazenam respectivamente os exemplos, os exercícios e os materiais complementares de cada conceito.

A matriz contendo dados descritores do conceito e a estrutura do conceito é a entrada de dados para a etapa de armazenamento em XML, servindo como base para a geração dos arquivos utilizados nas demais fases do ambiente AdaptWeb. Maiores informações sobre o módulo de autoria e sua respectiva estrutura de dados podem ser encontrados em [FRE 2003].

4.2.2 Módulo 2: Armazenamento em XML

O módulo de armazenamento é responsável pela organização dos arquivos fornecidos pelo autor, através do módulo de autoria, para XML. Também é nesta fase que os dados da autoria serão organizados para posterior utilização no ambiente.

Para criar os arquivos XML, este módulo utiliza como entrada a estrutura de dados passada pela fase de autoria, onde todos os dados definidos pelo autor estão armazenados. Essa representação serve como base para a geração dos arquivos utilizados nas demais etapas do ambiente, a fim de ser apresentado ao usuário adequadamente, observando seu perfil.

A saída de dados da presente fase é composta pelos arquivos XML. Existe um arquivo XML para a estrutura dos conceitos e um arquivo XML para organizar o conteúdo de cada conceito cadastrado pelo autor.

Para a criação dos documentos XML foram definidas duas DTD (*Document Data Definition*). Uma delas retrata a estrutura de conceitos definida pelo autor, já que na fase de autoria todos os conceitos de uma disciplina devem conter suas devidas informações como descrição, número, lista de pré-requisitos, lista de palavras-chaves relacionadas, entre outros elementos. A outra DTD tem por finalidade descrever os conteúdos armazenados para cada conceito. Esses conteúdos educacionais estão divididos em conceito, exemplos, exercícios e materiais complementares.

Algoritmo Gerador dos arquivos XML

Com base nas DTD foi definido e implementado um algoritmo para conversão da representação em memória, gerado na fase de autoria, em arquivos XML.

Este algoritmo cria um arquivo XML para cada disciplina com sua respectiva estrutura de conceitos, a qual define as características de cada conceito da disciplina. O conteúdo instrucional é armazenado em outro arquivo XML, dividido em conceitos, exemplos, exercícios e materiais complementares, como exposto na figura 4.2.

O algoritmo também contempla o tratamento dos dados recebidos da fase de autoria. Os arquivos XML só são gerados se os dados da autoria, armazenados em memória, estiverem todos inseridos. Caso exista um elemento da autoria que não esteja cadastrado, como uma descrição de conceito, por exemplo, o algoritmo sinaliza o erro e não conclui o processo de geração dos arquivos XML. Assim que o autor atualiza os elementos que estão faltando o algoritmo faz uma nova validação na estrutura gerada pela autoria e com os dados corretos gera os arquivos XML. Os elementos que devem estar inseridos na autoria são definidos nas DTD como elementos obrigatórios.

Na figura 4.2 é visualizado o esquema de geração de arquivos XML que possuem como entrada as DTD de estrutura de conceitos e a DTD de conteúdo do conceito. São gerados: um arquivo XML contendo a estrutura geral da disciplina, que é totalmente orientada ao conceito; e outros documentos XML referentes aos demais conteúdos relacionados ao conceito, como: exemplos, exercícios e material complementar.

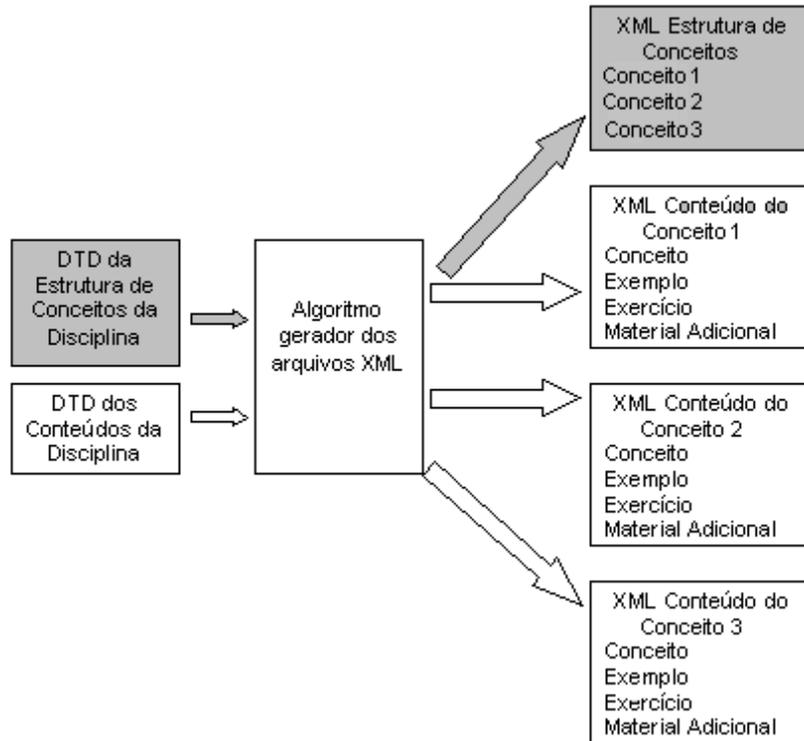


FIGURA 4.2 – Esquema de geração de arquivos XML

Esta organização permite uma estruturação dos dados de forma hierárquica, pois sempre existe um único arquivo XML com a estrutura de conceitos da disciplina e tantos arquivos XML com conteúdo quantos forem os conceitos definidos. Para exemplificar: se o autor insere dez conceitos na ferramenta de autoria então são criados onze documentos XML através do algoritmo, um para armazenar toda a estrutura de conceitos (como um índice), e um arquivo XML para cada um dos dez conceitos.

A figura 4.3 mostra a exemplificação de uma representação de uma estrutura, onde cada nó é um conceito. A utilização desta estrutura de nós leva a uma organização hierárquica de elementos [AMA 2002].

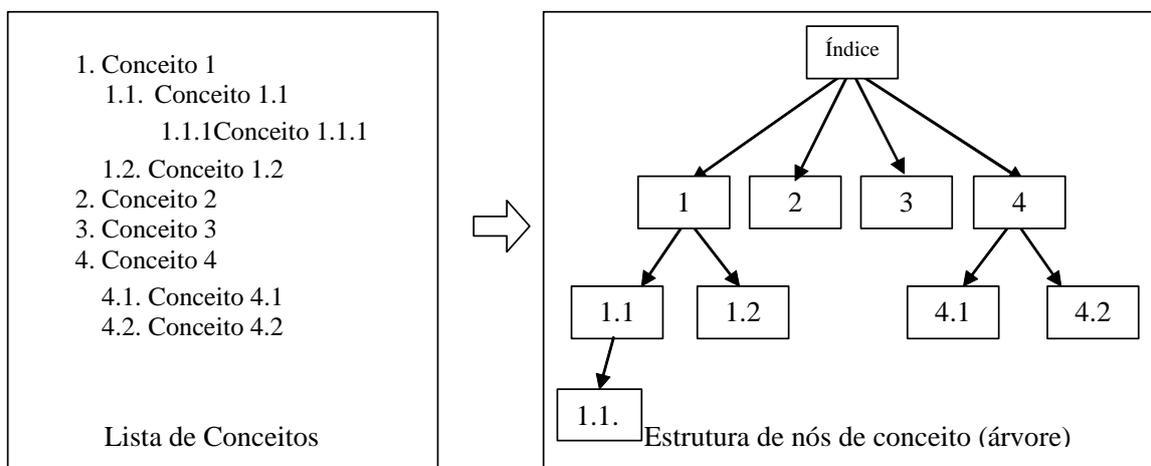


FIGURA 4.3 – Organização de um conteúdo institucional em lista de conceitos e em árvore de conceito

Todo o processo de geração de arquivos XML sempre é validado através de um analisador que percorre os documentos. Este *parser* utiliza a API (*Application Programming Interface*) DOM (*Document Object Model*) [DOM 98] para o processo de validação. Os documentos XML só são gravados no servidor se forem validados pelo *parser*.

A API DOM foi utilizada como padrão de interface entre os documentos XML e o algoritmo descrito, pois foi confirmada como uma recomendação da W3C em 1998 e é amplamente utilizada como uma interface independente de plataforma e linguagem, que provê formas de acesso para os programas e *scripts* acessarem e atualizarem dinamicamente os conteúdos, as estruturas e os estilos dos documentos XML [ROY 2001], [DOM 98].

O modelo DOM representa um documento XML como uma árvore, cujos nós são elementos, texto, etc. O processador XML gera a árvore na memória, e a entrega ao *parser* para compará-la às regras de criação de documento descritas nas DTD [SEL 2001].

Uma descrição mais detalhada do desenvolvimento do módulo de armazenamento de dados em XML pode ser vista em [AMA 2002].

4.2.3 Módulo 3: Adaptação de conteúdo baseado no modelo do aluno

Os módulos apresentados até o momento dizem respeito ao processo de autoria do material instrucional, ou seja, como os conteúdos existentes são organizados, estruturados, incluídos e finalmente armazenados. Os módulos de adaptação de conteúdo e de interface adaptativa, descritos nas seções que seguem são responsáveis por adaptar o ambiente para o usuário, tanto para recuperar e adaptar o material instrucional, quanto para adaptar sua navegação e apresentação do ambiente de acordo com o modelo do usuário.

Os conteúdos instrucionais disponibilizados pelo AdaptWeb precisam ser adaptados dinamicamente de acordo com o modelo do aluno. Este modelo representa o relacionamento entre o aluno e a estrutura de conceitos resultantes do processo de autoria, levando em consideração o curso, a preferência pelo modo de navegação e o ambiente de trabalho do aluno.

A primeira vez que o aluno acessa o ambiente AdaptWeb, deve cadastrar-se e solicitar matrícula nas disciplinas de seu interesse associadas ao seu curso. Desta forma, assim que o professor liberar sua matrícula, este estará apto para interagir com o ambiente.

Durante o início da interação o aluno passa pelo processo de *login* e a seguir seleciona a disciplina/curso que deseja e o modo de navegação de sua preferência, que pode ser tutorial ou livre. A partir deste momento, o módulo de identificação do ambiente deve verificar dinamicamente a característica tecnológica do ambiente de trabalho do aluno e repassar esta informação ao módulo de Adaptação de conteúdo.

Desta maneira, antes do acesso ao conteúdo educacional, são coletadas informações a respeito da identificação do aluno, da disciplina e do curso que deseja interagir, do modo de navegação escolhido e da característica tecnológica de seu ambiente de trabalho naquele instante de tempo. Estas informações são utilizadas no processo de filtragem dos documentos XML a fim de possibilitar a adaptabilidade na apresentação do conteúdo instrucional, das mídias e da interface.

Filtros de documentos XML conteúdo do conceito

Cada conceito da estrutura hierárquica global possui um arquivo XML de conteúdo associado. Nestes arquivos estão armazenados o conceito teórico, os exemplos, os exercícios e os materiais complementares associados a este conceito e suas respectivas características, como quais cursos podem resolver um determinado exercício ou se o exercício é fácil, médio, complexo, ou não tem classificação.

Estes arquivos passam pelo filtro durante a navegação do aluno pelos conceitos, sendo que estes possuem quatro formas de apresentação: conceito (teoria), exemplo, exercício e material complementar. De acordo com a forma e o conceito escolhido, o filtro é executado sobre o respectivo arquivo XML de conteúdo e é gerada dinamicamente a apresentação para o aluno.

Existem quatro filtros, um para cada forma de apresentação do conteúdo: conceito, exemplo, exercício e material complementar. O filtro de conceito filtra as mídias deste determinado conceito observando o ambiente de trabalho do usuário no momento; o filtro de exemplo filtra de acordo com o curso, pois podem existir vários exemplos, além de observar as características da conexão de rede do ambiente de trabalho do aluno; para filtrar os exercícios, observam-se os mesmos aspectos utilizados pelo filtro de exemplos, e utiliza-se também a classificação deste exercício, podendo este ser “fácil”, “médio”, “complexo” ou ainda não possuir classificação. O aluno pode observar somente os exercícios que correspondem com seu modelo no momento. O filtro para material complementar também filtra de acordo com o curso e o ambiente de trabalho do usuário. Os filtros apresentados também são considerados como critérios para a geração adaptada do conteúdo das páginas HTML. Maiores informações sobre o módulo de adaptação de conteúdo encontram-se em [MAR 2003].

4.2.4 Módulo 4: Interface Adaptativa

O módulo de interface adaptativa e o módulo de adaptação do conteúdo são responsáveis pela apresentação adaptativa baseando-se no modelo do usuário. O módulo de adaptação do conteúdo cuida do conteúdo das páginas, ou seja, adapta o conteúdo do autor observando o curso e o ambiente de trabalho do usuário. Ele faz os filtros baseados nos XML de conteúdo de conceitos. Já o módulo de interface adaptativa adapta a navegação e a apresentação da interface do ambiente, mostrando versões diferentes de acordo com o curso, as preferências pelo modo de navegação e o conhecimento do usuário correspondentes ao modelo do usuário. Os filtros deste módulo são baseados no XML de estrutura de conceitos. Este XML possui todas as informações sobre uma disciplina, mostrando todos os conceitos relacionados com cada curso. No capítulo 5 o módulo de Interface Adaptativa será apresentado com mais detalhes, tema deste trabalho.

5 Interface Adaptativa

A seguir serão apresentadas as tecnologias utilizadas para o desenvolvimento do módulo de Interface Adaptativa (seção 5.1), as características dos usuários utilizadas no ambiente AdaptWeb para prover adaptabilidade (seção 5.2), a modelagem do ambiente (seção 5.3), as técnicas adaptativas utilizadas (seção 5.4) e a implementação do módulo em conjunto com o monitoramento (seção 5.5, 5.6 e 5.7). A arquitetura do ambiente AdaptWeb com o módulo de interface adaptativa em destaque pode ser visualizado na figura 5.1.

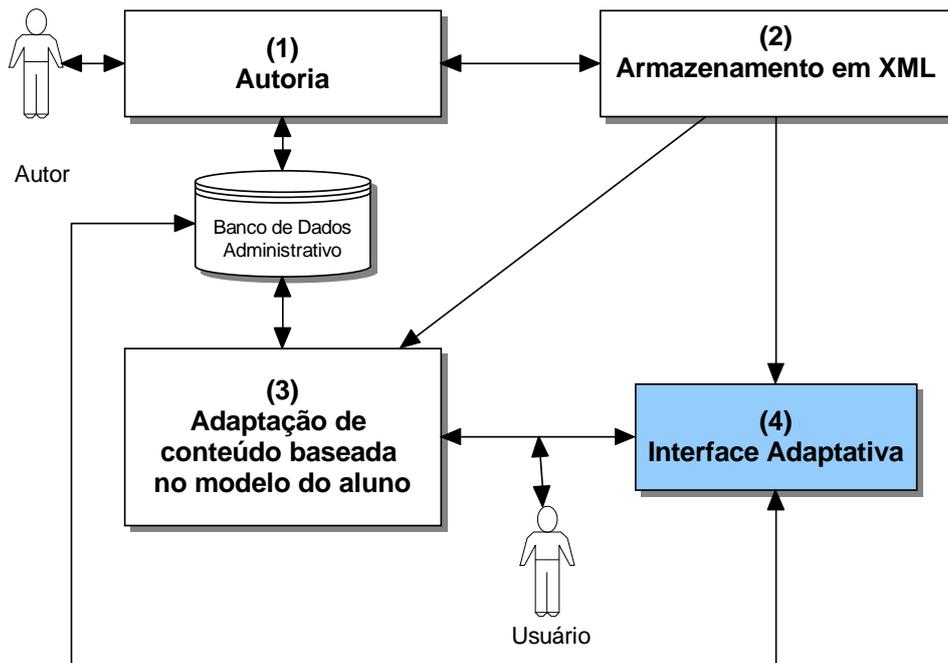


FIGURA 5.1 – Arquitetura do ambiente AdaptWeb com o módulo de Interface Adaptativa em destaque

O ambiente AdaptWeb é um SHA que permite a adaptabilidade de navegação, de apresentação e de conteúdo. A adaptabilidade de navegação é uma característica particularmente importante em ambientes educacionais, por permitir a seleção e o acesso a conteúdos diferentes e em ordens distintas. No AdaptWeb, a interface do ambiente de ensino foi estruturada visando auxiliar o aluno na navegação, proporcionando simplicidade operacional e visão geral das opções disponíveis.

Pode-se dizer que o modelo do usuário no ambiente AdaptWeb é um modelo cooperativo, pois sua construção e manutenção é feita por vários módulos que interagem em conjunto. A criação do modelo do usuário é realizada quando o usuário se cadastra em alguma disciplina (por exemplo, no estudo de caso adotado, a disciplina de Computação Algébrica e Numérica) e escolhe um dos cursos cadastrados para esta disciplina em questão, sendo estes cursos cadastrados pelo professor/autor da disciplina na fase de autoria (por exemplo, no estudo de caso, o autor cadastrou os cursos de Matemática, Ciência da Computação e Engenharia). Neste instante o módulo 3: adaptação de conteúdo cadastra o novo usuário no banco de dados MySQL, respeitando o curso selecionado. Quando o aluno começa a assistir a aula, o módulo 4: interface adaptativa também coleta dados do usuário, como seu conhecimento, sua preferência

navegacional e seu histórico navegacional. Esse procedimento será explicado nas próximas seções.

O módulo de interface adaptativa, tema deste trabalho, se encarrega da adaptação navegacional no ambiente, visto que, para ambientes educacionais, o aluno, ao ser auxiliado pelo sistema na escolha do melhor caminho navegacional, acelera seu processo de aprendizagem. O módulo de adaptação de conteúdo baseado no modelo do usuário é o responsável pela adaptação de conteúdo. Os dois módulos em conjunto adaptam a apresentação, sendo que o primeiro adapta o *layout* do ambiente e dos *links* e o segundo o *layout* do conteúdo instrucional. O módulo de interface adaptativa é o responsável direto pela adaptação navegacional do AdaptWeb, observando as características do usuário contidas em seu modelo. Um exemplo do uso dessas características pode ser visto quando se observa a preferência pelo modo de navegação do usuário. Se o aluno optar pela navegação tutorial, o módulo de Interface adaptativa observa o modelo do aluno no momento para buscar o conhecimento já adquirido por este, para adaptar assim a navegação, liberando conceitos de acordo com os pré-requisitos. Uma explicação mais detalhada sobre as características do usuário utilizadas em seu modelo pode ser vista na seção 5.2.

5.1 Tecnologias utilizadas para o desenvolvimento do módulo de Interface Adaptativa

O módulo de Interface Adaptativa foi implementado essencialmente com a linguagem de programação PHP, versão 4.2.X. Utilizou-se o banco de dados MySQL para monitoramento e atualização dinâmico de *log* do usuário. Para realizar a apresentação adaptativa da navegação, o módulo faz o monitoramento do aluno. Cada vez que o aluno adquire um novo conhecimento, o módulo ativa uma função que atualiza a tabela de *log* cadastrando o novo conceito visitado/adquirido.

O MySQL é conhecido por manipular grandes bases de dados. Existem bases de usuários em MySQL com 60.000 tabelas e cerca de 5.000.000.000 de registros. As funções SQL (*Structured Query Language*) são implementadas através de uma biblioteca de classes altamente otimizada e são muito rápidas atingindo uma alta performance e dinamismo [DEM 2001]. Como o ambiente é uma aplicação baseada na *Web*, a alta performance e a dinamicidade se tornam essenciais, e o MySQL foi escolhido para resolver essas questões.

Na etapa de autoria, os conteúdos educacionais do AdaptWeb são organizados pelo professor através de uma árvore de conceitos (tópico 1, 1.1, 1.2, 2, 2.1, 2.1.1, etc.) específica para cada área de conhecimento. Neste contexto, a linguagem XML desempenha um importante papel, visto que um documento XML também possui uma estrutura hierárquica em árvore e seu conteúdo é mantido separado do *layout* de apresentação. Desta maneira, através da API DOM, torna-se possível acessar e manipular um documento XML, gerando diferentes apresentações para o mesmo [DOM 98].

No ambiente AdaptWeb foi utilizado a API DOM como padrão de interface entre os documentos XML e o programa de aplicação desenvolvido em PHP. A API DOM foi confirmada como uma recomendação da W3C em 1998 e é amplamente utilizada como uma interface neutra de plataforma e linguagem, que provê formas de acesso para os programas e *scripts* acessarem e atualizarem dinamicamente os conteúdos, as estruturas e os estilos dos documentos XML [ROY 2001], [W3C 2000].

O modelo DOM representa um documento XML como uma árvore hierárquica na memória, cujos nós são elementos, texto, páginas HTML, etc. O processador XML gera a árvore na memória, e a entrega ao programa de aplicação, que pode ser escrito em uma linguagem *client-scripting* (linguagem cujo código é executado no navegador do visitante), como *JavaScript* ou *VBScript*, ou *server-scripting* (linguagem cujo código é executado no servidor *Web*), como o PHP [SEL 2001].

O módulo de interface adaptativa utiliza a API DOM para gerar a navegação entre os conceitos para o usuário, como uma árvore hierárquica, utilizando para isso o XML de estrutura de conceitos proveniente do módulo de armazenamento XML. Utiliza-se o PHP como a linguagem de aplicação que faz as chamadas.

Também é utilizada a linguagem *JavaScript* para a manipulação do menu de navegação (maximizar, minimizar, restaurar) aplicada junto ao HTML.

5.2 Aspectos que podem ser adaptados no Ambiente AdaptWeb

Todas as técnicas que promovem adaptabilidade são baseadas nas características do usuário, representadas no modelo do usuário [DEB 99a]. Brusilovsky identificou alguns aspectos importantes do usuário, que podem ser utilizados pelo sistema para prover adaptação [BRU 96], [BRU 2001]: seus conhecimentos, objetivos, experiências, preferências, a área de conhecimento e o ambiente de trabalho. No AdaptWeb, este modelo representa o relacionamento entre o aluno e a estrutura de conceitos do processo de autoria, levando em consideração a sua formação (curso), a preferência pelo modo de navegação (tutorial ou livre), o ambiente de trabalho do aluno e o conhecimento adquirido. O conhecimento, a formação e a preferência pelo modo de navegação são aspectos monitorados pelo módulo de interface adaptativa, e utilizados durante a adaptação da navegação. A formação e o tipo de conexão de rede (ambiente de trabalho) são informações coletadas pelo módulo de adaptação de conteúdo através de formulários, e utilizadas para a adaptação de conteúdos.

5.2.1 Formação do Usuário

A formação está relacionada à área de conhecimento do aluno. Um aluno do curso de física, que queira fazer a disciplina de Lógica de Programação, não possui a base de conhecimento para estudar o mesmo conteúdo educacional de um aluno formado em computação. Por isso, o AdaptWeb provê este nível de adaptação, que somente acontece porque o próprio autor através de sua experiência determina quais conteúdos de uma mesma disciplina serão vistos para cada grupo de alunos (denominados cursos), adaptando assim com base na formação de cada aluno.

Quando o usuário faz o *login* para acessar o ambiente, ele deve informar qual a disciplina que deseja aprender e também selecionar qual sua formação (curso).

Essa característica é utilizada para adaptar tanto o módulo de adaptação de conteúdo quanto o módulo de interface adaptativa. O módulo de adaptação de conteúdo preocupa-se em mostrar os conceitos, exemplos, exercícios e materiais complementares para cada curso, de acordo com a especificação do autor. Para os exercícios, ainda pode existir uma característica individual, que é a organização estrutural na apresentação, sendo que um exercício pode ser “fácil”, “médio”, “complexo”, ou não possuir classificação. Para isso o módulo de adaptação de conteúdo utiliza filtros no XML de conceito.

O módulo de interface adaptativa utiliza a formação do usuário para adaptar a navegação do usuário. Um grupo de usuários pode assistir a um determinado conjunto de conceitos, exemplos, exercícios e materiais complementares, enquanto outros não. Mesmo que um conceito esteja habilitado para um determinado curso, pode-se ou não ter exemplos, exercícios e materiais complementares para este mesmo conceito.

Para isso o módulo de interface adaptativa verifica quais os conceitos relacionados a um determinado grupo de usuários (curso) no XML de estrutura de conceitos.

Ele é o responsável em apresentar o menu de conceitos, de exemplos, de exercícios e de materiais complementares de acordo com essa característica. Vale lembrar que a estrutura hierárquica do material instrucional é orientada a conceito, que significa dizer que só podem existir exemplos, exercícios e materiais complementares de um determinado conceito (se o tópico conceito existir). Logo é obrigatório existir o conceito para existir qualquer outra categoria (exemplo, exercício, material complementar) de um conceito, mas a recíproca não é verdadeira. Então, o módulo de interface adaptativa deve primeiramente observar se existe um determinado conceito para um grupo de usuários, e se existir, aí verificar se existem exemplos, exercícios e materiais complementares.

5.2.2 Ambiente de Trabalho

A adaptação ao ambiente de trabalho do aluno surgiu com os sistemas baseados na *Web*. Os alunos que são usuários das aplicações *Web* do tipo *server-side* podem estudar virtualmente de qualquer lugar e através de diferentes equipamentos. Este fato é importante na escolha da mídia mais adequada para apresentar o conteúdo ao aluno. A adaptação de acordo com o ambiente de trabalho é feita pelo módulo de adaptação de conteúdo, onde este adapta um conteúdo de acordo com o modelo do usuário.

Por enquanto, o módulo de identificação do ambiente de trabalho ainda não está em funcionamento, então o ambiente ainda não tem condições para saber dinamicamente qual o perfil tecnológico do usuário. Para que exista a adaptação, quando o usuário faz o *login* para entrar no ambiente, ele seleciona a disciplina e seu curso, e posteriormente ele seleciona qual seu ambiente de trabalho (conexão *dial-up* ou *ADSL - Asymmetrical Digital Subscriber Line*). Desta forma o módulo de adaptação de conteúdo faz a adaptação dos conteúdos das páginas, das mídias e da apresentação como um todo, filtrando os HTML da autoria.

5.2.3 Conhecimento

O conhecimento atual e específico do aluno é a característica mais importante e variável para cada aluno em particular. Isto significa que o sistema deve observar as mudanças no conhecimento do aluno e atualizar o modelo adequadamente.

Na fase de autoria o autor pode optar por definir conceitos pré-requisitos a outros. Isso significa que o aluno deve ter conhecimento sobre os pré-requisitos de um determinado conceito para poder acessá-lo. Essa característica é armazenada pelo módulo de interface adaptativa (monitoramento) no banco de dados e utilizada para prover a adaptabilidade.

Na estrutura hierárquica de conceitos com pré-requisitos, as informações sobre os conceitos que já foram percorridos por um aluno são representadas por um valor binário: sim (visitado/conceito adquirido – 100%) ou não (não visitado/ não adquirido 0%). Para isso, o módulo de interface adaptativa armazena o *log* de acesso da navegação

do aluno. Se o conceito pré-requisito já foi visitado (ou “aprendido”) pelo usuário, então o conceito que depende deste pode ser liberado para ser visualizado e assistido. Um conceito pode assumir três categorias, a de “aprendido”, neste caso isso significa que o usuário já visitou o *link*; “em-trabalho”, que significa dizer que o conceito está sendo visitado, este conceito é chamado de atual; e conceito “não-conhecido”, que pode estar nesta categoria por duas razões, (1) o conceito ainda não foi liberado pois seus pré-requisitos ainda não foram atingidos; ou (2) o conceito ainda não foi visitado (“aprendido”), mas já está liberado pois seus pré-requisitos já foram visitados (“aprendidos”).

5.2.4 Preferência pelo modo de navegação

A adaptabilidade na navegação é um recurso importante especialmente para ambientes educacionais, visto que o aluno ao ser auxiliado pelo sistema na escolha do melhor caminho navegacional não se perde no hiperespaço, além de diminuir sua sobrecarga cognitiva, pois é o sistema que verifica o melhor caminho, e não o usuário, possibilitando assim um aceleração em seu processo de aprendizagem [SAN 98], [SAN 96].

O AdaptWeb possui duas formas de navegação no qual o usuário tem a opção de escolha: o modo tutorial e o modo livre.

O modo tutorial é o modo em que são considerados os pré-requisitos definidos pelo autor na fase de autoria. Neste modo o usuário só pode acessar um conceito se seus pré-requisitos já foram visitados/conhecidos, ou seja ele é direcionado pelo professor. Este modo de navegação auxilia a navegação de usuários dentro do hiperespaço. Por essa razão, ele é indicado para usuários com pouca experiência no ambiente, reduzindo problemas de orientação e de caminho, ajudando o usuário a entender seu posicionamento no hiperespaço.

O modo livre, como o próprio nome já diz, não considera os pré-requisitos definidos pelo autor, podendo o usuário navegar por todo o hiperespaço da disciplina, sem restrições.

Independente do modo de navegação escolhido pelo usuário, o módulo de interface adaptativa monitora o aluno na navegação sobre os conceitos, exemplos, exercícios e materiais complementares, fazendo atualizações constantes de *log*. Assim, cada vez que o aluno adquire um novo conhecimento, o módulo ativa uma função que atualiza a tabela de *log* do aluno, cadastrando o novo conceito visitado. Este procedimento possibilita que o professor realize análises sobre o *log* do aluno e descubra várias características sobre o aluno, como por exemplo, o seu percurso preferido.

São usadas cores diferentes para indicar quais conceitos foram visitados, quais não foram e qual conceito está sendo visitado no momento (atual). No modo tutorial os *links* desabilitados demonstram que o aluno ainda não tem o conhecimento necessário para visitá-los (pré-requisitos) e no modo livre todos os *links* do menu de navegação ficam disponíveis. As cores utilizadas para a indicação seguem as regras de usabilidade de Nielsen [NIE 2000], ou seja, foi adotado para *links* que o usuário ainda não viu, a cor azul, e para *links* que o usuário já visitou o roxo.

5.3 Modelagem do Ambiente

Essa seção é dedicada a mostrar a modelagem do ambiente. A seção 5.3.1 mostra as tabelas do banco de dados. A seção 5.3.2 mostra a modelagem do modelo do usuário e do domínio.

5.3.1 Modelagem do Banco de Dados Administrativo

Essa seção mostra a estruturação do banco de dados administrativo, onde este armazena dados da disciplina, do curso e dos usuários. A figura 5.2 mostra o relacionamento entre as tabelas do ambiente.

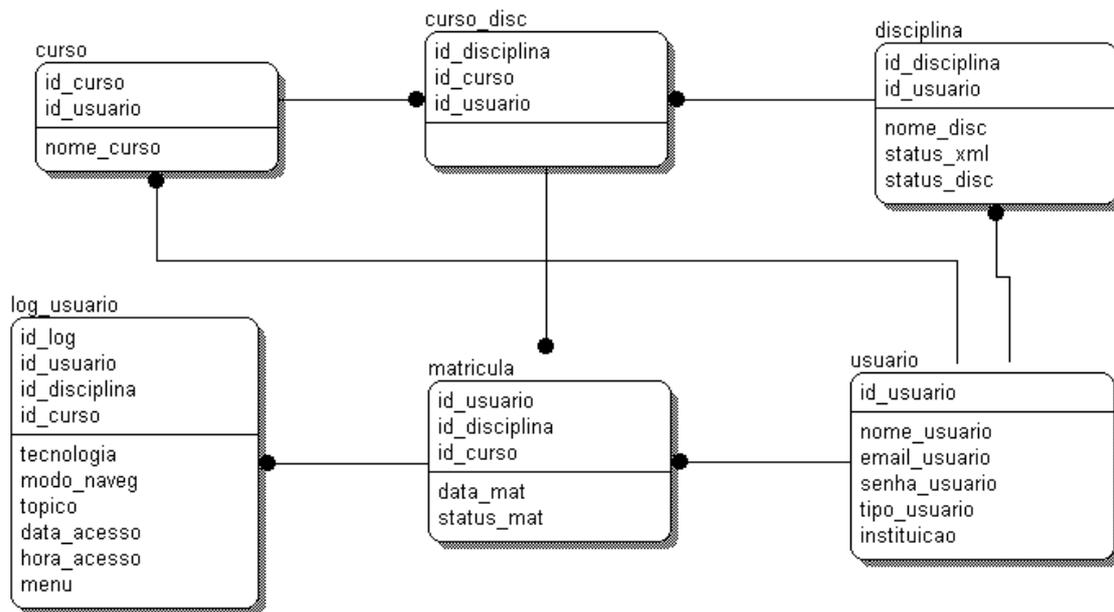


FIGURA 5.2 – Relacionamento entre as tabelas do banco de dados administrativo

A tabela criada e utilizada neste trabalho é a de *log_usuario*, para monitoração e adaptação do aluno de acordo com o conhecimento adquirido. A seção 5.6 explica o monitoramento com mais detalhes. As outras tabelas foram utilizadas pelos módulos de autoria e de adaptação de conteúdo.

5.3.2 Modelagem do Usuário e do Domínio

O modelo do aluno no ambiente AdaptWeb é um modelo colaborativo, no qual o módulo de Interface Adaptativa e o módulo de Adaptação de conteúdo baseado no modelo do aluno interagem. O módulo de adaptação de conteúdo cadastra o usuário, com seu curso e disciplina, já o módulo de Interface Adaptativa é o encarregado de atualizar o modelo de acordo com a aquisição de conhecimento do usuário.

Quando o aluno entra pela primeira vez no ambiente, ele possuirá uma identificação. Ele deve escolher a disciplina para estudo (por exemplo, no estudo de caso, a disciplina de Computação Algébrica e Numérica) juntamente com seu curso (por exemplo, no estudo de caso adotado, os cursos da Matemática, Ciência da Computação e Engenharia podem assistir as aulas da disciplina). Quando o aluno entra pela primeira vez no ambiente de ensino da disciplina selecionada, o sistema assume que seu conhecimento sobre a mesma é nulo.

A cada interação com o ambiente, ou seja, a cada novo conceito visitado, o módulo de Interface Adaptativa armazena um *log* de usuário, com a identificação do usuário, de seu curso, da disciplina, da tecnologia, modo de navegação (tutorial ou livre), data e hora de acesso, se ele está em um conceito, ou em um exemplo, exercício ou material complementar, e o mais importante, qual é o conceito que está sendo visitado (por exemplo, conceito 1.2.3). Todos esses dados ficam armazenados na tabela de *log_usuario*, e toda vez que o usuário entra novamente no ambiente, o módulo de Interface Adaptativa procura os conceitos já visitados pelo estudante nesta tabela para mostrá-los de forma diferenciada.

O ambiente AdaptWeb utiliza o modelo de *overlay* e o modelo de estereótipo para representar o aluno. O modelo de *overlay* é utilizado pelo módulo de Interface Adaptativa para representar o conhecimento do aluno. O módulo possui um atributo para cada conceito do domínio da aplicação. Esse atributo pode ser: (i) visitado, que quer dizer que o aluno já possui conhecimento sobre este conceito; (ii) atual, que significa que o estudante está adquirindo o conhecimento no momento; e (iii) não-visitado, que implica que o aluno ainda não possui conhecimento sobre o conceito.

Além disso, um aluno pertence a algum curso, e para cada curso, o professor determina quais conceitos vão existir. Desta forma, o modelo de estereótipo é utilizado pelos módulos de Interface Adaptativa e de Adaptação do Conteúdo. O módulo de Interface Adaptativa só mostra os conceitos para o curso selecionado, enquanto o módulo de adaptação de conteúdo verifica para cada conceito, quais exemplos, exercícios e material complementar podem existir. Esses exemplos e exercícios são dispostos como fácil, intermediário, difícil ou sem classificação (não possuem classificação).

O modelo do domínio é armazenado em documentos XML, pois o AdaptWeb possui uma estrutura hierárquica de conceitos, e os documentos XML são estruturados hierarquicamente. O módulo de Armazenamento em XML é encarregado de gerar estes arquivos, para posteriormente o módulo de Interface Adaptativa e o módulo de Adaptação de Conteúdo filtrarem os mesmos para adaptar o domínio de acordo com o perfil do usuário.

5.4 Técnicas Adaptativas utilizadas para prover adaptação baseada nas características do Usuário

A seguir serão apresentadas as técnicas adaptativas utilizadas para cada característica do usuário. Todos os módulos atuam no modelo do usuário, mas os módulos que realmente implementam a adaptação no ambiente são os módulos de adaptação de conteúdo baseado no modelo do usuário e o módulo de interface adaptativa. O primeiro adapta o conteúdo das páginas do autor, e o segundo adapta a navegação e a apresentação da interface do ambiente, sendo que a união desses módulos proporcionam uma apresentação adaptativa.

Formação do Usuário

Para adaptar o conteúdo das páginas de acordo com a formação do usuário, o módulo de adaptação de conteúdo utiliza a técnica de **fragmento variante**, que produz diversas variações de uma mesma página. Essa adaptação é feita para os exemplos, exercícios e materiais complementares. Por exemplo, existem vários exercícios sobre um conceito, mas somente um deles o autor selecionou para um determinado curso, ou seja, este curso só pode fazer este exercício. Para maiores explicações, ver [MAR 2003].

Para adaptar a navegação pela formação do usuário, o módulo de interface adaptativa aplica a técnica de **remoção de links**, removendo os *links* de todos os conceitos que o autor não selecionou para um determinado curso. Para remoção dos *links* deve-se verificar no modelo do usuário (mais especificamente, na tabela de usuário, de curso e de disciplina) qual é o curso do usuário para a disciplina ministrada. A seção 5.7 explica com isso é feito no ambiente. A figura 5.3 mostra uma remoção de alguns conceitos verificando o curso do usuário. Os conceitos em negrito são conceitos que o autor não selecionou para o curso A.

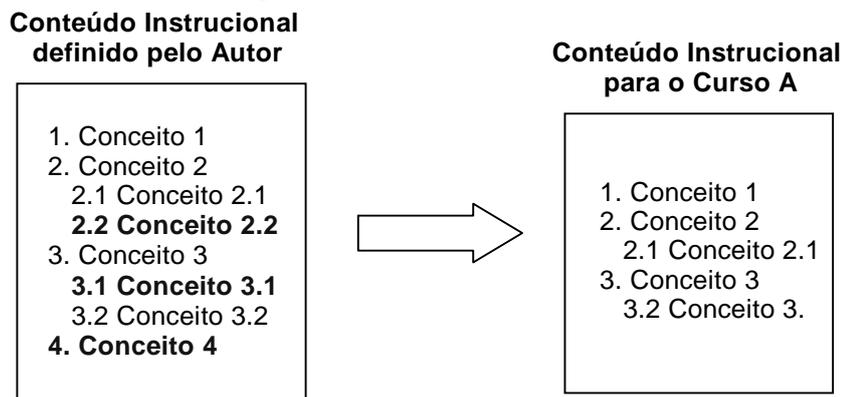


FIGURA 5.3 – Exemplificação da técnica de remoção de *links*

Utilizou-se a técnica de remoção de *links* para os conceitos pois o usuário não precisa saber qual material instrucional foi selecionado para ele, e qual não foi. Por isso, pode-se considerar que a remoção já vem desde a fase de autoria, pois o autor simplesmente faz uma marcação nos conceitos, selecionando-os para os cursos determinados. A técnica de **orientação direta** também foi utilizada para adaptar a apresentação dos tópicos (conceito, exemplo, exercício e material complementar) pela formação do usuário e é apresentada em dois elementos da interface. O primeiro é o “caminho hierárquico”, visível e clicável, apresentando ao usuário sua localização na hierarquia de acordo com a estrutura elaborada pela autoria e também se existem os tópicos da determinada categoria (conceito, exemplo, exercício e material complementar). Já o segundo elemento somente é apresentado no modo de navegação livre. Neste modo de navegação, o *layout* possui “setas” que apresentam os possíveis caminhos a serem seguidos pelo usuário em relação a sua hierarquia. Deste modo, as setas apresentam até quatro possíveis caminhos: (i) para o tópico pai; (ii) para o próximo irmão; (iii) para o irmão anterior e (iv) para o filho imediato. Caso não existam esses tópicos associados, a seta não aparece, utilizando-se a remoção de *links*.

Uma outra técnica explicada a seguir é utilizada para as categorias exemplo, exercício, material complementar. A estrutura da disciplina é baseada em conceitos, e por essa razão, se um conceito (por exemplo 1.2.2) não existir para um grupo de usuários, então também não existem exemplos, exercícios e material complementar deste conceito para aquele grupo. Mas se o conceito existir, os exemplos, exercícios e os materiais complementares podem ou não existir. Se não existir, é aplicada a técnica de **anotação de links** em conjunto com a técnica de **desabilitação de links**, para demonstrar para o usuário o mesmo menu hierárquico, com os mesmos tópicos, mas deixando-o a par de que não existem exemplos (ou exercícios, ou materiais complementares) para aquele tópico conceito. A figura 5.4 mostra um exemplo de uma disciplina, com seus conceitos. Os conceitos em negrito não possuem exemplos, então o menu de exemplos aparecerá com a mesma hierarquia do menu de conceitos, porém,

com destaque os tópicos exemplos que não possuem exemplos, mostrando para o usuário que naqueles tópicos não existem exemplos.

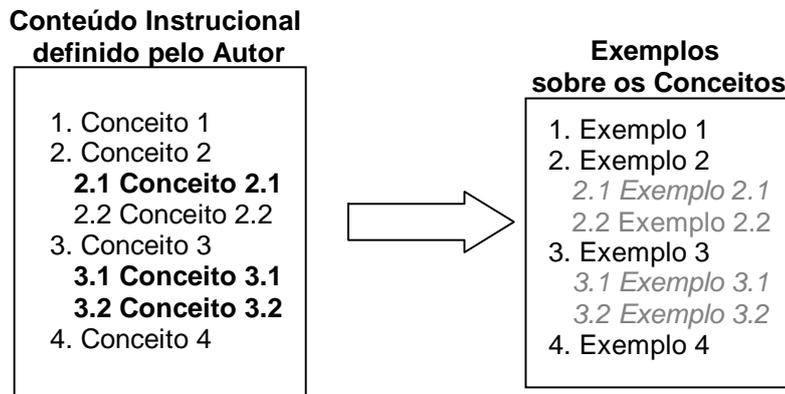


FIGURA 5.4 – Exemplificação da técnica de anotação unida com a desabilitação de *links*

Optou-se pela utilização das técnicas de anotação juntamente com a de desabilitação de *links* por dois aspectos principais, a consistência da interface e a associação entre a estrutura do domínio e o menu de tópicos que o usuário pode fazer sempre que utilizar o ambiente.

Como o menu de tópicos sempre contém todos os conceitos que o usuário pode acessar, quando o aluno seleciona a guia de exemplos, ou a guia de exercícios, ou ainda a de material complementar, e menu vai conter sempre os mesmos conceitos. Ou seja, a estrutura conceitual do domínio é preservada, contribuindo para o aluno assimilar a estrutura. Para o aluno diferenciar um conceito exemplo que possui material de um que não possui, é feita a anotação dos *links*, que mostra os conceitos que possuem exemplos em azul (*link* ativo ainda não visitado), ou em roxo (*link* ativo com exemplo visitado), e os conceitos que não possuem exemplo em cinza claro.

Ambiente de trabalho

A adaptação pelo ambiente de trabalho é feita somente pelo módulo de adaptação de conteúdo, que utiliza a técnica de **texto condicional** para apresentar somente as mídias que o usuário tem condição tecnológica no momento. Para maiores explicações, ver [MAR 2003].

Conhecimento

Para a adaptação navegacional observando o conhecimento adquirido pelo usuário o módulo de interface adaptativa aplica as técnicas de **anotação de links** em conjunto com a **desabilitação de links**, para diferenciar *links* visitados (conhecimento adquirido), não visitados (ainda não possui o conhecimento) e o atual (em aprendizagem). Essa característica é observada juntamente com o modo de navegação escolhido pelo usuário.

Quando o modo de navegação escolhido for o livre, o módulo de Interface Adaptativa deve somente verificar quais os conceitos que já foram visitados, ou seja, quais os conceitos que já foram aprendidos. Para esses conceitos, é utilizada a técnica de anotação de *links* mostrando os *links* na cor roxa. Optou-se por essa técnica e por essa cor, com base nas recomendações de Nielsen [NIE 2000], [NIE 2002], onde ele diz que

os *links* visitados de uma página devem ser mostrados na cor roxa (ou púrpura) ou vermelho. Para os conceitos ainda não visitados, optou-se pela anotação azul, que é a cor oficial de *links* na Web. E o conceito que está sendo visualizado no momento, ou seja, o usuário está adquirindo o conhecimento, está na cor vermelha, com a finalidade de destaque.

A diferença do modo tutorial, é que neste modo de navegação, o módulo de Interface Adaptativa, além de utilizar a técnica descrita no modo livre (anotação), também precisa verificar os pré-requisitos definidos pelo autor. Um conceito só pode ser assistido se seus pré-requisitos já foram conhecidos. Desta forma, o módulo de Interface Adaptativa também utiliza a técnica de desabilitação de *links*, pois o usuário deve saber que existe o conceito, mas que este conceito ainda não está liberado para ele por que ele ainda não adquiriu o conhecimento necessário. Também é feita a anotação dos *links* desabilitados pela falta de conhecimento, estes *links* ficam na cor preta, mostrando assim que quando o usuário adquirir o conhecimento necessário, o *link* ficará ativo.

Preferência pelo modo de navegação

Para adaptar a navegação e a apresentação da interface de acordo com a preferência navegacional do usuário, o módulo de interface adaptativa implementa, além das técnicas de adaptação dos *links* descritas sobre o conhecimento, a técnica de **página variante**, sobre o método *layout* variante.

Quando o usuário escolhe o modo de navegação tutorial, ele está ciente de que deve seguir os pré-requisitos definidos pelo autor. Quando o usuário opta pelo modo de navegação livre, além dos recursos navegacionais oferecidos pelo ambiente no modo tutorial, este modo dispõe de recursos adicionais: a busca por palavras-chaves e a navegação direta. Porém, o menu navegacional deste modo não obedece aos pré-requisitos definidos pelo autor, gerando assim um menu onde todos os conceitos são habilitados, sem o usuário precisar ter adquirido o conhecimento prévio. Optou-se por essas duas variações no *layout* para os usuários selecionarem uma interface simplificada, mas que contém todos os recursos para um usuário inexperiente (modo tutorial), e uma outra que contém recursos avançados, para usuários que gostam de navegar livremente pelo hiperespaço (modo livre).

A seguir é apresentada a tabela 5.1 onde são sintetizadas as técnicas utilizadas para prover adaptação no ambiente. As linhas são referentes as técnicas, enquanto nas colunas são apresentadas as características do modelo do usuário. O cruzamento entre elas é descrito em qual módulo do ambiente é realizada a adaptação, que pode ser no módulo de adaptação de conteúdo ou no módulo de interface adaptativa.

TABELA 5.1 –Técnicas adaptativas utilizadas no Ambiente AdaptWeb

Características/ Técnicas Adaptativas	Formação do Usuário	Ambiente de Trabalho	Conhecimento	Preferência Navegacional
Fragmento variante	Adaptação de Conteúdo			
Remoção de <i>links</i>	Interface Adaptativa			
Anotação de <i>links</i>	Interface Adaptativa		Interface Adaptativa	
Desabilitação de <i>links</i>	Interface Adaptativa		Interface Adaptativa	
Texto Condicional		Adaptação de Conteúdo		
Página Variante				Interface Adaptativa
Orientação Direta	Interface Adaptativa			

5.5 Interface do ambiente de ensino

A Interface do ambiente de ensino foi estruturada para obter simplicidade e consistência. A Interface foi planejada para exibir todas as categorias sobre um conteúdo, previstas na fase de autoria: o conceito (propriamente dito), os exemplos, os exercícios e os materiais complementares para cada conceito.

A seguir será apresentado um detalhamento da área física da interface quando o modo de navegação livre foi escolhido (figura 5.5), dividida pelo seu funcionamento e sua categoria.

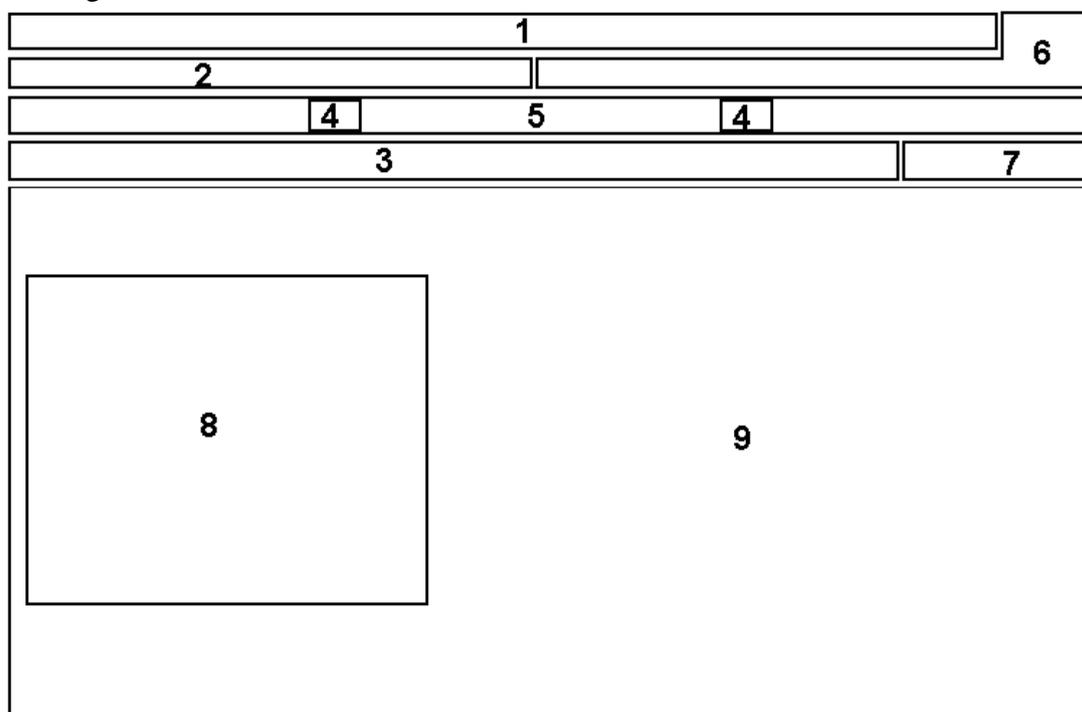


FIGURA 5.5 – Interface do ambiente de ensino no modo livre dividida por áreas de visualização

Área 1

Corresponde a identificação do ambiente AdaptWeb junto a identificação da disciplina. O nome da disciplina e do curso do usuário (inseridos pelo autor na fase de autoria) é adquirido pelo módulo de adaptação de conteúdo quando o usuário faz o *login* para entrar no ambiente da disciplina. O módulo então passa através de variáveis de sessão para módulo de interface adaptativa os dados de identificação correspondentes, e assim, dinamicamente o módulo de interface adapta a área 1 de acordo com a disciplina em questão.

Área 2

Esta área possui as guias que apresentam as categorias em que o aluno pode navegar sobre um mesmo conceito (conceito, exemplos, exercícios e material complementar). O AdaptWeb possui uma modelagem baseada em conceitos, isto é, para um mesmo conceito pode existir exemplos, exercícios e materiais complementares associados à ele, mas essas categorias não existem sem um conceito.

Área 3

Demonstra o caminho hierárquico do conceito que está sendo visualizado, ou seja, todos os conceitos ancestrais a ele. Por exemplo, se o usuário estiver no conceito 3.2.2.3, o caminho gera o *link* do conceito 3, do conceito 3.2, do conceito 3.2.2 e mostra conceito em questão 3.2.2.3 (por exemplo: 3 > 3.2 > 3.2.2 > 3.2.2.3).

Essa tecnologia é utilizada como um mecanismo de apoio à navegação para ajudar usuários com pouca experiência no hiperespaço, ajudando-os a entender a organização do material instrucional, auxiliando-os e identificando onde estão dentro do hiperespaço, reduzindo assim a desorientação. A desorientação ou "perdido no espaço" surge a partir da necessidade do leitor saber em que ponto da rede ele está, de onde veio, e como ir para outra parte do hiperdocumento. Para implementar o caminho a interface adaptativa possui uma função que busca recursivamente no XML de estrutura de conceitos da disciplina (por curso) os conceitos ancestrais do conceito em questão. Uma explicação mais detalhada da implementação é feita na seção 5.4.3.

Área 4

Identifica a navegação direta entre as páginas (conceitos) ligadas hierarquicamente sobre um determinado conceito (conceito pai, irmão à esquerda, irmão à direita e filho). Como exemplo, se o conceito que está sendo visualizado for o 3.2.2, os *links* dos conceitos que podem ser ativados (se existirem) são respectivamente o 3.2, o 3.2.1, o 3.2.3 e o 3.2.2.1. Se estes conceitos não existirem, o ícone correspondente fica desativado (desaparece), indicando assim a falta do conceito em questão. Esse mecanismo de navegação entre as páginas ajuda usuários evitando a sobrecarga cognitiva, utilizando a técnica orientação direta para prover adaptabilidade. A sobrecarga se deve às contínuas tomadas de decisão de quais *links* seguir e quais descartar, dado um grande número de escolhas disponíveis.

Esta área é reservada somente no modo livre, onde todos os conceitos podem ser aprendidos sem a necessidade de conhecer os pré-requisitos. Já no modo tutorial esse mecanismo de navegação entre as páginas não é utilizado pois não necessariamente um conceito é liberado sequencialmente, fazendo assim com que a navegação direta deste mecanismo não seja apropriada. A implementação dessa navegação é feita observando o

XML de estrutura de conceitos da disciplina, utilizando a API DOM para hierarquizar e assim, procurando os ancestrais, irmãos e descendentes para prover a navegação.

Área 5

Essa área é reservada para identificar o conceito corrente, mostrando o nome do conceito capturado do XML de estrutura de conceito da disciplina. Toda vez que o usuário muda de um conceito para outro, este campo também é modificado dinamicamente. O nome do conceito em questão é adquirido observando-se o XML de estrutura de conceitos. O XML possui um identificador para cada conceito (tópico), e em seguida uma descrição do conceito, que é utilizado pelo módulo de interface para a apresentação.

Área 6

Responsável pelas ferramentas que auxiliam na utilização do ambiente (os *links* Exibir Menu, Configurações, Mapa, Ajuda e *Logout*).

- Exibir Menu: Tem como funcionalidade abrir o menu de conteúdo. Existem quatro menus, cada um para as quatro categorias sobre um conceito, referentes às divisões do material instrucional (menu de conceito, menu de exemplos, menu de exercícios, e menu de materiais complementares), referente aos tópicos de conceito. Cada menu somente pode ser visualizado se sua guia estiver ativada na área 2;
- Configurações: Esse *link* ativará uma página onde o usuário poderá personalizar o ambiente e ainda possuirá ferramentas de apoio como enviar trabalho e imprimir determinado conceito. Neste protótipo ainda não implementado;
- Mapa: Mostra um mapa global da disciplina;
- Ajuda: Será um *link* para que o usuário obtenha informações sobre o ambiente de ensino, possuir uma lista de dúvidas mais frequentes. Neste protótipo ainda não implementado;
- *Logout*: *Link* para que o usuário saia do ambiente da disciplina.

Área 7

Indicada para fazer pesquisas sobre um determinado conteúdo. O autor pode colocar palavras-chaves para cada conceito na fase de autoria, e posteriormente fica armazenado no XML de estrutura de conceito. O módulo de interface busca no XML de estrutura de conceito a palavra pela *tag* correspondente e também pela descrição do conceito com sua respectiva abreviação. Como este recurso é um recurso mais avançado, ele foi adotado somente no modo de navegação livre.

Área 8

Corresponde ao menu de conteúdo. Existem quatro menus de conteúdo: o menu de conceitos, o menu de exemplo, o menu de exercícios e o menu de material complementar. O menu de cada categoria pode ser ativado se a guia correspondente estiver selecionada (Área 2). Os menus são baseados no conceito de cada tópico, e além disso, um mesmo conteúdo pode gerar diferentes menus de conceitos de acordo com o curso do usuário, seu modo de navegação e seu conhecimento. Para implementar os menus, leva-se em conta o XML de estrutura de conceito, a escolha pelo modo de

navegação na sessão e o *log* do usuário, sabendo assim o estado do modelo do usuário dinamicamente, e adaptando sempre que necessário.

Área 9

Essa área é reservada para o conteúdo do material instrucional do autor propriamente dito, que é filtrado apropriadamente pelo módulo de adaptação de conteúdo baseada no modelo do usuário.

Existem várias técnicas e métodos para a implementação da adaptabilidade, e na Interface do ambiente de ensino pode-se perceber a orientação global (Mapa) que é uma técnica utilizada para mostrar todo o hiperespaço, a condução global (na área 3 – caminho hierárquico) usada para informar o menor caminho, history list (menu tópico) que faz uma lista geral e também quando entra de novo no ambiente volta onde parou, o suporte a navegação da área 3 é uma orientação e condução local pois consiste em auxiliar o usuário a entender seu posicionamento no hiperespaço e este pode escolher modificar esta posição. Todas as técnicas são baseadas nas características dos alunos, representadas em seu modelo [BRU 99].

5.6 Monitoramento

O monitoramento do aluno é feito pelo módulo de interface adaptativa a cada tópico visitado. É adquirida toda a informação necessária sobre o usuário para adaptar a navegação. Essas informações sobre o usuário são armazenadas e incluídas no modelo do usuário, gerando posteriormente a adaptação pelo conhecimento do usuário. O módulo de interface adaptativa utiliza a tabela de *log* para verificar o conhecimento adquirido pelo aluno.

A tabela de *log* do usuário do banco de dados MySQL (*log_usuario*) foi estruturada com os seguintes campos:

- *id_log*: campo auto-incremental do banco de dados com valor único que identifica a posição de um registro do *log*;
- *id_usuario*: identificador do usuário;
- *id_disciplina*: identificador da disciplina;
- *id_curso*: identificador do curso cuja disciplina está sendo ofertada;
- *tecnologia*: identifica a tecnologia que o usuário optou (*dial-up* ou ADSL);
- *modo_naveg*: neste campo é marcado o modo de navegação do aluno (tutorial ou livre);
- *tópico*: mostra o número do tópico (por exemplo 2.2.3);
- *data_acesso*: apresenta a data de acesso;
- *hora_acesso*: marca a hora que acessou o *link*;
- *menu*: mostra por qual categoria está navegando (conceito, exemplo, exercício ou material complementar)

O módulo de interface adaptativa possui filtros que utilizam a tabela para adaptar a navegação do usuário, que serão vistos a seguir.

5.7 Filtros

Os conteúdos instrucionais no ambiente AdaptWeb são estruturados e armazenados em documentos XML pelo módulo de armazenamento em XML. Por essa razão, verificou-se o que poderia ser feito para adaptar o domínio para cada usuário. Chegou-se a conclusão que o processamento dos documentos XML podem ser feitos utilizando filtros, que são mecanismos para prover adaptação baseados nas regras de adaptação. Uma forma de se processar documentos XML de maneira a viabilizar a construção de aplicações que aproveitem ao máximo a riqueza de um documento é o modelo de objetos DOM (*Document Object Model*) [PIM 2001].

Segundo Pimentel [PIM 2001] “construir um programa para processar documentos XML e viabilizar aplicações sobre esses documentos pode se tornar uma tarefa árdua se não for utilizado alguma ferramenta de suporte. De fato, o processamento de um documento XML envolve:

- carregar os dados da origem;
- separar os diversos tipos de dados – elementos, cadeias de caracteres, comentários, etc;
- expandir todas as entidades usadas no arquivo;
- verificar se o documento é válido;
- montar uma estrutura que represente o documento;
- navegar por essa estrutura e acessar cada componente do documento, etc”.

No sentido de facilitar essas tarefas, o W3C (*World Wide Web Consortium*) estabeleceu a recomendação DOM. Neste contexto, os documentos XML são representados em uma estrutura de árvore que reflete a sua estrutura lógica hierárquica [PIM 2001]. DOM é uma API que define uma forma de acesso e manipulação à árvore de objetos correspondente ao documento. Assim, DOM permite que programas *scripts* acessem e manipulem dinamicamente a estrutura, conteúdo e estilo de apresentação de documentos XML e HTML [PIM 2001].

DOM foi proposto para facilitar a computação sobre os documentos, ou seja, um processador DOM recebe um arquivo XML e disponibiliza sua representação como uma árvore de objetos para aplicações. Ele suporta a manipulação da estrutura e do conteúdo do documento, permitindo que programas possam adicionar, modificar e apagar nós da árvore de objetos a ele associada [PIM 2001].

A seguir serão especificados os filtros desenvolvidos pelo módulo de interface adaptativa que utilizam o XML de estrutura de conceito da disciplina, a tabela de *log do usuário* do banco de dados MySQL, e se comunica com o módulo de adaptação de conteúdo por variáveis de sessão. Para testar os filtros desenvolvidos do módulo de interface adaptativa, foi desenvolvida um módulo da primeira disciplina no Ambiente AdaptWeb chamada de “Computação Algébrica e Numérica” para os cursos de ciência da computação, engenharia e matemática pela professora Dra. Maria Angélica C. Brunetto do departamento de Computação da UEL.

Os filtros implementados estão dispostos em quatro *scripts*, para cada categoria específica (conceito, exemplos, exercícios e material complementar). Esses *scripts* são utilizados para qualquer disciplina cadastrada no ambiente, desde que a mesma passe anteriormente pelo módulo de autoria e em seguida pelo módulo de armazenamento em XML.

A tabela 5.2 demonstra todos os tópicos conceitos pertencentes à disciplina em questão. Toda a modelagem foi feita baseando-se na estrutura de conceitos obtida na fase de autoria.

TABELA 5.2 – Estrutura Hierárquica de Conceitos da Disciplina Computação Algébrica e Numérica

1. Contexto e Objetivos da Computação Algébrica e Numérica
2. Sistemas Lineares de Equações Algébricas
2.1 Introdução
2.2 Métodos Diretos
2.2.1 Método de Gauss
2.2.1.1 Algoritmo da Triangularização
2.2.1.2 Algoritmo da Retrossubstituição
2.2.2 Método de Gauss com pivotamento
2.2.2.1 Condicionamento de Matrizes
2.2.3 Método da Decomposição LU
2.2.4 Método de Cholesky
2.3 Métodos Iterativos
2.3.1 Método de Jacobi
2.3.2 Método de Gauss-Seidel

A Tabela 5.3 foi estruturada para mostrar os cursos que visualizam cada conceito, e também para ilustrar os pré-requisitos de cada conceito para a disciplina Computação Algébrica e Numérica. Existem duas colunas para denotar os pré-requisitos, a do lado esquerdo significa os pré-requisitos navegacionais, pois no Ambiente AdaptWeb foi definido como uma árvore hierárquica, onde sempre o conceito pai é pré-requisito para os filhos, e a coluna do lado direito mostra os pré-requisitos conceituais definidos pelo autor. Pode-se perceber, por exemplo, que o conceito 2.2.2 *Método de Gauss com pivotamento* pode ser assistido pelos alunos dos cursos de Ciência da Computação e de Engenharia, mas não para os alunos de Matemática.

TABELA 5.3 – Estrutura dos Pré-requisitos e da Adaptabilidade por Cursos

	Pré-requisito	Comp	Eng	Mat
1. Contexto e Objetivos da Computação Algébrica e Numérica	--	X	X	X
2. Sistemas Lineares de Equações Algébricas	--	X	X	X
2.1 Introdução	2	X	X	X
2.2 Métodos Diretos	2	X	X	X
2.2.1 Método de Gauss	2.2	X	X	X
2.2.1.1 Algoritmo da Triangularização	2.2.1	X	X	X
2.2.1.2 Algoritmo da Retrossubstituição	2.2.1	X	X	X
2.2.2 Método de Gauss com pivotamento	2.2	2.2.1	X	X
2.2.2.1 Condicionamento de Matrizes	2.2.2	X	X	
2.2.3 Método da Decomposição LU	2.2	X	X	
2.2.4 Método de Cholesky	2.2	2.2.3	X	X
2.3 Métodos Iterativos	2	X	X	X
2.3.1 Método de Jacobi	2.3	X	X	X
2.3.2 Método de Gauss-Seidel	2.3	2.3.1	X	X

O XML de estrutura de conceito resultante desta disciplina está no anexo 1. Todos os filtros do módulo de interface que acessam o arquivo XML utilizam este documento XML da disciplina para buscar os dados.

Na primeira etapa para o aluno utilizar o ambiente de ensino, ele deve fazer a autenticação, selecionando seu *e-mail* e sua senha como demonstra a figura 5.6.

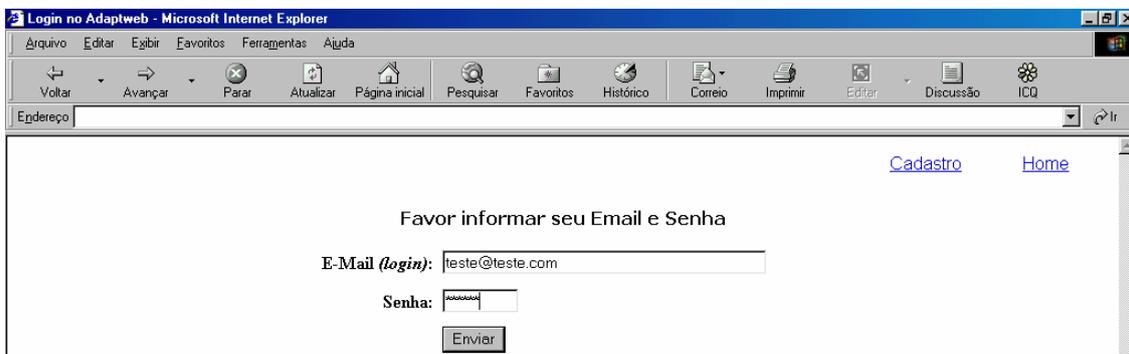


FIGURA 5.6 – Autenticação do usuário

Neste instante o módulo de adaptação de conteúdo identifica o usuário e mostra as disciplinas nas quais ele está cadastrado e o curso correspondente. A figura 5.7 mostra este processo para um aluno do curso de ciência da computação.

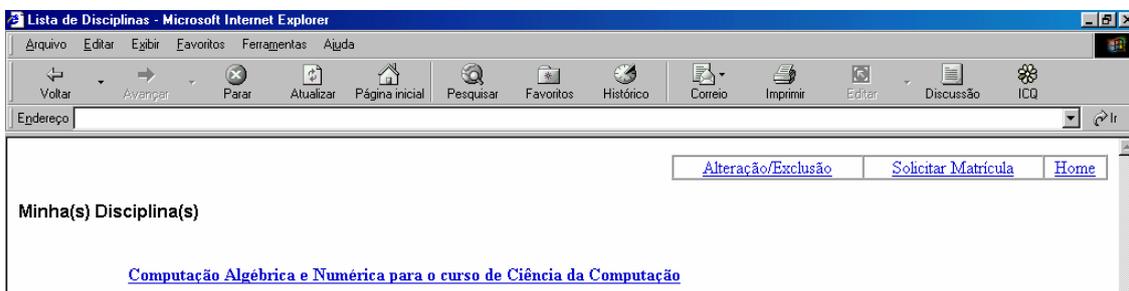


FIGURA 5.7 – Disciplinas que o aluno pode assistir

Após o aluno selecionar a disciplina a qual deseja assistir, ele deve escolher a sua conexão de rede (*dial-up/ADSL*) e o modo de navegação desejado (tutorial/livre). Essa escolha é armazenada no banco de dados pelo módulo de adaptação de conteúdo e implica na adaptação tanto do conteúdo (conexão de rede) pelo módulo de adaptação de conteúdo, quanto na adaptação da navegação e apresentação (modo de navegação) pelo módulo de interface adaptativa. A figura 5.8 apresenta a interface correspondente.

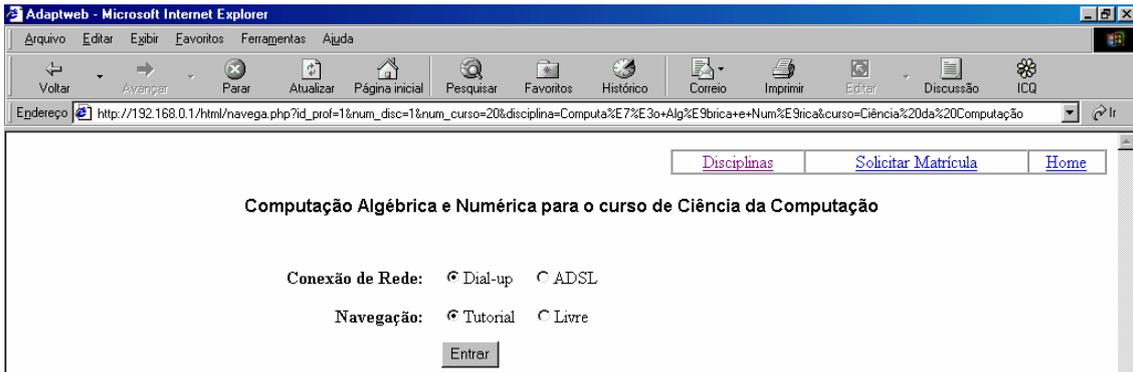


FIGURA 5.8 – Seleção da conexão de rede e do modo de navegação

Somente depois de ter selecionado todas as opções é que o usuário pode navegar pelo ambiente de ensino da disciplina. As informações necessárias sobre o usuário como a disciplina selecionada, seu curso correspondente e o modo de navegação são passadas para o módulo de interface adaptativa por variáveis de sessão.

Geração dinâmica do nome da disciplina

O nome da disciplina aparece na parte superior da interface do ambiente de ensino, e para colocar o nome especificado é passado por variável de sessão do módulo de adaptação de conteúdo o nome da disciplina cadastrada no banco de dados. A variável de sessão é a chamada $\$disciplina$. Assim, o nome que foi cadastrado no banco de dados como o nome da disciplina pelo professor é passado como variável de sessão e dinamicamente atualizado no *layout* do ambiente de ensino pelo módulo de interface adaptativa, ou seja, se o autor modificar novamente o nome da disciplina, a interface também mudará. A figura 5.9 mostra o esquema para fazer a geração do nome da disciplina e a figura 5.10 mostra a geração do nome da disciplina e sua respectiva localização na tela.

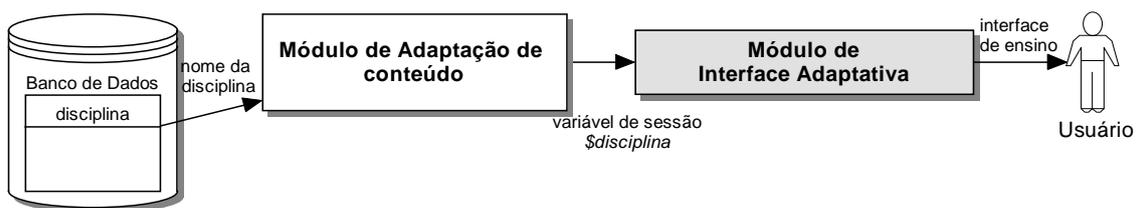


FIGURA 5.9 – Esquema para obtenção do nome da disciplina dinamicamente

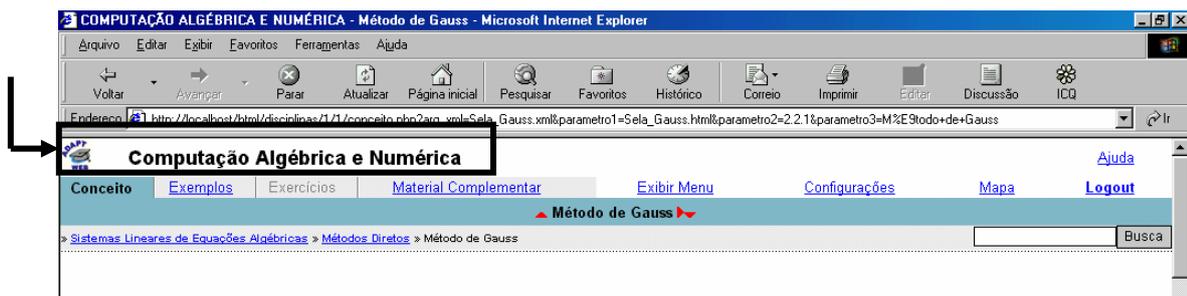


FIGURA 5.10 – Geração do nome da disciplina e sua localização

Filtro para cada grupo de usuários (por curso)

Este filtro busca no XML de estrutura de conceitos todos os tópicos que devem ser mostrados ao aluno de acordo com o seu curso descrito em seu modelo de usuário. Ele busca o tópico de conceito para posteriormente colocar esses tópicos navegacionais na interface (no menu, no caminho e no acesso direto as páginas). Para isso, é passada uma variável de sessão pelo módulo de adaptação de conteúdo que define qual curso o aluno pertence ($\$num_curso$), e o módulo de interface adaptativa percorre o XML e verifica em cada tópico, se existe o curso em questão observando a *tag curso identcurso*. Os cursos de Engenharia, Ciência da Computação e Matemática foram cadastrados no banco de dados e no XML como 10, 20 e 30 respectivamente. Pode-se observar na tabela 5.3 quais tópicos existem para cada curso. A figura 5.11 demonstra o esquema.

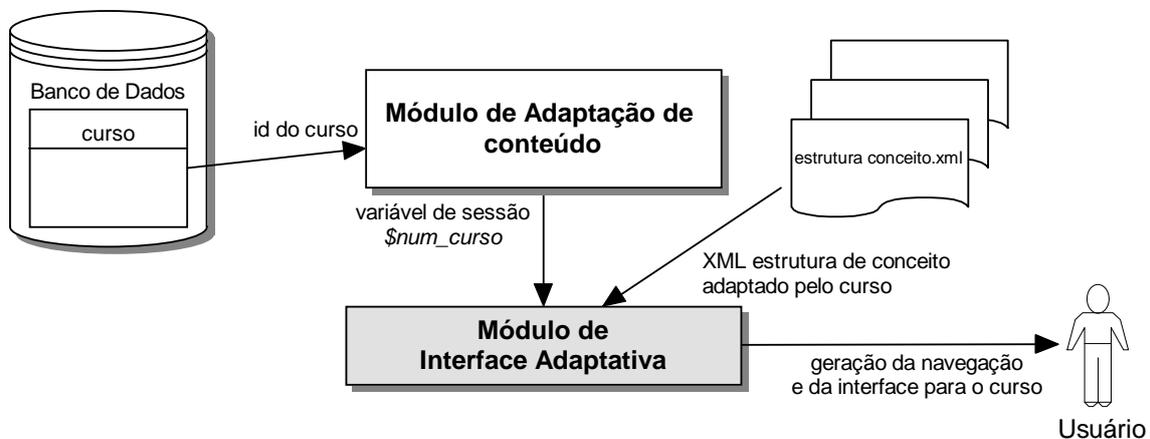


FIGURA 5.11 – Esquema para achar tópicos conceitos por curso

A figura 5.12 mostra a adaptação no menu de navegação para os cursos de ciência da computação e da matemática respectivamente.



FIGURA 5.12 – Menus de navegação para os cursos de Ciência da Computação e Matemática

Filtro para verificação das guias de exemplos, exercícios e material complementar de um determinado conceito

Após a verificação se um determinado curso pode visualizar um tópico conceito, deve-se procurar por exemplos, exercícios e material complementar deste tópico para este curso. Para isso, no XML, dentro de cada tópico conceito está cada curso, e dentro de cada curso estão os elementos que o compõe. Se a *tag elementos* estiver vazia, quer dizer que não existe nenhum exemplo, exercício, ou material complementar para este tópico e curso.

Por exemplo, quando o aluno está navegando pelo menu de conceitos e ele está acessando um conceito que possui exercícios para aquele curso, a interface dinamicamente libera na guia específica o *link* de exercício, para aquele tópico conceito. Pode-se citar que para o curso de Ciência da Computação existem exemplos, exercícios e material complementar de acordo com o que mostra a tabela 5.4.

TABELA 5.4 – Exemplos, exercícios e material complementar para o curso de Ciência da Computação

	Exemplos	Exercícios	Material Complementar
1. Contexto e Objetivos da Computação Algébrica e Numérica			
2. Sistemas Lineares de Equações Algébricas			
2.1 Introdução		X	
2.2 Métodos Diretos			
2.2.1 Método de Gauss	X		X
2.2.1.1 Algoritmo da Triangularização			
2.2.1.2 Algoritmo da Retrossubstituição			
2.2.2 Método de Gauss com pivotamento	X		X
2.2.2.1 Condicionamento de Matrizes			
2.2.3 Método da Decomposição LU			X
2.2.4 Método de Cholesky			X
2.3 Métodos Iterativos			
2.3.1 Método de Jacobi			X
2.3.2 Método de Gauss-Seidel			X

A figura 5.13 explica o esquema para a habilitação das guias de exemplo, exercícios e material complementar.

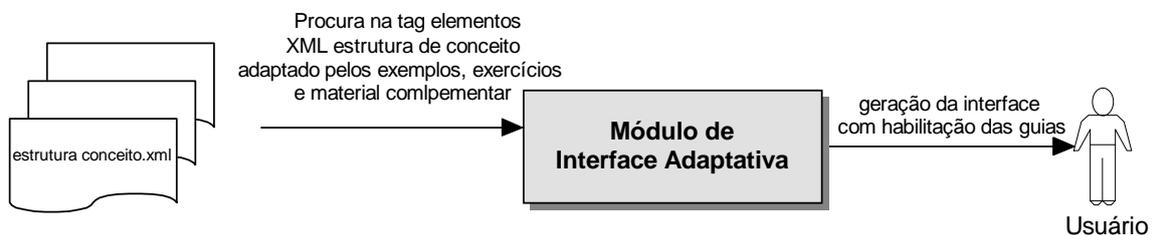


FIGURA 5.13 – Esquema para habilitação das guias

A figura 5.14 mostra uma tela da interface para um aluno do curso de Ciência da Computação, quando este está visualizando o tópico conceito “*Método de Gauss*”. Como existem *exemplos* e *material complementar* deste tópico, as guias correspondentes estão habilitadas, o que não ocorre com a guia *exercícios*.

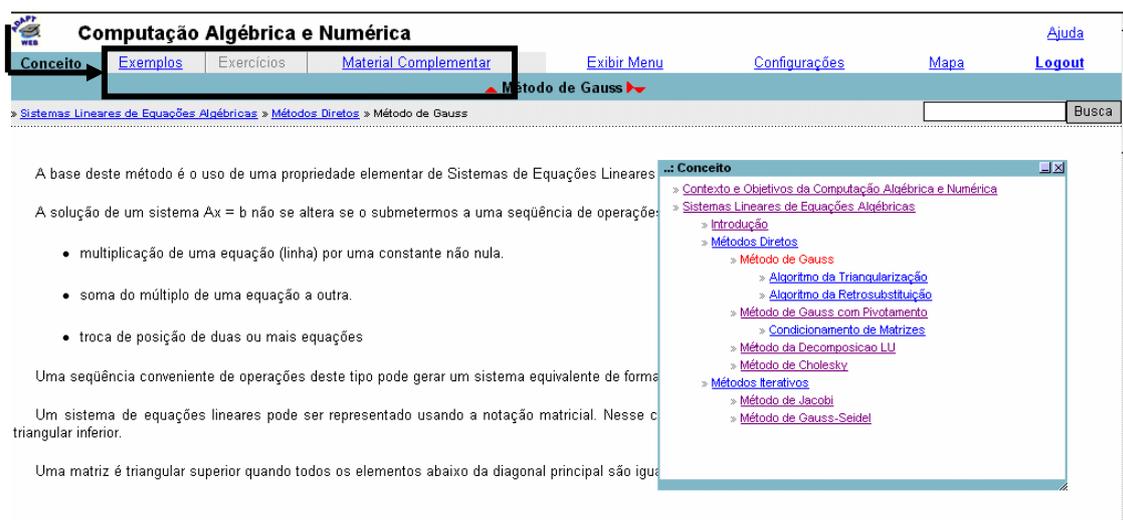


FIGURA 5.14 – Tela da interface com o tópico Método de Gauss visualizado e as guias de *exemplos* e *material complementar* habilitadas

Geração dinâmica do tópico conceito corrente

O tópico conceito que está sendo visualizado é chamado de tópico atual ou corrente, e ele é dinamicamente gerado no lugar apropriado para identificação.

Para gerar o nome do tópico corrente na Interface de ensino (área 5 figura 5.5) toda vez que o usuário selecionar um novo tópico (ou seja, clicar em seu *link*) é passado por parâmetro o tópico em questão para a atualização na Interface e geração do menu de navegação, caminho e navegação direta.

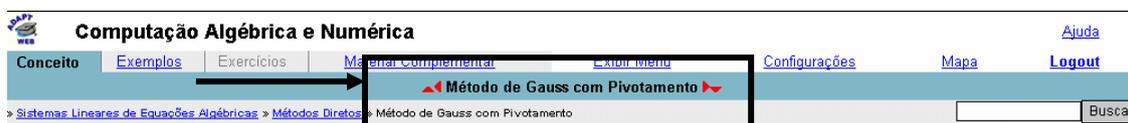


FIGURA 5.15 – Geração do nome do tópico que está sendo visualizado

Filtro para prover Caminho Hierárquico (Caminhamento)

Este filtro faz a busca hierárquica dos tópicos no XML, mostrando todos os ancestrais ao tópico que está sendo visualizado.

a) Filtro para conceito

Para os conceitos, os ancestrais mostrados no caminho são habilitados para navegação. Para isso, existe uma função que verifica no XML os conceitos ancestrais ao conceito que está sendo visualizado. A figura 5.16 explica o esquema para prover caminho na Interface.

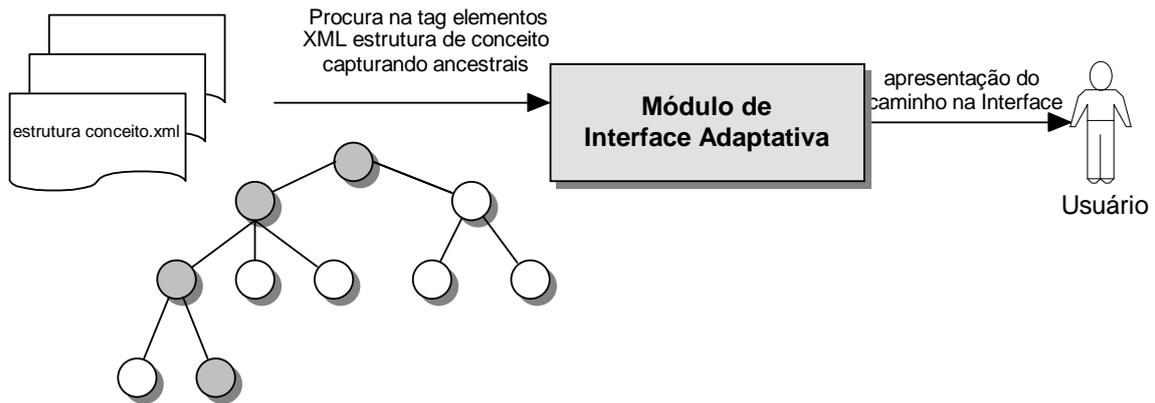


FIGURA 5.16 – Esquema para achar o caminho hierárquico do conceito

Para demonstrar este recurso navegacional, a figura 5.17 apresenta o caminho (ancestrais) ao tópico conceito 2.2.2 *Método de Gauss com Pivotamento* da disciplina Computação Algébrica e Numérica para o curso de ciência da computação. Todos os tópicos estão demonstrados na tabela 5.2.



FIGURA 5.17 – Caminho hierárquico

b) Filtro para exemplos, exercícios e material complementar

Caso o aluno esteja nas outras categorias (exemplos, exercícios, material complementar), são verificados se os tópicos ancestrais existem, por exemplo, se na categoria exercícios existem exercícios associados ao tópico ancestral para aquele modelo de usuário. Se o tópico não existir, o mesmo não pode ser habilitado para navegação. A figura 5.18 apresenta a esquemática para a verificação do caminho nessas categorias. O símbolo “X” representa nós na estrutura que não podem ser visualizados por não possuírem materiais da categoria.

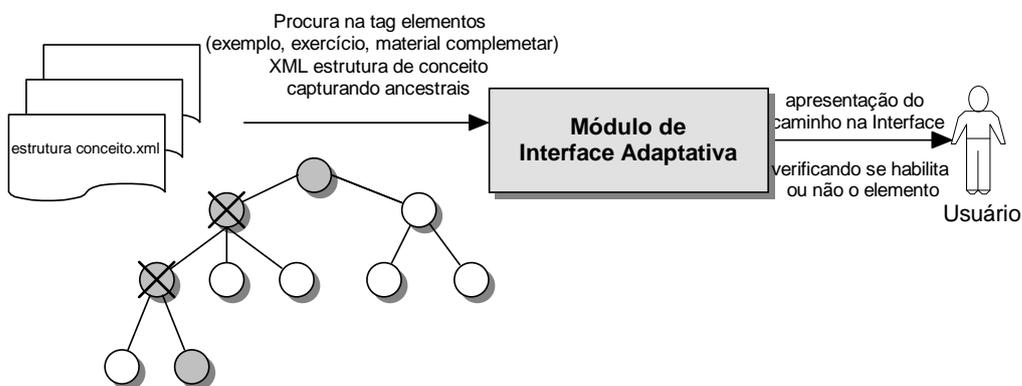


FIGURA 5.18 – Esquema para caminho em exemplo, exercício e material complementar

A figura 5.19 apresenta o caminho quando um aluno do curso de ciência da computação está navegando pelo material complementar. Observe que os tópicos ancestrais *Sistemas Lineares de Equações Algébricas* e *Método Diretos* não estão habilitados, pois não existe material complementar para estes tópicos.



FIGURA 5.19 – Caminho hierárquico quando navegando pelo material complementar

Filtro navegação direta

A navegação direta é a navegação entre os tópicos vizinhos mais próximos ao tópico (tópico pai, tópico irmão à esquerda, tópico irmão à direita e tópico filho, na figura 5.20 – 1, 2, 3 e 4 respectivamente). Primeiro deve-se verificar se os tópicos diretos existem, se não existir não aparece a seta correspondente, se existe verifica cada um, sendo que quando o usuário coloca o mouse na região da seta na interface, aparece o texto (*hint*) com o nome do tópico, como mostra a figura 5.21.

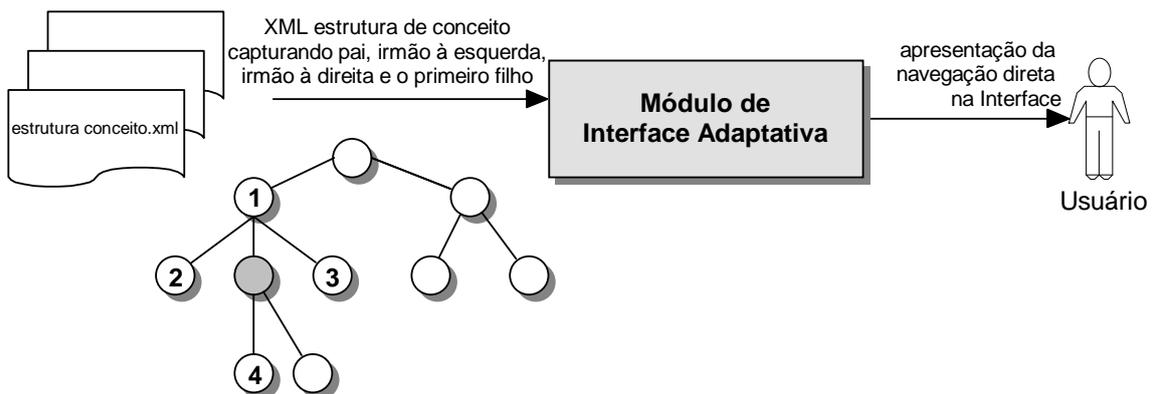


FIGURA 5.20 – Esquema para apresentação da navegação direta

A figura 5.21 mostra o tópico atual com a navegação direta e com o tópico filho escolhido.



FIGURA 5.21 – Navegação direta

Função escolha da interface no MODO TUTORIAL e MODO LIVRE

O aluno pode escolher dois modos de navegação: o tutorial ou o livre. No modo tutorial ele navega de acordo com os pré-requisitos definidos pelo professor na fase de autoria. Já no modo livre ele pode navegar por onde escolher, sem precisar conhecer um tópico específico.

Para isso, quando o aluno faz a autenticação para entrar no ambiente da disciplina, ele seleciona um modo de navegação. Assim, o módulo de adaptação do conteúdo verifica sua opção e o módulo de interface adaptativa captura o valor por uma variável de sessão ($\$naveg$). Posteriormente, o módulo de interface adaptativa apresenta a interface escolhida pelo usuário. A figura 5.22 demonstra o esquema para a apresentação da Interface no modo de navegação escolhido.

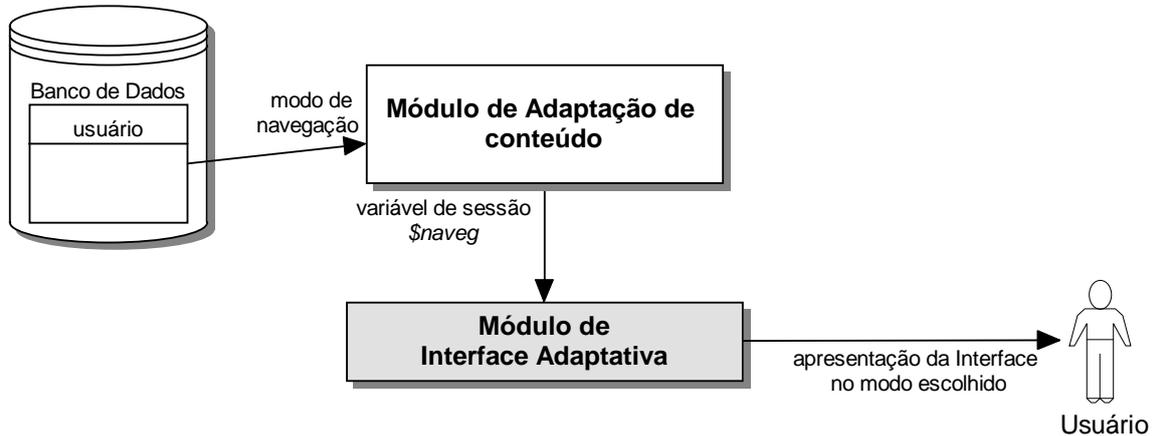


FIGURA 5.22 – Esquema para a apresentação da Interface

A figura 5.23 e a figura 5.24 mostram as duas interfaces iniciais (no modo tutorial e no modo livre respectivamente) para um aluno de ciência da computação.

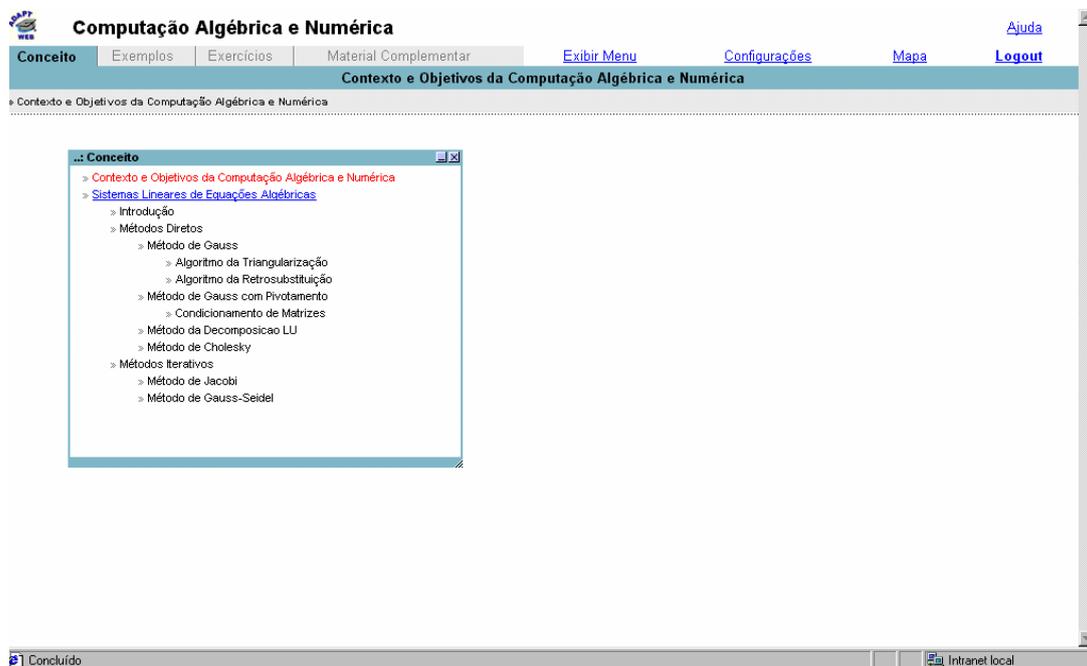


FIGURA 5.23 – Interface inicial da disciplina no modo TUTORIAL

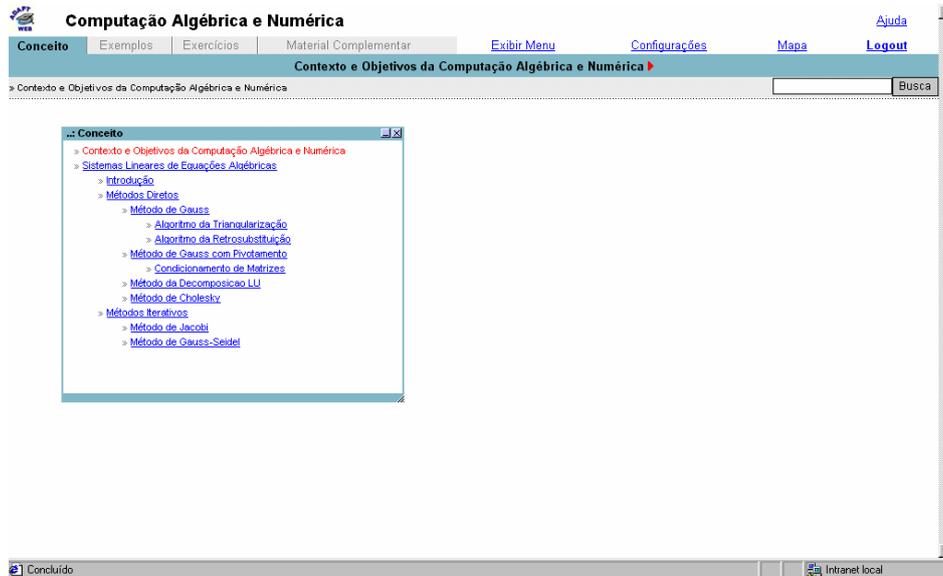


FIGURA 5.24 – Interface inicial da disciplina no modo LIVRE

Filtro de Adaptação do Menu de Navegação

Este filtro é utilizado tanto no modo tutorial, quanto no livre. Ele obtém o tópico do arquivo XML, analisa a tabela de *log* do usuário no banco de dados e gera a navegação pelo menu fazendo *links* distintos para tópicos visitados (conhecimento adquirido) e não visitados.

a) Geração do menu de navegação de conceito no modo livre

A cada tópico conceito analisado, o filtro gera a identificação (hierarquia) do menu de acordo com o nível do nó, analisa a tabela de *log* de usuário, e apresenta o menu observando o resultado obtido nesta verificação. Os *links* para os conceitos que já foram visitados (adquiridos) são apresentados na cor roxa, e os *links* conceitos que ainda não foram são apresentados na cor azul. Também é passado por parâmetro o tópico conceito que está sendo visualizado, este conceito é mostrado na cor vermelha.

A figura 5.25 mostra como é realizada a hierarquia de cada nó.

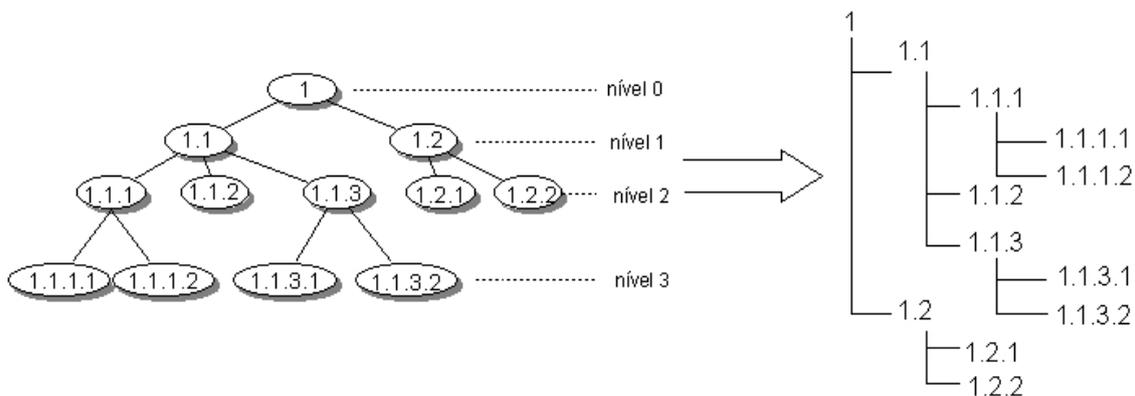


FIGURA 5.25 – Esquema da hierarquia do menu

A figura 5.26 apresenta o menu de navegação de um aluno do curso de ciência da computação, cujos conceitos aprendidos (ou seja, já inseridos na tabela de *log*) são:

Contexto e Objetivos da Computação Algébrica e Numérica; Sistemas Lineares de Equações Algébricas; e Introdução, que estão em roxo. Os conceitos que ainda não foram aprendidos estão em azul, e o conceito que está sendo aprendido no momento (*Método de Gauss*) em vermelho.

Como os tópicos visitados/aprendidos são armazenados no banco de dados, não importa em qual computador o usuário se encontra, as cores serão sempre as correspondentes. Ou seja, o AdaptWeb pode ser acessado de qualquer computador, até mesmo de um que o usuário nunca havia acessado, que as cores ficam correspondentes.

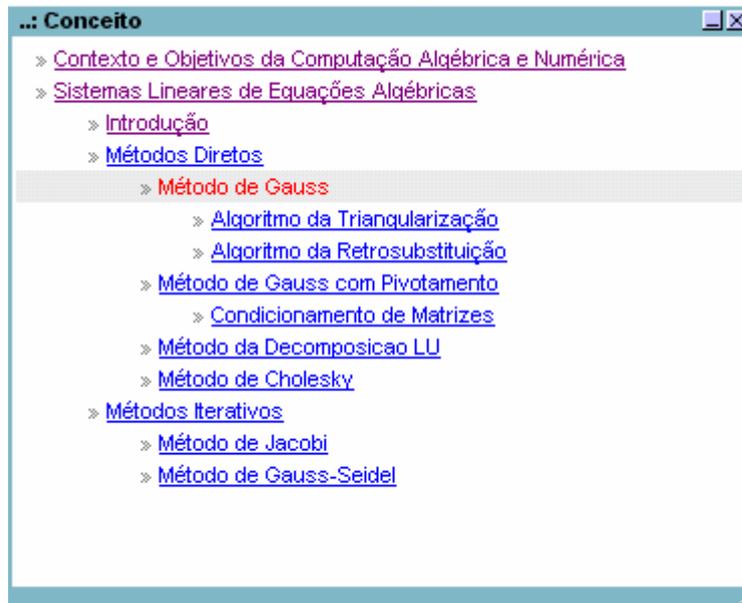


FIGURA 5.26 – Menu de navegação para tópicos conceitos no modo livre identificando conceitos visitados/aprendidos, não visitados/não aprendidos e atual.

b) Geração do menu de navegação de exemplos, exercícios e material complementar no modo livre

Para gerar o menu de navegação de exemplos, exercícios e material complementar, o módulo de interface adaptativa faz o mesmo processo apresentado anteriormente para o tópico conceito (seção a), e também faz a verificação se existe um tópico de exemplos, exercícios ou material complementar correspondente, pois se não existir, este deve ficar desabilitado, na cor cinza claro identificando assim que o tópico não existe.

A figura 5.27 demonstra os menus de conceito, exemplos, exercícios e material complementar para um aluno do curso de ciência da computação. Optou-se por mostrar os tópicos, mesmo desabilitados (sem exemplos, ou exercícios ou material complementar), para que o aluno tenha consistência, ficando os menus correspondentes ao menu de conceito.



FIGURA 5.27 – Menus dos tópicos para o curso de ciência da computação (a) Conceito (b) Exemplos (c) Exercícios (d) Material Complementar

c) Geração do menu de navegação de conceito no modo tutorial

Quando o aluno está no modo tutorial, o menu de navegação de conceito, além de fazer todo o processo apresentado na seção (a), também deve apresentar o menu de acordo com os pré-requisitos do autor. Somente é habilitado um tópico conceito para o aluno, se ele já possui o conhecimento necessário para o aprendizado, ou seja, se ele já visitou os pré-requisitos.

Para isso, o módulo de interface adaptativa observa o XML para a verificação dos pré-requisitos (elemento *prereq identprereq*) de cada tópico conceito. Ele também analisa a tabela de *log* do usuário no banco de dados para observar os tópicos conceitos já visitados. Somente serão habilitados os conceitos cujos pré-requisitos já foram visitados. As cores seguem o mesmo padrão, roxo para tópicos visitados, azul para tópicos habilitados ainda não visitados e vermelho para tópico atual. Ainda existe uma outra cor para os tópicos cujos pré-requisitos ainda não foram satisfeitos e ainda não estão habilitados, que estão em preto. O esquema para geração do menu de navegação é apresentado na figura 5.28.

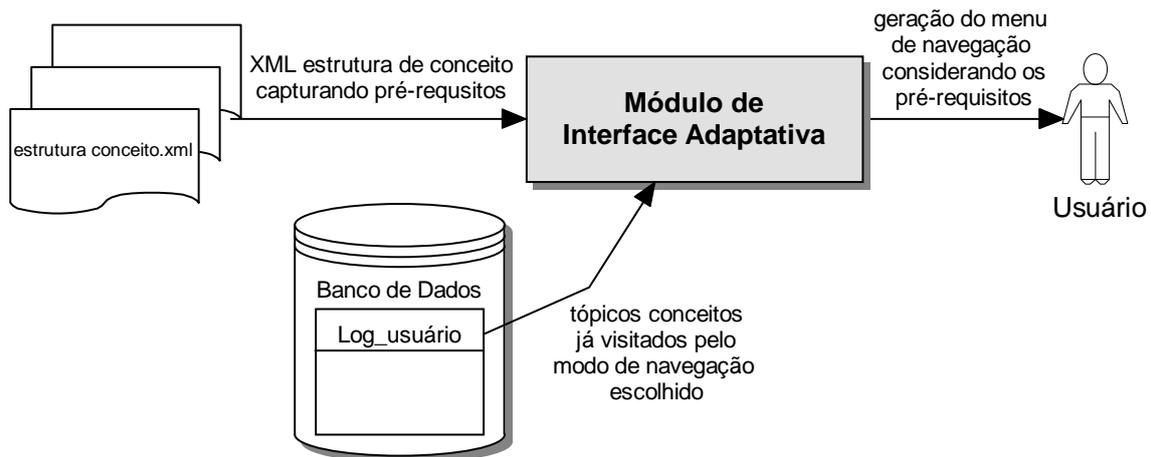


FIGURA 5.28 – Esquema para geração do menu de navegação

A figura 5.29 mostra o menu de navegação no modo tutorial para um aluno do curso de ciência da computação. Os conceitos que estão em roxo já foram visitados/aprendidos, os conceitos em azul estão habilitados mas ainda não foram vistos, o conceito que está em vermelho é o atual, e os conceitos que estão em preto ainda não podem ser visitados pois seus pré-requisitos ainda não foram atingidos.

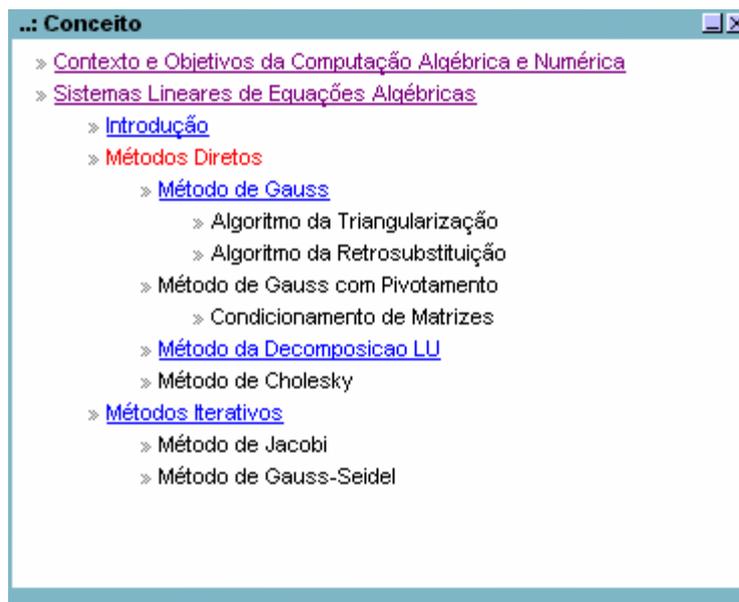


FIGURA 5.29 – Menu de navegação de conceito no modo tutorial

d) Geração do menu de navegação de exemplos, exercícios e material complementar no modo tutorial

A geração do menu de navegação das categorias exemplos, exercícios e material complementar no modo tutorial utiliza a mesma função apresentada na seção (b), e também somente deve habilitar um tópico se o(s) seu(s) pré-requisito(s) já foram visitados. Os pré-requisitos são para os tópicos de conceitos, porém, se não habilitar para o tópico conceito, então não habilita para os demais (exemplos, exercícios e material complementar).

A figura 5.30 apresenta o menu de navegação do material complementar para o curso de ciência da computação. Os tópicos em cinza significam que não existe material complementar para eles. O tópico na cor vermelha significa que é este tópico que está sendo visualizado. Os tópicos na cor azul são os que estão habilitados, mas ainda não foram visitados, os em roxo já foram visitados e os em preto ainda não possuem os pré-requisitos liberados, ou seja, ainda não possuem o conhecimento necessário.

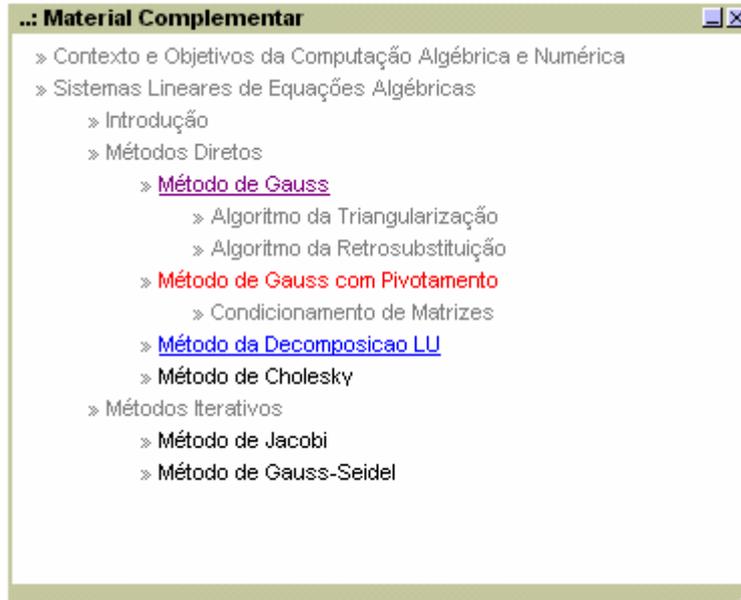


FIGURA 5.30 – Menu de navegação de Material Complementar no modo tutorial

Filtro de pesquisa

Este filtro é encarregado de buscar no XML todas as ocorrências da palavra buscada nos campos nome da disciplina (*desktop*), abreviação (*abreviacao*) e palavra-chave (*palchave*). Se encontrada, o módulo de interface adaptativa apresenta os tópicos conceitos que contêm a palavra buscada. A figura 5.31 apresenta o esquema do filtro para pesquisa.

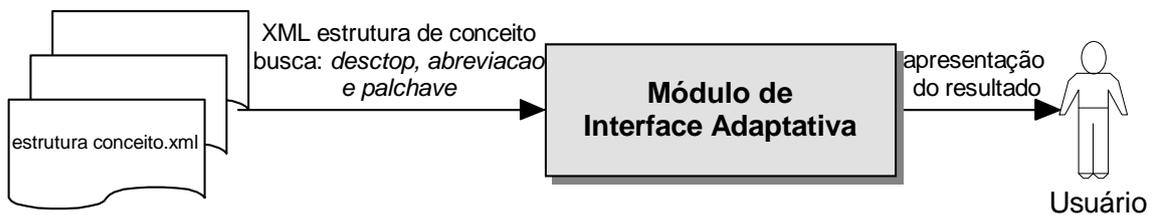


FIGURA 5.31 – Esquema para pesquisa

A figura 5.32 mostra o resultado da pesquisa quando o aluno procura pela palavra “*Méto*”.

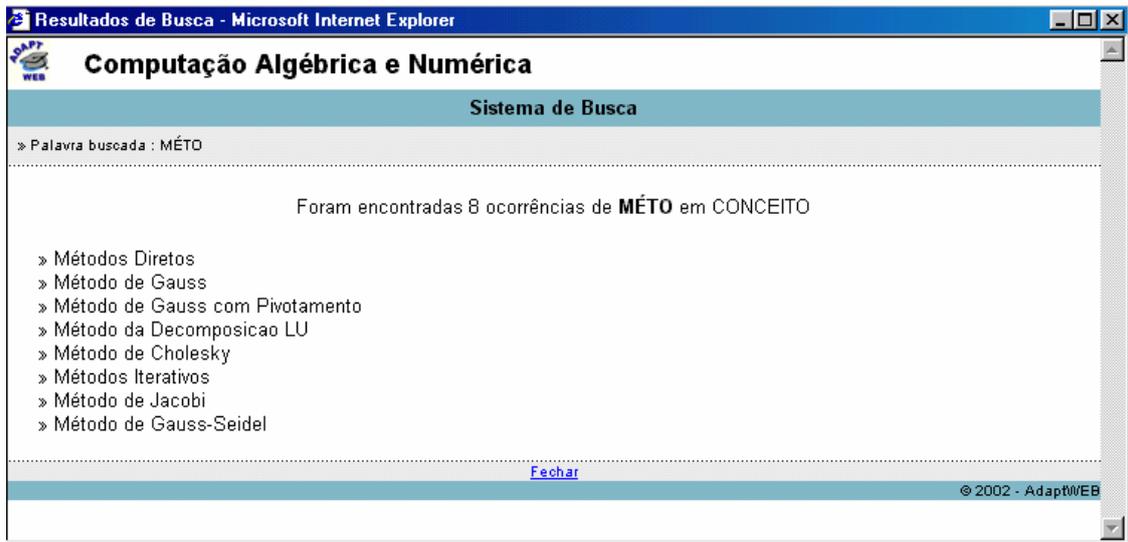


FIGURA 5.32 – Resultado da pesquisa

Filtro Mapa

Este filtro funciona basicamente igual ao filtro de adaptação do menu navegacional, porém todos os menus são mostrados, e o usuário pode navegar em qualquer um deles. No mapa, fica destacado na cor vermelha onde o usuário se encontra no momento. O mapa é de acordo com modo de navegação do aluno, sendo que se ele estiver no modo tutorial somente será habilitado o que ele pode ver no momento, e no modo livre todos os *links* ficam habilitados.

A figura 5.33 apresenta uma porção do mapa da disciplina.

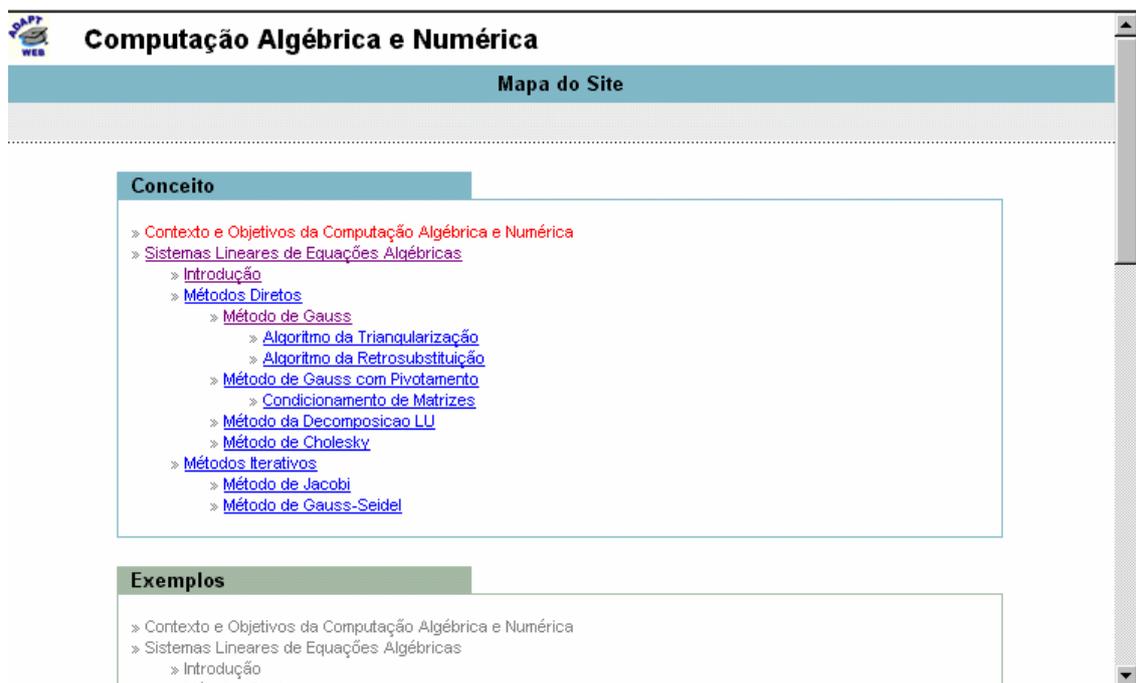


FIGURA 5.33 – Mapa da disciplina Computação Algébrica e Numérica

6 Conclusão

As aplicações de sistemas hipermídia convencionais oferecem as mesmas páginas e os mesmos *links* para os usuários, porém os usuários diferem muito em seus objetivos, experiências e conhecimento, necessitando assim de informação e navegação diferenciada [SAN 2000]. Sistemas Hipermídia Adaptativos podem ser úteis na área educacional pois permitem que vários usuários utilizem o ambiente de forma adaptada. O sistema observa as características dos usuários representadas em seu modelo, adaptando a informação, a apresentação e a navegação para seus usuários. A adaptabilidade pode proteger o usuário da desorientação, pois identificados os objetivos e conhecimentos sobre os conteúdos, um SHA oferece apoio à navegação, limitando o espaço de navegação e sugerindo *links* a seguir [SAN 2000].

Existem duas classes de adaptação, a apresentação adaptativa e a navegação adaptativa. No AdaptWeb, a apresentação adaptativa é realizada pelos módulos de interface adaptativa que adapta o *layout* da interface e as informações sobre a navegação em cada unidade de apresentação, e pelo módulo de adaptação de conteúdo que adapta o conteúdo do autor (informação). Os dois módulos em conjunto adaptam a interface como um todo. A navegação adaptativa é realizada pelo módulo de interface adaptativa que adapta a estrutura de navegação em cada conceito e entre os conceitos de cada disciplina.

O trabalho desenvolvido utilizou técnicas de navegação (orientação direta, remoção, anotação e desabilitação de *links*) e apresentação adaptativa (página variante) para adaptar a navegação e a apresentação da interface no ambiente AdaptWeb. As características dos alunos coletadas e armazenadas no modelo do usuário utilizadas pelo módulo de interface adaptativa para prover a adaptação foram (1) o curso do usuário – sua formação; (2) sua preferência navegacional – modo tutorial e livre e; (3) conhecimento do usuário, sendo que para esta característica ser analisada o módulo de interface adaptativa faz o monitoramento do usuário, e a cada clique, armazena na tabela de *log* do usuário o novo conhecimento adquirido.

Utilizou-se várias técnicas em conjunto para prover a navegação e a apresentação adaptativa, sendo elas: orientação direta, remoção, anotação e desabilitação de *links* para adaptar pela formação do usuário; anotação e desabilitação de *links* para adaptar o conhecimento do usuário; e página variante para adaptar pela preferência do modo de navegação do usuário.

Para testar os conjuntos de técnicas utilizadas no módulo de interface adaptativa, e os demais módulos do ambiente, foi feito um estudo de caso com um módulo da disciplina Computação Algébrica e Numérica, ministrado pela professora Maria Angélica C. Brunetto da UEL, participante deste projeto. Esta disciplina foi implementada para os cursos de ciência da computação, engenharia e matemática, e todas as etapas foram realizadas com sucesso.

Os objetivos propostos foram atingidos: foi desenvolvido o módulo de Interface Adaptativa do ambiente AdaptWeb, além de feito os estudos específicos, como a investigativa de utilização de SHA dentro do contexto da educação a distância, estudo do projeto do modelo do usuário em conjunto com o módulo de adaptação de conteúdo, o projeto a interface do ambiente de ensino com usabilidade e consistência, e verificação das técnicas adaptativas que possibilitam uma posterior análise sobre o usuário.

Algumas limitações ainda são observadas, que podem ser indicadas como objetos de estudo de trabalhos futuros, como a implementação dos *links* de ajuda e de configurações, a inclusão de mais características dos usuários para uma adaptação mais personalizada e o aumento dos tipos de navegação, como por exemplo, uma totalmente guiada pelo professor, uma navegação do tipo linear em que o usuário somente pode ir para um “próximo conceito”, ou uma navegação diferenciada para cada menu de navegação (conceito, exemplos, exercícios, material complementar).

A próxima etapa a ser realizada pelos integrantes do projeto é a realização de avaliações com os usuários e análise dos resultados obtidos. Também é de interesse a integração do ambiente com o sistema AvalWeb³ [CAR 2001], pois por enquanto o módulo de interface adaptativa, descrito neste trabalho somente assume um conceito como conhecido ou não conhecido pelo usuário. A integração poderá ajudar a especificar diferentes perfis de usuários, fazendo sua avaliação de exercícios, e assim, aumentando consideravelmente os estilos cognitivos proporcionados pelo ambiente. Uma proposta pode ser a verificação do histórico navegacional do usuário e seu histórico de avaliações, tentando fazer um cruzamento dos dados obtidos para análise das técnicas de adaptabilidade, e de diferentes perfis cognitivos.

Também se propõe como trabalho futuro, a criação de um ambiente de pré-autoria onde o autor pode além de pré-estruturar sua disciplina, desenvolver diferentes *layout* para o ambiente de ensino com a utilização de folhas de estilo, tornando o ambiente mais funcional, além de fazer uso de mapas cognitivos para uma melhor representação do conhecimento.

³ Disponível no site <http://cemt.inf.ufrgs.br/avalweb/>

Anexo XML de estrutura de Conceito da disciplina Computação Algébrica e Numérica

```

<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE material SYSTEM "dtd\Estrutura_Topico.dtd">

<material disciplina="1">
  <topico numtop="1"
    desctop="Contexto e Objetivos da Computação Algébrica e Numérica"
    abreviacao="Cont. Obj. C.A.N."
    arquivoxml="Contexto.html"
    palchave="Objetivos, Computação, Algébrica, Numérica">
    <curso identcurso="10">
      <elementos>
      </elementos>
    </curso>
    <curso identcurso="20">
      <elementos>
      </elementos>
    </curso>
    <curso identcurso="30">
      <elementos>
      </elementos>
    </curso>
  </topico>
  <topico numtop="2"
    desctop="Sistemas Lineares de Equações Algébricas"
    abreviacao="Sist. Lin. de Eq. Alg"
    arquivoxml="Sistemas_de_Equacoes_Lineares_Algebraica.html"
    palchave="Sistemas, Equações">
    <curso identcurso="10">
      <elementos>
      </elementos>
    </curso>
    <curso identcurso="20">
      <elementos>
      </elementos>
    </curso>
    <curso identcurso="30">
      <elementos>
      </elementos>
    </curso>
  </topico>
  <topico numtop="2.1"
    desctop="Introdução"
    abreviacao="Intro"
    arquivoxml="Introducao.html"
    palchave="Sistemas, Equações, Introdução">
    <prereq identprereq="2"/>
    <curso identcurso="10">
      <elementos>
        <exercico possuiexerc="sim"/>
      </elementos>
    </curso>
    <curso identcurso="20">
      <elementos>
        <exercico possuiexerc="sim"/>
      </elementos>
    </curso>
    <curso identcurso="30">
      <elementos>
        <exercico possuiexerc="sim"/>
      </elementos>
    </curso>
  </topico>
  <topico numtop="2.2"
    desctop="Métodos Diretos"
    abreviacao="Metod. Dir."
    arquivoxml="Metodos_Diretos.html"
    palchave="Método, Direto">
    <prereq identprereq="2"/>
    <curso identcurso="10">
      <elementos>
      </elementos>
    </curso>
  </topico>

```

```

</curso>
<curso identcurso="20">
  <elementos>
    </elementos>
  </curso>
<curso identcurso="30">
  <elementos>
    </elementos>
  </curso>
<topico numtop="2.2.1"
  desctop="Método de Gauss"
  abreviacao="Metod. Gauss"
  arquivoxml="Sela_Gauss.html"
  palchave="Sistemas, Equações">
  <prereq identprereq="2.2"/>
  <curso identcurso="10">
    <elementos>
      <exemplo possuiexemp="sim"/>
      <matcomp possuiatcomp="sim"/>
    </elementos>
  </curso>
  <curso identcurso="20">
    <elementos>
      <exemplo possuiexemp="sim"/>
      <matcomp possuiatcomp="sim"/>
    </elementos>
  </curso>
  <curso identcurso="30">
    <elementos>
      <exemplo possuiexemp="sim"/>
      <matcomp possuiatcomp="sim"/>
    </elementos>
  </curso>
<topico numtop="2.2.1.1"
  desctop="Algoritmo da Triangularização"
  abreviacao="Alg. Triang."
  arquivoxml="Sela_Triang.html"
  palchave="Triangularização">
  <prereq identprereq="2.2.1"/>
  <curso identcurso="10">
    <elementos>
      </elementos>
    </curso>
  <curso identcurso="20">
    <elementos>
      </elementos>
    </curso>
  <curso identcurso="30">
    <elementos>
      </elementos>
    </curso>
  </topico>
<topico numtop="2.2.1.2"
  desctop="Algoritmo da Retrosubstituição"
  abreviacao="Alg. Retrosub."
  arquivoxml="SEla_Retro.html"
  palchave="Retrosubstituição">
  <prereq identprereq="2.2.1"/>
  <curso identcurso="10">
    <elementos>
      </elementos>
    </curso>
  <curso identcurso="20">
    <elementos>
      </elementos>
    </curso>
  <curso identcurso="30">
    <elementos>
      </elementos>
    </curso>
  </topico>
</topico>
<topico numtop="2.2.2"
  desctop="Método de Gauss com Pivotamento"
  abreviacao="Gauss Pivot."
  arquivoxml="Sela_Gaussp.html"
  palchave="Gauss, Pivotamento">

```

```

<prereq identprereq="2.2"/>
  <prereq identprereq="2.2.1"/>
<curso identcurso="10">
  <elementos>
    <exemplo possuiexemp="sim"/>
    <matcomp possuiatcomp="sim"/>
  </elementos>
</curso>
<curso identcurso="20">
  <elementos>
    <exemplo possuiexemp="sim"/>
    <matcomp possuiatcomp="sim"/>
  </elementos>
</curso>
<topico numtop="2.2.2.1"
  desctop="Condicionamento de Matrizes"
  abreviacao="Cond. Mat."
  arquivoxml="Condicioamento_Matrizes.html"
  palchave="Condicionamento, Matriz">
  <prereq identprereq="2.2.2"/>
<curso identcurso="10">
  <elementos>
  </elementos>
</curso>
<curso identcurso="20">
  <elementos>
  </elementos>
</curso>
</topico>
</topico>
<topico numtop="2.2.3"
  desctop="Método da Decomposicao LU"
  abreviacao="Decomp. LU"
  arquivoxml="Sela_LU.html"
  palchave="Decomposicao">
<prereq identprereq="2.2"/>
<curso identcurso="10">
  <elementos>
    <matcomp possuiatcomp="sim"/>
  </elementos>
</curso>
<curso identcurso="20">
  <elementos>
    <matcomp possuiatcomp="sim"/>
  </elementos>
</curso>
</topico>
</topico>
<topico numtop="2.2.4"
  desctop="Método de Cholesky"
  abreviacao="Metod. Chol."
  arquivoxml="Sela_Cholesky.html"
  palchave="Cholesky">
<prereq identprereq="2.2"/>
<prereq identprereq="2.2.3"/>
<curso identcurso="10">
  <elementos>
    <matcomp possuiatcomp="sim"/>
  </elementos>
</curso>
<curso identcurso="20">
  <elementos>
    <matcomp possuiatcomp="sim"/>
  </elementos>
</curso>
</topico>
</topico>
<topico numtop="2.3"
  desctop="Métodos Iterativos"
  abreviacao="Metod. Iterat."
  arquivoxml="Metodos_Iterativos.html"
  palchave="Iterativos">
<prereq identprereq="2"/>
<curso identcurso="10">
  <elementos>
  </elementos>
</curso>
<curso identcurso="20">

```

```

    <elementos>
  </elementos>
</curso>
<curso identcurso="30">
  <elementos>
  </elementos>
</curso>
<topico numtop="2.3.1"
  desctop="Método de Jacobi"
  abreviacao="Metod. Jac."
  arquivoxml="Sela_Jacobi.html"
  palchave="Jacobi">
  <prereq identprereq="2.3"/>
  <curso identcurso="10">
    <elementos>
      <matcomp possuimatcomp="sim"/>
    </elementos>
  </curso>
  <curso identcurso="20">
    <elementos>
      <matcomp possuimatcomp="sim"/>
    </elementos>
  </curso>
    <curso identcurso="30">
      <elementos>
        <matcomp possuimatcomp="sim"/>
      </elementos>
    </curso>
  </topico>
</topico numtop="2.3.2"
  desctop="Método de Gauss-Seidel"
  abreviacao="Metod. GS"
  arquivoxml="Sela_GaussSeidel.html"
  palchave="Gauss-Seidel">
  <prereq identprereq="2.3"/>
  <prereq identprereq="2.3.1"/>
  <curso identcurso="10">
    <elementos>
      <matcomp possuimatcomp="sim"/>
    </elementos>
  </curso>
  <curso identcurso="20">
    <elementos>
      <matcomp possuimatcomp="sim"/>
    </elementos>
  </curso>
</topico>
</topico>
</topico>
</material>

```

Referências

- [AMA 2002] AMARAL, Marília A. **Armazenamento Persistente de Dados no Ambiente AdaptWeb utilizando XML**. 2002. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [BER 96] BERCHT, Magda. **Estudos sobre Representações de Estratégias de Ensino para Tutores Inteligentes**. 1996. Trabalho Individual – Instituto de Informática, UFRGS, Porto Alegre.
- [BIL 2002] BILLSUS, Daniel; BRUNK Clifford A.; EVANS, Craig; GLADISH, Brian; PAZZANI, Michael. Adaptive Interfaces for Ubiquitous Web Access. 2002. **Communications of the ACM**, New York, v. 45, n. 5, p. 34-38.
- [BOT 2000] BOTICARIO, J. G.; GAUDIOSO, E. Towards a Personalized Web-Based Educational System. In: MEXICAN INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, MICAI, 1, 2000, Acapulco. **MICAI 2000: Advances in Artificial Intelligence: proceedings**. Berlin: Springer, 2000. p. 729-740. (Lecture Notes in Artificial Intelligence 1793).
- [BRU 2001] BRUSILOVSKY, Peter. Adaptive Hypermedia. **User Modeling and User Adapted Interaction** - UMUAI, Netherlands, v.11, n. 1-2, p. 87-110, 2001. Special Issue Ten Anniversary Issue.
- [BRU 2001a] BRUSILOVSKY, Peter. Adaptive Educational Hypermedia. In: INTERNATIONAL PEG CONFERENCE, PEG, 10. , 2001, Tampere, Finland. **Proceedings...** Disponível em: <<http://www2.sis.pitt.edu/~peterb/papers/PEG01.html>>. Acesso em: maio 2002.
- [BRU 99] BRUSILOVSKY, Peter. **Adaptive and Intelligent Technologies for Web-based Education**. 1999. Disponível em: <<http://www.contrib.andrew.cmu.edu/~plb/papers/KI-review.html>>. Acesso em: set. 2001.
- [BRU 98] BRUSILOVSKY, Peter. Adaptive Educational Systems on the World-Wide-Web: A Review of Available Technologies. In: INTERNATIONAL CONFERENCE ON INTELLIGENT TUTORING SYSTEMS, ITS, 4., 1998, San Antonio. **Intelligent Tutoring Systems: proceeding**. Berlin: Springer-Verlag, 1998.
- [BRU 98a] BRUSILOVSKY, Peter; EKLUND, John. A Study of User Model Based Link Annotation in Educational Hypermedia. **Journal of Universal Computer Science**, [S.l.], v. 4, n. 4, p. 429-448, 1998.
- [BRU 97] BRUSILOVSKY, Peter. Efficient Techniques for Adaptive Hypermedia. In: NICHOLAS, Charles K.; MAYFIELD, James. **Intelligent Hypertext: Advanced Techniques for the World Wide Web**. Berlin: Springer-Verlag, 1997. p. 12-30.
- [BRU 96] BRUSILOVSKY, Peter. Methods and techniques of adaptive hypermedia. **User Modeling and User Adapted Interaction**, [S.l.], v.6, n. 2-3, p. 87-129, 1996. Special Issue on adaptive hypertext and hypermedia.
- [CAR 2001] CARDOSO, Rodrigo Ferrugem; LIMA, José Valdeni de. AvalWeb -

- Sistema Interativo para Gerência de Questões e Aplicação de Avaliações na Web. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, SBIE, 12., 2001, Vitória. **Anais...** Vitória: UFES, 2001.
- [CAS 2001] CASTAGNETTO, Jesus et al. **Professional PHP: Programando**. São Paulo: Makron Books, 2001.
- [CHE 2002] CHEN, Sherry Y.; MACREDIE, Robert D. Cognitive Styles and Hypermedia Navigation: Development of a Learning Model. **Jornal of the American Society for Information Science And Technology**, New York, v. 53, n. 1, p. 3 –15, 2002.
- [COS 9?] COSTA, Rosa Maria E. M; WERNECK, Vera Maria B. **Sistemas Tutoriais: aplicação das Tecnologias de Hipermídia e de Inteligência Artificial em Educação**. (199?). COPPE – UFRJ, Rio de Janeiro.
- [COS 96] COSTA, Rosa Maria E. M.; WERNECK, Vera Maria B. **Tutores Inteligentes**. 1996. Relatório Técnico ES – 392/96, COPPE/UFRJ.
- [DEB 2000] DE BRA, Paul. Pros and Cons of Adaptive Hypermedia in Web-based Education. **Journal on Cyber Psychology and Behavior**, [S.l.], v. 3, n. 1, p. 71-77, 2000. Disponível em: <<http://wwwis.tue.nl/~debra/cyber.html>> Acesso em ago. 2001.
- [DEB 99] DE BRA, Paul. Design Issues in Adaptive Web-Site Development. In: WORKSHOP ON ADAPTIVE SYSTEMS AND USER MODELING ON THE WWW, 1999, Canada. **Proceedings...** Disponível em: <<http://wwwis.win.tue.nl/~debra/asum99/debra/debra.html>>. Acesso em: ago. 2001.
- [DEB 99a] DE BRA, P.; WU, H.; HOUBEN, G. J. P. User Modeling in Adaptive Hypermedia Applications. In: INTERDISCIPLINAIRE CONFERENTIE INFORMATIEWETENSCHAP, 1999, Amsterdam. **Proceedings...** Disponível em: <<http://wwwis.win.tue.nl/~debra/public.html>>. Acesso em: set. 2001.
- [DEM 2001] DEMATRINI, Giovanni. **Autentificação de Aluno e Geração e Análise de log de Acessos em Curso de Ensino a Distância**. 2001. Trabalho de Diplomação (Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [DOM 98] DOCUMENT OBJECT MODEL (DOM). **Level 1 Specification Version 1.0**. October 1, 1998. Disponível em: <<http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/>>. Acesso em: jun. 2002.
- [FRA 91] FRAINER, Antônio S. **Interfaces Inteligentes**. 1991. Trabalho Individual (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [FRE 2003] FREITAS, Veronice. **Autoria Adaptativa de Hipermídia Educacional**. 2003. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [GAS 2002] GASPARINI, Isabela; PIMENTA, Marcelo S.; PALAZZO, M. de Oliveira, José; LIMA, José Valdeni de; FREITAS, Veronice de; MARÇAL, Viviane P.; AMARAL, Marília A.; PROENÇA Jr., Mario Lemes; BRUNETTO, Maria Angélica C.; RIBEIRO, Cora H. F. Pinto. AdaptWeb: an Adaptive Web-based Courseware. In:

- INTERNATIONAL CONFERENCE OF INFORMATION AND COMMUNICATION TECHNOLOGIES IN EDUCATION, 2002. **Proceedings...** [S.l.: s.n.], 2002.v.1, p. 131-134.
- [GAS 2002a] GASPARINI, Isabela; PIMENTA, Marcelo S. Concepção de Interfaces WWW Adaptativas para EAD. **Cadernos de Informática**, Porto Alegre, v.2, n.1, p.71-76, mar. 2002. Artigo apresentado nos Seminários sobre Tecnologias de Informática para Ensino a Distância, 2001.
- [GAS 2001] GASPARINI, Isabela. **Concepção de Interfaces Adaptativas para Educação a Distância: uma revisão bibliográfica**. 2001. Trabalho Individual (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [HOO 96] HÖÖK, Kristina et al. A Glass Box Approach to Adaptive Hypermedia. **Jornal User Modelling and User-Adaptive Interaction**, Netherlands, v.6, n. 2/3, p. 157-184, 1996.
- [KOB 2001] KOBSA, Alfred; KOENEMANN, Jürgen; POHL, Wolfgang. Personalised hypermedia presentation techniques for improving online customer relationships. **The Knowledge Engineering Review**, Cambridge, v. 16, n. 2, p. 111–155, 2001.
- [KOC 2001] KOCH, Nora P. **Software Engineering for Adaptive Hypermedia Systems – Reference Model, Modeling Techniques and Development Process**. 2001. PhD. Thesis. Ludwig-Maximilians-Universität München. Disponível em: <<http://www.pst.informatik.uni-muenchen.de/~kochn/>>. Acesso em: abr. 2002.
- [KUL 2000] KULES, Bill. **User Modeling for Adaptive and Adaptable Software Systems**. Maryland: Department of Computer Science, University of Maryland, College Park, 2000. Disponível em: <<http://www.otal.umd.edu/UUGuide/wmk/>>. Acesso em: dez. 2001.
- [LIM 2002] LIMA, P. S. R.; PIMENTA, M. S. Personalização de Interfaces Web para Sites Institucionais com Base em Perfis de Usuários. In: CONGRESSO INTERNACIONAL DE ERGONOMIA E USABILIDADE, DESIGN DE INTERFACES E INTERAÇÃO HUMANO-COMPUTADOR, 1., 2002, Rio de Janeiro. **Anais...** Rio de Janeiro: PUC-Rio, 2002.
- [LIM 2002a] LIMA, P. S. R. **Personalização de Interfaces Web para Sites Institucionais com Base em Perfis de Usuários**. 2002. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [LYN 2002] LYNCH, Patrick J.; HORTON, Sarah. **Web Style Guide**. 2nd ed. 2002. Disponível em: <<http://www.webstyleguide.com/index.html>>. Acesso em: jul. de 2002.
- [MAR 2003] MARÇAL, Viviane. **Adaptação de Conteúdo baseada no modelo do aluno em um Ambiente de Ensino Adaptativo**. 2003. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [MAY 2002] MAYBURY, Mark; BRUSILOVSKY, Peter. From Adaptive Hypermedia to the Adaptive Web. **Communications of the ACM**, New York, v. 45, n. 5, p. 31-33, 2002.

- [MYK 92] MYKA, A.; SARRE, F.; GÜNTZER, F. Rule-Based Machine Learning of Hypertext Links. In: INTERNATIONAL WORKSHOP ON DATABASE SYSTEM INTELLECTUALIZATION, 1992. **Proceedings...** [S.l.:s.n], 1992. Disponível em:<<http://www-db.informatik.uni-tuebingen.de/mitarbeiter/myka.shtml>>. Acesso em: set. 2002.
- [NIE 2002] NIELSEN, Jakob; TAHIR, Marie. **HomePage: 50 websites desconstruídos**. Rio de Janeiro: Campus, 2002.
- [NIE 2000] NIELSEN, Jakob. **Designing Web Usability: The Practice of Simplicity**. Indianapolis: New Riders Publishing, 2000.
- [OLI 2002] OLIVEIRA, José M. P.; FERNANDES, Clovis T. Arquitetura de Adaptação em Sistemas Hipermídia Adaptativos Educacionais. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, SBIE, 13., 2002, São Leopoldo. **Anais...** São Leopoldo: UNISINOS, 2002.
- [PAL 2002] PALAZZO, Luiz Antônio M. Sistemas de Hipermídia Adaptativa. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 22., 2002, Florianópolis. **Convergências Tecnológicas: redesenhando as fronteiras da ciência e da educação**. Florianópolis: SBC, 2002. p. 1-38.
- [PES 96] PESIN, Leonid; BRUSILOVSKY, Peter. **ISIS-Tutor: An Intelligent Learning Environment for CDS/ISIS Users**. 1996. Disponível em: <http://www.cs.joensuu.fi/~mtuki/www_clce.270296/Brusilov.html>. Acesso em: set. 2001.
- [PIM 2001] PIMENTEL, Maria da Graça C.; TEIXEIRA, Cesar Augusto C. A Prática XML em aplicações Web. In: SIMPÓSIO BRASILEIRO DE SISTEMAS MULTIMÍDIA E HIPERMÍDIA, SBMIDIA, 7., 2001, Florianópolis. **Anais...** Florianópolis: UFSC, 2001. Disponível em: <<http://coweb.icmc.sc.usp.br/incaserve>>. Acesso em: set 2002.
- [ROU 99] ROUSSEAU, Franck; GARCIA-MACÍAS, J. Antonio, LIMA, José Valdeni de; DUDA, Andrzej. User Adaptable Multimedia Presentations for the WWW. In: INTERNATIONAL. WORLD WIDE WEB CONFERENCE, 8., 1999, Toronto, Canadá. **Proceedings...** Disponível em: <<http://drakkar.imag.fr/awww>>. Acesso em: abr. 2002.
- [ROY 2001] ROY, Jaideep; RAMANUJAN, Anupana. XML Schema Language Taking XML to the Next Level, **Proceeding of the IEEE**, New York, v. 1, p. 37-40, Apr. 2001.
- [SAN 2000] SANTOS, Neide. **Interfaces de Ambientes Educacionais: Diretrizes de Projeto**. Rio de Janeiro: Instituto de Matemática e Estatística, Universidade do Rio de Janeiro (UERJ), 2000. Disponível em: <<http://www.ime.uerj.br/professores/neidenew/Interfaces.htm>>. Acesso em: jan. 2002.
- [SAN 98] SANTIBAÑEZ, Miguel. R Flores; FERNANDES Clovis Torres. Hacia Um Ambiente De Aprendizagem Hipermídia Adaptativo no WWW. In: CONGRESSO IBEROAMERICANO DE INFORMÁTICA NA EDUCAÇÃO, 4., 1998. Brasília. **Anais...** Disponível em: <<http://www.niee.ufrgs.br/ribie98/TRABALHOS/204.PDF>>. Acesso em: out. 2002.

- [SAN 96] SANTOS, Neide; CRESPO, Sérgio da Silva Pinto; ROCHA, Ana R. C. **Navegação em Documentos Hipermissão:** Estado da Arte. Rio de Janeiro: COPPE/Sistemas, 1996. (Relatório Técnico ES-373/96). Disponível em: <<http://www.ime.uerj.br/professores/neidenew/Reltec.htm>>. Acesso em: jan. 2002.
- [SEL 2001] SELIGMAN, Len. XML's Impact on Databases and Data Sharing. **IEEE Computer**, [S. l.], v.34, n. 6, p. 59-67, June 2001.
- [SOU 2002] SOUTO, Maria Aparecida Martins; VERDIN, Regina; SOUZA, Karine Beschoren de; ZANELLA, Renata; NICOLAO, Mariano; MADEIRA, Milton J. P.; VICARI, Rosa Maria; OLIVEIRA, Jose Palazzo Moreira de. Web-adaptive training system based on cognitive student style. In: IFIP WORLD COMPUTER CONGRESS, 17., 2002, Montreal. **Tele-learning**. Boston: Kluwer Academic, 2002. p.281-288.
- [SOU 2002a] SOUTO, Maria Aparecida M.; VERDIN, Regina; WAINER, Ricardo; MADEIRA, Milton; VICARI, Rosa M.; OLIVEIRA, José Palazzo de. 2002. Um Estudo Empírico dos Comportamentos de Navegação por Estilo Cognitivo de Aprendizagem em um ambiente de treinamento na Web. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, SBIE, 13., 2002, São Leopoldo. **Anais...** São Leopoldo: UNISINOS, 2002.
- [SOU 2000] SOUTO, Maria Aparecida Martins, OLIVEIRA, José Palazzo Moreira de; VICCARI, Rosa Maria; DIEHL, Eduardo; MADEIRA, Milton; VERDIN, Regina; WAINER, Ricardo; SIMONEENGLER. Modelo de ensino adaptativo na internet baseado em estilos cognitivos de aprendizagem. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, SBIE, 2000, Maceió, Alagoas. **Anais...** Disponível em: <http://www.inf.ufrgs.br/~palazzo/docs/Artigos/00_XI_SBIE.pdf>. Acesso em: set. 2002.
- [SPE 99] SPECHT, Marcus; KOBZA, Alfred. Interaction of Domain Expertise and Interface Design in Adaptive Educational Hypermedia. In: WORKSHOP ON ADAPTIVE SYSTEMS AND USER MODELING ON THE WWW. 2., 1999. **Proceedings...** Disponível em: <http://www.contrib.andrew.cmu.edu/~plb/WWWUM99_workshop/specht/specht.html>. Acesso em: nov. 2001.
- [SPE 97] SPECHT, Marcus; WEBER, Gerhard. User Modeling and Adaptive Navigation Support in WWW-Based Tutoring Systems. In: INTERNATIONAL CONFERENCE ON USER MODELING, UM, 6., 1997, Sardinia. **User Modeling: proceedings**. Sardinia: Springer-Verlag, 1997.
- [VIC 2000] VICARI, Rosa; GIRAFFA, M. L. **Introdução aos Sistemas Tutores Inteligentes**. Material disponibilizado nas aulas do Programa de Pós-Graduação em Ciência da Computação, UFRGS, 2000.
- [VIC 90] VICCARI, Rosa; MOUSSALE, Neila. Tutores Inteligentes para o Ensino da Linguagem PROLOG. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, SBIE, 1., 1990, Rio de Janeiro. **Anais...** Rio de Janeiro: SBC, 1990. p. 203-220.
- [WU 2001] WU, Hongjing. A Reference Architecture for Adaptive Hypermedia

Systems. In: WORKSHOP ON ADAPTIVE HYPERTEXT AND HYPERMEDIA, 3., 2001. **Proceedings...** Disponível em: <<http://wwwis.win.tue.nl/ah2001/proceedings.html>>. Acesso em: maio 2002.

[WU 98] WU, H.; HOUBEN, G.J.; DE BRA, P. AHAM: A Reference Model to Support Adaptive Hypermedia Authoring, In: ZESDE INTERDISCIPLINAIRE CONFERENTIE INFORMATIEWETENSCHAP, 1998, Antwerp. **Proceedings...** Disponível em: <<http://wwwis.win.tue.nl/~debra/infwet98/paper.pdf>>. Acesso em: abr. 2002.

[W3C 2000] EXTENSIBLE Markup Language - XML. W3C Recommendation. 06 October 2000. Disponível em: <<http://www.w3.org/TR/REC-xml>>. Acesso em: jul. 2001.