

**UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL**

**MODELI: UM AMBIENTE DE GERAÇÃO DE MATERIAL  
DIDÁTICO NA WEB BASEADO NA AIM-CID**

TÂNIA REGINA RAMÍRES BEZERRA

Dissertação de Mestrado em Ciência da Computação

Área de Concentração: Engenharia de Software

Departamento de Computação e Estatística

**TÂNIA REGINA RAMÍRES BEZERRA**

**MODELI: UM AMBIENTE DE GERAÇÃO DE MATERIAL  
DIDÁTICO NA WEB BASEADO NA AIM-CID**

Dissertação apresentada como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Curso de Pós-Graduação em Ciência da Computação, Departamento de Computação e Estatística, Universidade Federal de Mato Grosso do Sul.

**Orientador: Prof. Dr. Marcelo Augusto Santos Turine**

**Julho 2008**

A minha mãe.

# Agradecimentos

Ao longo desses anos de estudo tenho muito a agradecer. Primeiramente, a Deus, que não me deixou fraquejar e desistir.

Ao Juliano, meu namorado, pelo apoio incondicional em todos os momentos, apresentando sempre uma palavra de incentivo e carinho com uma enorme paciência nos momentos de estresse.

A minha mãe que desde cedo inculcou em mim a importância do estudo e aperfeiçoamento contínuo.

Ao meu orientador que com apoio, disposição e firmeza conduziu este trabalho. Obrigado pela amizade e confiança.

A Sanesul e a DSF que permitiram que eu me ausentasse para cumprir os créditos exigidos no mestrado.

A todos os amigos, colegas e professores, que direta ou indiretamente contribuíram para a realização deste trabalho, em especial ao amigo cido (Charles Viegas) pelo apoio, compreensão, descontração e amizade.

Seja a mudança que você quer ver no mundo.  
Dalai Lama

# Sumário

1. INTRODUÇÃO .....	2
1.1. Considerações Iniciais .....	2
1.2. Motivação .....	5
1.3. Objetivos .....	6
1.4. Organização do Texto .....	7
2. REVISÃO BIBLIOGRÁFICA .....	8
2.1. Considerações Iniciais .....	8
2.2. Abordagens de Apoio à Modelagem de Conteúdos Didáticos .....	8
2.2.1. Mapas Conceituais .....	9
2.2.2. Modelo HMBS .....	10
2.2.3. HMBS/M .....	13
2.3. Ambientes e Ferramentas de Apoio .....	14
2.3.1. WebCT (Blackboard) .....	15
2.3.2. AulaNet .....	18
2.3.3. Teleduc .....	20
2.3.4. Tidia-Ae .....	22
2.3.5. CoTeia .....	23
2.3.6. Moodle .....	25
2.3.7. WebSCharts .....	26
2.4. Considerações Finais .....	29
3. Abordagem AIM-CID .....	30
3.1. Considerações Iniciais .....	30
3.2. Modelagem de Conteúdos Educacionais .....	30
3.2.1. Modelagem Conceitual .....	34
3.2.2. Modelagem Instrucional .....	38
3.2.3. Modelagem Didática .....	40
3.3. Considerações Finais .....	42
4. AMBIENTE MODELI .....	43
4.1. Considerações iniciais .....	43
4.2. Arquitetura .....	43
4.2.1. Visão Geral .....	43

4.2.2. Especificação .....	44
a. JSF – JavaServer Faces.....	45
b. EMF – Eclipse Modeling Framework .....	47
4.3. Ambiente.....	48
4.4. Considerações Finais .....	54
5. CONCLUSÃO .....	55
5.1. Visão Geral da Pesquisa .....	55
5.2. Contribuições de Pesquisa .....	55
5.3. Trabalhos Futuros.....	56
ANEXO I – PADRONIZAÇÃO NA MODELAGEM DE CASOS DE USO .....	58
1.1. Considerações Iniciais .....	58
1.2. Caso de Uso CRUD .....	59
1.3. Caso de Uso Manter Questão .....	65
1.4. Caso de Uso Manter Conceito.....	66
1.5. Considerações Finais.....	67
ANEXO II – PLUGIN PARA EDIÇÃO DO MODELO NO ECLIPSE .....	69
2.1. Considerações Iniciais .....	69
2.2. Download e Instalação do Plugin.....	69
2.3. Projetos para Edição.....	71
2.4. Edição de um Modelo Existente .....	73
2.5. Upload das Modificações realizadas .....	74
2.6. Considerações Finais.....	74

# Lista de Figuras e Tabelas

Figura 2.1 Exemplo de Mapa Conceitual (Konrath, 2008).....	9
Figura 2.2 Exemplo de especificação de uma ligação (Turine, 1998). ....	11
Figura 2.3 Fases do HMBS/M (Carvalho, 1998).....	14
Figura 2.4 Interface do ambiente <i>BlackBoard</i> acessada pelo Administrador. ....	17
Figura 2.5 Interface principal de um curso no ambiente Aulanet (Aulanet, 2005). ....	19
Figura 2.6 Interface de um curso no ambiente TelEduc (TelEduc, 2008).....	22
Figura 2.7 Interface Tidia-Ae (Tidia-Ae, 2008).....	23
Figura 2.8 Visualização da página da CoTeia. ....	25
Figura 2.9 Visualização da página do Moodle. ....	26
Figura 2.10 Arquitetura da Ferramenta WebSCharts (Brito, 2003).....	28
Tabela 3.1 Requisitos de Modelagem de Conteúdos X Perspectivas de Desenvolvimento de Módulos Educacionais (Barbosa, 2004).....	32
Tabela 3.2 Perspectivas e Requisitos para Modelagem de Conteúdos Educacionais (Barbosa, 2004).....	33
Figura 3.1 AIM–CID - Abordagem Integrada para Modelagem de Conteúdos Educacionais (Barbosa, 2004).....	33
Figura 3.2 Estrutura Geral do Módulo Modelagem de Software.....	36
Figura 3.3 Modelo Conceitual do Sub-módulo Diagramas da UML.....	37
Figura 3.4 Modelo instrucional parcial do Módulo Modelagem de Software.....	40
Figura 3.5 Modelagem Didática com Especificação Aberta.....	41
Figura 4.1 Arquitetura da ferramenta ModELi.....	44
Figura 4.2 Diagrama de Classes ModELi: Parte 1.....	45
Figura 4.3 Diagrama de Classes ModELi: Parte 2.....	46
Figura 4.4 Diagrama de Classes ModELi: Parte 3.....	46
Figura 4.5 Padrão de arquitetura MVC.....	47
Figura 4.6 Consulta do Módulo Educacional. ....	49
Figura 4.7 Consulta do Módulo Educacional. ....	49

Figura 4.8 Manutenção de um Módulo Educacional.....	49
Figura 4.9 Manutenção do Item de Informação Conceito.....	50
Figura 4.10 Manutenção do Item de Informação Conceito associando a outro Item de Informação	50
Figura 4.11 Manutenção do Elemento Instrucional Exemplo associado a um Item de Informação.	51
Figura 4.12 Geração do Conteúdo Didático. ....	52
Figura 4.13 Geração do Conteúdo Didático em HTML.....	52
Figura 4.14 Geração do Conteúdo Didático em PDF .....	53
Figura 4.15 Geração de Conteúdo Didático em <i>PowerPoint</i> . ....	53
Figura I.1 Caso de Uso CRUD demonstrando os cenários/requisitos que podem ser sobrescritos.	59
Figura I.2 Modelo de Caso de Uso: Manter Questão. ....	65
Figura I.3 Modelo de Caso de Uso: Manter Conceito.....	66
Figura II.1 Download do Arquivo XMI e do Plugin .....	69
Figura II.2 Importação do Plugin.....	70
Figura II.3 Seleção do diretório do Plugin.....	70
Figura II.4 Escolha do Plugin.....	71
Figura II.5 Projeto ModELi EMF .....	71
Figura II.6 Projetos que permitem a edição do modelo .....	72
Figura II.7 Arquivo do meta-modelo .....	72
Figura II.8 Projeto vazio.....	73
Figura II.9 Modelo sendo editado .....	73
Figura II.10 Upload do Modelo Alterado .....	74

# Lista de Siglas

<b>ADV</b>	<i>Abstract Data Views</i>
<b>AIM-CID</b>	Abordagem Integrada de Modelagem – Conceitual, Instrucional e Didática
<b>API</b>	<i>Application Programming Interface</i>
<b>COWEB</b>	<i>Collaborative Web Site</i>
<b>CRUD</b>	<i>Create, Read, Update and Delete</i>
<b>DCT</b>	Departamento de Computação e Estatística
<b>DSF</b>	Desenvolvimento de Sistemas Fiscais
<b>EAD</b>	Educação a Distância
<b>EMF</b>	<i>Eclipse Modeling Framework</i>
<b>EORM</b>	<i>Enhanced Object Relationship Model</i>
<b>GATECH</b>	Laboratório de Software Colaborativo do Instituto de Tecnologia da Geórgia
<b>GMF</b>	<i>Graphical Modeling Framework</i>
<b>GPL</b>	<i>GNU General Public License</i>
<b>HDM</b>	<i>Hypertext Design Model</i>
<b>HMBS</b>	<i>Hypermedia Model Based on Statecharts</i>
<b>HMBS/M</b>	<i>Hypermedia Model Based on Statecharts/Method</i>
<b>HySCharts</b>	<i>Hypermedia System based on Statecharts</i>
<b>HTML</b>	<i>Hypertext Markup Language</i>
<b>ICMC-USP</b>	Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo
<b>IDE</b>	<i>Integrated Development Environment</i>
<b>IMM</b>	<i>Information Management Metamodel</i>
<b>J2EE</b>	<i>Java 2 Platform, Enterprise Edition</i>
<b>JMI</b>	<i>Java Metadata Interface</i>
<b>JSF</b>	<i>JavaServer Faces</i>
<b>LCMS</b>	<i>Learning Content Management Systems</i>

<b>LEDES</b>	Laboratório de Engenharia de Software
<b>LES</b>	Laboratório de Engenharia de Software
<b>LMS</b>	<i>Learning Management Systems</i>
<b>MDT</b>	<i>Model Development Tools</i>
<b>MVC</b>	<i>Model View Controller</i>
<b>NIED</b>	Núcleo de Informática Aplicada à Educação
<b>OOHDM</b>	<i>Object-Oriented Hypermedia Design Method</i>
<b>ODF</b>	<i>OpenDocument Format</i>
<b>OMT</b>	<i>Object Modeling Technique</i>
<b>PUC-RIO</b>	Pontifícia Universidade Católica - Rio de Janeiro
<b>SDO</b>	<i>Service Data Objects</i>
<b>SOA</b>	<i>Service Oriented Architecture</i>
<b>UML</b>	<i>Unified Modeling Language</i>
<b>UFMS</b>	Universidade Federal de Mato Grosso do Sul
<b>UNICAMP</b>	Universidade de Campinas
<b>XMI</b>	<i>XML Metadata Interchange</i>
<b>XML</b>	<i>Extensible Markup Language</i>
<b>XSD</b>	<i>XML Schema Definition</i>
<b>XSLT</b>	<i>Stylesheet Language Transformations</i>
<b>W3C</b>	<i>World Wide Web Consortium</i>
<b>WWW</b>	<i>World Wide Web</i>

# Resumo

Nos últimos anos, a área de educação vem passando por mudanças importantes em relação à introdução das tecnologias de informação e comunicação no processo de ensino e de aprendizagem. Como resultado dessas mudanças e dos esforços para aproveitar os benefícios dessas tecnologias, diversos sistemas ou ambientes virtuais de apoio à educação foram desenvolvidos. Apesar do vasto conjunto de ferramentas oferecidas por esses ambientes, muitos educadores e/ou pesquisadores ainda optam pelo desenvolvimento de suas próprias aplicações de apoio à educação em razão da diversidade das técnicas de ensino e avaliação. Devido ao fato dos ambientes educacionais tradicionais não serem projetados para auxiliar na autoria de conteúdos educacionais, os usuários são desmotivados a utilizar esses sistemas. A modelagem de conteúdos é considerada uma das atividades relevantes do processo educacional, sendo fundamental para a estruturação do conhecimento que se deseja ensinar. Neste contexto, a abordagem AIM–CID (Abordagem Integrada de Modelagem – Conceitual, Instrucional e Didática) foi proposta e define requisitos e perspectivas para a modelagem conceitual, instrucional e didática de conteúdos educacionais. Assim, neste trabalho objetiva-se especificar e implementar uma ferramenta Web intitulada **ModELi (Módulos Educacionais Livres)** para apoiar a modelagem de conteúdos educacionais e gerar conteúdos didáticos livres na forma de arquivos de textos (formatos *PDF* e *HTML*) e/ou arquivos de apresentações (formato *PowerPoint* e Apresentação *ODF*), segundo a modelagem formal baseada em Statecharts subjacente à abordagem AIM-CID. Para validar a ferramenta foi realizado um estudo de caso do curso “Diagramas da UML: Uma Visão Geral”.

**Palavras-chave:** *AIM-CID, Autoria de Conteúdos Educacionais, Modelagem Educacional, Modelagem Conceitual, Modelagem Instrucional, Modelagem Didática, Statecharts.*

# Abstract

In recent years, the education area is going through major changes in relation to the introduction of information technology in the process of teaching and learning. As a result of these changes and efforts to take advantage of these technologies, several systems (or virtual environments) to support education have been developed. Despite the wide range of tools offered by these environments, many educators and/or researchers still choose for developing their own applications to support education because of the diversity of techniques for teaching and evaluation. Due to the fact that traditional educational environments not offer features to help authors of educational content, users are disappointed to use these systems and end up creating their own applications. The modeling of contents present itself as one of the relevant activities to be considered, it's fundamental to the structure of knowledge what would to teach. In this context, the approach IMA–CID (Integrated Modeling Approach – Conceptual, Instructional and Didactic) was proposed and defines requirements and prospects for conceptual modeling, instructional and teaching of educational content. Thus, this study aims at specifying and implementing a tool on Web **ModELi** (Free Educational Modules) to support the modeling of educational content and generate open educational content in the form of text files (PDF and HTML format) and / or files of presentations (*PowerPoint* and Presentation ODF format), according to the formal modeling based on Statecharts underlying the approach IMA-CID. To validate the tool was conducted a case study of the course "UML Diagrams: Overview."

**Key-Words:** *IMA-CID, Author of Educational Content, Design Education, Conceptual Design, Instructional Design, Modeling Didactic, Statecharts.*

# 1. INTRODUÇÃO

## 1.1. Considerações Iniciais

É notável que o computador está cada vez mais sendo utilizado como uma ferramenta auxiliar no processo de ensino/aprendizagem, fomentando a busca pela construção do conhecimento de maneira fácil e agradável mediante a utilização de ambientes virtuais educacionais (Oliveira *et al.*, 2008).

Nesse sentido, surgem novas possibilidades geradas em função das tecnologias da informação e comunicação, criando-se novas formas educativas. Contudo, para que esses novos meios tecnológicos sejam de fato educativos, é preciso que estejam ligados primordialmente a um conjunto de intenções e a uma prática de ensino que tenha como função a construção de uma ação educativa (Garbin & Amaral, 2008).

Dessa forma, é possível obter ambientes virtuais educacionais, variando desde a simples informatização dos métodos tradicionais de instrução (paradigma instrucionista) até o desenvolvimento de novas formas de aprendizagem (paradigma construcionista). Valente (1995 a) menciona que as primeiras experiências de uso dos computadores na educação enfatizavam a estruturação de um conjunto de informações sobre assuntos específicos, de modo que os estudantes eram condicionados a estudar essas informações numa seqüência pré-determinada pelo projetista do software. Posteriormente, com o aumento no nível de interação e com o uso de novas tecnologias, especialmente as de comunicação, o enfoque das pesquisas em informática na educação passou a ser o projeto de ambientes e ferramentas que servissem, não apenas como instrutores ou apresentadores de conteúdo, mas também como auxiliares do processo de ensino-aprendizagem.

Neste contexto, torna-se cada vez mais comum a adoção de ambientes interativos, e essa interatividade tornou-se a principal atração desses espaços virtuais. Cada vez mais, softwares e ambientes na Internet utilizam a interatividade para atrair usuários, entre eles estão bate-papos;

softwares de mensagens instantâneas, como MSN Messenger<sup>1</sup>; ambientes de relacionamento como o Orkut<sup>2</sup>, dentre outros. Tais recursos são bastante comuns entre os usuários de Internet, na atualidade.

Sendo assim, a Internet passou a ser intensamente explorada em programas de treinamento e educação em função das facilidades oferecidas para seus usuários. Programas de educação/treinamento presencial, caracterizados pela sala de aula, lousa e giz, podem utilizar os ambientes virtuais educacionais baseados na Internet para apoio ao processo de ensino (Abowd *et al.*, 2000; Silva *et al.*, 2000). Nesse caso, os ambientes virtuais educacionais são utilizados para auxiliar no gerenciamento das atividades propostas e realizadas pelas entidades envolvidas no processo educacional (professores, estudantes, monitores, etc) e para armazenar e organizar materiais didáticos. Atualmente, grande parte dos professores, especialmente em universidades, faz uso das facilidades da Internet para apoio às suas aulas presenciais, seja para disponibilizar conteúdos ou para estabelecer mecanismos de comunicação com os estudantes (e entre eles).

Além do apoio à sala de aula, a Internet também passou a ser utilizada na área de Educação a Distância (EAD), modalidade educacional prevista no artigo 80 da Lei de Diretrizes e Bases da Educação - Lei 9.394/96 na qual a mediação didática pedagógica nos processos de ensino e de aprendizagem ocorre com a utilização de meios e tecnologias de informação e comunicação, com estudantes e professores desenvolvendo atividades educativas em lugares ou tempos diversos.

Inicialmente, a EAD era realizada por correspondência, televisão, vídeo ou rádio (Roberts, 1996). O uso da EAD via Internet vai além do simples apoio à tarefa realizada na sala de aula. Nesse caso, os recursos oferecidos nos sistemas são os principais meios pelos quais o conhecimento é transmitido aos estudantes e pelos quais estes têm acesso aos professores e aos outros estudantes, já que nessa forma de educação não existe a noção de sala de aula física, mas sim o conceito de sala de aula virtual.

Mediante a expansão da EAD através da utilização pela Internet, surge o sistema UAB (Universidade Aberta do Brasil) cujo principal objetivo é desenvolver a modalidade de educação à distância, com a finalidade de expandir e interiorizar a oferta de cursos e programas de

---

<sup>1</sup> O programa permite que um usuário da Internet se comunique com outro que tenha o mesmo programa em tempo real, podendo ter uma lista de amigos "virtuais".

<sup>2</sup> É uma rede social, com objetivo de ajudar seus membros a criar novas amizades e manter relacionamentos.

educação superior no país, além de ampliar o acesso à educação superior pública levando tais cursos às diferentes regiões do país. É objetivo, também, oferecer cursos superiores para capacitação de dirigentes, gestores e trabalhadores em educação básica dos estados, do distrito federal e dos municípios e apoiar a pesquisa em metodologias inovadoras de ensino superior respaldados em tecnologias de informação e comunicação. Além disso, pretende-se atingir objetivos sócio-educacionais com a colaboração da União com entes federativos, e estimular a criação de centros de formação permanentes por meio dos pólos de apoio presencial (UAB, 2008).

De acordo com Oreste Preti (2000) e Correia e Antony (2003) a EAD, enquanto prática-educativa deve considerar a realidade e comprometer-se com os processos de libertação do ser humano em direção a uma sociedade mais justa, solidária e igualitária. Enquanto prática mediatizada, deve fazer recurso à tecnologia, entendida como um processo lógico de planejamento, como um modo de pensar os currículos, os métodos, os procedimentos, a avaliação, os meios, na busca de tornar possível o ato educativo.

Kenski (2001) relata a preocupação necessária à formação dos professores, que necessita praticar uma educação de qualidade independente das ferramentas disponíveis como recursos didáticos:

*“Como levar um professor a praticar um ensino de qualidade em meio às mudanças velozes e estruturais nas esferas dos conhecimentos, saberes e práticas que ocorrem na atualidade? O grande desafio inicial, em termos de realidade brasileira, é a formação de professores capazes de lidar com alunos e situações extremas: alunos que já possuem conhecimentos tecnologicamente avançados e acesso pleno ao universo de informações disponíveis nos múltiplos espaços virtuais; alunos que se encontram em plena exclusão tecnológica, sem oportunidade para vivenciar e aprender nesta nova realidade; instituições de ensino equipadas com as mais modernas tecnologias digitais aos espaços educacionais precários e com recursos mínimos para se trabalhar”.*

No que diz respeito aos objetivos pedagógicos, pretende-se que o aluno faça parte do desenvolvimento das estratégias de uma educação crítica, que privilegie uma abordagem construtivista de aprendizagem, desenvolvendo a compreensão na assimilação de habilidades solicitadas pelo atual perfil profissional: criatividade, raciocínio crítico, boa qualidade de informações, posição integradora na dinâmica das relações, com cooperações e parcerias on-line, visões empreendedoras e de projetos auto-sustentáveis, dentre outras (Pagano, 2002).

Como resultado dessas mudanças e dos esforços para aproveitar os benefícios dos ambientes virtuais educacionais no processo de ensino e de aprendizagem diversos sistemas de apoio a EAD foram desenvolvidos, tais como: WebCT (WebCT, 2006)/Blackboard (Blackboard, 2006), TelEduc (Rocha, 2002b; TelEduc, 2008), AulaNet (Aulanet, 2005), TopClass (TopClass, 2006), WebCoM (Silva & Moreira, 2003b), Col (COL, 2006), iClass (iClass, 2006), Angel (Jafari, 2002), *AdaptWeb* (Freitas et. al. 2002) e Moodle (Dougiamas & Taylor, 2003).

Em geral, esses sistemas oferecem uma variedade de ferramentas para a criação e acompanhamento de cursos baseados na Web, tais como gerenciamento de material didático, usuários, atividades, notas, comunicação, entre outras. Porém, não oferecem recursos computacionais e pedagógicos para auxiliar na autoria e no desenvolvimento de módulos educacionais (material didático), que segundo Barbosa (2004) são unidades concisas de estudo, compostas por conteúdos teóricos integrados a atividades práticas e avaliações, cuja disponibilização aos aprendizes é apoiada por recursos tecnológicos e computacionais. Neste contexto, uma ferramenta para auxiliar a autoria dos módulos educacionais é extremamente relevante tanto para a EAD como para aulas presenciais, possibilitando a reutilização de materiais didáticos, e tal ferramenta é o escopo do presente trabalho de mestrado.

## **1.2. Motivação**

Apesar do vasto conjunto de ferramentas oferecidas pelos ambientes colaborativos, muitos usuários ainda optam pelo desenvolvimento de suas próprias ferramentas de apoio à educação. Isso acontece em função da diversidade do currículo e das técnicas de ensino e avaliação existentes entre diferentes instituições e, até mesmo, entre professores dentro de uma mesma instituição, o que muitas vezes dificulta a adoção de um sistema existente e sua completa adaptação a necessidades específicas.

Em geral, os ambientes educacionais existentes visam oferecer suporte para apresentação e disponibilização de material didático, além da condução de atividades de comunicação, colaboração, avaliação e acompanhamento dos aprendizes. Segundo Barbosa (2004), tais ambientes limitam-se somente à criação da estrutura do curso, armazenamento e controle de acesso ao conteúdo e monitoração do aprendiz, deixando a tarefa de autoria do material didático, ou seja, modelagem do conteúdo educacional inteiramente a cargo do autor do curso, sem quaisquer diretrizes ou mesmo um processo sistemático que apóie sua realização.

Assim, uma limitação comum à maioria dos trabalhos conduzidos neste contexto refere-se ao fato dos ambientes e técnicas subjacentes concentrarem-se na criação da estrutura, armazenamento do conteúdo e controle de acesso ao material didático, deixando a tarefa de modelagem do conteúdo educacional inteiramente a cargo do autor do curso, sem qualquer tipo de suporte ou mesmo diretrizes e atividades sistemáticas que apoiem sua elaboração.

Desta forma, é fundamental o desenvolvimento de novas ferramentas complementares aos ambientes educacionais existentes para apoiar a modelagem, o planejamento e a organização do conteúdo didático. Tal proposta é uma atividade essencial para a melhoria da qualidade dos materiais didáticos disponíveis nos ambientes colaborativos.

Neste contexto, a principal motivação para realização deste trabalho é dar continuidade ao trabalho de doutorado de Barbosa (2004), que propôs a abordagem AIM-CID (Abordagem Integrada de Modelagem – Conceitual, Instrucional e Didática) contendo requisitos e perspectivas para a modelagem conceitual, instrucional e didática de conteúdos educacionais. No nível conceitual são utilizados mapas conceituais estendidos; no nível instrucional é estabelecido o modelo *HMBS/Instrucional*; e no nível didático propõe-se o modelo *HMBS/Didático*. Em seu trabalho de doutorado, Barbosa não implementou uma ferramenta para interpretar e executar tais modelos a fim de gerar automaticamente conteúdos educacionais.

Além disso, a utilização da técnica de especificação formal baseada em Statecharts (Harel, 1987) subjacente ao modelo HMBS (*Hypermedia Model Based Statecharts*) (Turine et al., 1997; Turine, 1998, Turine et al., 1998), utilizado na abordagem AIM-CID, é de interesse do grupo de pesquisadores do Laboratório de Engenharia de Software (LEDES) do Departamento de Computação e Estatística da Universidade Federal de Mato Grosso do Sul (DCT-UFMS).

Assim, surge uma oportunidade para especificar e implementar uma ferramenta Web baseada na abordagem AIM-CID para apoiar a autoria e a gestão de conteúdos educacionais.

### **1.3. Objetivos**

Este trabalho tem como objetivo principal especificar e implementar um ambiente Web intitulado **ModELi (Módulos Educacionais Livres)** para apoiar a modelagem de conteúdos educacionais e gerar conteúdos didáticos livres na forma de arquivos de textos (formatos *PDF* e *HTML*) e/ou arquivos de apresentações (formato *PowerPoint* e Apresentação ODF), segundo a

modelagem formal baseada em Statecharts subjacente à abordagem AIM-CID. A partir da interpretação e da execução dos modelos *HMBS/Instrucional* e *HMBS/Didático* da AIM-CID e da geração de módulos educacionais em diferentes formas de apresentação para o aprendiz, pretende-se estabelecer um cenário para o desenvolvimento de módulos educacionais livres (*open learning materials*). Para validar a ferramenta, será modelado o curso “Diagramas da UML: Uma Visão Geral”.

#### **1.4. Organização do Texto**

Este trabalho está organizado nos seguintes capítulos. No Capítulo 2 são apresentados modelos e métodos de modelagem de conteúdos, bem como alguns ambientes de apoio à educação. A abordagem AIM-CID é apresentada no Capítulo 3. No Capítulo 4 é apresentada a ferramenta ModELi para apoio à geração conteúdos educacionais livres sob as perspectivas conceituais, instrucionais e didáticas baseado na abordagem AIM-CID. No Capítulo 5 é apresentada a conclusão deste trabalho abordando as principais contribuições e limitações e algumas citações de relevância ao projeto em questão são descritos no Anexo.

## 2. REVISÃO BIBLIOGRÁFICA

### 2.1. Considerações Iniciais

A motivação do presente trabalho de mestrado foi originada no contexto de desenvolvimento de material didático e, especificamente, na proposta tecnológica de uma plataforma Web para apoiar a autoria de conteúdos educacionais livres. Desta forma apresenta-se, neste capítulo, uma visão geral sobre abordagens, modelos e métodos de apoio a autoria de material didático, além de ambientes utilizados para gerenciamento de cursos de EAD.

Na Seção 2.22.2 são apresentadas algumas abordagens de apoio a modelagem de conteúdos educacionais utilizados na AIM-CID. Na Seção 2.32.3 são descritas as formas mais comuns de uso da tecnologia no contexto educacional e apresentados exemplos de ambientes virtuais que apóiam ou viabilizam essas práticas, e por fim, na Seção 2.4. as considerações finais do capítulo.

### 2.2. Abordagens de Apoio à Modelagem de Conteúdos Didáticos

A elaboração e o projeto de material didático envolvem, sobretudo, a organização e a estruturação das informações sobre o domínio de conhecimento que se deseja ensinar. O uso de mecanismos e abordagens que apóiem a modelagem dos conteúdos educacionais permite ao autor capturar e organizar a estrutura de um domínio complexo e torná-la clara e acessível aos aprendizes.

Nas próximas seções são apresentados a técnica Mapas Conceituais (técnica utilizada em várias abordagens como base para a representação do domínio de conhecimento), o modelo HMBS (*Hypermedia Model Based on Statecharts*), o modelo HMBS/M (*Model Based on Statecharts / Method*) subjacentes à abordagem AIM-CID utilizada neste trabalho.

### 2.2.1. Mapas Conceituais

Um mapa conceitual consiste na representação de conceitos referentes a um domínio de conhecimento na forma de estruturas hierárquicas ou grafos orientados. Cada nó contém o nome de um conceito e cada arco entre dois conceitos indica o relacionamento existente entre eles.

Os mapas conceituais foram desenvolvidos, inicialmente, por Novak (Moreira & Buchweitz, 1987; Novak, 1981; Novak & Gowin, 1984; Novak, 1990). Sua proposta está baseada na idéia fundamental da Psicologia Cognitiva de Ausubel (Ausubel; Novak; Hanesian, 1980), onde a aprendizagem ocorre por assimilação de novos conceitos e proposições na estrutura cognitiva do acadêmico. Novas idéias e informações são aprendidas, na medida em que existem pontos de ancoragem, conforme ilustra a Figura 2.1. Aprendizagem implica em modificações na estrutura cognitiva e não apenas em acréscimos (Konrath, 2008).

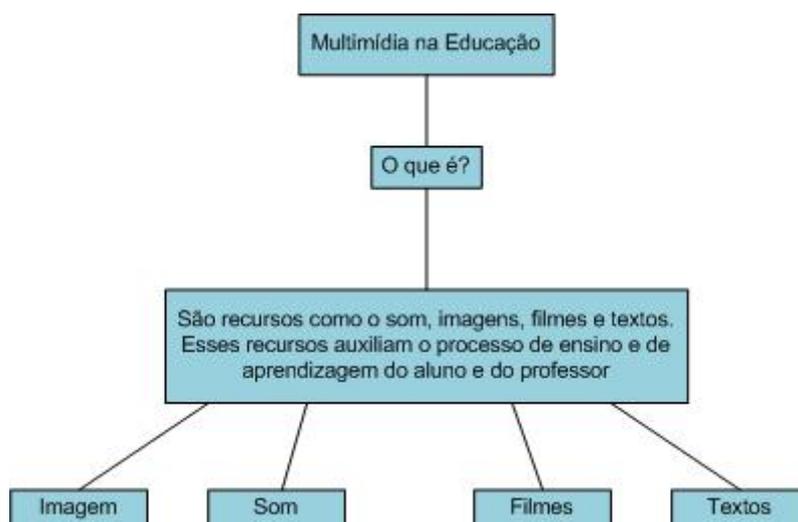


Figura 2.1 Exemplo de Mapa Conceitual (Konrath, 2008).

A utilização dos mapas conceituais no processo educacional auxilia a ordenação e a seqüenciação dos conteúdos de ensino, de forma a oferecer estímulos adequados ao acadêmico. Podem ser usados como um instrumento que se aplica às diversas áreas do ensino e da aprendizagem escolar, como planejamentos de currículo, sistemas e pesquisas em educação.

Nesta perspectiva, esta teoria procura explicar os mecanismos internos que ocorrem na mente dos seres humanos. Para os professores, os mapas conceituais podem constituir-se em poderosos auxiliares nas suas tarefas rotineiras, conforme destaca (Konrath,2008), tais como:

- Tornar claro os conceitos difíceis, arranjados em uma ordem sistemática;
- Auxiliar os professores a manterem-se mais atentos aos conceitos chaves e às relações entre eles;
- Auxiliar os professores a transferir uma imagem geral e clara dos tópicos e suas relações para seus estudantes;
- Reforçar a compreensão e aprendizagem por parte dos alunos;
- Permitir a visualização dos conceitos chave e resumir suas inter-relações;
- Verificar a aprendizagem e identificar conceitos mal compreendidos pelos acadêmicos;
- Auxiliar os professores na avaliação do processo de ensino;
- Possibilitar aos professores avaliar o alcance dos objetivos pelos acadêmicos por meio da identificação de conceitos mal entendidos e dos que estão faltando.

### **2.2.2. Modelo HMBS**

O HMBS (*Hypermedia Model Based on Statecharts*) [(Turine et al., 1997; Turine, 1998; Turine et al., 1998)] consiste em um modelo para especificar a estrutura e a semântica navegacional de aplicações Web utilizando a técnica *Statecharts* (Harel, 1987) como modelo de especificação formal subjacente. Os modelos instrucional e didático da abordagem AIM-CID, utilizada neste trabalho, utilizam o modelo HMBS como base para sua especificação.

De acordo com o HMBS, uma aplicação Web é composta por três tipos de objetos: estruturais, navegacionais e de apresentação. Os objetos estruturais (estados, transições e eventos) são especificados a partir da estrutura organizacional definida pelo *Statecharts* subjacente. Os objetos navegacionais (páginas, ligações e âncoras) definem a estrutura navegacional da aplicação Web. Os objetos de apresentação são definidos por meio de canais de apresentação, que correspondem a abstrações de dispositivos pelos quais é possível especificar os requisitos associados à apresentação da aplicação. Os canais são invocados para visualizar as informações contidas nas páginas durante a navegação, atuando como interpretadores da página.

Nesse sentido, uma aplicação Web é definida no HMBS segundo uma organização hierárquica dos estados do *Statecharts* que representam a sua estrutura organizacional e navegacional, os quais são associados a porções de informação ou páginas. Os eventos contidos nos rótulos das transições representam as âncoras que disparam as possíveis ligações entre as páginas, definindo assim os caminhos de navegação disponíveis ao usuário, conforme ilustrado na Figura 2.2.

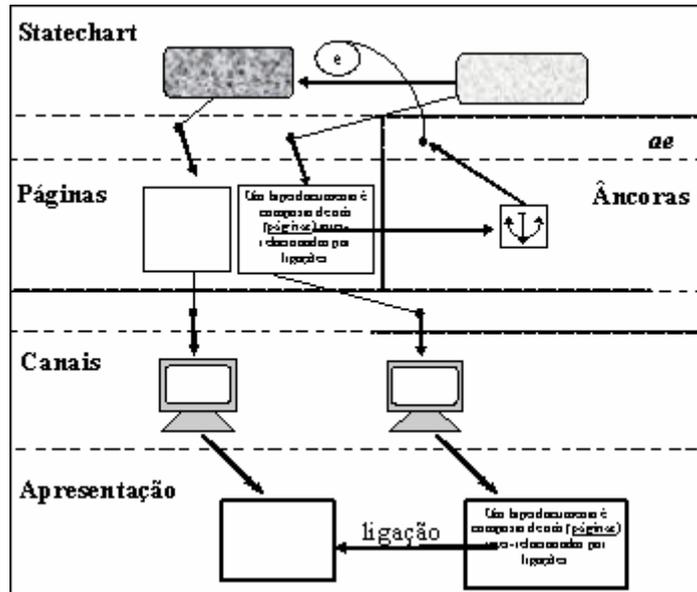


Figura 2.2 Exemplo de especificação de uma ligação (Turine, 1998).

Na especificação formal do modelo HMBS, uma aplicação Web H é definida como uma 7-tupla (Turine, 1998):

$$H = \langle ST, P, m, L, pl, ae, N \rangle, \text{ na qual:}$$

- a) ST representa o Statechart que especifica a estrutura organizacional e navegacional subjacente da aplicação. O Statechart é definido como uma 11-tupla,  $ST = \langle S, \rho, \psi, \gamma, \delta, V, C, E, A, R, T \rangle$ , representando, respectivamente, estados, função de hierarquia, função tipo de decomposição, função história, função default, conjunto de expressões, conjunto de condições, conjunto de eventos, conjunto de ações, conjunto de rótulos e conjunto de transições. É interessante ressaltar algumas simplificações importantes no Statechart adotado: um símbolo de história não pode ser default de um estado, o evento  $time\_out^3$  não é permitido, e os eventos não podem ser formados por uma combinação

<sup>3</sup>  $time\_out(e,m)$ : determina a ocorrência de um evento “e” no instante em que “m” passos transcorreram desde o momento em que o estado origem da transição foi ativado.

lógica de outros eventos, ou seja, se  $f \in E$  e  $g \in E$  então  $(f \text{ or } g) \notin E$  e  $(f \text{ and } g) \notin E$ ; (Turine, 1998).

- b)  $P = \{p_1, p_2, \dots, p_n\}$ ,  $n > 0$ , é o conjunto finito de páginas de informação que define o conteúdo da aplicação. Cada página  $p \in P$ , comumente denominada nó de informação, contém uma porção de informação relevante em um determinado contexto da aplicação e é definida conceitualmente pela tripla  $\langle c, t, Anc_p \rangle$ , tal que “c” é a porção do conteúdo de informação, que pode ser composta por mídias estáticas ou independentes de tempo (texto, gráfico ou imagem, por exemplo), ou por mídias dinâmicas ou dependentes de tempo (vídeo, áudio ou animação); “t” representa o título associado à página, o qual deve identificar o conceito que melhor descreve a informação contida na página a fim de auxiliar na busca por uma informação específica durante o processo de navegação;  $Anc_p = \{a_1, a_2, \dots, a_m\}$ ,  $m > 0$  define uma coleção de âncoras associadas à página. O conjunto de páginas pode incluir uma página nula especial ( $P_\lambda$ ), sem qualquer conteúdo, título e/ou âncora, a qual pode ser associada a estados que não modelam apresentação de informação;
- c)  $m: SS \rightarrow P$  denota uma função que mapeia estados “s” de um subconjunto  $SS$  ( $SS \subset S$ ) a páginas  $p \in P$  da aplicação. O subconjunto “SS” é definido por:  $S_S: \{x \in S \mid \psi(x) = OR \vee \rho(x) = \phi\}$ , isto é, “ $S_S$ ” é o conjunto formado pelos estados compostos do tipo OR e pelos estados atômicos do Statechart. Estados do tipo AND não são mapeados para páginas, pois são utilizados unicamente para especificar concorrência de informações na apresentação;
- d)  $L = \{l_1, l_2, \dots, l_k\}$ ,  $k > 0$ , define o conjunto de canais de apresentação (ou leitores), que são abstrações de dispositivos pelos quais é possível especificar os requisitos relativos à apresentação da informação contida nas páginas da aplicação. Os canais são invocados para visualizar as informações contidas nas páginas durante a navegação, atuando como interpretadores da página. Dependendo do tipo da mídia, os canais invocam um aplicativo para visualizar as informações. Por exemplo, editores de texto podem ser utilizados como dispositivos para visualizar informações textuais, editores gráficos como dispositivos para imagens estáticas e manipuladores de áudio e vídeo como dispositivos para animação.

Para testar e validar a aplicação do HMBS, foi desenvolvido o ambiente HySCharts (*Hyperdocument System based on Statecharts*), que possui recursos para autoria e simulação interativa de aplicações Web (Turine et. al. 1998), e a plataforma WebSCharts (Brito, 2003).

### **2.2.3. HMBS/M**

O *Hypertext Model Based on Statecharts / Method* (HMBS/M), desenvolvido por Carvalho (Carvalho, 1998) e estendido por (Brito, 2003), é um método para apoiar o projeto e o desenvolvimento estruturado de aplicações hipermídia. O HMBS/M é composto por quatro fases, sendo que o produto de cada fase é incrementado ou melhorado na fase posterior. As melhorias introduzidas no(s) modelo(s) de uma das fases devem ser especificadas no(s) modelo(s) da(s) fase(s) anterior (es), permitindo assim um desenvolvimento iterativo. Na Figura 2.3 são apresentadas as fases do método: Modelagem Conceitual, Modelagem Navegacional, Modelagem da Interface, Implementação e Teste.

Na fase de modelagem conceitual o objetivo é analisar o domínio e representá-lo em dois modelos: de classes e de fatias. No modelo de classes são representadas as informações relevantes do domínio e o relacionamento entre elas, e o modelo de fatias enriquece o modelo de classes com a primitiva de modelagem “fatia”, que contém as unidades de informação que serão apresentadas e exploradas durante a navegação. Tais modelos utilizam como base o documento de especificação de requisitos que define os objetivos e os requisitos necessários. O método não oferece técnicas e ferramentas para realizar a tarefa de especificação de requisitos.

Na fase de modelagem navegacional os modelos da fase anterior são reorganizados, considerando os aspectos navegacionais por meio da especificação de três modelos: (1) contextos navegacionais, (2) navegacional de tipos e (3) composição. No modelo de contextos navegacionais são especificados os contextos de navegação que definem as possíveis estruturas de navegação a serem utilizadas para orientar o usuário. Os contextos definidos e os diagramas das fases anteriores são combinados para a criação do modelo navegacional de tipos, que define os caminhos de navegação que orientam o usuário. A seguir é definido o modelo de composição que define as páginas de informação e as possíveis ligações entre as páginas. A partir deste modelo, um protótipo pode ser gerado automaticamente.

Na modelagem de interface é especificada e implementada a interface de cada uma das páginas utilizando a linguagem *Extensible Stylesheet Language Transformations* (XSLT), uma das partes da família de recomendações da *World Wide Web Consortium* (W3C) para apresentação e transformação de documentos *Extensible Markup Language* (XML). O conjunto de todos os templates de interface forma o modelo de estilos. A última fase do método é a de Publicação/Teste onde deve ser cadastrado o conteúdo da aplicação de forma estruturada, tendo como produto a aplicação povoada.

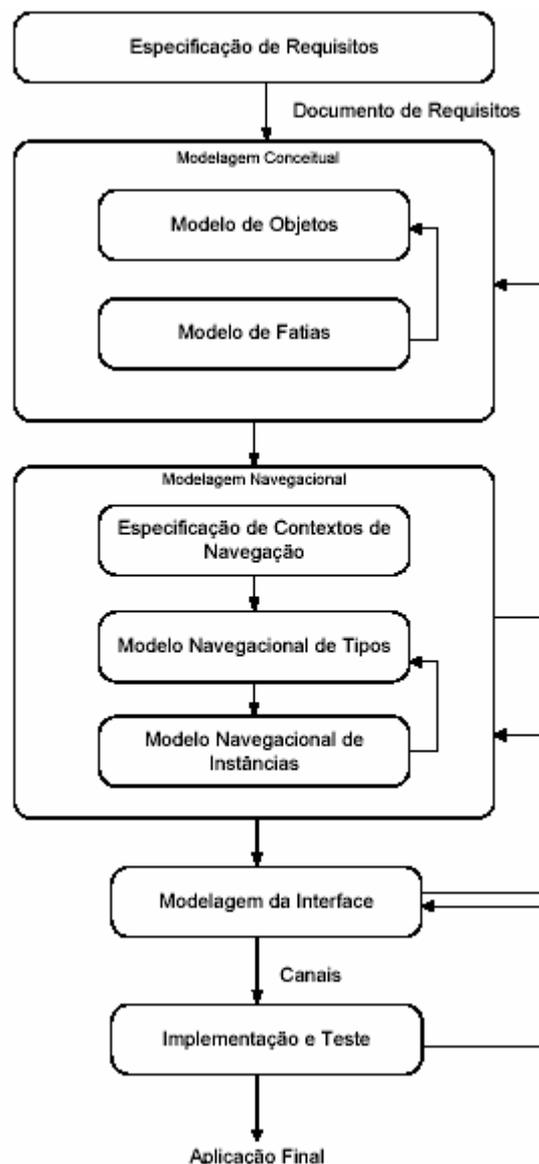


Figura 2.3 Fases do HMBS/M (Carvalho, 1998).

### 2.3. Ambientes e Ferramentas de Apoio

Nos últimos anos, diversos projetos de aplicação das tecnologias computacionais na educação foram propostos e implementados. A maioria desses projetos, geralmente caracterizados por ambientes virtuais de educação, teve como foco principal fornecer

ferramentas síncronas e assíncronas, baseadas na Web, para dar suporte a educação à distância no que tange à criação e gerenciamento de cursos *online*. Dentre as funcionalidades mais comuns desses ambientes pode-se citar ferramentas para disponibilizar materiais didáticos, exercícios e avaliações, e para gerenciar atividades (por exemplo, entrega de trabalhos ou realização de avaliações), e a comunicação entre professores, estudantes e entre os próprios estudantes, o acompanhamento de estudantes, entre outras.

Apesar da variedade de ambientes virtuais de apoio à educação existente na literatura, tais como, WebCT, AulaNet, TelEduc, Tidia-Ae, CoTeia e Moodle, a maioria deles apresenta um conjunto de funcionalidades bastante semelhante, já que foram desenvolvidos para serem utilizados por diversos tipos de usuários.

Nas subseções seguintes são apresentados alguns ambientes ressaltando que a proposta do ambiente ModELi não é ser mais um ambiente nesta categorização e sim permitir a sua integração com as ferramentas já existentes.

Ainda nesta seção ainda é apresentada a ferramenta de geração de páginas *Web*, denominada WebScharts, que utiliza em sua implementação o modelo HMBS e que contribuiu também no desenvolvimento da ferramenta MoDELi.

### **2.3.1. WebCT (Blackboard)**

O WebCT (*Web Course Tools*), proposto por Murray W. Goldberg da University of British Columbia (Canadá) em 1996 (Goldberg, 1996, 1997), é um dos mais conhecidos entre os ambientes computacionais destinados à educação baseada na *Web*. Ao lado de outros ambientes citados na literatura, o ambiente é tido como um dos mais completos para a manutenção de cursos na *Web* por fornecer, aos estudantes e professores, a maioria dos recursos necessários para a realização do curso, incluindo ferramentas de autoria de conteúdos e de gerenciamento de material e de tarefas didáticas.

Apesar de ter sido inicialmente desenvolvido como um projeto de pesquisa, o WebCT se tornou um produto comercial em 1997. Em fevereiro de 2006, foi adquirida pela rival Blackboard Inc. que progressivamente vem trabalhando na marca *Blackboard*, eliminando o nome WebCT.

Existem várias versões do produto disponíveis onde a mais completa é a *Blackboard Academic Suite* que além de suportar as funcionalidades básicas para gerência de

cursos, também oferece recursos para gerência acadêmica, por exemplo, para o controle de várias unidades de uma instituição de forma centralizada.

Nas últimas versões do produto há seis grupos principais de usuários (papéis) que possuem direitos de acesso diferentes sobre as ferramentas do ambiente:

- *Administrador*: é o usuário que administra o servidor do ambiente. Ele tem acesso a um conjunto de páginas para criação, inicialização e remoção de cursos e para controle de senha dos usuários. O administrador cria os cursos e estabelece seus membros, inclusive o *designer*.
- *Designer*: é o usuário que define o projeto gráfico das seções de cursos (cada ocorrência de um mesmo curso é chamado de uma seção daquele curso) e determina as ferramentas a serem utilizadas. Porém, ele não tem acesso às ferramentas de gerenciamento do curso, tais como manipulação de notas ou conteúdo didático.
- *Instrutor*: é o usuário responsável pelo gerenciamento de uma seção de um curso. É esse usuário que manipula o conteúdo didático, cria avaliações e exercícios, estabelece notas e acompanha o progresso dos estudantes, cria grupos de trabalho e manipula as contas dos estudantes da seção do curso, entre outras atividades.
- *Monitores*: são os usuários que auxiliam os professores a gerenciar algumas atividades de uma seção de curso. Eles possuem os mesmos privilégios dos estudantes, mas podem realizar correção de testes e atribuir notas aos estudantes.
- *Estudantes*: são os usuários que realizam o curso. Eles podem visualizar conteúdos, fazer testes, submeter trabalhos, modificar sua página pessoal, etc.
- *Auditores*: são os usuários que participam do curso mas não são avaliados.

O uso do *Blackboard* não requer instalação de nenhum *software* no computador do usuário, pois ele é totalmente baseado na *Web* e executado a partir de um navegador. O ambiente *Blackboard* foi desenvolvido utilizando tecnologias da plataforma J2EE (*Java 2 Enterprise Edition*), sendo portanto independente de plataforma. Por ser um produto

comercial, não é disponibilizada nenhuma documentação mais detalhada sobre a estrutura interna do ambiente.

Entre as ferramentas educacionais oferecidas pelo ambiente pode-se citar a gerência de calendário, autoria dos conteúdos, mecanismos de comunicação (sistema de conferência, *chat*, correio eletrônico), acompanhamento do aluno, suporte para projetos colaborativos (área compartilhada de dados entre os alunos), auto-avaliação (*quiz*), distribuição e visualização de notas, glossário, etc. A título de ilustração, na Figura 2.4 é apresentada uma interface de um curso no *Blackboard* na visão de um Administrador.

O conjunto de funcionalidades oferecidas pelo *Blackboard* permite que esse ambiente virtual seja utilizado tanto como ferramenta de ensino em programas de educação à distância quanto como ferramenta de apoio à sala de aula em cursos presenciais.



Figura 2.4 Interface do ambiente *BlackBoard* acessada pelo Administrador.

### 2.3.2. AulaNet

O AulaNet é um ambiente virtual para criação, participação e administração de cursos baseados na *Web*, cujo desenvolvimento foi iniciado em 1997 no Laboratório de Engenharia de Software (LES) do Departamento de Informática da Pontifícia Universidade Católica do Rio de Janeiro (PUC-RIO) (Aulanet, 2005). De acordo com seus desenvolvedores, o AulaNet foi concebido tendo-se como principal objetivo a colaboração entre os seus participantes, daí a característica de *groupware* do ambiente. (Lucena et al., 1998).

No ambiente Aulanet, existem três atores (Lucena *et al.*, 1998; Lucena *et al.*, 1999):

- *Administrador*: é quem atua na operação do ambiente, facilitando a integração professor/ambiente/estudante. Ele é responsável por cadastrar novos departamentos e/ou instituições, fazer matrícula de estudantes, registrar professores, entre outras tarefas. O administrador também pode definir o *layout* do ambiente que afeta todos os cursos.
- *Professor*: é o criador do curso, participando desde a sua descrição inicial até a entrada do conteúdo didático. Um professor pode ou não ser o responsável pela aplicação do curso, podendo contar com o auxílio de um professor co-autor.
- *Estudante*: é o usuário final do curso que acessa o material didático, faz as atividades, participa das discussões, entre outras tarefas.

A criação de um curso no ambiente Aulanet é feita passo a passo como um *wizard* e inclui a descrição das informações gerais do curso (nome, descrição sumário e ementa), a seleção dos recursos que serão utilizados (há um conjunto pré-selecionado que pode ser modificado), a preparação de um plano de aulas e, posteriormente, a entrada do conteúdo didático.

Os recursos oferecidos pelo ambiente AulaNet (para serem utilizados nos cursos) estão divididos em três categorias (Lucena *et al.*, 1998): mecanismos de comunicação, de coordenação e de cooperação. Os mecanismos de comunicação compreendem os recursos para comunicação entre professores e estudantes e entre os estudantes, tais como: correio eletrônico, grupos de interesse (*newsgroup*), grupos de discussão (*forum*) e debates (bate-papo e videoconferência). Os mecanismos de coordenação envolvem os

recursos para agendamento de tarefas e avaliações, tais como agenda, notícias do curso (*news*), prova, trabalho e exercício. Os mecanismos de cooperação<sup>4</sup>, que incluem bibliografias, webliografias, documentações e *downloads*, correspondem ao instrumental pedagógico utilizado durante o curso (devem ser previamente selecionados pelo professor).

Para o acesso ao curso é necessário que o estudante esteja cadastrado no ambiente e que sua matrícula seja solicitada para cada curso desejado. Os recursos selecionados pelo professor ficam disponíveis para acesso dos estudantes conforme mostra a Figura 2.5 que apresenta a interface principal de um curso oferecido no ambiente AulaNet.



Figura 2.5 Interface principal de um curso no ambiente Aulanet (Aulanet, 2005).

Apesar de toda a interação com o AulaNet ser por meio de um navegador *Web*, alguns recursos mais sofisticados de interação, como a videoconferência, requerem a instalação de programas específicos (*plug-ins*) no computador do cliente.

<sup>4</sup> No AulaNet, o conceito de cooperação deve ser entendido como a preparação do material pelo professor ou, de um ponto de vista mais construtivista, a criação de um espaço para a contribuição de outros participantes (outros professores e os próprios estudantes) (Lucena *et al.*, 1999).

O ambiente Aulanet, como o WebCT, também pode ser utilizado tanto para a educação a distância como para a complementação das atividades de educação presencial ou treinamento de profissionais.

Atualmente o AulaNet está na versão 2.x, e por uma limitação da tecnologia de desenvolvimento, a sua versão atual roda somente num servidor Windows (Fuks et al., 2003). Porém, uma nova versão, a 3.0, está sendo projetada e promete mudar significativamente a estrutura interna desse ambiente.

### **2.3.3. Teleduc**

O TelEduc (TelEduc, 2008) é um ambiente de suporte ao ensino-aprendizagem a distância, cujo desenvolvimento teve início em 1998 no Núcleo de Informática Aplicada à Educação (NIED) da Universidade de Campinas (UNICAMP). Sua proposta inicial teve como alvo o processo de formação de professores para Informática Educativa baseado numa metodologia construída com base na análise de experiências de profissionais formadores do NIED (Rocha, 2002a).

De acordo com seus desenvolvedores, o ambiente TelEduc (Rocha, 2002b) (cujo nome vem da junção das palavras Telemática e Educação) foi desenvolvido tendo como elemento central a ferramenta “Atividades”. Essa abordagem considera que o aprendizado de conceitos de qualquer domínio do conhecimento acontece a partir da resolução de problemas com subsídio de diferentes materiais colocados à disposição do aluno, tais como textos, *software* e instruções de uso.

As ferramentas oferecidas pelo TelEduc são divididas em três grupos: *coordenação*, *comunicação* e *administração* (Rocha, 2002b).

O grupo de *coordenação* engloba as ferramentas que organizam e oferecem suporte para as ações do curso, tais como agenda, histórico e dinâmica do curso (tempo de duração do curso, objetivos, o que é esperado dos estudantes, etc). As ferramentas que disponibilizam material didático de apoio às atividades do estudante, como as ferramentas leituras, material de apoio, atividades e perguntas freqüentes, também fazem parte do grupo de coordenação.

O grupo de *comunicação* inclui ferramentas como correio eletrônico, bate-papo, grupos de discussão, mural de recados e portfólio (área que o estudante tem para compartilhar

informações pessoais) que veiculam a comunicação entre os estudantes com seus formadores e com outros estudantes do curso.

No grupo de *administração* estão as ferramentas que fornecem apoio ao formador para o gerenciamento da parte administrativa do curso:

- Gerenciamento de alunos e formadores (professores) e grupos de estudantes, inscrições, datas de início e término de curso;
- Apoio à autoria, ou seja, inserção de conteúdo para as ferramentas: leituras, material de apoio, atividades;
- Atualização de informações das ferramentas: agenda e dinâmica de curso;
- Seleção de ferramentas disponíveis no curso;
- Acompanhamento dos acessos diários dos estudantes às ferramentas disponíveis no curso.

O ambiente TelEduc conta basicamente com dois tipos de usuários que são os formadores e os estudantes. As ferramentas dos grupos de coordenação e comunicação podem ser acessadas por formadores e estudantes, mas as ferramentas de administração só estão disponíveis ao usuário formador.

Além dos formadores e estudantes, há também a figura do administrador do ambiente TelEduc, que é o responsável pelas funções administrativas do servidor. O administrador é responsável por instalar o servidor do ambiente TelEduc, fazer atualização de versões, criar áreas de cursos para os formadores e, também, emitir relatórios de utilização do ambiente.

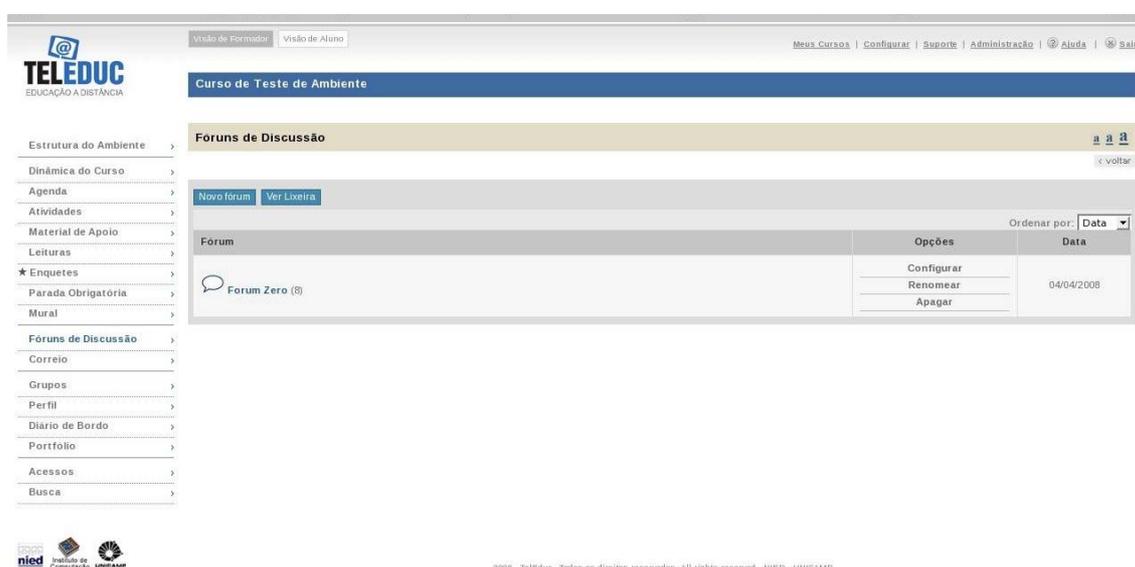
Uma característica marcante no ambiente TelEduc é a sua preocupação com a avaliação formativa<sup>5</sup> dos estudantes. Algumas pesquisas nesse ambiente, visando esse tipo de avaliação, tentam favorecer a análise de dados quantitativos (Romani, 2003) e qualitativos (Ferreira *et al.*, 2003) das participações dos estudantes num curso, principalmente nas ferramentas de comunicação (interação).

---

<sup>5</sup> Entende-se por avaliação formativa o tipo de avaliação que tem como objetivo principal melhorar o processo de ensino-aprendizagem em andamento e não apenas verificar as aprendizagens adquiridas, como acontece na maioria dos ambientes virtuais de educação (Ferreira *et al.*, 2003).

O ambiente TelEduc é um *software* distribuído livremente sob os termos da licença GPL<sup>6</sup> (*GNU General Public License*). Ele foi desenvolvido utilizando-se a linguagem PHP<sup>7</sup> e o gerenciador de banco de dados MySQL<sup>8</sup>, sendo que sua versão atual, a v3.3.5, é suportada por servidores Linux (TelEduc, 2008).

Apesar de ter sido inicialmente projetado para a educação a distância, o TelEduc também tem sido utilizado para apoiar aulas presenciais. A Figura 2.6 apresenta uma interface do ambiente TelEduc sendo acessada por um formador. O acesso ao ambiente é feito por meio de um navegador sem necessidade de instalação de *software* específico no computador do usuário.



**Figura 2.6 Interface de um curso no ambiente TelEduc (TelEduc, 2008).**

### **2.3.4. Tidia-Ae**

O Projeto TIDIA-Ae (Tecnologia da Informação para o Desenvolvimento da Internet Avançada) [(TIDIA-Ae, 2008)] visa o desenvolvimento de ferramentas de suporte e apoio ao ensino e aprendizagem com interações presenciais e a distância, síncronas e assíncronas, amparado pelas redes de alta velocidade. Os objetivos incluem a especificação, projeto e implementação de um amplo conjunto de ferramentas para o ensino à distância, baseado em soluções flexíveis de baixo custo – que teriam como resultado um profundo impacto social por serem baseadas em ferramentas de código livre, que podem ser combinadas e estendidas conforme necessário.

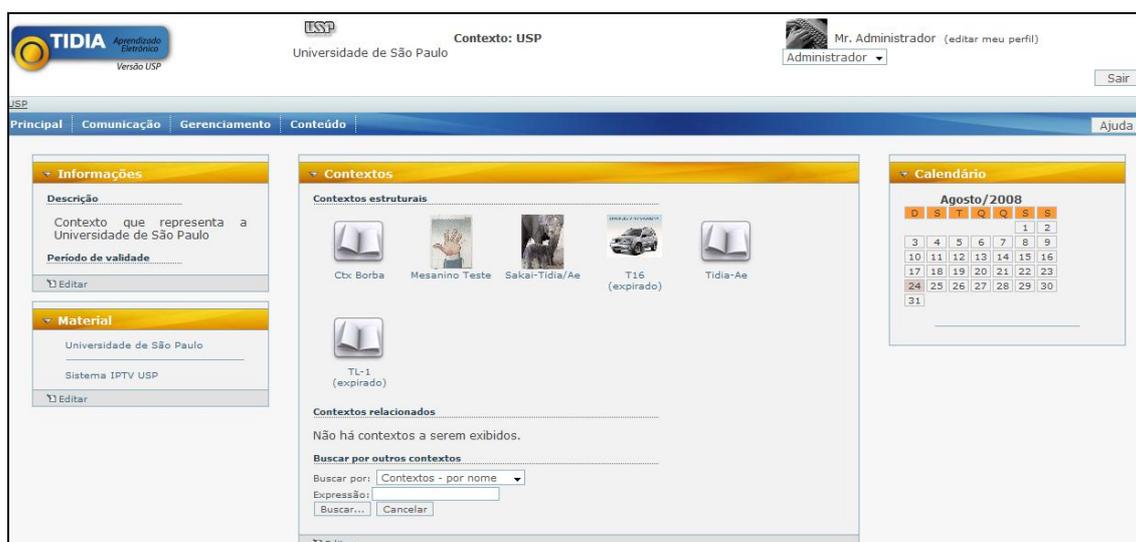
<sup>6</sup> <http://www.gnu.org/copyleft/gpl.html>

<sup>7</sup> Hipertext Processor - <http://www.php.net/>

<sup>8</sup> <http://www.mysql.com>

O desenvolvimento baseado em uma arquitetura de componentes de software facilitará a reaplicação e a extensão de uma infra-estrutura resultante, apontando para um desenvolvimento colaborativo que sustenta pesquisas em formatação, gestão e testes de novas ferramentas que podem então ser incorporadas na infra-estrutura global – com economia em termos de esforços de pesquisa e tempo.

A criação da infra-estrutura do TIDIA-Ae representaria um avanço para o desenvolvimento do Ensino à Distância no estado de São Paulo, integrando muitas realizações que estão atualmente dispersas. Um requisito essencial no projeto é a sustentação do ensino à distância através de código livre - herdando não apenas a tecnologia, mas também a metodologia e a teoria que tradicionalmente caracterizam o desenvolvimento de código livre. O projeto investigará e posicionará o conhecimento da teoria da educação a distância – e também aplicará e desenvolverá técnicas de áreas que incluem Engenharia de Software, Interação Homem-Máquina, Inteligência Artificial e Multimídia entre outras, no formato, implementação e testes de componentes de software (TIDIA-Ae, 2008). A Figura 2.7 ilustra a interface do ambiente TIDIA-Ae.



**Figura 2.7 Interface Tidia-Ae (Tidia-Ae, 2008).**

### **2.3.5. CoTeia**

A CoTeia (COTEIA, 2008) é uma ferramenta de trabalho cooperativo para edição compartilhada e assíncrona de hiperdocumentos na Web (Arruda-Jr. *et al.*, 2002). Ela foi desenvolvida no ICMC-USP com base nas funcionalidades do sistema CoWeb (*Collaborative Web Site*) (Guzdial, 1999), proposto inicialmente por pesquisadores do

Laboratório de Software Colaborativo do Instituto de Tecnologia da Geórgia (GATECH).

No ICMC-USP, a CoTeia é utilizada como uma ferramenta para apoio a aulas presenciais. Cada professor solicita a abertura de um repositório (*Swiki*) na CoTeia (uma área em que ele pode compartilhar hiperdocumentos) e a utiliza para disponibilizar variados tipos de informação: material didático, exercícios, notas de trabalhos e avaliações, trabalhos dos estudantes, etc. Os repositórios criados pelo administrador da CoTeia são preenchidos com páginas *Web* (hiperdocumentos) que podem ser editadas colaborativamente por todos os usuários que tiverem a autorização para edição, ou seja, a senha para desbloqueio da página.

A linguagem para a edição de páginas da CoTeia é baseada em XML (*Extended Markup Language*) (W3C, 2004a). A CoTeia disponibiliza algumas *tags* específicas como `<upload></upload>` (para referenciar arquivos que foram colocados *no* repositório da CoTeia) e `<lnk></lnk>` (para criar *links* entre as páginas da CoTeia). Entretanto, a formatação das páginas pode ser feita por meio das *tags* de formatação da linguagem HTML (*Hypertext Markup Language*) (W3C, 1999a). Na Figura 2.8 é apresentada uma página da CoTeia sendo visualizada pelo usuário final .

Dentre as funcionalidades oferecidas pela CoTeia estão: gerenciamento de repositórios *Swiki* pelo administrador, criação e edição de páginas, visualização de histórico e versão das páginas, busca, *upload* e ligação de documentos, anotações, etc.

Na CoTeia não há a preocupação com educação a distância e nem com aspectos para gerenciamento de cursos na *Web*. Entretanto, suas características permitem que ela seja utilizada para realizar algum tipo de gerenciamento de recursos e atividades didáticas, por exemplo: criação de calendários, *upload* de trabalhos, divulgação de notas e bate-papo (há uma ferramenta de bate-papo integrada à CoTeia).

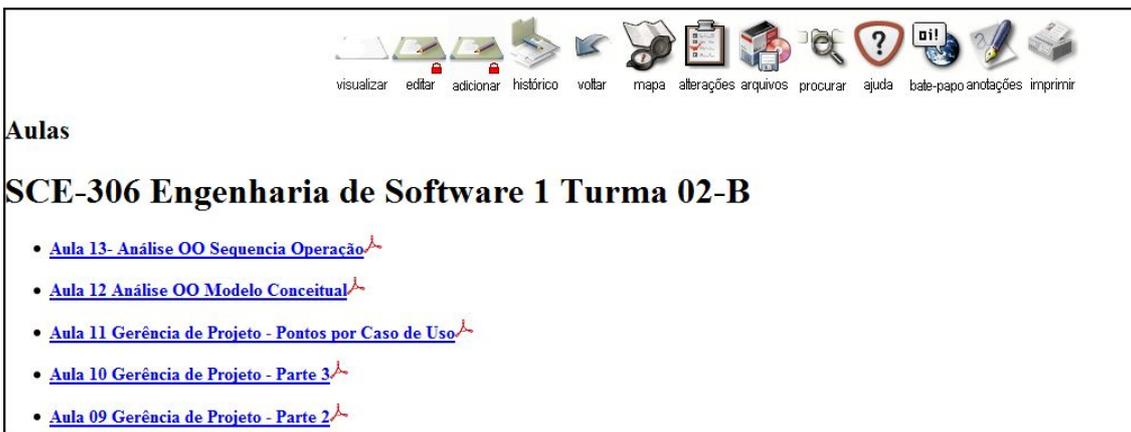


Figura 2.8 Visualização da página da CoTeia.

### 2.3.6. Moodle

Moodle (*Modular Object Oriented Dynamic Learning Environment*) [(MOODLE, 2008)] é um software livre de apoio à aprendizagem, executado num ambiente virtual. Permite gerenciar cursos via Internet, com uso de recursos voltados para publicação de conteúdo, interatividade entre alunos via *Web* e avaliação do processo de ensino-aprendizado, além de permitir um minucioso controle das atividades desenvolvidas pelos usuários via Internet.

O Moodle dispõe de um conjunto de ferramentas que podem ser selecionadas pelo professor de acordo com seus objetivos pedagógicos. Dessa forma pode-se conceber cursos que utilizem fóruns, diários, chats, questionários, textos, objetos de aprendizagem, publicar materiais de quaisquer tipos de arquivos, dentre outras funcionalidades. (Alves & Brito, 2005) Interessante destacar neste ambiente em particular, que tais mecanismos sejam oferecidos ao aluno de forma flexibilizada, ou seja, o professor, além de poder definir a sua disposição na interface, poderá utilizar metáforas que imputem a estas ferramentas diferentes perspectivas, que apesar de utilizarem a mesma funcionalidade, se tornem espaços didáticos únicos. Assim, um simples Chat, pode ser utilizado com um espaço para discussão de conceitos relacionados a um tema, como pode ser chamado de “Ponto de Encontro” e ser utilizado para estimular o estabelecimento de vínculos entre os participantes do curso ou comunidade.

Da mesma forma pode-se criar metáforas para outras ferramentas como o fórum, que tornar-se um portfólio, um repositório de atividades, um relatório de atividades de campo, além de um espaço para discussão de conceitos. Ao mesmo tempo, um glossário pode ser usado com um dicionário, uma FAQ, um pequeno manual, dentre outras

alternativas. É bom lembrar, que o uso de uma ação ou atividade para uma ferramenta não inviabiliza outras possibilidades, pois cada uma delas pode ser inserida no mesmo curso quantas vezes e em que posição ou momento o professor achar necessário (Alves & Brito, 2005).

Nesta perspectiva, o ambiente virtual é concebido como mais do que um simples espaço de publicação de materiais, permeado por interações pré-definidas, mas como um local onde o professor espelhe as necessidades de interação e comunicação que cada contexto educacional lhe apresente em diferentes momentos e situações, conforme ilustra a Figura 2.9.

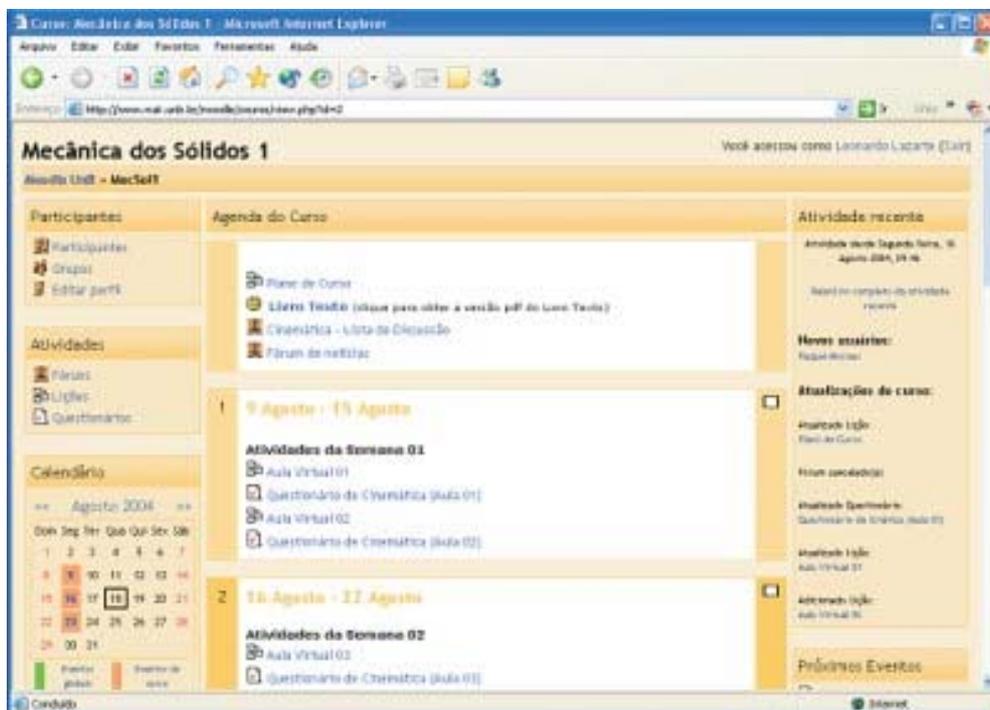


Figura 2.9 Visualização da página do Moodle.

### 2.3.7. WebSCharts

A ferramenta CASE *WebSCharts* (Brito, 2003; Brito, Turine, 2006) foi desenvolvida para auxiliar o projeto e o desenvolvimento estruturado de WebApps segundo o método HMBS/M (*Hypermedia Model Based on Statecharts/Method*) (Carvalho; Oliveira, Masiero, 1999). A ferramenta fornece suporte ao método orientado a objetos HMBS/M que usa o modelo HMBS (Turine, 1998), baseado no formalismo *Statecharts*, para especificar a estrutura organizacional e navegacional de WebApps.

A arquitetura da ferramenta WebSCharts é composta por três camadas principais, denominadas camadas de aplicação, de estrutura e de armazenamento, conforme ilustra a Figura 2.10. A camada de aplicação permite ao projetista criar a estrutura organizacional e navegacional da WebApp e, ao usuário, navegar pela aplicação. Esta camada está subdividida nos ambientes de autoria, de projeto navegacional e de publicação. A camada de estrutura é considerada o núcleo da ferramenta, pois compreende a estrutura interna subjacente à formalização da WebApp com base nos modelos de classes, de fatias, de contextos navegacionais, navegacional de tipos, de composição, de estilos, subjacente ao modelo HMBS (Turine; Oliveira; Masiero, 1999). Toda a representação desses modelos é armazenada em bases de dados gerenciadas por funções da camada de armazenamento, que contém as bases internas da ferramenta capaz de gerar o conteúdo em formato *XML (Extensible Markup Language)* da WebApp.

Para especificar a ferramenta utilizou-se os diagramas da UML. A implementação utilizou-se o ambiente Borland Delphi 6.0 para Microsoft Windows, recompilada no Kylix (KYLIX, 2002) para ambiente Linux, garantindo sua portabilidade. As informações da camada de estrutura são armazenadas no banco de dados relacional FireBird 1.5, e a aplicação é gerada em documentos XML e XSLT, necessitando de um servidor Web, por exemplo, Apache. Um guia para download, instalação, e utilização do WebSCharts está disponível em Brito (Brito, 2003).

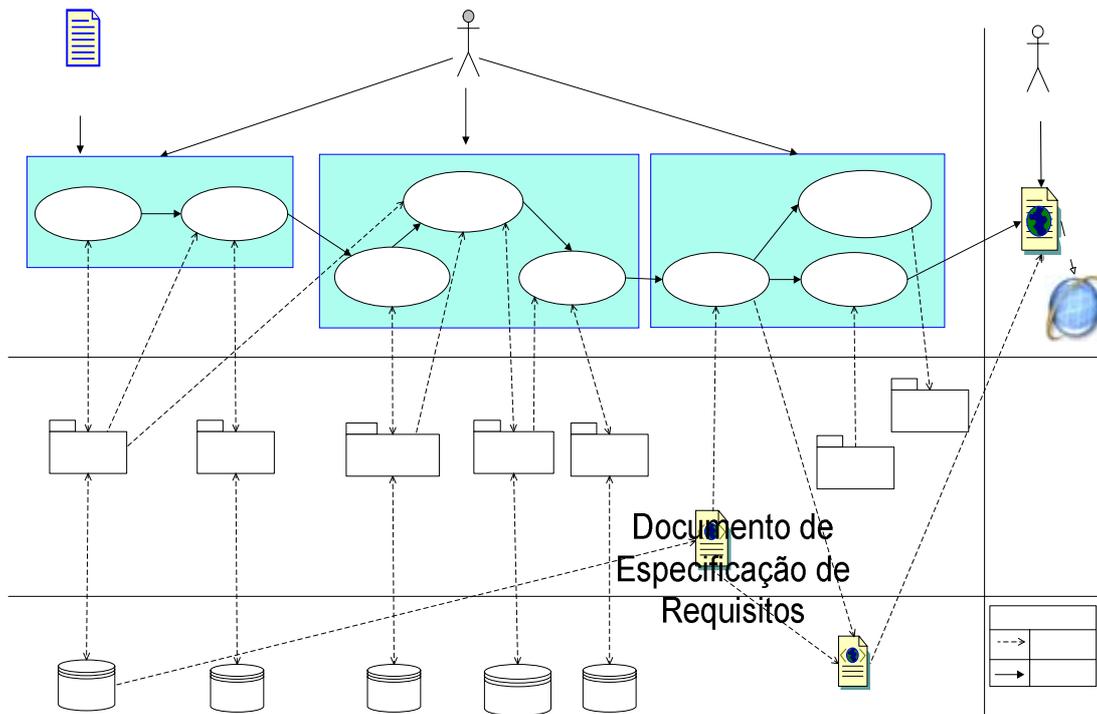


Figura 2.10 Arquitetura da Ferramenta WebSCharts (Brito, 2003).

**Ambiente de Autoria**

Resumidamente, uma WebApp contém classes, fatias, páginas Web, contextos e diversas outras informações. As classes são especificadas por meio de várias atributos (simples ou de perspectiva) e relacionamentos, formando assim o modelo de classes.

**Editor de Classes**

**Editor de Fatias**

Para formar o modelo de fatias, os atributos das classes são agrupados em fatias, sendo que uma classe é composta por várias fatias que se inter-relacionam. O principal elemento de navegação do HMBS/M é a página Web, que é formada por unidades classe e estrutural. Um contexto está sempre associado a uma classe e a seus atributos, podendo estar associada a condições que restringem as instâncias apresentadas no contexto.

A ferramenta contém três ambientes distintos: autoria, projeto navegacional e publicação, com diversos editores gráficos que permitem a especificação de WebApp desde a modelagem conceitual até a geração final da aplicação. No ambiente de autoria o projetista, a partir do documento de especificação de requisitos da aplicação, especifica os modelos da fase de modelagem conceitual do HMBS/M utilizando os editores de classe e de fatias. As primitivas do modelo de classes são armazenadas no repositório “Classe” e as primitivas do modelo de fatias no “Fatias”.

**Modelo de Classes**

**Modelo de Fatias**

No ambiente de projeto navegacional, os editores de modelo de contextos navegacionais, de modelo navegacional de tipos e de modelo de composição são utilizados para especificar os modelos da fase de modelagem navegacional. A partir da

definição dos contextos, o modelo navegacional de tipos é construído. Este modelo é a primeira visão abstrata da estrutura navegacional da WebApp. A especificação das páginas Web e o relacionamento entre estas são definidos no modelo de composição. Desta forma a WebApp pode ser gerada automaticamente no ambiente de publicação. O cadastro/atualização do conteúdo da WebApp é feito diretamente no WebSCharts ou utilizando uma ferramenta de povoamento desenvolvida especialmente para este fim. A aplicação é gerada em documentos XML que são apresentados em navegadores sem a realização de nenhum tipo de conversão ou mapeamento.

Finalmente, para gerar a aplicação e verificar as consistências é utilizado o gerador de aplicação, que por meio das estruturas criadas anteriormente e da associação de templates *XSLT (Extensible Stylesheet Language Transformations)* aos documentos XML gera a WebApp. Para especificar os aspectos comportamentais utiliza-se o modelo HMBS. Assim, uma vez povoada a aplicação, o projetista pode automaticamente gerar uma versão comportamental da WebApp por meio do gerador de especificação comportamental.

## **2.4. Considerações Finais**

Neste capítulo foram apresentados os principais modelos, métodos e ambientes colaborativos utilizados para auxiliar o processo de ensino e de aprendizagem. A partir da revisão bibliográfica, percebe-se que poucas ferramentas automatizadas auxiliam na autoria de conteúdos educacionais. No próximo capítulo será apresentada a abordagem AIM-CID, baseada na técnica de Mapas Conceituais e no HMBS, que permite a modelagem de conteúdos didáticos conforme as perspectivas conceitual, instrucional e didática.

## **3. ABORDAGEM AIM-CID**

### **3.1. Considerações Iniciais**

A definição de mecanismos adequados à elaboração e disponibilização de material didático, incluindo ainda recursos para avaliação e acompanhamento do aprendiz e apoio à condução de atividades práticas, têm sido explorados a fim de fornecer subsídios ao estabelecimento de ambientes de ensino e treinamento efetivos quanto aos objetivos e metas de aprendizados especificados. Módulos educacionais inserem-se nessa perspectiva, correspondendo a unidades concisas de estudo, compostas essencialmente por conteúdos teóricos e práticos, integrados e disponibilizados aos aprendizes por meio de recursos tecnológicos e computacionais (Barbosa, 2004).

O desenvolvimento de módulos educacionais requer ainda que outros aspectos técnicos, somados a fatores gerenciais e organizacionais, também sejam considerados. Em decorrência da diversidade e complexidade dos aspectos envolvidos, a construção de tais módulos exige tempo e recursos, não podendo ser considerada uma atividade trivial. Nesse sentido a definição e a padronização de práticas sistemáticas e disciplinadas de desenvolvimento também são necessárias na construção de módulos educacionais, garantindo, com isso, produtividade ao processo de desenvolvimento e qualidade aos módulos elaborados.

Neste capítulo será apresentada a abordagem AIM-CID (*Abordagem Integrada de Modelagem – Conceitual, Instrucional e Didática*), desenvolvida por Barbosa (2004), que será a base de apoio para a implementação da ferramenta *Web* de modelagem e geração de conteúdos educacionais livres.

### **3.2. Modelagem de Conteúdos Educacionais**

A construção de módulos educacionais pode ser investigada mediante quatro perspectivas correlatas (Barbosa, 2004): (1) estruturação do conhecimento, que envolve a agregação adequada das informações do domínio e a criação de uma estrutura para organizá-las; (2) evolução do conhecimento, que leva em consideração o caráter dinâmico e evolutivo do conhecimento; (3) transferência do conhecimento, referente à forma como este é transmitido

aos aprendizes e (4) reuso do conhecimento, caracterizando os módulos como unidades independentes, passíveis de reutilização em diferentes cenários e contextos de ensino e treinamento, atendendo aos aprendizes com perfis e objetivos de aprendizados distintos.

É interessante observar que essas perspectivas de desenvolvimento estão relacionadas entre si: a estruturação dos conteúdos é fundamental para garantir a transferência, reuso e evolução do conhecimento; a evolução, por sua vez, é decorrente da transferência e do reuso, podendo ainda resultar na reestruturação do conhecimento; o reuso, além de conteúdos estruturados, requer mecanismos de transferência bem definidos (Barbosa, 2004).

A construção de modelos para a representação de conteúdos educacionais requer que uma série de características associadas à atividade de modelagem seja considerada, envolvendo desde a determinação de aspectos específicos do domínio de conhecimento, passando pela definição de atividades práticas e mecanismos para a avaliação do aprendiz, até o estabelecimento de uma ordem pedagógica para a apresentação das informações modeladas. Diante da diversidade de características e aspectos a serem considerados, perspectivas e requisitos de modelagem tornam-se fundamentais (Barbosa, 2004).

No que se refere aos aspectos de modelagem, estabeleceu-se um conjunto de nove requisitos especificamente voltados a essa atividade:

- (1) taxonomia de conceitos, estabelecendo uma estrutura conceitual e hierárquica, ao qual pode ser empregada como base para a definição e organização das informações relevantes ao domínio do conhecimento;
- (2) composição de conceitos, estabelecendo relacionamentos de composição, haja vista que alguns conceitos são melhores compreendidos quando analisados em função de suas partes;
- (3) relacionamentos específicos, essenciais na representação do conhecimento, facilitando a compreensão do assunto que se deseja ensinar;
- (4) decomposição hierárquica, possibilitando o agrupamento de conceitos em módulos, facilitando a sua leitura e compreensão;
- (5) diferenciação entre categorias do conhecimento, possibilitando maior compreensão do domínio;

- (6) ordem pedagógica, aspecto fundamental na elaboração de conteúdos didáticos;
- (7) contextos de aprendizado, a fim de permitir que aprendizes com perfis e objetivos distintos tenham acesso ao mesmo material didático por meio de perspectivas de ensino e treinamento diferenciados;
- (8) história, possibilitando que contextos de aprendizados anteriormente visitados possam ser recuperados; e
- (9) propagação de eventos, fornecendo suporte à sincronização em diversos tipos de aplicações multimídia.

Na Tabela 3.1 são apresentados os requisitos para modelagem de conteúdos propostos, ilustrando sua adequação em relação às perspectivas de desenvolvimento de módulos educacionais discutidas anteriormente. É interessante observar que, embora a maioria dos requisitos identificados enfatize a estruturação e o reuso do conhecimento, as perspectivas de evolução e transferência, por estarem intrinsecamente relacionadas, também podem ser indiretamente beneficiadas (Barbosa, 2004).

<b>Requisitos de Modelagem</b>	<b>Perspectivas de Desenvolvimento</b>
Taxonomia de Conceitos	Estruturação
Composição de Conceitos	Estruturação
Relacionamentos Específicos	Estruturação
Decomposição Hierárquica	Estruturação, Reuso
Diferenciação entre Categorias do Conhecimento	Estruturação, Evolução, Transferência, Reuso
Ordem Pedagógica	Estruturação, Reuso
Contextos de Aprendizado	Estruturação, Reuso
História	Estruturação, Reuso
Propagação de Eventos	Evolução, Reuso

**Tabela 3.1 Requisitos de Modelagem de Conteúdos X Perspectivas de Desenvolvimento de Módulos Educacionais (Barbosa, 2004).**

Além dos requisitos apresentados, perspectivas de modelagem também foram identificadas e investigadas visando a caracterizar os modelos para representação de conteúdos educacionais (Barbosa, 2004):

- Perspectiva Conceitual, que trata especificamente da modelagem dos conceitos relevantes do domínio de conhecimento e da forma com que tais conceitos se inter-relacionam nesse domínio;
- Perspectiva Instrucional, que aborda a modelagem de informações adicionais e elementos complementares sobre o domínio;

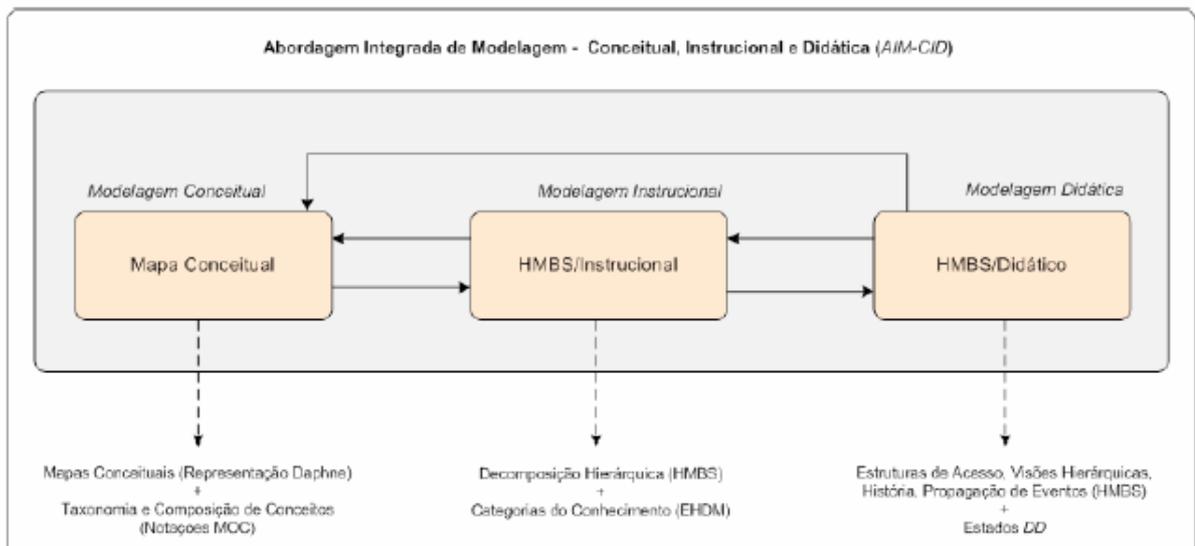
- Perspectiva Didática, responsável por associar os objetos anteriormente modelados, estabelecendo uma seqüência de apresentação entre eles.

Na Tabela 3.2 é ilustrada a relação entre as perspectivas de modelagem definidas e os requisitos propostos.

Perspectivas de Modelagem	Requisitos de Modelagem
Conceitual	Taxonomia de Conceitos Composição de Conceitos Relacionamentos Específicos Decomposição Hierárquica
Instrucional	Diferenciação entre Categorias do Conhecimento
Didática	Ordem Pedagógica Contextos de Aprendizado História Propagação de Eventos

**Tabela 3.2 Perspectivas e Requisitos para Modelagem de Conteúdos Educacionais (Barbosa, 2004).**

A partir das perspectivas e requisitos de modelagem identificados, foi estabelecida a abordagem AIM-CID, caracterizando um conjunto de modelos genéricos para a representação de conteúdos educacionais. Cada modelo aborda aspectos distintos da atividade de modelagem. Na Figura 3.1 são apresentados os principais aspectos da abordagem:



**Figura 3.1 AIM-CID - Abordagem Integrada para Modelagem de Conteúdos Educacionais (Barbosa, 2004).**

A abordagem proposta foi baseada na técnica de Mapas Conceituais e no modelo HMBS a fim de permitir que os modelos conceitual, instrucional e didático fossem representados com maior riqueza de detalhes. Mapas conceituais foram estendidos com primitivas para a representação de taxonomia e composição de conceitos. O modelo HMBS foi estendido considerando diferentes níveis de abstração.

No nível instrucional, foram considerados apenas os aspectos de decomposição hierárquica fornecidos pelo HMBS, acrescidos da possibilidade de representação de diferentes categorias de conhecimento. O HMBS estendido no nível instrucional foi denominado HMBS/Instrucional. No nível didático, foi incorporada ao modelo HMBS a idéia de estados DD (*Dynamically Defined* – Definido Dinamicamente), utilizados como apoio à especificação e implementação dos aspectos de navegação. As extensões estabelecidas no nível instrucional também foram mantidas no nível didático. O modelo HMBS estendido no nível didático foi denominado HMBS/*Didático* (Barbosa, 2004).

### **3.2.1. Modelagem Conceitual**

O Modelo Conceitual consiste em uma descrição de alto-nível do domínio de conhecimento que se deseja ensinar. Sua construção envolve a definição dos conceitos relevantes para a compreensão do domínio e a especificação da forma pela qual os mesmos se relacionam (Barbosa, 2004).

A construção e/ou avaliação de um modelo conceitual deve levar em consideração os seguintes requisitos de modelagem:

- *Taxonomia de Conceitos*: o modelo deve fornecer primitivas que permitam representar relacionamentos de classificação;
- *Composição de Conceitos*: o modelo deve fornecer primitivas que permitam representar relacionamentos de composição;
- *Relacionamentos Específicos*: o modelo deve fornecer primitivas que permitam representar relacionamentos específicos, com semântica dependente do domínio;
- *Decomposição Hierárquica*: o modelo deve fornecer mecanismos que permitam representar a decomposição do domínio de conhecimento em hierarquias.

Barbosa (2004) apresenta em sua tese *guidelines* associadas ao processo de construção dos modelos conceituais:

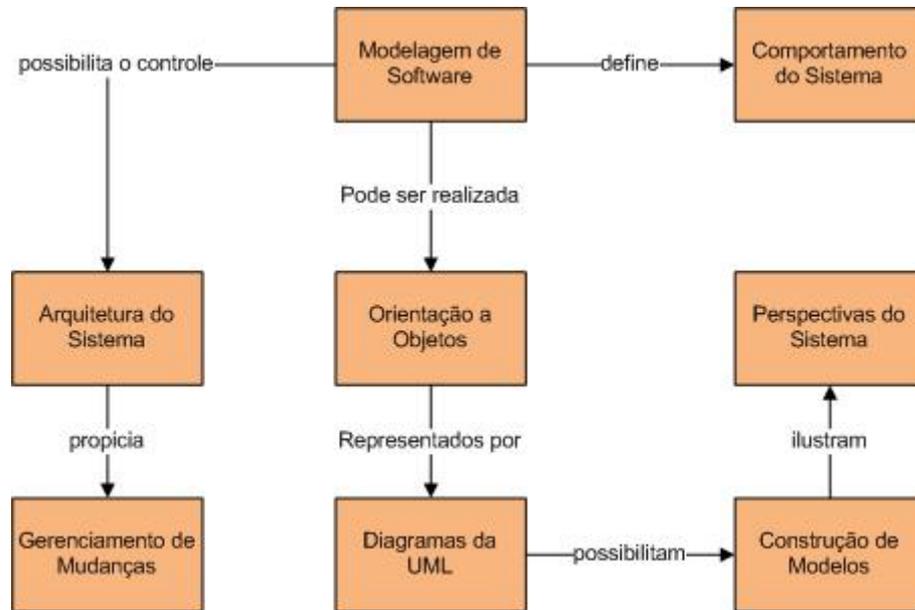
1. Identificar, de maneira genérica, os principais tópicos (assuntos) do domínio de conhecimento que se deseja ensinar, caracterizando unidades coesivas de estudo (módulos).
2. Para cada módulo:

- (a) Identificar o conceito principal referente ao módulo.
- (b) Identificar os conceitos diretamente relacionados ao conceito principal. Substantivos utilizados na descrição do domínio de conhecimento podem auxiliar na identificação dos conceitos.
- (c) Identificar relacionamentos estruturais entre os conceitos selecionados.
- i. Identificar relações de taxonomia (classificação). Verbos e expressões tais como *é classificado em* e *é um tipo de* associando dois conceitos podem auxiliar na determinação das relações da taxonomia.
  - ii. Identificar relações de composição. Verbos e expressões tais como *é formado por*, *é composto de* e *pertence a* associando dois conceitos podem auxiliar na identificação de tais relações.
- (d) Identificar relacionamentos específicos entre os conceitos selecionados. Verbos e expressões tais como *implica em*, *é consequência de*, *utiliza*, *assume*, entre outras, podem auxiliar na identificação de relacionamentos específicos.
- (e) Repetir os passos (2b), (2c) e (2d) até que todos os conceitos relevantes para o módulo em questão tenham sido representados.

Na Figura 3.2 é ilustrado um mapa conceitual geral do domínio de *Modelagem de Software* que evidencia a estrutura do módulo em termos de seus sub-módulos (Barbosa, 2004). Os sub-módulos definidos enfatizam os seguintes aspectos da atividade de modelagem:

- *Modelagem de Software*: introduz conceitos referentes à atividade de modelagem bem como a sua importância dentro do contexto de desenvolvimento de software;
- *Comportamento do Sistema*: caracteriza a atividade de modelagem quanto aos seus objetivos em relação ao software a ser desenvolvido;
- *Arquitetura do Sistema*: apresenta um conjunto de decisões significativas acerca da organização do sistema, os elementos estruturais e suas interfaces, comportamento, dentre outros aspectos inerentes;

- *Gerenciamento de Mudanças*: discorre sobre o processo de verificação e aprovação de pedidos de mudança, e também a coordenação de mudanças aprovadas a serem implementadas;



**Figura 3.2 Estrutura Geral do Módulo Modelagem de Software.**

- *Orientação a Objetos*: apresenta os conceitos sobre o paradigma de análise, projeto e programação de sistemas de *software* baseado na composição e interação entre diversas unidades de software chamadas de objetos;
- *Diagramas da UML*: discorre sobre os diagramas da UML que apresentam as diferentes versões e níveis de detalhamento do sistema;
- *Construção de Modelos*: ilustra as principais técnicas e metodologias que auxiliam na construção dos diversos modelos presentes na UML;
- *Perspectivas do Sistema*: apresenta os modelos que representam diferentes pontos de vista, ou seja, as visões da UML que apresentam foco em ângulos diferentes de um mesmo sistema.

É importante observar que, embora os sub-módulos tenham sido projetados para possibilitar sua utilização de forma independente, relacionamentos entre os conceitos de sub-módulos diferentes podem ser estabelecidos. Na Figura 3.3 é apresentado um modelo conceitual do domínio do conhecimento *Diagramas da UML*, representando os conceitos com as extensões

notações de taxonomia e composição de conceitos. Uma leitura do modelo elaborado é apresentada a seguir:

- UML é formado por diversos diagramas, que podem ser classificados em *Diagrama Estrutural* e *Diagrama Comportamental*;
- O *Diagrama Estrutural* pode ser classificado em *Diagrama de Classe*, *Diagrama de Objeto*, *Diagrama de Pacote*, *Diagrama de Estrutura Composta*, *Diagrama de Componentes* e *Diagrama de Implantação*;
- O *Diagrama Comportamental* pode ser classificado em *Diagrama de Caso de Uso*, *Diagrama de Atividade*, *Diagrama de Estado* e *Diagrama de Interação*;
- O *Diagrama de Interação* pode ser classificado em *Diagrama de Seqüência*, *Diagrama de Comunicação* e *Diagrama de Tempo*.

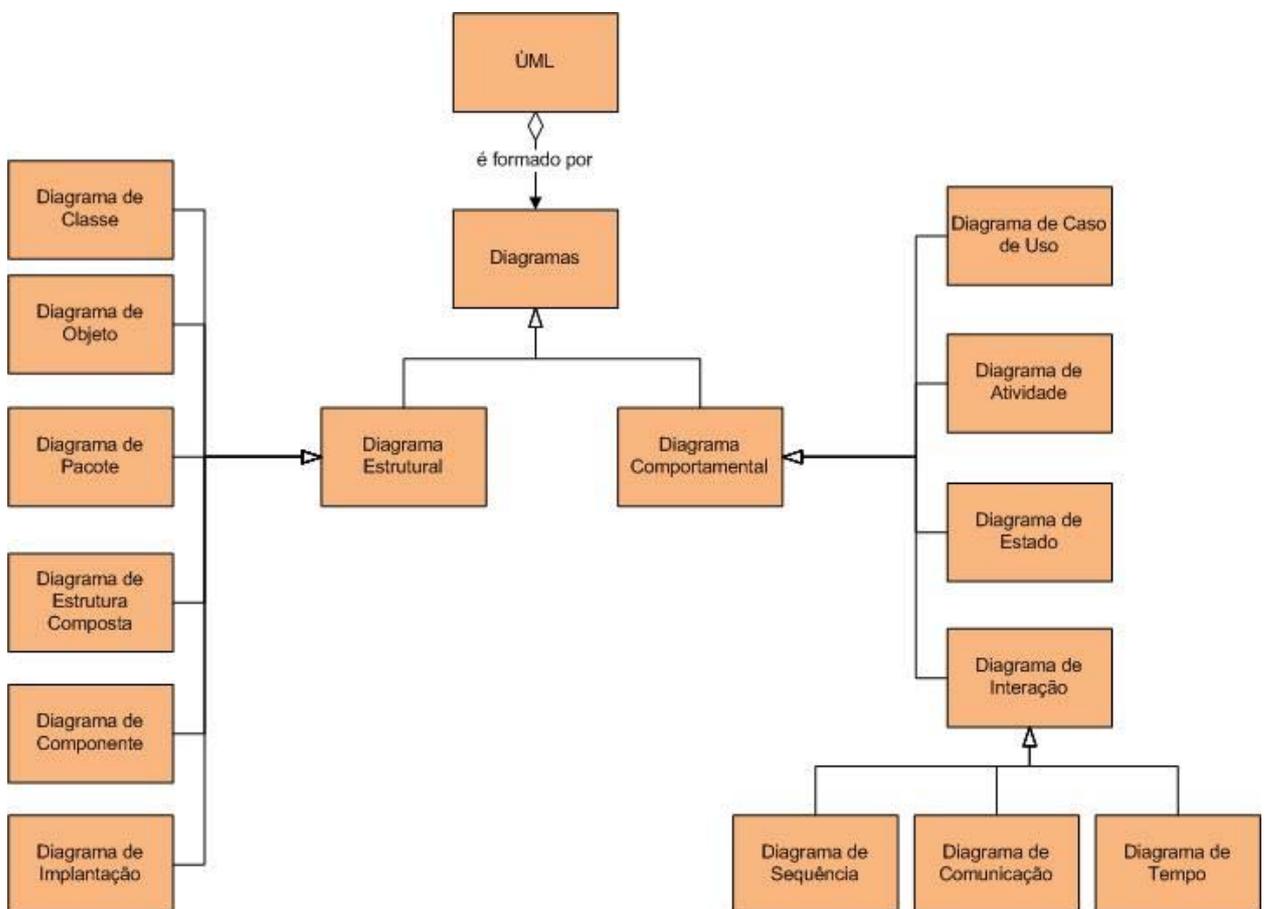


Figura 3.3 Modelo Conceitual do Sub-módulo Diagramas da UML.

### **3.2.2. Modelagem Instrucional**

O Modelo Instrucional tem como objetivo definir os itens de informação<sup>1</sup> (conceitos, fatos, procedimentos e princípios) e elementos instrucionais<sup>2</sup> (exemplos, informações complementares, exercícios e avaliações), associando-os aos conceitos já identificados (Barbosa, 2004).

Salienta-se que foi utilizado a Teoria de Merrill (MERRIL, 1983) conforme sugerida pela abordagem AIM-CID onde são definidos quatro tipos de itens de informação: (1) Fato define parte de informação logicamente associada, como nomes, data e eventos; (2) Conceito define eventos e objetos que compartilham características cujo entendimento é essencial para a comunicação; (3) Procedimento corresponde ao conjunto de passos que visam atingir um objetivo e (4) Princípio apresenta o porquê de determinados acontecimentos.

A abordagem AIM-CID classifica os elementos instrucionais da seguinte forma (Barbosa, 2004): (1) Elementos Explanatórios que correspondem a informações complementares utilizadas na explicação de um dado item de informação (exemplos, dicas, sugestões de estudo, referências); (2) Elementos Exploratórios que permitem que o aprendiz “navegue” pelo domínio de conhecimento, praticando os itens de informação relacionados; (3) Elementos de Avaliação que permitem que se avalie tanto o aprendiz como a efetividade do aprendizado ocorrido.

Barbosa (2004) apresenta o processo de construção de modelos instrucionais com o apoio das seguintes *guidelines*:

1. Para cada conceito representado no modelo conceitual, identificar as informações adicionais relevantes que possam ser associadas ao mesmo. Expressões associadas ao conceito caracterizando nomes, datas, eventos, conjunto de passos, etapas, deduções, entre outros elementos, podem auxiliar na identificação dos itens de informação.
2. Associar os itens de informação entre si, preservando/estabelecendo a estruturação hierárquica entre eles.

---

<sup>1</sup> A abordagem AIM-CID considera que um item de informação é utilizado para tratar, tanto conceitos como os demais tipos de informação adicionais associadas.

<sup>2</sup> A abordagem AIM-CID define que os elementos instrucionais são complementos aos itens de informação previamente identificados.

3. Para cada conceito representado no modelo conceitual, definir elementos instrucionais que possam ser associados ao mesmo.

(a) Definir elementos explanatórios. Concentrar-se em aspectos que exemplifiquem e complementem os conceitos pertinentes ao domínio de conhecimento.

(b) Definir elementos exploratórios. Concentrar-se em aspectos que permitam explorar o domínio de conhecimento de maneira prática.

(c) Definir elementos de avaliação. Concentrar-se em aspectos que permitam avaliar o aprendizado ocorrido tanto em termos gerais, envolvendo o domínio de conhecimento como um todo, como específicos, envolvendo assuntos pontuais pertinentes ao domínio.

4. Associar os elementos instrucionais entre si, preservando/estabelecendo a estruturação hierárquica entre eles.

5. Associar, de forma genérica, os elementos instrucionais aos itens de informação pertinentes. As associações estabelecidas não precisam ser rotuladas.

A fim de exemplificar, na Figura 3.4 é apresentado um modelo instrucional parcial sobre informações relevantes da Modelagem de Software. Observa-se que este modelo deveria enfatizar outros aspectos inerentes ao módulo que foram explicitamente apresentados na modelagem conceitual, tais como Orientação a Objetos, Diagrama da UML, dentre outros.

Para o modelo a seguir foram identificados vários itens de informação (conceitos) relevantes ao domínio do problema. Além disso, foram identificados três elementos explanatórios (exemplos) que seriam interessantes para exemplificar a necessidade de modelar.

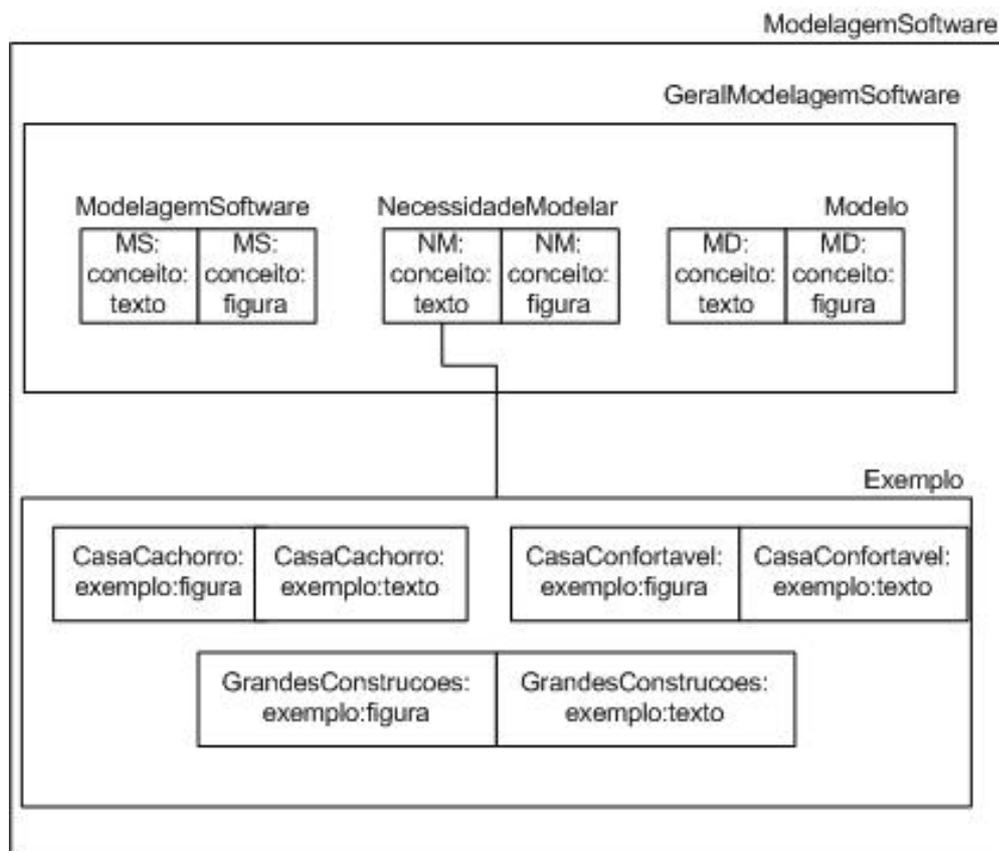


Figura 3.4 Modelo instrucional parcial do Módulo Modelagem de Software.

### 3.2.3. Modelagem Didática

O Modelo Didático tem como objetivo associar os objetos anteriormente modelados, estabelecendo uma seqüência de apresentação entre eles. Basicamente são responsáveis pelo estabelecimento de relações de precedência (pré-requisitos) e relacionamentos didáticos, definindo seqüências de apresentação entre os objetos caracterizados no modelo instrucional (itens de informação e elementos instrucionais). Por relacionamentos didáticos entendem-se relações tais como *complementa*, *exemplifica*, *ilustra*, *motiva*, *exercita* e *avalia*, entre outras. Estruturas de acesso, tais como, índices, roteiros e visitas guiadas, também podem ser utilizadas (Barbosa, 2004).

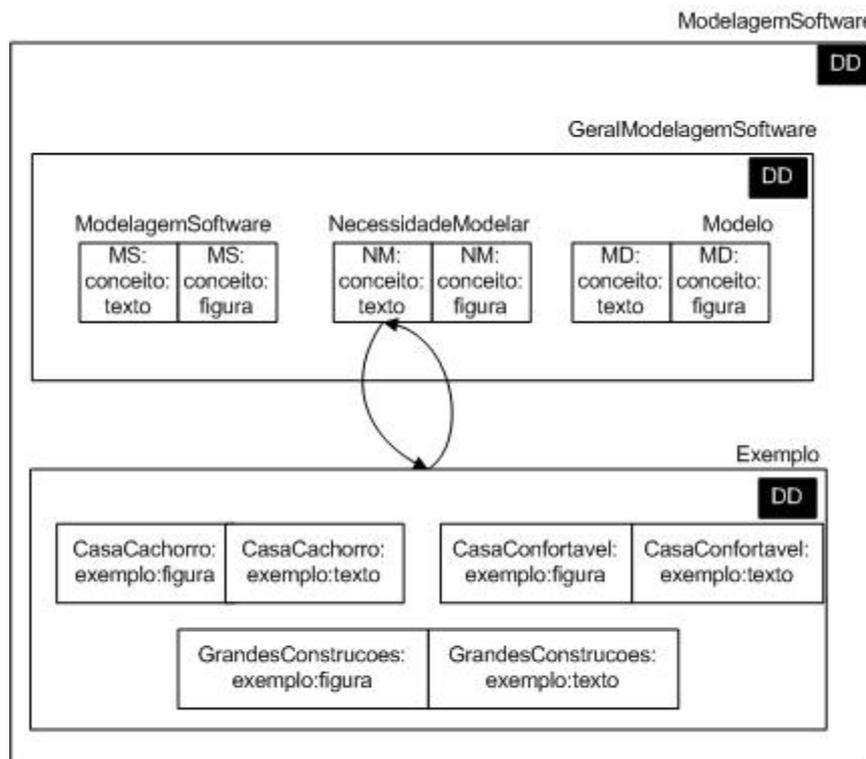
Os modelos didáticos incorporam em sua construção os seguintes requisitos de modelagem:

- *Ordem Pedagógica*: o modelo deve fornecer primitivas que permitam representar relacionamentos de precedência entre as informações do domínio;
- *Contextos de Aprendizado*: o modelo deve dispor de recursos e mecanismos que permitam especificar diferentes contextos de aprendizagem;

- *História*: o modelo deve fornecer primitivas que permitam recuperar contextos de aprendizado anteriormente visitados, bem como armazenar os caminhos percorridos pelo aprendiz durante a exploração do conteúdo;
- *Propagação de Eventos*: o modelo deve fornecer mecanismos que permitam tratar aspectos de concorrência e sincronização entre as diferentes mídias que compõem o conteúdo.

Barbosa (2004) ressalta que diversos modelos didáticos podem ser derivados a partir de um único modelo instrucional, considerando, por exemplo, três tipos de aspectos navegacionais: (1) Especificação Aberta onde são apresentadas todas as possíveis seqüências de apresentação entre os objetos modelados; (2) Especificação Parcialmente Aberta onde somente algumas possibilidades de navegação são consideradas e (3) Especificação Fechada onde os aspectos de navegação são pré-determinados, ou seja, a ordem de apresentação dos objetos é fixa.

Na Figura 3.5 é apresentado o modelo didático baseado na especificação aberta do módulo *Modelagem de Software*. Ressalta-se que somente a especificação aberta foi modelada em razão da limitação do ambiente ModelEli que permite a geração somente de módulos com este tipo de especificação.



**Figura 3.5 Modelagem Didática com Especificação Aberta**

### **3.3. Considerações Finais**

A utilização de métodos, modelos e ferramentas da Engenharia de Software propiciam melhorias significativas na qualidade e produtividade no processo de desenvolvimento dos ambientes educacionais. A abordagem AIM-CID, baseada na técnica de Mapas Conceituais e no HMBS, permite a modelagem conforme as perspectivas conceitual, instrucional e didática.

A modelagem do módulo educacional *Modelagem de Software* foi especificada conforme as diretrizes e requisitos da abordagem AIM-CID mediante a elaboração de conceitos, fatos, princípios, procedimentos, exemplos, informações complementares e exercícios que podem ser apresentados em páginas HTML, documentos textos em formato PDF ou *PowerPoint*/Apresentação ODF.

De modo geral, foram consideradas no processo de desenvolvimento do módulo as perspectivas de estruturação, transferência, evolução e reuso do conhecimento. Em especial, aspectos de estruturação foram enfatizados, sendo construídos modelos conceituais, instrucionais e didáticos para a representação dos conteúdos educacionais pertinentes.

Salienta-se que o módulo apresenta características de especificação e implementação abertas, ou seja, o usuário tem total liberdade para decidir em tempo de execução quais tópicos e assuntos devem ser tratados. No próximo capítulo será apresentado o ambiente ModELi, que visa gerar os conteúdos didáticos apresentados conforme as especificações da abordagem AIM-CID.

## **4. AMBIENTE MODELI**

### **4.1. Considerações iniciais**

No contexto educacional, o desenvolvimento de materiais didáticos de aprendizagem a partir do reuso de materiais de um repositório educacional é fundamental para individualizar o processo de ensino e de aprendizagem. Porém, haja vista que o processo de elaboração destes materiais é demorado e de alto custo, o que se procura promover é a reutilização de materiais e estabelecer parcerias entre instituições para compartilhar conteúdos e serviços.

Assim, muitas pesquisas estão sendo desenvolvidas a fim de alcançar esse compartilhamento de conteúdos. A abordagem AIM-CID, baseada na técnica de mapas conceituais e no modelo HMBS, é uma destas pesquisas que objetiva estabelecer a padronização de processos para o desenvolvimento de módulos educacionais juntamente com os aspectos associados à especialização e instanciação de um processo. Porém, é fundamental a automatização das atividades e fases proposta pela abordagem, e neste caso, o ambiente ModELi contribui para automatizar uma parte destas ações.

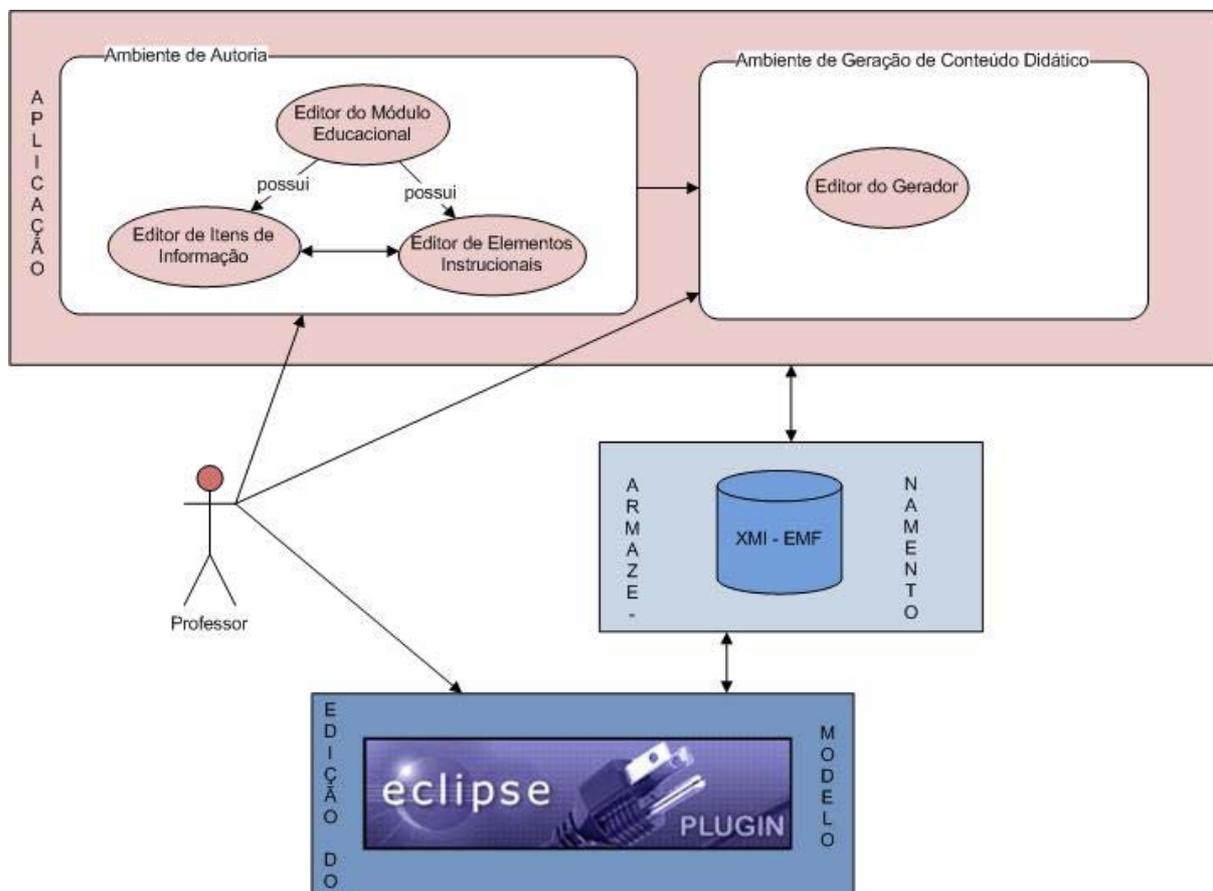
Neste contexto, o presente projeto de implementação do ambiente ModELi visa apoiar a modelagem de conteúdos educacionais segundo a abordagem AIM-CID, com ênfase na geração automática de conteúdos educacionais, personalizados de acordo com interesses pedagógicos, perfil do usuário, características do módulo, entre outros requisitos.

### **4.2. Arquitetura**

#### **4.2.1. Visão Geral**

A arquitetura do ModELi é composta por três camadas principais denominadas camadas de aplicação, de armazenamento e de integração (Modelo de Edição), como ilustrada na Figura 4.1. A camada de aplicação permite ao professor a edição do módulo educacional. Esta camada está subdividida no Ambiente de Autoria e no Ambiente de Geração do Conteúdo Didático. O Ambiente de Autoria possibilita o cadastro dos dados referentes aos módulos educacionais,

descrevendo os itens de informação (fato, conceito, procedimento e princípio) e os elementos instrucionais (informações complementares, exemplos, elementos exploratórios, exercícios, elementos de avaliação e avaliações). O Ambiente de Geração do Conteúdo Didático, através da utilização dos dados do Ambiente de Autoria, possibilita a geração de conteúdos didáticos nos formatos *PDF*, *HTML* e *PowerPoint*.



**Figura 4.1** Arquitetura da ferramenta ModELi.

Toda a representação dos modelos descritos anteriormente é armazenada num banco de dados projetado em tempo de execução (memória) que, posteriormente, gera um arquivo XMI (OMG, 2006) no modelo de classes EMF (EMF, 2007). Este arquivo, por sua vez, pode ser editado no IDE Eclipse através de um *plugin* desenvolvido para edição deste modelo. No Anexo II é apresentado um roteiro descrevendo passo a passo a edição do XMI através da utilização deste plugin no IDE Eclipse.

#### **4.2.2. Especificação**

O ambiente ModELi foi implementado na linguagem de programação JAVA utilizando o padrão MVC (MVC, 2008) mediante a utilização da tecnologia JavaServer Faces (JSF, 2007) conforme

o modelo de classes proposto por Barbosa (2004) com pequenas modificações na navegabilidade dos relacionamentos e nos atributos da bibliografia. As Figuras 4.2, 4.3 e 4.4 ilustram o diagrama de classes do ambiente ModELi. A persistência dos dados foi implementada de acordo com o framework EMF por meio de um meta-modelo gerado conforme este mesmo modelo de classes. A seguir são descritas informações referentes às tecnologias utilizadas.

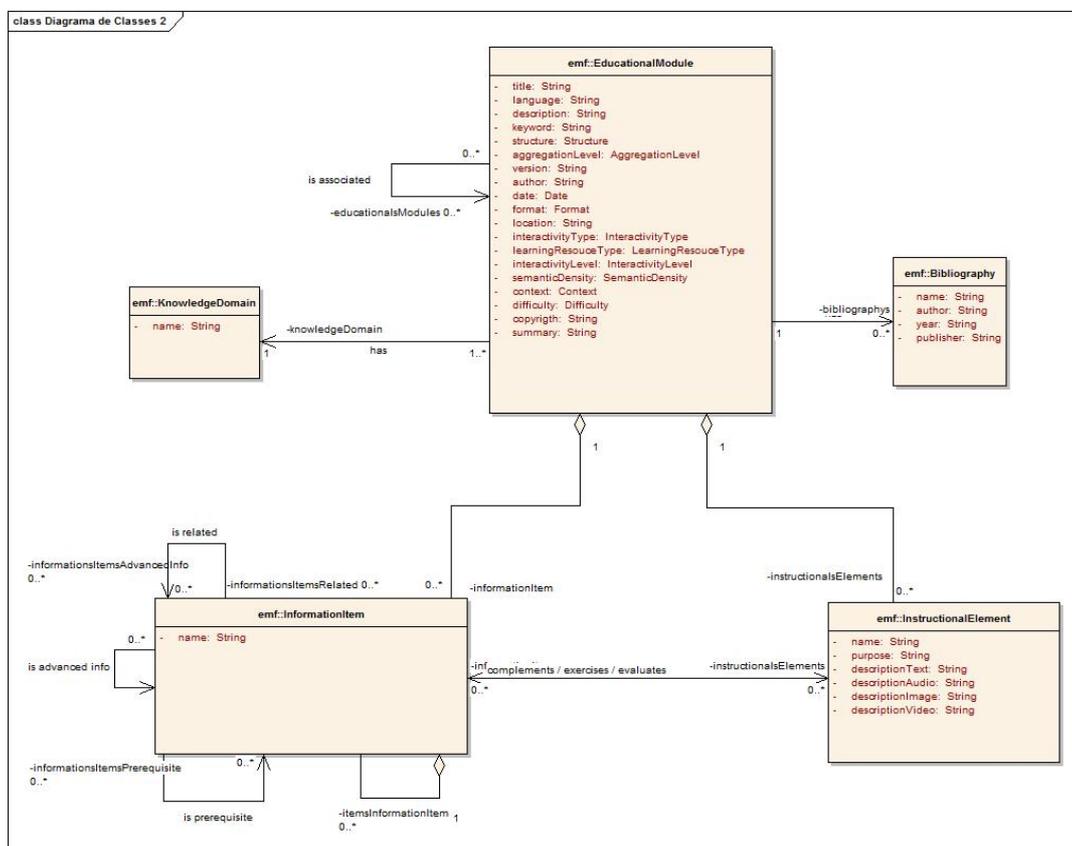


Figura 4.2 Diagrama de Classes ModELi: Parte 1.

**a. JSF – JavaServer Faces**

JSF é uma tecnologia desenvolvida pela Sun Microsystems que incorpora características do padrão de arquitetura *Model-view-controller* (MVC) por meio de um modelo de interfaces gráficas baseado em eventos. Por basear-se no padrão de projeto MVC, uma de suas melhores vantagens é a clara separação entre a visão e regras de negócio.

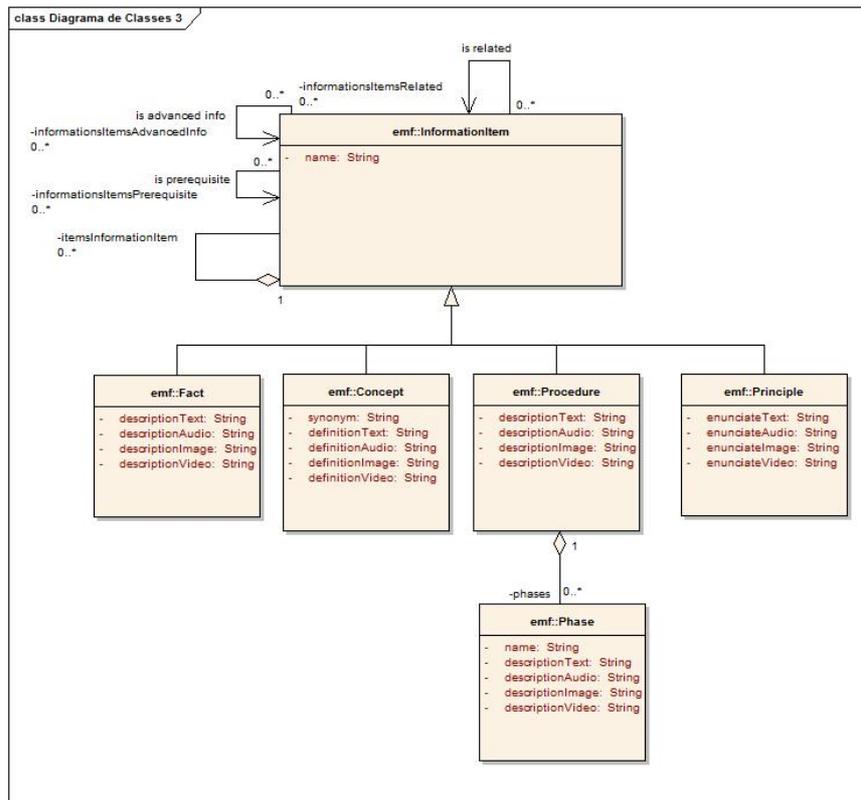


Figura 4.3 Diagrama de Classes ModELi: Parte 2.

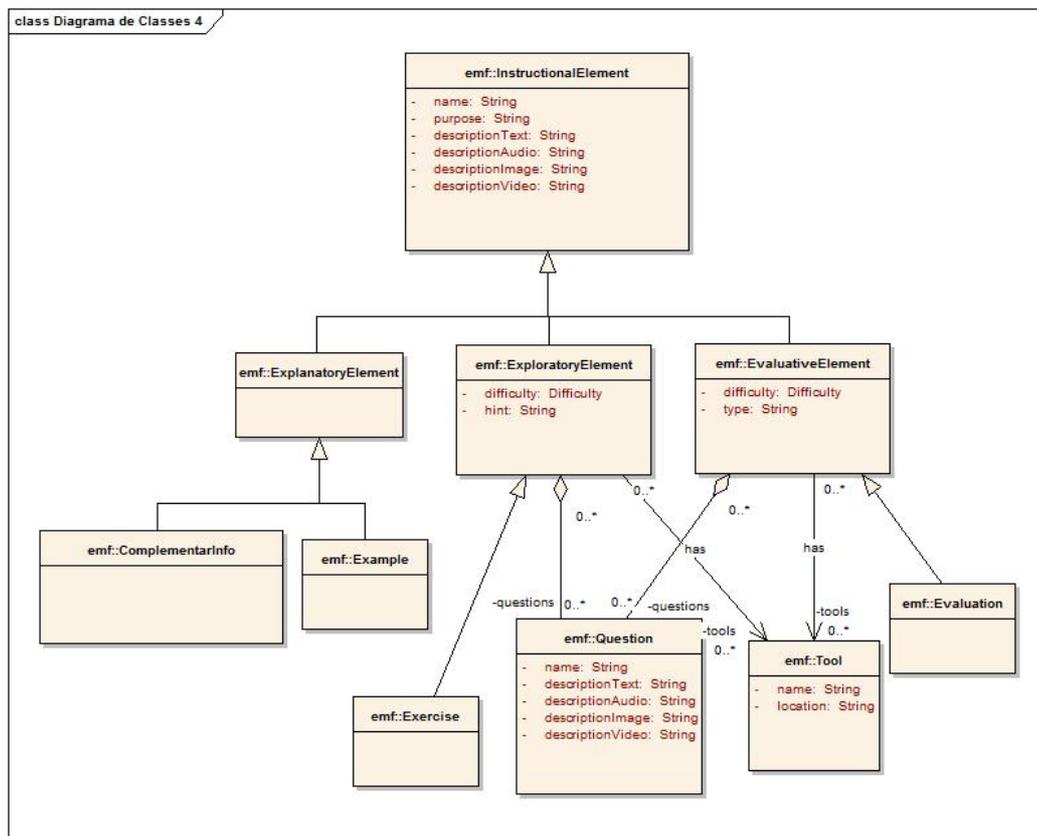
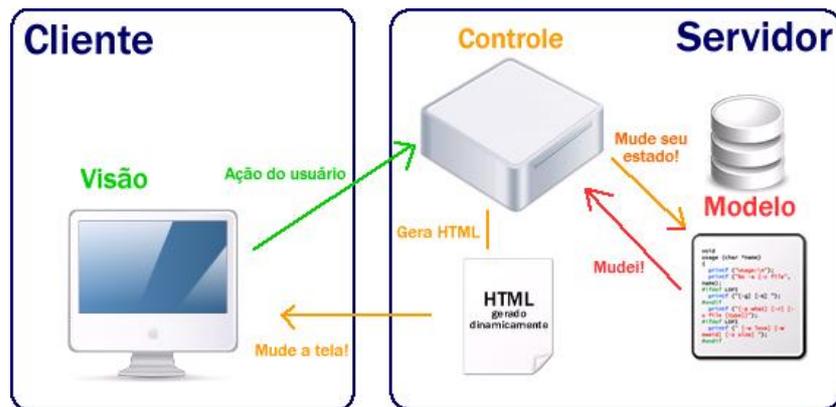


Figura 4.4 Diagrama de Classes ModELi: Parte 3.

O padrão MVC sugere a divisão da aplicação em três camadas: modelo, visão e controle, conforme ilustra a Figura 4.5. O modelo é responsável por representar os objetos de negócio, manter o estado da aplicação e fornecer ao controlador o acesso aos dados. A visão representa a interface com o usuário, sendo responsável por definir a forma como os dados serão apresentados e encaminhar as ações dos usuários para o controlador. Já a camada de controle é responsável por fazer a ligação entre o modelo e a visualização, além de interpretar as ações do usuário e as traduzir para uma operação sobre o modelo, onde são realizadas mudanças e, então, gerar uma visão apropriada.



**Figura 4.5 Padrão de arquitetura MVC.**

O controle desenvolvido na tecnologia JSF é composto por um *Servlet* denominado *FacesServlet*, por arquivos de configuração e por um conjunto de manipuladores de ações e observadores de eventos. O *FacesServlet* é responsável por receber requisições da WEB, redirecioná-las para o modelo e então remeter uma resposta. Os arquivos de configuração são responsáveis por realizar associações e mapeamentos de ações e pela definição de regras de navegação. Os manipuladores de eventos são responsáveis por receber os dados vindos da camada de visualização, acessar o modelo, e então devolver o resultado para o *FacesServlet*.

O modelo representa os objetos de negócio e executa uma lógica de negócio ao receber os dados vindos da camada de visão. Finalmente, a visão é composta por *Component Trees* (hierarquia de componentes), tornando possível unir um componente ao outro para formar interfaces mais complexas.

#### **b. EMF – Eclipse Modeling Framework**

O projeto *Eclipse Modeling Framework* (EMF) é um framework que facilita a geração e construção de ferramentas e outras aplicações baseadas em um modelo de objetos. A partir deste

modelo, descrito na especificação XMI, este framework fornece ferramentas de apoio para produzir um conjunto de classes Java para o modelo, junto com um adaptador do conjunto de classes que permitem visualizar e editar o modelo através de um editor básico, permitindo:

- Geração de esquemas XSD: recomendação oficial da W3C para validação de documentos XML. Esse padrão consegue suprir as limitações da DTD, além de fornecer diversas funcionalidades. Através do XSD é possível construir tipos próprios derivados de tipos mais básicos, realizar relacionamentos entre elementos de dados dentro do XML (de forma similar aos relacionamentos entre tabelas). Desde o seu surgimento, o padrão XSD vem ganhando uma adesão cada vez maior. As ferramentas para trabalhar com XML, classes nas linguagens de programação e diversos aplicativos incluem hoje um amplo suporte ao padrão XSD. Esse efeito acaba por disseminá-lo ainda mais entre a comunidade de desenvolvedores como solução preferida para validação de documentos XML;
- Utilização de componentes *Model Development Tools* (MDT, 2008) tais como *Information Management Metamodel* (IMM), UML2 Tools, dentre outros;
- Implementação de *Service Data Objects* (SDO): framework para desenvolvimento de aplicativos que simplificam a programação J2EE abstraindo a Arquitetura Orientada a Serviços (SOA) mediante a utilização das melhores práticas de programação.

A ferramenta ModELi possui em sua implementação o padrão EMF para persistência dos dados, o que possibilita, por meio de um *plugin* desenvolvido no IDE Eclipse, a edição do modelo. Esta edição pode ser visualizada no Anexo II.

### **4.3. Ambiente**

O ambiente ModELi disponibiliza ao professor no Ambiente de Autoria o cadastro das informações referente ao módulo educacional, seus itens de informação e seus elementos instrucionais. Também estão disponíveis o cadastro de ferramentas, questões e domínio do conhecimento. A Figura 4.6 apresenta a tela inicial do ambiente ModELi.



**Figura 4.6 Consulta do Módulo Educacional.**

Procurou-se disponibilizar inicialmente em todas as suas telas, uma pesquisa pelo *Nome* e pelo *Módulo Educacional* conforme ilustra a Figura 4.7.

Educational Module List				
<input type="button" value="New"/> Name: <input type="text"/> Educational Module: <input type="text" value="None"/> <input type="button" value="Search"/>				
Id	Name	Language	Aggregation Level	Action
10	UML		raw media data fragments	<a href="#">edit</a>
11	Modelagem de software	Português	raw media data fragments	<a href="#">edit</a>

**Figura 4.7 Consulta do Módulo Educacional.**

Na tela de manutenção, como ilustra a Figura 4.8 procurou-se adotar um padrão de ações disponíveis (*Save*, *Remove*, *New* e *Search*), bem como um *help* (imagem ?) para que o usuário possa esclarecer algumas dúvidas durante o cadastro de alguma informação.

Educational Module Crud	
<input type="button" value="Save"/> <input type="button" value="Remove"/> <input type="button" value="New"/> <input type="button" value="Search"/>	
Name: *	<input type="text" value="Modelagem de software"/>
Description: (HTML/PDF) *	<input type="text" value="Modelagem de software é a atividade de construir modelos que expliquem as características ou o comportamento de um software ou de um sistema de software. Na construção do software os modelos podem ser usados na identificação das características e funcionalidades que"/>
Summary: (PPT) *	<input type="text" value="Modelagem de software é a atividade de construir modelos que expliquem as características ou o comportamento de um software. Frequentemente usa algum tipo de notação gráfica e são apoiados pelo uso de Ferramentas CASE."/>
Language:	<input type="text" value="Português"/>
Keyword:	<input type="text" value="Modelagem"/>
Version:	<input type="text" value="1.0"/>
Author:	<input type="text" value="Tânia Ramíres"/>
Date:	<input type="text" value="11/07/2008"/>

**Figura 4.8 Manutenção de um Módulo Educacional**

A aplicação possibilita o cadastro dos itens de informação associado a um determinado módulo. Também é possível associar este item de informação com outros itens de informação ou elementos instrucionais previamente cadastrados, conforme ilustra a Figura 4.9 e a Figura 4.10.

Concept Crud

Save Remove New Search

Educational Module \* ? Modelagem de software

Name: \* ? Modelamos para...

Synonym: ?

Text: (HTML/PDF) ?

- validar nossas teorias ou tentar uma maneira de minimizar os riscos e custos;
- entender melhor o sistema que pretendemos desenvolver;
- construir modelos de sistemas complexos que não podem ser

Slide: (PPT) ?

- validar nossas teorias ou tentar uma maneira de minimizar os riscos e custos;
- entender melhor o sistema que pretendemos desenvolver;
- construir modelos de sistemas complexos que não podem ser

Image: ?

Information Item Instructional Element

Related	
Name	Type
Advanced Info	
Name	Type
Prerequisite	
Name	Type
Composite	
Name	Type
Instructional Element	
Name	Type
Casa de cachorro	Example
Casa confortável	Example
Grandes construções	Example

Figura 4.9 Manutenção do Item de Informação Conceito.

Concept Crud

Save Remove New Search

Educational Module \* ? Modelagem de software

Name: \* ? Modelamos para...

Instructional Element

None  
 Modelagem de software - Casa de cachorro - Example  
 Modelagem de software - Casa confortável - Example  
 Modelagem de software - Grandes construções - Example

Figura 4.10 Manutenção do Item de Informação Conceito associando a outro Item de Informação

A aplicação possibilita o cadastro dos elementos instrucionais associado a um determinado módulo. Também é possível associar este elemento instrucional com outros itens de informação previamente cadastrados, conforme pode ser visualizado a Figura 4.11.

Example Crud

Save Remove New Search

Educational Module \* ? Modelagem de software

Name: \* ? Casa de cachorro

Purpose: ? Ilustrar a necessidade de um modelo

Text: (HTML/PDF) ? Pode ser feito por uma única pessoa;  
Requer:  
- O mínimo de modelagem;  
- Um processo simples;

Slide: (PPT) ? Pode ser feito por uma única pessoa;  
Requer:  
- O mínimo de modelagem;  
- Um processo simples;

Image: ?

Information Item

Name	Type
Modelamos para...	Concept

**Figura 4.11 Manutenção do Elemento Instrucional Exemplo associado a um Item de Informação.**

Em todas as telas de manutenção procurou-se disponibilizar dois tipos de descrição, sendo a primeira para módulos educacionais gerados em formatos HTML ou PDF, pois possibilitam textos maiores sem a preocupação com o layout e, a segunda, para módulos educacionais gerados em formato Slides, sendo extremamente necessário à preocupação com o layout em virtude da limitação do espaço.

O Ambiente de Geração do Conteúdo Didático disponibiliza ao professor a geração dos conteúdos didáticos livres conforme a necessidade, sendo possível especificar o módulo a ser trabalhado bem como os itens de informação e os elementos instrucionais requisitados, conforme ilustra a Figura 4.12.



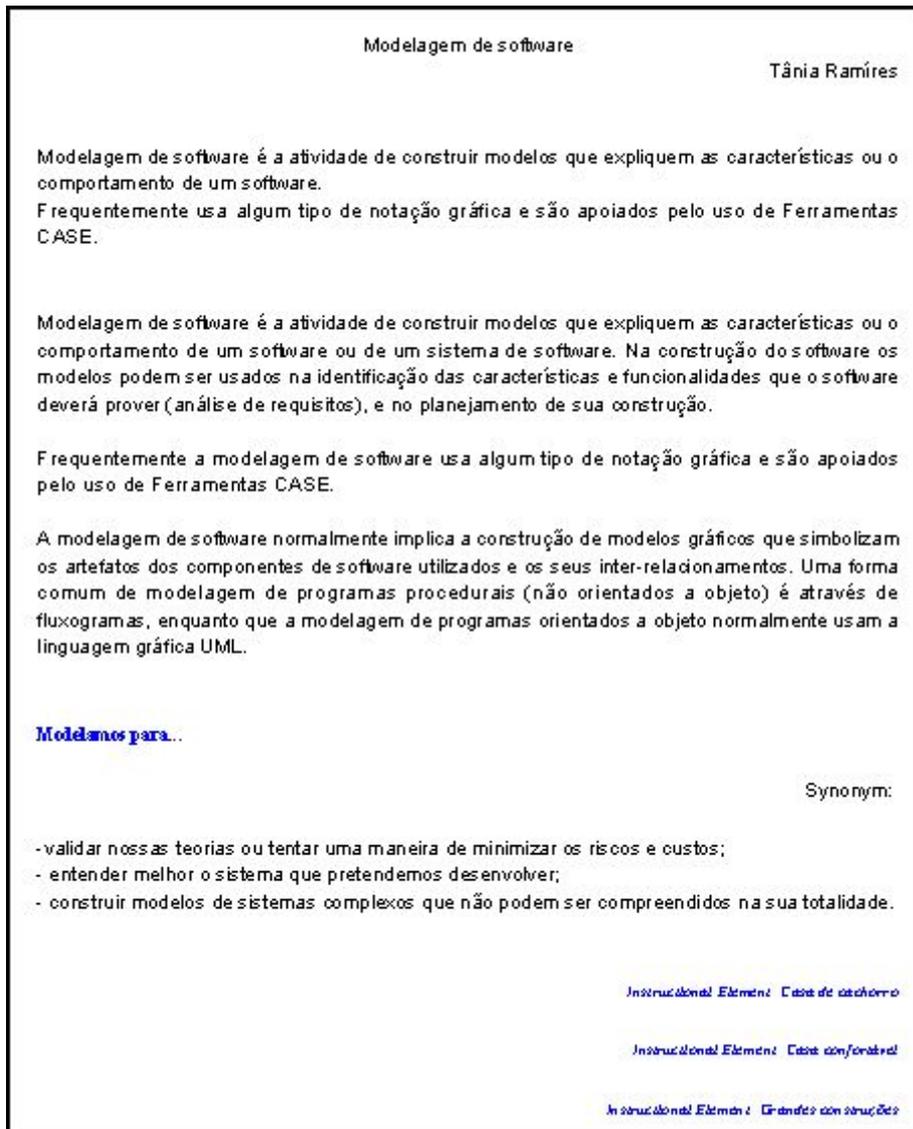


Figura 4.14 Geração do Conteúdo Didático em PDF



Figura 4.15 Geração de Conteúdo Didático em PowerPoint.

#### **4.4. Considerações Finais**

Neste capítulo foram apresentadas a arquitetura e as características principais da ferramenta ModELi, desenvolvida para permitir ao professor criar e gerar conteúdos didáticos livres utilizando a abordagem AIM-CID apresentada no Capítulo 3. A ferramenta auxilia o professor na autoria do material didático em todas as fases da abordagem desde a modelagem conceitual até a geração final do conteúdo, fornecendo materiais em formatos HTML, PDF e *Slide*.

## 5. CONCLUSÃO

### 5.1. Visão Geral da Pesquisa

Para viabilizar e facilitar o processo educacional e a produção de material educacional de forma organizada, sistemática e compartilhada, o presente trabalho teve como objetivo principal implementar um ambiente Web intitulado **ModELi (Módulos Educacionais Livres)** para apoiar a modelagem e a geração de conteúdos educacionais, segundo a modelagem formal baseada em *Statecharts* subjacente à abordagem AIM-CID. Este ambiente, por meio de uma ferramenta de geração automática, permite gerar material didático hipermídia em vários formatos, tais como, arquivos textos (formatos PDF e HTML) e/ou arquivos de apresentações (formatos Slides).

Tendo como entrada os modelos *HMBS/Instrucional* e *HMBS/Didático* da AIM-CID, elaborados em uma linguagem padronizada, mediante o cadastro de tais informações ou através do padrão estabelecido pela OMG denominado XMI (OMG, 2006), o ambiente interpreta e executa tais modelos a fim de gerar os módulos educacionais conforme a vontade e necessidade do autor. A partir deste modelo, diferentes formas de apresentação são geradas, estabelecendo assim um cenário para o desenvolvimento de módulos educacionais.

### 5.2. Contribuições de Pesquisa

Dentro do contexto apresentado, as principais contribuições deste trabalho são:

- Especificação e implementação do ambiente ModELi: uma ferramenta livre;
- Desenvolvimento de um *framework* interno que possibilita que diversas aplicações sejam desenvolvidas a partir dele, possibilitando integração com *frameworks* de persistência, como o Hibernate;
- Geração de material didático nos formatos HTML, PDF e Slides com especificação aberta nos aspectos de navegação;

- Desenvolvimento conforme o *framework* EMF que possibilita a geração de interfaces de desenvolvimento que auxilia na padronização do projeto;
- Importação e exportação dos módulos educacionais para o padrão recomendado pela W3C (XMI);
- Desenvolvimento de um *plugin* que possibilita a edição do modelo numa outro ambiente: IDE Eclipse;
- Proposição de um padrão de Modelagem de Casos de Uso para possibilitar a otimização do processo de desenvolvimento, reduzindo o tempo de escrita para os casos de uso mais comuns (CRUD e Mestre-Detalhe).

### 5.3. Trabalhos Futuros

Dando continuidade às atividades deste trabalho, podem-se destacar como perspectivas futuras de pesquisa as seguintes linhas de atuação:

- Projeto de HCI (*Human-computer interaction*) para o ambiente Modeli para proporcionar ao ambiente melhor interação com o usuário;
- Viabilizar estudos em relação ao projeto *Graphical Modeling Framework* (GMF, 2008) para possibilitar a edição de modelos EMF de forma gráfica;
- Aprofundar os estudos em relação à API de geração de apresentações no formato *PowerPoint*. A API utilizada, denominada Apache POI-HSLF (Apache POI Project, 2007), possui algumas limitações em relação à geração de *links* navegacionais, limitando o ambiente na geração de somente 10 conteúdos em cada módulo;
- Desenvolver um Sistema de Segurança que possibilite a edição e a consulta dos módulos educacionais conforme o Nível de Acesso;
- Detalhar as descrições disponíveis no *Help On-Line* na aplicação;
- Desenvolvimento de uma funcionalidade que possibilite a edição do *Help On-Line*;
- Desenvolvimento de uma funcionalidade que possibilite o Cadastro de Autor;

- Aprofundar a geração dos módulos educacionais para que seja possível a geração de materiais com especificação parcialmente aberta e fechada no que tange aos aspectos navegacionais;
- Possibilitar a integração entre os módulos educacionais cadastrados;
- Possibilitar que o material didático seja gerado em diversos formatos num único material, sendo que, caberia ao autor solicitar o tipo de mídia a ser gerado (HTML, PDF ou Slide);
- Viabilizar estudos em relação à usabilidade do sistema, para permitir que usuário não encontre muitas dificuldades em sua utilização;
- Internacionalização do ambiente para que seja possível adaptá-lo conforme a língua e a cultura de um país;
- Possibilitar a integração com algumas ferramentas de modelagem tais como JUDE, que podem ser utilizados para as modelagens conceitual, instrucional e didática;
- Viabilizar estudos em relação ao projeto *Model Development Tools* (MDT, 2008) para possibilitar a integração com o meta-modelo da UML 2.x gerado conforme o *framework* EMF. Esta integração poderia possibilitar a geração de *Statecharts* baseado no meta-modelo gerado pelo ambiente ModELi.

# ANEXO I - PADRONIZAÇÃO NA MODELAGEM DE CASOS DE USO

## **1.1. Considerações Iniciais**

A utilização de padrões de projeto é considerada uma importante técnica utilizada no desenvolvimento de software que permite a produção de projetos com qualidade e produtividade. Uma forma de otimizar todo o processo de desenvolvimento de software é através da utilização de padrões na Modelagem de Casos de Uso

Sabe-se que escrever um Caso de Uso não é uma tarefa tão simples quanto se imagina. Além disso, a maneira que cada Analista de Sistemas escreve pode impedir a eficácia no desenvolvimento, visto que, um determinado texto pode ter diversas interpretações, o que acarreta muitas vezes no retrabalho do desenvolvimento de alguns Casos de Uso.

Neste projeto, buscou-se a padronização no desenvolvimento de cada Caso de Uso de tal forma que todos os Casos de Uso foram escritos de forma padronizada conforme o funcionamento do ambiente ModELi.

Interessante observar que os estudos referentes à padronização na Modelagem de Casos de Uso foram propostos pela empresa DSF (Desenvolvimento de Sistemas Fiscais) que buscava produtividade na Área de Projetos (Modelagem de Negócio, Modelagem de Casos de Uso e Modelagem de Classes). Mediante estes estudos e mediante os sucessos obtidos tornou-se interessante incorporá-lo ao presente projeto com diversas adaptações para atender a necessidade do ambiente.

Essencialmente foram projetados três tipos de Casos de Uso: (1) CRUD (*Create, Read, Update and Delete*) que corresponde à lógica Ciclo de Vida, dedicada a classes que, por terem grande população (acima de 100 objetos), necessitam de lógicas auxiliares de filtragem e seleção para a edição e alteração de objetos, (2) Mestre-Detalhe que é dedicada a classes que se relacionam em agregações fortes, do tipo "parte-de" (agregação simples) ou "composite" (composição ou

agregação composta) e (3) Outros que é dedicado a casos de uso que não correspondem a CRUD ou Mestre-Detalhe.

Esses Casos de Uso foram escritos de tal forma que, mediante a utilização do conceito de herança, é necessário sobrescrever somente alguns cenários ou requisitos internos de cada Caso de Uso para que ele possa ser desenvolvido, não sendo necessário escrevê-lo por completo. Por exemplo, para demonstrar quais os atributos que estarão disponíveis no Caso de Uso, basta sobrescrever o requisito interno “Lista de atributos”, ao passo que, para demonstrar alguma ação realizada após salvar um determinado registro, basta sobrescrever o cenário classificado como Extensão “Depois salvar”.

Em virtude de limitação de espaço, a seguir é apresentado o Caso de Uso CRUD, escrito de forma padronizada e dois exemplos que utilizaram o conceito de herança do mesmo.

## 1.2. Caso de Uso CRUD

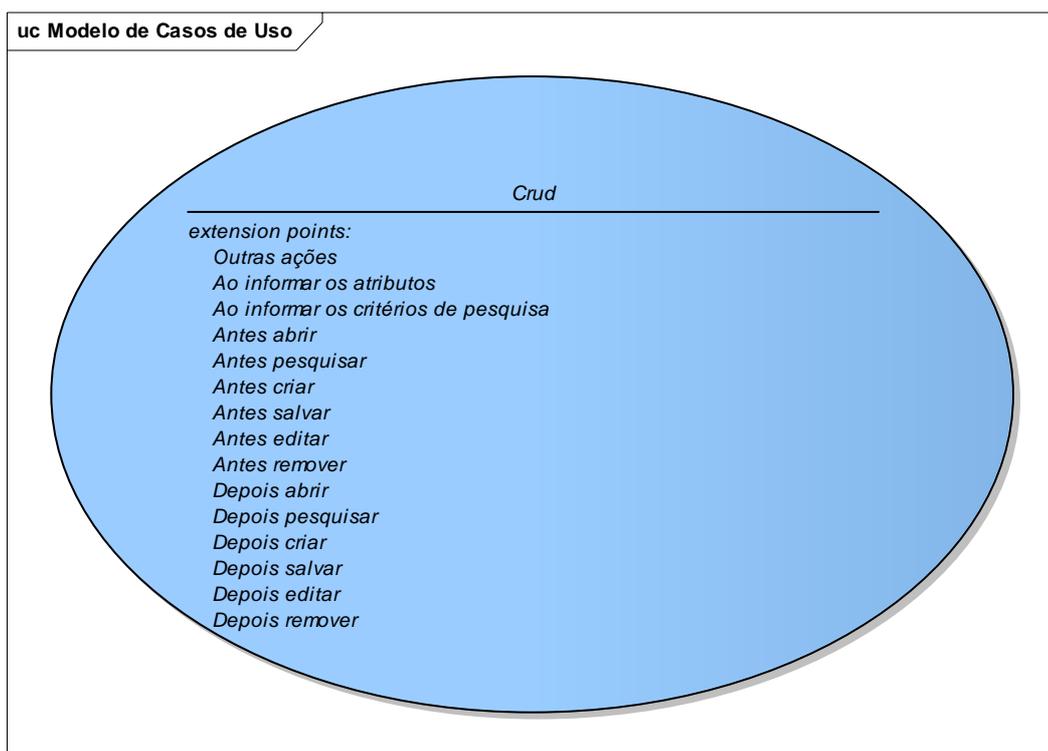


Figura I.1 Caso de Uso CRUD demonstrando os cenários/requisitos que podem ser sobrescritos.

### Descrição breve:

Lógica de “CRUD” (Ciclo de Vida) dedicada a classes que, por terem grande população (acima de 100 objetos), necessitam de lógicas auxiliares de filtragem e seleção para a edição e

alteração de objetos. É o tipo de lógica mais comum em aplicações comerciais, devendo representar, em média, 50% das lógicas.

## Cenários

### Manter cadastro <Fluxo Básico>

1. O caso de uso começa quando o usuário deseja consultar ou alterar um cadastro do sistema;
2. O subfluxo “Antes abrir” é executado;
3. O sistema apresenta as ações ao usuário;
4. O subfluxo “Depois abrir” é executado;
5. O sistema apresenta os critérios de pesquisa conforme o requisito interno “Critérios de pesquisa”;
6. De acordo com o tipo de ação desejada pelo usuário, um dos subfluxos é executado:
  - 6.1. Se o usuário deseja incluir informações, é executado o subfluxo “Novo”;
  - 6.2. Se o usuário deseja consultar informações, é executado o subfluxo “Pesquisar”;
  - 6.3. Se o usuário deseja executar outra ação, é executado o subfluxo “Outras ações”.
7. O caso de uso termina quando o usuário solicita ao sistema a sua finalização.

### Novo <Subfluxo>

1. O usuário seleciona a ação: “Novo”;
2. O subfluxo “Antes criar” é executado;
3. O sistema apresenta a tela de manutenção do cadastro conforme os atributos especificados no requisito interno “Lista de atributos”;
4. O subfluxo “Depois criar” é executado;
5. O sistema apresenta as ações “Novo” e “Pesquisar”;
6. O subfluxo “Ao informar os atributos” é executado;
7. O usuário informa os atributos solicitados;
8. O usuário seleciona a ação: “Salvar”;
9. O subfluxo “Antes salvar” é executado;
10. O sistema salva os dados informados;
11. O subfluxo “Depois salvar” é executado;
12. O sistema exibe a mensagem M1 do requisito interno “Lista de mensagens”;
13. O sistema apresenta as ações ao usuário;

14. De acordo com o tipo de ação desejada pelo usuário, um dos subfluxos é executado:
  - 14.1. Se o usuário deseja incluir informações, é executado o subfluxo “Novo”;
  - 14.2. Se o usuário deseja consultar informações, é executado o subfluxo “Pesquisar”;
  - 14.3. Se o usuário deseja remover informações, é executado o subfluxo “Remover”;
  - 14.4. Se o usuário deseja executar outra ação, é executado o subfluxo “Outras ações”.
15. Este subfluxo termina.

#### Pesquisar <Subfluxo>

1. O subfluxo “Ao informar os critérios de pesquisa” é executado;
2. O usuário informa os critérios de pesquisa;
3. O usuário seleciona a ação: “Pesquisar”;
4. O subfluxo “Antes pesquisar” é executado;
5. O sistema pesquisa os registros conforme os critérios de pesquisa informados;
6. O subfluxo “Depois pesquisar” é executado;
7. O sistema apresenta o resultados encontrados conforme a Lista de atributos “Resultados da pesquisa”
8. O sistema apresenta ao usuário a ação “Editar” para cada registro encontrado;
9. De acordo com o tipo de ação desejada pelo usuário, um dos subfluxos é executado:
  - 9.1. Se o usuário deseja incluir informações, é executado o subfluxo “Novo”;
  - 9.2. Se o usuário deseja consultar informações, é executado o subfluxo “Pesquisar”;
  - 9.3. Se o usuário deseja editar informações, é executado o subfluxo “Editar”;
  - 9.4. Se o usuário deseja executar outra ação, é executado o subfluxo “Outras ações”.
10. Este subfluxo termina.

#### Editar <Subfluxo>

1. O usuário seleciona a ação: “Editar”;
3. O subfluxo “Antes editar” é executado;
4. O sistema apresenta a tela de manutenção do cadastro com os atributos preenchidos especificados no requisito interno “Lista de atributos”;
5. O sistema apresenta as ações “Novo”, “Pesquisar” e “Remover”;
6. O subfluxo “Depois editar” é executado;
7. O subfluxo “Ao informar os atributos” é executado;
8. O usuário informa os atributos solicitados;

9. O usuário seleciona a ação: “Salvar”;
10. O subfluxo “Antes salvar” é executado;
11. O sistema salva os dados informados;
12. O subfluxo “Depois salvar” é executado;
13. O sistema exibe a mensagem M1 do requisito interno “Lista de mensagens”;
14. O sistema apresenta as ações ao usuário;
15. De acordo com o tipo de ação desejada pelo usuário, um dos subfluxos é executado:
  - 15.1. Se o usuário deseja incluir informações, é executado o subfluxo “Novo”;
  - 15.2. Se o usuário deseja consultar informações, é executado o subfluxo “Pesquisar”;
  - 15.3. Se o usuário deseja remover informações, é executado o subfluxo “Remover”;
  - 15.4. Se o usuário deseja executar outra ação, é executado o subfluxo “Outras ações”.
16. Este subfluxo termina.

#### Remover <Subfluxo>

1. O usuário seleciona a ação: “Remover”;
2. O subfluxo “Antes remover” é executado;
3. O sistema exclui o(s) registro(s) selecionado(s);
4. O subfluxo “Depois remover” é executado;
5. O caso de uso retorna ao passo 2 do fluxo básico exibindo a mensagem M3 do requisito interno “Lista de mensagens”;

#### Campos obrigatórios <Alternativo>

1. No passo 10 do subfluxo “Novo” e no passo 11 do subfluxo “Editar”, o sistema verifica se os campos obrigatórios foram informados. Caso contrário, o sistema exibe a mensagem M2 do requisito interno “Lista de mensagens “ para todos o primeiro campo não informado, substituindo o texto “{0}” pelo rótulo do campo.
2. Se o fluxo executado for o “Novo”, o caso de uso retorna ao passo 7 deste subfluxo. Caso contrário, o caso de uso retorna ao passo 8 do subfluxo “Editar”.

#### Registros não encontrados <Alternativo>

1. No passo 7 do subfluxo “Pesquisar”, o sistema verifica que nenhum registro foi encontrado na pesquisa.

2. O caso de uso retorna ao passo 2 do fluxo básico exibindo a mensagem M4 do requisito interno “Lista de mensagens”;

Outras ações <Extensão>

Ao informar os atributos <Extensão>

Ao informar os critérios de pesquisa <Extensão>

Antes abrir <Extensão>

Antes pesquisar <Extensão>

Antes criar <Extensão>

Antes salvar <Extensão>

Antes editar <Extensão>

Antes remover <Extensão>

Depois abrir <Extensão>

Depois pesquisar <Extensão>

Depois criar <Extensão>

Depois salvar <Extensão>

Depois editar <Extensão>

Depois remover <Extensão>

## **Requisitos internos**

## Pontos de extensão

P1.

## Lista de mensagens

M1 – Registro salvo com sucesso.

M2 – {0} é obrigatório.

M3 – Registro removido com sucesso.

M4 – Nenhum registro que atende aos critérios informados foi encontrado.

## Lista de atributos

## Critérios de pesquisa

## Resultados da pesquisa

## **Restrições**

### Pré-Condição

### Pós-Condição

### 1.3. Caso de Uso Manter Questão

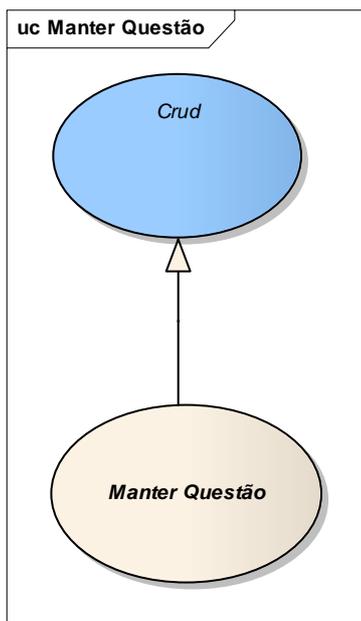


Figura I.2 Modelo de Caso de Uso: Manter Questão.

#### Descrição breve:

Caso de uso para manutenção das questões utilizadas no apoio aos elementos instrucionais.

#### Requisitos internos

##### Lista de atributos

- Nome: Obrigatório; Nome da questão
- Texto (HTML/PDF): Opcional; Descrição textual que será exibido em módulos gerados em HTML ou PDF
- Slide (PPT): Opcional; Descrição textual que será exibido em módulos gerados em *PowerPoint*
- Imagem: Opcional; Imagem que será exibida em todos os módulos gerados

##### Critérios de pesquisa

- Nome: Opcional. Pesquisar de forma igual
- Módulo educacional: Opcional. Pesquisar de forma igual

##### Resultados da pesquisa

- Nome
- Texto (HTML/PDF)

#### Restrições

Pré-Condição

## 1.4. Caso de Uso Manter Conceito

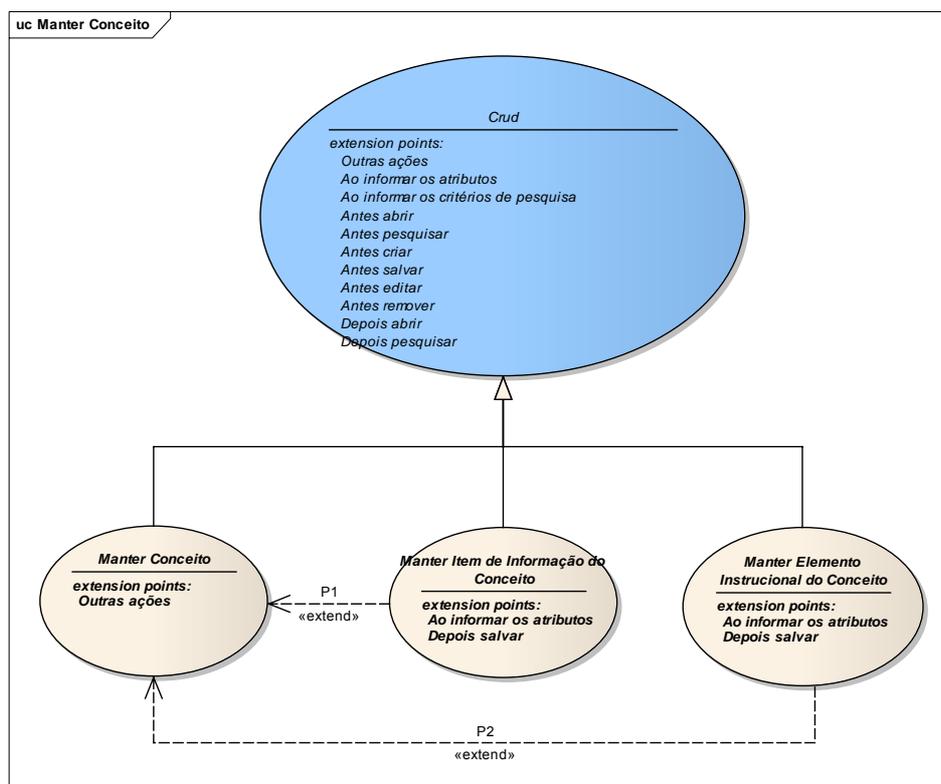


Figura I.3 Modelo de Caso de Uso: Manter Conceito.

### Descrição breve:

Caso de uso para manutenção dos itens de informação denominados de conceitos.

### Cenários

Outras ações <Extensão>

1. Se o usuário deseja adicionar ou remover Itens de Informação ao Conceito, o sistema executa o caso de uso P1, através da ação “Item de Informação”.
2. Se o usuário deseja adicionar ou remover Elementos Instrucionais ao Conceito, o sistema executa o caso de uso P2, através da ação “Elemento Instrucional”.

### Requisitos internos

Lista de atributos

- Módulo Educacional: Obrigatório; Módulo ao qual o conceito pertence; Exibir numa Lista de Seleção
- Nome: Obrigatório; Nome do conceito
- Sinônimo: Opcional; Sinônimo do conceito
- Texto (HTML/PDF): Opcional; Descrição textual que será exibido em módulos gerados em HTML ou PDF
- Slide (PPT): Opcional; Descrição textual que será exibido em módulos gerados em *PowerPoint*
- Imagem: Opcional; Imagem que será exibida em todos os módulos gerados
- Lista de Itens de Informação Relacionados: exibir em modo leitura o Nome e o Tipo (Item de Informação)
- Lista de Itens de Informação Informações Avançadas: exibir em modo leitura o Nome e o Tipo (Item de Informação)
- Lista de Itens de Informação Pré-requisitos: exibir em modo leitura o Nome e o Tipo (Item de Informação)
- Lista de Itens de Informação Composição: exibir em modo leitura o Nome e o Tipo (Item de Informação)
- Lista de Elementos Instrucionais: exibir em modo leitura o Nome e o Tipo (Elemento Instrucional)

### Critérios de pesquisa

- Nome: Opcional. Pesquisar de forma igual
- Módulo educacional: Opcional. Pesquisar de forma igual

### Resultados da pesquisa

- Módulo educacional
- Nome
- Tipo (Item de Informação)

### Pontos de extensão

- P1. Manter Item de Informação do Conceito
- P2. Manter Elemento Instrucional do Conceito

## Restrições

### Pré-Condição

Usuário logado.

### **1.5. Considerações Finais**

A utilização da padronização na modelagem de casos de uso tem oferecido soluções flexíveis para problemas comuns da modelagem de software, que de outra forma seriam resolvidos apenas de acordo com a experiência e intuição de cada um. Cada padrão “é constituído por diversas partes, incluindo aplicabilidade, os objetivos, a estrutura da solução e implementações de demonstração”. No entanto, a sua utilização nem sempre alcança da melhor forma os objetivos do padrão, por vários motivos como o domínio do problema.

Todavia, a utilização da padronização neste projeto auxiliou a redução da complexidade e tempo de modelagem dos Casos de Uso promovendo a reutilização dos esforços de modelagem. A utilização de padrões também aumentou a confiabilidade da aplicação uma vez que faz uso de padrões que visaram projetar o domínio do problema conforme o ambiente ao qual o ModELi foi desenvolvido.

# ANEXO II – PLUGIN PARA EDIÇÃO DO MODELO NO ECLIPSE

## 2.1. Considerações Iniciais

O presente projeto apresenta o desenvolvimento de um plugin desenvolvido na IDE Eclipse para que seja possível editar o modelo através de outra ferramenta e posteriormente exportá-lo para o ambiente Modeli ou para qualquer outra aplicação que possa trabalhar com o meta-modelo desenvolvido.

A seguir é apresentado um roteiro para edição de um módulo educacional.

## 2.2. Download e Instalação do Plugin

A aplicação Modeli disponibiliza um *link* para que seja possível realizar o *download* do *plugin* e o *download* do arquivo XMI que permite a edição do modelo, conforme ilustra a Figura 6.4.

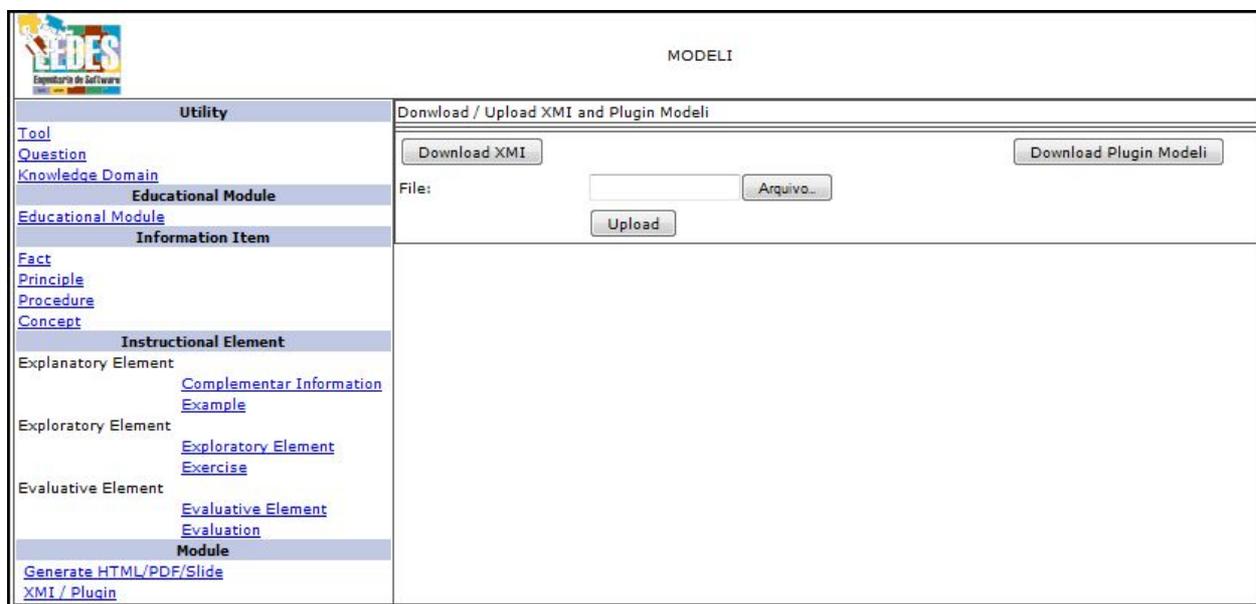
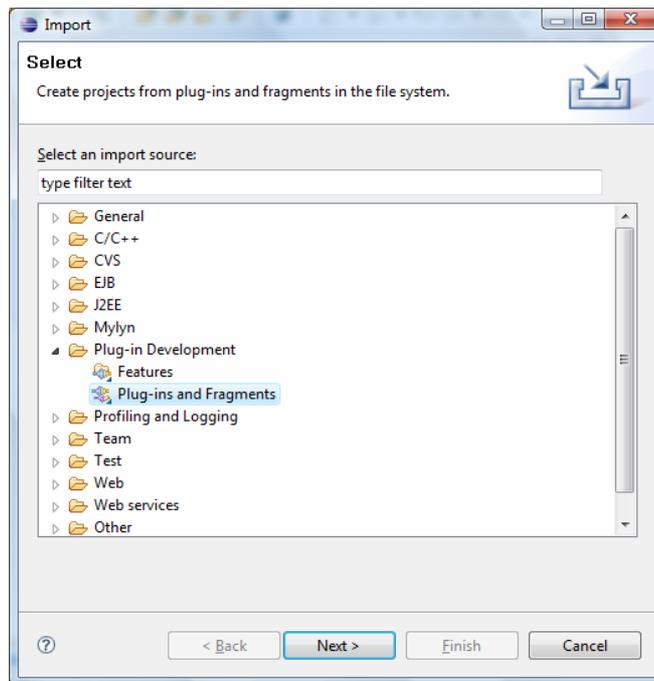


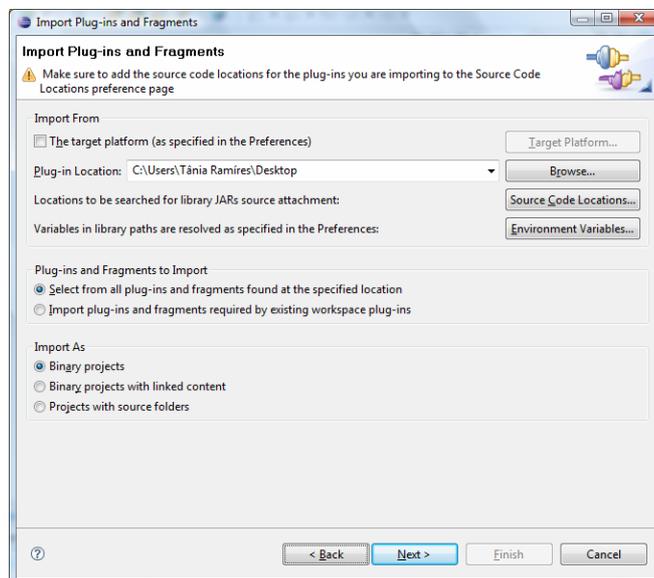
Figura II.1 Download do Arquivo XMI e do Plugin

Após a realização do *download* do arquivo é necessário importá-lo para a IDE Eclipse através do menu *File-> Import* e realizar os seguintes passos:

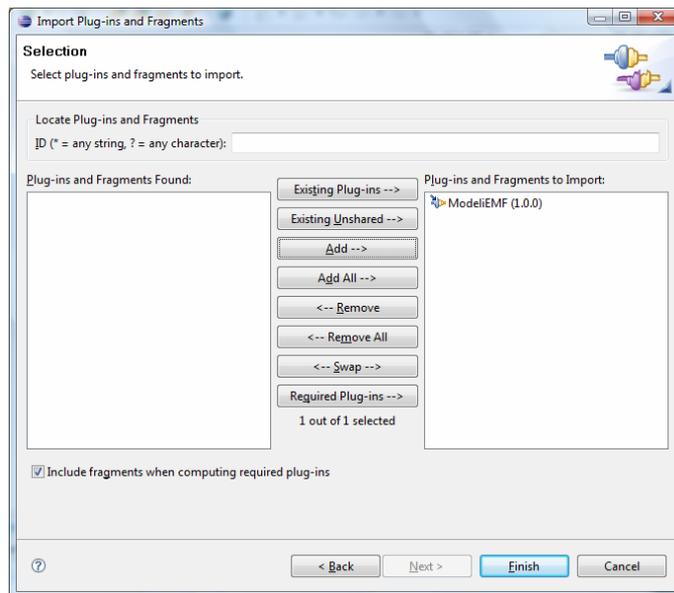


**Figura II.2 Importação do Plugin**

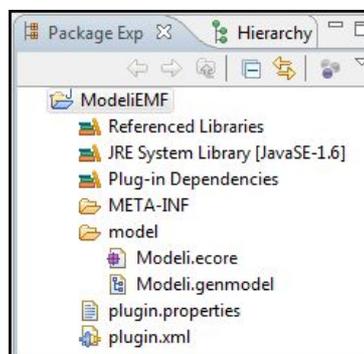
1. Selecione a opção *Plug-in and Fragments* e clique no botão *Next*. Será exibida a tela apresentada na Figura 6.6 para seleção do plugin;
2. Informe o diretório que está localizado o plugin e clique em *Next*;
3. Adicione o *plugin* a ser importado conforme ilustra a Figura 6.7 e clique em *Next*. Pronto, um projeto foi criado no eclipse com as especificações do *plugin*, como ilustra a Figura 6.8.



**Figura II.3 Seleção do diretório do Plugin**



**Figura II.4 Escolha do Plugin**



**Figura II.5 Projeto Modeli EMF**

### **2.3. Projetos para Edição**

O *framework* EMF possui uma meta-modelo (Ecore) para a descrição de modelos em tempo de execução, incluindo suporte para notificação de alterações, apoio à persistência e serialização conforme o padrão XMI e um eficiente API (*Application Programming Interface*) que permite manipular objetos EMF genericamente.

Além disso, disponibiliza projetos que permitem a edição de tais modelos, disponibilizando o conteúdo e os rótulos das classes, apoio ao código fonte, dentre outros benefícios.

Para isso, é necessário a criação de projetos que permitem a edição deste modelo em tempo de execução, sendo necessário abrir o arquivo *Modeli.genmodel* e selecionar as opções *Generate Edit Code* e *Generate Editor Code* no menu *Generator*. A Figura 6.9 ilustra o resultado final após a criação dos projetos que permitem a edição do modelo.

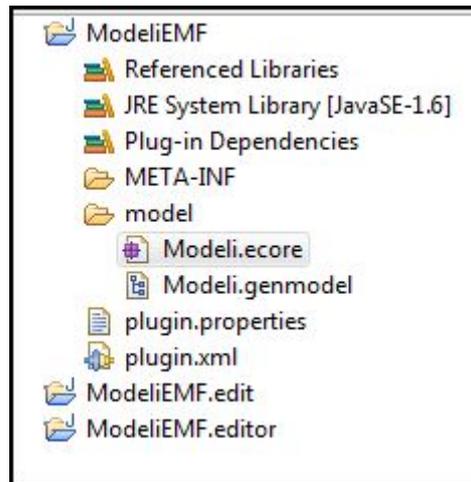


Figura II.6 Projetos que permitem a edição do modelo

Interessante observar o arquivo *Modeli.genmodel* na Figura 6.10. É apresentado todas as classes fazem parte deste meta-modelo, podendo ainda alterá-los conforme a necessidade.

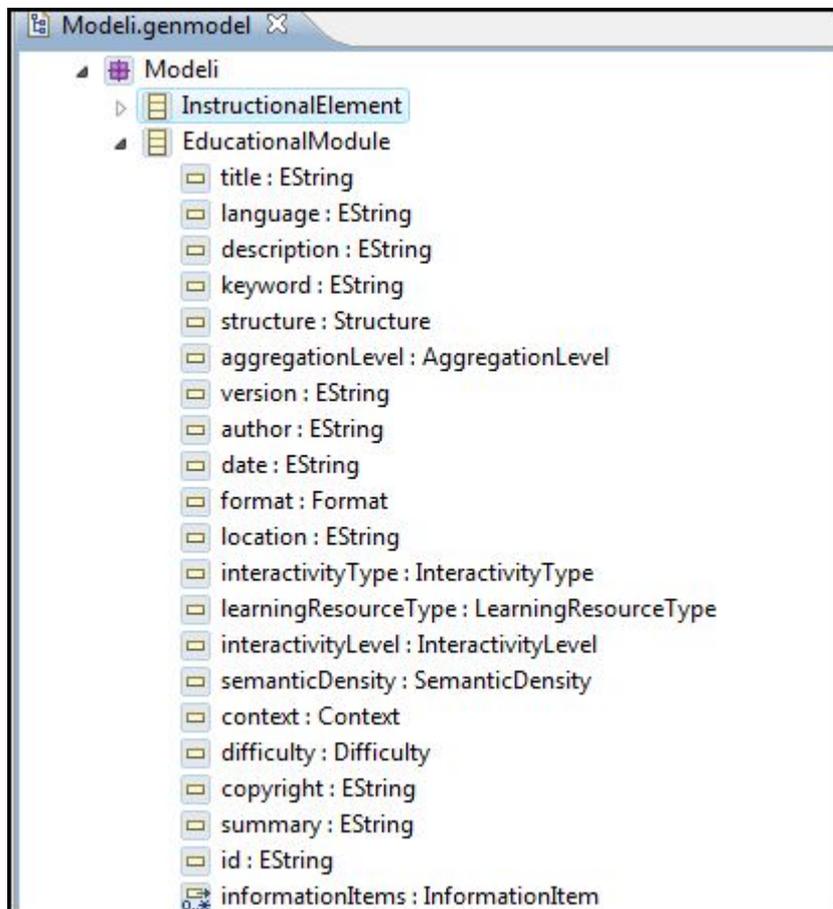


Figura II.7 Arquivo do meta-modelo

## 2.4. Edição de um Modelo Existente

Após a compilação de todos os projetos existentes, para que seja possível a edição do modelo é necessário apenas executar o projeto principal (ModeliEMF) através do próprio Eclipse, através do menu *Run -> Run As -> Eclipse Application*.

Na primeira execução do projeto é necessário criar um projeto geral para a permitir a edição de qualquer modelo ou a criação de um novo modelo. O projeto poderá ser criado no menu *File -> New -> Other -> General -> Project*. Um projeto vazio foi criado conforme ilustra a Figura 6.11.

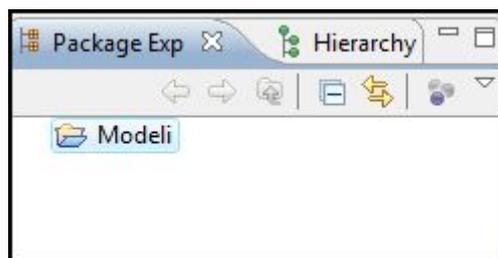


Figura II.8 Projeto vazio

Para a edição de um modelo existente, basta copiar (CTRL+C) o arquivo importado do ambiente MODELi e colar (CTRL+V) neste projeto em execução ou importar o arquivo para este mesmo projeto. A Figura 6.12 ilustra a edição de um Elemento Instrucional Exemplo “Casa Confortável” que está relacionado ao Item de Informação Conceito “Modelamos para...”.

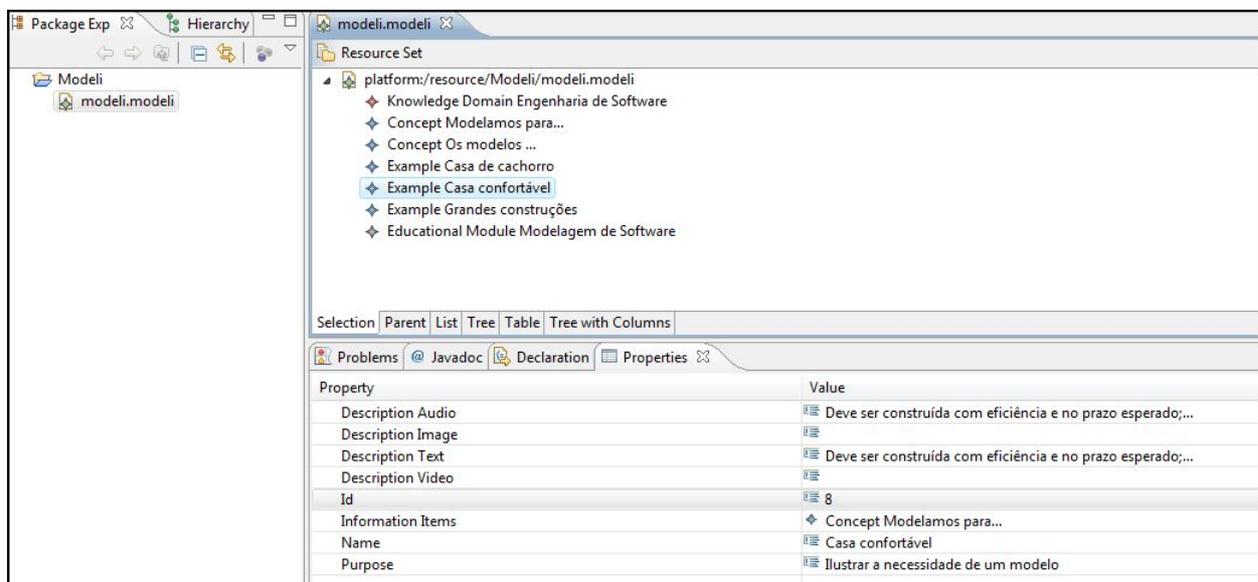


Figura II.9 Modelo sendo editado

## 2.5. Upload das Modificações realizadas

O ambiente Modeli permite que um modelo editado na IDE Eclipse possa ser importado, através do *link de acesso* XMI/Plugin. Para isto, basta selecionar o arquivo a ser importado e clicar no botão *Upload*.

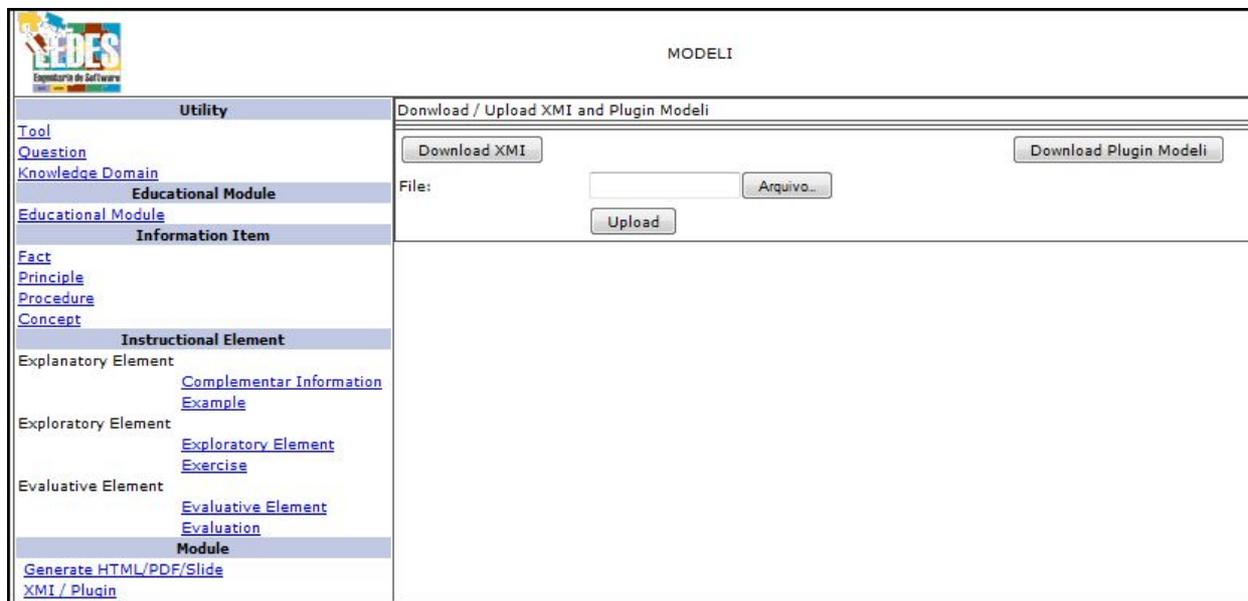


Figura II.10 Upload do Modelo Alterado

## 2.6. Considerações Finais

A edição do modelo através de outras ferramentas é interessante devido à necessidade de utilização de outras ferramentas que possibilitem a Modelagem Conceitual, Instrucional e Didática conforme sugere a abordagem AIM-CID.

O *plugin* construído no presente projeto irá auxiliar nessa edição, porém sabe-se que são necessários ainda diversos estudos que possam disponibilizar ao usuário final uma ferramenta mais robusta e uma integração mais transparente.

# REFERÊNCIAS BIBLIOGRÁFICAS

ABOWD, G. D. *Classroom 2000: An experiment with the instrumentation of a living educational environment*. IBM Systems Journal, v. 38, n. 4, p. 508–530, 1999.

AGUIAR, G. M. *Um pouco além do XML: Introdução ao XML Schema (XSD) - Parte 1*. Disponível em <<http://www.plugmasters.com.br/sys/materias/413/5/Um-pouco-al%E9m-do-XML%3A-Introdu%E7%E3o-ao-XML-Schema-%28XSD%29---Parte-1>>. Acesso em 01/03/2007.

APACHE POI Project. *POI-HSLF - Java API To Access Microsoft Powerpoint Format Files*. Disponível em: <<http://poi.apache.org/hslf/index.html>>. Acesso em 02/08/2007.

KONRATH, M. L. P. *Mapas conceituais*. Disponível em: <<http://penta2.ufrgs.br/edutools/mapasconceituais/>>. Acesso em 24/03/2008.

ALTOÉ, A. *Formação de professores para o uso do computador em sala de aula*. In: Revista Teoria e Prática da Educação: Educação e Informática. Vol. 6, n 14, edição especial, 2003.

ALTOÉ, A. *O papel do facilitador no ambiente logo: formação e atuação*. Campinas, SP: UNICAMP/NIED, 1996.

ALVES, L.; BRITO, M. *O Ambiente Moodle como Apoio Presencial*. Bahia, 2005.

AUSUBEL, D.P.; NOVAK, J.D. e HANESIAN, H. *Psicologia educacional*. Rio de Janeiro: Interamericana, 1980.

BARBOSA, E. F.. *Uma contribuição ao processo de desenvolvimento e modelagem de módulos educacionais*. Tese de Doutorado, ICMC-USP, 2004.

BEHRENS, M. A. *O paradigma emergente e a prática pedagógica*. Curitiba: Champagnat, 2000.

BELLONI, M. L. *Educação à distância*. Campinas, SP: Editora: Autores Associados, 1999.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *Unified Modeling Language User Guide*. 1. ed.: Addison Wesley, 1998.

BRITO, L. S. F. *WebScharts: Uma Ferramenta de Desenvolvimento de Aplicações Web baseada no HMBS/M*. Dissertação de Mestrado, DCT-UFMS, Campo Grande-MS, 2003.

CARVALHO, M. R. C. *HMBS/M - Um método orientado a objetos para o projeto e o desenvolvimento de aplicações hipermídia*. Dissertação de Mestrado - Instituto de Ciências Matemáticas e de Computação (ICMC-USP). São Carlos-SP, 1998.

- CORREIA, A. A.; ANTONY, G. *Educação hipertextual: diversidade e interação como materiais didáticos*. In: FIORENTINI, Leda Maria Rangel; MORAES, Raquel de Almeida (orgs) *Linguagens e interatividade na educação à distância*. Rio de Janeiro: DP& A, 2003.
- COTEIA. Disponível em <<http://incubadora.fapesp.br/projects/coteia>>. Acesso em 13/02/2008.
- EMF, *Eclipse Modeling Framework Project*. Disponível em: <<http://www.eclipse.org/modeling/emf/>>. Acesso em 15/07/2007.
- FILHO, C. S. S. and MACHADO, E. D. C.: *O computador como Agente Transformador da Educação e o Papel do Objeto de Aprendizagem.*, Portal Universia, Disponível em: <<http://www.universiabrasil.net/materia/materia.jsp?id=5939>>. Acesso em 28/01/2006.
- FREIRE, P. *Pedagogia da autonomia. Saberes necessários à prática educativa*. 22 ed. Rio de Janeiro: Paz e Terra, 2002.
- GARBIN, M. C.; AMARAL, S.F: *Construção de um Ambiente Interativo na Internet: a Biblioteca Escolar Digital*. *Revista Iberoamericana de Educación*, 10/04/2008.
- GMF, *Eclipse Graphical Modeling Framework*. Disponível em: <<http://www.eclipse.org/modeling/gmf/>>. Acesso em 03/05/2008.
- HAREL, D. *Statecharts: A visual formalism for complex systems*. *science of computer programming*, n.8, 1987.
- HARRIS, A.Lúcia Nogueira de Camargo. *Análise comparativa entre a utilização dos ambientes WEBCT e TELEDUC como apoio didático às disciplinas do curso de arquitetura e urbanismo da FEC/UNICAMP*, UNICAMP, 2003.
- KEEGAN, D.: *Foundations of distance education*. 3ed. London, Routledge, 1996.
- KENSKI, V. M. *Tecnologias educacionais e educação à distância: Avaliando Políticas e Práticas*. Rio de Janeiro: Quartet, 2001.
- KONRATH, M. L. P. *Mapas conceituais*. Disponível em: <<http://penta2.ufrgs.br/edutools/mapasconceituais/>>. Acesso em 24/03/2006.
- ISO/IEC, International Organization for Standardization. *Information Technology – Software Life Cycle Processes*. *Padrão ISO/IEC 12207*, ISO/IEC, December 1995.
- JSF, *JavaServer Faces Technology*. Disponível em: <<http://java.sun.com/javaee/jaserverfaces/>>. Acesso em 02/02/2007.
- LANGE, D. B. *An object-oriented design method for hypermedia information systems*. In: *International Conference on System Sciences*, New York, NY, 1994.
- LEIVA, W. D. *Um modelo de hipertexto para apoio ao ensino mediado pela web*. *Tese de Doutorado*, USP, SP, 2003.
- LÉVY, P. *As tecnologias da inteligência: O futuro do pensamento na era da informática*. Rio de Janeiro: 1993.

- LOCATELLI, M. H. *Engenharia de software para o desenvolvimento de webapps e as metodologias OOHDM e WEBML*. UFSC, 2003.
- MASETTO, M. T. *Docência na universidade*. Campinas, SP: Papyrus, 1998. MEC- Ministério da Educação. *Regulamentação da EAD no Brasil*. Disponível em: <<http://portal.mec.gov.br/seed/>> . Acesso em 15/03/2006.
- MERRIL, M. D. *Component display theory*. In: *Instructional Design Theories and Models: An Overview of their Current States*, Hillsdale, NJ: Lawrence Erlbaum, 1983.
- MDT, Model Development Tools. Disponível em: <<http://www.eclipse.org/modeling/mdt/?project=uml2>> . Acesso em 08/06/2008.
- MELO, S. S. W. *TV escola: práticas, reflexões e pesquisa*. Dissertação de Mestrado em Educação, UFMS, 2005.
- MOODLE, *Modular Object Oriented Dynamic Learning Environment*. Disponível em: <<http://moodle.org/>>. Acesso em 12/08/2008.
- MVC, *Model-View-Controller*. Disponível em: <<http://java.sun.com/developer/technicalArticles/javase/mvc/index.html>> Acesso em 12/02/2008.
- NEITZEL, L. C. *Novas tecnologias e práticas docentes: o hipertexto no processo de construção do conhecimento (uma experiência vivenciada na rede pública estadual de Santa Catarina)*, UFSC, Florianópolis. 2001.
- NOVAK, J. D.; GOWIN, D. B. *Learning how to learn*. New York: Cambridge University Press, 1984.
- OLIVEIRA, G. F.; RODELLO, I. A.; BREGA, J. R. F; SEMENTILLE, A. C. *Integrando Mapas Conceituais e Recursos Multimídia em Ambientes de Realidade não Imersiva*. Simpósio Brasileiro de Computação. 2008.
- OMG, O. M. G. *OMG XML Metadata Interchange (XMI) Specification*. Janeiro 2002. Disponível em: <<http://www.omg.org/docs/formal/02-01-01.pdf>>. Acesso em: 28/08/2006.
- PAGANO, L. A. N. *A importância do ensino a distância – “EAD” como ferramenta para a educação continuada: um estudo de caso para a escola técnica Tsedakah em Dias D’Ávila – Bahia*, UFSC, 2002.
- PANSANATO, L. T. E. *EHDM: um método para o projeto de aplicações hipermídia para ensino*. Dissertação de Mestrado, ICMC-USP, 1999.
- PANSANATO, L. T. E.; NUNES, M. G. V. *Autoria de aplicações hipermídia para ensino*. *Revista Brasileira de Informática na Educação*, v. 1, n. 5, 1999.
- PETERSON, J. L. Petri nets. *ACM Computing Surveys*, v. 9, n. 3, 1977.
- PITANGA, T. *JavaServer Faces: A mais nova tecnologia Java para desenvolvimento WEB*. Disponível em <<http://www.guj.com.br/content/articles/jsf/jsf.pdf>>. Acesso em 05/05/2008.

- PRADO, M. E. B.; VALENTE, J. A. *A formação na ação do professor: uma abordagem na e para uma nova prática pedagógica*. In: VALENTE, José Armando (org) *Formação de educadores para o uso da informática na escola*. Campinas, SP: UNICAMP/NIED, 2003.
- PRETI, O. *Educação à distância – construindo significados*. Brasília: Plano, 2000.
- PRIMO, A. F. T. *Interação mútua e interação reativa: uma proposta de estudo*. In: CONGRESSO DA INTERCOM, 21, 1998, Recife.
- ROBERTS, J. M.: *The Story of Distance Education: a Practitioner's Perspective*. *Journal of the American Society for Information Science*, v.47, n.11, pp. 811-816, 1996.
- SANTINELLO, J. *O professor universitário utilizando um ambiente virtual de EAD como apoio pedagógico na educação presencial*, UNICENTRO, 2005.
- SCHWABE, D.; ROSSI, G. *The object-oriented hypermedia design model*. *Communications of the ACM*, v. 38, n. 8, 1995.
- SILVA, V. T.; FERRAZ, F.; CARVALHO, G. R.; DAFLON, L. *AulaNet – ajudando os professores a fazerem seu dever de casa*. In: *XXVI Seminário Integrado de Software e Hardware (SEMISH99)*, Rio de Janeiro, RJ, 1999, p. 105–117.
- SOUZA, R. S.; MENEZES, C.S. *Um sistema inteligente para apoio à interação em ambientes cooperativos de aprendizagem*. SBIE, UFES, 2000.
- STOTTS, P. D.; FURUTA, R. *Petri net based hypertext: document structure with browsing semantics*. *ACM Transactions on Information Systems*, v. 7, n. 1, 1989.
- STOTTS, P. D.; FURUTA, R. *Dynamic adaptation of hypertext structure*. In: *Third ACM Conference on Hypertext (Hypertext 91)*, San Antonio, Texas, 1991.
- TELEDUC Ensino à Distância. Disponível em: < <http://www.teleduc.org.br/> > Acesso em 16/01/2008.
- TIDIA-AE. *Tecnologia da Informação no Desenvolvimento da Internet Avançada*. Disponível em: < <http://www.tidia.fapesp.br/portal> > Acesso em 27/03/2008.
- TURINE, M. A. S. *HMBS - Um modelo Baseado em Statecharts para a Especificação Formal de Hiperdocumentos*. Tese de Doutorado, Instituto de Física de São Carlos (IFSC), São Carlos-SP. 1998.
- TURINE, M. A. S.; OLIVEIRA, M.C.F.; MASIERO, P. C. *HySCharts: A Statechart-Based Environment for Hyperdocument Authoring and Browsing*. In *Multimedia Tools and Applications*, v. 8, p. 309-324, 1999.
- UAB. *Universidade Aberta do Brasil*. Disponível em: <<http://uab.capes.gov.br>>. Acesso em 03/02/2008.
- UML. *OMG Unified ModELing Language Specification version 2.0*. Object Management Group. Disponível em: <<http://www.omg.org/uml>>. Acesso em 25/03/2006.

VALENTE, J.A. *Diferentes usos do computador na educação*, NIED, UNICAMP, 1995a. Disponível em: <[http://www.nied.unicamp.br/publicacoes/pub.php?classe=separata&cod\\_publicacao=10](http://www.nied.unicamp.br/publicacoes/pub.php?classe=separata&cod_publicacao=10)>. Acesso em 26/08/2006.

VALENTE, J. A. *Por quê o computador na educação?*, NIED, UNICAMP, 1995b. Disponível em:<[http://www.nied.unicamp.br/publicacoes/pub.php?classe=separata&cod\\_publicacao=11](http://www.nied.unicamp.br/publicacoes/pub.php?classe=separata&cod_publicacao=11)>. Acesso em 23/07/2006

VALENTE, J. A. *O uso inteligente do computador na educação*, NIED, UNICAMP, 2002. Disponível em: <<http://www.unidavi.edu.br/~afischer/conteudo.html>> . Acesso em 24/03/2006.

WEBCT. *Web Course Tools*. Disponível em <<http://www.webct.com/>>. Acesso em 27/08/2006.