

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**SISTEMA HIPERMÍDIA ADAPTATIVO: A SUA UTILIZAÇÃO PARA
O ENSINO DA MATEMÁTICA**

Douglas Kaminski

Trabalho de conclusão de curso
submetido à Universidade Federal de
Santa Catarina como parte dos
requisitos para obtenção do grau de
Bacharel em Sistemas de Informação.

Florianópolis - SC

2006/1

Douglas Kaminski

**SISTEMA HIPERMÍDIA ADAPTATIVO: A SUA UTILIZAÇÃO PARA O ENSINO
DA MATEMÁTICA**

Trabalho de conclusão de curso submetido à Universidade Federal de Santa Catarina como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

Orientadora: Profa. Marília A. Amaral, Msc.

Universidade Federal de Santa Catarina

marilia@inf.ufsc.br

Banca Examinadora

Prof. Fernando Ostuni Gauthier, Dr.

Universidade Federal de Santa Catarina

gauthier@inf.ufsc.br

Profa. Vania Ribas Ulbricht, Dra.

Universidade Federal de Santa Catarina

ulbricht@floripa.com.br

AGRADECIMENTOS

Agradeço aos meus pais pelo investimento inicial e pela convivência verdadeira que sempre tivemos, que continuam a conversar e guiar-me ao melhor caminho.

Aos meus irmãos pelas dicas que me deram na convivência em relação ao dia a dia de uma Universidade.

À minha namorada Giséle que me incentivou em algumas vitórias importantes da minha vida durante o tempo em que estamos juntos.

Aos Professores do INE e em especial à Marília que durante as aulas que ela ministrou neste Departamento como Professora Substituta, aceitou ser a minha orientadora deste projeto numa área de pesquisa sugerida por ela e que me propiciou novos conhecimentos.

Por último, aos meus colegas Servidores Técnico-Administrativos desta Universidade da qual faço parte como aluno desde 2001 e como funcionário desde 2004. Um pessoal que se mostrou muito acolhedor e, acima de tudo, grandes companheiros de trabalho.

RESUMO

KAMINSKI, Douglas. **Sistema Hipermídia Adaptativo: a sua utilização para o ensino da matemática**. 2006. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação). Curso de Sistemas de Informação, Universidade Federal de Santa Catarina, Florianópolis.

Os usuários que navegam pela *web* estão cada vez mais ansiosos por tecnologias que permitam uma personalização das suas aplicações, diminuindo a sobrecarga cognitiva e a desorientação durante o processo de navegação. Nos ambientes de ensino a distância a procura já é grande por aplicações que reconheçam estes anseios dos usuários para não perder o interesse do seu maior cliente que é o aluno. Este trabalho está na busca pela realização destas idéias através do uso das técnicas de hipermídia adaptativa. O Sistema Hipermídia Adaptativo e os modelos de referência Dexter, AHAM e Munich são detalhados para um prévio conhecimento da estrutura, sendo proposta a implementação de um protótipo que utiliza a modelagem UWE (*UML-based Web Engineering*), que especifica a construção de seis modelos: diagramas de casos de uso, modelo conceitual, modelo de usuário, modelo de navegação, modelo de apresentação e modelo de adaptação. O estudo de caso utilizado é a disciplina de matemática e a estrutura de suas aulas está armazenada num arquivo XML (*Extensible Markup Language*) que é validado por uma DTD (*Document Type Definition*). Para passar ao aluno a visualização desta estrutura, é feita uma leitura do arquivo XML através da API DOM embutida no código PHP.

Palavras-Chave: EAD, Adaptabilidade, XML, DOM, SHAE.

ABSTRACT

KAMINSKI, Douglas. **Adaptive Hypermedia System: its use for the education of the mathematics**. 2006. Final Dissertation. Information Systems Graduation Course, Universidade Federal de Santa Catarina, Florianópolis.

The users who sail for web are each time more anxious for technologies that they allow a personalization of its applications, diminishing the cognitive overload and the disorientation during the navigation process. In education environments in the distance the search already is great for applications that recognize these yearnings of the users not to lose the interest of its bigger customer who is the pupil. This work is in the search for the accomplishment of these ideas through the use of the techniques of adaptive hypermedia. The System Hypermedia Adaptative and the models of reference Dexter, AHAM and Munich are detailed for a previous knowledge of the structure, having been proposed the implementation of an archetype that uses the modeling UWE (UML-based Web Engineering), that it specifies the construction of six models: diagrams of use cases, conceptual model, model of user, model of navigation, model of presentation and model of adaptation. The study of used case it is disciplines it of mathematics and the structure of its lessons is stored in an archive XML (Extensible Markup Language) that it is validated by a DTD (Document type definition). To pass to the pupil the visualization of this structure, a reading of archive XML through API DOM is made inlaid in code PHP.

Palavras-Chave: Distance Learning Environment, Adaptive, XML, DOM, SHAE.

SUMÁRIO

LISTA DE FIGURAS	7
LISTA DE TABELAS	8
LISTA DE ABREVIATURAS E SIGLAS	9
1 INTRODUÇÃO	10
1.1 DESCRIÇÃO DO PROBLEMA.....	11
1.2 OBJETIVOS.....	11
1.3 JUSTIFICATIVA.....	12
1.4 ESTRUTURA DO DOCUMENTO.....	13
2 FUNDAMENTOS DE UM SISTEMA HIPERMÍDIA ADAPTATIVO	14
2.1 FORMAS DE ADAPTAÇÃO.....	14
2.1.1 APRESENTAÇÃO ADAPTATIVA.....	15
2.1.2 NAVEGAÇÃO ADAPTATIVA.....	16
2.1.3 CARACTERÍSTICAS VOLTADAS PARA A ADAPTAÇÃO.....	17
3. MODELOS DE REFERÊNCIA PARA SHA.....	18
3.1 DEXTER.....	18
3.2 AHAM.....	20
3.3 MUNICH.....	22
4. XML EM APLICAÇÕES HIPERMÍDIA ADAPTATIVAS	25
4.1 A LINGUAGEM XML.....	25
4.2 ESTRUTURA E SINTAXE DA XML.....	26
4.3 XML E SUAS FOLHAS DE ESTILO.....	29
4.4 EXEMPLOS DE SHAS QUE UTILIZAM XML.....	30
5. DESENVOLVIMENTO DA APLICAÇÃO PROPOSTA	33
5.1 OS MODELOS ADAPTATIVOS DA UWE.....	33
5.1.1 MODELO DE CASO DE USO.....	35
5.1.2 MODELO CONCEITUAL.....	37
5.1.3 MODELO DE USUÁRIO.....	38
6. PROTÓTIPO DE UM SISTEMA HIPERMÍDIA ADAPTATIVO EDUCACIONAL	33
6.1 METODOLOGIA APLICADA.....	48
6.2 TÉCNICAS DE ADAPTAÇÃO.....	52
6.3 MODELO DE USUÁRIO.....	55
7. CONCLUSÃO	57
7.1 TRABALHOS FUTUROS.....	58
ANEXO 1 – INTERFACES DO PSHAE	59
ANEXO 2 – ARQUIVO XML – ESTRUTURA DOS CONCEITOS	60
REFERÊNCIAS	62
APÊNDICE A - CÓDIGO FONTE	625
PÊNDICE B - ARTIGO	85

LISTA DE FIGURAS

FIGURA 1: ARQUITETURA DO MODELO DEXTER. FONTE: HALASZ, 1990	18
FIGURA 2: ARQUITETURA DO MODELO AHAM. FONTE: DE BRA, 1998	20
FIGURA 3: ARQUITETURA DO MODELO MUNICH. FONTE: KOCH, 2002	23
FIGURA 4: EXEMPLO DE UM ARQUIVO XML.....	26
FIGURA 5: PROCESSAMENTO DE UM ARQUIVO XSLT.....	30
FIGURA 6: MODELO DE DOMÍNIO. FONTE: DE BRA, 2002	31
FIGURA 7: ARQUIVO XML GERADO PELA ARGOUWE. FONTE: IYENGAR, 1999.....	32
FIGURA 8: MODELOS NA FASE DE PLANEJAMENTO DE UM SHA. FONTE: KOCH, 2002 ...	34
FIGURA 9: MODELO DE CASO DE USO	36
FIGURA 10: DIAGRAMAS DE ATIVIDADES	36
FIGURA 11: MODELO CONCEITUAL	38
FIGURA 12: MODELO DE USUÁRIO	40
FIGURA 13: MODELO DE HIPERESPAÇO DE NAVEGAÇÃO	42
FIGURA 14: MODELO DE ESTRUTURA DE NAVEGAÇÃO	43
FIGURA 15: CÓDIGO DTD	50
FIGURA 16: CÓDIGO DE INTERAÇÃO PHP E API DOM.....	51
FIGURA 17: TELA DO PSHAE – NÍVEL INTERMEDIÁRIO.	52
FIGURA 18: ESTRUTURA DO PHSAE	54
FIGURA 19: TELA DO PSHAE: NÍVEL INICIANTE	55

LISTA DE TABELAS

TABELA 1 - SEQÜÊNCIA DAS AULAS DA DISCIPLINA DE MATEMÁTICA NO PSHAE	53
TABELA 2: ATRIBUTOS DA TABELA DO MODELO DE USUÁRIO	55

LISTA DE ABREVIATURAS E SIGLAS

- **CSS** - *Cascading Style Sheets*: é um arquivo que possui referências de formatação, oferecendo um maior controle sobre a apresentação e layout de elementos de HTML.
- **DOM** - *Document Object Model*: especificação desenvolvida pelo W3C que permite a manipulação de documentos XML através de um conjunto de propriedades e métodos especificados como se fossem para um objeto de um programa completo.
- **DTD** - *Definition Type Document*: são regras que definem quais as *tags* que podem ser usadas em um documento XML e quais seus valores válidos.
- **SAX** - *Simple API for XML*: provê uma interface de manipulação de arquivos XML.
- **SHA** – Sistema Hipermedia Adaptativo: área da ciência da computação que se ocupa do estudo e desenvolvimento de sistemas capazes de promover a adaptação de hiperdocumentos e hipermedia em geral.
- **XML** - *Extensible Markup Language*: linguagem que utiliza *tags* para organização dos dados de forma hierárquica.
- **XML Schema** – similar a DTD com a adição de algumas funcionalidades, como o suporte a tipo de dados.
- **XSL** - *The Extensible Stylesheet Language Family*: é um grupo de linguagens responsáveis por transformar documentos escritos em XML.
- **XSL-FO** - *XSL Formatting Objects*: ideal para descrever um layout preciso de livros e imagens, gerando documentos bem formatados para posterior impressão.
- **UML** - *Unified Modeling Language*: é uma linguagem para especificação, documentação, visualização e desenvolvimento de sistemas orientados a objetos.
- **URL** - *Universal Resource Locator*: endereço em forma de número ou nome que identifica as páginas na web.
- **UWE** - *UML-based Web Engineering*: uma abordagem de engenharia de software para o desenvolvimento de SHAs.
- **W3C** - *World Wide Web Consortium*: órgão responsável por manter a especificação XML.

1 INTRODUÇÃO

Os ambientes de ensino virtuais se propagam com o aumento do número de acessos a rede mundial de computadores. Como atualmente, qualquer dispositivo conectado a Internet possibilita acesso às informações disponíveis em um *site* na *web*, tornou-se possível o surgimento de ambientes de ensino virtuais. Existem várias pesquisas sobre este assunto que buscam uma tecnologia mais apropriada para este fim.

O Ensino a Distância ou EAD é um meio de comunicação que utiliza uma ação conjunta de diversos recursos didáticos que propiciam a aprendizagem mais independente e autônoma dos alunos. Com os recursos de hipermídia atuais, o professor pode manter um contato constante com os seus alunos através de fóruns, *chat* e vídeo-conferência, diminuindo a fronteira de comunicação que existe numa sala de quarenta alunos, por exemplo.

No entanto, há ainda alguns empecilhos para uma maior disseminação desta nova forma de ensinar. O acesso a Internet no Brasil ainda é demasiadamente caro e a aquisição de computadores não está ao alcance de todos. Além desta exclusão digital, alguns cursos existentes na *web* não atendem as expectativas dos alunos, sendo que a tecnologia aplicada no desenvolvimento é precária ou faltam profissionais especializados.

Este é o grande desafio: combater a “maldição” do formato taylorista e fordista de transmissão de informações, que não assegura a construção do conhecimento e, ao contrário, promete demagogicamente uma capacitação que o formato de tempo disponível e a qualificação dos envolvidos não atende [1].

Um Sistema Hipermídia Adaptativo (SHA) é uma tecnologia que vem para facilitar a aplicação de um curso on-line de uma forma mais dinâmica. Assim, os usuários podem seguir no *site* direções totalmente diferentes em relação ao outro e nem por isso, prejudicar o aprendizado. Entretanto, a apresentação do conteúdo bem como a quantidade de *links* acompanha o modelo do usuário dependendo da evolução do seu aprendizado, sendo o sistema responsável por aplicar e modificar o conteúdo da disciplina de acordo com o perfil do aluno.

Procura-se com este trabalho, propor um ambiente de ensino usando os conceitos de hipermídia adaptativa para os alunos da oitava série do ensino

fundamental. Espera-se, também, difundir esta tecnologia como uma ferramenta a mais na avaliação da disciplina de matemática trazida pela solução proposta, buscando uma melhor avaliação dos alunos, seguindo as suas características e personalizando o ambiente de acordo com estes usuários.

1.1 DESCRIÇÃO DO PROBLEMA

Como a cada ano aumenta o número de alunos em sala de aula, fica cada vez mais difícil para o professor identificar as dificuldades de aprendizado. Portanto, é necessário que surjam ferramentas de auxílio ao docente, para que ele possa avaliar os seus alunos de uma forma eficiente e propor novas metodologias de ensino para evitar os déficits de aprendizado.

A disciplina de matemática continua sendo a maior causa de reprovações e deficiências no ensino fundamental. O Sistema Nacional de Avaliação da Educação Básica em Matemática apurou que:

A maioria dos alunos, 51,71%, encontra-se no estágio crítico, e apenas 2,65% se situam no estágio adequado ... 97,21% dos alunos de 8ª série não conseguiram atingir o nível adequado, estando, portanto, aquém do nível exigido para a 8ª série. Esses alunos não interpretam e nem resolvem problemas de forma competente e, portanto, não fazem o uso correto da linguagem matemática [2].

Um SHA se enquadra entre as ferramentas que possibilitará um intercâmbio maior de informações via *web*, vindo a ser utilizado como uma avaliação adicional na disciplina. Algumas técnicas de SHA foram implementadas em alguns ambientes já desenvolvidos, seguindo apenas as suas características individuais, mas já foram desenvolvidos modelos para a implementação de qualquer sistema e o seu uso tende cada vez mais a se popularizar.

1.2 OBJETIVOS

Pretende-se utilizar os conceitos gerais da hipermídia adaptativa para desenvolver um EAD, procurando facilitar o aprendizado dos alunos da disciplina de matemática, bem como proporcionar uma avaliação adicional pelo professor da disciplina.

Para alcançar os resultados esperados com este trabalho de conclusão de curso, foram definidos os seguintes objetivos específicos:

- comparar as aplicações existentes sobre hipermídia adaptativa.
- abordar as arquiteturas de hipermídia adaptativa.
- propor um aplicativo que facilite o aprendizado dos alunos que cursam a disciplina de matemática no ensino fundamental.
- modelar o aplicativo proposto utilizando UWE.
- utilizar XML como forma de armazenamento deste conteúdo.

1.3 JUSTIFICATIVA

Os ambientes de ensino na *web* vêm crescendo a cada ano e as pesquisas nesta área buscam atingir os objetivos de professores e alunos. Como a tendência é o aumento crescente do acesso à Internet, principalmente na educação, fica visível a importância de se complementar cada vez mais os estudos via um EAD. Além disso, não é preciso espaço físico como uma sala de aula normal, sendo necessário apenas que cada aluno tenha acesso a qualquer dispositivo conectado a rede mundial de computadores.

Por meio de um ambiente hipermídia adaptativo disponível na *web*, o professor terá como acompanhar individualmente o desempenho e as dificuldades de seus alunos, o que na sala de aula só seria possível através de provas ou testes tradicionais. Através de um SHA, o docente não precisaria adequar o conteúdo de forma manual levando em conta a dificuldade de seus estudantes.

Uma das mais importantes características do usuário para os sistemas hipermídia adaptativos é o conhecimento deste sobre o assunto representado. O conhecimento é uma característica variável e particular ao usuário, pois os sistemas que a utilizam devem reconhecer as modificações no estado do conhecimento do usuário e devem, de acordo com estas informações, atualizar o modelo do usuário [3].

As várias formas de reação de uma pessoa ao sistema dificultam o processo de adaptação. Além disso, a navegação em um ambiente como este, deve facilitar a evolução do aluno no decorrer dos tópicos da disciplina e de maneira alguma prejudicá-lo. É preciso, sobretudo, manter o interesse do aluno não apenas através

de cobranças, mas também por atrativos que possibilitem uma contribuição mútua para a manutenção desta ferramenta no decorrer da disciplina.

A adaptação procura solucionar os seguintes problemas: quebra de fluxo conceitual; desorientação; sobrecarga cognitiva e quebra de fluxo narrativo. A solução destas questões podem auxiliar no processo de ensino e aprendizagem, na utilização de SHA voltados à Educação [4].

1.4 ESTRUTURA DO DOCUMENTO

Este trabalho relaciona as teorias sobre SHA e uma aplicação proposta para o ensino da matemática na oitava série do ensino fundamental, sendo dividido da seguinte forma: o Capítulo 2 aborda os conceitos, arquiteturas e aplicações desenvolvidas com Sistemas Hiperídia Adaptativos. No Capítulo 3 são analisados alguns Modelos de Referência para a hiperídia adaptativa. O capítulo 4 descreve a importância do uso XML em aplicações hiperídia adaptativas. O Capítulo 5 descreve uma visão geral do projeto, seus requisitos, metodologias e modelagem da aplicação proposta. No Capítulo 6 é descrita a arquitetura do protótipo desenvolvido seguindo a modelagem UWE. Por fim, as considerações finais e as propostas para trabalhos futuros são apresentados no capítulo 7.

2 FUNDAMENTOS DE UM SISTEMA HIPERMÍDIA ADAPTATIVO

Os Sistemas Hiperfídia Adaptativos (SHA) vêm com o intuito de facilitar a navegação e apresentação de conteúdo na Internet, passando para o usuário final uma interface mais personalizada aos seus interesses, tornando o ambiente em uso mais interessante.

O SHA busca resolver a desorientação e os problemas de sobrecarga cognitiva pela adoção de uma aproximação centrada no usuário. O usuário é observado pelo sistema através do modelo do usuário e o sistema se adapta a ele, mostrando um conteúdo e uma apresentação adequada ao mesmo [5].

Com isso, o sistema consegue se moldar em relação aos objetivos e ao conhecimento dos usuários, não apresentando uma estrutura estática linear, facilitando a navegação pelo *site* e mesmo assim mantendo uma estrutura de conteúdo com início, meio e fim.

As teorias sobre SHA, suas arquiteturas e exemplos de aplicações são apresentadas neste capítulo.

2.1 FORMAS DE ADAPTAÇÃO

Antes de implementar um SHA deve ser levado em conta o que pode ser adaptado com a hiperfídia adaptativa e como proceder ao uso de técnicas e métodos já desenvolvidos em outros sistemas. Procura-se com as adaptações não dificultar o processo de aprendizagem do aluno, apresentando um ambiente que seja um guia para os seus objetivos. Como um hiperdocumento é formado por *links* que apontam para nodos, que são partes pertencentes a um *site*, como uma página em *html* ou um simples documento, este deve ser modelado seguindo as técnicas e métodos que serão descritos a seguir a fim de que se tenha um SHA.

A apresentação adaptativa e o suporte a navegação adaptativa são as duas classes utilizadas para a adaptação na *web*. A primeira atua em relação ao conteúdo da página, personalizando a mesma em relação ao modelo do usuário, modificando, por exemplo, as informações, sendo mais detalhadas ou não, levando em conta o conhecimento do usuário. Já a segunda classe de adaptação, modificará os *links*

dando suporte ao usuário na navegação do ambiente em si, orientando-o na melhor forma de acesso às informações.

2.1.1 APRESENTAÇÃO ADAPTATIVA

A apresentação do conteúdo de um *site* pode ser feita através da mudança do idioma ou alteração das fontes do texto, suas cores, novas imagens, fragmentos de texto diferenciados, inclusão ou não de arquivos multimídia. São relacionados os seguintes métodos que implementam a apresentação adaptativa [6]:

- Explicação Adicional (EA): oculta certas partes do texto para o usuário que ainda não tem interesse ou o conhecimento adequado;
- Explicação Requerida (ER): ordena o conteúdo seguindo pré-requisitos;
- Explicação Comparativa (EC): utiliza a similaridade para a comparação entre os tópicos da disciplina (conceitos);
- Explicação Variante (EV): algumas variantes do conteúdo de uma página são apresentados aos usuários de acordo com o seu modelo;
- Classificação de Fragmentos (CF): ordena fragmentos de informação do conceito passando ao usuário as partes mais relevantes para os mesmos.

Algumas técnicas aplicadas em SHA definem de forma mais clara a adaptação existente numa página hipermídia. A *Strechtext* condensa o conteúdo em palavras-chaves e o expande quando necessário, passando uma sensação de resumo do conteúdo, facilitando até mesmo a localização do aluno no ambiente de ensino. O Texto Condicional (TC), por exemplo, divide o texto em partes menores que são mostradas seguindo algumas condições relacionadas ao nível de conhecimento do usuário. Assim, é possível ocultar as informações que não são relevantes para o usuário naquele momento. As Páginas Variantes (PV) mantém um certo número de páginas para cada conceito, discriminando as mesmas para os usuários do nível iniciante, intermediário ou avançado. Por último, o *frame-based approach* que permite a exibição de toda a informação em um *frame* que pode ser

mostrado, ocultado, apresentado alternativamente ou em ordem, seguindo regras para decidir que *frames* serão exibidos.

2.1.2 NAVEGAÇÃO ADAPTATIVA

A navegação num *site* no *hiperespaço* pode às vezes tornar-se uma tarefa nada fácil para o usuário, pois pode sobrecarregá-lo com um conteúdo extenso sem uma boa conectividade dos *links*, dificultando a leitura. Portanto, para minimizar estes problemas, foram desenvolvidos os seguintes métodos de suporte à navegação adaptativa [5]:

1. *Local Guidance* ou Condução Local;
2. *Local Orientation* ou Orientação Local;
3. *Global Guidance* ou Condução Global;
4. *Global Orientation* ou Orientação Global;
5. *Personalised views* ou Visões Personalizadas.

A Condução Local (CL) tem como objetivo direcionar para o usuário qual o melhor *link* da página corrente que ele deve acessar. A Condução Global (CG), ao contrário da CL, apresenta ao usuário que já possui um objetivo a alcançar com o sistema o menor caminho para se guiar em busca deste objetivo. A Orientação Local (OL) insere informações adicionais sobre o link ou oculta certos trechos de conteúdo para evitar que o usuário fique perdido diante de tanta informação. A Orientação Global (OG) tem como objetivo facilitar o conhecimento da estrutura do *hiperespaço* e a localização do usuário no mesmo. E, por último, as Visões Personalizadas mantém as interfaces personalizadas com o passar do tempo através de agentes inteligentes que teriam a responsabilidade de encontrar os novos *links* para o usuário.

Dentre as técnicas que coexistem com a navegação adaptativa, podem ser destacadas as seguintes [6]:

- Condução Direta: existe apenas um link para acesso naquele nodo determinado pelo próprio sistema;

- Anotação Adaptativa: são realizadas diferentes modificações em um *link* com o intuito de aumentar suas informações, informando ao usuário o que virá nos próximos nodos;
- Ocultação: consiste na ocultação dos *links* que levam para os nodos menos relevantes naquele momento. É a técnica mais comumente usada e diminui consideravelmente a sobrecarga cognitiva do usuário ao não expor *links* que não sejam interessantes para o usuário de acordo com o seu modelo;
- Classificação dos *links*: a partir do nodo mais relevante os *links* começam a ser classificados de acordo com o modelo do usuário, sendo apresentados logo após em ordem decrescente.

2.1.3 CARACTERÍSTICAS VOLTADAS PARA A ADAPTAÇÃO

Além destas técnicas utilizadas num SHA para se prover a adaptação, deve-se levar em conta outras não menos importantes que serão úteis para a definição do modelo do usuário. Para Brusilovsky [7], existem cinco características voltadas para um SHA que são: os objetivos do usuário, conhecimento, *background*, experiência no *hiperespaço* e preferências.

Os objetivos do usuário são definidos em relação ao sistema utilizado. Por exemplo, num ambiente educacional as metas são de aprendizado, já num ambiente de recuperação de informações pode-se ter como meta a pesquisa. Como as metas podem variar de sessão para sessão, torna-se importante manter as mesmas atualizadas no modelo do usuário. Em relação ao conhecimento, ele pode ser muito bem aproveitado tanto nas mudanças de conteúdo como na navegação do ambiente, pois certas partes podem ser ocultadas ou melhor aproveitadas para o usuário em questão, diminuindo consideravelmente a sobrecarga cognitiva. Já o *background* leva em consideração a profissão, a experiência anteriormente adquirida na área em questão do SHA, facilitando a implementação das técnicas de navegação adaptativa. Por último, as preferências dos usuários adquiridas diretamente através de questionários pré-definidos ou indiretamente através de suas ações constantes, podem facilitar a adaptabilidade do sistema consideravelmente.

3. MODELOS DE REFERÊNCIA PARA SHA

Nas pesquisas sobre SHA foram definidos alguns modelos de referência que formam a estrutura para o desenvolvimento das aplicações. Alguns pesquisadores da área de hipermídia adaptativa defendem um modelo em relação ao outro devido às diversas características ou ferramentas que cada um pode assumir. Serão discutidos neste capítulo os seguintes modelos Dexter, AHAM e Munich. Todos eles possuem módulos que atuam nas duas classes de adaptação descritas no capítulo anterior, abrangendo as técnicas e os métodos utilizados nos SHA.

3.1 DEXTER

O Dexter, modelo criado em um workshop por um grupo de *designers* hipermídia na cidade de New Hampshire, Estados Unidos em 1988, procurou integrar e formalizar aspectos encontrados em diferentes sistemas de hipertexto. Apesar de não descrever uma arquitetura adaptativa, serviu como base para outros modelos, sendo dividido nas três camadas descritas a seguir e na figura 1 [8]:

- camada de armazenamento: núcleo do Dexter, descreve a rede com os nodos e links, sendo que estes últimos são tratados como qualquer componente presente num hiperdocumento;
- camada interna aos componentes: abrange o conteúdo e a estrutura junto com os nodos;
- camada de execução: contém a descrição da apresentação dos nodos e dos links, interações com o usuário e a dinâmica da aplicação em si.

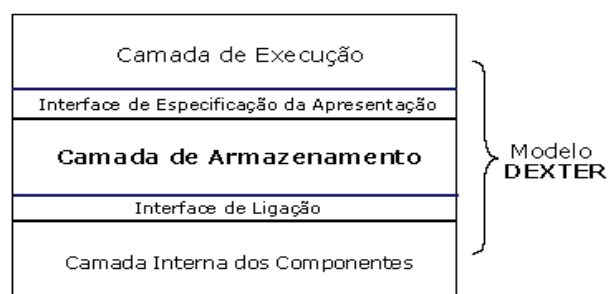


Figura 1 - Arquitetura do Modelo Dexter. Fonte: Halasz, 1990.

A camada de armazenamento é a mais importante deste modelo e é formada por *atoms*, *links* e *composite components*. Os *atoms* são simples nodos. Os *composite components* são estruturas hierárquicas de nodos e os *links* são entidades que representam as relações entre os componentes especificados por duas ou mais âncoras. As âncoras são definidas na parte interna dos componentes que provê um mecanismo para endereçamento. Elas consistem em duas partes: um identificador único para que seja possível a alocação correta no espaço de um hipertexto e um valor que informa a atual região em que se encontra no interior de um componente.

Ao contrário das outras camadas, a camada interna dos componentes está mais ligada ao conteúdo e a estrutura dos componentes de um hipertexto como: textos, imagens, gráficos, sons; não apresentando mais detalhes neste modelo. Já a camada de execução representa a parte dinâmica dos sistemas *hipertexto*. Sua principal idéia está no fato de instanciar um componente ao usuário do sistema. Instanciar um componente resulta em instanciar âncoras denominadas de *link marker*. Em um dado momento, um usuário poderá acessar o hipertexto possuindo algumas funcionalidades como ver ou editar um componente através da entidade *session*. Dessa forma, ele poderá criar, editar e destruir instâncias de componentes no hipertexto. E, no final desta interação a *session* é fechada.

Para efetuar a comunicação entre estas camadas o Dexter propôs interfaces de comunicação. Este modelo não detalha os mecanismos de apresentação. Entretanto, ele pode conter informações sobre como um componente numa rede é apresentado ao usuário na interface de especificação da apresentação, que está presente entre a camada de execução e a camada de armazenamento. Nesta última, é informado ao usuário a relação de um componente numa rede como o caminho para chegar de encontro a ele, ou seja, uma forma de diferenciar qual o usuário que utiliza o ambiente para o estudo ou acessa para editar as aulas, por exemplo. A interface de ligação composta pelas âncoras, localizada entre a camada de armazenamento e a camada interna dos componentes, assume um papel importante ao disponibilizar um mecanismo de endereçamento de *links*, tanto os que são parte integrante do texto quanto aos que interligam os nodos [8].

Apesar de algumas restrições que este modelo apresenta, seu uso facilitou a integralização de ambientes hipermídia desenvolvidos em diferentes sistemas. Além disso, ao separar as funcionalidades do sistema em camadas, demonstrou a clara

separação entre elas e um meio de comunicação entre as mesmas, mostrando ao projetista e ao usuário, respectivamente, as especificidades e a visão geral dos conteúdos e estruturas deste modelo.

3.2 AHAM

O *Adaptive Hypermedia Application Model* (AHAM) foi proposto pelo pesquisador Paul de Bra, professor da Universidade de Antuérpia na Bélgica. Este modelo, mostrado na figura 2, já aborda arquiteturas de hipermídia adaptativas, reformulando as camadas do Dexter, ao inserir módulos que irão ser responsáveis pela adaptação. Segundo DE BRA, para identificá-lo num sistema é necessário prestar atenção a três fatores [9]:

- a aplicação deve ser baseada no modelo de domínio, descrevendo como o conteúdo num hiperdocumento é estruturado, similar à camada de armazenamento do modelo Dexter;
- o sistema deve criar e manter um modelo do usuário puro que represente suas preferências, conhecimentos, objetivos, histórico de navegação e outros aspectos relevantes;
- o sistema, através do modelo de ensino que consiste em regras pedagógicas, terá que ser capaz de efetuar a adaptação de conteúdo e a estrutura dos *links* baseado nos modelos de domínio e do usuário. Assim, surge o modelo de adaptação que segue regras para gerar uma apresentação adaptativa e atualizar o modelo do usuário.

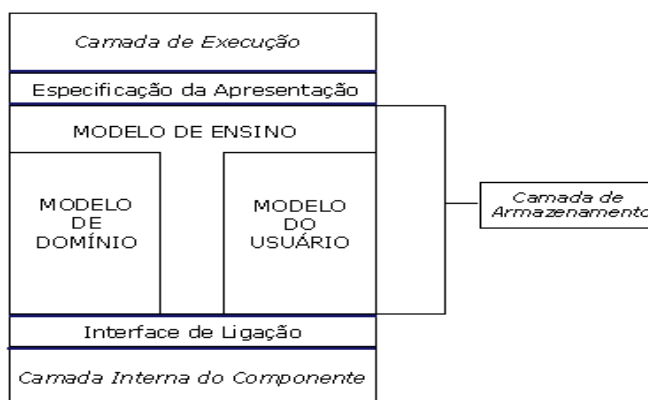


Figura 2: Arquitetura do Modelo AHAM. Fonte: De Bra, 1998.

O modelo do usuário descreve como um usuário reconhece o modelo de domínio, armazenando o quanto cada um sabe sobre os conceitos do domínio da aplicação. As regras nesta relação são responsáveis por orientar a *adaptive engine*, módulo do AHAM, na geração no que o Dexter denomina de especificação da apresentação de um SHA [10].

O AHAM tem como foco a camada de armazenamento, a interface de ligação e a interface de especificação da apresentação, sendo que através de dez definições é possível a elaboração do modelo. São elas [9]:

- 1) em um sistema, o conceito é a representação abstrata de um item de informação do domínio, formado por um par de atributos: um identificador(id) e a informação sobre um componente(info);
- 2) assim como no Dexter, nos conceitos há a diferenciação entre *atoms* e *composites components*. Um conceito atômico pode ser um fragmento de informação, seus atributos e valores pertencem à camada interna dos componentes. Já com os *composites components* os conceitos são estruturados numa árvore, estrutura de dados que representa uma hierarquia de informações de fácil acesso;
- 3) uma âncora é um par id (único) e valor (que especifica alguma localização, região, item ou subestrutura de um conceito);
- 4) um especificador é formado pela tupla: *c_id*, *a_id*, *dir* e *pres*. Onde o *c_id* identifica um componente, o *a_id* identifica uma âncora, o *dir* define a direção que o usuário poderá seguir e o *pres* é especificação da apresentação;
- 5) o conceito é definido por *id*, *seq*, e *info*, que respectivamente representam o identificador, uma seqüência de especificadores e um par de atributo-valor, uma seqüência de âncoras e uma especificação da apresentação;
- 6) os conceitos atômicos junto com os *composites components* e os conceitos relacionados formam o modelo de domínio de uma aplicação hipermídia adaptativa;
- 7) um SHA associa os atributos de um modelo de usuário para cada conceito no modelo de domínio. Cada usuário possui uma tabela onde são armazenados os seus atributos referentes a cada conceito. Uma tabela específica para um determinado usuário é uma instância de seu modelo;
- 8) uma regra pedagógica genérica é formada pela tupla (*trig*, *fase*, *propag*), onde *trig* é uma regra *trigger*, *fase* é uma execução da regra e *propag* assume um

valor booleano informando quando uma regra pode receber um gatilho de outras regras. As regras possibilitam alterações nas variáveis de conceitos e o elemento *fase* é recebido antes ou durante a geração das páginas;

- 9) uma regra pedagógica específica possui a tupla (*strig*, *scomp*, *spropag*), o *strig* e o *spropag* possuem os mesmos tipos de valores da regra anterior, sendo que o *scomp* representa um conjunto de componentes utilizados pelas regras. Devido ao usuário possuir um estado corrente e depois de atualizado o seu estado passar para outro antes da geração das páginas do aplicativo, foram definidas estas duas etapas;
- 10) o modelo de ensino de um SHA é um conjunto regras pedagógicas genéricas e específicas.

O objetivo central deste modelo foi propor um *framework* para o desenvolvimento de SHAs. Além de o AHAM possuir uma clara separação entre: o espaço em que se encontra o conceito, a informação atual, a estrutura dos *links* e as dependências entre os conceitos. Com isso, introduziu a atualização constante da aplicação do modelo de usuário num intercâmbio com os conceitos, promovendo a adaptação.

3.3 MUNICH

O Modelo de Munich, proposto por Nora Koch [5], define uma extensão do modelo Dexter ao incluir certas características como uma especificação mais formal e algumas alterações nas camadas definidas no modelo anterior. Através da notação UML (*Unified Modeling Language*) e do modelo orientado a objeto. Este foi o primeiro modelo a propor uma especificação visual de um SHA constituindo a UWE (*UML-based Web Engineering*).

A UWE tem como principal função modelar os elementos, notações e métodos. Os modelos propostos são: conceitual, usuário, navegação, apresentação e o modelo de adaptação. Além da UML, que vem sendo considerada uma das linguagens mais expressivas para modelagem orientada a objetos, é utilizado a linguagem OCL (*Object Constraint Language*), parte integrante da UML, que tem a

capacidade de descrever de forma precisa através de uma linguagem textual, as limitações do modelo orientado a objetos, definindo uma representação semântica.

A principal camada do modelo Dexter que era responsável apenas pelo armazenamento das informações referentes à estrutura hipermídia, passa agora a ser dividida em três sub-modelos que podem ser visualizados na Figura 3 [11]:

- meta-modelo de domínio – responsável pela estrutura da rede composta por nodos e *links*, sendo que o primeiro é considerado como recipiente de dados;
- meta-modelo de usuário – características dos usuários armazenadas para uma posterior adaptação;
- meta-modelo de adaptação – constitui numa série de regras que implementam as formas de adaptação.

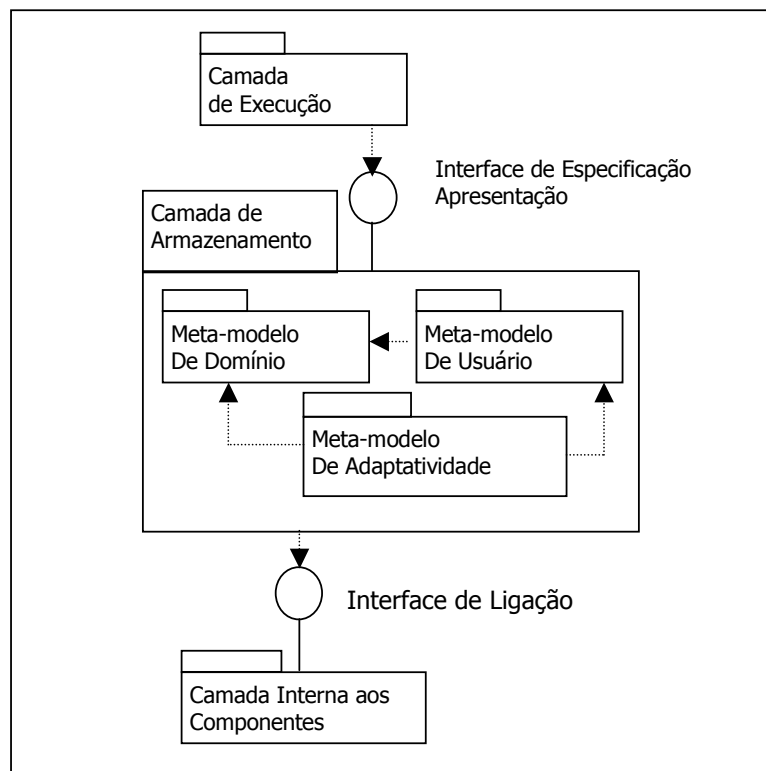


Figura 3: Arquitetura do Modelo Munich. Fonte: Koch, 2002.

Para prover as funcionalidades de um SHA, o modelo define três operações: a autorização, necessária para atualizar componentes, regras e atributos dos usuários; a recuperação útil para o acesso à estrutura do domínio hipermídia e ao Modelo do Usuário e as operações para adaptar de forma dinâmica o Modelo do Usuário, ou

seja, personalizar a apresentação de um aplicativo de acordo com o estado atual do usuário.

Na especificação do Modelo de Domínio também foram criadas três operações: *resolver*, *accessor* e *constructor*. Com elas é possível recuperar e criar componentes adaptativos. Através de um identificador é possível o acesso ao componente pela operação *accessor*, garantindo o endereçamento de um componente no domínio hipermídia. No domínio são criadas classes e componentes que representam as interações presentes nesta arquitetura, como as funcionalidades da navegação adaptativa [12].

Já os mecanismos de adaptação são descritos no Modelo de Adaptação que é responsável pela adaptação do conteúdo, dos *links* e da apresentação. O Modelo do Usuário representa a estrutura de cada modelo de usuário e como estes são administrados. Ou seja, modelar um usuário consiste em inicializar, atualizar e recuperar o corrente estado deste usuário no modelo. Assim, como no modelo de domínio, são criadas classes como a classe *User*, por exemplo, que identifica um usuário específico e possui o seu conjunto de atributos. Como cada usuário contém as suas características como: conhecimento, preferências, experiências; as mesmas são divididas em duas categorias: conhecimentos do usuário relatados para os componentes do domínio e as características gerais do usuário, não dependentes do domínio.

Por fim, o objetivo central do Munich foi apresentar um modelo de referência para um SHA, constituindo na base da UWE, apesar de ter na arquitetura semelhanças com o AHAM e o Dexter, estende algumas importantes funcionalidades adaptativas no modelo de armazenamento.

Neste sentido, este estudo opta pela utilização do Modelo de Munich para a definição das classes de navegação e de adaptação, devido a este modelo ter sido desenvolvido especificamente para SHAs. Além disso, sua especificação formal e metodologias empregadas vêm de encontro aos interesses deste trabalho. Por isso, este modelo será detalhado mais a fundo no capítulo cinco onde o desenvolvimento da aplicação proposta será efetivado.

4. XML EM APLICAÇÕES HIPERMÍDIA ADAPTATIVAS

Aplicações hipermídia adaptativas requerem tecnologias que propiciem uma adaptação de forma automática e um armazenamento dinâmico das informações. No padrão XML, os dados são estruturados numa árvore onde não se leva em questão os aplicativos empregados ou a plataforma utilizada, sendo possível a criação de inúmeras *tags*. Então, a estrutura de um SHA pode ser representada por XML e suas folhas de estilo (XSL), pois todos os nodos devem ser organizados numa hierarquia que facilite uma eficiente localização de seus componentes.

Neste capítulo será abordado as características, estrutura e exemplos desta linguagem aplicados em alguns SHAs. No capítulo seguinte, XML é utilizada como uma forma de armazenamento de conteúdo para a aplicação proposta neste trabalho.

4.1 A LINGUAGEM XML

A XML (*Extensible Markup Language*) foi desenvolvida pelo consórcio internacional de pesquisadores, o W3C [13], responsáveis por vários protocolos presentes na *web*. O foco desta linguagem é a descrição de dados através de uma linguagem de marcação e não a sua visualização como HTML o faz. Além disso, ela foi criada para estruturar, guardar e enviar informações seja em qual for o ambiente da aplicação.

Tanto o HTML quanto o XML derivam da linguagem de marcação SGML e, portanto, apresentam características similares. A SGML é um padrão internacional de código aberto. Um dos seus objetivos é garantir que os documentos tenham uma estrutura de marcação formal. A XML foi idealizada por Jon Bosak, engenheiro da Sun Microsystems, que teve a idéia de explorar o SGML em aplicações voltadas para Internet. Em 1996, foi criado o XML, inicialmente como uma versão simplificada do SGML, e, em fevereiro de 1998, o XML tornou-se uma especificação formal pelo consórcio W3C [14].

Uma das possíveis abordagens para resolver os problemas na troca de dados entre aplicações é por meio de um armazenamento num banco de dados relacional. E através de uma linguagem de programação, junto com a SQL, acessar e

manipular os dados. Para solucionar este e outros problemas a XML foi criada e o consórcio W3C mantém este projeto, sendo responsável pela sua especificação, além de uma contínua busca na evolução desta linguagem.

4.2 ESTRUTURA E SINTAXE DA XML

Na figura 4, estão presentes as *tags* que estruturam este documento no formato XML. O nome dado a cada *tag* corresponde a um elemento. Assim, para que este documento não possua erros na sua sintaxe, cada elemento deve marcar um início e um fim, sendo as suas respectivas *tags* idênticas. Para verificar se tudo está certo ou bem-formado, basta abrir o arquivo dentro de qualquer navegador de Internet compatível com esta tecnologia, ou seja, que já possui um analisador de sintaxe (*parser*) incluso. Caso não possua erros, a mesma estrutura de marcação será mostrada na tela como foi definida.

```
<cardapio>
  <prato>
    <nome>Alcatra</nome>
    <preco>35</preco>
    <descricao>refeição para duas pessoas,
acompanha arroz, maionese e salada</descricao>
    <calorias>1500</calorias>
  </prato>

  <prato>
    <nome>Picanha</nome>
    <preco>48</preco>
    <descricao>refeição para duas pessoas,
acompanha arroz, maionese e salada</descricao>
    <calorias>1800</calorias>
  </prato>
</cardapio>
```

Figura 4: Exemplo de um arquivo XML.

Nota-se que a estrutura é hierárquica, existindo sempre um elemento raiz, por exemplo, o elemento *cardapio*. Cada elemento pode ser composto por um ou mais elementos, chamados de “elementos filhos”, como o elemento *prato*. E, cada elemento filho também pode possuir a sua própria estrutura, como em *nome*, *preco*, *descrição* e *calorias*, que são elementos filhos do elemento *prato* (elemento pai).

Por ser uma linguagem extensível, as *tags* podem ser elaboradas pelo autor do documento, facilitando a compreensão da mesma e a sua definição estruturada em

texto e marcação. Entretanto, devem seguir algumas regras para a definição de nomes, presente na especificação da XML [15]:

- podem conter letras, números e outros caracteres.
- não iniciam com números ou caracteres de pontuação.
- não iniciam com as letras xml, Xml ou XML, etc.
- não contém espaços.
- devem ser curtos e simples.
- utilizar, preferencialmente, os mesmos nomes dados aos campos do banco de dados que a aplicação trabalha.

Os elementos podem conter atributos além dos seus nomes. Os dados tanto podem ser armazenados em elementos como em atributos. Mas, o autor de um documento XML deve tomar cuidado na sua utilização, pois um atributo deve apenas prover informações que não são relevantes para um elemento em especial. A preferência pelo uso de elementos ao invés de atributos se deve às seguintes limitações deste último em relação ao primeiro [14]:

- atributos não contém múltiplos valores.
- não descrevem estruturas.
- a sua manipulação é mais difícil.

Uma DTD (*Document Definition Type*) procura formalizar quais *tags* podem ser usadas na elaboração de um documento XML, sendo declarada internamente ou numa referência externa.. Possui uma sintaxe não tão simples à primeira vista, mas ganha importância quando os documentos precisam ser trabalhados por diferentes membros de uma equipe de projeto, facilitando o intercâmbio das informações ou para validar documentos XML descritos por outros setores de uma empresa, por exemplo.

Para uma DTD, um documento XML é formado basicamente por uma quantidade de blocos definidos como: elementos, atributos, entidades, *PCDATA* e *CDATA*. Os elementos são considerados o bloco principal. Os atributos provêm informações adicionais a um elemento. Entidades são variáveis usadas para se definir um trecho de código comum. *PCDATA*, que é reconhecido pelo *parser*, especifica o que é elemento e as entidades podem ser expandidas. Por último, ao

contrário do *PCDATA*, o *CDATA* não é reconhecido pelo *parser* e define o que não é elemento [15].

Devido à rigidez desta estrutura, o W3C desenvolveu um novo recurso para suportar algumas deficiências de uma DTD, como a falta de suporte a tipo de dados e a *namespaces*, que são nomes pré-definidos em outros documentos e podem ser reutilizados. Assim, criou-se a recomendação XML *Schema* similar a DTD, mas muito mais poderosa. Pois, com ela fica mais fácil trabalhar com dados oriundos de um banco de dados, definir restrições para um determinado dado e os seus respectivos formatos e converter um tipo de dado em outro sem muitos problemas. Uma outra grande vantagem é que não há a necessidade de aprendizado de uma nova linguagem, porque XML *Schemas* é escrita em XML.

A seguir, os dois trechos de código apresentam uma referência a um mesmo documento XML, diferenciadas apenas pela forma de validação:

```

<!--Arquivo DTD-->
<!ELEMENT prato (nome, preco, descricao, calorias)>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT preco (#PCDATA)>
<!ELEMENT descricao (#PCDATA)>
<!ELEMENT calorias (#PCDATA)>
<!--Arquivo XML Schema-->
1 <?xml version="1.0"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
3 targetNamespace="http://www.selfservice.com.br"
4 xmlns="http://www.selfservice.com.br"
5 elementFormDefault="qualified">
6 <xs:element name="prato">
7   <xs:complexType>
8     <xs:sequence>
9       <xs:element name="nome" type="xs:string"/>
10      <xs:element name="preco" type="xs:integer"/>
11      <xs:element name="descricao" type="xs:string"/>
12      <xs:element name="calorias" type="xs:integer"/>
13    </xs:sequence>

```

```
14 </xs:complexType>
15 </xs:element>
16 </xs:schema>
```

Nota-se que as definições no arquivo XML *Shema* são mais amplas e claras em relação ao arquivo da DTD. A linha 2 do código acima, que referencia um *namespace*, indica a URL onde é definido quais os tipos que podem ser usados no documento (elementos, tipos de dados, string, *boolean*, etc.). Já na linha 3, define que os elementos *prato*, *nome*, *preco*, *descricao* e *calorias* são oriundos da URL (<http://www.selfservice.com.br>). E, entre as linhas 9 e 12, os elementos filhos são declarados junto com o suporte de tipos de dados (*string*, *integer*, *date*, *boolean*, etc.)

4.3 XML E SUAS FOLHAS DE ESTILO

Para um arquivo em XML ser bem aplicado no intercâmbio de dados ou numa apresentação visual, deve atuar em conjunto com outras linguagens padronizadas. No grupo das três linguagens que fazem parte da XSL (*Extensible Stylesheet Language Family*), uma das recomendações de XML definidas pelo W3C, semelhante a ligação dos arquivos CSS e HTML, existe a XSLT que possui a funcionalidade de transformar um arquivo XML em outro, além de poder gerar qualquer outro formato de texto [16]. Assim, é possível a geração de novos *layouts* a partir de um mesmo arquivo em XML. Já para encontrar uma *tag* específica neste arquivo, a XSL utiliza a Xpath, linguagem responsável pela navegação. Por último, a XSL-FO integra uma variedade de atributos que definem as propriedades de formatação.

A XSLT vai atuar no aplicativo proposto a fim de facilitar a geração das diferentes apresentações necessárias para as adaptações definidas pelo perfil de cada usuário. Além de promover diferentes apresentações, ela pode ser útil no envio de dados entre as aplicações, caso o software no destino necessite um documento em conformidade com uma versão diferente de uma DTD ou XML *Schema* do que aquela que está sendo utilizada. Sabe-se que a interoperabilidade pode ser alcançada de várias maneiras. Por exemplo, talvez seja preciso alterar alguns

elementos em atributos ou modificar o espaço de nomes utilizado para qualificar os elementos e atributos. A linguagem de programação Java poderia ser utilizada nesta transformação se o processamento através do SAX ou DOM não fosse tão difícil, tornando-se como a melhor alternativa nesta situação a XSLT [17].

A Fig. 5 descreve o funcionamento deste mecanismo de transformação. Um documento escrito em XML mais um programa XSLT, escrito também em XML, são as entradas para um processador XSLT que gera a saída de um novo documento. Há muitos processadores deste tipo disponíveis no mercado, como o Saxon, o MSXML e o Xalan. Mais adiante na implementação, as estrutura de um programa XSLT serão mais detalhadas, assim como outras linguagens de programação que serão necessárias na etapa de desenvolvimento.

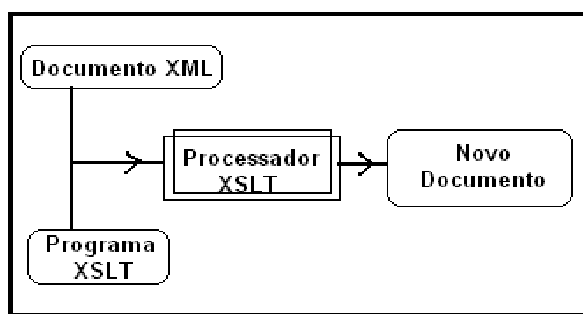


Figura 5: Processamento de um arquivo XSLT.

4.4 EXEMPLOS DE SHAS QUE UTILIZAM XML

Esta tecnologia é importante para este estudo, pois já existem várias ferramentas que utilizam XML para o armazenamento das regras adaptativas e para a estruturação do conteúdo, principalmente nos SHAs educacionais. Mas, antes de serem apresentados aos alunos, o documento em XML é verificado e analisado, obedecendo ao seu modelo de usuário. Isso permite um fácil intercâmbio de informações entre as páginas HTML e a exibição correta dos conceitos.

O AHA! [18], aplicação de código aberto que se baseia no AHAM como modelo de referência, fornece um bom exemplo ao utilizar arquivos em XML para armazenar as regras de adaptação, modelo de domínio, modelo de usuário e os conceitos. Cada página é criada em HTML com o XML possuindo uma série de regras definidas usadas para a adaptabilidade da navegação, por exemplo. O AHA!

lê o arquivo XML através do Java *Servlets*, armazenando dentro de um objeto a fim de manter as interações adaptativas com o usuário no decorrer do processo.

Na figura 6 [19], é mostrado um código XML do AHA! para representar um conceito sobre o interesse em relação a uma página da cerveja *de-koninck* da Antuérpia. Nota-se que esta página foi pré-definida como interessante para os usuários que possuem uma idade superior a vinte anos. A página ainda não acessada assume um valor inicial *false* de tipo *boolean* passando para *true* no momento em que é acessada. O interesse nesta marca por ser um atributo persistente, deve ser armazenado no modelo do usuário. Além disso, também há a atualização dos atributos através de ações condicionais. Assim, ao acessar a página sobre cerveja há um acréscimo no interesse do usuário e ao mesmo tempo um decréscimo no seu interesse por chocolate.

```

<concept>
  <name>de-koninck</name>
  <desc>Beer from Antwerp</desc>
  <requirement>beer.interest > 20</requirement>
  <attribute name="access" type="bool" isPersistent="false">
    <desc>standard attr: true when page accessed</desc>
    <generate>
      <requirement>beer.interest < 100</requirement>
      <trueAction>beer.interest += 10</trueAction>
    </generate>
    <generate>
      <requirement>chocolate.interest >= 5 and
        chocolate.interest < 50</requirement>
      <trueAction>chocolate.interest -= 5</trueAction>
    </generate>
  </attribute>
  <attribute name="interest" type="int" isPersistent="true">
    ...
  </concept>

```

Figura 6: Arquivo XML – Modelo de Domínio. Fonte: De Bra, 2002.

Um outro exemplo, é a ferramenta ArgoUWE [20], que engloba a UWE e XML para o desenvolvimento de aplicações *web* de forma automática. Foi implementada para ser adicionada à ferramenta de modelagem UML ArgoUML. Funcionando da seguinte forma: novos tipos de diagramas são adicionados ao ArgoUML para incluir a representação dos modelos definidos pelo método UWE. Nesta transição, são usados arquivos em XML para a geração destes modelos [21].

```

<Model>
  <name>Business</name>
  <visibility xmi.value="public"/>
  <Class>
    <name>Customer</name>
    <feature>
      <Attribute>
        <name>id</name>
        <multiplicity>
          <XMI.field>1</ XMI. field>
          < XMI. field>1</ XMI. field>
        </multiplicity>
      </Attribute>
    </feature>
  </Class>
</Model>

```

Figura 7: Arquivo XML gerado pela ArgoUWE. Fonte: Iyengar, 1999.

Na figura 7, aparece um código gerado a partir do fragmento de um modelo elaborado através de um diagrama em UML, sendo feito um detalhamento dos dados num simples arquivo XML. O nome do modelo, a classe envolvida e os seus atributos são projetados para um arquivo texto, tornando possível, posteriormente, a geração automática de aplicações para a *web*.

Já o XHAM [22], um SHA baseado em XML, as camadas são compostas por documentos em XML que descrevem cada página e os segmentos de conteúdo. Assim, é possível o acesso a dados e às funções de forma independente, gerando conteúdos em tempo de execução. O XML pode conter, por exemplo, uma imagem com diferentes níveis de detalhamento e no decorrer do texto ela é organizada numa hierarquia de fragmentos, alcançando diferentes versões e localizada no espaço adaptado seguindo o modelo do usuário. Para isso, este sistema utiliza os esquemas de XML de um repositório de dados implementando logo após a interface personalizada.

Por fim, seja no intercâmbio de dados ou na apresentação de novos *layouts*, XML e a criação de novos padrões a partir desta linguagem, facilita o desenvolvimento de aplicações robustas e que operam em diferentes sistemas operacionais ou aplicativos. Além disso, os sistemas de grande porte existentes no mercado não sobrevivem mais sem o uso desta tecnologia e suas complementações, por torna-se um padrão para quem busca flexibilidade, portabilidade e um amplo emprego na *web*.

5. DESENVOLVIMENTO DA APLICAÇÃO PROPOSTA

Para mostrar as vantagens do uso de um SHA num ambiente de ensino, este trabalho propõe um aplicativo que segue as características do Modelo de Munich, descrito previamente no capítulo três. O Munich trabalha com a complexidade de um SHA ao seguir uma abordagem de engenharia de software (UWE) que é composta por uma série de modelos a fim de promover um adequado desenvolvimento da aplicação. Somando-se a isso, a representação visual destes modelos através da UML, possibilita uma fácil abstração de toda a estrutura.

Este capítulo tem como objetivo descrever a modelagem e a implementação da aplicação proposta, seguindo o modelo de Munich como metodologia de desenvolvimento, bem como a utilização do XML como forma de armazenamento de conteúdo.

5.1 OS MODELOS ADAPTATIVOS DA UWE

A UWE atua na definição de um projeto sistemático, personalizado e de geração semi-automática para as aplicações WEB. É orientado a objeto, tendo uma abordagem interativa e incremental, utilizando exclusivamente a linguagem UML, suas técnicas, notações e extensões.

Segundo Koch [5], a UWE compõe as diversas etapas do ciclo de vida de um SHA e consiste na modelagem de elementos, notações e métodos. A notação é definida pela UML que representa um conjunto de estereótipos e propriedades específicos num SHA. Os métodos definidos por esta metodologia de desenvolvimento são: modelo de casos de uso, modelagem conceitual, navegacional e de apresentação. Cada modelo é representado na Figura 8 a seguir, por meio de pacotes de classes.

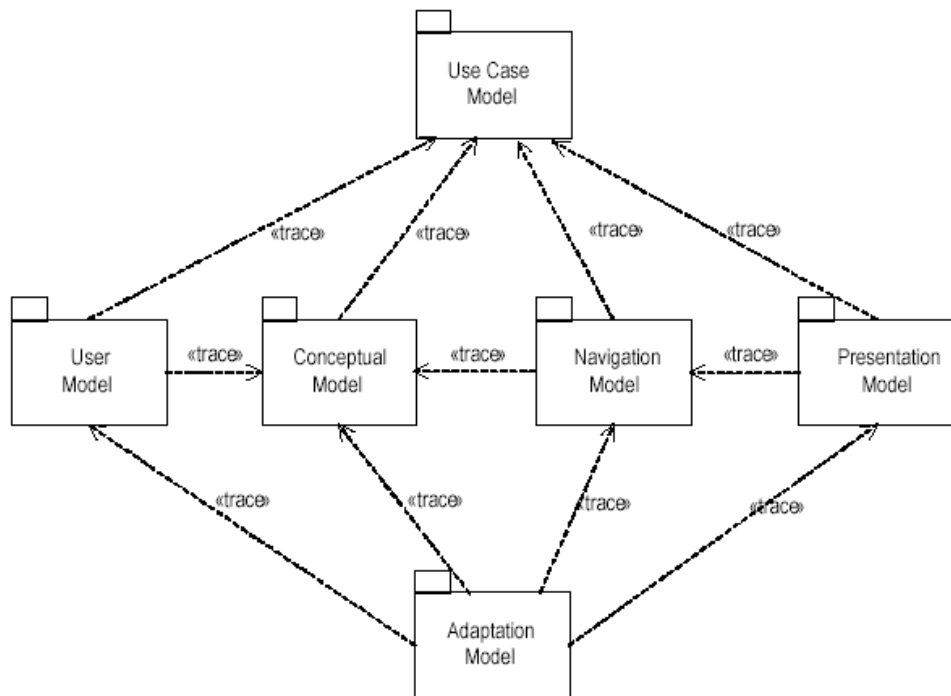


Figura 8: Modelos na fase de planejamento de um SHA. Fonte: Koch, 2002.

Devido aos aspectos dinâmicos encontrados num SHA são incluídos os diagramas de atividades e de estado. Na modelagem conceitual e navegacional são definidos *profiles* da UML que permitem estender a notação para incorporar novos aspectos por meio de estereótipos. Já no modelo conceitual são representados os elementos do domínio através de um diagrama de classes numa estrutura estática.

A estrutura de navegação é constituída de dois modelos: modelo navegacional de espaço e modelo navegacional de estrutura. O primeiro tem a função de especificar quais objetos podem ser alcançados pela navegação. O segundo define como estes objetos são alcançados. Na fase de modelagem de apresentação é utilizado um tipo particular de diagrama de classes.

Todos estes modelos e suas respectivas notações em UML são detalhados a seguir de acordo com as características da aplicação proposta. Esta análise da aplicação torna-se uma parte fundamental para uma boa modelagem e implementação, pois procura identificar todas as características que estão presentes. Além disso, a UWE possibilita uma seqüência de passos e, por conseqüência, desmistifica a complexidade do desenvolvimento de um SHA, que se não fosse feito através de uma formalização, poderia dificultar a manutenção e a reusabilidade do aplicativo em questão.

5.1.1 MODELO DE CASO DE USO

Neste modelo as interações do sistema com os seus usuários (atores) ou outro sistemas são revelados. Um ator é parte integrante do sistema e interage com ele, por isso é importante identificar todos os atores e as operações que eles processam no ambiente, não sendo necessário mostrar ainda como estas seriam implementadas. Além disso, este modelo passa para o sistema uma visão da hierarquia dos usuários, relatando o que cada um pode fazer no decorrer do processo de utilização do aplicativo.

Na seqüência a seguir, são mostrados alguns requisitos da aplicação:

- os usuários são matriculados primeiramente antes de possuir o acesso às aulas;
- os visitantes acessam apenas algumas informações sobre o sistema;
- o modelo do usuário (MU) é adaptado dinamicamente;
- as aulas são personalizadas seguindo o MU;
- a edição das aulas e o controle de usuários são acessos restritos.

Apesar de possuir um alto nível de abstração, no diagrama de casos de uso são demonstradas algumas situações de processamento da aplicação, suas funcionalidades e responsabilidades. Com isso, o modelo não permanece estático na maioria dos sistemas. Com a evolução das atividades presentes na aplicação, novos atores podem ser definidos como responsáveis por novas funções. Apesar de ser relativamente simples a construção deste modelo, a sua importância é fundamental para as definições das classes de domínio da aplicação e as relações que cada uma possui em relação à outra.

No aplicativo proposto, foram identificados os seguintes atores e suas respectivas associações: Usuario, Visitante, UsuarioConstante, UserAutor, UserMatriculado, Autor e Administrador. Na Figura 9 tem-se a estrutura e as associações entre estes atores. Um Usuário (pai) pode ser tanto um Visitante (filho) como um UsuarioConstante (filho), sendo que os elementos filhos herdam as funcionalidades do elemento pai. Já o UsuarioConstante é dividido em UserAutor e UserMatriculado. Por último, o UserAutor é um Autor ou um Administrador.

Na Figura 10 são definidas as atividades que os atores deste sistema podem realizar. O Usuário (Figura 10 (a)), elemento que possui certas funcionalidades referenciadas pelos elementos restantes, como o acesso inicial à aplicação, a visualização de notícias e o envio de e-mails para o sistema. O Visitante (Figura 10 (b)), além de herdar as atividades do Usuário, pode solicitar o cadastro, visualizar uma aula demonstrativa, onde são mostrados os elementos atrativos de uma página hipermídia adaptativa e, por fim, pode optar pelo recebimento de notícias do sistema, como avisos da próxima matrícula e etc.

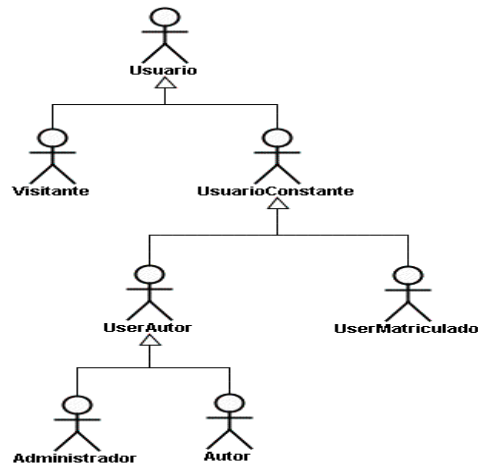


Figura 9: Modelo de Caso de Uso.

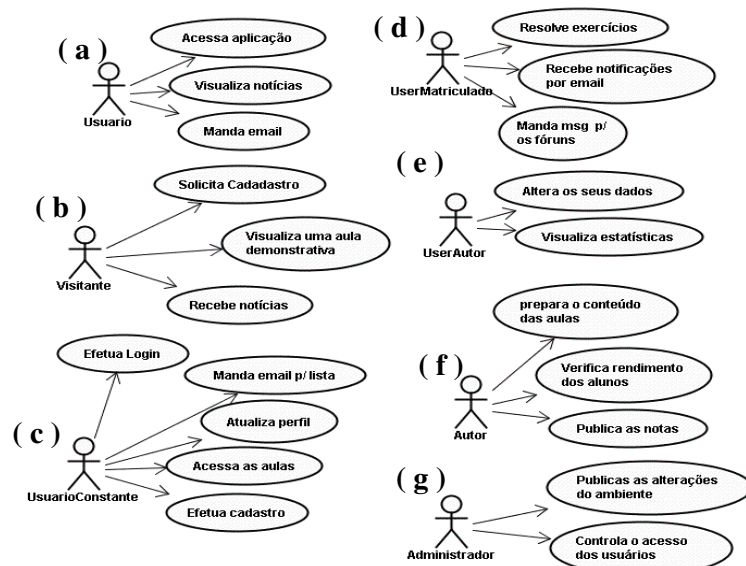


Figura 10: Diagrama de Atividades.

Já um *UsuarioConstante* (Figura 10 (c)), elemento pai dos elementos relatados a seguir, efetua cadastro e *login* no ambiente, acessa as aulas, manda e-mails para a lista de contatos, além de atualizar o seu perfil no modelo de usuário. Um *UserMatriculado* (Figura 10 (d)) representa os alunos da disciplina que resolvem os exercícios, recebem notificações por e-mail do professor (*Autor*) ou do Administrador e manda mensagens para os fóruns. Ao *UserAutor* (Figura 10 (e)) cabe apenas as funções básicas de poder alterar os seus dados como a sua senha, por exemplo, e a visualização de estatísticas, que mostra a este ator informações referentes aos usuários que estão *on-line* no momento, quantos exercícios foram feitos e quais são os que mais participam das atividades propostas. O *Autor* (Figura 10 (d)) fica responsável pela preparação do conteúdo das aulas, a verificação dos rendimentos de seus alunos e a publicação das notas. Por último, o *Administrador* (Figura 10 (e)) que publica as alterações feitas na aplicação e controla o acesso dos usuários.

O ator *UserMatriculado* é o referencial para atualização do modelo do usuário. O mesmo representa um aluno da disciplina da matemática que procura o entendimento dos conceitos da disciplina e uma correta resolução dos exercícios. As relações deste ator com os outros atores presentes no sistema e as suas ações, definem onde o SHA deve atuar para satisfazer as necessidades de seus usuários.

5.1.2 MODELO CONCEITUAL

Como em outros sistemas, este modelo revela uma visão geral do domínio da aplicação. São definidas as classes e os seus atributos existentes aos quais os usuários interagem. Assim, como nos casos de uso, o modelo conceitual ignora as formas de apresentação, navegação e interação. Ele apenas procura representar todas as associações entre as classes e as hierarquias de herança. Estas classes, posteriormente no modelo de navegação, derivam os nodos da estrutura hipermídia e as associações definem os *links*.

A Figura 11 demonstra o modelo conceitual da aplicação proposta. A classe *Aulas* é parte principal deste sistema. É nela que ocorre a maior parte das adaptações necessárias. Sendo composta por um título, um resumo (pequena descrição do que será explicado), a data de publicação efetuada pelo *UserAutor*, um professor responsável, número de acessos até o momento e o seu endereço na *web*. Uma aula também pode conter um ou mais tópicos sobre o assunto que será

discutido. Nesta classe podem existir comentários dos usuários ou sugestões de *sites* sobre o assunto. Além disso, numa aula o *UserAutor* pode anexar arquivos e responder a fóruns, assim como o *UserMatriculado*. Por fim, a classe *Exercicios* é importante para as avaliações dos alunos e possui níveis de dificuldade de resolução que só são alterados seguindo o MU.

Apesar deste modelo ainda não alcançar a representação navegacional, possui todos os recursos que os alunos utilizam no aplicativo. E, por isso, torna-se importante para os modelos seguintes que utilizam as suas classes e relações para a construção de suas estruturas.

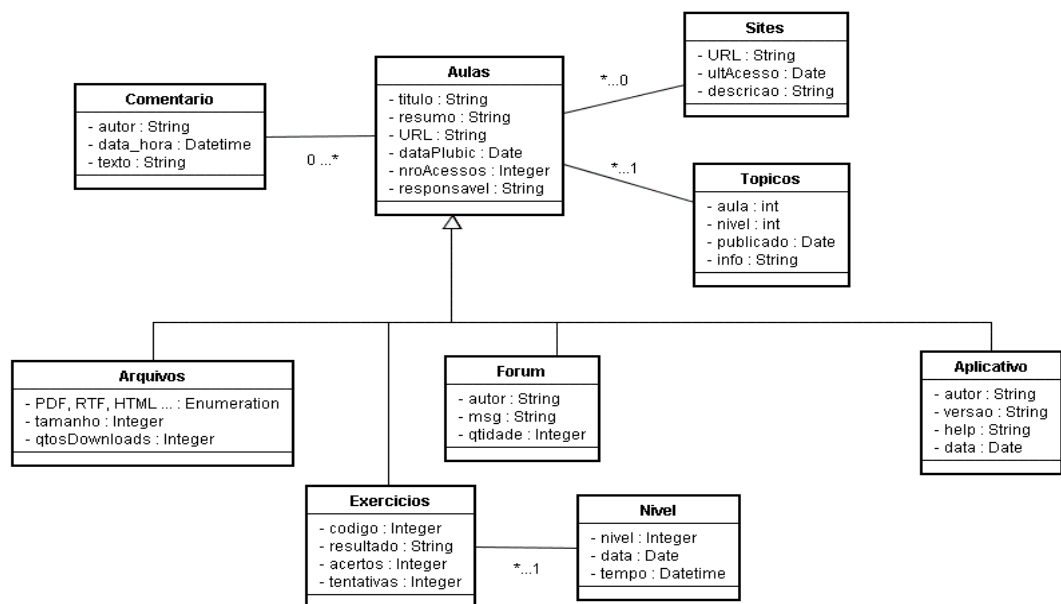


Figura 11: Modelo Conceitual.

5.1.3 MODELO DE USUÁRIO

O Modelo de Usuário (MU) tem a função de armazenar o perfil do aluno, suas características, preferências, interesses, tarefas e a navegação feita por ele no sistema. Os atributos dos alunos devem ser relacionados uns com os outros e interagir com as classes do domínio. De acordo com Koch [5], o MU deve orientar um usuário no aprendizado de um determinado tópico, oferecer informações adaptadas ao aluno, ajudá-lo a encontrar as informações, oferecer um *feedback* sobre o seu desempenho e prestar assistência ao uso do sistema.

A aplicação ficará responsável pelas modificações deste modelo de acordo com as observações feitas sobre o perfil do aluno nas interações que o mesmo pratica no uso do sistema, principalmente em relação ao caminho percorrido pela estrutura navegacional. Sendo esta maneira a mais simples para se obter o que um aluno almeja, sem que ocorram interrupções no uso do aplicativo para a coleta destes objetivos. Já que no início do uso da ferramenta é passado para um aluno um questionário a fim de conhecer as mais importantes características do seu perfil para fornecer algumas adaptabilidades necessárias, sendo mais fácil nesta fase inicial a coleta das informações. Já com o passar do tempo, fica cada vez mais onerosa a manutenção desta personalização, pois a alteração do MU pode seguir uma série de regras ou observações complexas, que nem sempre retratam o perfil atual do aluno.

Representado a seguir na Figura 12, no MU é utilizado o diagrama de classes com suas respectivas associações. Portanto, um sistema para prover uma adaptabilidade adequada para os seus usuários, deve abstrair todas as informações relevantes através deste modelo e atualizá-lo constantemente.

No MU cada usuário possui um nível de acesso ao conteúdo, além de possuir preferências que podem alterar a visualização das aulas, definindo a classe denominada de apresentação adaptativa. Neste modelo também são definidas algumas pendências em relação à resolução de exercícios e a mudança de nível, efetuando as mudanças necessárias para o aluno prosseguir no aprendizado ou exigir que ele refaça algumas atividades que ainda não tenha atingido o conhecimento necessário.

Assim, a classe Usuario, possui todas as informações referentes a um aluno da disciplina a fim de facilitar o seu processo de identificação na aplicação. Cada aluno tem um nível de acesso que possibilita o acesso apenas aos recursos presentes naquele nível, ou seja, ele só poderá ir para a próxima aula se possuir o conhecimento adequado, comprovado pela resolução de exercícios ou de outras atividades definidas pelo professor.

É na classe Aulas que o aluno acessa o conteúdo através de um título, resumo e texto. Nesta classe, também são definidos mais alguns atributos como o seu endereço na *web* (URL), data da sua publicação, o número de acessos e o responsável pela aula em questão. Em cada aula há exercícios para serem resolvidos pelos alunos que formam um dos meios de avaliação mais importantes da

aplicação, influenciando de maneira direta na mudança de nível de acesso no decorrer do uso desta ferramenta de ensino.

Cada aluno, ao nível de personalização do conteúdo, possui preferências como: configurações da apresentação (cores, fontes de texto); visualização ou não de informações adicionais sobre *links* (anotação); visualização ou não de resumos sobre as aulas; acesso a todos os *links* ou somente a alguns (ocultação). Além disso, o aplicativo armazenará a data em que foram feitas estas alterações e caso o aluno tenha informado um período de vigência para a permanência destas.

Além de este modelo ser a peça chave na modelagem deste SHA, o MU assume um papel de importância na aplicação proposta, pois com os seus atributos sendo atualizados constantemente, o aluno pode visualizar a disciplina de uma forma mais personalizada aos seus interesses.

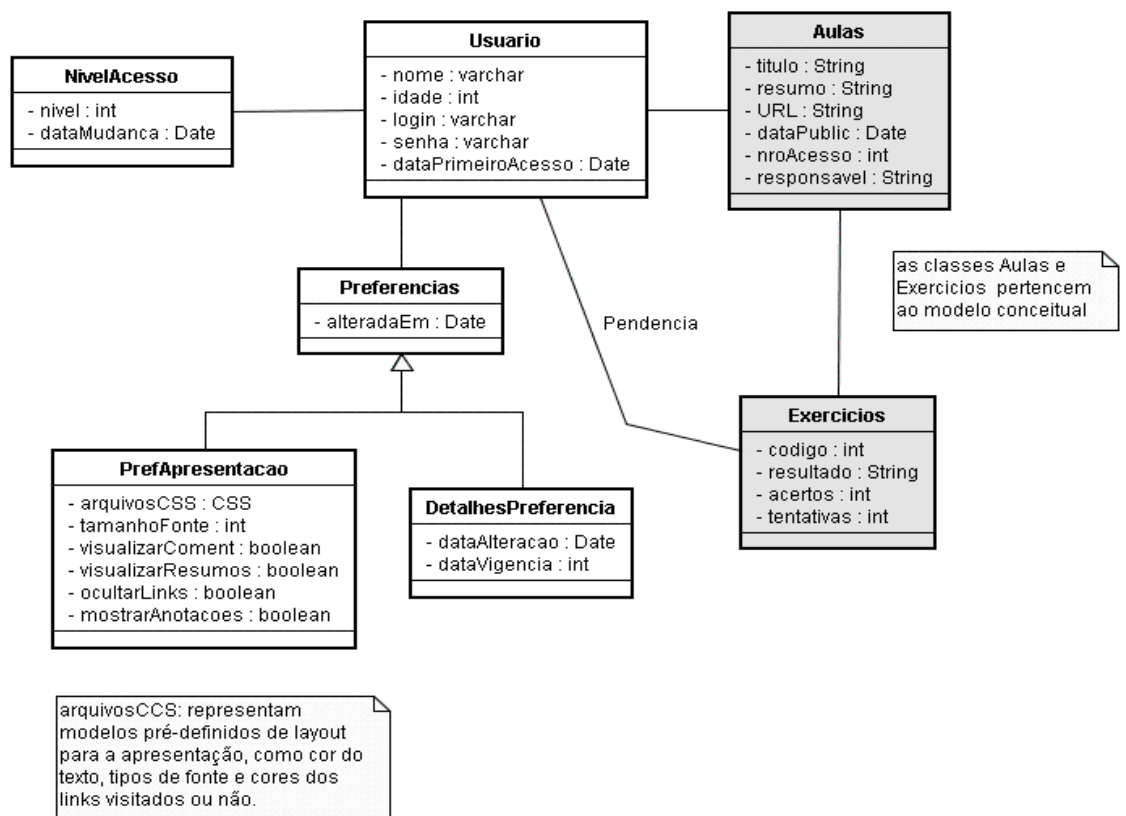


Figura 12: Modelo de Usuário.

5.1.4 MODELO DE NAVEGAÇÃO

Um dos maiores problemas existentes na *web* é a desorientação que a maioria dos sites provocam aos seus visitantes. Excesso de *links* ou a falta deles propicia ao internauta uma perda considerável de rendimento na leitura das informações que lhe interessa. Além disso, nem sempre a sua identificação representa o que um usuário queria saber ou o caminho adequado para uma continuação da resolução das suas dúvidas que o levaram a visitar o site.

Sendo assim, procura-se com esta aplicação amenizar estes problemas através de dois modelos de navegação propostos por Koch [5] que são: o modelo de Hiperespaço de Navegação e o modelo de Estrutura de Navegação. O primeiro define quais objetos podem ser acessados durante a navegação e o segundo especifica como estes objetos são localizados dentro da estrutura.

Segundo Koch [5], durante a construção do modelo de hiperespaço de navegação, algumas classes da modelagem definem esta estrutura: classes navegacionais (CN), nodos externos (NE) e as associações navegacionais ou navegação direta (ND). As CN modelam quais as instâncias de uma classe pode ser acessada durante a navegação no aplicativo e os seus nomes podem ser iguais às suas respectivas classes do modelo conceitual, de onde estas classes derivam atributos que não são incluídos no modelo navegacional. Os NE representam *links* para outras aplicações hipermídias, não fazendo parte do aplicativo modelado inicialmente. Por último, a ND define diferentes associações das que são usadas pelo modelo conceitual. Na representação das direções a serem criadas neste modelo, a ND utiliza associações diretas, que é uma flecha em ambos os lados ou em apenas um. Dependendo da multiplicidade das relações uma regra de nome é definida. A Figura 13 expõe este modelo de hiperespaço de navegação referente a aplicação proposta neste trabalho.

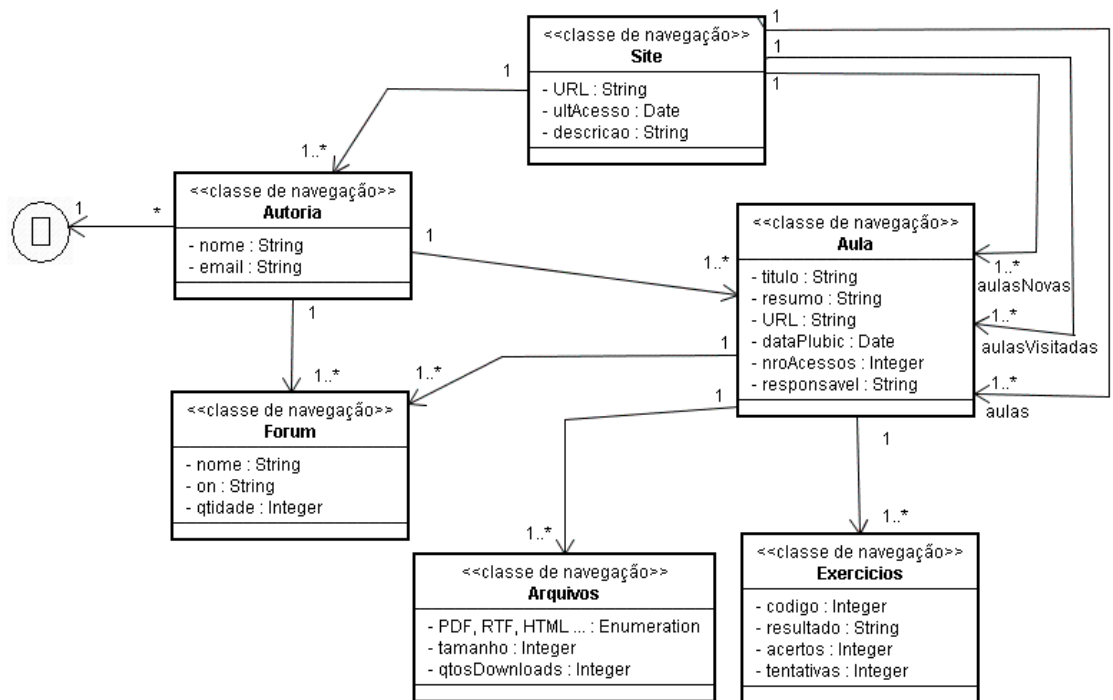


Fig. 13: Modelo de Hiperespaço de Navegação.

Na construção deste modelo, deve-se prestar atenção as seguintes regras propostas em Koch (2000):

- as classes do modelo conceitual são mapeadas para este modelo, sendo omitidas as classes não utilizadas pelo modelo de caso de uso. Por isso, foram omitidas as classes *Topicos* e *Nivel*, definidas anteriormente pelo modelo conceitual.
- os atributos do modelo conceitual são mantidos, sendo excluídos apenas os que não são relevantes neste modelo.
- as associações do modelo conceitual são mantidas e novas associações são criadas, como as existentes entre *Site* e *Aula*, para representar a navegação direta.
- estas associações que foram adicionadas representam alguns cenários descritos no diagrama de casos de uso, como no exemplo de um professor que esteja acessando o ambiente de ensino e descubra a necessidade de publicar novas aulas.

O modelo de estrutura de navegação, especificado na Fig. 14 a seguir, não é um modelo UML puro, mas sim um modelo construído com uma concepção próxima da UML, pois adiciona os estereótipos criados por Koch [5] às suas classes para representar os elementos utilizados no processo de navegação. Este modelo é

construído através do uso de páginas principais, menus, *links* externos e listas de itens [23].

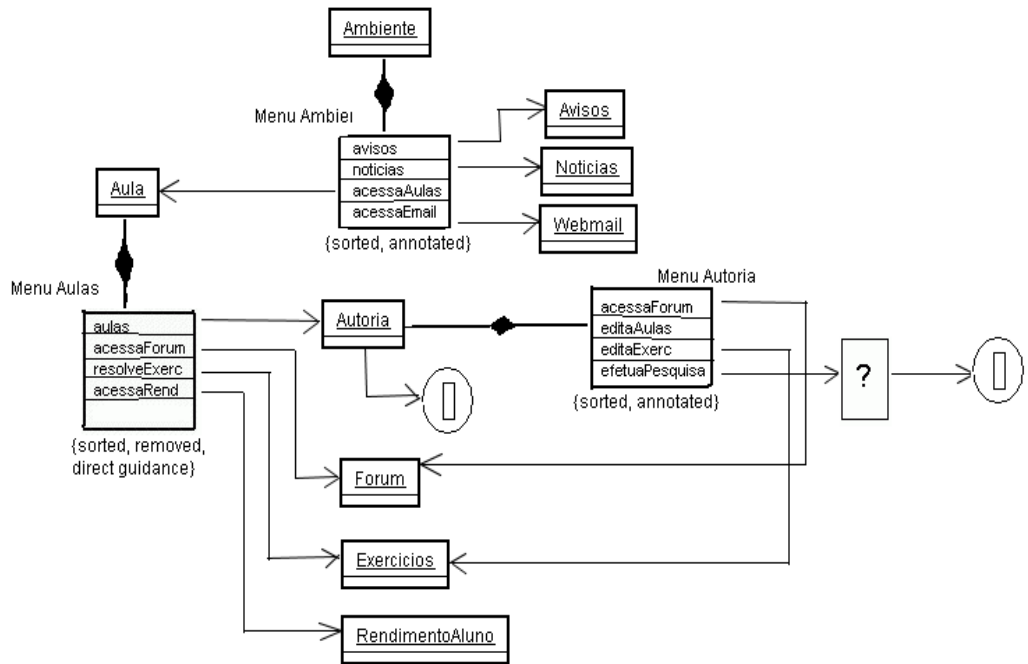


Fig. 14: Modelo de Estrutura de Navegação.

O modelo de estrutura de navegação traz uma série de alterações feitas no modelo de navegação anterior a fim de mostrar um caminho inicial na localização destes objetos dentro desta estrutura. Na figura acima, o objeto *Ambiente* possui uma relação de agregação por composição ao *Menu Ambiente*. Este último, com a sua lista de itens, representará mais tarde os *links* e nodos de uma página, que pode ligar a um ponto final do ambiente em questão, como *avisos* apontando para o objeto *Avisos*, ou para um novo nodo, como em *acessaAulas*.

Para demonstrar onde as adaptações podem ocorrer neste modelo de navegação são inseridos estereótipos referentes às técnicas que pode ser aplicadas, algumas delas descritas no capítulo 2.1.2, como a condução direta e anotação adaptativa, e outras que são [5]:

- ordenação (*sorted*): quando numa lista de menu, os seus itens são dispostos em ordem, indicando uma relevância para o usuário.
- remoção (*removed*): quando alguns itens podem ser removidos pelo sistema por não serem relevantes ao usuário.

- navegação passiva (*passive navigation*): define que a navegação é realizada pelo sistema diante de uma reação passiva do usuário.

5.1.5.1 MODELO DE ADAPTAÇÃO

O modelo de adaptação foi criado com este propósito: definir como e onde as adaptações podem ser aplicadas. Para isso, ele precisa elaborar uma série de regras para orientar-se no seu objetivo que é apresentar ao aluno desta aplicação um sistema personalizado levando em conta o seu modelo de usuário e ao mesmo tempo o modelo navegacional e de apresentação apropriado às suas características.

Ao elaborar os elementos desta modelagem, Koch [5] descreveu suas principais funcionalidades:

- regras – mostra como um conceito é encontrado, como as páginas são construídas e apresentadas ao aluno e, por fim, como o modelo de usuário é atualizado. Sendo classificadas em três tipos:
 - construção: encontram os conceitos
 - aquisição: atualizam o MU através de informações sobre o aluno
 - adaptação: conteúdo, *links* e/ou apresentação são adaptados
- ação do usuário – é uma classe que modela as ações do usuário percebidas pelo sistema. São encontrados três tipos de ações em sistemas hipermídias:
 - *browsing*: visita a um nodo hipermídia como consequência do acesso a um *link* específico.
 - *input*: formulário para envio de informações.
 - *user inactivity*: mecanismo utilizado para registrar um inatividade do usuário por um determinado intervalo de tempo.

Para especificar quais as regras de uma aplicação, Koch [5] resolveu definir algumas orientações importantes para o seu desenvolvimento que são:

- deixar claro todas as ações do usuário perante ao sistema.
- definir ao menos uma regra para cada classe de navegação do modelo de estrutura formando uma ou mais propriedades.
- estabelecer regras para adaptar a apresentação de acordo com as preferências escolhidas pelos usuários.

- representar através de diagramas de colaboração as situações mais complexas de adaptação, mostrando como a união das regras, das ações do usuário, dos seus atributos e dos objetos da apresentação contribuem para a construção do modelo de apresentação.

- verificar se cada a ação do usuário é usada numa condição de uma regra.
- verificar se os atributos de usuário são atualizados por pelo menos uma regra.

- verificar se a aplicação recursiva das regras pode garantir a finalização de todos os possíveis casos.

A seguir, é mostrado um conjunto de regras sobre as ações do usuário aluno para a aplicação desenvolvida neste trabalho:

- Regra 1 – O aluno deve acessar o sistema através do seu *login* e senha. Depois disso, o sistema efetua uma comparação dos seus dados com o MU, que está persistente numa base de dados, e verifica qual o nível e a aula que aquele aluno se encontra.
- Regra 2 - Se $\text{exercAula01} = 0$ então $\text{categ.MU} = \text{"iniciante"}$.
- Regra 3 - Se $\text{cod_Aula} < \text{cod_ProxAula}$ então $\text{linksProxAula} = \text{"desabilitado"}$.
- Regra 4 - Se $\text{categ.MU} = \text{"iniciante"}$ então $\text{corFundo} = \text{"branca"}$ e $\text{anotaLinks} = \text{verdadeiro}$.
- Regra 5 - Se $0 < \text{exercAula01} < 5$ então $\text{categ.MU} = \text{"intermediario"}$.
- Regra 6 - Se $\text{categ.MU} = \text{"intermediario"}$ então $\text{corFundo} = \text{"amarela"}$ e $\text{anotaLinks} = \text{falso}$.
- Regra 7 - Se $\text{categ.MU} = \text{"intermediario"}$ então alguns trechos das aulas são ocultados.
- Regra 8 – Se $\text{nroAula} > 5$ então $\text{categ.MU} = \text{"avancado"}$.
- Regra 9 - Se $\text{categ.MU} = \text{"avancado"}$ então $\text{corFundo} = \text{"azulEscuro"}$ e $\text{tamTexto} = \text{"resumido"}$.

Numa descrição textual, caso o aluno acesse pela primeira vez o ambiente, categoria iniciante, a cor de fundo é branca e ocorre a anotação dos *links* na área restrita de exibição das aulas. Depois que o aluno acessar a primeira aula e resolver os exercícios corretamente, a sua categoria é alterada para intermediário. Na

categoria intermediário, a cor de fundo passa a ser amarelo e os *links* que aparecem no texto ficam sem anotação. Caso o rendimento do aluno não seja satisfatório durante o intermediário, o *link* para as próximas aulas é desabilitado até o momento que a sua pontuação fique no nível desejável. Depois de alcançar a aula cinco, a categoria do aluno passa para avançado. A cor de fundo é azul escuro e o conteúdo pode ser mais resumido que as outras categorias.

Estas regras implementadas na aplicação demonstram como o sistema age diante de algumas situações do usuário, aumentando rapidamente o rendimento e o interesse do aluno em relação aos outros sistemas de ensino a distância que não utilizam técnicas de adaptação [1].

6. PROTÓTIPO DE UM SISTEMA HIPERMÍDIA ADAPTATIVO EDUCACIONAL

Como estudo de caso, foi escolhido um sistema para os alunos da oitava série do ensino fundamental que possibilite acessar os recursos de forma personalizada. Cada aluno possui uma interface de acordo com o seu modelo adaptativo. Nela, um ambiente de navegação e o conteúdo das aulas são apresentados. Conforme as interações do aluno com a ferramenta, todos os dados possíveis e úteis para uma posterior adaptação são armazenados no perfil deste aluno. Assim, os exercícios em cada aula formam a base para avaliação e para a alteração do nível do seu aprendizado.

O PSHAE (Protótipo de um Sistema Hipermedia Adaptativo Educacional) desenvolvido neste trabalho para demonstrar o funcionamento do conjunto de técnicas de um Sistema Hipermedia Adaptativo, utiliza o Modelo UWE para a modelagem da sua estrutura. Assim, a partir dos modelos descritos, o protótipo foi elaborado a fim de demonstrar algumas funcionalidades de um sistema que prevê uma adaptabilidade segundo o modelo de Munich.

No desenvolvimento do PSHAE foi definido que apenas a visão do ator UserMatriculado seria passada ao sistema para as posteriores adaptações de acordo com o seu modelo de usuário. Por apresentar uma ampla quantidade de técnicas de modelagem, as mais importantes e de entendimento simples, foram aplicadas às ações deste usuário diante do sistema, para deixar bem claro que o seu uso já evidencia a separação entre os SHA e os sistemas estáticos de EAD existentes no mercado.

Assim, ao listar os conceitos da disciplina, o aluno só pode seguir adiante caso resolva os exercícios da respectiva aula em que se encontra. Este controle é efetivado por um arquivo XML que contém a seqüência dos tópicos numa estrutura hierárquica e, através da API DOM em conjunto com a linguagem de *script* PHP, é feita uma comparação com MU para propagar as adaptações pelos seus diferentes níveis: iniciante, intermediário e avançado. Com isso, existe primeiro a preocupação em como acessar e modelar a estrutura hierárquica do XML e depois como apresentar os resultados aos usuários, que neste caso são os alunos da disciplina

de matemática, mas que também poderiam ser clientes de um portal de *e-commerce* ou leitores de um *web site* de notícias com os seus casos particulares de adaptação.

6.1 METODOLOGIA APLICADA

Na realização do protótipo foram utilizadas tecnologias *Open Source* e de fácil aquisição. Quanto ao local de funcionamento do aplicativo, foi utilizado o servidor *web* do Departamento de Informática e Estatística da Universidade Federal de Santa Catarina, disponibilizado aos alunos através do SSH (*Secure Transfer Protocol*). Este servidor pode atuar tanto como um repositório de arquivos como de aplicações. Nele, estão configurados as linguagens de programação e um banco de dados para a persistência do modelo de usuário, suficientes para a construção do aplicativo que são: PHP versão 4.4.0, XML (descrito no capítulo 4), DOM/XML API e o banco de dados Mysql versão 4.1.14.

Para a geração dos modelos da UWE utilizou-se a JUDE (*Java and UML Developer Environment*), ferramenta para a criação de diagramas UML, implementada em Java. Com um menu bem visual dos seus componentes, a modelagem da UWE foi implementada sem maiores problemas.

Com o PHP é feito o controle de acesso ao sistema através da função *session*. A cada acesso do aluno confirmado pela comparação do seu *login* e senha com os valores armazenados no Mysql, uma sessão é inicializada. Caso se confirme o não registro destes valores na sessão, o acesso é bloqueado, evitando que um aluno tenha como entrar no sistema sem que seja feita uma verificação do seu perfil. E, pelas funções *mysql_connect*, *mysql_select_db* e *mysql_query*, esta linguagem seleciona, atualiza e busca os dados armazenados no Mysql, que representa na sua totalidade o Modelo do Usuário.

Com o uso do PHP e a da API DOM foi possível percorrer a estrutura hierárquica de um arquivo XML na procura dos elementos *num*, *titulo* e *prerequisito*. Ao analisar o código descrito abaixo (ver o código completo no Anexo 2), na linha 1 aparece a versão do XML e o código de reconhecimento dos caracteres utilizados no documento. Na linha 2 é determinado o caminho para o arquivo DTD, onde é definida a estrutura hierárquica dos elementos e os seus respectivos tipos. Mas, é a partir da linha 4 que aparecem os elementos que são armazenados num *array* após a interação com a API DOM e PHP. Ao guardar os valores de cada um destes

elementos num *array* fica viável a sua comparação através dos dados provenientes do MU presentes no Mysql, para elaborar um algoritmo que mostre ao usuário apenas as informações que seguem as regras de adaptação.

```

1 <?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
2 <!DOCTYPE listaAulas SYSTEM
http://wwwexe.inf.ufsc.br/~kaminski/shae/xmIs/estruturaAulas.dtd">
3 <aula disciplina="1">
4     <conceito num="1" titulo="Conjuntos Numéricos" abrev="CjtoNum" >
5         <prerequisito idprerequis=""/>
6     </conceito>
7     <conceito num="2" titulo="Pi, o número mais famoso" abrev="NumPI" >
8         <prerequisito idprerequis="1"/>
9     </conceito>
10    <conceito num="2.1" titulo="O Papel da Trigonometria" abrev="PapTrig" >
11        <prerequisito idprerequis="1"/>
12    </conceito>
13    <conceito num="2.2" titulo="Arcos da Circunferência" abrev="ArcCircun" >
14        <prerequisito idprerequis="1"/>
15    </conceito>
16    <conceito num="2.3" titulo="O número Pi" abrev="PI" >
17        <prerequisito idprerequis="1"/>
18    </conceito>
19    <conceito num="2.4" titulo="Mudança de Unidades" abrev="MudUnid" xmlURL="02-
01" >
20        <prerequisito idprerequis="1"/>
21    </conceito>
        ■ ■ ■
23 </aula>

```

A estrutura deste XML, portanto, é construída e publicada de acordo com o arquivo DTD que segue com o seu respectivo código na figura 15. Nota-se que a informação é bem resumida neste arquivo, mas o seu poder de representação através de palavras reservadas e operadores serve bem ao propósito. Na linha 1, o elemento aula é declarado como o elemento raiz e possui como elemento filho o

conceito, que deve aparecer pelo menos uma vez no documento, devido ao operador +. Na linha 2, assim como na linha 6, são listados os atributos de aula e conceito respectivamente. Neste caso, todos os atributos são declarados como *string* (CDATA) e podem ser obrigatórios (*#REQUIRED*) ou opcionais (*#IMPLIED*). Por fim, todo o conjunto de declarações auxilia na confirmação se o documento XML está bem formado ou não, assegurando que a aplicação lidará apenas com documentos em conformidade com a DTD.

```

1 <!ELEMENT aula (conceito+)>
2 <!ATTLIST aula
3   disciplina CDATA #REQUIRED
4 >
5 <!ELEMENT conceito (requisito*,conceito*)>
6 <!ATTLIST conceito
7   num CDATA #REQUIRED
8   titulo CDATA #REQUIRED
9   abrev CDATA #REQUIRED
10  xmlURL CDATA #IMPLIED
11 >
12 <!ELEMENT requisito EMPTY>
13 <!ATTLIST requisito
14  idRequis CDATA #REQUIRED
15 >

```

Fig. 15: Código DTD.

No *link* Aulas, ver interface do protótipo em anexo, é visualizado como fica esta estrutura para o aluno após a interação do código PHP e da API DOM a partir do documento XML (ver a figura 16 abaixo). Sendo possível a leitura do documento (linhas 1 - 6), a API DOM reconhece o elemento principal e em seguida os seus elementos filhos (linhas 7 - 9) da figura 6.2. Em seguida, a API varre o arquivo durante um *loop*, pela função *foreach()*, a procura dos elementos *prerequisito* e ao encontrar uma ocorrência, busca o seu atributo (*idprerequis*) e armazena na variável *\$elemento*, em seguida incrementa um posição no *array* *\$prereq* e armazena esta mesma variável. Com isso, este *array* possuirá todas as ocorrências na ordem em que elas aparecem no documento. Desta mesma forma, é feito para buscar todos os elementos *conceito* e os seus atributos *titulo* e *num* (linhas 17 – 25).

```

1 if (!$dom =
2 domxml_open_file("http://wwwexe.inf.ufsc.br/~kaminski/shae/xmllistaAulas.xml"))
3     {
4         echo "Error while parsing the document\n";
5         exit;
6     }
7     $root = $dom->document_element();
8     $root = $dom->root();
9     $children = $root->children();
10
11     $prerequisitos = $dom->get_elements_by_tagname("prerequisito");
12     foreach($prerequisitos as $prerequisito) {
13         $element = $prerequisito;
14         $elemento = $element->get_attribute("idprerequis");
15         $prereq[$pre++] = "$elemento";
16     }
17     $conceitos = $dom->get_elements_by_tagname("conceito");
18     foreach($conceitos as $conceito) {
19         $element = $conceito;
20         $elemento01 = $element->get_attribute("titulo");
21         $titulos[$tit++] = "$elemento01";
22
23         $elemento02 = $element->get_attribute("num");
24         $nums[$num++] = "$elemento02";
25     }

```

Fig. 16: Código de interação PHP e API DOM.

Depois destas variáveis estarem armazenadas no *array*, por meio de código PHP a estrutura é percorrida e simultaneamente é feita a comparação com o MU, ficando transparente ao aluno apenas a estrutura de aulas disponível para ele naquele exato momento. Como neste protótipo, os arquivos referentes à disciplina são arquivos *html*, separa-se a estrutura do conteúdo ao ser utilizado o arquivo XML.

Observando a Figura 17, nota-se que os *links* a partir da aula 3 ficam bloqueados até a resolução correta dos exercícios da aula 2. E, para deixar claro o motivo desta ação, o sistema aplica a técnica de anotação nos *links* bloqueados. O aluno passa a não ter uma outra alternativa a não ser estudar mais aula atual ou resolver os exercícios corretamente. Entretanto, neste estudo de caso, optou-se pela

idéia de que cada assunto deveria possuir um pré-requisito, tornando-se obrigatório que o aluno percorra toda a árvore de conteúdo através de uma seqüência pré-definida pelo professor da disciplina.

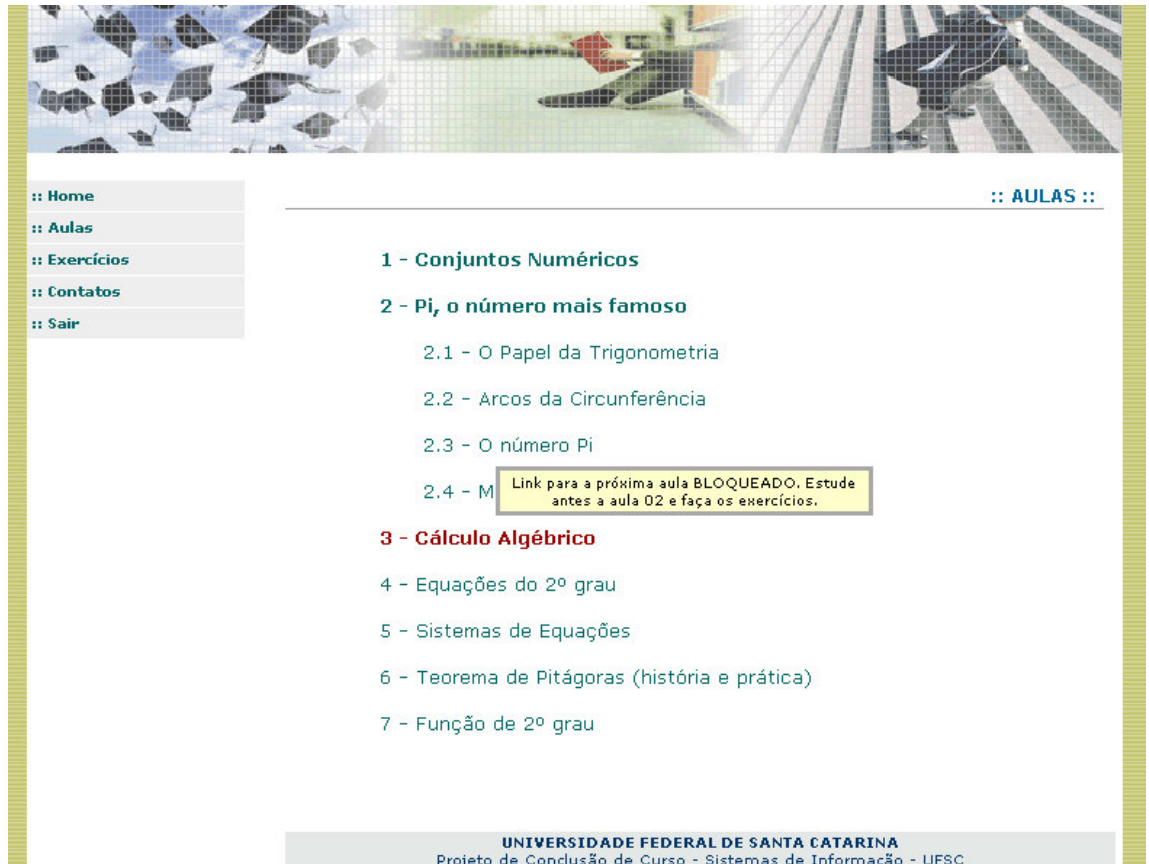


Fig. 17: Tela do PSHAE – Nível Intermediário.

6.2 TÉCNICAS DE ADAPTAÇÃO

No PSHAE através de uma arquitetura bem simples, foram aplicadas algumas técnicas adaptativas. Para isso, as regras de adaptação e as ações dos alunos precisam ser totalmente seguidas durante a navegação no sistema. Há a interação com o modelo do usuário, pois a cada regra ou ação do usuário efetuada, o MU é consultado para a confirmação do caminho a ser seguido pelo usuário.

Mas, antes de descrever as técnicas, é importante ressaltar as funcionalidades do protótipo perante às ações do usuário. O aluno ao ter contato pela primeira vez com o PSHAE se depara com uma tela de *login*. Isso é necessário,

para identificar qual o usuário e o seu respectivo modelo. Depois de confirmado o acesso, o aluno pode ler apenas o conteúdo da primeira aula. Para liberar a próxima aula, o aluno deve resolver os exercícios da aula anterior. Alcançando o número de questões corretas definidas pelo professor, ele pode dependendo de sua posição no ambiente, alterar ou não o seu nível de adaptação. No PSHAE o aluno já muda do nível iniciante para o intermediário na resolução correta dos exercícios da primeira aula. E, para o nível avançado, a partir da resolução dos exercícios da aula 5 (observar a tabela 1).

TABELA 1 – Seqüência das Aulas da Disciplina de Matemática no PSHAE

1. Conjuntos Numéricos
2. Pi, o número mais famoso
2.1 O Papel da Trigonometria
2.2 Arcos da Circunferência
2.3 O número Pi
2.4 Mudança de Unidades
3. Cálculo Algébrico
4. Equações do 2º grau
4.1 Resolução de Equações do 2º grau
4.2 Fórmula de BhÁskara
4.3 Propriedades
4.4 Equações do 2º grau Fracionárias
4.5 Equações Literias de 2º grau
5. Sistemas de Equações
5.1 Método da Adição
5.2 Método da Substituição
5.3 Método da Comparação
6. Teorema de Pitágoras (história e prática)
7. Função de 2º grau
7.1 Gráfico de uma Função de 2º grau
7.2 Raízes da Função de 2º grau
7.3 Concavidade da Parábola

É durante a apresentação do conteúdo da disciplina, que as primeiras adaptações são percebidas pelos alunos. O aluno pode definir o tamanho para a fonte do texto durante a exibição dos conteúdos. A cada mudança de nível a cor de fundo do PSHAE é diferente, começando como branca para o iniciante, amarelo para o intermediário e azul escuro para o avançado. Além disso, os tipos de fonte são alterados seguindo esta mesma idéia. A seqüência de aulas da disciplina também é modificada pelo método explicação requerida, que nada mais é do que ordenar o conteúdo seguindo os pré-requisitos definidos no arquivo de estrutura XML e compará-los com o MU.

Na Figura 18, é observada a principal idéia de funcionamento do protótipo. Os *layouts* são definidos levando em conta o modelo do usuário. No entanto, é ele que vai assumir o papel de maior importância ao definir onde e quando as adaptações irão surgir apesar da estrutura dos conceitos já ter sido pré-definida no arquivo XML. Portanto, as ações do usuário são consideradas sempre em primeiro plano.

Todos os recursos que podem ser aplicados pelas técnicas de hipermídia adaptativa devem levar em consideração este fato: o usuário, que neste caso é o aluno, deve ser o principal ator deste sistema e nunca pode ir de encontro ao seu MU. Por isso, pode se garantir as adaptações necessárias para que o protótipo propicie aos seus usuários a melhor forma de apresentação e navegação implementadas.

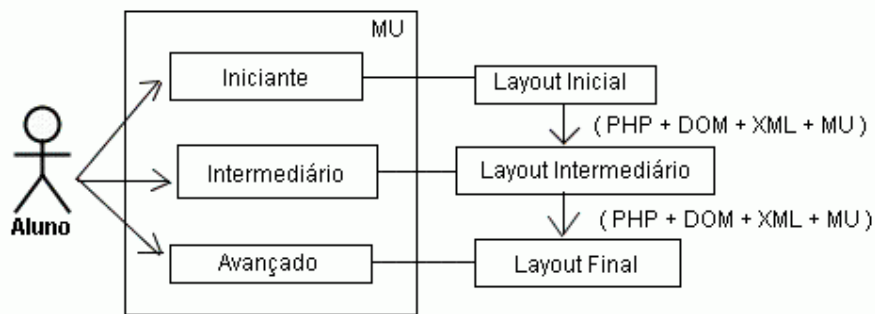


Fig. 18: Estrutura do PHSAE.

Já com relação ao ambiente de navegação, o aluno no nível iniciante possui algumas anotações sob os *links* a fim de prover uma orientação local (Figura 19). No nível intermediário, alguns trechos do conteúdo são ocultados para diminuir a sobrecarga de informações. Nesta arquitetura existem várias tecnologias que trabalham em conjunto para tornar a tarefa de navegação do aluno a mais simples possível, definindo uma boa conectividade dos *links* além da definição de conteúdos para os diferentes níveis de aprendizado.



Fig. 19: Tela do PSHAE – Nível Iniciante.

6.3 MODELO DE USUÁRIO

As informações dos alunos são armazenadas numa tabela no banco de dados *mysql*. Na tabela 2 abaixo, aparecem os atributos de um aluno que utiliza o PSHAE. Esta tabela passar a ser atualizada quando o aluno acessa o protótipo através do *login* e senha. Com estes dois atributos o sistema identifica o perfil completo do usuário e fica responsável por passar as informações necessárias para as adaptações segundo a estrutura definida.

TABELA 2 – Atributos da Tabela do Modelo de Usuário

Nome	Tipo	Descrição
user_id	integer	atributo de controle
nro_matr	integer	armazena o número de matrícula
user_name	varchar	nome do aluno
user_user	varchar	login
user_senha	varchar	senha
nivel	integer	0,1,2 (iniciante, intermediário e avançado)
dataPrimAcesso	date	data do primeiro acesso
dataMudNivel	date	data da mudança de nível
cod_aula	integer	armazena o código da aula atual
cod_exerc	integer	armazena o código do último exercício resolvido
dataAtualiz	date	data da última atualização da tabela

Com o atributo *cod_aula* é possível bloquear as aulas que ainda não podem ser acessadas pelo aluno. E, através do código do último exercício resolvido é

identificado o nível ao qual o aluno pertence naquele exato momento. Com a verificação destes dados persistentes no *mysql*, as ações dos usuários perante o sistema passam a serem vigiadas constantemente para que em certas partes do aplicativo comecem a surgir as personalizações de acordo com o seu nível de aprendizado.

Portanto, cada ator que interage com um sistema hipermídia adaptativo deve possuir um modelo de usuário que defina com clareza as suas preferências e as suas limitações perante a utilização de um aplicativo. O controle de todas as adaptações presentes é realizado pelo modelo de usuário. Sendo que os seus atributos assumem as responsabilidades de identificar o estado atual de um aluno e os possíveis caminhos que ele pode vir a seguir dependendo das suas ações e expectativas.

7. CONCLUSÃO

No desenvolvimento de aplicativos *web* direcionados para o ensino a distância, a aplicação das técnicas de hipermídias adaptativas mostrou-se muito eficaz. Este trabalho propôs a elaboração de um protótipo seguindo a modelagem UWE, que facilita o início do planejamento de projetos nesta área, orientando o projetista através de modelos UML referentes às camadas do modelo de referência de Munich. Além disso, foi realizada uma pesquisa com relação a outros dois modelos de referência: o Dexter e o AHAM, sendo o modelo de Munich o escolhido para compor o núcleo do protótipo.

Após a visualização da estrutura do sistema, a implementação do protótipo foi executada num servidor *web* para alcançar os resultados esperados que foram: a personalização do conteúdo e a melhora na navegação. Para isso, o uso de XML como forma de estruturação do conteúdo, possibilitou que as aulas respeitassem esta estrutura, sem a inclusão de nenhum controle nos seus arquivos em *html*.

O estudo da UWE permitiu visualizar a complexidade da aplicação dos conceitos de hipermídia adaptativos. A definição dos atores envolvidos na utilização do PSHAE foi decisiva para se decidir pela implementação das regras e ações de apenas um ator. Pois, a complexidade de desenvolvimento aumenta consideravelmente a medida que mais atores participam do sistema, sendo necessário a duplicação dos esforços em relação aos cuidados que deve se ter com o cruzamento das diversas ações executadas em conjunto pelos atores. Assim, apenas o ator *UserMatriculado* representou as ações de um aluno da disciplina de matemática e partir daí foram elaboradas as regras e as demais adaptações escolhidas para o estudo de caso.

A disciplina de matemática, utilizada como estudo de caso, foi escolhida com o intuito de propor um diferencial no material didático que o professor utiliza como ferramenta de ensino. Como o acesso à internet tem demonstrado um crescimento gradual, os alunos teriam mais este meio disponível para os seus estudos, e o PSHAE se porta como um auxílio ao ensino presencial. Além disso, o professor teria como verificar o andamento da disciplina e propor novos exercícios caso o aluno tivesse dificuldades em algum contexto. Assim, com os SHAEs é possível adaptar o sistema de acordo com o perfil do aluno, partindo do princípio que os usuários são

diferentes e, por isso, necessitam de uma personalização que se enquadre às suas necessidades.

Por fim, os ambientes de ensino a distância devem seguir uma modelagem que proporcione o maior número de ferramentas para que os seus usuários fiquem satisfeitos com os resultados e colaborem para o enriquecimento das mesmas. Os Sistemas Hiperfídias Adaptativos Educacionais ou SHAEs estão na busca destes objetivos e a continuação das pesquisas nesta área é de suma importância para o avanço dos programas educacionais na *web*, trazendo mais recursos para o aumento da aceitação de alunos e professores por esta tecnologia.

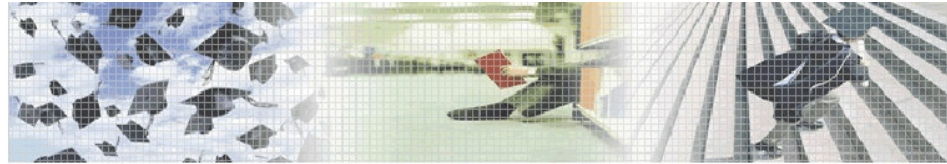
7.1 TRABALHOS FUTUROS

As regras descritas no trabalho em conjunto com a aplicação de novas tecnologias existentes na *web*, como sugestão o AJAX (*Asynchronous Javascript And XML*), permitiria uma resposta do sistema perante às ações dos usuários mais eficiente, devido a esta tecnologia possibilitar a visualização dos resultados sem a necessidade de um *refresh* no *browser*. Com isso, o aluno poderia realizar várias tarefas de modo assíncrono, ou seja, treinar os seus conhecimentos ao mesmo tempo em que estivesse lendo um arquivo numa página, não sendo necessário o envio do mesmo a outro local.

Já quanto aos exercícios, no PSHAE eles são utilizados para definir a passagem de um aluno para a próxima aula, sendo os mesmos responsáveis pela comprovação do nível de aprendizado. Por isso, a idéia seria elaborar a sua geração de forma automática através da utilização de redes neurais artificiais, enriquecendo o processo de aprendizagem do aluno, que não teria mais como prever o enunciado ou as respostas dos exercícios como acontece no protótipo desenvolvido.

Anexo 1 – Interfaces do PSHAE

Interface de Login e da Área Restrita do Protótipo de Sistema Hiperídia Educativa



:: PSHAE - PROTÓTIPO DE SISTEMA HIPERMÍDIA ADAPTATIVO EDUCACIONAL::

USUÁRIO:

SENHA:

UNIVERSIDADE FEDERAL DE SANTA CATARINA
Projeto de Conclusão de Curso - Sistemas de Informação - UFSC



- [:: Home](#)
- [:: Aulas](#)
- [:: Exercícios](#)
- [:: Contatos](#)
- [:: Sair](#)

>> SEJA BEM VINDO !!!

APRESENTAÇÃO

Aluno (a): Adriano da Silva - nº 54453453



Protótipo desenvolvido como
Trabalho de Conclusão de Curso

O PSHAE - **Protótipo de Sistema Hiperídia Adaptativo Educativa**, visa desenvolver ferramentas relacionadas com **Sistemas Hiperídias Adaptativas**. Ou seja, proporcionar ao aluno da disciplina de matemática, um ambiente personalizado e adaptável às suas necessidades.

A Equipe.

UNIVERSIDADE FEDERAL DE SANTA CATARINA
Projeto de Conclusão de Curso - Sistemas de Informação - UFSC

Anexo 2 – Arquivo XML – Estrutura dos Conceitos

Documento XML que demonstra a estrutura hierárquica dos conceitos expostos na disciplina de Matemática utilizada como Estudo de Caso

```

<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<!DOCTYPE listaAulas SYSTEM
"http://wwwexe.inf.ufsc.br/~kaminski/shae/xmls/estruturaAulas.dtd">
<aula disciplina="1">
  <conceito num="1" titulo="Conjuntos Numéricos" abrev="CjtoNum" >
    <prerequisito idprerequis=""/></conceito>
  <conceito num="2" titulo="Pi, o número mais famoso" abrev="NumPI" >
    <prerequisito idprerequis="1"/></conceito>
  <conceito num="2.1" titulo="O Papel da Trigonometria" abrev="PapTrig" >
    <prerequisito idprerequis="1"/></conceito>
  <conceito num="2.2" titulo="Arcos da Circunferência" abrev="ArcCircun" >
    <prerequisito idprerequis="1"/></conceito>
  <conceito num="2.3" titulo="O número Pi" abrev="PI" >
    <prerequisito idprerequis="1"/> </conceito>
  <conceito num="2.4" titulo="Mudança de Unidades" abrev="MudUnid"
xmlURL="02-01" >
    <prerequisito idprerequis="1"/></conceito>
  <conceito num="3" titulo="Cálculo Algébrico" abrev="CalcAlg" >
    <prerequisito idprerequis="2"/></conceito>
  <conceito num="4" titulo="Equações do 2º grau" abrev="EquaSegGrau" >
    <prerequisito idprerequis="3"/></conceito>
    <conceito num="4.1" titulo="Resolução de Equações do 2º grau"
abrev="EquaSegGrau" >
      <prerequisito idprerequis="3"/></conceito>
    <conceito num="4.2" titulo="Fórmula de BhÁskara" abrev="EquaSegGrau" >
      <prerequisito idprerequis="3"/></conceito>
    <conceito num="4.3" titulo="Propriedades" abrev="Propriedades" >
      <prerequisito idprerequis="3"/></conceito>
    <conceito num="4.4" titulo="Equações do 2º grau Fracionárias"
abrev="EquaSegGrauFrac" >

```

```

    <prerequisito idprerequis="3"/></conceito>
    <conceito num="4.5" titulo="Equações Literias de 2º grau"
abrev="EquaSegGrauLit" >
    <prerequisito idprerequis="3"/></conceito>
    <conceito num="5" titulo="Sistemas de Equações" abrev="SistEquac">
    <prerequisito idprerequis="4"/></conceito>
    <conceito num="5.1" titulo="Método da Adição" abrev="MetodAdic">
    <prerequisito idprerequis="4"/></conceito>
    <conceito num="5.2" titulo="Método da Substituição" abrev="SistEquac">
    <prerequisito idprerequis="4"/></conceito>
    <conceito num="5.3" titulo="Método da Comparação" abrev="SistEquac">
    <prerequisito idprerequis="4"/></conceito>
    <conceito num="6" titulo="Teorema de Pitágoras (história e prática)" abrev="Pitag" >
    <prerequisito idprerequis="5"/></conceito>
    <conceito num="7" titulo="Função de 2º grau" abrev="FuncSegGrau" >
    <prerequisito idprerequis="6"/></conceito>
    <conceito num="7.1" titulo="Gráfico de uma Função de 2º grau"
abrev="GrafFuncSegGrau" >
    <prerequisito idprerequis="6"/></conceito>
    <conceito num="7.2" titulo="Raízes da Função de 2º grau"
abrev="raizFuncSegGrau" >
    <prerequisito idprerequis="6"/></conceito>
    <conceito num="7.3" titulo="Concavidade da Parábola"
abrev="ParabolaSegunGrau" >
    <prerequisito idprerequis="6"/>
    </conceito>

</aula>

```

REFERÊNCIAS

1. Portal Inclusão Digital. **Inclusão digital: algumas promessas e muitos desafios.** Artigo publicado na Revista Soluções sobre alguns desafios a serem enfrentados na promoção da Inclusão Digital no Brasil, Março de 2005. Disponível por WWW em: <http://www.idbrasil.gov.br/docs_prog_gesac/artigos_entrevistas/cidec1>. Acesso em: 15 jul. 2005.
2. Sistema Nacional de Avaliação da Educação Básica (Saeb). **Qualidade da Educação: uma Nova Leitura do Desempenho dos Estudantes da 8ª Série do Ensino Fundamental,** Dezembro de 2003. Disponível por WWW em: <http://www.inep.gov.br/download/cibec/2003/saeb/qualidade_educa.pdf>. Acesso em: 21 jun. 2005.
3. AMARAL, Marília A.. **Organização e Armazenamento de Conteúdo Instrucional no Ambiente AdaptWeb Utilizando XML.** Mestrado. 2002. UFRGS.
4. OLIVEIRA, José M. Parente; FERNANDES, Clovis Torres. **Arquitetura de Adaptação em Sistemas Hiperídia Adaptativos Educacionais.** XIII Simpósio Brasileiro de Informática na Educação - SBIE. São Leopoldo: UNISINOS, 2002.
5. KOCH, Nora Parcus de. **Software Engineering for Adaptive Hypermedia Systems.** Tese - PhD. 2000. Universität München, Alemanha.
6. PALAZZO, L. A. M. **Modelos Proativos para Hiperídia Adaptativa.** Tese. 1999. UFRGS.
7. BRUSILOVSKY, Peter. **Methods and techniques of adaptive hypermedia.** User Modeling and User Adapted Interaction. Kluwer academic publishers, 1996, v 6, n 2-3, pp 87-129.
8. HALASZ, Frank G.; SCHWARTZ, Mayer **The Dexter Hypertext Reference Model.** NIST Hypertext Standardization Workshop, Gaithersburg, MD, January 16-18, 1990.
9. DE BRA, P.; HOUBEN, G. J.; WU, H. **AHAM: A Dexter-based Reference Model for Adaptive Hypermedia.** In: Proceedings of 10th ACM Conference on Hypertext and hypermedia (Hypertext'99), Darmstadt, Germany, February 21 – 25, 1999, ACM Press, pp. 147-156.

10. AMARAL, Marília A.; FERREIRA, Cláudio L.; ZANCHETTI, Pedro; ULBRICHT, Vânia R. **Modelo de Referência em Sistemas Hiperídia Adaptativos**. In: Anais do Congresso Nacional de Ambientes Hiperídia para Aprendizagem (CONAHPA). Florianópolis, 2006.
11. KOCH, Nora P. **The Munich Reference Model for Adaptive Hypermedia**. 2nd International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, 2002.
12. KOCH, N.; WIRSING, M. **The Munich Reference Model for Adaptive Hypermedia Applications**. In: 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, May , pp. 213-222.
13. *Word Wide Web Consortium*. Disponível por WWW em: <http://www.w3.org/> . Acesso em: 03 mai. 2006.
14. ALMEIDA, M.. **Uma introdução ao XML , sua utilização na Internet e alguns conceitos complementares**. Ci. Inf., Brasília, v. 31, n. 2, maio/ago. 2002.
15. *XML Tutorials*. Disponível por WWW em: <http://www.w3schools.com/> . Acesso em: 13 mai. 2006.
16. *The Extensible Stylesheet Language Family*. Disponível por WWW em: <http://www.w3.org/Style/XSL/> . Acesso em: 03 mai. 2006.
17. Tod, Nick. **JavaServer pages: o guia do desenvolvedor** / Nick Todd, Mark Szolkowski; tradução de Edson Furmankiewicz. – Rio de Janeiro: Elsevier, 2003.
18. De Bra, P., Aerts, A., Berden, B., De Lange, B., Rousseau, B., Santic, T., Smits, D., Stash, N., **AHA! The Adaptive Hypermedia Architecture**. Proceedings of the ACM Hypertext Conference, Nottingham, UK, August 2003, pp. 81-84.
19. De Bra, P, A. T. M. Aerts, David Smits, Natalia Stash: **AHA! the next generation**. Hypertext: 2002: 21-22. Disponível por WWW em: <http://portal.acm.org/citation.cfm?doid=513338.513347>. Acesso em: 21 out. 2005.

20. KNAPP, A.; KOCH, N.; MOSER F.; ZHANG, G. **ArgoUWE: A CASE Tool for Web Applications**. First Int. Workshop on Engineering Methods to Support Information Systems Evolution (EMSISE'03), set. 2003.
21. IYENGAR, Sridhar. **XML Metadata Interchange (XMI) From UML Object Models to XML DTDs and Documents**. OMG TC Meeting, Cambridge, 1999.
22. M. Cannataro, A. Cuzzocrea, A. Pugliese. **XAHM: an Adaptive Hypermedia Model based on XML**. International Conference on Software Engineering and Knowledge Engineering (SEKE), Ischia, Italy, 2002. ACM, New York, USA, 2002.
23. AMARAL, Marília A. ORIUNDI, Walter. **Modelo de Referência em Sistemas Hiperídia Adaptativos** In: Ambientes Adaptativos. Editora Ciência Moderna. Rio de Janeiro 2006.

APÊNDICES

APÊNDICE A – CÓDIGO FONTE

```
<?php // Arquivo index.php ?>
<table width="755" border="0" align="center" cellpadding="0"
cellspacing="0" bgcolor="#FFFFFF">
  <tr>
    <td colspan="4">
      <?php include('header.php'); ?>    </td>
    </tr>
  <tr>
    <td width="120" valign="top">
      </td>
    <td width="32" align="center" valign="top">&nbsp;</td>
    <td width="581" align="center" valign="top"><br>
    <?php
      include('query.php');
      if( (isset($pagina)) and (file_exists($pagina)) ) {
        include ($pagina);
      } else {
        echo "<br><b>P&acutegina solicitada n&atilde;o
existente.<br><a href=\"javascript:history.back();\">Voltar</a></b>";
      }
    ?></td>
    <td width="22" align="center" valign="top">&nbsp;</td>
  </tr>
  <tr align="center" bgcolor="#E4E9E8">
    <td colspan="4" class="rodape"><b>UNIVERSIDADE FEDERAL DE SANTA
CATARINA</b><br><br>
    Projeto de Conclus&atilde;o de Curso - Sistemas de
Informa&ccedil;&atilde;o - UFSC <br></td>
  </tr>
</table>
```

```
<?php

// Arquivo query.php

@$categoria = $_REQUEST['categoria'];
@$sub = $_REQUEST['sub'];

if ($categoria == "" && !$sub) {
    $pagina = "login.php";
}

elseif ($categoria == "redireciona" && !$sub) {
    $pagina = "redireciona.php";
}

?>
```

```

<?php

// Arquivo redireciona.php

$login = $_POST["login"];
$senha = $_POST["senha"];

include "aCesso.php"; //Conecta com a nosso banco de dados MySQL

$sql_user = mysql_query("SELECT * FROM tb_user where user_user = '$login'
and user_senha = '$senha'")
or die("ERRO no comando SQL:".mysql_error());

if(mysql_num_rows($sql_user) == 0) {

echo "<br><br><b>Login ou Senha incorretos</b><br><br>"

<html><meta http-equiv=refresh
content=1;URL=http://wwwexe.inf.ufsc.br/~kaminski/shae/index.php></html>

";

}
else{

echo "<html><meta http-equiv=refresh
content=1;URL=http://wwwexe.inf.ufsc.br/~kaminski/shae/index2.php?categoria
=abertura></html>";
}

if(mysql_num_rows($sql_user) > 0) {

    $user_idx = mysql_result($sql_user,0,"user_id");
    $user_userx = mysql_result($sql_user,0,"user_user");

    include ("sessao.php");
    //Registra as variáveis na sessão
}

?>

```

```

<?

// Arquivo aCesso.php

$dbname="kaminski";
$usuario="kaminski";
$password="?????";

//1° passo - Conecta ao servidor MySQL
if(!($id = mysql_connect("localhost", $usuario, $password))) {
//mysql.capitulocriciuma.com.br
    echo "<p align=\"center\"><big><strong>Não foi possível estabelecer uma
conexão com o gerenciador MySQL. Favor Contactar o Administrador.

```

```

    </strong></big></p>";
    exit;
}

//2° passo - Seleciona o Banco de Dados
if(!($con=mysql_select_db($dbname,$id)) {
    echo " <p align=\"center\"><big><strong>Não foi possível estabelecer uma
conexão com o gerenciador MySQL. Favor Contactar o Administrador.
</strong></big></p>";
    exit;
}
?>

```

```

<?php

// Arquivo sessao.php

session_start("usuarios"); //Inicializa uma sessão
//session_register("user_idx","user_userx");
$_SESSION['user_idx'] = $user_idx;
$_SESSION['user_userx'] = $user_userx;

?>

```

```

<?php

// Arquivo index2.php

session_start("usuarios"); //Inicializa uma sessão
include ("ver_sessao.php");
include "aCesso.php"; //Conecta com o banco de dados MySQL

$sql_user = mysql_query("SELECT * FROM tb_user where user_id =
'$user_idx'")
    or die("ERRO no comando SQL:".mysql_error());

if(mysql_num_rows($sql_user) > 0) {

    $user_idx = mysql_result($sql_user,0,"user_id");
    $user_userx = mysql_result($sql_user,0,"user_user");
    $user_nivelx = mysql_result($sql_user,0,"nivel");

}

?>
<?php if ($user_nivelx == 2) { ?>
<body background="imagens/fundo.gif" leftmargin="0" topmargin="0"
marginwidth="0">
<?php
} elseif ($user_nivelx == 3) { ?>
<body background="imagens/fundo2.gif" leftmargin="0" topmargin="0"
marginwidth="0">
<?php }

```

```

?>
<table width="755" border="0" align="center" cellpadding="0"
cellspacing="0" bgcolor="#FFFFFF">
  <tr>
    <td colspan="4">
      <?php include('header2.php'); ?>    </td>
    </tr>
  <tr>
    <td width="120" valign="top">
      <?php include ('menu2.php'); ?>    </td>
    <td width="24" align="center" valign="top">&nbsp;   </td>
    <td width="595" align="center" valign="top"><br>
      <?php
        include('query2.php');
        if( (isset($pagina)) and (file_exists($pagina)) ) {
          include ($pagina);
        } else {
          echo "<br><b>P&agrave;gina solicitada n&atilde;o
existente.<br><br><a href=\"javascript:history.back();\">Voltar</a></b>";
        }
      ?></td>
    <td width="16" align="center" valign="top">&nbsp;   </td>
  </tr>
  <tr align="center" bgcolor="#E4E9E8">
    <td colspan="4" class="rodape"><b>UNIVERSIDADE FEDERAL DE SANTA
CATARINA</b><br><br>
      Projeto de Conclus&atilde;o de Curso - Sistemas de
Informa&ccedil;&atilde;o - UFSC <br></td>
  </tr>
</table>

```

```

<?php
// Arquivo ver_sessao.php

$user_idx=$HTTP_SESSION_VARS['user_idx'];
$user_userx=$HTTP_SESSION_VARS['user_userx'];

session_start("usuarios");
if(!(session_is_registered("user_idx") AND
session_is_registered("user_userx"))) {
  echo "<center><h1><b>Essa é uma área restrita.</b></h1></center>
  <html><meta http-equiv=refresh
content=1;URL=http://wwwexe.inf.ufsc.br/~kaminski/shae/index.php></html>

  ";
  exit;
}
?>

```

```

<?php // Arquivo menu2.php ?>

<link href="teste.css" rel="stylesheet" type="text/css"> <body
bgcolor="#FFFFFF" leftmargin="0" topmargin="0" marginwidth="0"
marginheight="0">
<br><table width="150" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td class="menu"><a href="?categoria=abertura">&nbsp;:: Home</a></td>
  </tr>
  <tr>
    <td class="menu"><a href="?categoria=aulas">&nbsp;:: Aulas</a></td>
  </tr>
  <tr>
    <td class="menu"><a href="?categoria=exercicios">&nbsp;::
Exercícios</a> </td>
  </tr>
  <tr>
    <td class="menu"><a href="?categoria=contatos">&nbsp;::
Contatos</a></td>
  </tr>
  <tr>
    <td class="menu"><a href="?categoria=sair">&nbsp;:: Sair</a></td>
  </tr>
</table>
<br>

```

```

<?php Arquivo aulas.php ?>

<script type="text/javascript" src="anota.js"></script>
<link href="formatacao.css" rel="stylesheet" type="text/css">
<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
<table width="99%" align="center" bgcolor="#FFFFFF">

  <tr>
    <td colspan="2" class="titulo">:: Aulas ::&nbsp;</td>
  </tr>
  <tr>
    <td colspan="2">&nbsp;</td>
  </tr>
  <tr>
    <td width="11%" class="texto3"><div align="center">
      <p><br>
      </p>
      <p>          <br>
      <br>
      <br>
      </p>
    </div></td>
    <td width="89%" class="texto3">

<?php

include ("inc_variaveisBD.php");

```

```

if (!$dom =
domxml_open_file("http://wwwexe.inf.ufsc.br/~kaminski/shae/xm1s/listaAulas.
xml")) {
    echo "Error while parsing the document\n";
    exit;
}

$root = $dom->document_element();

$root = $dom->root();

$children = $root->children();

$pre++;
$tit++;
$num++;
$url="?categoria=aula";

$prereq = array();
$titulos = array();
$num = array();

$numTop = array();
$titTop = array();

$prerequisitos = $dom->get_elements_by_tagname("prerequisito");

    foreach($prerequisitos as $prerequisito) {

        $element = $prerequisito;
        $elemento = $element->get_attribute("idprerequis");
        $prereq[$pre++] = "$elemento";

    }

$conceitos = $dom->get_elements_by_tagname("conceito");

    foreach($conceitos as $conceito) {

        $element = $conceito;
        $elemento01 = $element->get_attribute("titulo");
        $titulos[$tit++] = "$elemento01";

        $elemento02 = $element->get_attribute("num");
        $nums[$num++] = "$elemento02";

    }

$tit=0;
$pre=1;
$num=1;

for ($x=1; $x<=sizeof($prereq); $x++){

    if ($prereq[$x]<=$user_cod_aulax){

$tit++;

```



```

if(nav) skin=document.pup;
if(iex) skin=pup.style;
if(n_6) skin=document.getElementById("pup").style;

// park modifier
var yyy=-1000;

// capture pointer
if(nav) document.captureEvents(Event.MOUSEMOVE);
if(n_6) document.addEventListener("mousemove",get_mouse,true);
if(nav||iex) document.onmousemove=get_mouse;

// set dynamic coords
function get_mouse(e)
{
    var x,y;

    if(nav || n_6) x=e.pageX;
    if(iex) x=event.x+document.body.scrollLeft;

    if(nav || n_6) y=e.pageY;
    if(iex)
    {
        y=event.y;
        if(navigator.appVersion.indexOf("MSIE 4")==-1)
            y+=document.body.scrollTop;
    }

    if(iex || nav)
    {
        skin.top=y+yyy;
        skin.left=x+Xoffset;
    }

    if(n_6)
    {
        skin.top=(y+yyy)+"px";
        skin.left=x+Xoffset+"px";
    }
    nudge(x);
}

// avoid edge overflow
function nudge(x)
{
    var extreme,overflow,temp;

    // right
    if(iex) extreme=(document.body.clientWidth-popwidth);
    if(n_6 || nav) extreme=(window.innerWidth-popwidth);

    if(parseInt(skin.left)>extreme)
    {
        overflow=parseInt(skin.left)-extreme;
        temp=parseInt(skin.left);
        temp-=overflow;
        if(nav || iex) skin.left=temp;
        if(n_6) skin.left=temp+"px";
    }

    // left

```

```

if(parseInt(skin.left)<1)
{
    overflow=parseInt(skin.left)-1;
    temp=parseInt(skin.left);
    temp-=overflow;
    if(nav || iex) skin.left=temp;
    if(n_6)skin.left=temp+"px";
}
}

// write content & display
function popup(msg,bak)
{
    var content="<TABLE WIDTH='"+popwidth+"' BORDER='2'
    BORDERCOLOR="+bcolor+" CELLPADDING=2 CELLSPACING=0 "+ "BGCOLOR="+bak+"><TD
    ALIGN='center'><FONT COLOR="+fcolor+" FACE="+fface+"
    SIZE='3'>"+msg+"</FONT></TD></TABLE>";

    if(old)
    {
        alert(msg);
        return;
    }

    yyy=Yoffset;
    skin.width=popwidth;

    if(nav)
    {
        skin.document.open();
        skin.document.write(content);
        skin.document.close();
        skin.visibility="visible";
    }

    if(iex)
    {
        pup.innerHTML=content;
        skin.visibility="visible";
    }

    if(n_6)
    {
        document.getElementById("pup").innerHTML=content;
        skin.visibility="visible";
    }
}

// park content box
function kill()
{
    if(!old)
    {
        yyy=-1000;
        skin.visibility="hidden";
        skin.width=0;
    }
}
}

```

```
//-->
```

Arquivo listaAulas.xml

```
<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<!DOCTYPE listaAulas SYSTEM
"http://wwwexe.inf.ufsc.br/~kaminski/shae/xm1s/estruturaAulas.dtd">
<aula disciplina="1">
    <conceito num="1" titulo="Conjuntos Numéricos" abrev="CjtoNum" >
        <prerequisito idprerequis=""/>
    </conceito>
    <conceito num="2" titulo="Pi, o número mais famoso" abrev="NumPI" >
        <prerequisito idprerequis="1"/>
    </conceito>
    <conceito num="2.1" titulo="O Papel da Trigonometria" abrev="PapTrig"
    >
        <prerequisito idprerequis="1"/>
    </conceito>
    <conceito num="2.2" titulo="Arcos da Circunferência"
abrev="ArcCircun" >
        <prerequisito idprerequis="1"/>
    </conceito>
    <conceito num="2.3" titulo="O número Pi" abrev="PI" xmlURL="aula02-
01" >
        <prerequisito idprerequis="1"/>
    </conceito>
    <conceito num="2.4" titulo="Mudança de Unidades" abrev="MudUnid"
xmlURL="02-01" >
        <prerequisito idprerequis="1"/>
    </conceito>
    <conceito num="3" titulo="Cálculo Algébrico" abrev="CalcAlg" >
        <prerequisito idprerequis="2"/>
    </conceito>
    <conceito num="4" titulo="Equações do 2º grau" abrev="EquaSegGrau" >
        <prerequisito idprerequis="3"/>
    </conceito>
    <conceito num="4.1" titulo="Resolução de Equações do 2º grau"
abrev="EquaSegGrau" >
        <prerequisito idprerequis="3"/>
    </conceito>
    <conceito num="4.2" titulo="Fórmula de BhÁskara"
abrev="EquaSegGrau" >
        <prerequisito idprerequis="3"/>
```

```

    </conceito>
    <conceito num="4.3" titulo="Propriedades" abrev="Propriedades"
  >
    <prerequisito idprerequis="3"/>
  </conceito>

    <conceito num="4.4" titulo="Equações do 2º grau Fracionárias"
abrev="EquaSegGrauFrac" >
    <prerequisito idprerequis="3"/>
  </conceito>

    <conceito num="4.5" titulo="Equações Literias de 2º grau"
abrev="EquaSegGrauLit" >
    <prerequisito idprerequis="3"/>
  </conceito>

  <conceito num="5" titulo="Sistemas de Equações" abrev="SistEquac">
    <prerequisito idprerequis="4"/>
  </conceito>

    <conceito num="5.1" titulo="Método da Adição"
abrev="MetodAdic">
    <prerequisito idprerequis="4"/>
  </conceito>

    <conceito num="5.2" titulo="Método da Substituição"
abrev="SistEquac">
    <prerequisito idprerequis="4"/>
  </conceito>

    <conceito num="5.3" titulo="Método da Comparação"
abrev="SistEquac">
    <prerequisito idprerequis="4"/>
  </conceito>

  <conceito num="6" titulo="Teorema de Pitágoras (história e prática)"
abrev="Pitago" >
    <prerequisito idprerequis="5"/>

  </conceito>

  <conceito num="7" titulo="Função de 2º grau" abrev="FuncSegGrau" >
    <prerequisito idprerequis="6"/>
  </conceito>

    <conceito num="7.1" titulo="Gráfico de uma Função de 2º grau"
abrev="GrafFuncSegGrau" >
    <prerequisito idprerequis="6"/>
  </conceito>

    <conceito num="7.2" titulo="Raízes da Função de 2º grau"
abrev="raizFuncSegGrau" >
    <prerequisito idprerequis="6"/>
  </conceito>

    <conceito num="7.3" titulo="Concavidade da Parábola"
abrev="ParabolaSegunGrau" >
    <prerequisito idprerequis="6"/>

```

```

        </conceito>
</aula>

```

Arquivo estruturaAulas.dtd

```

<!ELEMENT aula (conceito+)>
  <!ATTLIST aula
    disciplina CDATA #REQUIRED
  >

  <!ELEMENT conceito (requisito*,conceito*)>
  <!ATTLIST conceito
    num CDATA #REQUIRED
    titulo CDATA #REQUIRED
    abrev CDATA #REQUIRED
    xmlURL CDATA #IMPLIED
  >

  <!ELEMENT requisito EMPTY>
  <!ATTLIST requisito
    idRequis CDATA #REQUIRED
  >

```

```

<?php
// Arquivo exercicioAula01.php

include ("ver_sessao.php");
?>
<LINK
href="http://wwwexe.inf.ufsc.br/~kaminski/shae/aulas/formatacao.css"
type=text/css
rel=stylesheet>

<style type="text/css">
<!--
.style1 {font-family: Verdana, Arial, Helvetica, sans-serif}
-->
</style>
<form action="?categoria=registra" method="post" enctype="multipart/form-
data" name="form" id="form">

  <table width="600" border="0" align="center" cellpadding="0"
cellspacing="2" bgcolor="#FFFFFF">
  <tr>
    <td><div align="center">
      <p class="titulo"><br>
        Exercícios Aula 01 - Conjuntos Numéricos <br>
      <br>
    </p>
    </div></td>
  </tr>
  <tr>
    <td><table width="600" align="center" cellpadding="0"
cellspacing="2">

```

```

        <tr>
          <td width="604" bgcolor="#FFFFFF" class="titulo2"><label>1 -
Asinale a alternativa correta: <br>
          <strong><font size="2" face="Verdana, Arial, Helvetica,
sans-serif">
          <input type="hidden" name="aula" size="30" value="aula01"
/>
          <input type="hidden" name="resp01" size="30" value="a" />
</font></strong><span class="texto"><br>
          a)
          <input name="exerc01" type="radio" value="a" >
          Todo n&uacute;mero inteiro &eacute; um n&uacute;mero real.
<br>
          b)
          <input name="exerc01" type="radio" value="b">
          Nem todo n&uacute;mero inteiro &eacute; um n&uacute;mero real.
<br>
          c)
          <input name="exerc01" type="radio" value="c">
          5 / 2 &eacute; um n&uacute;mero inteiro. <br>
          d)
          <input name="exerc01" type="radio" value="d">
nenhuma alternativa. </span><br />
<br />
2 - Asinale a alternativa correta: <br />
<input type="hidden" name="resp02" size="30" value="b" />
<br />
<span class="texto">a)
<input name="exerc02" type="radio" value="a">
Os n&uacute;meros decimais peri&ocedil;dicos n&atilde;o s&atilde;o
racionais. <br />
b)
<input name="exerc02" type="radio" value="b" />
A uni&atilde;o do conjunto dos n&uacute;meros irracionais com o dos
racionais forma o grupo de reais. <br />
c)
<input name="exerc02" type="radio" value="c" />
o n&uacute;mero pi &eacute; um n&uacute;mero inteiro. <br />
d)
<input name="exerc02" type="radio" value="d" />
nenhuma alternativa. </span><br />
<br />
3 - Os n&uacute;meros irracionais s&atilde;o aqueles: <br />
<input type="hidden" name="resp03" size="30" value="c" />
<br />
<span class="texto">a)
<input name="exerc03" type="radio" value="a" />
que podem ser expressos na forma a/b, com a e b inteiros e b
diferente de 0. <br />
b)
<input name="exerc03" type="radio" value="b" />
compostos pela a uni&atilde;o do conjunto dos n&uacute;meros reais com o
dos racionais. <br />
c)
<input name="exerc03" type="radio" value="c" />
que podem ser compostos por d&iacute;zimas infinitas n&atilde;o
peri&ocedil;dicas. <br />
d)
<input name="exerc03" type="radio" value="d" />
nenhuma alternativa. </span><br />
<br />

```



```

if (($exerc01 == $resp01) and ($resp01!="")){
    $result01 = "sim";
    $acerto++;
}
else{
    $result01 = "não";
}

if (($exerc02 == $resp02) and ($resp02!="")){
    $acerto++;
    $result02 = "sim";
}
else{
    $result02 = "não";
}

if (($exerc03 == $resp03) and ($resp03!="")){
    $acerto++;
    $result03 = "sim";
}
else{
    $result03 = "não";
}

if (($exerc04 == $resp04) and ($resp04!="")){
    $acerto++;
    $result04 = "sim";
}
else{
    $result04 = "não";
}

if ($aula=="aula01") {

if ($acerto == 4 and $user_cod_exercx == 0) {
echo "$user_namex, meus parabéns!!!!";
$nivel="2";
$cod_exerc= "1";
$cod_aula= "1";
$sql = mysql_query("UPDATE tb_user SET
nivel='$nivel',cod_aula='$cod_aula',cod_exerc='$cod_exerc',
dataMudNivel='$datahora', dataAtualizado='$datahora' WHERE
user_id='$user_idx'")
    or die("Erro no comando SQL:".mysql_error());
}
elseif ($acerto < 4) {
echo "Estude mais a aula 01. <b>Vc não pode passar adiante</b> !!!!";
}
elseif ($user_cod_exercx >= 1) {
echo "<b>Vc já fez este exercício</b> !!!!";
}
}

elseif(($aula=="aula02")){
if ($acerto == 3 and $user_cod_exercx == 1) {
echo "$user_namex, meus parabéns!!!!";
$nivel="2";
$cod_exerc= "2";
$cod_aula= "2";
$sql = mysql_query("UPDATE tb_user SET
nivel='$nivel',cod_aula='$cod_aula',cod_exerc='$cod_exerc',

```



```

dataMudNivel='$datahora', dataAtualizado='$datahora' WHERE
user_id='$user_idx'")
        or die("Erro no comando SQL:".mysql_error());

}
elseif ($acerto < 3) {
echo "Estude mais a aula 02. <b>Vc não pode passar adiante</b> !!!!!";
}
elseif ($user_cod_exercx >= 2) {
echo "<b>Vc já fez este exercício</b> !!!!!";
}
}

elseif(($aula=="aula03")){
if ($acerto == 2 and $user_cod_exercx == 2) {
echo "$user_namex, meus parabéns!!!!";
$nivel="2";
$cod_exerc= "3";
$cod_aula= "3";
$sql = mysql_query("UPDATE tb_user SET
nivel='$nivel',cod_aula='$cod_aula',cod_exerc='$cod_exerc',
dataMudNivel='$datahora', dataAtualizado='$datahora' WHERE
user_id='$user_idx'")
        or die("Erro no comando SQL:".mysql_error());

}
elseif ($acerto < 2) {
echo "Estude mais a aula 03. <b>Vc não pode passar adiante</b> !!!!!";
}
elseif ($user_cod_exercx >= 3) {
echo "<b>Vc já fez este exercício</b> !!!!!";
}
}

elseif(($aula=="aula04")){
if ($acerto == 3 and $user_cod_exercx == 3) {
echo "$user_namex, meus parabéns!!!!";
$nivel="2";
$cod_exerc= "4";
$cod_aula= "4";
$sql = mysql_query("UPDATE tb_user SET
nivel='$nivel',cod_aula='$cod_aula',cod_exerc='$cod_exerc',
dataMudNivel='$datahora', dataAtualizado='$datahora' WHERE
user_id='$user_idx'")
        or die("Erro no comando SQL:".mysql_error());

}
elseif ($acerto < 3) {
echo "Estude mais a aula 04. <b>Vc não pode passar adiante</b> !!!!!";
}
elseif ($user_cod_exercx >= 4) {
echo "<b>Vc já fez este exercício</b> !!!!!";
}
}

elseif(($aula=="aula05")){
if ($acerto == 2 and $user_cod_exercx == 4) {
echo "$user_namex, meus parabéns!!!!";
$nivel="3";
$cod_exerc= "5";
$cod_aula= "5";
}
}

```

```

$ssql = mysql_query("UPDATE tb_user SET
nivel='$nivel',cod_aula='$cod_aula',cod_exerc='$cod_exerc',
dataMudNivel='$datahora', dataAtualizado='$datahora' WHERE
user_id='$user_idx'")
        or die("Erro no comando SQL:".mysql_error());

}

elseif ($acerto < 2) {
echo "Estude mais a aula 05. <b>Vc não pode passar adiante</b> !!!!!";
}
elseif ($user_cod_exercx >= 5) {
echo "<b>Vc já fez este exercício</b> !!!!!";
}
}

elseif(($aula=="aula06")){
if ($acerto == 1 and $user_cod_exercx == 5) {
echo "$user_name, meus parabéns!!!!";
$nivel="3";
$cod_exerc= "6";
$cod_aula= "6";
$ssql = mysql_query("UPDATE tb_user SET
nivel='$nivel',cod_aula='$cod_aula',cod_exerc='$cod_exerc',
dataMudNivel='$datahora', dataAtualizado='$datahora' WHERE
user_id='$user_idx'")
        or die("Erro no comando SQL:".mysql_error());

}
elseif ($acerto < 1) {
echo "Estude mais a aula 06. <b>Vc não pode passar adiante</b> !!!!!";
}
elseif ($user_cod_exercx >= 6) {
echo "<b>Vc já fez este exercício</b> !!!!!";
}
}

elseif(($aula=="aula07")){
if ($acerto == 3 and $user_cod_exercx == 6) {
echo "$user_name, meus parabéns!!!!";
$nivel="3";
$cod_exerc= "7";
$cod_aula= "7";
$ssql = mysql_query("UPDATE tb_user SET
nivel='$nivel',cod_aula='$cod_aula',cod_exerc='$cod_exerc',
dataMudNivel='$datahora', dataAtualizado='$datahora' WHERE
user_id='$user_idx'")
        or die("Erro no comando SQL:".mysql_error());

}
elseif ($acerto < 3) {
echo "Estude mais a aula 07. <b>Vc não pode passar adiante</b> !!!!!";
}
elseif ($user_cod_exercx >= 7) {
echo "<b>Vc já fez este exercício</b> !!!!!";
}
}

?>

```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<LINK
href="http://wwwexe.inf.ufsc.br/~kaminski/shae/aulas/formatacao.css"
type=text/css
rel=stylesheet>
</head>

<body>
<p>&nbsp;</p>
<table width="235" border="1" align="center">
  <tr>
    <td width="64" class="texto3"><div
align="center">Exerc&iacute;cio</div></td>
    <td width="67" class="texto3"><div align="center">Asinalada</div></td>
    <td width="82" class="texto3"><div align="center">Correta ? </div></td>
  </tr>
  <tr>
    <td><div align="center" class="texto3"><strong>01</strong></div></td>
    <td><div align="center" class="texto3"><?php echo
"$exerc01";?></div></td>
    <td><div align="center" class="texto3">
      <?php
        echo "$result01";
      ?>
    </div></td>
  </tr>
  <?php if ($exerc02 != ""){?>
  <tr>
    <td><div align="center" class="texto3"><strong>02</strong></div></td>
    <td><div align="center" class="texto3"><?php echo
"$exerc02";?></div></td>
    <td><div align="center" class="texto3">
      <?php echo "$result02";
    ?>
    </div></td>
  </tr> <?php } ?>
  <?php if ($exerc03 != ""){?>
  <tr>
    <td><div align="center"><strong>03</strong></div></td>
    <td><div align="center"><span class="texto3"><?php echo
"$exerc03";?></span></div></td>
    <td><div align="center"><span class="texto3"><?php echo "$result03";
?></span></div></td>
  </tr> <?php } ?>
  <?php if ($exerc04 != ""){?>
  <tr>
    <td><div align="center"><strong>04</strong></div></td>
    <td><div align="center"><span class="texto3"><?php echo
"$exerc04";?></span></div></td>
    <td><div align="center"><span class="texto3"><?php echo "$result04";
?></span></div></td>
  </tr> <?php } ?>
</table>
<br />
<br />
<br />

```

```
<br />  
</body>  
</html>
```

```
<?php  
  
// Arquivo inc_variaveisBD.php  
  
include ("ver_sessao.php");  
include "aCesso.php"; //Conecta com a nosso banco de dados MySQL  
  
$sql_user = mysql_query("SELECT * FROM tb_user where user_id =  
'$user_idx'")  
    or die("ERRO no comando SQL:".mysql_error());  
  
if(mysql_num_rows($sql_user) > 0) {  
  
    $user_idx = mysql_result($sql_user,0,"user_id");  
    $user_userx = mysql_result($sql_user,0,"user_user");  
    $nro_matrx = mysql_result($sql_user,0,"nro_matr");  
    $user_namex = mysql_result($sql_user,0,"user_name");  
    $user_emailx = mysql_result($sql_user,0,"user_email");  
    $user_idadx = mysql_result($sql_user,0,"idade");  
    $user_nivelx = mysql_result($sql_user,0,"nivel");  
    $user_dataPrimAcessox = mysql_result($sql_user,0,"dataPrimAcesso");  
    $user_dataMudNivelx = mysql_result($sql_user,0,"dataMudNivel");  
    $user_rendimentox = mysql_result($sql_user,0,"rendimento");  
    $user_cod_aulax = mysql_result($sql_user,0,"cod_aula");  
    $user_cod_exercx = mysql_result($sql_user,0,"cod_exerc");  
    $user_dataAtualizadox = mysql_result($sql_user,0,"dataAtualizado");  
}  
  
?>
```

APÊNDICE B – ARTIGO

Sistema Hipermissão Adaptativo: a sua utilização para o ensino da matemática**Douglas Kaminski¹**

¹Universidade Federal de Santa Catarina – Discente do Curso de Sistemas de Informação
Campus Trindade – UFSC – Florianópolis – SC – Brasil

kaminskidk@gmail.com

***Resumo.** Os usuários que navegam pela web estão cada vez mais ansiosos por tecnologias que permitam uma personalização das suas aplicações, diminuindo a sobrecarga cognitiva e a desorientação durante o processo de navegação. Nos ambientes de ensino a distância a procura já é grande por aplicações que reconheçam estes anseios dos usuários para não perder o interesse do seu maior cliente que é o aluno. Este trabalho, está na busca pela realização destas idéias através do uso das técnicas de hipermissão adaptativa. É apresentada uma proposta de implementação de um protótipo modelado com base nos preceitos da UWE (UML-based Web Engineering), que especifica a construção de seis modelos: diagramas de casos de uso, modelo conceitual, modelo de usuário, modelo de navegação, modelo de apresentação e modelo de adaptação.*

***Abstract.** The users who sail for web are each time more anxious for technologies that they allow a personalization of its applications, diminishing the cognitive overload and the disorientation during the navigation process. In education environments in the distance the search already is great for applications that recognize these yearnings of the users not to lose the interest of its bigger customer who is the pupil. This work, is in the search for the accomplishment of these ideas through the use of the techniques of adaptativa hipermission. It is presented a proposal of implementation of a shaped archetype on the basis of the rules of UWE (UML-based Web Engineering), that it specifies the construction of six models: diagrams of use cases, conceptual model, model of user, model of navigation, model of presentation and model of adaptation.*

1. Introdução

Os ambientes de ensino se propagam com o aumento do número de acessos a rede mundial de computadores. Como atualmente, qualquer dispositivo conectado a Internet possibilita acesso às informações disponíveis em um site na web, tornou-se viável o surgimento de ambientes de ensino virtuais. Existem várias pesquisas sobre este assunto que buscam uma tecnologia mais apropriada para este fim.

O Ensino a Distância ou EAD é um meio de comunicação que utiliza uma ação conjunta de diversos recursos didáticos que propiciam a aprendizagem mais independente e autônoma dos alunos. Com os recursos de hipermídia atuais, o professor pode manter um contato constante com os seus alunos através de fóruns, chat e vídeo-conferência, diminuindo a fronteira de comunicação que existe numa sala de quarenta alunos, por exemplo [Portal Inclusão Digital, 2005].

No entanto, há ainda alguns empecilhos para uma maior disseminação desta nova forma de ensinar. O acesso a Internet no Brasil ainda é demasiadamente caro e a aquisição de computadores não está ao alcance de todos. Além desta exclusão digital, alguns cursos existentes na web não atendem as expectativas dos alunos, sendo que a tecnologia aplicada no desenvolvimento é precária ou faltam profissionais especializados.

Um Sistema Hipermídia Adaptativo (SHA) é uma tecnologia que surgiu para facilitar a aplicação de um curso on-line de uma forma mais dinâmica. Assim, os usuários podem ter processos navegacionais personalizados, em tempo de execução [Brusilovsky, 1996]. Entretanto, a apresentação do conteúdo bem como a quantidade de links acompanha o modelo do usuário dependendo da evolução do aprendizado do mesmo, sendo o sistema responsável por aplicar e modificar o conteúdo da disciplina de acordo com o perfil do aluno.

Procura-se com este trabalho, propor um ambiente de ensino usando os conceitos de hipermídia adaptativa para apoiar o ensino de matemática aos alunos da oitava série do ensino fundamental [Saeb, 2003]. As próximas seções apresentam o processo de modelagem deste SHA, utilizando UWE [Koch, 2000] para tal.

2. Aplicação Proposta

Para mostrar as vantagens do uso de um SHA em um ambiente de ensino, este trabalho propõe um aplicativo que segue as características estabelecidas pelo Modelo de Munich. O Munich trabalha com a complexidade de um SHA ao seguir uma abordagem de engenharia de software (UWE) que é composta por uma série de modelos a fim de promover um adequado desenvolvimento da aplicação. Somando-se a isso, a representação visual destes modelos através da UML, possibilita uma fácil abstração de toda a estrutura [Koch, 2000], [Koch, 2002].

2.1 Os Modelos Adaptativos da UWE

A UWE atua na definição de um projeto sistemático, personalizado e de geração semi-automática para as aplicações WEB. É orientado a objeto, tendo uma abordagem interativa e incremental, utilizando exclusivamente a linguagem UML em suas técnicas, notações e extensões [Koch, 2002].

Segundo Koch (2000), a UWE compõe as diversas etapas do ciclo de vida de um SHA e consiste na modelagem de elementos, notações e métodos. A notação é definida pela UML que representa um conjunto de estereótipos e propriedades específicos em um SHA. Os

métodos definidos por esta metodologia de desenvolvimento são: modelo de casos de uso, modelagem conceitual, navegacional e de apresentação.

2.2 Modelo de Caso de Uso

Neste modelo as interações do sistema com os seus usuários (atores) ou outro sistemas são reveladas. Um ator é parte integrante do sistema e interage com ele, por isso é importante identificar todos os atores e as operações que eles processam no ambiente, não sendo necessário mostrar ainda como estas seriam implementadas (Amaral, 2006). Além disso, este modelo passa para o sistema uma visão da hierarquia dos usuários, relatando o que cada um pode fazer no decorrer do processo de utilização do aplicativo.

Na seqüência a seguir, são mostrados alguns requisitos da aplicação:

- os usuários são matriculados primeiramente antes de possuir o acesso às aulas;
- os visitantes acessam apenas algumas informações sobre o sistema;
- modelo do usuário (MU) é adaptado dinamicamente;
- as aulas são personalizadas seguindo o MU;
- a edição das aulas e o controle de usuários são acessos restritos.

Apesar de possuir um alto nível de abstração, no diagrama de casos de uso são demonstradas algumas situações de processamento da aplicação, suas funcionalidades e responsabilidades. Assim, este modelo não permanece estático na maioria dos sistemas. Com a evolução das atividades presentes na aplicação, novos atores podem ser definidos como responsáveis por novas funções.

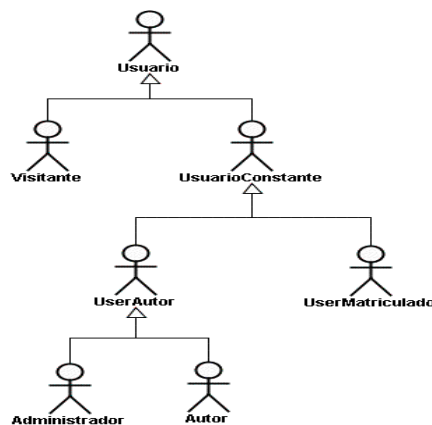


Figura 2: Modelo de Caso de Uso.

No aplicativo proposto, foram identificados os seguintes atores e suas respectivas associações: Usuário, Visitante, UsuarioConstante, UserAutor, UserMatriculado, Autor e Administrador. Na Figura 2 tem-se a estrutura e as associações entre estes atores. Um Usuário (pai) pode ser tanto um Visitante (filho) como um UsuarioConstante (filho), sendo que os elementos filhos herdam as funcionalidades do elemento pai. Já o UsuárioConstante é dividido em UserAutor e UserMatriculado. Por último, o UserAutor é um Autor ou um Administrador.

Na Figura 3 são definidas as atividades que os atores deste sistema podem realizar. O Usuário (Figura 3 (a)), elemento que possui certas funcionalidades referenciadas pelos elementos restantes, como o acesso inicial à aplicação, a visualização de notícias e o envio de e-mails para o sistema. O Visitante (Figura 3 (b)), além de herdar as atividades do Usuário, pode solicitar o cadastro, visualizar uma aula demonstrativa, onde são mostrados os elementos atrativos de uma página hiper-mídia adaptativa e, por fim, pode optar pelo recebimento de notícias do sistema, como avisos da próxima matrícula e etc.

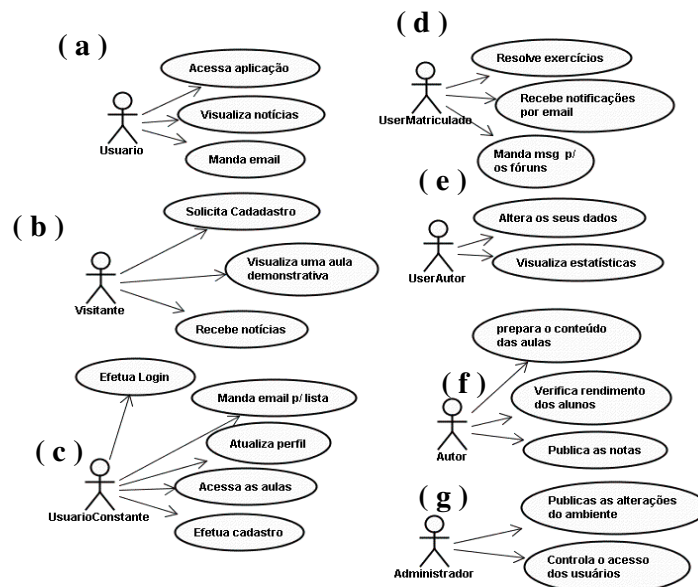


Figura 3: Diagrama de Atividades.

Já um UsuarioConstante (Figura 3 (c)), elemento pai dos elementos relatados a seguir, efetua cadastro e login no ambiente, acessa as aulas, manda e-mails para a lista de contatos, além de atualizar o seu perfil no modelo de usuário. Um UserMatriculado (Figura 3 (d)) representa os alunos da disciplina que resolvem os exercícios, recebem notificações por e-mail do professor (Autor) ou do Administrador e manda mensagens para os fóruns. Ao UserAutor (Figura 3 (e)) cabe apenas as funções básicas de poder alterar os seus dados como a sua senha, por exemplo, e a visualização de estatísticas, que mostra a este ator informações referentes aos usuários que estão on-line no momento, quantos exercícios foram feitos e quais são os que mais participam das atividades propostas. O Autor (Figura 3 (d)) fica responsável pela preparação do conteúdo das aulas, a verificação dos rendimentos de seus alunos e a publicação das notas. Por último, o Administrador (Figura 3 (e)) que publica as alterações feitas na aplicação e controla o acesso dos usuários.

O ator UserMatriculado é o referencial para atualização do modelo do usuário. O mesmo representa um aluno da disciplina da matemática que procura o entendimento dos conceitos da disciplina e uma correta resolução dos exercícios. As relações deste ator com os outros atores presentes no sistema e as suas ações, definem onde o SHA deve atuar para satisfazer as necessidades de seus usuários.

2.3 Modelo Conceitual

Como em outros sistemas, este modelo revela uma visão geral do domínio da aplicação. São definidas as classes e os seus atributos. Assim, como nos casos de uso, o modelo conceitual ignora as formas de apresentação, navegação e interação. Ele apenas procura representar todas as associações entre as classes e as hierarquias de herança. Estas classes, posteriormente no modelo de navegação, derivam os nodos da estrutura hipermídia e as associações definem os links.

A figura 4 apresenta o modelo conceitual da aplicação proposta. A classe Aulas é parte principal deste sistema. É nela que ocorre a maior parte das adaptações necessárias. Sendo composta por um título, um resumo (pequena descrição do que será explicado), a data de publicação efetuada pelo UserAutor, um professor responsável, número de acessos até o momento e o seu endereço na web. Uma aula também pode conter um ou mais tópicos sobre o assunto que será discutido. Nesta classe podem existir comentários dos usuários ou sugestões de sites sobre o assunto. Além disso, numa aula o UserAutor pode anexar arquivos e responder a fóruns, assim como o UserMatriculado. Por fim, a classe Exercicios é importante para as avaliações dos alunos e possui níveis de dificuldade de resolução que só são alterados seguindo o MU.

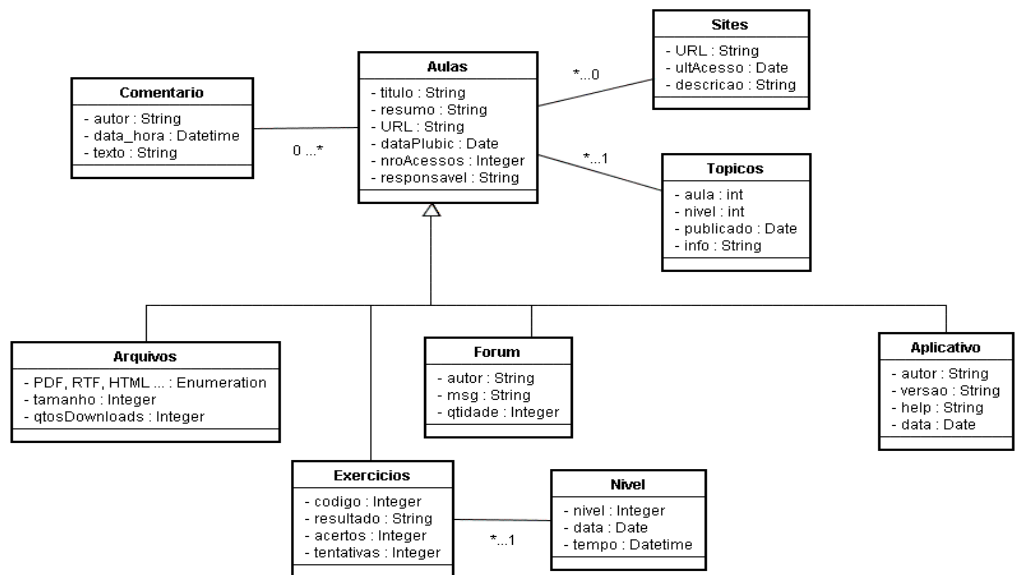


Figura 4: Modelo Conceitual.

Apesar deste modelo ainda não alcançar a representação navegacional, possui todos os recursos que os alunos utilizam no aplicativo. E, por isso, torna-se importante para os modelos seguintes que utilizam as suas classes e relações para a construção de suas estruturas.

2.4 Modelo de Usuário

O Modelo de Usuário (MU) tem a função de armazenar o perfil do aluno, suas características, preferências, interesses, tarefas e a navegação feita por ele no sistema [De Bra, 1999]. Os atributos dos alunos devem ser relacionados uns com os outros e interagir com as classes do domínio. De acordo com Koch (2000), o MU deve orientar um usuário no aprendizado de um determinado tópico, oferecer informações adaptadas ao aluno, ajudá-lo a encontrar as

informações, oferecer um feedback sobre o seu desempenho e prestar assistência ao uso do sistema.

Representado na figura 5, no MU é utilizado o diagrama de classes com suas respectivas associações. Portanto, um sistema para prover uma adaptabilidade adequada para os seus usuários, deve abstrair todas as informações relevantes através deste modelo e atualizá-lo constantemente.

No MU cada usuário possui um nível de acesso ao conteúdo, além de possuir preferências que podem alterar a visualização das aulas, definindo a classe denominada de apresentação adaptativa. Neste modelo também são definidas algumas pendências em relação à resolução de exercícios e a mudança de nível, efetuando as mudanças necessárias para o aluno prosseguir no aprendizado ou exigir que ele refaça algumas atividades que ainda não tenha atingido o conhecimento necessário.

Assim, a classe Usuario, possui todas as informações referentes a um aluno da disciplina a fim de facilitar o seu processo de identificação na aplicação. Cada aluno tem um nível de acesso que possibilita o acesso apenas aos recursos presentes naquele nível, ou seja, ele só poderá ir para a próxima aula se possuir o conhecimento adequado, comprovado pela resolução de exercícios ou de outras atividades definidas pelo professor.

É na classe Aulas que o aluno acessa o conteúdo através de um título, resumo e texto. Nesta classe, também são definidos mais alguns atributos como o seu endereço na web (URL), data da sua publicação, o número de acessos e o responsável pela aula em questão. Em cada aula há exercícios para serem resolvidos pelos alunos que formam um dos meios de avaliação mais importantes da aplicação, influenciando de maneira direta na mudança de nível de acesso no decorrer do uso desta ferramenta de ensino.

Cada aluno, ao nível de personalização do conteúdo, possui preferências como: configurações da apresentação (cores, fontes de texto); visualização ou não de informações adicionais sobre links (anotação); visualização ou não de resumos sobre as aulas; acesso a todos os links ou somente a alguns (ocultação). Além disso, o aplicativo armazenará a data em que foram feitas estas alterações e caso o aluno tenha informado um período de vigência para a permanência destas.

Além de este modelo ser a peça chave na modelagem deste SHA, o MU assume um papel de importância na aplicação proposta, pois com os seus atributos sendo atualizados constantemente, o aluno pode visualizar a disciplina de uma forma mais personalizada aos seus interesses.

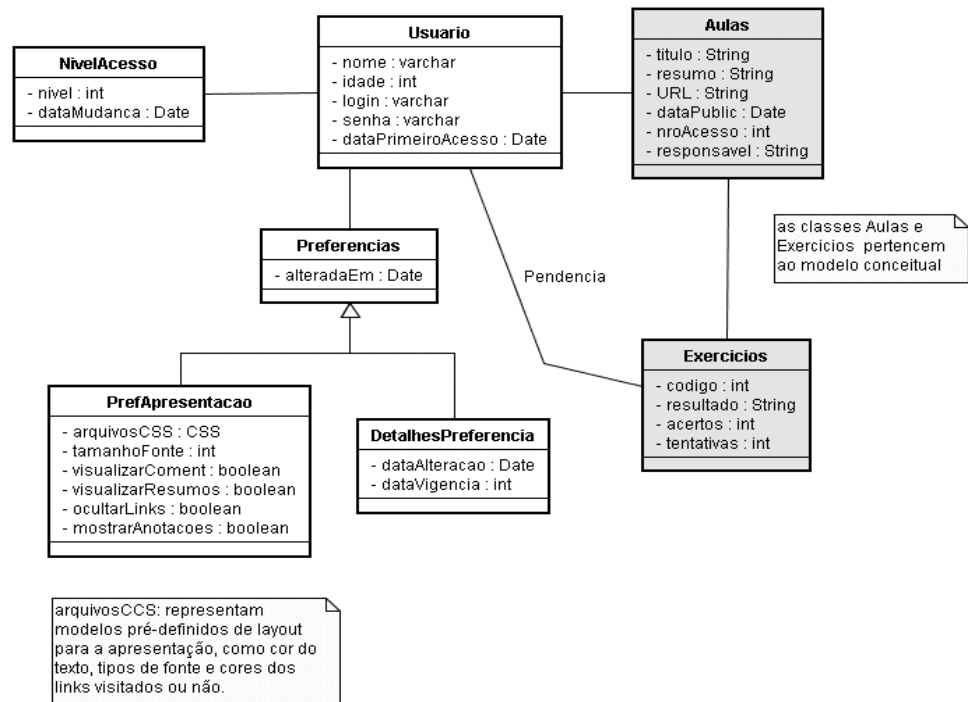


Figura 5: Modelo de Usuário.

2.5 Modelo de Navegação

Um dos maiores problemas existentes na web é a desorientação que a maioria dos sites provocam aos seus visitantes. Excesso de links ou a falta deles propicia uma perda considerável de rendimento na leitura das informações. Além disso, nem sempre a sua identificação, representa o que um usuário queria saber ou o caminho adequado para uma continuação da resolução das suas dúvidas que o levaram a visitar o site.

Sendo assim, procura-se amenizar estes problemas através de dois modelos de navegação propostos por Koch (2002) que são: o modelo de Hiperespaço de Navegação, figura 6, e o modelo de Estrutura de Navegação, figura 7. O primeiro define quais objetos podem ser acessados durante a navegação e o segundo especifica como estes objetos são localizados dentro da estrutura.

Para demonstrar onde as adaptações podem ocorrer neste modelo de navegação são inseridos estereótipos referentes às técnicas que pode ser aplicadas, como a condução direta e anotação adaptativa, e outras que são [Koch, 2000]:

- ordenação (sorted): quando numa lista de menu, os seus itens são dispostos em ordem, indicando uma relevância para o usuário.
- remoção (removed): quando alguns itens podem ser removidos pelo sistema por não serem relevantes ao usuário.
- navegação passiva (passive navigation): define que a navegação é realizada pelo sistema diante de uma reação passiva do usuário.

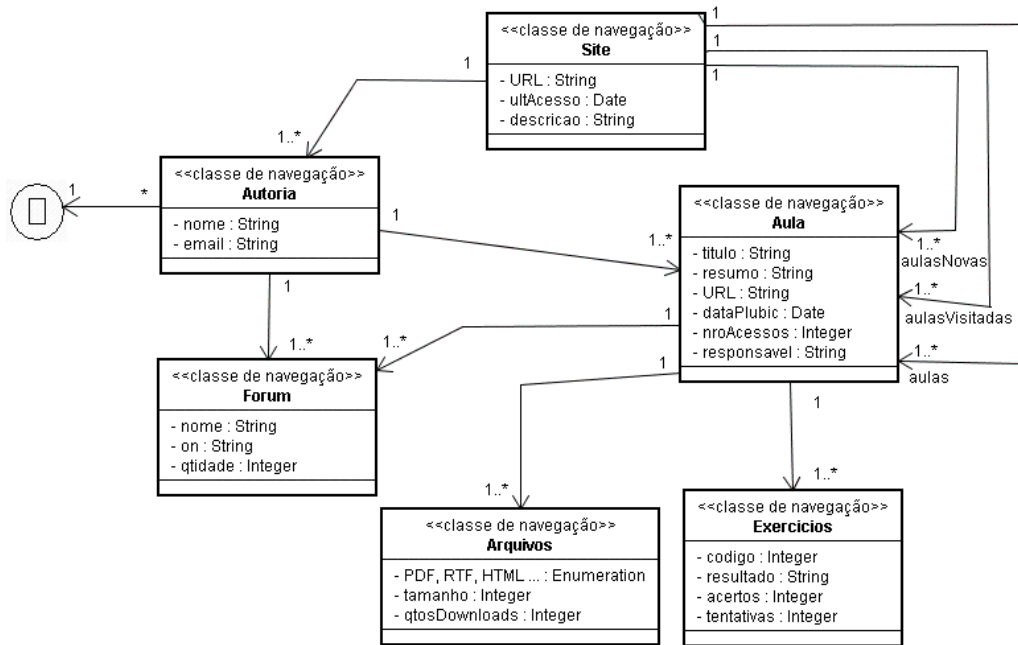


Figura. 6: Modelo de Hiperespaço de Navegação.

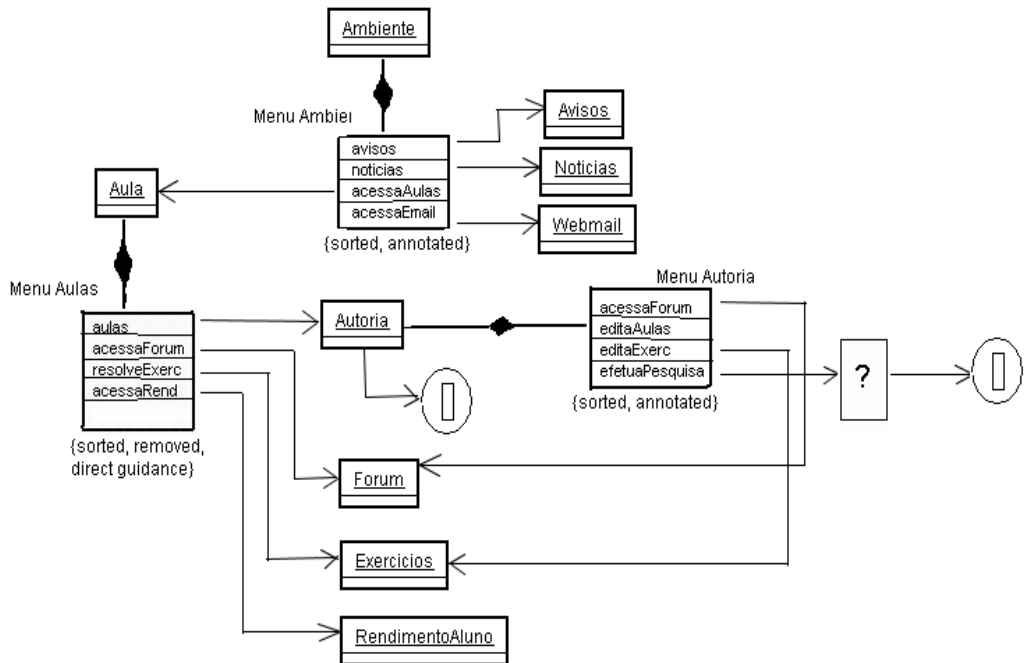


Figura 7: Modelo de Estrutura de Navegação.

2.6 Modelo de Adaptação

O modelo de adaptação foi criado com este propósito: definir como e onde as adaptações podem ser aplicadas. Para isso, ele precisa elaborar uma série de regras para orientar-se no seu objetivo que é apresentar ao aluno desta aplicação um sistema personalizado levando em conta o seu modelo de usuário e ao mesmo tempo o modelo navegacional e de apresentação apropriado às suas características.

A seguir, é mostrado um conjunto de regras sobre as ações do usuário aluno para a aplicação desenvolvida neste trabalho:

- Regra 1 – O aluno deve acessar o sistema através do seu login e senha. Depois disso, o sistema efetua uma comparação dos seus dados com o MU, que está persistente numa base de dados, e verifica qual o nível e a aula que aquele aluno se encontra.
- Regra 2 - Se $\text{exercAula01} = 0$ então $\text{categ.MU} = \text{"iniciante"}$.
- Regra 3 - Se $\text{cod_Aula} < \text{cod_ProxAula}$ então $\text{linksProxAula} = \text{"desabilitado"}$.
- Regra 4 - Se $\text{categ.MU} = \text{"iniciante"}$ então $\text{corFundo} = \text{"branca"}$ e $\text{anotaLinks} = \text{verdadeiro}$.
- Regra 5 - Se $0 < \text{exercAula01} < 5$ então $\text{categ.MU} = \text{"intermediario"}$.
- Regra 6 - Se $\text{categ.MU} = \text{"intermediario"}$ então $\text{corFundo} = \text{"amarela"}$ e $\text{anotaLinks} = \text{falso}$.
- Regra 7 - Se $\text{categ.MU} = \text{"intermediario"}$ então alguns trechos das aulas são ocultados.
- Regra 8 – Se $\text{nroAula} > 5$ então $\text{categ.MU} = \text{"avancado"}$.
- Regra 9 - Se $\text{categ.MU} = \text{"avancado"}$ então $\text{corFundo} = \text{"azulEscuro"}$ e $\text{tamTexto} = \text{"resumido"}$.

Em uma descrição textual, caso o aluno acesse pela primeira vez o ambiente, categoria iniciante, a cor de fundo é branca e ocorre a anotação dos links na área restrita de exibição das aulas. Depois que o aluno acessar a primeira aula e resolver os exercícios corretamente, a sua categoria é alterada para intermediário. Na categoria intermediário, a cor de fundo passa a ser amarelo e os links que aparecem no texto ficam sem anotação. Caso o rendimento do aluno não seja satisfatório durante o intermediário, o link para as próximas aulas é desabilitado até o momento que a sua pontuação fique no nível desejável. Depois de alcançar a aula cinco, a categoria do aluno passa para avançado. A cor de fundo é azul escuro e o conteúdo pode ser mais resumido que as outras categorias.

Estas regras implementadas na aplicação demonstram como o sistema age diante de algumas situações do usuário, aumentando rapidamente o rendimento e o interesse do aluno em relação aos outros sistemas de ensino a distância que não utilizam técnicas de adaptação.

3. Conclusão

No desenvolvimento de aplicativos web direcionados para o ensino a distância, a aplicação das técnicas de hipermídias adaptativas mostrou-se muito eficaz. Este trabalho propôs a elaboração de um protótipo seguindo a modelagem UWE, que facilita o início do planejamento de projetos nesta área, orientando o projetista através de modelos UML referentes às camadas do modelo de referência de Munich. Além disso, foi realizada uma pesquisa com relação a outros dois modelos de referência: o Dexter e o AHAM, sendo o modelo de Munich o escolhido para compor o núcleo do protótipo.

O estudo da UWE permitiu visualizar a complexidade da aplicação dos conceitos de hipermídia adaptativos. A definição dos atores envolvidos na utilização do PSHAE foi decisiva para se decidir pela implementação das regras e ações de apenas um ator. Pois, a complexidade de desenvolvimento aumenta consideravelmente a medida que mais atores participam do sistema, sendo necessário a duplicação dos esforços em relação aos cuidados que deve se ter com o cruzamento das diversas ações executadas em conjunto pelos atores. Assim, apenas o ator UserMatriculado representou as ações de um aluno da disciplina de matemática e partir daí foram elaboradas as regras e as demais adaptações escolhidas para o estudo de caso.

A disciplina de matemática, utilizada como estudo de caso, foi escolhida com o intuito de propor um diferencial no material didático que o professor utiliza como ferramenta de ensino. Como o acesso à internet tem demonstrado um crescimento gradual, os alunos teriam mais este meio disponível para os seus estudos, e o PSHAE se porta como um auxílio ao ensino presencial. Além disso, o professor teria como verificar o andamento da disciplina e propor novos exercícios caso o aluno tivesse dificuldades em algum contexto. Assim, com os SHAEs é possível adaptar o sistema de acordo com o perfil do aluno, partindo do princípio que os usuários são diferentes e, por isso, necessitam de uma personalização que se enquadre às suas necessidades.

Por fim, os ambientes de ensino a distância devem seguir uma modelagem que proporcione o maior número de ferramentas para que os seus usuários fiquem satisfeitos com os resultados e colaborem para o enriquecimento das mesmas. Os Sistemas Hipermídias Adaptativas Educacionais ou SHAEs estão na busca destes objetivos e a continuação das pesquisas nesta área é de suma importância para o avanço dos programas educacionais na web, trazendo mais recursos para o aumento da aceitação de alunos e professores por esta tecnologia.

Referências Bibliográficas

- AMARAL, Marília A.; FERREIRA, Cláudio L.; ZANCHETTI, Pedro; ULBRICHT, Vânia R. Modelo de Referência em Sistemas Hiperídia Adaptativos. In: Anais do Congresso Nacional de Ambientes Hiperídia para Aprendizagem (CONAHPA). Florianópolis, 2006.
- BRUSILOVSKY, Peter. Methods and techniques of adaptive hypermedia. User Modeling and User Adapted Interaction. Kluwer academic publishers, 1996, v 6, n 2-3, pp 87-129.
- DE BRA, P.; HOUBEN, G. J.; WU, H. AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. In: Proceedings of 10th ACM Conference on Hypertext and hypermedia (Hypertext'99), Darmstadt, Germany, February 21 – 25, 1999, ACM Press, pp. 147-156.
- KOCH, Nora Parcus de. Software Engineering for Adaptive Hypermedia Systems. Tese - PhD. 2000. Universität München, Alemanha.
- KOCH, Nora P. The Munich Reference Model for Adaptive Hypermedia. 2nd International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, 2002.
- Portal Inclusão Digital. Inclusão digital: algumas promessas e muitos desafios. Artigo publicado na Revista Soluções sobre alguns desafios a serem enfrentados na promoção da Inclusão Digital no Brasil, Março de 2005. Disponível por WWW em: <http://www.idbrasil.gov.br/docs_prog_gesac/artigos_entrevistas/cidec1>. Acesso em: 15 jul. 2005.
- Sistema Nacional de Avaliação da Educação Básica (Saeb). Qualidade da Educação: uma Nova Leitura do Desempenho dos Estudantes da 8ª Série do Ensino Fundamental, Dezembro de 2003. Disponível por WWW em: <http://www.inep.gov.br/download/cibec/2003/saeb/qualidade_educ.pdf>. Acesso em: 21 jun. 2005.