Uma contribuição à modelagem e à geração
automática de conteúdos educacionais

Vanessa Araujo Borges

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 26 de maio de 2010

Assinatura: _____

Uma contribuição à modelagem e à geração automática de conteúdos educacionais

Vanessa Araujo Borges

Orientadora: *Profa. Dra. Ellen Francine Barbosa*

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação — ICMC/USP como parte dos requisitos para obtenção do título de Mestre em Ciências de Computação e Matemática Computacional.

USP - São Carlos
Maio/2010

Agradecimentos

Agradeço a Deus, por me conceder força e determinação para buscar por mais este objetivo.

À Ellen Francine Barbosa, professora e orientadora, pela confiança, palavras de incentivo e, principalmente, pela amizade. Seu apoio constante e disponibilidade irrestrita fizeram toda a diferença. Obrigada Mammy!

Aos professores da pós-graduação do ICMC por toda a contribuição para a minha formação e aos funcionários pelo constante auxílio. Em especial, aos professores José Carlos Maldonado, Simone do Rocio Senger de Souza, Maria da Graça Campos Pimentel e Solange Oliveira Rezende pelas valiosas ideias, dicas e sugestões que foram essenciais para a condução deste trabalho.

Aos professores da UFMS, Marcelo Augusto Santos Turine, Edson Norberto Caceres e Nahri Balesdent Moreano, por todo o apoio e incentivo tanto na graduação quanto para a continuidade da minha vida acadêmica.

Aos meus pais, Cida e Borges e irmãs, Clarissa e Sylvia Carolina, pela compreensão, carinho e amor incondicional. A presença de vocês tornam minha caminhada sempre mais suave. Ao André, cunhadinho querido, pelo constante incentivo.

Ao Bruno por todo o amor, carinho, paciência e incentivo. Sua presença em todos os momentos e apoio constante, foram fundamentais para a concretização deste trabalho.

À Dona Antonieta e família que fizeram com que os dias em São Carlos fossem mais do que especiais.

Aos amigos do mestrado e agregados, em especial: Mel, Victor, Sandro, Marília, Leandro, André Abe, Gambi, Marllós, Rodolfo, Marco, Gondim, Jorge, Diogo, Kenji, Mário, Rafael, Rodrigo, Fabiano, Draylson, Kátia, Sorim, Alê, Endo, Resina, Maluquinho, Otávio, Vânia, Aninha, Gaúcho, Edson e Viviane, que tornaram o ambiente de trabalho sempre descontraído e muito agradável.

Aos amigos de Campo Grande: Marcela, Japa, Renata, Chris e Felipe; e de São Paulo: Kika, Carlos, Paula, Otávio, Andreza, Luiz, Laurinha, Camila, Moa, Marcella.

Às funcionárias do setor de Pós-Graduação ICMC/USP, Ana Paula, Beth, Laura e Livia, pela atenção e apoio.

Ao CNPq pelo apoio financeiro.

Ao Universo Online, representado por Daniel Ambrósio e Stella Ota, pela compreensão e incentivo nessa fase final do mestrado.

Por fim, deixo minha sincera gratidão a todas as pessoas que contribuíram, direta ou indiretamente, para a concretização deste trabalho.

Ambientes e sistemas de apoio ao ensino e aprendizado estão sendo adotados como mecanismos facilitadores no processo de ensino e aprendizado. No entanto, uma limitação comum à maioria desses ambientes está relacionada ao fato de concentrarem-se apenas na criação da estrutura, armazenamento e controle de acesso ao material didático, sem oferecer qualquer tipo de suporte à atividade de modelagem do conteúdo. Nesse contexto, em trabalhos anteriores foi proposta a abordagem *ALM-CTD* – uma abordagem integrada para modelagem de conteúdos educacionais. Dando continuidade às pesquisas já realizadas, este trabalho aborda o estudo de mecanismos de apoio à modelagem de conteúdos educacionais, propondo extensões às etapas de modelagem conceitual e instrucional da abordagem *ALM-CTD*. A ideia é incorporar à abordagem aspectos relacionados ao reuso e compartilhamento de conteúdos educacionais. Além disso, também foi considerado no escopo deste trabalho o projeto e o desenvolvimento da *ALM-Tool* – uma ferramenta Web, de apoio à modelagem e geração automática de conteúdos educacionais, com ênfase na construção distribuída dos modelos estendidos da abordagem *ALM-CTD*. A ferramenta fornece mecanismos para a geração automática dos conteúdos modelados em diferentes formatos, além de tratar aspectos relacionados ao compartilhamento do material gerado, por meio da adoção do padrão LOM.

Palavras-chave: Módulos educacionais, modelagem de conteúdos, ontologias, ferramentas de apoio, geração de conteúdos.

Learning environments and systems has been investigated in order to facilitate the learning processes in general. However, one common limitation of most of these environments is related to the fact that they focus only on the creation of the structure, storage and learning material access control, with no support to the content modeling activity in the educational process. In this regard, in a previous work we proposed the *IMA-CID* – an integrated approach for modeling educational content. Motivated by this scenario, in this work we investigate supporting mechanisms for educational content modeling, proposing extensions to the conceptual model and instructional model of the *IMA-CID* approach. The idea is to incorporate aspects of sharing and reuse for educational content creation. Moreover, we have also considered the development of *IMA-Tool* – a Web tool that supports the modeling and automatic generation of educational content, emphasizing the distributed construction of the extended models of the *IMA-CID* approach. This tool provides mechanisms for the automatic generation in various formats, besides considering aspects related to content sharing through the adoption of the LOM pattern.

Keywords: Educational modules, content modeling, ontology, supporting tool, content generation.

Sumário

Resumo	iii
Abstract	v
Lista de Figuras	ix
Lista de Tabelas	xi
Lista de Abreviaturas	xiii
1 Introdução	1
1.1 Contexto e Motivação	1
1.2 Objetivos	4
1.3 Organização do Trabalho	5
2 Fundamentação Teórica	7
2.1 Considerações Iniciais	7
2.2 Mecanismos de Apoio ao Desenvolvimento de Módulos Educacionais	7
2.2.1 Abordagem Integrada para a Modelagem de Conteúdos Educacionais	8
2.2.2 Processo Padrão para o Desenvolvimento de Módulos Educacionais	18
2.3 Conteúdos Educacionais Livres	19
2.4 Padrões e Interoperabilidade	21
2.4.1 IEEE 1484 – LOM	22
2.4.2 SCORM	24
2.5 Ambientes e Ferramentas de Apoio ao Ensino e Aprendizado	27
2.6 Técnicas e Ferramentas de Apoio à Estruturação do Conhecimento	32
2.6.1 Mapas Conceituais	32
2.6.2 Modelo HMBS	36
2.7 Considerações Finais	42
3 Evoluções à Abordagem AIM-CID	45
3.1 Considerações Iniciais	45
3.2 Ontologias para a Modelagem de Conteúdos	45
3.3 Extensões ao Modelo Conceitual	48

3.4	Extensões ao Modelo Instrucional	53
3.5	Considerações Finais	58
4	AIM-Tool: Apoio Automatizado à Abordagem AIM-CID	61
4.1	Considerações Iniciais	61
4.2	Visão Geral de Ferramenta <i>AIM-Tool</i>	62
4.3	Aspectos de Modelagem	63
4.4	Aspectos Arquiteturais	71
4.5	Aspectos de Implementação	74
4.6	Aspectos Operacionais	76
4.7	Considerações Finais	87
5	Conclusões	89
5.1	Visão Geral da Pesquisa Realizada	89
5.2	Contribuições de Pesquisa	90
5.3	Trabalhos Futuros	91
	Referências	93

Lista de Figuras

2.1	<i>AIM-CID</i> : abordagem integrada para modelagem de conteúdos educacionais	9
2.2	Modelo conceitual do módulo <i>Teste de Software: Teoria e Prática</i>	11
2.3	Modelo instrucional do módulo <i>Teste de Software: Teoria e Prática</i>	13
2.4	Transparências referentes à Análise de Mutantes (módulo educacional <i>Teste de Software: Teoria e Prática</i>)	14
2.5	Modelo didático do módulo <i>Teste de Software: Teoria e Prática</i> (especificação aberta)	17
2.6	Transparências referentes a Mutante – versão aberta, parcialmente aberta e fechada	18
2.7	<i>MIT Open Course Ware</i> – lista de cursos disponíveis por departamento	20
2.8	Árvore resumida do LOM	24
2.9	Modelo de colaboração 3C	28
2.10	Página principal e algumas ferramentas do <i>Ae</i>	31
2.11	Modelagem do conhecimento utilizando a ferramenta <i>CmapTools</i>	34
2.12	Tela principal da ferramenta <i>Inspiration</i>	36
2.13	Arquitetura da ferramenta <i>HySCharts</i>	40
2.14	EGS - Editor Gráfico de <i>Statecharts</i>	41
2.15	Arquitetura da ferramenta <i>WebScharts</i>	42
3.1	Camadas da Web semântica	47
3.2	(a) Mapa Conceitual para o Critério Análise de Mutantes . (b) Mapa Conceitual para o mesmo domínio de conhecimento, caracterizando uma topologia em árvore	50
3.3	OntoTest: ontologia principal de Teste de Software	51
3.4	Modelo conceitual considerando a sub-ontologia de Procedimento de Teste da OntoTest	53
3.5	ALoCoM-Ontology	54
3.6	Composição do objeto de aprendizado	55
3.7	Integração da ALoCoM-Ontology com itens de informações e elementos instrucionais do modelo instrucional da abordagem <i>AIM-CID</i>	55
3.8	Estrutura definida para o modelo <i>HMBIS/Instrucional</i> para caracterizar o conceito e as informações complementares	57

3.9	Modelo instrucional (<i>HMBIS/Instrucional</i>) da abordagem <i>AIM-CID</i> para o domínio de Teste de Software	58
4.1	Funcionalidades da ferramenta <i>AIM-Tool</i>	63
4.2	Diagrama de caso de uso da ferramenta <i>AIM-Tool</i>	64
4.3	Diagrama de atividades para a gerência de domínio, modelo conceitual e modelo instrucional definidos para a ferramenta <i>AIM-Tool</i>	66
4.4	Diagrama de atividades para o modelo didático e módulo de geração de conteúdo da ferramenta <i>AIM-Tool</i>	67
4.5	Diagrama UX definido no projeto da ferramenta <i>AIM-Tool</i>	69
4.6	Diagrama de classes definido no projeto da ferramenta <i>AIM-Tool</i>	70
4.7	Arquitetura da ferramenta <i>AIM-Tool</i>	71
4.8	Interface do modelo instrucional da ferramenta <i>AIM-Tool</i> ilustrando a composição do seu contexto navegacional	77
4.9	Página principal da ferramenta <i>AIM-Tool</i> com caminhos para adição de novos autores e novos domínios de conhecimento	77
4.10	Interface da ferramenta <i>AIM-Tool</i> para a inserção manual de conceitos	78
4.11	Interface da ferramenta <i>AIM-Tool</i> para a inserção de conceitos por meio de ontologias	78
4.12	Interface da ferramenta <i>AIM-Tool</i> ilustrando a visualização textual ao adicionar novo conceito	79
4.13	Interface da ferramenta <i>AIM-Tool</i> ilustrando as diferentes formas de visualização gráfica de conceitos.	80
4.14	Interfaces da ferramenta <i>AIM-Tool</i> ilustrando as funcionalidades do modelo instrucional	81
4.15	Interface da ferramenta <i>AIM-Tool</i> para a definição de estados do tipo OR do modelo <i>HMBIS/Instrucional</i>	82
4.16	Interface da ferramenta <i>AIM-Tool</i> para a definição de estados com tipo AND e OR do modelo <i>HMBIS/Instrucional</i>	82
4.17	Modelo didático definido para a ferramenta <i>AIM-Tool</i> considerando a seleção do conteúdo que será gerado	83
4.18	Interface da ferramenta <i>AIM-Tool</i> com o relatório gerado após a seleção, no modelo didático, do conteúdo	84
4.19	Transição do <i>HMBIS/Didático</i> para um SCXML	84
4.20	Geração de conteúdo pela ferramenta <i>AIM-Tool</i>	85
4.21	Exemplo de material gerado pela ferramenta <i>AIM-Tool</i> no formato PDF-Latex	86

Lista de Tabelas

2.1	Categorias e atributos do LOM.	23
4.1	Informações de implementação da ferramenta <i>AZM-Tool</i>	76

Lista de Abreviaturas

ADL	Advanced Distributed Learning
AIM-CID	Abordagem Integrada de Modelagem - Conceitual, Instrucional e Didática
ALOC_oM	Abstract Learning Object Content Model
API	Application Program Interface
FLOSS	Free Libre and Open Source Software
HMBS	Hyperdocument Model Based on Statecharts
HTML	HyperText Markup Language
MVC	Model View Controller
OER	Open Educational Resources
OWL	Web Ontology Language
LMS	Learning Management Systems
LOM	Learning Object Metadata
RDF	Resource Description Framework
SCXML	State Chart extensible Markup Language
SCORM	Sharable Content Object Reference Model
URI	Uniform Resource Identifier
WAE	Web Application Extension
XML	Extensible Markup Language

Introdução

1.1 Contexto e Motivação

Inúmeras pesquisas destinadas ao projeto e desenvolvimento de ferramentas e recursos computacionais para apoiar o ensino e o aprendizado, contemplando as modalidades presencial, híbrida (*blended*) e a distância, têm sido desenvolvidas ao longo dos anos (Dougiamas e Taylor, 2003; Farmer e Dolphin, 2005; IWT, 2006; Tidia, 2006). De modo geral, tais ambientes têm por objetivo auxiliar a automatização das tarefas relativas ao ensino, oferecendo suporte à disponibilização de material didático, à condução de atividades colaborativas e à avaliação e acompanhamento dos aprendizes.

No entanto, tarefas típicas do professor, como selecionar o conteúdo e preparar o material didático, continuam em segundo plano, sem qualquer tipo de ferramenta de apoio ou diretrizes capazes de auxiliá-lo. Nesse sentido, a organização e a estruturação do domínio de conhecimento, de modo a auxiliar o professor na tarefa de modelagem e produção do material didático, possibilitando a elaboração de material de qualidade, construído com base em princípios educacionais previamente estabelecidos, também constitui um fator determinante no processo educacional.

Segundo Barbosa (2004), os conteúdos trabalhados pelo aprendiz devem possuir, entre outras características: (1) abrangência em relação ao entendimento dos conceitos e teorias que servem como base para o domínio de conhecimento a ser ensinado; (2) profundidade para motivar o aprendiz, a partir dos conceitos já abordados, a explorar por si mesmo outros

tópicos e sub-áreas relacionadas; e (3) orientação, com o objetivo de facilitar a aplicação prática do conhecimento adquirido.

Sob essa perspectiva, a definição de mecanismos adequados à autoria e apresentação de conteúdos educacionais têm sido explorados a fim de fornecer subsídios ao estabelecimento de ambientes efetivos quanto aos objetivos de aprendizado especificados. Procurando abordar tais aspectos, uma das linhas de pesquisa que vem sendo investigada no contexto de ensino e aprendizado refere-se ao desenvolvimento de módulos educacionais – unidades de estudo, compostas por conteúdos teóricos integrados a avaliações e atividades práticas, cuja disponibilização aos aprendizes é apoiada por meio de recursos tecnológicos e computacionais (Barbosa, 2004; Barbosa e Maldonado, 2004, 2006a,b,c).

Considerando o contexto de desenvolvimento de módulos educacionais, a modelagem de seus conteúdos associados representa uma atividade essencial para a estruturação do conhecimento apoiando a identificação e a definição de conceitos e informações pertinentes, possibilitando que estes sejam disponibilizados de modo coerente e ordenado. De fato, a simples utilização de recursos computacionais não garante, necessariamente, o sucesso e a efetividade do aprendizado; a qualidade do material didático utilizado constitui um fator determinante nessa perspectiva e, com isso, a modelagem de conteúdos torna-se um aspecto fundamental.

Além disso, é importante observar que a modelagem de conteúdos é apenas uma das atividades a serem consideradas. O desenvolvimento de módulos educacionais requer que, além de aspectos essencialmente técnicos, fatores gerenciais e organizacionais também sejam considerados. Adicionalmente, tais módulos podem ser disponibilizados aos aprendizes de diferentes maneiras, evoluídos conforme o aprendizado, reutilizados em diferentes contextos de ensino e aprendizado e para aprendizes com perfis distintos. Em virtude da diversidade e complexidade dos fatores envolvidos, a construção desses produtos demanda tempo e recursos (humanos, técnico-administrativos e econômicos), não sendo uma atividade trivial (Barbosa, 2004).

Por fim, o desenvolvimento de módulos educacionais pode ainda ser abordado segundo características e princípios do desenvolvimento de software livre (Barbosa, 2004). Traçando um paralelo entre ambos, algumas características de desenvolvimento em comum podem ser observadas. A primeira está relacionada à evolução contínua, em que versões atualizadas do software são lançadas com mais frequência, em resposta a necessidades e opiniões dos usuários. Essa mesma característica também constitui um aspecto essencial no desenvolvimento de módulos educacionais considerando o caráter dinâmico e evolutivo associado ao conhecimento, em que os módulos são constantemente evoluídos como resultado das experiências de aprendizado.

Outra característica refere-se à dispersão geográfica dos desenvolvedores e, como consequência, a necessidade de uma participação distribuída na construção do produto. A ideia é que qualquer desenvolvedor possa contribuir para o produto (software ou módulo educacional) e que as contribuições sejam filtradas de forma que o melhor código (no caso dos módulos, o melhor conteúdo) permaneça, sendo incorporado ao produto em questão.

Além disso, o desenvolvimento de produtos livres requer um conjunto de tecnologias colaborativas de modo a garantir a comunicação e a interação entre os vários desenvolvedores. No caso de módulos educacionais, o uso de tecnologias colaborativas é fundamental não apenas no processo de desenvolvimento, mas também durante a disponibilização e utilização do módulo, como forma de viabilizar a condução das atividades e avaliações propostas aos aprendizes.

Os aspectos discutidos anteriormente motivaram a investigação de mecanismos de apoio específicos dentro do contexto de ensino e aprendizado. Tais mecanismos foram explorados no trabalho de doutorado de Barbosa (2004), tendo sido caracterizados nas seguintes linhas de atuação:

1. Estudo e definição de mecanismos para a modelagem de conteúdos educacionais, estabelecendo subsídios para a identificação e estruturação das partes relevantes do domínio de conhecimento a serem ensinadas.
2. Definição de parâmetros para comparação e seleção de modelos para representação de conteúdos e estabelecimento de uma abordagem integrada de modelagem, procurando ressaltar pontos fortes e minimizar limitações identificadas nas abordagens de modelagem existentes.
3. Padronização de processos para a elaboração de módulos educacionais, fornecendo diretrizes para o desenvolvimento sistemático de módulos educacionais.
4. Especialização dos processos definidos e sua instanciação para projetos específicos.

Especificamente com relação à modelagem de conteúdos, foi proposta a *AIM-CID* (Abordagem Integrada de Modelagem – Conceitual, Instrucional e Didática), caracterizando um conjunto de modelos genéricos para a representação de conteúdos educacionais (Barbosa, 2004; Barbosa e Maldonado, 2004, 2006b). Cada modelo aborda aspectos distintos da atividade de modelagem. O Modelo Conceitual consiste em uma descrição de alto-nível do domínio de conhecimento. Sua construção trata a definição dos conceitos relevantes para a compreensão do domínio e a especificação de como esses conceitos se relacionam. O Modelo Instrucional define itens de informação (fatos, procedimentos e princípios) e elementos instrucionais (exemplos, informações complementares, exercícios, dicas e ferramentas),

associando-os aos conceitos já identificados. O Modelo Didático é responsável por associar os objetos anteriormente modelados, estabelecendo uma sequência de apresentação entre eles.

Os modelos estabelecidos propiciaram o desenvolvimento e a reestruturação de materiais didáticos na área de Engenharia de Software, mais especificamente dentro da temática de Teste de Software, a fim de proporcionar o domínio e a disseminação de conhecimentos técnico-científicos nesse contexto. Os módulos “Teste de Software: Teoria e Prática” (Barbosa et al., 2008c), “Fundamentos de Programação e Teste de Software” (Barbosa et al., 2008b) e “Teste de Mutação” (Barbosa e Maldonado, 2006a) foram desenvolvidos e aplicados no contexto educacional sendo que sua efetividade pode ser evidenciada, principalmente em relação ao desempenho dos alunos e à utilização de aplicações-exemplos. Ressalta-se, entretanto, que tais módulos foram desenvolvidos de forma manual, sem qualquer tipo de apoio automatizado na aplicação da abordagem *AIM-CTD*. Além disso, aspectos de desenvolvimento distribuído também não foram considerados.

Dentro da perspectiva apresentada, pretende-se investigar a evolução dos mecanismos descritos a fim de possibilitar sua aplicação de forma compartilhada e distribuída, associada ao projeto e desenvolvimento de uma ferramenta para apoiar as atividades de modelagem e a geração automática de conteúdos educacionais livres relacionados aos módulos. A ideia é evoluir e automatizar os modelos estabelecidos na abordagem *AIM-CTD* de modo a possibilitar a modelagem compartilhada de conteúdos, minimizar os possíveis erros produzido durante sua aplicação manual, além de possibilitar a geração automática dos conteúdos modelados em diferentes formatos. O presente trabalho de mestrado insere-se nesse contexto, tendo seus objetivos detalhados a seguir.

1.2 Objetivos

Os objetivos deste trabalho podem ser caracterizados seguindo duas principais linhas de atuação. A primeira delas está relacionada à evolução dos modelos que constituem a abordagem *AIM-CTD* por meio da proposição de extensões e adequações aos mesmos. Procurou-se evoluir aspectos relacionados à caracterização e componentização de mídias para a modelagem de conteúdos dinâmicos. Além disso, considerou-se a possibilidade de fornecer subsídios à construção compartilhada e reuso do conhecimento por meio da modelagem de conteúdos estruturados com base em um senso comum, além de diretrizes para possibilitar a construção do conhecimento de forma distribuída, por meio da definição de um ambiente automatizado.

A segunda linha refere-se ao desenvolvimento da *AIM-Tool* – uma ferramenta para viabilizar a construção dos modelos conceitual, instrucional e didático da abordagem *AIM-CTD*. Ainda, a *AIM-Tool* deve possibilitar a geração automática dos conteúdos modelados em diferentes formatos, além de tratar aspectos relacionados ao reuso do material didático re-

sultante, por meio da adoção de metadados definidos para o desenvolvimento de conteúdos portáteis (Borges e Barbosa, 2008, 2009a,b).

Ressalta-se que os aspectos investigados foram explorados no desenvolvimento de um módulo educacional para o domínio de Teste de Software, ilustrando a aplicação das evoluções propostas a partir da utilização da ferramenta *ALM-Tool*.

1.3 Organização do Trabalho

Neste capítulo foram apresentados o contexto no qual este trabalho se insere, as motivações para a sua realização e os principais objetivos relacionados. No Capítulo 2, é apresentada a fundamentação teórica a respeito das principais pesquisas relacionadas. Inicialmente é apresentada a ideia de módulos educacionais com ênfase na abordagem *ALM-CTD* para a modelagem dos conteúdos associados aos módulos. Em seguida, é apresentada a ideia de conteúdos educacionais livres e uma breve descrição sobre os padrões LOM (*Learning Object Metadata*) e SCORM (*Sharable Content Object Reference Model*), considerados nesta pesquisa a fim de garantir a portabilidade do material didático gerado pela ferramenta *ALM-Tool*. Além disso, é apresentada uma visão geral a respeito de alguns dos ambientes e sistemas educacionais existentes, em especial ferramentas que apoiam a estruturação e representação do conhecimento utilizando a técnica de Mapas Conceituais e o modelo HMBS (*Hyperdocument Model Based on Statecharts*).

No Capítulo 3, são apresentadas as evoluções propostas à abordagem *ALM-CTD*, com ênfase na evolução dos modelos conceitual e instrucional. Tais modelos foram evoluídos para incorporar aspectos relacionados ao compartilhamento do conteúdo modelado. Outra evolução refere-se à caracterização de mídias necessárias à produção de conteúdos dinâmicos, além da definição de diretrizes ao modelo instrucional, a fim de possibilitar sua construção de forma sistemática.

No Capítulo 4, é apresentada a ferramenta *ALM-Tool*, desenvolvida para viabilizar a modelagem e a geração automática de conteúdos educacionais livres, segundo as especificações da abordagem *ALM-CTD*. Para o projeto e desenvolvimento da ferramenta *ALM-Tool*, foram especificadas sua arquitetura, além de seus aspectos operacionais de modelagem e de implementação.

Por fim, no Capítulo 5, são apresentadas as conclusões deste trabalho, destacando-se as principais contribuições e trabalhos futuros relacionados.

Fundamentação Teórica

2.1 Considerações Iniciais

Neste capítulo é apresentada a fundamentação teórica relevante à condução do trabalho. Na Seção 2.2, são descritos os mecanismos para o desenvolvimento de módulos educacionais, com ênfase na abordagem *AIM-CID*. Na Seção 2.3, é apresentada a definição de conteúdos educacionais livres, também considerada na geração de conteúdos educacionais. Na Seção 2.4, são descritos os padrões LOM e SCORM, investigados para compor a geração automática de conteúdos, garantindo interoperabilidade e reusabilidade aos mesmos. Na Seção 2.5, são brevemente descritos alguns dos principais ambientes educacionais existentes. Por fim, na Seção 2.6, são apresentadas algumas técnicas e ferramentas específicas utilizadas na estruturação e representação do conhecimento.

2.2 Mecanismos de Apoio ao Desenvolvimento de Módulos Educacionais

Uma das linhas que tem sido objeto de pesquisa no contexto de ensino e aprendizagem refere-se ao desenvolvimento de módulos educacionais. Entende-se por módulo educacional uma unidade concisa de estudo, composta basicamente por conteúdos teóricos integrados a

atividades práticas e avaliações, cuja disponibilização aos aprendizes é apoiada por recursos tecnológicos e computacionais (Barbosa, 2004).

Um módulo educacional pode ser caracterizado em termos de seus principais componentes: material didático e infra-estrutura. O material didático é constituído por conteúdos teóricos e conteúdos práticos. Os conteúdos teóricos podem ser caracterizados como livros, artigos, informações e referências na Web, transparências, anotações de aula, áudio, vídeo entre outros. Os conteúdos práticos são caracterizados em função das atividades e avaliações conduzidas e de seus produtos resultantes (documentos, códigos-fonte, programas executáveis, discussões, estudos experimentais). Ferramentas específicas pertinentes ao domínio de conhecimento bem como os resultados obtidos a partir de sua utilização também são considerados conteúdos práticos.

Recursos computacionais e tecnológicos, tais como ambientes e sistemas educacionais, são caracterizados como parte da infra-estrutura necessária à integração do material didático e à consequente disponibilização do módulo educacional.

A modelagem de material didático de qualidade representa um aspecto essencial no desenvolvimento de módulos educacionais, sendo uma tarefa complexa que requer a consideração de uma série de fatores – desde a determinação de aspectos específicos do domínio de conhecimento e a 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, no trabalho de Barbosa (2004) foi definida uma abordagem integrada de modelagem (*AIM-CID*) a qual tem como objetivo reunir, de forma consistente, os diferentes aspectos – conceitual, instrucional e didático – relacionados à modelagem de conteúdos educacionais. Ainda como parte do trabalho de Barbosa (2004), também foi definido um Processo Padrão para o Desenvolvimento de Módulos Educacionais (PP-DME). A seguir é apresentada uma visão geral a respeito da abordagem *AIM-CID* e do PP-DME, com ênfase nos aspectos de modelagem de conteúdos educacionais.

2.2.1 Abordagem Integrada para a Modelagem de Conteúdos Educacionais

Considerando o contexto de desenvolvimento de módulos educacionais, a modelagem dos conteúdos associados representa uma atividade essencial para a estruturação do conhecimento, apoiando a identificação e a definição de conceitos e informações pertinentes, possibilitando em última análise que os mesmos sejam disponibilizados de modo coerente e ordenado, com base em teorias e princípios educacionais previamente definidos. Desse modo, a simples utilização de recursos computacionais não garante necessariamente o sucesso e a efetividade do

aprendizado. Assim, a qualidade do material didático utilizado também constitui um fator determinante nessa perspectiva e, com isso, a modelagem de conteúdos torna-se um aspecto fundamental (Barbosa, 2004).

Motivada por esse cenário, em seu trabalho de doutorado Barbosa (2004) definiu três perspectivas de modelagem distintas e inter-relacionadas a fim de caracterizar os principais aspectos na representação de conteúdos educacionais: (1) conceitual – modelagem de conceitos relevantes do domínio de conhecimento e seus inter-relacionamentos; (2) instrucional – modelagem de informações adicionais e elementos complementares ao domínio de conhecimento; e (3) didática – associação entre os objetos já modelados – conceituais e instrucionais – estabelecendo uma sequência de apresentação entre eles. Na Figura 2.1 são ilustrados os modelos que constituem a abordagem *AIM-CID*, que serão descritos a seguir.

Figura 2.1: *AIM-CID*: abordagem integrada para modelagem de conteúdos educacionais (Barbosa, 2004).

Modelo Conceitual

O modelo conceitual, segundo Barbosa (2004), consiste em uma descrição em alto nível do domínio que se deseja ensinar. Sua construção envolve a definição dos conceitos para a compreensão do domínio, a especificação da forma pela qual os mesmos se relacionam e o estabelecimento de uma estrutura para representar esses elementos.

Os elementos do domínio de conhecimento são caracterizados essencialmente em termos de seus conceitos. Os relacionamentos entre conceitos também devem ser especificados, podendo ser divididos em duas classes: relacionamentos estruturais e relacionamentos específicos do domínio. Os relacionamentos estruturais representam uma categoria genérica de relacionamento, independentes do domínio, podendo ainda ser pré-definidos de acordo com o modelo conceitual utilizado. Os relacionamentos de taxonomia, relacionados à representação hierárquica do domínio de conhecimento, e de composição, quando um conceito pode ser explicado a partir de outro, são exemplos de relacionamentos estruturais. Os relaciona-

mentos específicos do domínio, por sua vez, representam as relações específicas em que as interpretações dependem do domínio de conhecimento.

Após a definição dos conceitos e dos relacionamentos entre eles, faz-se necessário a organização e a representação estruturada dos mesmos a fim de facilitar a compreensão do domínio de conhecimento sendo modelado. Para a construção do modelo conceitual, optou-se pela utilização de Mapas Conceituais (Moreira e Buchweitz, 1987; Novak, 1981, 1990; Novak e Gowin, 1984), desenvolvidos para representar, de forma gráfica, conceitos e seus inter-relacionamentos. A técnica de Mapas Conceituais foi escolhida, sobretudo, devido à facilidade de modelar a estrutura hierárquica do domínio de conhecimento, além de serem intuitivos para usuários não familiarizados. Os Mapas Conceituais, utilizados no nível conceitual da abordagem *ALM-CID*, foram estendidos com primitivas para a representação de taxonomia e composição de conceitos.

A título de ilustração na Figura 2.2 é apresentado o modelo conceitual construído para representar o domínio de conhecimento envolvendo o critério de teste Análise de Mutantes (Barbosa e Maldonado, 2006a). De modo geral, o critério Análise de Mutantes tem por objetivo testar a corretude de um programa por meio da geração de programas mutantes (programas criados com pequenas alterações sintáticas) e da construção de casos de teste capazes de diferenciar o comportamento do programa original e dos seus mutantes (Agrawal et al., 1989).

De acordo com a Figura 2.2 a Análise de Mutantes, um dos Critérios de Teste pertencentes à Técnica Baseada em Erros, assume a validade da Hipótese do Programador Competente e do Efeito de Acoplamento. O critério utiliza informações sobre Erros Típicos para derivar casos de teste e possui um Escore de Mutação associado. Erros Típicos são modelados por meio de Operadores de Mutação, dando origem a programas Mutantes que caracterizam Transformações Sintáticas de Programas, podendo ser classificados em Mutantes Mortos, Mutantes Equivalentes e Mutantes Error-Revealing.

É importante ressaltar o caráter subjetivo da modelagem conceitual, em que a construção dos modelos pode variar de acordo com o perfil do projetista envolvido na sua construção. De fato diversos outros modelos distintos podem ser definidos para representar o mesmo domínio de conhecimento.

Figura 2.2: Modelo conceitual do módulo *Teste de Software: Teoria e Prática* – Critério Análise de Mutantes (Barbosa, 2004).

Modelo Instrucional

O modelo instrucional é responsável pela definição de informações adicionais relativas aos conceitos previamente identificados. Nessa etapa são definidos e modelados itens de informação e elementos instrucionais. Por itens de informação compreendem-se os elementos significativos ao domínio de conhecimento que devem ser representados e incorporados ao conteúdo educacional (Barbosa, 2004). Considerando-se a teoria de Merrill (1983), tem-se que itens de informação podem ser caracterizados em: (1) fatos – partes de informação logicamente associadas – nomes, datas e eventos são exemplos de fatos; (2) conceitos – constituem grande parte da linguagem e seu entendimento é essencial para a comunicação – símbolos, eventos e objetos que compartilham características e são identificados pelo mesmo nome;

(3) procedimentos – conjuntos ordenados de passos, que visam a resolver um problema ou atingir um objetivo; e (4) princípios – explicações e deduções sobre o porquê de determinados acontecimentos e das maneiras específicas com que os mesmos ocorrem.

Os elementos instrucionais são informações que complementam os conceitos e demais itens de informações já definidos. Desse modo, por elementos instrucionais entendem-se as descrições, exemplos, exercícios, sugestões de estudo e quaisquer outros mecanismos que possam ser associados aos conceitos do domínio de conhecimento. Partindo disso, Barbosa (2004) representou os elementos instrucionais em três categorias: (1) elementos explanatórios – utilizados para ilustrar e complementar um dado item de informação tais como, exemplos, explicações, referências; (2) elementos exploratórios – permitem que o aprendiz navegue pelo domínio de conhecimento e pratique os conceitos relacionados tais como, problemas com sugestões, dicas para resolução; e (3) elementos de avaliação – permitem avaliar o desempenho e o comportamento do aprendiz, podendo ser classificados em elementos de avaliação diagnóstica, formativa ou somativa.

De modo geral, os itens de informação e elementos instrucionais diferem no fato de que os itens de informação correspondem às partes essenciais do conteúdo educacional enquanto os elementos instrucionais são elementos complementares, que podem ser incorporados ou não ao conteúdo dependendo do contexto didático relacionado.

Para caracterizar o modelo instrucional definido na abordagem *AIM-CID* é utilizado o modelo HMBS (Turine, 1998) – modelo para a especificação da estrutura semântica de hiperdocumentos, que utiliza a técnica de *Statecharts* (Harel, 1987) como modelo de especificação formal subjacente. O HMBS, desvinculado da representação de transições, eventos e mecanismos de história, e estendido para a representação de diferentes categorias de conhecimento (itens de informação e elementos instrucionais), foi denominado *HMBS/Instrucional*. Na Figura 2.3 é ilustrado o modelo instrucional para o critério Análise de Mutantes.

Para a construção do modelo instrucional, são identificadas outras informações relevantes que possam ser associadas aos conceitos representados no modelo conceitual da Figura 2.2. Por exemplo, de acordo com a Figura 2.3, o estado **AnáliseMutantes** deve conter o conceito propriamente dito (**AM:conceito:texto**), fatos (**AM:fato:texto**) e princípios (**ProgramadorCompetente:princípio:texto** e **Acoplamento:princípio:texto**) associados. A Figura 2.4 ilustra as diferentes categorias de conhecimento consideradas no modelo instrucional e sua representação final nas transparências referentes ao estado **AnáliseMutantes**, no módulo *Teste de Software: Teoria e Prática* (Barbosa, 2004).

Modelo Didático

O modelo didático é responsável por associar os objetos anteriormente modelados, estabelecendo uma sequência de apresentação entre eles. A partir de um mesmo modelo instrucional

Figura 2.3: Modelo instrucional (*HMBS/Instrucional*) do módulo *Teste de Software: Teoria e Prática* – critério Análise de Mutantes (Barbosa, 2004).

é possível derivar diferentes modelos didáticos, ou seja, modelos didáticos correspondem a diferentes visões de um modelo instrucional. Desse modo, diferentes maneiras de utilização e

Figura 2.4: Transparências referentes à Análise de Mutantes (módulo educacional *Teste de Software: Teoria e Prática*) (Barbosa, 2004).

disponibilização de conteúdo educacional, estabelecidas segundo objetivos do curso, estratégia pedagógica do professor e perfil de cada aprendiz podem, por exemplo, ser representadas por meio de modelos didáticos distintos (Barbosa, 2004; Barbosa et al., 2002).

Basicamente, os modelos didáticos são responsáveis pelo estabelecimento de relações de precedência (pré-requisitos) e relacionamentos didáticos, definindo sequências de apresentação entre os objetos caracterizados no modelo instrucional (itens de informação e elementos instrucionais).

Além disso, aspectos comportamentais também podem ser especificados no modelo didático. As transformações didáticas mostram a dinâmica do conteúdo educacional, representando a forma com que o espaço didático é modificado na medida em que é percorrido pelo aprendiz, ou seja, quais informações são ativadas e quais são desativadas quando um dado relacionamento é percorrido. Desse modo, a especificação dos aspectos comportamentais pode auxiliar no estabelecimento de contextos dinâmicos de aprendizado, em que os elementos do contexto podem variar conforme o desempenho dos aprendizes.

É importante observar que nenhuma abordagem pedagógica específica é considerada na construção e/ou avaliação dos modelos didáticos segundo a *AIM-CID* (Barbosa, 2004). Sendo assim, aspectos pedagógicos relacionados a diferentes paradigmas educacionais (construtivismo, cognitivismo, comportamentalismo, etc) devem ser estabelecidos pelo próprio

instrutor/mediador e transpostos para o modelo didático por meio das sequências de apresentação (ordem pedagógica) especificadas.

No nível didático da abordagem *AIM-CTD*, além dos aspectos já tratados no nível instrucional, elementos associados à navegação e à especificação de aspectos comportamentais do hiperdocumento também foram representados. Desse modo, com base nas informações comportamentais, o modelo possibilita a personalização e o controle do caminho a ser percorrido pelo aprendiz em tempo de execução. Sob essa perspectiva, uma extensão sugerida ao modelo HMBS original refere-se a ideia de estados *DD* (*Dynamically Defined* – Definido Dinamicamente), que apresentam as seguintes propriedades (Barbosa, 2004):

1. Apenas um único estado pode estar ativo em um dado momento;
2. Um estado *DD* não incorpora a representação de estado inicial (*default*), o qual é definido dinamicamente em tempo de execução;
3. Todos os subestados de um estado *DD* são totalmente conexos (estão totalmente conectados uns aos outros);
4. A saída de um estado *DD* pode ativar quaisquer estados da hierarquia de estados-pai-*DD*.

No modelo didático definem-se três tipos de especificações sobre os aspectos de navegação. Na especificação aberta estão representadas todas as possíveis sequências de apresentação entre os objetos modelados. Na especificação fechada os aspectos de navegação são pré-determinados, ou seja, a ordem de apresentação dos objetos é fixa, devendo ser previamente definida pelo especialista e/ou instrutor responsável pelo módulo. Na especificação parcialmente aberta algumas possibilidades de navegação são consideradas enquanto outras são pré-determinadas (Barbosa, 2004).

Os estados *DD* são utilizados como apoio à representação de especificação aberta dos aspectos de navegação. A ideia de especificação aberta refere-se à possibilidade de representar as sequências de apresentação entre conceitos e informações pertinentes de maneira flexível e personalizada. Dependendo de fatores como duração do curso, objetivos de aprendizado e público-alvo, maneiras diferenciadas de apresentação e navegação pelo mesmo conteúdo são exigidas. Uma especificação aberta permite que todas as possíveis sequências de apresentação do conteúdo sejam representadas no mesmo modelo didático. Assim, a partir de um único modelo, várias implementações do mesmo conteúdo podem ser geradas em função dos fatores mencionados.

A especificação aberta apoia, em nível de modelagem e projeto, o estabelecimento de contextos de aprendizado diferenciados. Desse modo, quando fisicamente implementada no

módulo educacional ela possibilita ao usuário uma total liberdade para decidir, dinamicamente, quais tópicos devem ser abordados e em que ordem os mesmos devem ser apresentados. O modelo HMBS, estendido para a representação de estados *DD* no nível didático da abordagem *AIM-CID*, foi denominado *HMBS/Didático*.

A Figura 2.5 ilustra o modelo didático elaborado considerando-se o domínio de Análise de Mutantes. O módulo foi derivado utilizando uma especificação aberta dos aspectos de navegação, na qual estão representadas todas as possíveis sequências de apresentação entre os objetos modelados.

Para garantir a clareza e legibilidade do modelo, a Figura 2.5 não representa as transições de um estado *DD*, as quais ficam implícitas uma vez que as propriedades de um estado *DD* estabelecem que todos os seus subestados são totalmente conexos. Um exemplo de estado *DD* é o estado *DetalhesAnáliseMutantes*. Esse estado deve comportar-se da seguinte maneira (Barbosa, 2004):

- De acordo com a Propriedade (3), todos os subestados de *DetalhesAnáliseMutantes* – *OperadorMutaçao*, *EscoreMutaçao*, *Aplicação*, *GeralMutante* e *GeralAbordagens* – estão conectados entre si por transições implícitas, as quais estabelecem a navegação entre eles. A partir de *OperadorMutaçao*, por exemplo, pode-se atingir os estados *EscoreMutaçao*, *Aplicação*, *GeralMutante* e *GeralAbordagens*.
- De acordo com a Propriedade (4), considerando a sequência (*GeralMutante*, *DetalhesAnáliseMutantes*, *GeralAnáliseMutantes*, *TécnicaBaseadaErros*, *TécnicaTeste*, *TesteSoftware_TeoriaPratica*) como sendo a hierarquia de estados-pai-*DD* de *Mutante*, tem-se, por exemplo, que a partir de *Mutante* podem ser atingidos todos os estados de *DetalhesAnáliseMutantes*: *OperadorMutaçao*, *EscoreMutaçao*, *Aplicação*, *GeralMutante* e *GeralAbordagens*. Para definir o conjunto completo de estados que podem ser atingidos a partir do estado *Mutante*, a mesma análise deve ser realizada para todos os estados da hierarquia de estados-pai-*DD* de *GeralMutante*.

É importante observar que caso sejam utilizados os modelos estendidos do HMBS – *HMBS/Instrucional* e *HMBS/Didático* – nenhum suporte automatizado à sua construção é fornecido e, desse modo, as atividades de modelagem instrucional e modelagem didática devem ser conduzidas manualmente. Tal limitação motivou a condução de atividades envolvendo o estabelecimento de mecanismos automatizados que apoiem a construção dos modelos propostos, explorados no escopo deste trabalho. A ferramenta *AIM-Tool*, descrita no Capítulo 4, apoia a construção de modelos da abordagem *AIM-CID*, possibilitando a edição e interpretação dos mesmos, além de permitir sua execução, explorando aspectos modelagem compartilhada do domínio de conhecimento e apresentação de conteúdos educacionais.

Figura 2.5: Modelo didático (*HMBS/Didático*) do módulo *Teste de Software: Teoria e Prática* (especificação aberta) – critério Análise de Mutantes (Barbosa, 2004).

Figura 2.6: Transparências referentes a **Mutante** – versão aberta, parcialmente aberta e fechada (Barbosa, 2004).

2.2.2 Processo Padrão para o Desenvolvimento de Módulos Educacionais

Além da abordagem *AZM-CID*, também foi proposto no trabalho de Barbosa (2004) um Processo Padrão para o Desenvolvimento de Módulos Educacionais (PP-DME).

O processo foi estabelecido com base na norma ISO/IEC 12207 (International Organization for Standardization, 1995), a qual foi adaptada ao contexto de produção de módulos

educacionais por meio da inclusão de práticas específicas de projeto instrucional (Dick et al., 2001; Lee e Owens, 2000; Seels e Glasgow, 1998) e modelagem de conteúdos educacionais (Barbosa e Maldonado, 2004, 2006b; Barbosa et al., 2002). Questões associadas ao desenvolvimento cooperativo e distribuído (Maidantchik, 1999; Maidantchik e Rocha, 2002) também foram preliminarmente investigadas.

As atividades do ciclo de vida de um módulo educacional foram estabelecidas por meio de três classes de processos (Barbosa, 2004): (1) processos fundamentais; (2) processos de apoio; e (3) processos organizacionais.

Comparados aos processos fundamentais da norma ISO/IEC 12207, os processos fundamentais para módulos educacionais diferem, sobretudo, com respeito à inclusão do Processo de Disponibilização, que trata das atividades e tarefas do instrutor/mediador do módulo. Já os processos de apoio diferem da norma pela inclusão do Processo de Gerência do Conhecimento, responsável por tratar os aspectos pertinentes ao gerenciamento e controle do conhecimento sob o qual o módulo está fundamentado. Quanto aos processos organizacionais, estes se diferenciam basicamente pela definição do Processo de *Copyright* e Licença, no qual são abordados aspectos pertinentes aos direitos autorais e aos termos de uso e distribuição do módulo. Observa-se que a atividade de modelagem de conteúdos educacionais é abordada como parte do Processo de Desenvolvimento, pertencente a categoria de Processos Fundamentais.

Além da padronização de processos para o desenvolvimento de módulos educacionais, aspectos associados à especialização e instanciación do processo padrão definido também foram abordados. Adicionalmente, foi proposto um modelo de maturidade de processos para a elaboração de módulos educacionais – *CMMI/Educacional* – visando a apoiar a especialização do processo padrão em diferentes níveis de maturidade. Mais informações a respeito do processo padrão definido podem ser encontrados nos trabalhos (Barbosa, 2004; Barbosa e Maldonado, 2006c, 2008, 2010).

2.3 Conteúdos Educacionais Livres

O movimento de software livre (FLOSS - *Free Libre and Open Source Software*) impulsionou várias iniciativas de projetos como o Apache, Linux, Mozilla, entre outros, têm inspirado pesquisadores a adotarem a concepção da liberdade de utilização dos diversos recursos para a disponibilização de material educacional, principalmente no contexto de educação a distância.

Com a evolução dos objetos de aprendizado, Wiley definiu, em 1998, o termo *Open Content* e as licenças *Open Content License* e *Open Publication License* para serem aplicadas no desenvolvimento de conteúdos digitais educacionais com base nos conceitos do movimento FLOSS.

A *Open Content License* foi criada seguindo as premissas do GNU GPL com o objetivo de expressar três graus de liberdade para o conteúdo: direito de cópias, direito de redistribuição e direito de modificação. Na sequência, foi criada a *Open Publication Licence* melhorando pontos importantes da *Open Content License* como, por exemplo, possibilitar a total reprodução e redistribuição de qualquer publicação no formato de livro com a correta citação do autor ou editor original, ou seja, a propriedade da licença é de seu autor.

Com a idéia de conteúdos abertos, em 2001, Lawrence Lessig, professor da Universidade de *Stanford*, idealizou a *Creative Commons*, que representa um conjunto de licenças para a gestão aberta, livre e compartilhada de conteúdos. Paralelamente, o *Massachusetts Institute of Technology* (MIT) criou iniciativas como o MIT *OpenCourseWare* (MIT OCW), tendo como principal objetivo oferecer o acesso *on-line* ao conteúdo dos cursos do MIT, de forma livre e aberta a todo o público. Na Figura 2.7 é ilustrada a lista de cursos que disponibilizam material didático livre do MIT OCW.

Figura 2.7: MIT Open Course Ware – lista de cursos disponíveis por departamento.

Esses movimentos instigaram a iniciativa de inúmeras instituições acadêmicas do mundo no sentido de disponibilizar conteúdos abertos desenvolvidos de forma compartilhada, assim como é feito no desenvolvimento de software livre. Entre as instituições que adotaram a disponibilização de conteúdos abertos destacam-se: (Kirkpatrick, 2006): *Utah State University*, *Johns Hopkins School of Public Health*, *Tufts University*, *University of Notre Dame*, entre outras.

Em 2002, a UNESCO promoveu um evento destinado à discussão de disponibilização de recursos educacionais abertos, iniciando o termo *Open Educational Resources* (OER).

Segundo Hylén (2006), esses recursos são caracterizados como materiais educacionais digitais disponibilizados de forma livre e aberta, com a finalidade de ensino, aprendizado e pesquisa.

Os OERs abrangem todo e qualquer conteúdo de aprendizagem como cursos, módulos de conteúdo, objetos de aprendizagem entre outros, além de incluir ferramentas de apoio como ambientes e sistemas educacionais e repositórios de objetos de aprendizado. Além disso, os OER contemplam os recursos de implementação como licenças para o reuso e disseminação do material aberto (Hylén, 2006).

É importante observar que o desenvolvimento de conteúdos educacionais livres podem ser apoiados pela atividade de modelagem sobretudos a partir da construção dos modelos conceitual, instrucional e didático da abordagem *AIM-CTD*. Uma característica importante associada está na possibilidade da construção de conteúdos de forma distribuída, de modo que equipes dispersas geograficamente possam contribuir efetivamente para sua elaboração. Além disso, aspectos de evolução contínua, em resposta às necessidades e opiniões dos usuários, também são considerados na construção de conteúdos educacionais, devido o caráter dinâmico e evolutivo associado ao conhecimento (Barbosa, 2004; Barbosa e Maldonado, 2006b).

2.4 Padrões e Interoperabilidade

Com o avanço da informática e da comunicação, a quantidade de informações digitais disponíveis, que integram multimídia e interatividade, têm aumentado de forma significativa de modo ampliar a diversidade de recursos educacionais (objetos de aprendizado) disponíveis. No entanto, o projeto e desenvolvimento de tais objetos demandam muito esforço, envolvendo grandes investimentos em recursos humanos e financeiros. Uma das alternativas para enfrentar essa dificuldade é o compartilhamento de objetos de aprendizado, de forma organizada e eficiente, garantindo sua reutilização entre projetos distintos de aprendizagem (Prata e Nascimento, 2007).

Segundo o IEEE LSTC (*Learning Technology Standards Committee*) (Hodgins et al., 2002), um objeto de aprendizado corresponde à qualquer entidade, digital ou não digital, que possa ser usada, reutilizada ou referenciada pelas tecnologias que apoiem o aprendizado. Exemplos de objetos de aprendizado incluem conteúdos multimídia, objetivos de aprendizado, ferramentas de software específicas ao domínio de conhecimento e, ainda, pessoas e organizações ou outras entidades que utilizem das tecnologias citadas durante o processo de aprendizado. É importante observar que módulos educacionais também podem ser desenvolvidos como uma combinação de objetos de aprendizado, os quais podem ser utilizados mais de uma vez, por mais de um autor, em sequências e com objetivos completamente diferentes (Melo et al., 2003).

Atualmente existem diversas especificações que normatizam o desenvolvimento e identificação desses objetos. Entre as principais especificações destacam-se o LOM (Hodgins et al., 2002) e o SCORM (ADL – SCORM, 2006). Essas especificações têm por objetivo preservar as características de reusabilidade, portabilidade e interoperabilidade dos objetos de aprendizado (Hamel e Ryan-Jones, 2002; Muzio et al., 2002; Wiley, 2000, 2001), permitindo que sejam desenvolvidos conteúdos independentes de uma ferramenta de autoria, de um ambiente de aprendizagem ou mesmo de uma plataforma de hardware e software. Uma visão geral sobre o LOM e o SCORM é apresentada na próxima seção.

2.4.1 IEEE 1484 – LOM

O LTSC (*Learning Technology Standards Committee*) foi instituído pelo IEEE, pela *Computer Society* e pelo *Standards Activity Board* com o objetivo de desenvolver normas, orientações e práticas recomendadas na área de aprendizado apoiado por computador. O resultado de tais esforços foi o padrão IEEE 1484, usualmente referenciado como *Learning Object Metadata* (LOM) (Hodgins et al., 2002).

Em linhas gerais, tal padrão foi desenvolvido por meio da especificação de um esquema conceitual comum, para garantir que *bindings* (representações de metadados por meio de linguagens conhecidas, como XML por exemplo) representem uma maior interoperabilidade semântica entre metadados de objetos de aprendizado.

O padrão tem por objetivo facilitar a busca, evolução, aquisição e uso de aprendizado por parte de aprendizes, tutores ou processos automatizados de software, além de facilitar o compartilhamento e intercâmbio de objetos de aprendizado, possibilitando o desenvolvimento de catálogos e inventários, considerando a diversidade cultural e de idiomas (Hodgins et al., 2002).

Uma das motivações que levam à necessidade de um sistema de cadastramento de objetos de aprendizado é o resultado insatisfatório que se obtém ao tentar recuperar material para compor uma unidade de aprendizado, pois as ferramentas de busca nem sempre expressam com fidelidade os requisitos desejados. Assim, são recuperadas muitas referências indicando materiais que não são apropriados ou mesmo passíveis de utilização.

O metadado de um objeto educacional descreve características relevantes que são utilizadas para sua catalogação em repositórios para que os mesmos possam ser reusáveis e posteriormente recuperados por sistemas de busca e utilizados por algum LMS (*Learning Management Systems*) com o propósito de compor unidades de aprendizado.

Organizações para a padronização, como o IEEE (1484.12.1 *Standard for Learning Object Metadata*) e o ISO (SC 36 WG 2 – *Information Technology for Learning, Education, and Training*), cujo objetivo é oferecer suporte à catalogação dos objetos de aprendizado, elaboraram propostas para a estruturação e categorização dos metadados. A proposta do padrão

IEEE 1484 é descrever um conjunto de atributos para categorizar objetos educacionais. Esses atributos são organizados em um esquema conceitual de dados (categorias) que definem a estrutura da instância de metadados de um objeto de aprendizado. Uma instância de metadados de um objeto de aprendizado descreve as características relevantes desse objeto por meio de atributos relacionados a cada categoria, como ilustrado na Tabela 2.1.

Tabela 2.1: Categorias e atributos do LOM.

Categorias	Descrição	Exemplos de Atributos
Geral	Agrupar informações gerais que descrevem o objeto.	Identificador, título do objeto, linguagem, descrição, palavras-chave, cobertura, estrutura, nível de agregação.
Ciclo de vida	Agrupar informações que descrevem as características relacionadas ao histórico e estado atual dos objetos e todos aqueles que o têm afetado durante sua evolução.	Versão, status, tipo de contribuição, entidades que contribuíram e data.
Metadados	Agrupar dados sobre a instância de metadados em si.	Esquema de catalogação, referência de catalogação.
Técnica	Agrupar os requisitos e características técnicas do objeto.	Formato, tamanho, localização, tipo de tecnologia, nome da tecnologia, requisitos, duração, comentários sobre a instalação.
Educacional	Agrupar as características educacionais e pedagógicas do objeto.	Tipo de interatividade, recurso de aprendizagem, nível de interatividade, usuário final esperado, ambiente de utilização, faixa etária, contexto, dificuldade, descrição.
Direitos	Agrupar os direitos de propriedade intelectual e as condições de uso do objeto.	Custo, direito autoral e condições de uso.
Relacionamento	Agrupar características que definem o relacionamento entre objetos de aprendizado	Tipo (natureza do relacionamento), recurso (identificador e descrição).
Anotação	Provê comentário sobre o uso educacional do objeto de aprendizado.	Entidade, data e descrição.
Classificação	Descreve o objeto de aprendizado em relação a algum sistema de classificação.	Propósito, caminho taxonômico, descrição e palavra-chave.

É importante observar que todos os elementos de dados (categorias e seus atributos) podem ser descritos no formato de uma “árvore”, conforme ilustrado na Figura 2.8. Isso significa que para incluir um elemento de dados, por exemplo, um **Catálogo** pertencente ao atributo **Identificador** na categoria **Geral**, deve-se obedecer ao esquema de um relacionamento de hierarquia, ou seja, caso se decida definir o conceito mais “externos”, é preciso que se tenha também definido os componentes que o antecedem na sequência de derivação a fim de garantir que não haverá “folhas” soltas na estrutura hierárquica.

Figura 2.8: Árvore resumida do LOM (Pereira et al., 2003).

Ressalta-se que alguns tipos de extensão podem ser feitas utilizando-se a categoria de **Classificação**, de tal forma que dentro desse item qualquer outro sistema de classificação possa ser criado (Hodgins et al., 2002).

2.4.2 SCORM

O SCORM (*Sharable Content Object Reference Model*) (ADL – SCORM, 2006) é um padrão desenvolvido e distribuído pela *Advanced Distributed Learning* (ADL), cujo principal objetivo é permitir o reúso de material didático disponibilizado em ambientes de ensino. O padrão define e descreve um conjunto de normas que especificam tanto os requisitos para o material didático (objeto de aprendizado) como para os ambientes de ensino que suportam esse material. Além disso, o SCORM é caracterizado como um modelo de referência, ou seja, um conjunto unificado de especificações para a disponibilização de conteúdos e serviços. Na

Na última versão, SCORM 2004, a ADL publicou as especificações em quatro conjuntos (ADL – SCORM, 2006):

- Visão Geral: contém uma visão geral, define os conceitos fundamentais do SCORM e um sumário das especificações técnicas e orientações contidas nas demais seções;
- Modelo de Agregação de Conteúdo: especifica o dicionário de metadados (modelo de conteúdo), o empacotamento de conteúdos e o XML dos metadados do pacote.
- Ambiente de Execução: responsável pela comunicação do objeto de aprendizado com o LMS, ou seja, comunicações sobre o andamento do curso e sobre a evolução do aluno; e
- Sequenciamento e Navegação: orienta sobre como os objetos de aprendizado devem ser sequenciados por um LMS e quais as avaliações que habilitam o aprendiz a progredir no módulo.

Segundo Vieira e Nicoleit (2007), o Modelo de Agregação de Conteúdo permite criar um objeto de aprendizado com informações completas sobre o conteúdo, a localização e a distribuição. Esse modelo define os recursos que podem ser utilizados na montagem de um objeto de aprendizado e como esses recursos devem ser agrupados para formar componentes maiores. Um *asset* é o recurso digital mais básico que pode ser utilizado dentro do conteúdo educacional. Ele é formado por páginas Web, imagens, recursos de áudio, animações, entre outros. O *Sharable Content Object* (SCO) representa uma associação de vários *assets*. Ele possui a capacidade de se comunicar com os LMS e deve ser construído como unidades pequenas que possam ser reutilizadas em contextos diversos.

A organização do conteúdo é dividida em dois grupos: atividades e recursos. Os recursos são formados por *assets* e SCOs. As atividades são como módulos ou tópicos do objeto de aprendizado. Cada objeto de aprendizado pode ser composto por várias atividades; uma atividade, por sua vez, pode conter outras atividades ou simplesmente conter os recursos (ADL – SCORM, 2006; Prata e Nascimento, 2007).

O encapsulamento de conteúdo garante a disponibilização do objeto de aprendizado no LMS para facilitar a busca e o acesso. O pacote é composto de dois elementos: um arquivo XML (que descreve a estrutura do objeto e o seu conteúdo) e arquivos multimídia (animações, textos, html, applets Java, entre outros). Os LMS utilizam as informações do arquivo XML para conhecer a organização do conteúdo do objeto.

O arquivo de metadados é o recurso utilizado para rotular o objeto de aprendizado. O esquema de criação dos metadados SCORM segue o estabelecido pelo IEEE-LOM (Hodgins et al., 2002), para garantir interoperabilidade com a maioria dos LMS. Os metadados têm

por objetivo criar um tipo de cabeçalho para o objeto de aprendizado com informações padronizadas (título do objeto, descrição do seu conteúdo, autoria, tempo de vida, entre outras características) para facilitar a busca e acesso ao conteúdo pelos LMS (ADL – SCORM, 2006).

O SCORM também especifica os métodos para conduzir a comunicação entre o curso e o LMS. Para tal comunicação, o SCORM utiliza o Ambiente de Execução no qual são incluídas as comunicações sobre a situação do curso, ou seja, quais materiais estão sendo apresentados ao aprendiz, bem como informações sobre o progresso do aprendiz durante o curso. A padronização dessas comunicações minimiza os problemas associados com a migração de cursos entre diferentes LMS, uma vez que cada ambiente utiliza a própria forma de gerenciamento e gravação do progresso do aprendiz durante um curso. Desse modo, para que um conteúdo educacional seja acessado e a sua evolução seja monitorada, é necessário que o próprio objeto se comunique com o LMS a fim de fornecer informações necessárias ao gerenciamento do conteúdo (ADL – SCORM, 2006; Prata e Nascimento, 2007).

O Sequenciamento e Navegação descrevem como os conteúdos podem ser organizados e sequenciados e como um LMS compatível deve interpretar essas regras de sequenciamento. É importante destacar que o SCORM, na sua versão atual, preocupa-se basicamente com o conteúdo, em como ele é organizado e sequenciado, como será mostrado e como rastrear as ações do aprendiz com o conteúdo. Falta, porém, uma maneira de especificar como esse “pacote” de conteúdo pode ser incorporado em contextos relacionados a outras atividades de interação do aprendiz, além da interação autônoma entre o aprendiz e os objetos de aprendizado (ADL – SCORM, 2006).

Por fim, uma limitação encontrada no SCORM refere-se à falta de suporte a abordagens de serviços comuns à utilização de um LMS, tais como sessões de fórum, sessões de *chat*, atividades desenvolvidas pelos aprendizes individualmente ou em grupo entre outros. Para oferecer uma forma de superar essas lacunas, surgiram outras iniciativas, como o *Learning Design* (Koper et al., 2003), a fim de buscar uma maior abrangência em pesquisas relacionadas ao desenvolvimento de objetos de aprendizado (ADL – SCORM, 2006; Prata e Nascimento, 2007).

Ressalta-se que como parte da geração de conteúdos da ferramenta *ALM-Tool* foi considerado o LOM como modelo de conteúdo especificado segundo um dicionário definido em um metadado. Tal padrão foi escolhido por possuir atributos adequados à caracterização de conteúdos educacionais. É importante observar que o padrão LOM compõe o modelo de agregação de conteúdos do SCORM, ou seja, os requisitos definidos pelo SCORM para o material didático são representados por meio do LOM. Desse modo, o conteúdo modelado e produzido pela ferramenta *ALM-Tool* possui características de reusabilidade e interoperabilidade garantidas por meio da adoção do LOM.

2.5 Ambientes e Ferramentas de Apoio ao Ensino e Aprendizado

Conforme discutido anteriormente, ambientes educacionais e ferramentas de apoio, são caracterizados como parte da infra-estrutura necessária à integração e disponibilização dos materiais didáticos associados aos módulos educacionais.

Nesse sentido, uma das linhas que têm sido objeto de pesquisa refere-se ao desenvolvimento de ambientes educacionais (Calvo et al., 2003; Counterman et al., 2004; Eberspächer et al., 1999; IWT, 2006; Lucena et al., 1998; Moodle, 2006; Rocha, 2003; Tidia, 2007; Yaskin e Everhart, 2002). De modo geral, esses ambientes têm por objetivo apoiar o processo de ensino e aprendizado fornecendo um conjunto de ferramentas de comunicação síncrona e assíncrona para apoiar a condução de trabalhos colaborativos. Além disso, tais ambientes oferecem suporte à disponibilização de material didático e à avaliação e acompanhamento dos aprendizes, constituindo mecanismos importantes ao desenvolvimento instrucional (Eberspächer et al., 1999; Goldberg et al., 1996).

A seguir são apresentados alguns dos principais ambientes educacionais existentes, fornecendo uma visão geral a respeito das pesquisas conduzidas na área.

AulaNet

O *AulaNet* (Fuks, 2000; Lucena et al., 2000, 1999, 1998) consiste em um ambiente fundamentado numa abordagem *groupware* para o aprendizado cooperativo baseado na Web. Esse ambiente foi desenvolvido pelo Laboratório de Engenharia de Software (LES) do Departamento de Informática da Pontifícia Universidade Católica do Rio de Janeiro (PUC/RJ).

O *AulaNet* dispõe de recursos para a criação e manutenção de cursos para atender a educação a distância ou para complementação atividades presenciais ou treinamento de profissionais. Além disso, é baseado nas seguintes premissas (Fuks, 2000; Lucena et al., 1999, 1998): (1) os cursos criados devem possuir grande capacidade de interatividade, de forma a atrair a participação intensa do aluno no processo de aprendizado (*learningware*); (2) o autor do curso não precisa ser necessariamente um especialista em Internet; (3) os recursos oferecidos para a criação de cursos devem corresponder aos de uma sala de aula convencional, acrescidos de outros normalmente disponíveis no ambiente Web; e (4) deve ser possível a reutilização de conteúdos já existentes em mídia digital, por exemplo, por meio da importação de arquivos.

A fim de estruturar o *AulaNet*, foi utilizado o modelo de colaboração denominado ‘3C’ (Ellis et al., 1991), como mostra a Figura 2.9. O modelo oferece três tipos de serviços integrados: comunicação, coordenação e colaboração (Fuks, 2000; Gerosa et al., 2004).

Figura 2.9: Modelo de colaboração 3C (Gerosa et al., 2004).

A comunicação, em um grupo de trabalho, envolve a negociação de compromissos e conhecimento. Esse serviço inclui ferramentas de correio eletrônico e lista de discussão, ferramentas de conferência assíncrona textual (*newsgroup*), ferramentas de conferência síncrona textual (*chat*) e ferramentas de videoconferência.

Por meio da coordenação, o grupo lida com conflitos e se organiza de maneira a evitar que os esforços de comunicação e cooperação sejam perdidos. Dentro desse contexto, os serviços de coordenação são baseados no agendamento de eventos e na competência dos aprendizes. Os principais serviços oferecidos sob essa perspectiva são: administrativos (agenda, notícias do curso, cadastro de instrutores, matrícula e mensagens automáticas), relatórios de participação, planos de aula, tarefas e avaliações.

A cooperação é a operação conjunta dos membros do grupo em um espaço compartilhado. A tecnologia da Internet possibilita a criação de espaços de compartilhamento e troca de informações. Assim, os serviços de cooperação fornecidos pelo *AulaNet* oferecem suporte à bibliografia, webliografia (*links* para sites na Internet), documentação, *download*, co-autoria de docentes (indicação de outros docentes para participarem como co-autores do curso) e co-autoria de aprendizes (indicação de aprendizes para criarem conteúdos para o curso).

Ao aprendiz é fornecido um menu, previamente configurado pelo professor, disponibilizando os mecanismos selecionados de comunicação, coordenação e cooperação. Esse menu proporciona ao aprendiz a facilidade navegacional, além de minimizar problemas de desorientação.

Ressalta-se que, apesar do *AulaNet* facilitar a criação, administração, manutenção e assistência de cursos à distância utilizando um alto nível de interatividade, este não possui dispositivos capazes de garantir a qualidade do material didático disponibilizado e dos cursos oferecidos (Fuks, 2000; Lucena et al., 1999, 1998).

Moodle

O Moodle (*Modular Object-Oriented Dynamic Learning Environment*) (Dougiamas e Taylor, 2003; Moodle, 2006; MoodleDocs, 2006) é um sistema de gestão de aprendizagem (LMS - *Learning Management System*) cujo desenvolvimento teve início em 1999.

As principais funcionalidades do ambiente são (Dougiamas e Taylor, 2003; Moodle, 2006; MoodleDocs, 2006):

- Gerenciamento do site: o site é gerenciado pelo usuário administrador, definido na instalação.
- Gerenciamento de usuários: cada usuário requer apenas uma conta para todo servidor. Desse modo é possível que o aluno participe de vários cursos utilizando a mesma conta.
- Gerenciamento dos cursos: o tutor de um curso possui controle sobre toda a configuração do mesmo. Atividades como fóruns, notícias, *quizz*, recursos, exercícios, avaliações, *chats*, *workshops*, também são programadas pelo tutor. A maior parte das áreas de texto – fontes, recursos, fóruns, notícias – podem ser editadas usando um editor HTML incorporado.
- Módulos de interação entre os usuários: permitem a troca de informações entre os usuários do Moodle. O sucesso na comunicação interativa ocorre especificamente por meio do conhecimento dos usuários e da manipulação adequada dos recursos oferecidos. Os seguintes módulos estão disponíveis no ambiente:
 - Módulo de tarefas: auxilia o professor a conduzir tarefas e avaliar os alunos. O *feedback* do professor é adicionado na página da tarefa para cada aluno e uma notificação é enviada por e-mail.
 - Módulo de *chat*: único módulo do sistema que permite interação síncrona por texto entre os alunos e o professor.
 - Módulo de enquete: similar a uma pesquisa, pode ser usado para votação ou para obter *feedback* de cada estudante.
 - Módulo de fórum: os diferentes tipos de fóruns são disponibilizados exclusivamente pelos professores e são utilizados para a comunicação assíncrona entre os participantes ou, ainda, para a publicação de notícias dos cursos. Podem ser criados quantos fóruns de discussão forem necessários.
 - Módulo *quizz*: permite a publicação e correções automáticas de questões em diversos formatos – respostas curtas, verdadeiro ou falso, relacionamento, aleatórias, numéricas, textos descritivos e com imagens integradas.

- Módulo de arquivo: serve para carregar arquivos para o servidor de um modo prático e simples. Suporta exibição de qualquer conteúdo eletrônico tais como documentos, apresentações, animações, vídeos, sons, entre outros.
- Módulo *workshop*: permite compartilhamento e trabalho conjunto dos alunos em documentos onde o professor pode gerenciar e avaliar o trabalho. Os professores podem prover exemplos de documentos para exercícios práticos.
- Módulo glossário: permite a criação de um dicionário com definições dos termos usados nos conteúdos dos cursos.

A ferramenta está disponível como um software livre (*open source*), sob as condições da *GNU Public License* (GPL) (Free Software Foundation, 2007).

Sakai e Ae

O *Sakai* (Counterman et al., 2004; Farmer e Dolphin, 2005) é um sistema desenvolvido por meio da colaboração das universidades: *University of Michigan*, *Stanford University*, *Indiana University* e *Massachusetts Institute of Technology*, que iniciaram o desenvolvimento deste sistema para substituir os sistemas desenvolvidos por cada uma das instituições separadamente. Esse projeto tem como objetivo desenvolver e distribuir um novo ambiente de aprendizagem colaborativo.

O projeto *Sakai* é composto por softwares educacionais pré-integrados, ou seja, um grupo de ferramentas *open source* a fim de produzir um ambiente de aprendizagem colaborativo. Além disso, o *Sakai* é considerado um *framework* descrito sobre o padrão IMS (IMS Global Learning Consortium, 2006; Koper et al., 2003; Leo et al., 2004) e as normas da OKI OSID (OKI, 2008) para prover a integração entre suas camadas e assegurar a portabilidade das ferramentas.

As principais funcionalidades do *Sakai* são (Counterman et al., 2004): (1) suporte à avaliação e à atividades colaborativas (fórum, grupos de discussão, *chat*); (2) compartilhamento de arquivos; (3) lista de emails; (4) suporte às atividades gerenciais (gráficos, estatísticas de grupos, classes e estudantes).

O projeto *Ae* (Aprendizado Eletrônico) é uma iniciativa financiada pela FAPESP (Processo 05/60651-9) dentro do programa *Tidia* (Tecnologia da Informação para o Desenvolvimento da Internet Avançada). O *Tidia* é um programa amplo que integra três linhas de projeto com o objetivo de desenvolvimento da Internet avançada, ou seja, a nova geração da Internet. As três linhas são: (1) o *Kyatera*, voltado para a construção de uma rede de alta velocidade voltado para pesquisas; (2) o próprio *Ae*; e (3) Incubadora Virtual de Conteúdos Digitais, voltada à criação e disponibilização cooperativa de conteúdos digitais abertos.

As funcionalidades da plataforma do *Ae* são voltadas para alunos e professores visando dar suporte ao conteúdo aprendido em aula presencial ou à distância, como mostra a Figura 2.10 (Tidia, 2007). Essa plataforma foi desenvolvida para promover a geração de conhecimento, a formação de recursos humanos e a produção científica, tecnológica e de produtos na área de tecnologia da informação aplicada à educação, explorando as características da Internet avançada apoiada pelo uso de redes de alta velocidade. Nesse ambiente, cada pessoa, pode criar um grupo de atividade personalizado que pode conter recursos como: integração de vídeos do Youtube, edição de textos colaborativos e, no caso dos professores da USP, integração com os sistemas de graduação e pós-graduação, o *Júpiter* e o *Fênix* (Martins, 2009).

Figura 2.10: Página principal e algumas ferramentas do *Ae*.

O desenvolvimento e implantação do ambiente de suporte ao aprendizado eletrônico do projeto *Ae* é desenvolvido em parceria com o projeto *Sakai*, utilizando o núcleo básico do *Sakai* para desenvolver sua plataforma e suas ferramentas colaborativas. Sua arquitetura foi desenvolvida baseada em componentes a fim de facilitar sua construção, implementação, manutenção e evolução, permitindo que novas funcionalidades sejam acrescidas ao longo do tempo.

O processo de gerência de aprendizagem do *Ae* é realizado pelo seguinte conjunto de ferramentas: gerenciador de contexto de aprendizagem, gerenciador de usuário, gerenciador de ações de aprendizagem e gerenciador de ferramentas.

Além das ferramentas de gerência, foram desenvolvidas algumas ferramentas básicas de ambiente de ensino e aprendizagem, tais como: (1) *whiteboard* – ferramenta responsável pela captura de informações do professor durante a aula de forma síncrona, por meio de uma

lousa eletrônica; tais informações são posteriormente disponibilizadas aos alunos locais ou remotos, utilizando *tablet PC* ou computadores; (2) *chat*; (3) comunicador instantâneo; (4) laboratório remoto – ferramenta que engloba todo o ciclo de vida de um experimento, que vai desde a sua criação até a sua efetiva realização pelo usuário; (5) hipertexto; (6) correio eletrônico; e (7) fórum de discussão.

Dentre as inovações propostas pelo *Ae*, está presente a Áudio Conferência em Realidade Aumentada (AC 3D). A proposta da AC 3D é dar uma noção espacial às conversas *on-line* de áudio. Pesquisadores do projeto afirmam que essa técnica pode aumentar a capacidade de aprendizado devido à capacidade do cérebro humano de conseguir focar o áudio. A intenção ao agregar tal ferramenta ao *Ae* é justamente para verificar a veracidade dessa hipótese (Martins, 2009).

Ressalta-se por fim, que o projeto *Ae* envolve pesquisa e desenvolvimento. Nesse sentido, os produtos tecnológicos resultantes foram disponibilizados para serem utilizados por universidades públicas e privadas e por empresas interessadas em treinamento eletrônico, sendo liberados sob a licença GPL (Tidia, 2006).

2.6 Técnicas e Ferramentas de Apoio à Estruturação do Conhecimento

Conforme discutido anteriormente, no trabalho de Barbosa (2004) foi estabelecida uma abordagem integrada de modelagem (*ALM-CTD*), tendo como base a técnica de Mapas Conceituais e o modelo HMBS.

Considerando a atividade de modelagem de conteúdos educacionais, a elaboração de material didático de qualidade requer, sobretudo, a organização e estruturação do domínio de conhecimento que se deseja ensinar. Sob essa perspectiva, técnicas e ferramentas descritas para estruturar o conhecimento, permitem ao autor organizar um domínio complexo de forma a torná-lo mais claro ao aprendiz.

Nas próximas seções são brevemente apresentadas a técnica de Mapas Conceituais e o modelo HMBS, além de algumas iniciativas específicas desenvolvidas para a sua construção. Nesse contexto, são apresentadas ferramentas de apoio à estruturação do conhecimento, as quais foram investigadas para o desenvolvimento da ferramenta *ALM-Tool*, que utiliza Mapas Conceituais e o modelo HMBS em conformidade com a abordagem *ALM-CTD*.

2.6.1 Mapas Conceituais

A técnica de Mapas Conceituais, proposta por Novak (Moreira e Buchweitz, 1987; Novak, 1981, 1990; Novak e Gowin, 1984), foi fundamentada na Teoria da Aprendizagem Significativa

de Ausubel et al. (1978) para representar graficamente, na forma de estruturas hierárquicas ou grafos orientados, um conjunto de conceitos e seus inter-relacionamentos.

Dentro desse contexto, Mapas Conceituais são classificados como diagramas que indicam relações entre conceitos. Os conceitos aparecem dentro de caixas ou nós do grafo enquanto que as relações entre os conceitos são especificadas por meio de textos nos arcos que unem os conceitos (Moreira e Buchweitz, 1987; Novak, 1990).

Mapas Conceituais têm sido amplamente utilizados para muitos fins e em diferentes situações. Eles são mais conhecidos dentro do contexto educacional como uma ferramenta metacognitiva que auxilia o aprendiz a compreender e a modelar o conhecimento. Também são eficientes para (Cañas et al., 2004b): (1) fornecer a síntese do conhecimento do aprendiz; (2) identificar os equívocos; (3) revelar lacunas no conhecimento a ser modelado; (4) promover reflexões sobre o domínio do conhecimento; (5) criar material didático; (6) avaliar o aprendizado; (7) avaliar a eficácia do programa; (8) facilitar a comunicação e promover um entendimento compartilhado entre membros de grupos ; e (9) compreender os processos utilizados pelos pesquisadores para definir novos conhecimentos.

Em relação ao embasamento teórico envolvido, ao propor a Teoria da Aprendizagem Significativa, Ausubel et al. (1978) relaciona um conceito por meio de dois princípios (Moreira, 1997): Diferenciação Progressiva e Reconciliação Integrativa.

Na Aprendizagem Significativa, os conceitos que interagem com o novo conhecimento e servem como base para a atribuição de novos significados modificam-se em função dessa interação, ou seja, adquirem novos significados e se diferenciam progressivamente. Esse processo é dito Diferenciação Progressiva.

Outro processo que ocorre na Aprendizagem Significativa é o estabelecimento de relações entre ideias, conceitos e proposições já estabelecidas na estrutura cognitiva. Elementos existentes na estrutura cognitiva com determinado grau de clareza, estabilidade e diferenciação são percebidos como relacionados, adquirem novos significados e levam a uma reorganização da estrutura cognitiva. Assim, a recombinação de elementos, a reorganização cognitiva e o tipo de relação significativa é referido como Reconciliação Integrativa.

Por serem relativamente simples, Mapas Conceituais são instrumentos que podem levar a profundas modificações na maneira de ensinar, avaliar e aprender. A seguir, duas ferramentas de apoio à construção de Mapas Conceituais são apresentadas: *CMapTools* e *Inspiration*.

2.6.1.1 Ferramenta CMapTools

O *CMapTools* (Cañas et al., 2004a) é uma ferramenta desenvolvida pelo IHMC (*Institute for Human Machine Cognition*) da *University of West Florida* que permite ao usuário construir, navegar e compartilhar modelos de conhecimento representados por Mapas Conceituais. Além do *CMapTools*, o IHMC desenvolveu o *CmapServer*, cujo principal objetivo é armaze-

nar Mapas Conceituais e seus recursos, permitindo ao usuário compartilhá-los por meio da Internet para trabalhar de forma colaborativa com outros usuários.

A ferramenta é distribuída gratuitamente pelo IHMC e desenvolvida utilizando a tecnologia Java, podendo ser executada em várias plataformas. O *CMapTool* possibilita a exportação dos seus mapas no formato HTML, o que garante que os mesmos possam ser trabalhados em diversas ferramentas hipermídia; além dos formatos JPEG, GIF, HTML, XML, entre outros.

Para a representação do conhecimento por meio de Mapas Conceituais, a ferramenta apresenta como estratégia cognitiva a possibilidade de agregar recursos como a adição de sons, imagens, vídeos, textos e até mesmo outros mapas. Na Figura 2.11 é ilustrada uma proposição de modelagem do conhecimento utilizando o *CmapTools* com suporte a recursos hipermídia.

Figura 2.11: Modelagem do conhecimento utilizando a ferramenta *CmapTools* (Cañas et al., 2004a).

Em linhas gerais, o *CMapTools* foi desenvolvido para ser capaz de (Cañas et al., 2004a):

- Fornecer uma boa usabilidade, podendo ser aprendida em pouco tempo por meio de uma interface intuitiva que apóie a construção de grandes conjuntos de conceitos.

- Promover a construção do conhecimento por meio de modelos, permitindo ao usuário expressar sua compreensão sobre um determinado domínio de conhecimento de forma gráfica.
- Permitir a colaboração, por meio da Internet, na qual usuários de todos os domínios possam colaborar e compartilhar os seus conhecimentos na construção de seus modelos.
- Fornecer uma arquitetura flexível, com componentes que podem ser adicionados ou removidos conforme a necessidade do usuário. Permite-se ainda que diferentes versões do programa possam ser adaptadas às necessidades dos diferentes usuários.

2.6.1.2 Ferramenta Inspiration

O *Inspiration* (Inspiration Software Inc., 2003) é uma ferramenta de aprendizagem desenvolvida para proporcionar ao usuário expressar suas ideias em forma de diagramas ou mapas que representam visualmente conceitos e seus relacionamentos. A ferramenta foi desenvolvida para auxiliar na organização de raciocínio e na economia de tempo durante o desenvolvimento de ideias. Além disso, o *Inspiration* foi projetado de forma a oferecer consultas rápidas e fáceis durante a criação de diagramas e esquemas.

Em relação às suas principais funcionalidades, o *Inspiration* possui os seguintes recursos (Inspiration Software Inc., 2003):

- Os diagramas produzidos pela ferramenta podem ser facilmente transferidos para um editor de texto para serem formatados e editados.
- É possível apresentar no diagrama informações de diversas fontes, incluindo documentos criados no *Inspiration* ou em outros aplicativos, fazendo um link para qualquer arquivo.
- A ferramenta permite gravar um áudio e associá-lo a algum símbolo ou tópico, bem como ouvir textos incluídos nos trabalhos (basta clicar em qualquer texto no documento para ouvi-lo).
- A interface do *Inspiration* fornece uma barra de ferramenta intuitiva para usuários não familiarizados, conforme ilustra a Figura 2.12.
- A exportação de arquivos acontece em vários formatos. Destaca-se o formato HTML que permite a publicação de diagramas e esquemas como páginas Web.

O *Inspiration*, em sua versão 7.6, está otimizado para ser compatível com lousas interativas e outros sistemas baseados em caneta digital. Além disso, a ferramenta é compatível com o *Microsoft PowerPoint*. Ressalta-se, entretanto, que a ferramenta *Inspiration* não possui os

Figura 2.12: Tela principal da ferramenta *Inspiration* (Inspiration Software Inc., 2003).

recursos para trabalho cooperativo e, além disso, não possui uma rigidez na construção dos mapas, permitindo que o autor inclua elementos que não são pertinentes à ideia original de Mapas Conceituais apresentada por Novak (1990).

2.6.2 Modelo HMBS

O HMBS (*Hyperdocument Model Based on Statecharts*) (Turine, 1998), utilizado como base para a construção dos modelos conceitual, instrucional e didático da abordagem *AIM-CTD*, consiste em um modelo para a especificação da estrutura da semântica navegacional de hiperdocumentos, usando a técnica de *Statecharts* (Harel, 1987) como modelo de especificação formal subjacente.

Turine (1998) define formalmente um hiperdocumento H como uma 7-tupla $H = \langle ST, P, m, L, pl, ae, N \rangle$, onde:

- ST representa o *Statechart* subjacente ao hiperdocumento definido no HMBS como uma 11-tupla $ST = \langle S, \rho, \psi, \gamma, \delta, V, C, E, A, R, T \rangle$ representando, respectivamente, estado, 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;

- P é o conjunto finito de páginas de informação que define o conteúdo do hiperdocumento. Cada página $p \in P$ é definida conceitualmente pela tripla $\langle c, t, Anc_p \rangle$, tal que c é a porção de informação que pode ser composta por mídias estáticas ou dinâmicas; t representa o título associado à página e Anc_p define uma coleção de âncoras contidas na 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 a apresentação da informação;
- $m: Ss \rightarrow P$ é uma função que mapeia estados de um subconjunto $Ss (Ss \subset S)$ em páginas $p \in P$ do hiperdocumento. O conjunto Ss é formado pelos estados compostos do tipo *OR* e pelos estados atômicos do *Statechart*. Estados do tipo *AND* não são mapeados em páginas, sendo utilizados unicamente para especificar concorrência de informações na apresentação;
- L define o conjunto de canais de apresentação invocados para interpretar e visualizar as informações contidas nas páginas durante a navegação;
- $pl: P \rightarrow L$ define a função de visualização que associa cada página $p \in P$ a um único canal $l \in L$ capaz de interpretá-la;
- $ae: Anc_p \rightarrow E$ é a função que associa âncoras de uma página a eventos do *Statechart* que, por sua vez, definem as transições a serem disparadas;
- N ($N \geq 0$) define o nível de visibilidade do hiperdocumento. O autor pode utilizar o valor de N para definir a profundidade da hierarquia a ser apresentada durante a navegação.

Semântica de Navegação

A semântica de navegação adotada no HMBS (Turine et al., 1997, 1998) fornece mecanismos que realçam a estrutura hierárquica típica de muitas aplicações definindo quais as páginas serão apresentadas durante a navegação, quais âncoras serão habilitadas e quais as transformações navegacionais irão ocorrer durante a interação com o usuário. O comportamento dinâmico do hiperdocumento é baseado na semântica operacional de *Statecharts* a partir de alguma configuração de estado.

Para visualizar as informações do hiperdocumento, os canais são invocados a fim de interpretar um número arbitrário de páginas durante a navegação, podendo ser exibidas diversas páginas concorrentemente. O número de páginas exibidas são determinadas de acordo com a configuração de estados corrente do *Statechart* subjacente. Assim, um hiperdocumento para o qual N é igual a zero ($N = 0$) serão exibidas em um determinado momento durante

a navegação todas as páginas associadas aos estados atômicos que estiverem ativos na configuração de estados do *Statechart*. Para N igual a 1 ($N = 1$) serão exibidas as mesmas páginas de $N = 0$ adicionando-se aquelas páginas associadas aos estados ancestrais do tipo *OR* imediatos da configuração de estados atual, e assim sucessivamente para $N > 1$.

Portanto, o conjunto de páginas apresentadas definem uma configuração de contexto do hiperdocumento. Essa configuração é formada pelas páginas associadas aos estados da configuração de estados corrente do *Statechart*, além das páginas que satisfazem o nível de visibilidade especificada pelo autor.

A configuração do contexto inicial é definida pela configuração do estado *default* do *Statechart*.

Baseado na semântica operacional de *Statecharts*, a semântica de navegação do HMBS faz com que o usuário selecione uma âncora, de modo que o sistema de hiperdocumento execute as seguintes ações (Turine et al., 1998):

- Gerar o evento relativo a âncora disparando a ligação correspondente à transição do *Statechart* a qual o evento está associado.
- Ativar todos os estados pertencentes ao conjunto destino da transição disparada gerando a próxima transição de estado e desabilitando a anterior.
- Definir a nova configuração de contexto do hiperdocumento. Essa nova configuração de contexto é formada pelas páginas associadas aos estados da nova configuração de estados, além das páginas que obedecem ao escopo estabelecido pelo nível de visibilidade. Em seguida, executa os canais da apresentação para cada página da configuração de contexto.

A fim de permitir a utilização do HMBS em um processo completo de construção de hiperdocumentos, apoiando tanto o projeto como o desenvolvimento estruturado das aplicações hipermídia, Carvalho (1998) definiu um método chamado HMBS/M (*Hypertext Model Based on Statecharts/Method*).

O método é composto por quatro fases (Modelagem Conceitual, Modelagem Navegacional, Modelagem da Interface e Implementação e Teste) na qual as melhorias introduzidas no modelo de uma certa fase devem ser especificadas nos modelos das fases anteriores possibilitando um desenvolvimento iterativo.

A fase de Modelagem Conceitual tem como objetivo analisar o domínio da aplicação e representá-lo em modelos que auxiliem na identificação de informações da aplicação. A próxima fase, a Modelagem Navegacional, é a etapa onde são especificados os contextos de navegação, o modelo navegacional de tipos e o modelo navegacional de instâncias. Os contextos de navegação definem formas para se navegar pela aplicação. O modelo navegacional

de tipos permite modelar os tipos de contextos definidos pelas estruturas de acesso. O modelo navegacional de instâncias modela as instâncias das classes e dos relacionamentos do modelo navegacional de tipos em termos de estados e transições. Nessa fase é que o HMBS é utilizado para representar a semântica de navegação da aplicação.

A fase de Modelagem da Interface é a etapa em que são definidas as características genéricas da interface, que não dependem do ambiente de programação. Por fim, a última fase é a de Implementação e Teste, na qual deseja-se mapear os modelos navegacionais e de interface em objetos concretos, cujo resultado é a aplicação hipermídia propriamente dita.

2.6.2.1 Ferramenta HySCharts

O *HySCharts* (*Hyperdocument System Based on StateCharts*) (Turine et al., 1998) é uma ferramenta gráfica que permite a criação, interpretação e execução de especificações formais de aplicações hipermídia segundo o Modelo HMBS. Ele foi desenvolvido como uma extensão da ferramenta *StatSim* (*Statechart Simulator*) (Masiero et al., 1991) e projetada para oferecer facilidades de prototipação e simulação interativa da estrutura navegacional subjacente do hiperdocumento.

A ferramenta possui uma arquitetura composta por três camadas, conforme ilustrado na Figura 2.13. A Camada de Aplicação está subdividida nos Módulos de Autoria e de Navegação, em adição a um ambiente de edição e simulação de *Statecharts*. A Camada de Estruturação é o núcleo do sistema no qual é definida a estrutura interna do hiperdocumento por intermédio dos objetos estruturais, navegacionais e de apresentação do HMBS. Todos esses objetos são armazenados em bases de dados gerenciadas por funções da Camada de Armazenamento.

Módulo de Autoria

Durante a atividade de autoria o autor especifica visualmente os objetos estruturais (objetos definidos pelo *Statechart*), os objetos navegacionais (páginas e ligações) e de apresentação (canais). Inicialmente, o autor especifica a estrutura organizacional da aplicação hipermídia (*Statechart* subjacente) utilizando um editor gráfico como mostra a Figura 2.14. Após especificar a estrutura, o autor utiliza um editor gráfico de objetos HMBS, que contém ferramentas para criar e gerenciar páginas, âncoras e canais. Tendo criado as páginas e seus objetos de apresentação canal, o autor associa-as aos estados. Em seguida, especificam-se os objetos navegacionais âncora que habilitam as ligações do hiperdocumento.

Módulo de Navegação

As páginas a serem apresentadas durante a navegação são definidas por meio da semântica de navegação adotada no HMBS, ou seja, que âncoras são habilitadas e quais as transformações navegacionais irão ocorrer durante a interação com o leitor. Para tornar disponível aos

Figura 2.13: Arquitetura da ferramenta *HySCharts* (Turine et al., 1999b).

leitores o conteúdo da aplicação hipermídia, as especificações em HMBS devem ser interpretadas e “executadas” no módulo de navegação segundo a semântica de navegação do modelo. O *HySCharts* dispõe de uma máquina de *Statecharts* associada ao *Browser* que recebe os eventos externos (ações do leitor) e os interpreta, modificando a apresentação (Turine et al., 1998).

De modo geral, a ferramenta *HySCharts* foi projetada e desenvolvida para validar a possibilidade prática de uso do HMBS num contexto real, oferecendo, além da navegação tradicional, um novo mecanismo de navegação hierárquica a fim de possibilitar ao leitor navegar pelas páginas associadas a estados acima ou abaixo do nível de hierarquia definido pelo autor (Turine et al., 1998).

2.6.2.2 Ferramenta WebScharts

A Ferramenta CASE *WebScharts* (Brito, 2003) foi desenvolvida para auxiliar o projeto e desenvolvimento estruturado de aplicações Web segundo o método HMBS/M (*Hypertext Model Based on Statecharts/Method*) (Carvalho, 1998; Carvalho et al., 1999). A ferramenta

Figura 2.14: EGS - Editor Gráfico de *Statecharts* (Turine et al., 1999a).

tem por objetivo apoiar todas as fases do método, desde a modelagem conceitual até a geração da aplicação final, além de fornecer facilidades de prototipação rápida. Sua arquitetura é composta por três camadas principais, como ilustra a Figura 2.15.

A camada de aplicação permite ao projetista criar a estrutura organizacional e navegacional da aplicação Web e, ao usuário, navegar pela aplicação. Esta camada está subdividida nos ambientes (1) de autoria; (2) de projeto navegacional; e (3) de publicação. No ambiente de autoria, o projetista (a partir do documento de especificação de requisitos da aplicação) desenvolve os modelos da fase de modelagem conceitual do HMBS/M utilizando os editores de classe e 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. Esse modelo é a primeira visão abstrata da estrutura navegacional da aplicação Web. A especificação das páginas Web e o relacionamento entre estas são definidos no modelo de composição e, dessa forma, a aplicação Web pode ser gerada automaticamente no ambiente de publicação.

A camada de estrutura é considerada o núcleo da ferramenta, pois compreende a estrutura interna subjacente à formalização da aplicação Web com base nos modelos de classes, de fatias, de contextos navegacionais, navegacional de tipos, de composição, de estilos e do modelo HMBS. Toda a representação desses modelos é armazenada em bases de dados

Figura 2.15: Arquitetura da ferramenta *WebCharts* (Brito, 2003).

gerenciadas por funções da camada de armazenamento, que contém as bases internas da ferramenta capaz de gerar o conteúdo XML, de aplicações Web (Brito, 2003).

Para gerar a aplicação final e verificar as consistências é utilizado o gerador de aplicação que, por meio das estruturas criadas anteriormente e da associação de templates XSLT aos documentos XML gera a aplicação.

Vale ressaltar que as ferramentas *CMapTools*, *Inspiration*, *HySCharts* e *WebCharts*, descritas para estruturar o domínio de conhecimento, foram investigadas para o desenvolvimento da ferramenta *ALM-Tool*, detalhada no Capítulo 4. A ferramenta utiliza a técnica de Mapas Conceituais e o modelo HMBS para a modelagem de conteúdos com base nos modelos que constituem a *ALM-CTD*.

2.7 Considerações Finais

Neste capítulo foi apresentada uma síntese das principais pesquisas que serviram como base para a realização deste trabalho. Inicialmente foi discutida a ideia de módulos educacionais e mecanismos de apoio à modelagem de conteúdos educacionais associados. Ênfase foi dada à abordagem integrada de modelagem *ALM-CTD*. Além disso, foi apresentada a ideia de conteúdos educacionais livre, explorando características necessárias à evolução contínua do material didático e à possibilidade de sua construção de forma distribuída.

Padrões que proporcionem reusabilidade, portabilidade, extensibilidade e interoperabilidade dos objetos de aprendizado (e dos próprios módulos e conteúdos educacionais associados) também foram considerados. Dentre os padrões investigados foi considerado o LOM por possibilitar à ferramenta *ALM-Tool* a geração de conteúdos educacionais portáteis que possuam características segundo especificações de metadados com o objetivo de desenvolver conteúdos reutilizáveis e extensíveis.

Adicionalmente, foi apresentada uma síntese a respeito de alguns ambientes educacionais, destacando suas principais características e funcionalidades. Um aspecto importante a ser observado, é que apesar desses ambientes fornecerem suporte a várias atividades, incluindo colaboração, avaliação e acompanhamento do aprendiz, nenhum apoio automatizado à modelagem e geração de conteúdos educacionais é oferecido. Dentro desse contexto também foram discutidas ferramentas específicas para a construção de Mapas Conceituais e do modelo HMBS, as quais foram investigadas no desenvolvimento da ferramenta *ALM-Tool*.

No próximo capítulo são discutidas adequações e extensões propostas aos modelos estabelecidos na abordagem *ALM-CID* de modo a contribuir para a sua evolução, sobretudo a fim de possibilitar sua aplicação de forma compartilhada.

Evoluções à Abordagem AIM-CID

3.1 Considerações Iniciais

Conforme discutido anteriormente, a abordagem *AIM-CID* visa a reunir, em uma única proposta, diferentes perspectivas de modelagem, caracterizando, assim, um conjunto de modelos para a representação de conteúdos educacionais (Barbosa, 2004). A abordagem foi proposta em 2004 e, no presente trabalho, foi adaptada e evoluída a fim de incorporar aspectos de desenvolvimento e aplicação distribuída na construção de módulos educacionais.

Em linhas gerais, as extensões propostas têm por objetivo evoluir a abordagem *AIM-CID* de modo que os conteúdos construídos possam ser mais dinâmicos e mais facilmente customizáveis e reutilizáveis, contribuindo assim para a melhoria do processo de ensino e aprendizado. Na Seção 3.2, é discutida a aplicação de ontologias para a modelagem de conteúdos educacionais, definindo sua caracterização no contexto da Web Semântica. Na Seção 3.3 e 3.4 são detalhadas as evoluções propostas para os modelos conceitual e instrucional da abordagem *AIM-CID*.

3.2 Ontologias para a Modelagem de Conteúdos

Segundo Gruber (1995), uma ontologia é uma especificação formal e explícita de uma conceitualização compartilhada. Uma *conceitualização* refere-se ao modelo abstrato de um fenô-

meno do mundo real; a palavra *formal* significa legível por máquina; *explícita* refere-se aos conceitos, propriedades e relações explicitamente definidas (Guarino, 1998).

As ontologias têm por objetivo possibilitar o reuso e a interoperabilidade semântica, de modo a promover o compartilhamento e a troca de informações. Além disso, propõem-se a fornecer uma compreensão comum e compartilhada do domínio de conhecimento de modo que esse domínio possa ser entendido por pessoas e computadores.

Nesse sentido, ontologias possuem um vocabulário específico, associados a um conjunto de suposições referentes aos significados dos termos deste vocabulário. De modo geral, esse conjunto de suposições tem o formato de uma teoria lógica de primeira ordem, em que os termos do vocabulário são predicados unários ou binários chamados, respectivamente, conceitos e relações. Uma ontologia, no seu modo mais simples, descreve uma hierarquia de conceitos relacionada por relações de classificação, enquanto que, nos casos mais sofisticados, axiomas são adicionados para expressar outras relações entre conceitos e restringir a interpretação pretendida (Guarino, 1998).

Marietto et al. (2002) apontam os principais benefícios no desenvolvimento de ontologias: (1) propicia uma melhor compreensão do domínio abordado; (2) possibilita o compartilhamento do conhecimento, considerando o compartilhamento de termos de um determinado domínio; (3) oferece suporte à interoperabilidade entre sistemas computacionais, considerando o relacionamento de diferentes paradigmas, métodos, linguagens, etc; (4) possibilita a troca de informações; e (5) auxilia no reuso do conhecimento.

Para a modelagem de conteúdos educacionais, o uso de ontologias pode ser aplicado para estruturar o domínio de conhecimento com pontos comuns de referência, utilizando os conceitos que foram definidos com base em um vocabulário padrão. Esse vocabulário, especificado formalmente por meio de termos descritos para sistematizar o domínio de conhecimento, permite a elaboração de todo o conteúdo que compõe o material didático interligados de forma coerente e ordenada.

A ontologia utilizada para a atividade de modelagem pode ser tratada no contexto da Web Semântica pois a Web Semântica define a estrutura para os dados, enquanto as ontologias descrevem a semântica desses dados. Desse modo, o conteúdo educacional pode ser definido de forma estruturada, com base em um senso comum a respeito dos termos que descrevem um determinado domínio, a fim de possibilitar sua melhor compreensão e facilitar o processo de ensino e aprendizado.

No contexto da Web Semântica, para fornecer suporte às informações disponíveis na Web e possibilitar o uso de ontologias, o W3C (*World Wide Web Consortium*) propôs uma arquitetura em camadas, ilustrada na Figura 3.1.

Figura 3.1: Camadas da Web semântica. Traduzida de Koivunen e Miller (2001).

De modo geral, a camada **Unicode** garante a interoperabilidade em relação à codificação de caracteres enquanto que o **URI** (*Uniform Resource Identifier*) representa um padrão para a identificação e unificação de recursos físicos.

A camada **XML + Namespace (NS) + XML-Esquema** fornece a interoperabilidade em relação à sintaxe de descrição de recursos da Web Semântica para possibilitar a criação de documentos com informações estruturadas contendo, o conteúdo e informações a respeito (metadados).

A camada **RDF + RDF-Esquema** torna possível a definição de vocabulários que podem ser atribuídos por **URIs** e também possibilitam estabelecer declarações sobre objetos com **URIs**. A **RDF** (*Resource Description Framework*) fornece um modelo de dados para expressar declarações simples sobre recursos. Além disso, a **RDF** fornece uma linguagem, denominada **RDF-Esquema**, para a definição de esquemas para os vocabulários utilizados nas declarações. A **RDF-Esquema** estende a especificação básica do **RDF** para permitir a definição de vocabulários. Assim, o **RDF-Esquema** é capaz de representar uma linguagem mínima para a representação de ontologias simples.

A camada de **Ontologia** tem por objetivo apoiar a evolução de vocabulários, modelando o domínio de conhecimento na forma de ontologias. A linguagem **RDF-Esquema** permite a construção de ontologias com expressividade e inferência limitadas, por meio de conjunto básico de elementos para a modelagem, sendo que poucos desses elementos podem ser utilizados para inferência. A **OWL** estende o vocabulário da **RDF-Esquema** para a inclusão de elementos com maior expressividade e inferência. Além disso, a linguagem **OWL** fornece três módulos: (1) *OWL Full* – fornece todas as primitivas da linguagem **OWL**, é totalmente compatível com o **RDF** tanto sintática como semanticamente, sendo poderosa em termos de expressividade e, por isso, indecidível; (2) *OWL DL* (*Description Logic*) – restringe a capacidade da *OWL Full* para que sua capacidade de inferência seja decidível; e (3) *OWL Lite* – restringe a *OWL DL* para um subconjunto de primitivas, ideal para a representação de simples taxonomia.

A camada **Lógica** fornece suporte à especificação de regras que atuam sobre recursos e instâncias, e as camadas **Prova** e **Confiança** têm por objetivo executar as regras especificadas

na camada de **Lógica** para que a camada de **Confiança** avalie se a prova está correta ou não.

Ressalta-se que a camada de **Ontologia**, apoiada pela linguagem OWL no seu módulo *OWL Lite*, foi considerada para compor o modelo conceitual, representando a entrada por ontologias, da abordagem *AIM-CID*. As extensões propostas são detalhadas a seguir.

3.3 Extensões ao Modelo Conceitual

Conforme discutido anteriormente, o modelo conceitual da abordagem *AIM-CID* consiste em uma descrição em alto nível do domínio do conhecimento. Sua construção envolve a definição dos conceitos relevantes para a compreensão do domínio, a especificação de como tais conceitos se relacionam e o estabelecimento de uma estrutura para sua representação (Barbosa, 2004).

Para a composição do modelo conceitual, Barbosa (2004) optou pela caracterização dos conceitos por meio de Mapas Conceituais (Novak, 1981). Mapas Conceituais têm sido amplamente utilizados no contexto de ensino e aprendizado por se tratar de uma técnica intuitiva, mesmo para usuários não familiarizados, devido à possibilidade de representação gráfica do domínio de conhecimento. Ferramentas de apoio à técnica, conforme apresentado na Seção 2.6, também estão disponíveis na literatura.

Devido ao caráter subjetivo da utilização de Mapas Conceituais, sua construção para um mesmo domínio de conhecimento, pode variar significativamente em função da perspectiva observada pelo autor. Neste trabalho o modelo conceitual proposto na abordagem *AIM-CID* foi adaptado à utilização da técnica de Mapas Conceituais, restrita a uma estrutura hierárquica, caracterizando assim uma topologia em árvore. Desse modo, os Mapas Conceituais definidos para o domínio de Teste de Software podem ser construídos de diversas formas para o mesmo domínio, ou seja, sua construção pode variar de acordo com as preferências e perfil do autor que está modelando o domínio de conhecimento.

Considerando a adaptação proposta, nas Figuras 3.2(a) e 3.2(b) são ilustrados os mapas definidos sob diferentes perspectivas para o domínio de Teste de Software. De modo geral, o Mapa Conceitual definido originalmente no trabalho de Barbosa (2004), foi alterado para caracterizar o modelo hierárquico de uma topologia em árvore (Borges e Barbosa, 2009a). Nesse contexto, alguns dos conceitos definidos como, por exemplo, na Figuras 3.2(a), **Critério de Teste**, **Hipótese do Programador Competente**, **Efeito do Acoplamento**, entre outros, foram migrados para o modelo instrucional, por considerar que tais conceitos podem ser representados como informações adicionais. Além disso, optou-se por adicionar novos conceitos ao domínio de conhecimento, conforme ilustrado na Figuras 3.2(b), de modo

que os conceitos **Técnica Funcional**, **Técnica Estrutural** e **Abordagens Alternativas**, fossem inseridos para compor um novo Mapa Conceitual.

Como extensão a abordagem, vislumbrou-se a possibilidade de utilizar como apoio à estruturação do domínio de conhecimento, conjuntos de termos estruturados de maneira hierárquica, os quais pudessem complementar os conceitos definidos no Mapa Conceitual. Desse modo, foi introduzida na etapa de modelagem conceitual a entrada de termos definidos em uma ontologia. Em linhas gerais, a caracterização de conceitos consiste na importação dos termos definidos na ontologia, por meio de um arquivo no formato OWL.

Conforme discutido anteriormente, a OWL foi desenvolvida no contexto da Web Semântica para representar um conjunto de conceitos, linguagens e tecnologias com um grande potencial para a criação de ferramentas de autoria capazes de facilitar a construção de objetos de aprendizado. Desse modo, a Web Semântica, representada pela OWL, possibilitou o registro de informações definidas em uma ontologia, por meio de linguagens processadas por computadores a fim de permitir a realização de inferências automáticas. A estrutura fornecida pela OWL permite descrever um conjunto de conceitos e seus relacionamentos para um determinado domínio de conhecimento. Desse modo, o nível das ontologias utilizado nessa etapa, refere-se a uma taxonomia que trata principalmente hierarquias de classificação e restrições simples, definido no módulo *OWL Lite*.

Ressalta-se que a entrada por ontologias foi escolhida, sobretudo, devido ao fato de uma ontologia descrever um vocabulário comum de um dado domínio de conhecimento, por meio da definição do significado de termos e suas relações. Esse vocabulário, disponível no formato OWL, ou seja, estruturado de maneira hierárquica, é capaz de representar integralmente as informações necessárias à modelagem conceitual. Nesse sentido, adoção de ontologias no nível conceitual abjetiva proporcionar um melhor entendimento e consequente compartilhamento e reuso dos conceitos e relacionamentos representados. Além disso, o desenvolvimento distribuído de conteúdos também pode ser facilitado pela utilização de um vocabulário comum a respeito de um determinado domínio de conhecimento.

Considerando a construção/reengenharia do módulo educacional para o domínio de Teste de Software, foi utilizada a OntoTest (Barbosa et al., 2008a) representando a entrada por ontologias como apoio à definição dos conceitos da etapa de modelagem conceitual.

Figura 3.2: (a) Mapa Conceitual para o Critério Análise de Mutantes definido por Barbosa (2004).
(b) Mapa Conceitual para o mesmo domínio de conhecimento, caracterizando uma topologia em árvore.

Em linhas gerais, a OntoTest é uma ontologia estabelecida com base na norma ISO/IEC 12207 para apoiar a obtenção, organização, reuso e compartilhamento do conhecimento explorando os diferentes aspectos envolvidos na atividade de teste. Tal ontologia foi dividida em dois níveis.

O nível principal foi denominado **Ontologia Principal de Teste de Software** (Figura 3.3). Nesse nível são definidos os principais conceitos e relações da OntoTest. Os subníveis **Processo de Teste**, **Procedimento de Teste**, **Passo de Teste**, **Artefato de Teste** e **Recurso de Teste**, tratam os conceitos específicos da ontologia principal com um maior nível de detalhes (Barbosa et al., 2008a). A partir dessa estrutura, é possível reutilizar a OntoTest por completo, somente com o nível principal, ou por algum de seus subníveis, descritos a seguir.

Figura 3.3: OntoTest: ontologia principal de Teste de Software.

- **Processo de Teste (*TestingProcess*):** são baseados em um paradigma de desenvolvimento, bem como no domínio da aplicação. Modelos de ciclo de vida de teste são usadas como referência na definição de um processo de teste, estabelecendo os principais passos de teste, e as relações de dependência entre eles.
- **Procedimento de Teste (*TestingProcedure*):** um exemplo de procedimento de teste são as técnicas de teste. As técnicas de teste são investigadas para apoiar a realização de testes de uma maneira sistemática de acordo com bases teóricas. De modo geral, as técnicas de teste selecionam sub-testes de um domínio de entrada preservando as possibilidades de falhas não cobertas. Exemplos de técnicas são: funcionais, estruturais, baseadas em erros, baseadas em máquinas de estados, randômica e *ad hoc*. Cada técnica primária estabelece um ou mais critérios de teste que podem ser caracterizados em critérios de adequação ou seleção.

- **Passo de Teste (*TestingStep*):** para que um teste seja executado, um conjunto de passos essenciais devem ser considerados sendo que cada etapa é composta por atividades de teste. Desse modo, as principais atividades de teste de software identificadas referem-se a: um testador deve ser capaz de planejar e construir casos de teste, executar testes e, analisar e avaliar os resultados obtidos. Além disso, passos de teste podem ser uma composição de atividade de teste.
- **Artefato de Teste (*TestingArtifact*):** um artefato de teste pode ser considerado um artefato de entrada ou de saída, dependendo caso seja utilizado (consumido ou produzido por um passo de teste). Cada passo de teste pode envolver diferentes artefatos como: documentos de teste, diagramas de teste, casos de teste, requisitos de teste, drivers e stubs.
- **Recursos de Teste (*TestingResource*):** conforme descrito nas sub-ontologias anteriores, um passo de teste pode ser visto como uma primitiva de transformação, em que as entradas e saídas dos passos de testes correspondem a um artefato de teste. No entanto, para a execução dos passos de teste são necessários outros elementos, denominados recursos de teste. Os recursos de teste podem ser recursos humanos (testador ou gerente de teste) e recursos de hardware e de software (ferramentas de teste ou sistemas de apoio).

A seguir, é apresentada a construção do modelo conceitual definido a partir da OntoTest. De acordo com a Figura 3.4, na sub-ontologia **Procedimento de Teste** são definidas as **Técnicas de Teste**. De modo geral, as **Técnicas de Teste** definem quais os tipos de informações serão selecionados para estabelecer os requisitos de teste. Exemplos de técnicas de testes são: **Técnica Funcional**, **Técnica Estrutural**, **Técnica Baseada em Erros**, **Técnica Baseada em Máquina de Estados**, **Técnica Randômica** e **Técnica ad hoc**. A **Técnica Baseada em Erros** estabelece a **Análise de Mutantes** como um dos **Critérios Baseado em Erros**. O critério utiliza informações sobre **Erros Típicos** para derivar casos de teste. **Erros Típicos** são modelados por meio de **Operadores de Mutação**, dando origem a programas **Mutantes**, podendo ser classificados em **Mutantes Mortos**, **Mutantes Equivalentes** e **Mutantes Error-Revealing**.

É importante destacar que o modelo conceitual descrito, a partir da sub-ontologia **Procedimento de Teste** da OntoTest foi utilizado na construção/reestruturação do módulo educacional para o domínio de Teste de Software. Sua construção, utilizando a ferramenta *AIM-Tool*, é apresentada no Capítulo 4.

Mais detalhes sobre conceitos, axiomas e restrições da OntoTest estão descritos em Barbosa et al. (2006, 2008a).

Figura 3.4: Modelo conceitual considerando a sub-ontologia de Procedimento de Teste da OntoTest.

3.4 Extensões ao Modelo Instrucional

O modelo instrucional da abordagem *AIM-CID* é responsável pela definição de informações adicionais relativas aos conceitos previamente identificados no modelo conceitual. Nessa etapa, são definidos e modelados itens de informação e elementos instrucionais (Barbosa, 2004; Barbosa e Maldonado, 2006b). Por itens de informação compreendem-se, além dos conceitos já identificados, todas as demais informações significativas ao domínio de conhecimento as quais também devem ser representadas e incorporadas ao conteúdo educacional.

Os elementos instrucionais, por sua vez, são informações adicionais que complementam os conceitos e demais itens de informação já definidos. Tais elementos são representados a fim de possibilitar uma melhor compreensão, além de facilitar a assimilação o conteúdo relativo ao domínio de conhecimento em questão. Como exemplo de elementos instrucionais tem-se: exercícios, dicas, exemplos, sugestões de estudo, ferramentas, simulações, entre outros (Barbosa e Maldonado, 2006b).

Como extensão ao modelo instrucional, neste trabalho foram definidos conjuntos para caracterizar as mídias que representam os itens de informação e elementos instrucionais. Tal caracterização teve como base a *ALOCoM-Ontology* (*Abstract Learning Object Content Model - Ontology*) (Gasevic et al., 2005) – uma ontologia de apoio à estruturação do conteúdo de um objeto de aprendizado, que possibilita o seu reuso, por meio da definição de suas partes (componentes).

A ALOCoM-Ontology é uma ontologia que provê uma definição explícita da estrutura do conteúdo de um objeto de aprendizado, especificando seus tipos de componentes e a forma com a qual tais objetos estão relacionados. Sua estrutura, ilustrada na Figura 3.5, separa os componentes em (Gasevic et al., 2005): (1) **Fragmentos de Conteúdo** (CF – *Content Fragments*); (2) **Objetos de Conteúdo** (CO – *Content Objects*); e (3) **Objetos de aprendizado** (LO – *Learning Objects*).

Figura 3.5: ALOCoM-Ontology (Gasevic et al., 2005).

Os **fragmentos de conteúdo** são os elementos do **objeto de aprendizado** na sua forma mais básica. Tais elementos são divididos em **contínuos** (*ContinuousCF*) e **descontínuos** (*DiscreteCF*). Elementos **contínuos** podem ser **animação**, **vídeo**, **áudio** e **simulação**; enquanto que elementos **descontínuos** podem ser divididos em **texto**, **gráfico**, **link** e **imagens**. Os **fragmentos de conteúdo** podem ser agregados em **objetos de conteúdo**. A definição de uma ontologia para **objetos de conteúdo** inclui a definição da estrutura do conteúdo, bem como as relações entre os seus componentes. Por fim, os **objetos de conteúdo** são agregados a fim de formar um **objeto de aprendizado**. Na Figura 3.6 é ilustrada a composição de um **objeto de aprendizado** pela ALOCoM-Ontology.

Figura 3.6: Composição do objeto de aprendizado (Gasevic et al., 2005).

A integração da ALOCoM-Ontology ao modelo instrucional da abordagem *AIM-CID* visa a melhor explorar o compartilhamento e reuso do conhecimento. A partir da adoção da ontologia, é possível agregar ao conteúdo diferentes objetos de aprendizado caracterizados em conjuntos de mídias que utilizam uma estruturação baseada componentes. Desse modo, para a definição das mídias adequadas à composição dos itens de informação e elementos instrucionais associados aos conceitos, neste trabalho foi considerada a ideia de **fragmentos de conteúdo** definida na ALOCoM-Ontology.

Na Figura 3.7 é ilustrada a integração dos itens de informação e elementos instrucionais com as mídias definidas na ALOCoM-Ontology. Observa-se que tanto itens de informação como elementos instrucionais podem ser associados a mídias contínuas e/ou descontínuas, dependendo das características e objetivos de aprendizado associados ao conteúdo educacional sendo desenvolvido.

Figura 3.7: Integração da ALOCoM-Ontology com itens de informações e elementos instrucionais do modelo instrucional da abordagem *AIM-CID*.

É importante destacar que a especificação de mídias no nível instrucional da abordagem *AIM-CID*, em especial das mídias contínuas, constitui um fator relevante no desenvolvimento de conteúdos educacionais, por definir conteúdos dinâmicos capazes de despertar ao aprendiz o interesse pelos tópicos abordados, transformando-os em agentes no processo de construção do conhecimento. Nesse contexto, a caracterização das mídias por meio da ALOCoM-Ontology permitiu à abordagem a diferenciação de objetos dinâmicos durante a definição de partes essenciais do conteúdo educacional e das informações complementares associadas a esse conteúdo. Além disso, a adoção da ALOCoM-Ontology possibilitou uma maior flexibilidade para a evolução contínua desses objetos de modo que a inserção de novos conteúdos dinâmicos possa ocorrer de forma natural dentro das categorias de mídias contínuas e descontínuas.

Adicionalmente à caracterização das mídias com base na ALOCoM-Ontology, fez-se necessário a definição de algumas diretrizes de modo a adaptar o modelo instrucional para proporcionar a caracterização dessas mídias, além de possibilitar sua definição de forma sistemática.

Segundo Barbosa (2004), para a representação do modelo instrucional da abordagem *AIM-CID*, considerou-se o modelo HMBS desvinculado da representação de transições, eventos e mecanismos de história, e estendido para a representação de diferentes categorias de conhecimento (itens de informação e elementos instrucionais). Esse modelo foi denominado *HMBS/Instrucional*.

De modo geral, as adaptações propostas foram baseadas no modelo original, seguindo um agrupamento de informações diferenciado. Nesse contexto, para a representação do modelo *HMBS/Instrucional* foi adicionado a cada conceito definido no modelo conceitual, um conjunto de informações que, associadas a esse conceito, são capazes de representar o domínio de conhecimento de maneira estruturada. Na Figura 3.8 é ilustrada a representação de um conceito genérico no modelo *HMBS/Instrucional*, definido nesse exemplo como **Conceito-Principal**. Desse modo, um conceito pode ser composto de:

1. Uma descrição textual para possibilitar a representação detalhada do conceito.
2. Um conjunto de mídias segundo a caracterização definida com base na ALOCoM-Ontology.
3. Um conjunto de itens de informação, sendo que cada item pode conter uma descrição textual e uma mídia associada.
4. Um conjunto de elementos instrucionais, sendo que cada elemento pode conter uma descrição textual e uma mídia associada.
5. Um conjunto de conceitos associados a ele, ou seja, de acordo com o modelo hierárquico definido no modelo conceitual representam os “conceitos filho”.

Figura 3.8: Estrutura definida para o modelo *HMBS/Instrucional* para caracterizar o conceito e as informações complementares.

Ressalta-se que, as adaptações propostas para a estruturação do modelo *HMBS/Instrucional*, foram fundamentais para possibilitar a execução desse modelo por meio de um ambiente automatizado. Nesse contexto, foi desenvolvida a ferramenta *AIM-Tool* que será detalhada no Capítulo 4.

Para representar o material didático a partir da estruturação definida, na Figura 3.9 é ilustrado o *HMBS/Instrucional* com ênfase no estado que representa o conceito **Operador de Mutação**. De modo geral, os estados do *HMBS/Instrucional* representam os conceitos (definidos no modelo conceitual), e os itens de informações e elementos instrucionais que foram definidos e associados a esses conceitos na etapa instrucional. Para cada estado definido, existe uma correspondência no conteúdo educacional final.

Mais detalhes sobre os mecanismos de apoio ao desenvolvimento de módulos educacionais definidos por Barbosa (2004) estão descritos na Seção 2.2 e sobre o modelo HMBS na Seção 2.6.

Figura 3.9: Modelo instrucional (*HMBS/Instrucional*) da abordagem *AIM-CID* para o domínio de Teste de Software.

3.5 Considerações Finais

Neste capítulo foram discutidos e investigados aspectos relacionados às adequações e evoluções propostas aos modelos que constituem a abordagem *AIM-CID*. De modo geral, foram detalhadas as extensões propostas aos modelos Conceitual e Instrucional da abordagem considerando cada novo requisito associado esses desses modelos.

Para caracterizar os aspectos de compartilhamento e reuso do conhecimento para a modelagem de conteúdos educacionais, foram investigados conceitos relacionados a ontologias e a Web Semântica para o modelo conceitual da abordagem *AIM-CID*. Além disso, para o modelo instrucional, foram discutidos o agrupamento de mídias proporcionado pela ALOCoM-Ontology e as diretrizes definidas para a sua construção. Vale ressaltar que a abordagem *AIM-CID*, considerando ainda as evoluções propostas, garante o desenvolvimento de conteúdos educacionais por meio de uma abordagem mista de desenvolvimento incremental e iterativo. Desse modo, em cada fase definida na abordagem, o conteúdo educacional pode ser construído ou enriquecido indefinidamente. Tal característica da abordagem torna relevante o desenvolvimento de um ambiente automatizado, capaz fornecer suporte à utilização de tais modelos. Nesse sentido, para apoiar a modelagem e geração automática de conteúdos

educacionais com base nos modelos que compõem a $\mathcal{AM}-\mathcal{CID}$, foi desenvolvida a ferramenta $\mathcal{AM}-Tool$, que será detalhada no próximo capítulo.

AIM-Tool: Apoio Automatizado à Abordagem AIM-CID

4.1 Considerações Iniciais

O desenvolvimento de módulos educacionais requer a determinação de conceitos e informações complementares, estruturados de forma coerente, a fim de garantir a produção de um material didático de qualidade. No entanto, os módulos educacionais desenvolvidos até então, como, por exemplo, “Teste de Software: Teoria e Prática” (Barbosa et al., 2008c) e “Teste de Mutação” (Barbosa e Maldonado, 2006a), foram produzidos a partir da aplicação manual da abordagem *AIM-CID*, sendo essa uma tarefa trabalhosa e propensa a erros. Desse modo, o projeto e desenvolvimento de uma ferramenta de apoio automatizado a esse processo é essencial para viabilizar a utilização da abordagem de forma sistemática, minimizando os possíveis erros decorrentes de sua aplicação manual, além de possibilitar a modelagem e construção compartilhada e distribuída de conteúdos educacionais.

Esse cenário serviu como motivação para o desenvolvimento da *AIM-Tool*, uma ferramenta para apoiar a edição, interpretação e execução dos modelos propostos na *AIM-CID*, além de fornecer apoio à geração automática de conteúdos educacionais em diferentes formatos.

Neste capítulo são apresentadas as características e funcionalidades da ferramenta *AIM-Tool*. Na Seção 4.2, é apresentada uma visão geral da ferramenta. Na Seção 4.3, são detalhados os principais aspectos de modelagem e, na Seção 4.4, são discutidos os aspectos arquiteturais

da *AIM-Tool*. Por fim, com a estrutura definida para a implementação da *AIM-Tool*, na Seção 4.5, são apresentados seus aspectos de implementação e, na Seção 4.6, seus aspectos operacionais.

4.2 Visão Geral de Ferramenta *AIM-Tool*

Conforme discutido anteriormente, para apoiar o desenvolvimento de um material didático de qualidade é necessário que conceitos e relacionamentos estejam bem definidos, tendo como base técnicas capazes de apoiar a estruturação do conhecimento. Para isso, considerou-se a modelagem de conteúdos proposta pela abordagem *AIM-CID*.

Como apoio automatizado à aplicação da abordagem *AIM-CID*, foi desenvolvida neste trabalho a *AIM-Tool* – uma ferramenta para a construção compartilhada e distribuída dos modelos conceitual, instrucional e didático da abordagem. Nesse sentido, a ferramenta *AIM-Tool* fornece recursos que permite especificar todas as informações necessárias à modelagem compartilhada de um domínio do conhecimento na qual os autores podem modelar e produzir conteúdos educacionais livres em conjunto com outros autores, contribuindo, assim, para a disseminação do conhecimento e considerando a construção e evolução do conteúdo modelado. Na Figura 4.1 são ilustradas as funcionalidades da ferramenta *AIM-Tool*.

Inicialmente, para a composição do modelo conceitual, a ferramenta possibilita tanto a associação individual de conceitos a cada domínio de conhecimento, ou ainda, de um conjunto de conceitos definidos em uma ontologia. Em seguida, para o modelo instrucional, a ferramenta permite a associação de itens de informação, elementos instrucionais e mídias, ao conceitos anteriormente definidos. Considerando o modelo didático, a *AIM-Tool* permite que o autor crie sequências distintas para a apresentação do conteúdo de acordo com as suas necessidades e preferências.

De modo geral, a ferramenta *AIM-Tool* possibilita ao autor a navegação irrestrita por todos os seus módulos, podendo alternar livremente sobre as etapas da modelagem estabelecidas na *AIM-CID*, de modo que o resultado de cada etapa sirva como apoio à execução das demais.

Adicionalmente, a ferramenta também fornece recursos para a geração automática de conteúdos educacionais em diferentes formatos, considerando ainda a especificação de tal conteúdos segundo o padrão LOM. Além disso, para todas as funcionalidades definidas estão disponíveis recursos de manutenção (edição, inserção e remoção) do conteúdo.

Figura 4.1: Funcionalidades da ferramenta *AIM-Tool* (Borges e Barbosa, 2009a,b).

4.3 Aspectos de Modelagem

A modelagem da ferramenta *AIM-Tool* foi realizada segundo o processo de desenvolvimento WAE (*Web Application Extension*), uma extensão para a UML criada por Conallen (1999). O processo de desenvolvimento WAE estende a notação UML, incorporando novos estereótipos com a adição de semântica e restrições que permitem a modelagem de elementos específicos da arquitetura envolvida em uma aplicação Web. Ressalta-se que tal processo foi escolhido a partir de estudos comparativos como os realizados por Gu et al. (2002) e Domingues et al. (2007), que selecionam o WAE como um dos principais métodos de desenvolvimento para aplicações Web.

Para a modelagem da ferramenta *AIM-Tool*, inicialmente foi realizada a especificação de requisitos, na qual foram identificados seus requisitos funcionais. A seguir, foram elaborados diagramas de caso de uso, diagrama de sequência, diagrama UX e diagrama de classes da para representar o sistema a partir de visões diferenciadas.

O diagrama de caso de uso foi utilizado com o objetivo de representar a interação entre os atores do sistema e as atividades que tais atores podem realizar. Na Figura 4.2, é ilustrado o diagrama de caso de uso elaborado para a ferramenta *AIM-Tool*, no qual o ator principal é o autor do material didático que será modelado, atuando sobre os módulos da ferramenta: gerência de domínio, modelo conceitual, modelo instrucional, modelo didático e geração de conteúdo.

Figura 4.2: Diagrama de caso de uso da ferramenta *AIM-Tool*.

A partir dos casos de uso, foram elaborados os diagramas de atividades, os quais permitem que sejam visualizados os fluxos dos eventos realizados pelos autores dentro do sistema. Para compor o diagrama de atividades, ilustrado na Figura 4.3, foram definidos os estados que representam as atividades estabelecidas para o fluxo que representa a gerência de domínio de conhecimento, modelo conceitual e modelo instrucional da ferramenta. De modo geral, a navegação pelos módulos da *AIM-Tool* é iniciada quando o autor escolhe o domínio de conhecimento que deseja modelar. Em seguida, é possível ou navegar pelos estados definidos para a etapa de modelagem conceitual, introduzindo novos conceitos ao domínio de conhecimento, ou navegar pelos estados do modelo instrucional para adicionar itens de informação e elementos instrucionais aos conceitos previamente definidos.

O fluxo ilustrado na Figura 4.4 representa os estados definidos para o modelo didático. O fluxo é iniciado com a listagem dos conceitos definidos para o domínio de conhecimento que está sendo modelado para que o autor possa, em seguida, selecionar os conceitos desse domínio. A partir da seleção de conceitos, é gerada uma nova lista de conceitos com todos os itens de informação e elementos instrucionais definidos para cada conceito. Tais informações devem ser selecionadas pelo autor para que a ferramenta *AIM-Tool* promova a geração do *statechart* referente ao *HMBSD/Didático*, representando todo o conteúdo que deverá ser gerado pelo módulo de geração de conteúdo.

Figura 4.3: Diagrama de atividades para a gestão de domínio, modelo conceitual e modelo instrucional definidos para a ferramenta *AIM-Tool*.

Em continuidade, o autor deve fazer o *upload* do arquivo SCXML para iniciar a navegação pelos estados do módulo de geração de conteúdo da ferramenta *AIM-Tool*. Após carregar as informações presentes no arquivo, a ferramenta lista todo o conteúdo que será gerado e solicita ao autor que adicione novas informações para compor a capa do material didático (título do material, nome da instituição de ensino, logo da instituição, entre outras). Após inserir tais informações, o autor passa para um novo estado no qual é possível informar dados necessários à definição do metadado especificado segundo o padrão IEEE-1484 LOM, de modo que o material gerado possa ser catalogado em repositórios, facilitando sua busca, aquisição e evolução.

Figura 4.4: Diagrama de atividades para o modelo didático e módulo de geração de conteúdo da ferramenta *AIM-Tool*.

Para modelar o fluxo de informações, por meio das principais rotas de navegação entre as páginas Web que compõem a aplicação, foi elaborado o diagrama UX (*User eXperience*) (Connallen, 1999). O modelo UX é composto por definições de telas (elementos apresentados ao usuário que contém a infra-estrutura padrão da interface), roteiros e mapas de navegação, sendo a tela o principal elemento a ser modelado. O diagrama UX definido para a ferramenta *AIM-Tool*, ilustrado na Figura 4.5, possui a representação do elemento tela, representado pelo esteriótipo << screen >>. Entre os elementos tela, existe ainda o fluxo de navegação, que trata a transição de uma tela à outra. Além do esteriótipo << screen >>, existem também os esteriótipos << inputform >>, utilizado para representar telas que contêm a entrada de dados do usuário, e << screencompartment >>, para representar porções reutilizáveis de tela.

Por fim, foi elaborado o diagrama de classes da ferramenta *AIM-Tool*, ilustrado na Figura 4.6. Neste diagrama são representadas todas as classes presentes no sistema e a forma com a qual estão relacionadas. A classe **author** representa o ator principal do sistema, que deve efetuar todas as ações de modelagem e geração automática dos conteúdos educacionais.

Essa classe possui relacionamento com as classes `EducationalModule`, `Concept`, `InformationItem`, `InstructionalElement`. Tal relacionamento deve-se ao caráter de desenvolvimento do domínio do conhecimento de forma compartilhada da ferramenta *AIM-Tool*, na qual é possível que o autor navegue pelo domínio de conhecimento para agregar conteúdos de sua autoria ou ainda, agregar ao seu material conteúdos de outros autores, construídos de forma compartilhada.

As subclasses `iiFact`, `iiConcept`, `iiPrinciple`, `iiProcedure` compõem a superclasse `InformationItem`. Os itens de informação representam os elementos significativos do domínio de conhecimento. Tais elementos foram compostos dessa forma, pois considerou-se a teoria proposta por Merrill (1983) na especificação dos itens de informação.

As subclasses `ieExercises`, `ieTool`, `ieComplementarInfo`, `ieExample` compõem a superclasse `InstructionalElement`. Os elementos instrucionais da *AIM-CID* representam as informações complementares aos conceitos. Segundo a abordagem *AIM-CID*, eles podem ser definidos como: exemplos, exercícios, dicas, sugestões de estudo, ferramentas, simulações, entre outros (Barbosa e Maldonado, 2006b).

As subclasses `ccfAnimation`, `ccfVideo`, `ccfAudio`, `ccfSimulation` compõem a superclasse `ContinuousCF` e as subclasses `dcfText`, `dcfGraphic`, `dctLink`, `dcfImage` compõem a superclasse `DiscreteCF`. Tais classes descrevem a estruturação das mídias definidas segundo a ALOCoM-Ontology, discutidas no Capítulo 3.

Figura 4.5: Diagrama UX definido no projeto da ferramenta *AIM-Tool*.

Figura 4.6: Diagrama de classes definido no projeto da ferramenta *AIM-Tool*.

4.4 Aspectos Arquiteturais

A arquitetura da *AIM-Tool* foi definida tendo como base o padrão arquitetural MVC (*Model-View-Controller*) (Buschmann et al., 1996), o qual consiste em uma tríade de camadas frequentemente usadas em sistemas interativos. Esse modelo vem sendo utilizado como padrão arquitetural para aplicações Web por promover a separação das funcionalidades nas seguintes camadas: (1) lógica de apresentação (camada do cliente); (2) lógica de negócio (camada de aplicação); e (3) lógica de acesso a dados (camada de persistência). Essa separação torna o sistema mais flexível, de modo que partes podem ser alteradas desde que o mecanismo de comunicação entre elas permaneça o mesmo. Com base no padrão MVC, a arquitetura da ferramenta *AIM-Tool*, ilustrada na Figura 4.7, foi estruturada em três camadas descritas a seguir.

Figura 4.7: Arquitetura da ferramenta *AIM-Tool* (Borges e Barbosa, 2009a).

A primeira camada da arquitetura definida para ferramenta *AIM-Tool* é a **lógica de apresentação**, representando a interface Web da ferramenta. Essa camada utiliza o *Framework* Ajax ZK (Open Source Ajax, 2007) para receber as requisições do autor (cliente) e enviar tais informações para a segunda camada, a da **lógica de negócio**. A interface da

ferramenta fornece dois métodos para a entrada de conceitos, por meio de um arquivo OWL, representando a entrada por ontologias ou pela inserção manual de conceitos.

Além das entradas fornecidas, a primeira camada apresenta ainda as saídas geradas após o processamento da **lógica de negócio**. As saídas geradas pela ferramenta podem ser desde a visualização das informações processadas até a disponibilização de arquivos, como: (1) SCXML resultante do modelo didático; (2) arquivos que representam o conteúdo educacional gerado, nos formatos PDF ou HTML, definidos no módulo gerador de conteúdo; e (3) o arquivo especificado segundo o padrão LOM, definido pelo módulo de geração de conteúdo com os dados referentes ao material modelado.

A segunda camada, a **lógica de negócio**, foi desenvolvida para processar informações com base em três sub-camadas.

- **Gerência de domínio de conhecimento:** sub-camada desenvolvida para processar informações a respeito do domínio de conhecimento que se deseja modelar, além de processar dados sobre a autoria do conteúdo inserido. informações sobre a autoria foram consideradas devido ao caráter distribuído da ferramenta *AIM-Tool*. Desse modo, é possível que o autor, ao navegar e construir o próprio conteúdo, possa identificar conteúdos modelados por outros autores, agregando-os ao material em construção.
- **Gerência da AIM-CID:** sub-camada que processa todo o conteúdo em elaboração tendo como base os modelos estendidos que constituem a abordagem *AIM-CID*, descritos no Capítulo 3. Assim, essa camada é subdividida em três módulos responsáveis pela representação (gerenciamento) dos modelos conceitual, instrucional e didático da abordagem.

O módulo que representa o modelo conceitual recebe, por meio da interface da aplicação (camada **lógica de apresentação**), um conjunto de conceitos contidos no arquivo OWL, ou definidos e inseridos pelo autor de forma manual. Para o arquivo OWL a extração dos dados é feita utilizando a OWL-API (*Web Ontology Language - Application Program Interface*) (University of Manchester, 2008) – uma interface Java para implementar aplicações que instanciam ontologias na Web, conforme a Web Semântica definida pela W3C.

Após a definição dos conceitos, a ferramenta *AIM-Tool* produz o mapa conceitual, possibilitando ao autor visualizar graficamente tais conceitos, e a forma com a qual estão relacionados. De modo geral, os conceitos processados pela OWL-API ou inseridos manualmente são convertidos em um modelo XML, que servirá de entrada para a API-Prefuse (Prefuse, 2008) – um *framework* Java utilizado para criar visualizações de dados de forma interativa. Esse *framework* provê a visualização dos conceitos por meio de uma estrutura hierárquica caracterizando uma topologia em árvore.

O módulo que representa o modelo instrucional dessa camada, gerencia os itens de informação associados aos conceitos modelados na etapa conceitual e os elementos instrucionais associados aos conceitos e aos itens de informação já modelados. Além disso, esse módulo gerencia as mídias definidas segundo a ALOCoM-Ontology que podem ser associadas diretamente aos conceitos definidos no módulo que gerencia o modelo conceitual, e também aos itens de informação e elementos instrucionais definidos neste módulo.

O módulo que representa o modelo didático gerencia a forma com a qual o conteúdo deve ser apresentado, ou seja, permite ao autor personalizar o material de acordo com os interesses pedagógicos, perfil do aprendiz, duração do curso, público-alvo, entre outros. Essa camada fornece informações necessárias a SCXML-API que irá traduzir tais informações em um modelo de SCXML (*State Chart extensible Markup Language*) (W3C, 2008) definido pela ferramenta *AIM-Tool* para representar o modelo *HMBB/Didático* da abordagem *AIM-CID*.

- **Gerador de conteúdo:** sub-camada necessária ao tratamento das informações produzidas no modelo didático por meio de um arquivo no formato SCXML. O arquivo SCXML representa o modelo *HMBB/Didático* da abordagem *AIM-CID*. Esse arquivo representa todo o conteúdo modelado que deve ser gerado pela sub-camada de geração de conteúdo. De modo geral, a SCXML-API extrai todo o conteúdo do SCXML, e processa tais informações pelo módulo de gerência de conteúdos. A gerência de conteúdo recupera as informações contidas nesse arquivo e busca o complemento dessas informações no banco de dados para gerar o material didático correspondente, tanto no formato PDF como HTML. Ressalta-se ainda que, juntamente ao material, é definido e disponibilizado ao autor um arquivo especificado segundo o padrão LOM (*Learning Object Metadata*). Esse padrão é gerado pelo processamento da LOM-API e transformado em um modelo XML padronizado segundo as especificações LOM.

A terceira camada, a **lógica de acesso aos dados**, é responsável pela persistência dos dados utilizando o Hibernate (Red Hat, 2008) – um *framework* para o mapeamento objeto-relacional (ORM - *Object-Relational Mapping*) para ambientes Java. Desse modo, todas as interações com o banco de dados são feitas por meio do Hibernate.

Vale ressaltar que a arquitetura da *AIM-Tool* foi elaborada de forma a desacoplar, na camada **lógica de negócio**, a etapa de modelagem que considera os modelos conceitual, instrucional e didático da *AIM-CID*, com o módulo de geração de conteúdos. Tal abordagem foi adotada para garantir a integração futura de novos módulos que possam ser agregados à ferramenta.

4.5 Aspectos de Implementação

Para a implementação da *AIM-Tool* foi utilizada a linguagem Java orientado a objetos, por meio da IDE Eclipse (Eclipse Foundation, 2004), seguindo diretrizes do modelo JavaBeans. O Sistema de Gerência de Banco de Dados (SGBD) adotado foi o MySQL (Microsystems, 1995). No entanto, é importante observar que a aplicação pode ser executada em qualquer outra base de dados relacional, devido à utilização do *Framework* Hibernate (Red Hat, 2008), para o mapeamento objeto relacional.

Considerando que a *AIM-Tool* foi desenvolvida como uma aplicação Web, faz-se necessária a utilização de um servidor de aplicações. Nesse caso, o servidor de aplicações adotado foi o Tomcat (Apache Software Foundation, 1999) – Web *container* de código aberto.

Vale ressaltar que todas as tecnologias agregadas a este trabalho estão dentro do contexto de software livre. Além das tecnologias de implantação já descritas, foram utilizados alguns *frameworks* e interfaces (API - *Application Program Interface*) para simplificar o desenvolvimento da ferramenta *AIM-Tool*.

- **Framework ZK (Open Source Ajax, 2007):** adotado como mecanismo para a geração de páginas Web, de modo a desacoplar a camada do cliente da lógica de negócios, reforçando assim a separação de interesses proposta no padrão MVC.
- **OWL-API (Web Ontology Language - Application Program Interface):** utilizada para traduzir o arquivo OWL para código Java, de modo que os conceitos e seus relacionamentos, definidos na ontologia, possam ser utilizados na etapa de modelagem conceitual.
- **Prefuse-API:** consiste em uma interface Java que foi utilizada para a visualização gráfica do modelo conceitual. A ferramenta *AIM-Tool* recupera do banco de dados os conceitos inseridos manualmente ou por meio de ontologias, constrói um modelo de XML necessário à interpretação do Prefuse, que irá criar a visualização gráfica dos conceitos e seus relacionamentos.
- **SCXML-API (State Chart extensible Markup Language - Application Program Interface):** utilizado após a recuperação das informações inseridas na base de dados, referente aos modelos conceitual e instrucional, para traduzi-las em um *State-chart* que serve como interface entre o modelo didático (*HMBSD/Didático*) e o módulo de geração de conteúdos.

Para a criação de apresentações dinâmicas foram desenvolvidos na ferramenta *AIM-Tool* contextos navegacionais. Desse modo, durante toda a navegação pelo material didático

gerado, é possível que o autor do material utilize diferentes caminhos durante a apresentação, podendo navegar de forma não linear entre os tópicos modelados.

Para promover a geração automática de conteúdos, considerou-se o formato PDF-Latex e HTML. A ferramenta *AIM-Tool* possui classes em código Java para traduzir dados inseridos no banco de dados e gerar arquivos Latex em conformidade com o pacote *beamer* – uma classe Latex para a composição de apresentações. A grande vantagem do Latex em relação aos editores convencionais, como o *Microsoft Power Point* e o *Impress*, está no fato do autor do material concentrar seus esforços apenas no conteúdo. Imagens, código-fonte e outros tipos de documentos externos são incluídos no documento no formato da própria mídia, ou em forma de *link*, deixando o processo de criação mais dinâmico. Outra vantagem está relacionada ao fato de que o arquivo da apresentação é gerado no formato PDF – um formato de documento que pode ser aberto em qualquer sistema operacional que tenha um leitor de PDF.

Nesse contexto, quando o formato do material a ser gerado é PDF-Latex, a ferramenta *AIM-Tool* retorna ao autor um pacote compactado contendo todas as mídias associadas aos conceitos, itens de informação e elementos instrucionais (selecionados pelo autor durante a modelagem do conteúdo). Também é retornado um arquivo TEX e um arquivo PDF, gerados após a compilação dos demais arquivos. Vale ressaltar que a principal vantagem do autor em possuir o arquivo TEX é a possibilidade de edição manual do material, ou seja, caso o autor deseje agregar novo conteúdo, ou mudar as dimensões de imagens, pode-se alterar esse arquivo e recompilá-lo utilizando algum editor Latex.

O formato HTML também possui características de navegação. Porém, a principal diferença em relação ao formato PDF-Latex é relacionada à profundidade da árvore de conceitos definida, ou seja, o esquema hierárquico do modelo. Nesse caso, o HTML consegue criar a hierarquia completa, de modo que o usuário possa utilizar a barra de rolagem do navegador para acessar o conteúdo, enquanto que no formato PDF-Latex é possível somente manter três níveis da hierarquia. O pacote retornado pela ferramenta *AIM-Tool* para o formato HTML contém os arquivos HTML e as mídias selecionadas, assim como no formato PDF-Latex.

Ressalta-se que tanto na geração de apresentações em PDF-Latex, como em páginas HTML, a *AIM-Tool* disponibiliza um arquivo segundo o padrão LOM. O LOM foi considerado principalmente por facilitar a busca, evolução, aquisição e uso do material, considerando um metadado estabelecido para representar as principais informações do material gerado.

Durante o desenvolvimento da *AIM-Tool*, foram realizados testes funcionais manuais, sendo feita a verificação e a validação das funcionalidades. Foram elaborados testes unitários apenas para classes com um maior grau de importância em termos de negócio. Para tal foi utilizado o *plugin* JUnit do Eclipse.

Na Tabela 4.1, são ilustradas algumas informações referentes à implementação da ferramenta *AIM-Tool*. Para a contagem do número de linhas de código (LOC - *Lines of Code*) não considerou-se comentários, linhas em branco e contagem de linhas referentes a classes de teste.

Atributos	Total
Número de Páginas Web	26
Número de Classes	73
Número de Atributos	141
Número de Métodos	467
Número de Linhas de Código	9288

Tabela 4.1: Informações de implementação da ferramenta *AIM-Tool*.

4.6 Aspectos Operacionais

A seguir são ilustrados os principais aspectos operacionais da ferramenta *AIM-Tool* considerando o desenvolvimento de um módulo educacional para o domínio de Teste de Software. Para compor tal módulo foi utilizado o módulo educacional desenvolvido manualmente no trabalho de Barbosa (2004), e como apoio à definição dos conceitos a OntoTest (Barbosa et al., 2006, 2008a).

A *AIM-Tool* foi desenvolvida para ser uma ferramenta para a modelagem compartilhada de conteúdos educacionais, na qual os autores podem se cadastrar e gerenciar as informações em conjunto com outros autores. Para seguir o fluxo completo da abordagem *AIM-CID*, a ferramenta possui um menu de navegação direta e indireta, utilizando *links* internos a páginas (Figura 4.8). O menu de navegação direta retrata o contexto dinâmico de navegação, ou seja, refere-se a um menu sempre presente, que possui ligações para todas as partes da ferramenta. Deste modo, o autor pode navegar pela ferramenta de forma não linear. O menu de navegação indireta, sugere alguns atalhos propondo uma navegação linear entre os módulos, ou seja, cada etapa possui ligações que sugerem o próximo passo para dar continuidade à atividade em questão. Tal menu foi definido para simplificar a navegação de autores não familiarizados com a abordagem *AIM-CID*, por propor uma sequência direta de modelagem de conteúdos.

Ao iniciar a navegação, é apresentada a tela principal da *AIM-Tool*, a partir da qual é possível seguir para o cadastro de autoria ou para o cadastro de um novo domínio de conhecimento, conforme ilustrado na Figura 4.9. Ainda nessa tela, o autor pode selecionar o domínio de conhecimento que desejar e, em seguida, navegar por esse domínio.

Figura 4.8: Interface do modelo instrucional da ferramenta *AIM-Tool* ilustrando a composição do seu contexto navegacional.

Figura 4.9: Página principal da ferramenta *AIM-Tool* com caminhos para adição de novos autores e novos domínios de conhecimento.

Após iniciar a navegação por um domínio, segue a etapa de modelagem conceitual. Nessa etapa os conceitos podem ser inseridos manualmente (Figura 4.10) ou por meio de uma ontologia, representada por um arquivo no formato OWL (Figura 4.11). Conforme discutido anteriormente, para a validação preliminar da ferramenta *AIM-Tool* foi desenvolvido um

módulo educacional para o domínio de Teste de Software. O arquivo OWL utilizado nessa etapa refere-se à taxonomia de conceitos definida na OntoTest.

É importante observar que, além de inserir os conceitos e definir seus relacionamentos de forma hierárquica, é possível agregar mídias a esses conceitos, definidas com base na ALOCoM-Ontology (Gasevic et al., 2005), conforme discutido no Capítulo 3.

Figura 4.10: Interface da ferramenta *AIM-Tool* para a inserção manual de conceitos.

Figura 4.11: Interface da ferramenta *AIM-Tool* para a inserção de conceitos por meio de ontologias.

Para simplificar a visualização dos conceitos definidos para o domínio de conhecimento em questão, foi desenvolvido na *AIM-Tool* a visualização textual e gráfica de tais conceitos.

A visualização textual foi definida para simplificar o acesso a informações, de modo que o autor, ao inserir um novo conceito, possa consultar a hierarquia para auxiliar a definição desse conceito. Além disso, foram implementadas diferentes formas de visualização gráfica de tais conceitos de modo que o autor possa interagir com a ferramenta manipulando graficamente esses conceitos. Na Figura 4.12, é ilustrada a visualização textual e na Figura 4.13 a visualização gráfica dos conceitos de teste utilizando a API-Prefuse.

Figura 4.12: Interface da ferramenta *AIM-Tool* ilustrando a visualização textual ao adicionar novo conceito.

Após a identificação dos principais tópicos do domínio de conhecimento (modelo conceitual), é iniciada a construção do modelo instrucional. Nessa etapa são identificadas as informações relevantes a serem associadas aos conceitos.

Na Figura 4.14 é ilustrada a adição de informações referentes ao modelo instrucional. Inicialmente, é apresentado ao autor uma tela com a lista de todos os conceitos definidos no modelo conceitual para aquele domínio de conhecimento. Nessa lista, para cada conceito, aparecem sua definição, sua descrição, e a quantidade de mídias, itens de informação e elementos instrucionais já cadastrados para esse conceito. Após selecionar o conceito, é apresentada uma segunda tela contendo quais tipos de informações podem ser adicionadas ao conceito. Nesse caso, pode-se adicionar novas mídias ao conceito, novos itens de informações e novos elementos instrucionais. Vale ressaltar que todo método para inserção de dados na ferramenta *AIM-Tool* possui também a gerência desses dados, ou seja, a possibilidade de edição e remoção dessas informações.

Ainda no modelo instrucional, é possível definir os estados para compor o *statechart* referente ao *HMBIS/Instrucional*. A representação dos estados pertencentes a esse modelo utiliza notações de estados do tipo AND e do tipo OR. As composições do tipo AND são

Figura 4.13: Interface da ferramenta *AIM-Tool* ilustrando as diferentes formas de visualização gráfica de conceitos.

utilizadas para agrupar informações que devem ser apresentadas simultaneamente, enquanto que as composições do tipo OR são utilizadas para agrupar informações de um mesmo grupo. Nas Figuras 4.15 e 4.16 é possível observar como é tratada a definição de tipos de estado utilizando a ferramenta, e como tais estados ficam agrupados em um modelo *statechart* correspondente à configuração feita pelo autor na ferramenta *ALM-Tool*.

A configuração inicial de estados definida pela ferramenta considera como padrão estados do tipo OR, sendo possível alterar tal definição em qualquer etapa da modelagem. Ressalta-se que as definições de estados dessa etapa são retomadas pela ferramenta *ALM-Tool* para a definição do contexto navegacional do modelo didático e na geração do SCXML resultante desse modelo, que será descrito a seguir.

Seguindo os modelos que constituem a abordagem *ALM-CTD*, o próximo passo é definir o modelo didático, selecionando os conceitos e demais informações associadas a fim de iniciar a composição do material didático que será gerado.

Figura 4.14: Interfaces da ferramenta *AIM-Tool* ilustrando as funcionalidades do modelo instrucional.

No modelo didático, contextualizado para a ferramenta *AIM-Tool*, as informações são selecionadas pelo autor em seqüências de passos, ilustrados na Figura 4.17. Inicialmente o autor deve selecionar os conceitos que foram definidos no modelo conceitual. Em seguida a ferramenta *AIM-Tool* apresenta ao autor todas as informações relativas a cada conceito (mídias do conceito, itens de informação e elementos instrucionais) que podem ser incluídas ao material. É importante destacar que, nessa etapa, para cada conceito são apresentadas todas as informações modeladas tanto pelo autor que está navegando, como as informações

Figura 4.15: Interface da ferramenta *AIM-Tool* para a definição de estados do tipo OR do modelo *HMBS/Instrucional*.

Figura 4.16: Interface da ferramenta *AIM-Tool* para a definição de estados com tipo AND e OR do modelo *HMBS/Instrucional*.

definidas por outros autores que contribuiriam de forma compartilhada para a evolução deste domínio de conhecimento.

Quando o autor finalizar a seleção das informações adicionais aos conceitos, a ferramenta apresenta na tela um relatório do que foi selecionado, conforme ilustrado na Figura 4.18. Além disso, nessa mesma tela é possível que o autor exporte o arquivo SCXML gerado pela ferramenta *AIM-Tool*. Esse arquivo deverá ser utilizado na próxima etapa da ferramenta, a geração de conteúdos.

O SCXML é um padrão definido pela W3C (2008) para a composição de máquinas de estados genéricas baseada no modelo proposto inicialmente por Harel (1987). Considerando que o modelo HMBS e consequentemente os modelos *HMBS/Instrucional* e *HMBS/Didático*

Figura 4.17: Modelo didático definido para a ferramenta *AIM-Tool* considerando a seleção do conteúdo que será gerado.

da abordagem *AIM-CID* também utilizam o *Statechart* com base na mesma abordagem, o SCXML torna-se adequado à representação desses modelos.

De modo geral, o SCXML é iniciado pela diretiva `<scxml>`; seus estados são mapeados como `<state>`, em que cada estado deve apontar qual deverá ser a sua transição inicial (`<initial>`), que representa o próximo estado ou sub-estado que será ativado. Além disso é possível mapear, dentro de cada estado, outros estados (sub-estados) ou, ainda, estados paralelos (`<parallel>`) (W3C, 2008). Na Figura 4.19 é ilustrado um exemplo de SCXML e seu modelo *HMB/Didático* correspondente. O SCXML ilustrado foi gerado após a seleção de conceitos no modelo didático do módulo educacional desenvolvido para o domínio de Teste de Software.

Em continuidade, para produzir um material didático navegável, o módulo de geração de conteúdos considera inicialmente um *statechart*, por meio do SCXML definido anteriormente no modelo didático. O autor deve importar para a *AIM-Tool* esse arquivo que, ao ser

Figura 4.18: Interface da ferramenta *AIM-Tool* com o relatório gerado após a seleção, no modelo didático, do conteúdo.

Figura 4.19: Transição do *HMBS/Didático* para um SCXML.

reconhecido pela ferramenta solicita ao autor informações adicionais para compor a página (transparência) principal do material. Após processar todas essas informações, a ferramenta

apresenta uma tela disponibilizando um pacote com todo o material didático gerado, conforme ilustrado na Figura 4.20.

Figura 4.20: Geração de conteúdo pela ferramenta *AIM-Tool*.

Para gerar o material didático foram definidas algumas diretrizes para mapear todo o conteúdo modelado. Para isso o conteúdo educacional modelado foi adaptado para ser agrupado de acordo com o formato do material a ser gerado, HTML ou PDF. Para materiais didáticos no formato HTML não é necessário agrupar/restringir a quantidade de conteúdo, pois é possível utilizar a barra de rolagem para navegar por todo o conteúdo. No entanto, para gerar apresentações em PDF, o conteúdo é agrupado seguindo uma sequência que considera apenas até o terceiro nível da hierarquia.

Como estrutura do material didático modelado, foram utilizados índices com acesso direto aos conceitos modelados na etapa conceitual. A especificação do contexto didático do material é fundamental, pois fornece uma estrutura de acesso mais rica permitindo considerar diferentes estratégias de navegação, bem definidas, de modo que o leitor possa percorrer de diferentes formas o conteúdo de seu interesse e mover-se ao longo do conteúdo para frente ou para trás.

A definição dos elementos instrucionais associados aos conceitos foram dispostos no material didático de Teste de Software como conteúdo navegável, por meio de ligações internas conforme ilustrado na Figura 4.21.

Figura 4.21: Exemplo de material gerado pela ferramenta *AIM-Tool* no formato PDF-Latex.

É importante observar que a adoção de SCXML para a geração de conteúdos torna o mesmo interoperável, por manter um padrão bem definido para máquinas de estado. Um exemplo está na possibilidade de visualização gráfica dos modelos por meio do alphaWorks (alphaWorks, 2007) – uma ferramenta de modelagem, que oferece mapeamento para a UML com suporte ao padrão SCXML. Tal ferramenta possibilita a criação e edição, de forma gráfica de modelos baseados em SCXML.

Por fim, embora nesta seção tenha sido apresentado o funcionamento da *AIM-Tool* seguido um fluxo linear da abordagem *AIM-CID*, o autor pode navegar e construir o conteúdo do modo que desejar, pois os modelos são totalmente desacoplados e inter-relacionados. Ressalta-se que a ferramenta *AIM-Tool* foi utilizada para a construção/reengenharia do módulo educacional para o domínio de Teste de Software. Tal módulo foi desenvolvido como uma validação preliminar da ferramenta, evidenciando a aplicabilidade dos mecanismos propostos.

Experimentos sistemático vêm sendo planejados e serão tratados com maior detalhamento em trabalhos futuros.

4.7 Considerações Finais

Neste capítulo foram apresentadas as principais características da ferramenta *AIM-Tool* ilustrada por meio da construção de um módulo educacional desenvolvido para o domínio de Teste de Software. Inicialmente foi discutida a modelagem de seus requisitos realizada em conformidade com as etapas definidas pelo método WAE (Conallen, 1999). Após a modelagem foi definida a arquitetura segundo o padrão MVC por promover a separação das funcionalidades em três camadas, de modo a garantir sua flexibilidade. Aspectos operacionais e de implementação também foram considerados.

Para ilustrar a aplicação automatizada da *AIM-CID* considerando as evoluções propostas aos modelos conceitual e instrucional, e como forma de validação preliminar da ferramenta *AIM-Tool*, foi desenvolvido um módulo educacional para o domínio de Teste de Software. Esse módulo utiliza parte dos conceitos considerados no módulo educacional desenvolvido no trabalho de doutorado de Barbosa (2004), evoluído a partir de novas informações encontradas na literatura. Além disso, como apoio à estruturação do conteúdo, também foi considerada parte da *OntoTest* – uma ontologia de teste de software, utilizada como apoio à definição e estruturação dos conceitos referentes ao domínio de Teste de Software. Ressalta-se que uma validação formal da ferramenta faz-se necessário e, nesse sentido, experimentos sistemáticos e controlados vêm sendo planejados.

No próximo capítulo é apresentada uma visão geral da pesquisa realizada no escopo deste trabalho, sendo sintetizadas as principais contribuições e os trabalhos futuros a serem conduzidos em continuidade ao presente trabalho.

Conclusões

5.1 Visão Geral da Pesquisa Realizada

Nos últimos anos, trabalhos relacionados ao ensino e aprendizado vêm sendo bastante discutidos pela comunidade científica. Em especial, novas técnicas e tecnologias de ensino e aprendizado. O estabelecimento de ambientes e sistemas educacionais, com suporte à disponibilização de material didático e à condução de atividades colaborativas, por meio de um conjunto de ferramentas síncronas e assíncronas associadas, constituem mecanismos importantes, sendo investigados nesse contexto. No entanto, uma limitação comum à maioria desses ambientes está associada à falta de mecanismos capazes de estruturar o domínio de conhecimento e apoiar a elaboração de material didático, com base em princípios educacionais previamente definidos.

Considerando o desenvolvimento de módulos educacionais, a modelagem de seus conteúdos associados representa uma atividade essencial para a estruturação do conhecimento. Especificamente com relação à modelagem de conteúdos educacionais, Barbosa (2004) estabeleceu a abordagem *ALM-CTD*. Tal abordagem foi desenvolvida para promover a modelagem de conteúdos educacionais utilizando modelos genéricos e inter-relacionados para a caracterização de conteúdos.

Procurando abordar os aspectos anteriormente relacionados, um dos objetivos deste trabalho visa a proposição de extensões aos modelos conceitual e instrucional da abordagem *ALM-CTD*. No nível conceitual da abordagem foi investigada a adoção de ontologias como

apoio ao compartilhamento e reuso de conceitos. No nível instrucional, ênfase foi dada à categorização das mídias associadas aos conceitos, itens de informação e elementos instrucionais. Tal categorização foi feita com base no agrupamento definido pela ALOCoM-Ontology, visando a componentização das mídias para a construção de conteúdos dinâmicos.

Além das extensões propostas, também foram definidas diretrizes para possibilitar o estabelecimento de um ambiente automatizado capaz de apoiar a construção dos modelos da *AIM-CTD* de forma integral, sem qualquer tipo de perda de informação. Nesse contexto, tornou-se possível o projeto e desenvolvimento da ferramenta *AIM-Tool*.

Além da possibilidade de aplicação automatizada da abordagem *AIM-CTD*, a ferramenta *AIM-Tool* também fornece mecanismos para a geração automática de conteúdos livres (OER), considerando ainda a possibilidade de geração de conteúdos educacionais em diferentes formatos. A ferramenta permite que, ao gerar automaticamente o material didático, tal material possua ainda uma especificação segundo o padrão LOM. Tal recurso foi considerado para garantir o reuso do material didático gerado pela ferramenta.

Por fim, é importante ressaltar que tanto as extensões à *AIM-CTD* como a ferramenta proposta no escopo deste trabalho foram aplicadas na construção/reengenharia de um módulo educacional para o domínio de Teste de Software. Os resultados obtidos evidenciaram, de forma preliminar, a aplicabilidade dos mecanismos propostos. Um experimento sistemático, visando a validação das ideias exploradas neste trabalho, vem sendo planejada e deverá ser conduzida em curto prazo.

5.2 Contribuições de Pesquisa

Como principais contribuições de pesquisa, podem-se destacar:

- **Proposição de extensões e evolução da abordagem *AIM-CTD*** . Foram investigados os principais mecanismos de apoio definidos no trabalho de Barbosa (2004), com ênfase nos aspectos relacionados à modelagem de conteúdos educacionais a partir dos modelos da abordagem *AIM-CTD*. Considerando os estudos realizados, foram propostas extensões à abordagem, incorporando ontologias à etapa de modelagem conceitual e, agregando ao modelo instrucional o agrupamento de mídias por meio da ALOCoM-Ontology. Além disso, foram definidas diretrizes para os modelos da abordagem *AIM-CTD* de modo a possibilitar sua aplicação de forma sistemática e, com isso, possibilitar a definição de um ambiente automatizado. Adicionalmente, como forma de validação preliminar às extensões propostas, foi desenvolvido/reestruturado, a partir da utilização da *AIM-Tool*, um módulo educacional para o domínio de Teste de Software.

- **Projeto e desenvolvimento da ferramenta *AIM-Tool*** . Tal ferramenta viabiliza a aplicação da abordagem *AIM-CID* de forma automatizada. Além disso, a ferramenta oferece recursos para a geração automática dos conteúdos educacionais modelados em diferentes formatos. Na concepção da ferramenta *AIM-Tool* foram realizados estudos sobre ambientes e sistemas de apoio ao ensino e aprendizado, arquiteturas de software para tais ambientes e tecnologias de desenvolvimento (*frameworks*, APIs) e implantação (banco de dados, servidor de aplicação) de sistemas desenvolvidos para a Web.

5.3 Trabalhos Futuros

Em continuidade às pesquisas realizadas podem-se destacar os seguintes trabalhos futuros:

- **Realizar estudos experimentais para o material gerado pela ferramenta *AIM-Tool***. Durante a condução das pesquisas realizadas neste trabalho de mestrado, ao desenvolver a ferramenta *AIM-Tool* foi definido, como exemplo, um módulo educacional para o domínio de teste de software, que utilizou na caracterização dos conceitos parte da OntoTest, e outros conceitos definidos *ad hoc*. A validação da ferramenta *AIM-Tool* foi realizada de modo informal, com a construção/reestruturação de um módulo educacional para o domínio de Teste de Software. Nesse sentido, experimentos sistemáticos e controlados para aplicar e validar a ferramenta na construção de módulos educacionais em diferentes domínios de conhecimento, e com diversos autores, dispersos geograficamente, vêm sendo planejados. Além disso, a efetividade dos módulos desenvolvidos também deverá ser investigada.
- **Definir um simulador gráfico entre os modelos instrucional e didático da abordagem *AIM-CID***. Os *statecharts*, como formalismo, possuem sintaxe e semântica definidas formalmente, possibilitando que a especificação do problema modelado seja analisada e validada automaticamente. Como linguagem, são visuais, podendo ser definidos e compreendidos graficamente utilizando uma notação visual. Para a especificação da *AIM-Tool*, o formalismo foi considerado com a adoção do SCXML (*State Chart extensible Markup Language*). No entanto, a parte gráfica pode ser melhor explorada com a definição de um simulador gráfico entre os modelos instrucional (*HMBIS/Instrucional*) e didático (*HMBIS/Didático*), considerando como entrada a especificação formal já definida.
- **Realizar estudos para a integração da ferramenta *AIM-Tool* em ambientes de apoio ao ensino e aprendizado**. A ferramenta *AIM-Tool* foi desenvolvida segundo uma arquitetura bem definida própria para aplicações Web. Desse modo, pretende-se investigar a integração da ferramenta *AIM-Tool* com ambientes de apoio ao ensino e

aprendizado disponibilizados como código aberto, como por exemplo, o *Moodle*, *Sakai*, entre outros.

- **Investigar evolução de interatividade dos conteúdos educacionais modelados.** Atualmente os itens de informação e elementos instrucionais da abordagem *AIM-CTD* agregam ao conteúdo mídias como áudio, vídeo, texto, simulações, entre outros. Para evoluir a interatividade do material didático modelado, principalmente no contexto de ensino a distância, pretende-se explorar a adoção de novos recursos de visualização agregando objetos virtuais com realidade aumentada aos itens de informação e elementos instrucionais, de modo a combinar elementos visuais (dados criados pelo computador) com o mundo real.
- **Investigar o uso dos modelos que constituem a *AIM-CTD* no estabelecimento de ontologias.** A abordagem *AIM-CTD* foi evoluída para possibilitar a entrada de ontologias no modelo conceitual. Em contrapartida, pode-se também investigar formas para definir e evoluir ontologias a partir dos modelos que constituem a *AIM-CTD*.

Referências

- ADL – SCORM Advanced distributed learning initiative – scorm specifications. 2006.
Disponível em: <http://www.adlnet.org/> [08/02/2009]
- AGRAWAL, H.; DEMILLO, R. A.; HATHAWAY, R.; HSU, W.; HSU, W.; KRAUSER, E. W.; MARTIN, R. J.; MATHUR, A. P.; SPAFFORD, E. H. *Design of mutant operators for the C programming language*. Relatório Técnico SERC-TR41-P, Software Engineering Research Center, Purdue University, West Lafayette, IN, 1989.
- ALBRIGHT, P. Open educational resources - final forum report. Internet Discussion Forum: Open Educational Resources - Open Content for Higher Education, 2005.
Disponível em: http://www.unesco.org/iiep/virtualuniversity/media/forum/oer_forum_final_report.pdf [14/12/2009]
- ALPHAWORKS alphaWorks : IBM modeling and integration tools for State Chart XML. 2007.
Disponível em: <http://www.alphaworks.ibm.com/tech/scxml/> [25/10/2009]
- APACHE SOFTWARE FOUNDATION Apache Tomcat. 1999.
Disponível em: <http://tomcat.apache.org/> [13/12/2009]
- AUSUBEL, D. P.; NOVAK, J. D.; HANESIAN, H. *Educational Psychology: a Cognitive View*. 2 ed. New York: Hold, Rinehart & Winston, 1978.
- BARBOSA, E. F. *Uma contribuição ao processo de desenvolvimento e modelagem de módulos educacionais*. Tese de Doutorado, ICMC – Universidade de São Paulo, São Carlos/SP, Brasil, 2004.
- BARBOSA, E. F.; MALDONADO, J. C. Mecanismos de apoio à modelagem de conteúdos: uma contribuição ao processo de desenvolvimento de módulos educacionais. In: *SBIE'04*:

-
- XV Simpósio Brasileiro de Informática na Educação*, Manaus/AM, Brasil, CD-ROM., 2004.
- BARBOSA, E. F.; MALDONADO, J. C. Establishing a mutation testing educational module based on IMA-CID. In: *Mutation'06: 2nd Workshop on Mutation Analysis*, Raleigh/NC, USA, 2006a.
- BARBOSA, E. F.; MALDONADO, J. C. An integrated content modeling approach for educational modules. In: *ICE'06: 1st International Conference on Education for the 21st Century - WCC'06: 19th IFIP World Computer Congress*, Santiago, Chile, 2006b, p. 17–26.
- BARBOSA, E. F.; MALDONADO, J. C. Towards the establishment of a standard process for developing educational modules. In: *FIE'06: 36th Annual Frontiers in Education Conference*, San Diego, CA, EUA, CD-ROM., 2006c.
- BARBOSA, E. F.; MALDONADO, J. C. Specialization and instantiation aspects of a standard process for developing educational modules. *ISOLA'08: 3rd International Symposium on Leveraging Applications of Formal Methods, Verification and Validation*, p. 16, CD-ROM., 2008.
- BARBOSA, E. F.; MALDONADO, J. C. Collaborative development of educational modules: a need for lifelong learning. *E-Infrastructures and Technologies for Lifelong Learning: Next Generation Environments – G. D. Magoulas. (Org.)*, p. 37, 2010.
- BARBOSA, E. F.; MALDONADO, J. C.; RICARTE, I. L. M. Learning materials: Towards the establishment of guidelines for domain modeling. In: *ICTEM'02: Informatics Curricula, Teaching Methods and best practice Working Conference*, Florianópolis/SC, Brasil, 2002, p. 193–206.
- BARBOSA, E. F.; NAKAGAWA, E. Y.; MALDONADO, J. C. Towards the establishment of an ontology of software testing. *SEKE'06: 18th International Journal of Software Engineering and Knowledge Engineering*, p. 522–525, 2006.
- BARBOSA, E. F.; NAKAGAWA, E. Y.; RIEKSTIN, A. C.; MALDONADO, J. C. Ontology-based development of testing related tools. In: *SEKE'08: 20th International Conference on Software Engineering and Knowledge Engineering*, San Francisco, CA, 2008a, p. 697–702.
- BARBOSA, E. F.; SILVA, M. A. G.; CORTE, C. K. D.; MALDONADO, J. C. Integrated teaching of programming foundations and software testing. In: *FIE'08: 38th Annual Frontiers in Education Conference*, Saratoga Springs, NY, 2008b, p. 5–10.

- BARBOSA, E. F.; SOUZA, S.; MALDONADO, J. C. An experience on applying learning mechanisms for teaching inspection and software testing. In: *CSEET'08: 21st IEEE-CS Conference on Software Engineering Education and Training*, Charleston, South Carolina, 2008c, p. 189–196.
- BORGES, V. A.; BARBOSA, E. F. Mecanismos de apoio ao desenvolvimento de módulos educacionais. 2008.
Disponível em: <http://www.labes.icmc.usp.br/moduloeducacional/> [24/01/2010]
- BORGES, V. A.; BARBOSA, E. F. Uma contribuição à modelagem e geração automática de conteúdos educacionais. In: *WEI'09: XIV Workshop de Educação em Computação – CSBC'09: XXIX Congresso da Sociedade Brasileira de Computação*, Bento Gonçalves/RS - Brasil: Sociedade Brasileira de Computação, 2009a, p. 601–610.
- BORGES, V. A.; BARBOSA, E. F. Using ontologies for modeling educational content. In: *SWEL'09: International Workshop on Ontologies and Semantic Web for E-Learning – AIED'09: Artificial Intelligence in Education*, Brighton, England, 2009b, p. 71–75.
- BRITO, L. S. F. *Webscharts: Uma ferramenta de desenvolvimento de aplicações web baseada no HMBS/M*. Dissertação de Mestrado, DCT – Universidade Federal do Mato Grosso do Sul, Campo Grande/MS, Brasil, 2003.
- BUSCHMANN, F.; MEUNIER, R.; ROHNERT, H.; SOMMERLAD, P.; STAL, M. *Pattern-oriented software architecture: a system of patterns*. John Wiley & Sons, Inc. New York, NY, USA, 1996.
- CAÑAS, A. J.; HILL, G.; CARFF, R.; SURI, N.; LOTT, J.; ESKRIDGE, T.; GÓMEZ, G.; ARROYO, M.; CARVAJAL, R. CmapTools: A knowledge modeling and sharing environment. *CMC'04: 1st International Conference on Concept Mapping - Concept maps: Theory, methodology, technology*, v. 1, p. 125–133, 2004a.
- CAÑAS, A. J.; NOVAK, J. D.; GONZÁLEZ, F. M. Using concept maps as a research tool in science education research. *CMC'04: 1st International Conference on Concept Mapping*, 2004b.
- CALVO, R.; GHIGLIONE, E.; ELLIS, R. The OpenACS e-learning infrastructure. In: *AusWeb'03: 9th Australian World Wide Web Conference*, 2003, p. 175–183.
- CARVALHO, M. R. *HMBS/M – um método orientado a objetos para o projeto e o desenvolvimento de aplicações hipermídia*. Dissertação de Mestrado, ICMC – Universidade de São Paulo, São Carlos/SP, Brasil, 1998.

- CARVALHO, M. R.; OLIVEIRA, M. C. F.; MASIERO, P. C. HMBS/M – an object oriented method for hypermedia design. In: *SBMIDIA'99: V Simpósio Brasileiro de Multimídia e Sistemas Hiperídia*, Goiânia/GO, Brasil, 1999, p. 43–62.
- CONALLEN, J. *Building Web applications with UML*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.
- COUNTERMAN, C.; GOLDEN, G.; GOLLUB, R.; NORTON, M.; SEVERANCE, C.; SPEELMON, L. Technical Report Sakai Project. 2004.
Disponível em: <http://sakaiproject.org/> [13/12/2009]
- DICK, W.; CAREY, L.; CAREY, J. O. *The systematic design of instruction*. 5 ed. Longman, 2001.
- DOMINGUES, A.; BIANCHINI, S.; COSTA, M.; FERRARI, F.; MALDONADO, J. Web application development methods: A comparison. In: *Webmedia'07: 13th Brazilian Symposium on Multimedia and the Web*, Gramado/RS, Brasil, 2007, p. 1–4.
- DOUGIAMAS, M.; TAYLOR, P. C. Moodle: Using learning communities to create an open source course management system. *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, 2003.
Disponível em: <http://dougiamas.com/writing/edmedia2003/> [13/12/2009]
- EBERSPÄCHER, H. F.; VASCONCELOS, C. D.; JAMUR, J. H.; ELEUTÉRIO, M. A. M. Eureka: Um ambiente de aprendizagem cooperativa baseado na Web para educação à distância. In: *SBIE'99: X Simpósio Brasileiro de Informática na Educação*, Curitiba/PR, Brasil, 1999, p. 341–348.
- ECLIPSE FOUNDATION Eclipse. 2004.
Disponível em: <http://www.eclipse.org/> [13/12/2009]
- ELLIS, C. A.; GIBBS, S. J.; REIN, G. L. Groupware: Some issues and experiences. *Communications of the ACM*, v. 34, n. 1, p. 38–58, 1991.
- FARMER, J.; DOLPHIN, I. Sakai: elearning and more. *EUNIS 2005-Leadership and Strategy in a Cyber-Infrastructure World*, 2005.
- FREE SOFTWARE FOUNDATION GNU General Public License. 2007.
Disponível em: <http://www.gnu.org/copyleft/gpl.html> [08/01/2010]
- FUKS, H. Aprendizagem e trabalho cooperativo no ambiente AulaNet. *RBIE'00: Revista Brasileira de Informática na Educação*, v. 1, n. 6, p. 53–73, 2000.

- GASEVIC, D.; DUVAL, E.; JOVANOVIĆ, J.; VERBERT, K.; MEIRE, M. Towards a global component architecture for learning objects: A slide presentation framework. In: *ED-MEDIA'05: 17th World Conference on Educational Multimedia, Hypermedia and Telecommunications*, 2005, p. 1429–1436.
- GEROSA, M.; RAPOSO, A.; FUKS, H.; LUCENA, C. Uma arquitetura para o desenvolvimento de ferramentas colaborativas para o ambiente de aprendizagem AulaNet. *SBIE'04: XV Simpósio Brasileiro de Informática na Educação*, p. 168–177, 2004.
- GOLDBERG, M. W.; SALARI, S.; SWOBODA, P. World Wide Web - Course Tool: An environment for building WWW-based courses. *Computer Networks and ISDN Systems*, v. 28, n. 7–11, p. 1219 – 1231, 1996.
- GRUBER, T. R. Toward principles for the design of ontologies used for knowledge sharing. *IJHCS'95: International Journal of Human Computer Studies*, v. 43, n. 5-6, p. 907–928, 1995.
- GU, A.; HENDERSON-SELLERS, B.; LOWE, D. Web modelling languages: the gap between requirements and current exemplars. In: *AusWeb'02: 8th Australian World Wide Web Conference*, 2002, p. 362–375.
- GUARINO, N. Some ontological principles for designing upper level lexical resources. In: *LREC'98: 1st International Conference on Lexical Resources and Evaluation*, 1998, p. 527–534.
- HAMEL, C. J.; RYAN-JONES, D. Designing instruction with learning objects. *IJET'02: International Journal of Educational Technology*, v. 3, n. 1, 2002.
- HAREL, D. Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, v. 8, n. 3, p. 231–274, 1987.
- HODGINS, W.; ET AL. Draft standard for learning object metadata. *W3C Candidate Recommendation IEEE P*, v. 1484, p. 08855–1331, 2002.
- HYLÉN, J. Open educational resources: Opportunities and challenges. *OECD'06: Centre for Educational Research and Innovation*, p. 49–63, 2006.
- IMS GLOBAL LEARNING CONSORTIUM IMS Project. 2006.
Disponível em: <http://www.imsproject.org/>[12/11/2009]
- INSPIRATION SOFTWARE INC. Inspiration tool. 2003.
Disponível em: <http://www.inspiration.com/vlearning>[13/12/2009]

- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION *Information technology – software life cycle processes*. Padrão ISO/IEC 12207, ISO/IEC, 1995.
- IWT Intelligent Web Teacher. 2006.
Disponível em: <http://www.didatticaadistanza.com/default.asp> [15/07/2009]
- KIRKPATRICK, K. OpenCourseWare. *Searcher*, v. 14, n. 6, p. 53–58, 2006.
- KOIVUNEN, M.; MILLER, E. W3C semantic Web activity. *Semantic Web Kick-Off in Finland*, p. 27–44, 2001.
- KOPER, R.; OLIVIER, B.; ANDERSON, T. IMS Learning Design Information Model. *IMS Global Learning Consortium*, 2003.
- LEE, W. W.; OWENS, D. L. *Multimedia-based instructional design: Computer-based training, web-based training, distance broadcast training*. San Francisco, CA: Pfeiffer, 384 p., 2000.
- LEO, D. H.; PEREZ, J. I. A.; DIMITRIADIS, Y. A. Ims learning design support for the formalization of collaborative learning patterns. *Advanced Learning Technologies, 2004. Proceedings. IEEE International Conference on*, p. 350–354, 2004.
- LUCENA, C. J. P.; FUKS, H.; LAUFER, C.; RIBEIRO, M. B.; CHOREN, R.; SILVA, V. T.; ASSIS, R. L.; FERRAZ, F.; CARVALHO, G. R. Applying digital information technology to education: a web-based course. In: *ICECE'00 - International Conference on Engineering and Computer Education*, São Paulo/SP, Brasil, 2000.
- LUCENA, C. J. P.; FUKS, H.; MILIDIU, R.; LAUFER, C.; RIBEIRO, M. B.; NOYA, R. C.; SILVA, V. T.; FERRAZ, F.; CARVALHO, G. R.; DAFLON, L. AulaNet – ajudando os professores a fazerem seu dever de casa. In: *SEMISH'99: XXVI Seminário Integrado de Software e Hardware*, Rio de Janeiro/RJ, Brasil, 1999, p. 105–117.
- LUCENA, C. J. P.; FUKS, H.; MILIDIU, R.; MACEDO, L.; SANTOS, N.; LAUFER, C.; RIBEIRO, M. B.; FONTOURA, M. F.; NOYA, R. C.; CRESPO, S.; TORRES, V.; DAFLON, L.; LUKOWIECKI, L. AulaNet – an environment for the development and maintenance of courses on the Web. In: *ICEE'98: International Conference on Engineering in Education*, Rio de Janeiro/RJ, Brasil, 1998.
- MAIDANTCHIK, C. L. L. *Gerência de processos de software para equipes geograficamente dispersas*. Tese de Doutorado, COPPE – Universidade Federal do Rio de Janeiro, Rio de Janeiro/RJ, Brasil, 1999.

- MAIDANTCHIK, C. L. L.; ROCHA, A. R. Managing a worldwide software process. In: *WGSD'02: Workshop on Global Software Development – ICSE'02: International Conference on Software Engineering*, Orlando, FL, 2002.
- MARIETTO, M.; DAVID, N.; SICHMAN, J.; COELHO, H. Requirements analysis of multi-agent-based simulation platforms: state of the art and new prospects. In: *MABS'02: 3rd International Workshop on Multi-Agent Based Simulation*, Bologna, Itália, 2002, p. 183–201.
- MARTINS, R. Projeto complementa a educação formal de forma interativa. 2009.
Disponível em: <http://www.e-educador.com/index.php/mundo-high-tech-mainmenu-99/5325-tidia> [25/10/2009]
- MASIERO, P. C.; FORTES, R. P. M.; BATISTA, N. Editing and simulating the behavioral aspects of real-time systems. *SEMISH'91: 18th Brazilian Integrated Seminar on Hardware and Software*, p. 45–61, 1991.
- MCGUINNESS, D.; VAN HARMELEN, F.; ET AL. OWL web ontology language overview. *W3C recommendation*, v. 10, p. 1–13, 2004.
- MELO, R. N.; BARUQUE, C. B.; BARUQUE, L. B.; PEREIRA, L. A. Instructional design: desenvolvimento de módulos de e-learning baseados em objetos de aprendizado (learning objects). Curso oferecido pelo Departamento de Informática da PUC-RIO, 2003.
- MERRILL, M. D. Component display theory. In: *Instructional Design Theories and Models: An Overview of their Current States*, Hillsdale, NJ: Lawrence Erlbaum, 1983.
- MICROSYSTEMS, S. MySQL. 1995.
Disponível em: <http://www.mysql.com/> [13/12/2009]
- MOODLE, D. G. Moodle – a free, open source course management system for online learning. 2006.
Disponível em: <http://moodle.org/> [14/12/2009]
- MOODLEDOCS Moodle Docs – the documentation for moodle that everyone may contribute to. 2006.
Disponível em: <http://docs.moodle.org/> [14/12/2009]
- MOREIRA, M. A. Mapas conceituais e aprendizagem significativa. *Cadernos de Aplicação*, v. 11, n. 2, 1997.
- MOREIRA, M. A.; BUCHWEITZ, B. *Mapas conceituais: Instrumentos didáticos, de avaliação e de análise de currículo*. São Paulo: Editora Moraes Ltda., 1987.

- MUZIO, J.; HEINS, T.; MUNDELL, R. Experiences with reusable e-learning objects: From theory to practice. *The Internet and Higher Education*, v. 5, n. 1, p. 21–34, 2002.
- NOVAK, J. D. *Uma teoria de educação*. São Paulo: Editora Pioneira, 1981.
- NOVAK, J. D. Concept mapping: A useful tool for science education. *Journal of Research in Science Teaching*, v. 27, p. 937–949, 1990.
- NOVAK, J. D.; GOWIN, D. B. *Learning how to learn*. New York: Cambridge University Press, 1984.
- OKI Open Knowledge Initiative – Accelerated interoperability through simplified integration. 2008.
Disponível em: <http://www.okiproject.org/> [20/10/2009]
- OPEN SOURCE AJAX ZK – Open Source Ajax. 2007.
Disponível em: <http://www.zkoss.org/> [14/12/2009]
- PEREIRA, L. A. M.; PORTO, F. A. M.; MELO, R. N. Objetos de Aprendizado Reutilizáveis (RLOs): conceitos, padronização, uso e armazenamento. *Monografias em Ciencia da Computação, PUC-RioInf. MCC10*, 2003.
- PRATA, C. L.; NASCIMENTO, A. C. A. A. *Objetos de aprendizagem: uma proposta de recurso pedagógico*. Ministério da Educação, 2007.
- PREFUSE Prefuse API – information visualization toolkit. 2008.
Disponível em: <http://prefuse.org/> [25/02/2009]
- RED HAT Hibernate – Relational Persistence for Java and .NET. 2008.
Disponível em: <http://www.hibernate.org/> [13/12/2009]
- ROCHA, H. V. Guia do ambiente virtual TelEduc. Rio de Janeiro: Núcleo de Pesquisa e Projetos de Educação a Distância, 2003.
- SEELS, B.; GLASGOW, Z. *Making instructional design decisions*. 2 ed. Columbus: Prentice-Hall, Inc., 1998.
- TIDIA Tecnologia da informação no desenvolvimento da internet avançada, 2006.
Disponível em: <http://www.tidia.fapesp.br/portal> [25/10/2009]
- TIDIA Tecnologia da informação para o desenvolvimento da internet avançada, 2007.
Disponível em: <http://tidia-ae.incubadora.fapesp.br/portal> [08/11/2009]

- TURINE, M. A. S. *HMBS: Um modelo baseado em statecharts para a especificação formal de hipertextos*. Tese de Doutorado, IFSC – Universidade de São Paulo, São Carlos/SP, Brasil, 1998.
- TURINE, M. A. S.; OLIVEIRA, M. C. F.; MASIERO, P. C. Designing structured hypertext with HMBS. In: *Hypertext'97: VIII International ACM Hypertext Conference*, Southampton, UK, 1997, p. 241–256.
- TURINE, M. A. S.; OLIVEIRA, M. C. F.; MASIERO, P. C. HySCharts: Um ambiente de autoria e navegação baseado no modelo HMBS. In: *SBMIDIA '98: IV Simpósio Brasileiro de Multimídia e Sistemas Hipermídia*, Rio de Janeiro/RJ, Brasil, 1998, p. 27–38.
- TURINE, M. A. S.; OLIVEIRA, M. C. F.; MASIERO, P. C. HySCharts: A statechart-based environment for hyperdocument authoring and browsing. *Multimedia Tools and Applications*, v. 8, n. 3, p. 309–324, 1999a.
- TURINE, M. A. S.; OLIVEIRA, M. C. F.; MASIERO, P. C. HySCharts: um ambiente de autoria e navegação para a especificação formal de aplicações hipermídia. *SBES'99: Simpósio Brasileiro de Engenharia de Software*, 1999b.
- UNIVERSITY OF MANCHESTER OWL API. 2008.
Disponível em: <http://owlapi.sourceforge.net/> [25/11/2009]
- VIEIRA, C. E. M.; NICOLEIT, E. R. Desenvolvimento de objetos de aprendizagem baseado em especificações de normatização scorm para o caso de suporte à aprendizagem de funções. *RENOTE'07: Revista Novas Tecnologias na Educação*, v. 5, n. 1, 2007.
- W3C SCXML – State Chart extensible Markup Language: State Machine Notation for Control Abstraction. 2008.
Disponível em: <http://www.w3.org/TR/scxml/> [25/10/2009]
- WILEY, D. A. *Learning object design and sequencing theory*. Tese de Doutorado, Brigham Young University, 2000.
- WILEY, D. A. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In: *The Instructional Use of Learning Objects*, Bloomington, IN: Association for Educational Communications and Technology, 2001.
- YASKIN, D.; EVERHART, D. Blackboard learning system (release 6): Product overview. 2002.
Disponível em: <http://www.blackboard.com> [08/04/2009]