

WILSON CASTELLO BRANCO NETO

**WEB SEMÂNTICA NA CONSTRUÇÃO DE
SISTEMAS DE APRENDIZAGEM ADAPTATIVOS**

**FLORIANÓPOLIS – SC
2006**

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA
DA COMPUTAÇÃO**

Wilson Castello Branco Neto

**WEB SEMÂNTICA NA CONSTRUÇÃO DE
SISTEMAS DE APRENDIZAGEM ADAPTATIVOS**

Tese submetida à Universidade Federal de Santa Catarina como parte dos requisitos para obtenção do grau de Doutor em Ciência da Computação.

Prof. Dr. Fernando Álvaro Ostuni Gauthier
Orientador

FLORIANÓPOLIS, FEVEREIRO DE 2006

WEB SEMÂNTICA NA CONSTRUÇÃO DE SISTEMAS DE APRENDIZAGEM ADAPTATIVOS

Wilson Castello Branco Neto

Esta Tese foi julgada adequada para a obtenção do título de Doutor em Ciência da Computação - Área de Concentração: Sistemas de Conhecimento - e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Banca Examinadora:

Prof. Raul Sidnei Wazlawick, Dr.
Coordenador do Curso

Prof. Fernando Álvaro Ostuni Gauthier, Dr.
Orientador

Prof. Luiz Antônio Moro Palazzo, Dr.

Prof. Marco Antônio Neiva Koslosky, Dr.

Prof. Rogério Cid Bastos, Dr.

Prof. Silvia Modesto Nassar, Dra.

Prof. Vânia Ribas Ulbricht, Dra.

“Coragem, coragem, se o que você quer é aquilo que pensa e faz. Coragem, coragem, eu sei que você pode mais. Muito mais...”

Raul Seixas.

A Deus, pela constante presença
através das diversas pessoas
que me rodeiam e fortalecem.

AGRADECIMENTOS

Ao cumprir mais uma importante etapa em minha vida acadêmica, olho para trás e percebo a grande quantidade de pessoas que me auxiliaram na realização deste sonho. Assim, deixo meu muito obrigado...

... aos meus pais, Ilson e Célia, e minhas irmãs, Vanessa e Fernanda, pelo apoio e confiança dispensados.

... a minha esposa, Mariana, pelo amor, compreensão e paciência nos momentos difíceis.

... aos professores Fernando Gauthier e Silvia Nassar, pelos conselhos e pela liberdade proporcionada durante a realização deste trabalho.

... à tia Marilda, tio Luiz, Carla, Breno e Gustavo que estavam sempre prontos para me receber durante minhas viagens a Florianópolis.

... a todos os guias espirituais e médiuns que de alguma forma me apoiaram durante a realização deste trabalho.

... aos meus colegas de trabalho da UNIPLAC, Alexandre e Ângelo, pelo companheirismo e amizade.

... aos meus amigos e amigas de São Joaquim, por me proporcionarem momentos agradáveis há tantos anos.

Mais uma vez peço desculpas àqueles que não foram citados, pois é impossível listar em uma página, todas as pessoas que ajudaram a construir uma vida.

SUMÁRIO

<i>Lista de Figuras</i>	<i>ix</i>
<i>Lista de Quadros</i>	<i>xi</i>
<i>Lista de Siglas</i>	<i>xiii</i>
<i>Resumo</i>	<i>xv</i>
<i>Abstract</i>	<i>xvi</i>
1. Introdução	1
1.1 Considerações Iniciais	1
1.2 Identificação do Problema	4
1.3 Objetivos	5
1.4 Justificativa	5
1.5 Estrutura do Trabalho	7
1.6 Metodologia e Delimitação da Pesquisa	8
2. Migrando da WEB para a WEB Semântica	10
2.1 Um Breve Histórico	10
2.2 Web: O Sonho e a Realidade	12
2.2.1 Limitações da Web Atual	13
2.2.2 Como Superar estas Dificuldades	16
2.3 Web Semântica	17
2.3.1 Aplicações da Web Semântica	19
2.4 Considerações	24
3. Tecnologias Necessárias para a Web Semântica	25
3.1 Estrutura da Web Semântica	25
3.2 XML	27
3.2.1 Estrutura de um Documento XML	31
3.2.2 Validando Documentos XML	33
3.2.3 Esquemas XML (XML Schemas)	35
3.2.4 Namespaces	38
3.2.5 Considerações sobre XML para a Web Semântica	40
3.3 Resource Description Framework	40
3.3.1 Representação de Dados em RDF	43
3.3.2 Outras Características do RDF	49
3.3.3 Representação de Sentenças RDF em XML	51
3.3.4 RDF Schema	55
3.3.5 Considerações sobre RDF para a Web Semântica	58
3.4 Ontologias	59
3.4.1 Aplicações de Ontologias	63
3.4.2 Web Ontology Language	64
3.4.3 Construção de Ontologias	72
3.4.4 Regras para a Web Semântica	73
3.4.5 Considerações sobre Ontologias para a Web Semântica	74

3.5 Web Services	74
3.5.1 Componentes Básicos de um Web Service	75
3.5.2 Considerações Sobre Web Services para a Web Semântica	76
3.6 Considerações	76
4. Educação Baseada na Web	78
4.1 Histórico e Conceitos	78
4.2 Sistemas Tutores Inteligentes	81
4.2.1 Modelo de Domínio	83
4.2.2 Modelo do Aluno	84
4.2.3 Modelo do Tutor	85
4.2.4 Classificação de Sistemas Tutores Inteligentes	86
4.3 Hipermídia Adaptativa	88
4.3.1 Apresentação Adaptativa	90
4.3.2 Navegação Adaptativa	91
4.3.3 Modelagem do Usuário em Hipermídia Adaptativa	92
4.4 Sistemas Gerenciadores de Aprendizagem e Objetos de Aprendizagem	93
4.5 Problemas da Educação Baseada na Web	94
4.6 Padrões de Metadados para Educação Baseada na Web	97
4.6.1 O Padrão IEEE 1484.12.1 sobre Metadados para Objetos de Aprendizagem	98
4.6.2 O Padrão IEEE 1484.2 Informações Públicas e Privadas de Alunos (PAPI)	102
4.6.3 O Padrão SCORM	103
4.7 Propostas de Ambientes Educacionais Baseados na Web Semântica	108
4.7.1 Multitutor	108
4.7.2 INES – Servidores Educacionais Inteligentes	110
4.7.3 OntAWare	112
4.7.4 OntoAIMS	113
4.7.5 MOT	114
4.8 Teorias Educacionais	114
4.9 Considerações	119
5. Descrição da Arquitetura Proposta	120
5.1 Definição dos Requisitos	120
5.2 Arquitetura do Sistema	122
5.3 Modelagem das Ontologias	124
5.3.1 Ontologia LOM	126
5.3.2 Ontologia Pedagógica	131
5.3.3 Ontologia do Estudante	137
5.3.4 Ontologia de Domínio	143
5.4 Transformação dos Dados da Ontologia em um Curso	156
5.5 Considerações	159
6. Atlas - Sistema Baseado na Web Semântica para Geração de Cursos	160
6.1 Caracterização do Sistema	160
6.2 Módulos do Sistema	160
6.2.1 O Módulo do Administrador	161
6.2.2 O Módulo para Iniciar a Construção de um Curso	162
6.2.3 O Módulo para Seleção de Conceitos	164
6.2.4 O módulo para Definição dos Perfis dos Alunos	166
6.2.5 O Módulo para Definição dos Objetivos de Aprendizagem	171
6.2.6 O Módulo para Definição das Atividades de Aprendizagem	175

6.2.7 O Módulo para Definição dos Objetos de Aprendizagem	181
6.2.8 O Módulo para Definição das Estruturas do Curso	186
6.3 Construção do Curso Exemplo	188
6.4 Geração e Disponibilização do Curso	197
6.5 Comparação dos Requisitos com a Arquitetura Desenvolvida	206
6.6 Considerações	208
7. Conclusões e Trabalhos Futuros	209
7.1 Conclusões	209
7.2 Trabalhos Futuros	211
8. Referências Bibliográficas	213

LISTA DE FIGURAS

FIGURA 1 -	Visão de Tim Berners-Lee da <i>Web</i> antes do seu desenvolvimento	14
FIGURA 2 -	Níveis da <i>Web Semântica</i> .	26
FIGURA 3 -	Exemplo de uma página para apresentação de livros.	29
FIGURA 4 -	Visualização dos dados de um arquivo XML através do Stylus Studio	33
FIGURA 5 -	Relação entre um esquema e documentos XML	35
FIGURA 6 -	Representação por grafos de declarações em RDF.	44
FIGURA 7 -	Declaração RDF com URIs representando recursos.	45
FIGURA 8 -	Declaração RDF com URIs representando recursos, predicados e objetos.	46
FIGURA 9 -	Nó em branco para representar relacionamentos n-ários.	49
FIGURA 10 -	Nó em branco para descrever recursos sem URI.	50
FIGURA 11 -	Grafo com informações sobre o livro “Desvendando XML”	52
FIGURA 12 -	Exemplo de Esquema RDF	56
FIGURA 13 -	Espectro de Ontologia	61
FIGURA 14 -	Relação entre ontologias e esquemas XML	62
FIGURA 15 -	Ontologia para publicações científicas	66
FIGURA 16 -	Instâncias da classe <i>Area_Conc_Pub</i>	70
FIGURA 17 -	Passos para acessar um <i>web service</i>	75
FIGURA 18 -	Arquitetura clássica de um STI	83
FIGURA 19 -	Expansão da arquitetura clássica de um STI, por SELF (1999)	86
FIGURA 20 -	Modelo de adaptação com base na modelagem do usuário.	89
FIGURA 21 -	Tecnologias de adaptação em Hipermídia Adaptativa	90
FIGURA 22 -	Modelo conceitual do PAPI	102
FIGURA 23 -	Conjunto de especificações SCORM	104
FIGURA 24 -	Arquitetura do Multitutor	109
FIGURA 25 -	Arquitetura desejada dos sistemas educacionais baseados na <i>Web</i>	111
FIGURA 26 -	Arquitetura interna de um servidor educacional	112
FIGURA 27 -	Visão geral da arquitetura proposta	123
FIGURA 28 -	Estrutura de classes da ontologia auxiliar LOM	126
FIGURA 29 -	Instâncias da classe <i>lom-edu:InteractivityType</i>	129
FIGURA 30 -	Classes da ontologia pedagógica	131
FIGURA 31 -	Instâncias da classe <i>Saskatchewan</i>	133
FIGURA 32 -	Instâncias da classe <i>GagneInstructionalEvents</i>	134
FIGURA 33 -	Instâncias da classe <i>BloomCognitiveTaxonomy</i>	135
FIGURA 34 -	Instâncias da classe <i>KnowledgeVerbs</i>	137
FIGURA 35 -	Estrutura de classes da ontologia do estudante.	139
FIGURA 36 -	Classes da ontologia de domínio	144
FIGURA 37 -	Instâncias da classe <i>Concepts</i> para o domínio de Estatística	146
FIGURA 38 -	Diagrama de navegação do sistema Atlas	161
FIGURA 39 -	Interface inicial do sistema	162
FIGURA 40 -	Interface para importação da ontologia	163
FIGURA 41 -	Identificação do curso	163
FIGURA 42 -	Interface para seleção de conceitos (1)	164
FIGURA 43 -	Interface para seleção de conceitos (2)	165
FIGURA 44 -	Interface para validação de conceitos	166
FIGURA 45 -	Interface para seleção de perfis pedagógicos	167
FIGURA 46 -	Interface para cadastro de perfis técnicos	168
FIGURA 47 -	Interface para definição do nível de conhecimento	169
FIGURA 48 -	Definição da estratégia instrucional	171
FIGURA 49 -	Interface para seleção dos objetivos de aprendizagem (1)	172
FIGURA 50 -	Interface para visualização dos dados de um objetivo de aprendizagem	173

FIGURA 51 -	Interface para seleção dos objetivos de aprendizagem (2)	174
FIGURA 52 -	Interface para criação de novos objetivos de aprendizagem	175
FIGURA 53 -	Diagrama de navegação para a definição das atividades e objetos de aprendizagem	176
FIGURA 54 -	Interface para seleção de atividades de aprendizagem (1)	176
FIGURA 55 -	Interface para visualização dos dados de uma atividade de aprendizagem	177
FIGURA 56 -	Interface para seleção de atividades de aprendizagem (2)	178
FIGURA 57 -	Interface para seleção de atividades de aprendizagem (3)	179
FIGURA 58 -	Interface para criação de atividades de aprendizagem	180
FIGURA 59 -	Interface para seleção dos Eventos Instrucionais	181
FIGURA 60 -	Interface para seleção dos Objetos de Aprendizagem	182
FIGURA 61 -	Interface para apresentar as informações de um LO	182
FIGURA 62 -	Interface para filtrar Objetos de Aprendizagem	183
FIGURA 63 -	Interface para cadastro de LO	184
FIGURA 64 -	Interface para estruturar as Atividades de Aprendizagem (1)	186
FIGURA 65 -	Interface para estruturar as Atividades de Aprendizagem (2)	187
FIGURA 66 -	Interface para estruturar as Atividades de Aprendizagem (3)	188
FIGURA 67 -	Interface principal do LMS compatível com o padrão SCORM	197
FIGURA 68 -	Interface com os cursos disponíveis após a importação do curso gerado	200
FIGURA 69 -	Apresentação do curso gerado para o perfil 1	203
FIGURA 70 -	Apresentação do curso gerado para o perfil 4	204
FIGURA 71 -	Apresentação do curso gerado para o perfil 4 com estratégia de estudo independente	205

LISTA DE QUADROS

QUADRO 1 -	Código HTML da figura 3. _____	29
QUADRO 2 -	Código XML da figura 3. _____	30
QUADRO 3 -	Exemplo de um documento XML _____	31
QUADRO 4 -	Outro exemplo de documento XML. _____	34
QUADRO 5 -	Componentes básicos de um esquema XML _____	36
QUADRO 6 -	Esquema XML _____	37
QUADRO 7 -	Código XML inválido _____	38
QUADRO 8 -	Ficha com informações de um livro. _____	43
QUADRO 9 -	Declarações sobre o livro apresentado no quadro 8 _____	43
QUADRO 10 -	Sentenças RDF apresentadas em triplas _____	48
QUADRO 11 -	Código do grafo da figura 11. _____	52
QUADRO 12 -	Código do grafo da figura 11 sem abreviaturas. _____	54
QUADRO 13 -	Código RDF/XML para nós em branco. _____	54
QUADRO 14 -	Código de um Esquema RDF _____	57
QUADRO 15 -	Instância do Esquema RDF. _____	59
QUADRO 16 -	Ontologia para o modelo da figura 15 (Início). _____	67
QUADRO 17 -	Ontologia para o modelo da figura 15 (Cabeçalho) _____	67
QUADRO 18 -	Ontologia para o modelo da figura 15 (Definição das Classes) _____	68
QUADRO 19 -	Ontologia para o modelo da figura 15 (Definição das Propriedades) _____	69
QUADRO 20 -	Código OWL com instâncias apresentadas na figura 16 _____	71
QUADRO 21 -	Diferenças entre STI e CAI _____	82
QUADRO 22 -	Componentes principais de um arquivo <i>imsmanifest.xml</i> _____	105
QUADRO 23 -	Estrutura das atividades de aprendizagem e regras de sequenciamento _____	106
QUADRO 24 -	Recursos utilizados no curso _____	107
QUADRO 25 -	Requisitos da arquitetura desenvolvida _____	121
QUADRO 26 -	Termos utilizados para representar as ontologias _____	125
QUADRO 27 -	Classes da ontologia LOM _____	127
QUADRO 28 -	Propriedades da ontologia LOM _____	128
QUADRO 29 -	Instâncias da ontologia LOM _____	130
QUADRO 30 -	Classes da ontologia pedagógica _____	132
QUADRO 31 -	Exemplos de Instâncias da ontologia pedagógica _____	138
QUADRO 32 -	Classes, propriedades e instâncias da dimensão técnica _____	140
QUADRO 33 -	Classes e Propriedades da dimensão pedagógica _____	142
QUADRO 34 -	Classes da ontologia de domínio _____	143
QUADRO 35 -	Propriedades e instâncias da classe <i>Concepts</i> _____	145
QUADRO 36 -	Propriedades da classe <i>LearningObjects</i> _____	148
QUADRO 37 -	Instância da classe <i>LearningObjects</i> _____	150
QUADRO 38 -	Propriedades da classe <i>LearningObjects</i> _____	151
QUADRO 39 -	Instância da classe <i>LearningActivities</i> _____	152
QUADRO 40 -	Propriedades da classe <i>LearningObjectives</i> _____	152
QUADRO 41 -	Instâncias da classe <i>LearningObjectives</i> _____	154
QUADRO 42 -	Propriedades da classe <i>Organizations</i> _____	154
QUADRO 43 -	Instâncias da classe <i>Organizations</i> _____	155
QUADRO 44 -	Propriedades da classe <i>Courses</i> _____	156
QUADRO 45 -	Instâncias da classe <i>Courses</i> _____	156
QUADRO 46 -	Relação entre os elementos da ontologia de domínio e do padrão SCORM _____	158
QUADRO 47 -	Perfis de estudantes criados _____	170
QUADRO 48 -	Metadados adicionados ao arquivo do LO <i>Exemplo de Dados Estatísticos</i> _____	185
QUADRO 49 -	Perfis técnicos selecionados _____	189
QUADRO 50 -	Perfis pedagógicos selecionados _____	190

QUADRO 51 -	Perfis pedagógicos x conceitos _____	191
QUADRO 52 -	Objetivos comuns entre os alunos com perfil básico e avançado _____	192
QUADRO 53 -	Objetivo específico para os alunos com perfil avançado _____	192
QUADRO 54 -	Combinações de perfis possíveis no curso de Introdução à Estatística _____	192
QUADRO 55 -	Atividades de Aprendizagem para o perfil 1 (Básico – Doméstico) _____	193
QUADRO 56 -	Atividades de Aprendizagem para o perfil 2 (Básico – Laboratório) _____	195
QUADRO 57 -	Atividades de Aprendizagem para o perfil 3 (Avançado – Doméstico) _____	195
QUADRO 58 -	Atividades de Aprendizagem para o perfil 4 (Avançado – Laboratório) _____	196
QUADRO 59 -	Componentes principais do arquivo <i>imsmanifest</i> gerado pelo Atlas _____	199
QUADRO 60 -	Objetivos de aprendizagem de uma atividade _____	200
QUADRO 61 -	Estrutura das atividades de aprendizagem para o perfil 1. _____	201
QUADRO 62 -	Estratégias instrucionais e forma de navegação _____	202
QUADRO 63 -	Definição dos recursos do curso gerado com LO copiados _____	205
QUADRO 64 -	Definição dos recursos do curso gerado com LO referenciados _____	206

LISTA DE SIGLAS

ADL	- <i>Advanced Distributed Learning</i>
ARPA	- <i>Advanced Research Projects Agency</i> (Agência de Projetos de Pesquisa Avançados)
B2B	- <i>Business to Business</i> (Negócio para Negócio)
B2C	- <i>Business to Consumer</i> (Negócio para Cliente)
CAI	- <i>Computer Aided Instruction</i> (Instrução Auxiliada por Computador)
CAM	- <i>Content Aggregation Model</i> (Modelo de Agregação do Conteúdo)
CD	- <i>Compact Disc</i> (Disco Compacto)
CERN	- <i>Nuclear Research European Center</i> (Centro Europeu de Pesquisa Nuclear)
DAML+OIL	- <i>DARPA Agent Markup Language+Ontology Inference Layer</i> (Linguagem para Marcação de Agentes DARPA + Camada de Inferência em Ontologias)
DARPA	- <i>Defense Advanced Research Projects Agency</i> (Agência de Projetos de Pesquisa Avançados de Defesa)
DTD	- <i>Datatype Definition</i> (Definição de Tipos de Dados)
EAD	- Educação a Distância
EBW	- Educação Baseada na <i>Web</i>
FTP	- <i>File Transfer Protocol</i> (Protocolo para Transferência de Arquivos)
HA	- Hipermídia Adaptativa
HTML	- <i>HyperText Markup Language</i> (Linguagem para Marcação de Hipertexto)
HTTP	- <i>HyperText Transfer Protocol</i> (Protocolo para Transferência de Hipertexto)
IA	- Inteligência Artificial
ILMS	- <i>Intelligent Learning Management System</i> (Sistemas Gerenciadores de Aprendizagem Inteligentes)
LIP	- <i>Learner Information Packaging</i> (Pacote de Informações do Aluno)
LMS	- <i>Learning Management System</i> (Sistemas Gerenciadores de Aprendizagem)
LO	- <i>Learning Object</i> (Objeto de Aprendizagem)
LOM	- <i>Learning Object Metadata</i> (Metadados para Objetos de Aprendizagem)
OWL	- <i>Web Ontology Language</i> (Linguagem para Ontologias Web)
PAPI	- <i>Public and Private Information for Learners</i> (Informações Públicas e Privadas de Alunos)
PLN	- Processamento de Linguagem Natural
RDF	- <i>Resource Description Framework</i> (<i>Framework</i> para Descrição de

	Recursos)
RDF/S	- <i>RDF Schema</i> (Esquema RDF)
RTE	- <i>Run Time Enviroment</i> (Ambiente em Tempo de Execução)
SCORM	- <i>Sharable Content Object Reference Model</i> (Modelo de Referência para Objetos de Conteúdo Compartilháveis)
SHA	- Sistemas de Hipermídia Adaptativa
SN	- <i>Sequencing and Navigation</i> (Sequenciamento e Navegação)
SOA	- Servidor de Objetos de Aprendizagem
SOAP	- <i>Simple Object Access Protocol</i> (Protocolo de Acesso a Objetos Simples)
STI	- Sistemas Tutores Inteligentes
SWRL	- <i>Semantic Web Rule Language</i> (Linguagem de Regras para a <i>Web Semântica</i>)
TCP/IP	- <i>Transmission Control Protocol/Internet Protocol</i> (Protocolo para Controle de Transmissões/Protocolo da Internet)
UDDI	- <i>Universal Description, Discovery, and Integration</i> (Descrição, Descoberta e Integração Universal)
UDI	- <i>Universal Document Identifier</i> (Identificador Universal de Documentos)
URI	- <i>Uniform Resource Identifier</i> (Identificador Uniforme de Recursos)
URL	- <i>Uniform Resource Locator</i> (Localizador Uniforme de Recursos)
URN	- <i>Uniform Resource Name</i> (Nome Uniforme de Recursos)
W3C	- <i>World Wide Web Consortium</i> (Consórcio da WWW)
WSDL	- <i>Web Service Definition Language</i> (Linguagem para Definição de <i>Web Service</i>)
WWW	- <i>World Wide Web</i> (Teia de Alcance Mundial)
XML	- <i>eXtensible Markup Language</i> (Linguagem de Marcação Expansível)

RESUMO

Esta tese aborda algumas dificuldades que surgiram a partir da utilização da *Web* na educação a distância, aqui referenciada como Educação Baseada na *Web* (EBW). Argumenta-se que um dos principais problemas é a lacuna existente entre dois grupos vinculados a ela. Os pesquisadores do primeiro grupo despendem seu tempo e energia estudando o processo de aprendizagem, mas não se preocupam em reduzir os custos e o esforço necessário para desenvolver sistemas educacionais. Por outro lado, os pesquisadores do segundo grupo visam facilitar a construção e a gerência de sistemas e cursos, mas não voltam sua atenção para as questões relacionadas à aprendizagem. Enquanto o primeiro grupo foi responsável pelo surgimento e aprimoramento de tecnologias como Sistemas Tutores Inteligentes (STI) e Sistemas de Hipermídia Adaptativa (SHA), o segundo deu a sua contribuição criando os Sistemas Gerenciadores de Aprendizagem (LMS) e os Objetos de Aprendizagem (LO). Esta divisão nos interesses de pesquisa tem feito com que os STI e SHA ainda sejam vistos como experimentos confinados aos laboratórios, sem uma aplicação prática em larga escala, devido ao custo de seu desenvolvimento. É eminente o predomínio dos LMS e LO devido às facilidades administrativas oferecidas por eles, apesar de não propiciarem condições de aprendizagem tão boas aos alunos. A pesquisa desenvolvida utiliza tecnologias da *Web Semântica* para auxiliar o desenvolvimento de sistemas que possuam benefícios oriundos das pesquisas dos dois grupos citados. Após identificar os principais problemas da EBW, foram modeladas três ontologias que contemplam informações sobre o domínio do conhecimento a ser ensinado, as teorias de aprendizagem e os alunos que participam de tais cursos. Estas informações são utilizadas para facilitar a construção de cursos para a EBW, possibilitando a reutilização de materiais já prontos e permitindo que aspectos das teorias de aprendizagem sejam considerados. Desta maneira, é possível reduzir os custos e o esforço necessário para construir os cursos e melhorar a aprendizagem dos alunos, atendendo aos objetivos dos pesquisadores dos dois grupos citados. Com o intuito de validar as ontologias desenvolvidas, construiu-se um sistema computacional, que utiliza as informações nelas disponíveis, para auxiliar o usuário na construção de um curso. Este curso é salvo em um pacote que segue o Modelo de Agregação de Conteúdo definido no padrão SCORM. Desta maneira, qualquer LMS com suporte ao referido padrão é capaz de executá-lo.

Palavras-chave: *Web Semântica*, Ontologias, Educação Baseada na *Web*, SCORM.

ABSTRACT

This thesis deals with some difficulties that arose from the use of Web in distance learning, referenced here as Web-Based Education (WBE). It argues that one of the main problem is the gap between two groups involved on WBE. The first group have spent their energy and time studying learning processes, although they are not concerned in reducing the costs and the work necessary for the development and management of educational systems. On the other hand, the second group aims at facilitating the construction and management of systems and courses, without turning its efforts towards learning subject. While the first group was responsible for creation and improvement of technologies such as Intelligent Tutoring Systems (ITS) and Adaptive Hypermedia Systems (AHS), the second one contribute by creating Learning Management Systems (LMS) and Learning Objects (LO). Such split between researches interests have made that ITS and AHS have still been seen as experiments restricteds to labs. The dominance of LMS and LO over ITS and AHS is evident, due to the facilities offered by them to manage courses and students, despite of not providing so good learning conditions. The research carried out makes use of Semantic Web technologies to aid in the development of systems that possess benefits provided by researches from the two cited groups. After to identify the main WBE problems, three ontology were modeled with information about domain to be taught in the courses, instructional and learning theories, and about students who participate of such courses. This information is used to make easier the construction of courses made available through the Web, making possible the reuse of existent material and considering aspects from instructional and learning theories. That way, a decrease on costs and effort needed to create courses is possible as well as an improvement on students learning, by satisfying the objectives of researchers from both groups. Aiming to validate ontology developed, a computational system that uses their information was built, in order to aid user to create courses. Such courses are saved in a package conformant with Content Aggregation Model defined in SCORM Standard. That way, any LMS conformant with that standard can run them.

Keywords: Semantic Web, Ontology, Web-Based Education, SCORM.

1. INTRODUÇÃO

1.1 Considerações Iniciais

A humanidade, ao longo de sua história, presenciou o lançamento de uma série de tecnologias de comunicação que causaram grandes impactos sociais, a exemplo do telefone, do rádio e da televisão. Entretanto, nenhuma delas teve um impacto tão grande quanto a Internet no processo de comunicação e interação entre os seres humanos.

A aplicação mais conhecida da Internet é a *World Wide Web* (WWW), ou simplesmente *Web*, nome pelo qual ficou conhecido o serviço de publicação de conteúdo através da criação de hipertextos. Este serviço tem sua força baseada em algumas características que lhe são intrínsecas, como a possibilidade de qualquer pessoa publicar informações e associá-las a outros conteúdos já existentes através de *hiperlinks* (KUROSE e ROSS, 2003). Isto é possível já que a Internet possui uma estrutura distribuída, sem um órgão regulador responsável por catalogar as informações nela disponibilizadas.

Todavia, esta liberdade tem seu preço. Com a ausência de controle sobre o desenvolvimento, publicação e vinculação de *sites*, é muito difícil encontrar alguma informação neste emaranhado sem o auxílio de um mecanismo automatizado de busca. Mesmo com o auxílio destes mecanismos, é notória a dificuldade para localizar informações úteis, devido à quantidade de resultados gerados, sendo grande parte deles irrelevantes.

O problema supracitado existe porque a *Web* que se conhece atualmente foi desenvolvida para ser utilizada apenas por humanos. Como os documentos e serviços nela disponibilizados são escritos em linguagem natural, é necessária inteligência em um grau que apenas os humanos possuem, para entender e utilizar completamente seus recursos (DACONTA, OBRST e SMITH, 2003).

Pesquisadores de Inteligência Artificial (IA) vêm, há anos, trabalhando no desenvolvimento de técnicas que possam dotar os computadores com a capacidade de entender textos escritos em linguagem natural. De acordo com RUSSEL e NORVIG

(2003), os avanços obtidos nesta área devem-se ao fato de que toda aplicação de Processamento de Linguagem Natural (PLN) possui como característica ser focada em um domínio específico. Entretanto, na *Web*, os resultados ainda não são os esperados, pois ela possui informações sobre praticamente todos os assuntos de interesse dos seres humanos. Sem desconsiderar os estudos na área de PLN, surgiu uma outra abordagem, caminhando no sentido oposto, com a finalidade de expandir as potencialidades da *Web*.

Web Semântica é o nome dado a um projeto sob coordenação do W3C (*World Wide Web Consortium*) que, segundo BERNERS-LEE, HENDLER e LASSILA (2001, p.1), disponibiliza “uma nova forma de conteúdo *Web* que possui significado para os computadores”. Enquanto as técnicas de PLN visam fazer com que os computadores entendam a linguagem humana, a *Web Semântica* busca meios mais simples para representar o conhecimento humano, de uma forma que ele possa ser facilmente entendido e manipulado pelos computadores. Ela, além de facilitar a busca das informações na *Web*, permite que os computadores manipulem-nas de forma autônoma, expandindo as possibilidades de aplicações da *Web* atual.

No relatório final do *Workshop* de Pesquisa Estratégica sobre Desafios de Pesquisa e Perspectivas da *Web Semântica* (EUZENAT, 2001) foram definidas algumas áreas chave para o seu desenvolvimento. De acordo com este relatório, o êxito da *Web Semântica* depende de esforços em áreas como Infra-estrutura, Linguagens e Inferência, Ontologias e Fatores Humanos.

Segundo EUZENAT (2001, p. 27), “Uma questão ainda em aberto está relacionada à necessidade de uma infra-estrutura específica para a *Web Semântica*, que seria fornecida pela modificação do protocolo *Web* atual ou por outros protocolos específicos”.

Em uma camada superior, o mesmo autor salienta que são necessárias linguagens para expressar o conteúdo da *Web Semântica*, sendo as semânticas destas linguagens exploradas pelos mecanismos de inferências.

“Ontologias são centrais para a *Web Semântica* porque elas permitem que aplicações concordem sobre termos utilizados enquanto estão se comunicando. Elas são o fator chave para possibilitar a interoperabilidade na *Web Semântica*” (EUZENAT, 2001, p. 33).

Finalizando, EUZENAT (2001, p. 37) expõe que “Pessoas não treinadas puderam ver os benefícios de publicar na *Web* e foram capazes de aprender por exemplos, bem o suficiente para trabalhar com a linguagem HTML (*HyperText Markup Language*)”. Por fim, o mesmo autor pergunta se a *Web Semântica* possui características semelhantes às da *Web*, para que ela também seja facilmente aceita pelos usuários. Esta dúvida mostra que

para a *Web Semântica* atingir o mesmo sucesso da *Web*, ela precisa ser amplamente incorporada pelos usuários e, para que isto aconteça, deve possuir um mecanismo onde pessoas com pouco conhecimento em computação sejam capazes de estruturar, anotar e publicar suas informações. Estas questões estão sendo pesquisadas e avanços significativos já foram realizados para permitir que a *Web Semântica* se torne uma realidade.

Diversas aplicações têm sido propostas para fazerem uso desta arquitetura. Dentre as mais citadas na literatura, FENSEL *et. al.* (2003) destaca a gerência de documentos, comércio eletrônico (*e-commerce*), ciência eletrônica (*e-science*), assistentes pessoais e, agora, estuda-se como a *Web Semântica* pode auxiliar na Educação Baseada na *Web* (EBW), ou seja, o foco deste trabalho.

O termo Educação Baseada na *Web* surgiu a partir da integração da *Web* às antigas práticas de Educação a Distância (EAD). Segundo MOORE e KEARSLEY (1996), educação a distância é uma forma de ensino-aprendizagem planejada que, normalmente, ocorre em lugar diverso do professor e como consequência requer técnicas especiais de planejamento de curso, técnicas instrucionais diferenciadas, métodos especiais de comunicação, sejam eles eletrônicos ou não, bem como estrutura organizacional e administrativa específica.

O enfoque da EAD foi modificado à medida que novas tecnologias para a realização da mesma surgiam. Até o início da década de 90, o ponto chave era a liberdade espaço-temporal que ela possibilitava, através de um processo de auto-aprendizagem baseado em livros, fitas de áudio e vídeo e CD-ROM. Após a disponibilização das tecnologias de rede para a EAD e, particularmente, dos ambientes virtuais de aprendizagem que passaram a integrar professores e alunos em tempo real, o conceito de liberdade espaço-temporal que estava já consolidado deu lugar à aproximação virtual.

Desde então, diferentes abordagens têm sido estudadas visando melhorar a aproximação virtual e auxiliar o aluno em seu processo de aprendizagem, o que implica na busca pela identidade deste aluno, com o intuito de adaptar o sistema as suas necessidades, características e dificuldades. Dentre estas abordagens podem ser destacadas a Hipermídia Adaptativa (HA) e os Sistemas Tutores Inteligentes (STI). Entretanto, vários fatores dificultam o desenvolvimento de sistemas com tais características, devido à forte relação que existe entre o domínio do conhecimento e as técnicas adaptativas e educacionais utilizadas. Isto dificulta a reutilização de sistemas, cursos inteiros ou de partes deles, o que torna o processo de desenvolvimento dispendioso e lento e faz com que a maioria dos sistemas de EBW comerciais não possuam tais características.

A proposta da *Web Semântica* pode vir ao encontro destas necessidades e contribuir para o desenvolvimento de novos cursos baseados na *Web*, permitindo que ontologias e conteúdos descritos por metadados sejam utilizados e reutilizados por qualquer pessoa interessada em criar um curso com características adaptativas e embasado nas teorias educacionais.

1.2 Identificação do Problema

Por ser a *Web Semântica* um campo de pesquisa recente, ainda existem muitas questões a serem pesquisadas, principalmente quanto a sua aplicação na EBW. A seguir, destacam-se alguns pontos abordados em eventos e periódicos da área, entre eles o *International Forum of Educational Technology & Society* (ET&S, 2004) e o *International Workshop on Applications of Semantic Web Technologies for E-Learning* (SW-EL, 2004):

- Criação de arquiteturas para sistemas educacionais baseados em ontologias;
- Desenvolvimento de ontologias para EBW;
- Construção de *Web Services* semânticos para EBW;
- Estudo da engenharia de ontologias e formas de representação do conhecimento sobre o processo educacional;
- Utilização de ontologias e outras tecnologias da *Web Semântica* para apoio à criação de ambientes educacionais e cursos baseados na *Web*;

Além destas necessidades, podem ser destacadas outras questões que requerem estudos, visando facilitar tanto a interação dos alunos com os cursos e ambientes educacionais quanto a construção dos mesmos. São elas:

- Como utilizar tecnologias da *Web Semântica* para possibilitar uma EBW personalizada e adaptativa;
- Como usar as tecnologias de *Web Semântica* e ontologias para evoluir dos STI tradicionais para STI baseados na *Web*, com maior capacidade para definição do modelo do aluno e maior grau de adaptabilidade ao perfil de cada um deles;
- Que protocolos e ontologias podem ser utilizados para troca de informações (interoperabilidade) entre sistemas educacionais adaptativos;
- Que características são necessárias às ontologias para que elas possibilitem a reutilização de conteúdos, técnicas adaptativas e educacionais para facilitar a construção de cursos baseados na *Web*.

A partir das dificuldades previamente expostas, pode-se definir que o problema abordado nesta tese é a escassez de métodos baseados nas tecnologias da *Web Semântica*, capazes de reduzir a complexidade de desenvolvimento de cursos e ambientes educacionais baseados na *Web*, adaptativos e fundamentados nas teorias educacionais, através da reutilização de componentes semanticamente descritos.

1.3 Objetivos

O objetivo principal desta tese é modelar uma arquitetura para ambientes de aprendizagem adaptativos, fundamentados nas teorias de aprendizagem que permitam criar materiais didáticos, utilizando descrições semânticas. São objetivos específicos deste trabalho:

- Criar a estrutura de uma ontologia capaz de representar as informações sobre diferentes domínios do conhecimento;
- Modelar duas ontologias, uma com informações advindas das teorias educacionais e outra com informações que descrevam os alunos participantes dos cursos;
- Apresentar uma arquitetura baseada nestas ontologias que possibilite a reutilização de materiais didáticos e siga as teorias de aprendizagem, durante a construção de cursos baseados na *Web*;
- Representar as informações sobre os componentes instrucionais existentes nas ontologias através do Modelo de Agregação de Conteúdo (CAM) definido pelo padrão SCORM (*Sharable Content Object Reference Model - Modelo de Referência para Objetos de Conteúdo Compartilháveis*);
- Avaliar a arquitetura proposta por meio do desenvolvimento de um curso na área de Estatística.

1.4 Justificativa

Os benefícios trazidos pela EAD aos indivíduos que não têm acesso ao ensino tradicional são óbvios, pois ela permite que estas pessoas tenham acesso a novos conhecimentos, expandindo, desta forma, seus horizontes e melhorando suas oportunidades no mercado de trabalho.

Além das oportunidades oferecidas àqueles que não tiveram acesso à educação

presencial, a EAD pode, também, contribuir de maneira significativa com as pessoas que já completaram seus estudos, seja em nível fundamental, médio, superior ou pós-graduação, possibilitando a formação continuada dos mesmos.

Contudo, ainda são muitos os obstáculos a serem superados para que a EAD e, mais especificamente, a EBW cumpram os seus propósitos, alcançando melhores resultados e proporcionando uma educação de qualidade aos alunos. Muitos esforços têm sido realizados para aumentar a eficiência dos ambientes de EBW, através dos STI e Sistemas de Hiperídia Adaptativa (SHA), como os apresentados em BRUSILOVSK (1999 e 2003), MURRAY (1999) e DEVEDZIC (2004).

Os resultados alcançados com STI e SHA são bastante positivos e a melhora na aprendizagem dos alunos é significativa, o que faz com que este tipo de ensino seja cada vez mais procurado. No entanto, alguns fatores ainda dificultam a criação de sistemas de EBW adaptativos e fundamentados nas teorias de aprendizagem. Segundo CALVO (2003), o maior problema é o alto custo para o desenvolvimento deste tipo de sistema, uma vez que as técnicas adaptativas e educacionais existentes atualmente são fortemente vinculadas ao domínio a ser ensinado e a taxa de reutilização das mesmas é muito baixa.

Entende-se que a utilização de recursos da *Web Semântica* seja primordial para a solução deste problema, pois ela permite a construção de serviços que tornam mais fácil o desenvolvimento de ambientes e cursos baseados na *Web* adaptativos e fundamentados nas teorias de aprendizagem, sem que a pessoa responsável pelo seu desenvolvimento precise despender muito tempo nem esforço de programação. Através da descrição semântica dos conteúdos, dos alunos e da definição de ontologias é possível construir um sistema com técnicas adaptativas e educacionais genéricas que podem ser utilizadas na construção de diferentes cursos baseados na *Web*, melhorando a aprendizagem do aluno.

O padrão SCORM foi escolhido para representar o curso gerado por englobar características importantes presentes em vários outros padrões de metadados, além de adicionar algumas outras relevantes não tratadas nos demais, e possuir uma documentação detalhada sobre todos os aspectos relacionados à estrutura do curso e às funcionalidades dos sistemas que os tornam disponíveis aos alunos.

A originalidade desta pesquisa reside na abordagem utilizada para representar as informações sobre os domínios do conhecimento, as teorias educacionais e os alunos que participam dos cursos. Enquanto alguns trabalhos desenvolvidos contemplam estas informações apenas em domínios específicos, outros que as generalizam não conseguem estabelecer as relações entre os diferentes aspectos (domínio, teorias educacionais e

alunos). A utilização de ontologias possibilita que uma estrutura genérica para todas estas informações seja criada, independente do domínio do conhecimento que venha a ser considerado posteriormente.

Para complementar a característica genérica das ontologias e reforçar a originalidade da pesquisa, as mesmas são descritas através da linguagem OWL (*Web Ontology Language*) e independentes de um modelo de metadados para representação de informações educacionais, como o Modelo de Agregação de Conteúdo - SCORM ou o IMS Learning Design. Desta maneira, sistemas baseados nas ontologias para gerar cursos em qualquer padrão de metadados podem ser construídos, desde que o seu desenvolvedor crie um mapeamento entre as informações dispostas nas ontologias e os elementos do padrão escolhido, uma vez que nesta tese é apresentado apenas um mapeamento para o padrão SCORM, utilizado na validação das ontologias e sistema desenvolvido.

1.5 Estrutura do Trabalho

Esta tese está dividida em 7 capítulos, organizados conforme segue.

O capítulo 1, inicialmente, faz uma breve apresentação da *Web Semântica* e destaca de que forma ela pode contribuir para melhorar a EBW. Na seqüência, descrevem-se o problema, os objetivos desta pesquisa e a motivação para o desenvolvimento da mesma. Por fim, consta este item com a estrutura dos capítulos, seguido da metodologia de desenvolvimento e das delimitações da pesquisa.

No capítulo 2 são descritos os conceitos relacionados à *Web Semântica*, partindo da criação das primeiras tecnologias de informação e comunicação, passando pelo surgimento da Internet e da *Web* e chegando ao “sonho” da *Web Semântica*. Finalizando este capítulo, estão alguns exemplos de aplicações que mostram o seu potencial.

O capítulo 3 está relacionado às tecnologias básicas necessárias para o desenvolvimento de aplicações nos moldes desejados pelos pesquisadores da área. Seguindo a arquitetura proposta pelo W3C, iniciou-se com as tecnologias que estão nos níveis mais baixos como URI (*Uniform Resource Identifier* – Identificador Uniforme de Recursos), Unicode, XML (*eXtensible Markup Language*- Linguagem de Marcação Expansível) e *Namespaces*, chegando às linguagens com maior poder de representatividade como RDF (*Resource Description Framework* – Framework para Descrição de Recursos), RDF/S (*RDF Schema* – Esquema RDF) e Ontologias. Além destas linguagens, este capítulo trata de *Web Services*, por serem eles os responsáveis por parte da autonomia

requerida pela *Web Semântica*.

O capítulo 4 inicia com um breve resgate da história da EAD, visando mostrar a sua importância para a sociedade atual. Em seguida, são apresentados os principais conceitos relacionados às tecnologias utilizadas na EBW como STI, SHA, Sistemas Gerenciadores de Aprendizagem (LMS) e Objetos de Aprendizagem (LO), e são discutidos os problemas relativos a cada uma delas, com vistas à definição dos requisitos da arquitetura desenvolvida. Na seqüência deste capítulo, são descritos padrões de metadados para representação de conteúdos educacionais e de alunos. Além disso, são listados alguns trabalhos relacionados, com o intuito de melhor definir os requisitos da arquitetura desenvolvida. Por fim, apresenta-se um resumo sobre as teorias de aprendizagem utilizadas na construção das ontologias.

No capítulo 5 apresenta-se a arquitetura proposta e descrevem-se as ontologias que subsidiaram o desenvolvimento do sistema para geração de cursos baseados na *Web*.

O capítulo 6 expõe o sistema desenvolvido com base nas ontologias descritas no capítulo 5. Também demonstra a utilização do referido sistema para a construção de um curso na área de estatística e a sua comparação com os requisitos da arquitetura, com o intuito de validá-la.

Finalizando, encontram-se no capítulo 7 as conclusões deste trabalho e algumas sugestões para a continuidade do mesmo.

1.6 Metodologia e Delimitação da Pesquisa

A pesquisa realizada que culminou com o desenvolvimento desta tese é de natureza aplicada, pois o conhecimento gerado deve auxiliar a resolver problemas específicos ligados à EBW. O problema foi abordado sob uma perspectiva qualitativa, sendo a validação realizada através da comparação com os requisitos propostos para o mesmo, sem a utilização de métodos quantitativos (SILVA e MENEZES, 2005).

De acordo com DODIG-CRNKOVIC (2002), do ponto de vista procedimental ela pode ser considerada uma pesquisa experimental, fundamentada em um levantamento bibliográfico sobre o assunto, como pode ser visto nas etapas descritas a seguir.

Para alcançar os objetivos propostos, inicialmente estudou-se a área de aplicação escolhida, a EAD e, mais detalhadamente, a EBW, com o objetivo de conhecer suas principais tendências, focando nos STI, SHA, LMS, LO e nas tecnologias capazes de integrá-la à *Web Semântica*. Com este estudo, verificou-se de que forma estas técnicas

estão beneficiando a EBW.

Em seguida, foi realizado um estudo sobre os principais conceitos, objetivos e métodos da *Web Semântica*, com o intuito de compreender a estrutura que já foi desenvolvida e a que está planejada para acontecer nos próximos anos. Através deste estudo, identificou-se o que é e o que “ainda” não é possível desenvolver com as tecnologias existentes.

Após o estudo destas áreas, iniciou-se o trabalho de modelagem da arquitetura proposta a partir das ontologias necessárias para a elaboração do sistema desenvolvido posteriormente. Na etapa seguinte, foi projetado o sistema para construção de cursos baseados na *Web* adaptativos e fundamentados nas teorias educacionais. Por fim, o sistema foi testado através da criação de um curso na área de Estatística e seus resultados comparados aos requisitos da arquitetura, com o intuito de validá-la.

Por ser a *Web Semântica* um ramo de pesquisa relativamente novo, ainda há muito a fazer até que as aplicações desenvolvidas alcancem o nível de autonomia desejado. Ao analisar as camadas projetadas para a *Web Semântica*, que são apresentadas posteriormente, isto fica evidente, pois, para apenas algumas delas existem tecnologias capazes de implementar por completo as suas funções. A presente tese não pretende propor um sistema que utilize os recursos das camadas ainda não desenvolvidas, por ser necessário buscar soluções para problemas como o desenvolvimento de provas lógicas para as informações obtidas ou a confiabilidade entre sistemas. Assim, seu escopo limita-se à utilização das tecnologias já existentes para melhorar o processo de ensino-aprendizagem em ambientes de EBW.

Também não se pretende desenvolver uma nova arquitetura para STI, SHA, LMS ou LO. O que se espera com este trabalho é facilitar a construção de cursos baseados na *Web*, através da utilização e reutilização das técnicas já existentes, devido à descrição semântica das mesmas.

2. MIGRANDO DA *WEB* PARA A *WEB* SEMÂNTICA

Este capítulo apresenta um breve histórico das principais transformações causadas pela tecnologia na maneira pela qual as pessoas interagem e se relacionam. Em seguida, destacam-se a Internet e a *Web*, que atualmente representam o ápice destas modificações, com seus sucessos e limitações. Por fim, apresenta-se de forma abrangente a *Web* Semântica, destacando algumas áreas que podem ser beneficiadas com as suas tecnologias.

2.1 Um Breve Histórico

Cada um dos últimos três séculos foi claramente dominado por uma nova tecnologia. O século XVIII foi o berço dos grandes sistemas mecânicos, característica da Revolução Industrial. No século XIX surgiram as máquinas a vapor. Por fim, o século XX causou uma revolução no campo da informação e das comunicações (TANENBAUM, 1997). No âmbito das comunicações destacaram-se, entre outros, a telefonia, o rádio e a televisão. Relacionado ao processamento de informação, foi, sem dúvida, o computador a ferramenta que causou as maiores transformações. Entretanto, nenhum deles individualmente causou mudanças tão profundas na forma com que as pessoas e corporações interagem como a Internet. A fusão dos computadores com os sistemas de comunicação abriu novos horizontes nas relações interpessoais, sejam estas para fins de entretenimento, estudos ou negócios.

A Internet tem seu nascimento datado da década de 60 do século passado. Pesquisadores, com o apoio da ARPA (*Advanced Research Projects Agency* – Agência de Projetos de Pesquisa Avançados), iniciaram o projeto da ARPANet, a primeira rede por comutação de pacotes e um ancestral direto da Internet atual. O primeiro uso da rede ocorreu em 1969, interligando computadores nas Universidades da Califórnia, Santa Bárbara, Utah e no Instituto de Pesquisa de Standford. Inicialmente, esta rede era fechada e única. Para ter acesso a um computador da ARPANet era necessário estar em um dos outros computadores ligados a ela (TANENBAUM, 1997).

Durante a década de 70, outras redes começaram a aparecer, porém, todas fechadas e

com tecnologias próprias. Em seu livro, KUROSE e ROSS (2003, p. 45) dizem “(...) com conhecimento do passado, podemos ver que já era chegada a hora de desenvolver uma arquitetura abrangente para comunicação de redes”. Este trabalho foi iniciado por Vinton Cerf e Robert Kahn, com o apoio do DARPA (*Defense Advanced Research Projects Agency* – Agência de Projetos de Pesquisa Avançados de Defesa), gerando o termo *internetting*, que define a essência do trabalho que era criar a rede das redes (KUROSE e ROSS, 2003).

Alguns dos princípios mais importantes do trabalho de Cerf e Kahn, como autonomia e controle descentralizado, foram englobados ao protocolo TCP/IP (*Transmission Control Protocol/ Internet Protocol*) e são mantidos até hoje. Enfim, o embrião estava formado. No final dos anos 70, existiam aproximadamente 200 computadores conectados à ARPANet. À década de 80 coube a responsabilidade de desenvolvê-la e disseminá-la. No dia 1º de Janeiro de 1983, o TCP/IP foi definido como o protocolo padrão para comunicação entre as máquinas da ARPANet e de todas as demais redes que desejassem se ligar a uma rede de longa distância. Ao mesmo tempo, ela foi dividida em duas redes: uma para fins militares, a MILNet; e outra, que continuou se chamando ARPANet, para fins de pesquisa. Foi esta última que em seguida passou a se chamar Internet e chegou aos anos 90 com 100.000 computadores conectados (COMER, 1998).

A última década do século XX também apresentou eventos que merecem destaque na história da Internet, pois foi nesta época que ela se tornou comercial e surgiu a *World Wide Web* (WWW), uma de suas principais aplicações.

As idéias básicas da *Web* foram se solidificando ao longo de várias décadas. Dentre os acontecimentos, cabe destacar o artigo sobre um dispositivo foto-eleto-mecânico para extensão de memória capaz de criar e seguir *links* entre documentos em microfichas, escrito em 1945, por Vannevar Bush. Em 1960, Doug Engelbart define o termo hipertexto e desenvolve um protótipo que permite criação e busca em hipertextos. Tim Berners-Lee começa a participar da história da *Web* quando, em 1980, cria um programa para uso particular que permite a criação de nós para armazenar informações, os quais possuíam um título, um tipo e uma lista de *links* bidirecionais para outros nós também cadastrados no sistema (CONNOLLY, 2000).

Em 1989, enquanto trabalhava no Laboratório Europeu de Partículas Físicas, no CERN (*Nuclear Research European Center* – Centro Europeu de Pesquisa Nuclear), Tim Berners-Lee propôs uma extensão de seu programa particular, criado em 1980, gerando, assim, um espaço global de hipertexto. Foi nesse momento que a idéia do hipertexto

deixou de ser utilizada em sistemas fechados, onde os arquivos só continham referências para outros arquivos já cadastrados no mesmo sistema. Ele passou, então, a ser empregado em espaços mais amplos, sendo os arquivos referenciados através das redes de computadores, independentemente de sua localização física, da plataforma e do sistema operacional utilizado na máquina em que está armazenado. Neste espaço, qualquer informação acessível pela rede poderia ser referenciada por um Identificador de Documentos Universal (UDI), hoje denominado URI (BERNERS-LEE, 2004).

O projeto intitulado “Information Management: A Proposal” (BERNERS-LEE, 1990), discutia o problema de perda de informação em sistemas complexos e dinâmicos e derivou uma solução baseada em sistemas de hipertexto distribuídos, que mais tarde se tornaria a *Web*. De acordo com (BERNERS-LEE, 1990, p. 2), o projeto visava responder uma pergunta constante nas reuniões do CERN, “Sim, mas como nós vamos manter os registros de um projeto tão grande?”.

Durante a execução do projeto, em 1990, foram escritos o primeiro editor de hipertexto e o primeiro servidor *Web*, que juntamente com a especificação de UDI, HTML e HTTP (*HyperText Transfer Protocol*), publicados no primeiro servidor *Web*, permitiram a ampla adoção e disseminação da *Web* (BERNERS-LEE, 2004).

De 1991 a 1994, a carga do primeiro servidor *Web* (info.cern.ch) cresceu a uma taxa de 10 vezes ao ano. Em 1992, a academia foi tomando conhecimento do projeto e, em 1993, a indústria (BERNERS-LEE, 2004). Os últimos anos presenciaram o vertiginoso crescimento da *Web*, atingindo, em 2005, mais de 970 milhões de usuários (INTERNET, 2005).

2.2 Web: O Sonho e a Realidade

Conforme descrito em BERNERS-LEE (1990), a proposta inicial do idealizador da *Web* visava criar uma forma para armazenar informações sobre grandes projetos no CERN, evitando, assim, a perda e facilitando o acesso às mesmas. Alguns requisitos deste projeto, tais como o acesso remoto através de redes, heterogeneidade, não centralização, acesso a dados já existentes, possibilidade de criar *links* particulares e analisar dados, definiram muitas das características da *Web* atual.

Para Tim Berners-Lee, “A *Web* que se vê hoje, como um glorificado canal de televisão é apenas uma parte do sonho original” (BERNERS-LEE, 2003, p. xiv). Embora ela tenha sido originada pela necessidade de organização do trabalho em grandes grupos,

foi fora do ambiente de trabalho, mais especificamente no ramo da publicação de informações que a *Web* teve seu maior crescimento.

O sonho do criador da *Web*, que se divide em duas partes, é o de um espaço de informação comum, onde as pessoas podem se comunicar e interagir através do compartilhamento de informações. Sua universalidade é essencial, pois um *link* pode referenciar tudo, seja isto pessoal, local ou global, apenas um esboço ou algo cuidadosamente acabado (BERNERS-LEE, 2003).

Com a grande quantidade de material disponível na *Web*, a primeira parte do sonho foi realizada. Entretanto, a segunda, que é ver a *Web* sendo usada tão genericamente que ela se torne um espelho real de como os humanos trabalham, se divertem e se relacionam ainda não aconteceu, mas existem sinais e planos que deixam os pesquisadores confiantes (BERNERS-LEE, 2004).

Segundo DACONTA, OBRST e SMITH (2003), o seu criador busca uma *Web* mais colaborativa e que possa ser entendida e processada por máquinas. Para que isto seja possível, é preciso repensar a forma com que os conteúdos são publicados na *Web*, pois páginas escritas em linguagem natural e formatadas através da linguagem HTML dificultam a criação de serviços com maiores níveis de autonomia, devido às dificuldades encontradas pelos programas para obtenção de informações relevantes e, conseqüentemente, para o processamento das mesmas.

A figura 1, extraída do projeto original de Berners-Lee, possui relacionamentos como “descreve”, “inclui” e “escreveu”, que não podem ser capturados pela *Web* atual. Ela deixa claro que na concepção de seu criador, os serviços disponíveis via *Web* devem ir muito além de, simplesmente, disponibilizar arquivos HTML.

2.2.1 Limitações da Web Atual

A *Web* que se popularizou nos últimos anos e modificou a forma com que as pessoas se comunicam e interagem é composta por serviços voltados, quase que totalmente, para os seres humanos. São inegáveis as novas possibilidades por ela apresentadas em todos os seguimentos, indo desde a área de negócios, sejam eles apenas entre empresas ou diretamente com os clientes, passando pelos setores de pesquisa e chegando até o entretenimento. Ainda restam, porém, alguns pontos não resolvidos que prejudicam a sua evolução e fazem com que os usuários não tenham acesso a serviços mais interessantes e, por que não dizer, mais úteis que os já conhecidos.

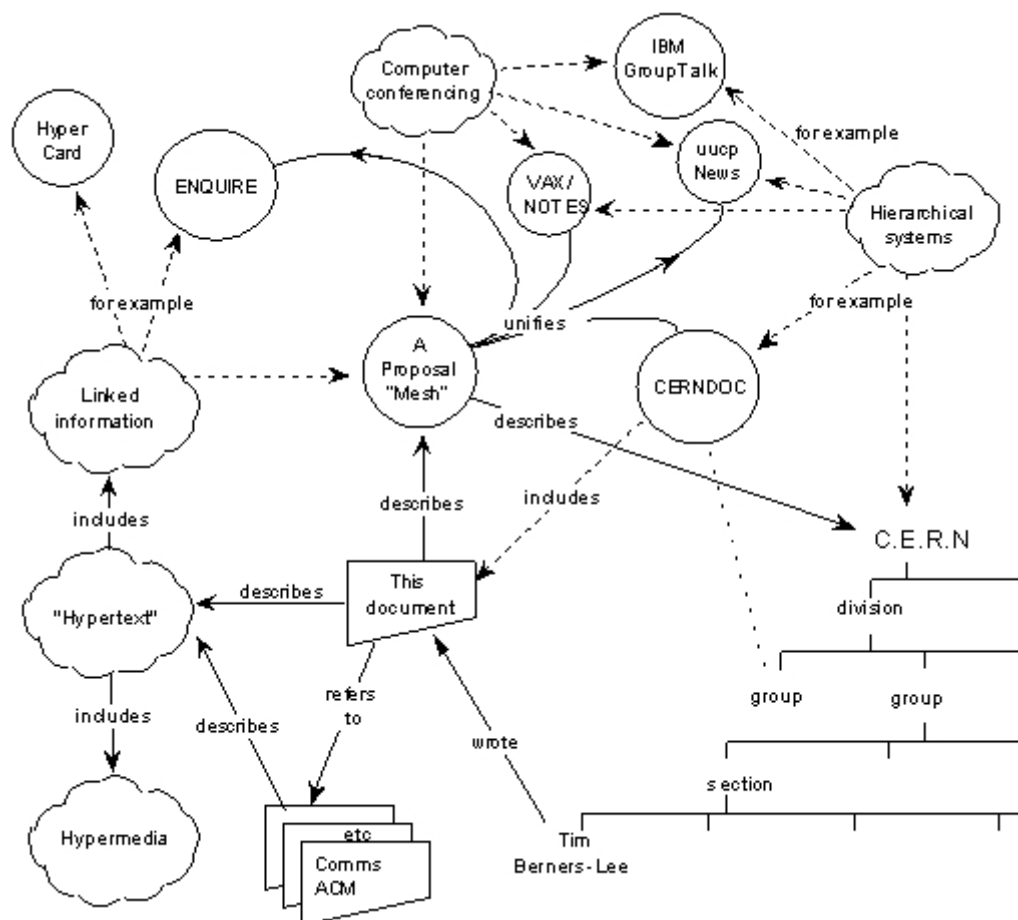


FIGURA 1 - Visão de Tim Berners-Lee da *Web* antes do seu desenvolvimento
 Fonte: (BERNERS-LEE, 1990).

O primeiro ponto está relacionado à busca de informações na *Web*. Esta tarefa, que atualmente é realizada com o auxílio de mecanismos de busca como Google, Yahoo e Altavista, apresenta vários problemas que dificultam o acesso ao conteúdo disponível. De acordo com ANTONIOU e HARMELEN (2004), as principais limitações associadas ao uso dos mecanismos de busca são:

- Alto número de resultados, com baixa precisão - mesmo que a página desejada seja localizada pelo mecanismo de busca, ela será de pouca valia se o usuário precisar encontrá-la entre outras dezenas de milhares de páginas apresentadas simultaneamente;
- Baixo número de resultados - embora seja menos comum, em alguns casos os mecanismos de busca não retornam nenhum resultado, principalmente quando o usuário utiliza termos muito específicos, com o objetivo de reduzir a quantidade de material irrelevante retornado.
- Resultados muito dependentes do vocabulário - muitas vezes algumas páginas interessantes não são retornadas, por utilizarem termos diferentes,

mas que possuem o mesmo significado do que está sendo procurado;

- Resultados são páginas *Web* isoladas - se as informações procuradas estão espalhadas em várias páginas, é necessário fazer várias buscas para coletar todo o material necessário.

Mesmo após encontrar as informações desejadas na *Web*, os usuários enfrentam problemas em razão da forma pela qual tais informações estão estruturadas. Como as páginas *Web* escritas em HTML são voltadas para a apresentação de conteúdo para os seres humanos, a construção de ferramentas que as manipulem é baseada, quase que totalmente, em heurísticas e metodologias oriundas da Inteligência Artificial.

Apesar dos avanços significativos obtidos nesta área, ainda há um longo caminho a ser trilhado para que se possa ter um computador com um grau de inteligência e autonomia próximo ao do ser humano. Esta dificuldade é bastante visível no ramo de processamento de linguagem natural, que é descrito como IA-Completo (em analogia ao NP-Completo). Isto é, produzir um sistema tão competente com linguagens quanto um humano exigiria que fosse resolvido “o problema da IA” (NILSSON, 1998).

Para RUSSEL e NORVIG (2003), os avanços obtidos no processamento de linguagem natural se devem ao fato de que toda aplicação possui como característica ser focada em um domínio específico e executar uma tarefa limitada como tradução; recuperação de informações; categorização de textos; extração de dados de textos e acesso a banco de dados.

Dos tipos de aplicações apresentados, os quatro primeiros são empregados com o intuito de facilitar e melhorar a tarefa de busca e processamento de informações na *Web*. Porém, ela falha no preceito básico para o sucesso de uma aplicação de processamento de linguagem natural, que é possuir informações sobre um domínio específico. Outro problema enfrentado é que estes sistemas são fortemente baseados no conhecimento sintático, sendo ele insuficiente na maioria dos casos, em que o significado das informações é fundamental e não pode ser obtido apenas através das regras sintáticas da linguagem (RUSSEL e NORVIG, 2003).

Este problema se deve principalmente a duas características presentes na linguagem natural, independente de idioma. A primeira é a possibilidade de utilização de sinônimos¹. Este recurso, que é aplicado para melhorar a legibilidade do texto e torná-lo mais agradável, por evitar a repetição excessiva de palavras, é um grande complicador no

¹ Palavras diferentes com mesmo significado.

momento que se deseja criar um sistema que manipule (entenda ou gere) textos em linguagem natural.

Além dos sinônimos, outro fator que dificulta a criação deste tipo de sistemas são os termos homônimos². Muitas vezes, não é suficiente entender o significado de uma palavra isolada, é necessário, também, a compreensão do contexto em que ela está sendo utilizada para poder concluir o seu verdadeiro sentido.

Os problemas citados mostram uma grande lacuna existente entre o tipo de informação que pode ser entendida e produzida pelos humanos e a informação que pode ser manipulada, atualmente, pelos computadores.

De acordo com os pesquisadores de processamento de linguagem natural, os avanços nesta área dependem, diretamente, da capacidade que os sistemas terão de entender o significado das palavras ou sentenças. Após anos de pesquisa, percebeu-se que de nada adianta uma aplicação possuir um extenso e completo dicionário, aliado às regras sintáticas de uma linguagem, se ela não é capaz de considerar o significado, ou a semântica do que está sendo processado, para atingir bons resultados, como destacam CHKLOVSKI e PANTEL (2004) e JURAFSKY e MARTIN (2000).

2.2.2 *Como Superar estas Dificuldades*

O primeiro passo em direção a nova *Web* consiste na criação de informações sobre informações. Tratando do excesso de lixo na *Web*, BERNERS-LEE (2003, p. xiv) diz que “(...) nós precisamos de informações sobre informações – Metadados para organizar isto”. Se faz necessária, também, uma mudança de paradigma quando se considera o dado como secundário em relação ao processamento. É importante que sejam desenvolvidos novos padrões para o armazenamento de dados, que contenham algum nível de inteligência, retirando um pouco da “responsabilidade” dos programas (processamento).

Esta evolução da importância e “inteligência” dos dados pode ser observada através de uma rápida retrospectiva feita por DACONTA, OBRST e SMITH (2003) sobre a forma de armazenamento de dados.

Nos estágios iniciais da computação os dados eram armazenados em arquivos (texto ou binário) e após em bases de dados. Estes dados eram proprietários das aplicações que os

² Palavras iguais com significados diferentes

manipulavam, sendo elas as possuidoras da “inteligência” do sistema. A importância começou a migrar da aplicação para os dados após a criação da linguagem XML, a qual tornou os dados independentes das aplicações dentro de um domínio específico. Em um terceiro estágio dessa evolução encontram-se as taxionomias, que permitem a composição de dados de diferentes domínios, sendo eles classificados hierarquicamente.

Ao final desta linha, encontram-se as ontologias, a partir das quais novos dados podem ser inferidos a partir dos existentes, seguindo regras lógicas. Este tipo de dado é definido por DACONTA, OBRST e SMITH (2003, p. 4) como “dado inteligente”. Segundo eles, “(...) pode-se definir dado inteligente como um dado que é independente de aplicação, componível, classificável e parte de um grande ecossistema de informações (ontologia)”.

É através da criação dos “dados inteligentes” que se pretende expandir as possibilidades da *Web* e trazê-la ao seu completo potencial de aplicação, criando, para isto, linguagens capazes de gerar documentos para uma *Web* projetada para ser processada por máquinas, além de humanos. A *Web* de documentos para humanos está sendo incorporada a uma *Web* de dados entendíveis pelas máquinas e o potencial do trabalho cooperativo entre humanos e máquinas trabalhando juntos e se comunicando através da *Web* pode ser imenso (BERNERS-LEE, 2004). O desejo de ver esta interação entre homens e máquinas ser levada a efeito e com uma intensidade ainda não vista levou os pesquisadores ao desenvolvimento do projeto da *Web Semântica*.

2.3 Web Semântica

“A *Web Semântica* é uma extensão da *Web* atual, na qual um significado bem definido é dado à Informação, permitindo que computadores e pessoas trabalhem em cooperação” (HENDLER, BERNERS-LEE e MILLER, 2002, p. 676). Um exemplo introdutório e que mostra todo o potencial da *Web Semântica* é apresentado a seguir. Nele, agentes inteligentes, utilizando informações estruturadas disponíveis na *Web*, interagem visando atingir um objetivo comum.

A história inicia quando o telefone de Pete toca. Ao atender, automaticamente o seu agente envia uma mensagem para o aparelho de som abaixar o volume. Era sua irmã, Lucy, que estava em um consultório médico com sua mãe, informando que ela precisaria de algumas sessões de fisioterapia. Com a aprovação de Pete, Lucy, através do navegador do seu computador de mão, determinou que seu agente agendasse as sessões. O agente de Lucy recebeu do agente do médico

as informações sobre o **tratamento prescrito**³. Em seguida, procurou-se na lista de **fisioterapeutas** os que atendiam pelo **plano de saúde** de sua mãe, a uma **distância de até 20 milhas** de sua casa e com serviços classificados como **excelente** ou **muito bom**. O Agente de Lucy negociou com os agentes dos fisioterapeutas os **horários** disponíveis, respeitando os compromissos de Lucy e Pete.

O plano apresentado pelo agente de Lucy não agradou a Pete, pelo fato do consultório escolhido ser do outro lado da cidade e por ele ter que voltar para casa em um horário de grande movimento. Pete configurou seu próprio agente para refazer a busca com restrições sobre **localização** e **horário**. O agente de Lucy, que confia no agente de Pete, passou todas as informações já obtidas para que a nova busca fosse realizada. Quase que instantaneamente, um novo resultado foi apresentado. Neste, a clínica era mais próxima e em horários mais cedo. Porém, haviam dois avisos. O primeiro dizia que Pete teria que desmarcar um **compromisso** que estava definido como de **pouca importância**. O segundo dizia que o especialista não estava incluído na lista do plano de saúde como fisioterapeuta, mas que isto foi verificado de outras formas pelo agente. Ao lado da mensagem o agente apresentava o link “details”, caso Pete desejasse verificar como ele concluiu que aquele profissional era um fisioterapeuta, mesmo não estando cadastrado como tal na lista do plano de saúde (BERNERS-LEE, HENDLER e LASSILA, 2001).

Embora o grau de autonomia apresentado neste exemplo ainda não seja encontrado nas aplicações atuais, já são notórios os avanços em diversas áreas que podem torná-lo realidade. É certo que os agentes de Pete e Lucy não executam suas tarefas na *Web* que existe nos dias de hoje. Os documentos escritos em linguagem natural e formatados através da linguagem HTML não possuem a estruturação necessária para que as máquinas possam tomar tantas decisões sem a intervenção humana.

Toda esta história é possível graças à *Web Semântica*, cujo objetivo é fornecer um *framework* capaz de permitir que dados sejam compartilhados e reutilizados entre aplicações, empresas e comunidades. O projeto para o seu desenvolvimento é um esforço colaborativo coordenado pelo W3C e conta com a participação de um grande número de empresas e pesquisadores. “A *Web Semântica* é uma visão: a idéia de ter dados na *Web* definidos e ligados de forma que eles podem ser usados por máquinas não apenas para apresentação, mas também para automação, integração e reutilização dos mesmos entre várias aplicações” (SEMANTIC WEB, 2004).

O principal objetivo da *Web Semântica* é permitir que agentes de software possam executar tarefas cada vez mais complexas em nome das pessoas e sem a intervenção das

³ As palavras em negrito também estão destacadas no artigo original e identificam termos cuja semântica, ou significado, foram definidos para o agente através da *Web Semântica*.

mesmas, o que possibilita a ampliação dos serviços já encontrados na *Web*, além da criação de muitos outros, até então não pensados.

A *Web Semântica* fornece uma infra-estrutura que habilitará, não apenas *homepages*, mas também bases de dados, serviços, programas, sensores, dispositivos pessoais e até eletrodomésticos a consumir e a produzir informações para a *Web*. Agentes de software poderão usar estas informações para buscar, filtrar e preparar informações de diferentes formas para ajudar as pessoas na execução das mais diversificadas tarefas (HENDLER, BERNERS-LEE e MILLER, 2002).

Cenários como o apresentado acima, segundo ANTONIOU e HARMELEN (2004), não requerem progressos científicos revolucionários. O desafio principal é mais de tecnologia e engenharia do que científico, uma vez que soluções parciais para todas as partes importantes do problema já existem. No momento, as maiores necessidades estão na área de integração de informações, padronização, desenvolvimento de ferramentas e adoção das mesmas pelos usuários.

2.3.1 Aplicações da *Web Semântica*

Tomando alguns exemplos bem conhecidos, é possível entender como velhos serviços foram completamente remodelados com base nos recursos fornecidos pela *Web*. Também, com estes mesmos exemplos, é possível visualizar como eles ainda podem ser consideravelmente melhorados com os recursos da *Web Semântica*.

2.3.1.1 Comércio Eletrônico

O relacionamento entre clientes e fornecedores é, sem sombra de dúvida, um dos ramos mais influenciados pelo surgimento da *Web*. Os termos comércio eletrônico (*e-commerce*) e negócio eletrônico (*e-business*) são conhecidos praticamente por todas as pessoas que têm acesso a um computador.

O número de usuários na Internet, por si só, é um bom indicativo do potencial destes ramos. Segundo INTERNET (2005), no final do mês de novembro 2005, a quantidade de usuários no mundo com acesso à Internet ultrapassava a barreira dos 970 milhões, atingindo de 60% a 70% da população em muitos países desenvolvidos. No Brasil, o número de pessoas com acesso à Internet em 2005 foi de aproximadamente 22,3 milhões, o que representa 12,3% da população. Além disto, o crescimento do número de usuários, só no Brasil foi de 346% entre 2000 e 2005, o que reforça ainda mais a importância que a

Web possui no mercado atual.

De acordo com SILVESTRE (2006), o comércio eletrônico no Brasil movimentou 2,5 bilhões de reais em 2005. Este número, além de mostrar a força do comércio eletrônico, indica a possibilidade de crescimento que ela ainda pode ter, considerando-se que menos de 25% dos usuários da Internet realizam compras através dela.

Cabe, ainda, destacar que as vendas diretas entre fornecedores e clientes, ou B2C (*business to consumer*) como é conhecido, representam apenas uma pequena parte do montante negociado eletronicamente. Grande parte desta quantia refere-se às transações entre empresas parceiras, o B2B (*business to business*), que, segundo E-COMMERCE.ORG (2005), foi de aproximadamente R\$ 195 bilhões em 2004.

Apesar dos expressivos resultados já alcançados, muitos pesquisadores ainda apontam o comércio eletrônico como um ramo que está longe de atingir todo seu potencial. Segundo FENSEL *et. al.* (2003), a automatização do comércio eletrônico é seriamente prejudicada pelo modo como o conteúdo é apresentado atualmente. O fato dos agentes responsáveis por buscar informações e realizar negócios necessitarem de *crawlers*⁴ e heurísticas para extrair informações de documentos textuais fracamente estruturados é um dos grandes obstáculos a serem superados. Embora a área de processamento de linguagem natural tenha evoluído, nem sempre a informação obtida é suficiente para tomar uma decisão correta.

Como exemplo, pode-se citar um agente ao qual foi designada a tarefa de procurar em diferentes lojas pelo preço de um determinado CD de música. Através de *crawlers*, o agente fará uma busca tendo como parâmetro o título do CD e algumas palavras-chave como “preço” e “valor”. Com as tecnologias atuais é possível, e muito provável, que o agente consiga descobrir o preço do CD nas várias lojas pesquisadas para informar à pessoa interessada qual é a mais barata.

Entretanto, sabe-se que o custo de uma mercadoria comprada via Internet não consiste apenas no preço do produto. Existem ainda o valor do frete e os impostos, dependendo do país onde se efetua a compra. Criar um agente capaz de extrair todas estas informações de um sistema de vendas *on-line*, formado por *homepages* com conteúdo desestruturado, já é uma tarefa significativamente mais complexa, contribuindo para que a qualidade da informação obtida nem sempre seja a desejada. Além disso, o agente tem

⁴ Componente de um mecanismo de busca que percorre a web armazenando URLs e indexando palavras e textos das páginas encontradas.

dificuldade para ir além de encontrar o menor preço e apresentá-lo ao cliente, sendo incapaz de realizar todo o processo sem intervenção humana.

De acordo com EUZENAT (2001), a *Web Semântica* é capaz de solucionar grande parte destes problemas, criando padrões para descrição de produtos, serviços e outros tipos de documentos. Mais importante que a criação destes padrões, é a utilização de ontologias para fazer os mapeamentos entre eles, relacionando diferentes terminologias e possibilitando o desenvolvimento de um comércio eletrônico mais flexível.

2.3.1.2 Ciência Eletrônica e *Grids* de Conhecimento

Da mesma forma que o comércio eletrônico e o negócio eletrônico refletem o processo de globalização e as alterações ocorridas na forma de efetuar transações comerciais, o termo *e-science* (Ciência Eletrônica) denota as mudanças ocorridas na forma de se fazer pesquisa. Segundo EUZENAT (2001), *e-science* indica o uso de recursos eletrônicos, como instrumentos, sensores, bases de dados e computadores, por cientistas que trabalham cooperativamente e de forma distribuída em grandes projetos.

Atualmente, a *e-science* tem como seu maior expoente o projeto do genoma humano. Devido à imensa quantidade de informação manipulada, nem sempre é possível ter em um mesmo laboratório todas as bases de dados necessárias ou os supercomputadores capazes de processar estes dados. Por esta razão, pesquisadores de todas as partes do mundo estão se unindo e compartilhando os dados e os recursos requeridos para a execução de um projeto comum.

Denomina-se *grid* de conhecimento a infra-estrutura capaz de fazer com que a *e-science* seja possível. De acordo com ROURE (2004), os *grids* surgiram como um novo e importante campo que difere da computação distribuída convencional por focar o compartilhamento de recursos em grande escala, aplicações inovadoras e, em alguns casos, a exigência de alta performance.

Para que o compartilhamento de dados e recursos seja possível, estão sendo desenvolvidos *middlewares*⁵, responsáveis por coletar e publicar informações sobre os recursos disponíveis em uma linguagem independente da plataforma sobre a qual ele é executado. O trabalho dos *middlewares* é fundamental para possibilitar a integração de diferentes sistemas. A definição e a padronização dos protocolos para comunicação entre

⁵ Software que media a comunicação entre dois sistemas diferentes através de uma rede de computadores, ocultando as diferenças existentes entre eles.

diferentes *grids* são coordenadas pelo *Global Grid Forum* (www.gridforum.org), de forma semelhante ao trabalho desenvolvido pelo W3C, na década de 90, nos protocolos e linguagens da *Web* (NEWHOUSE *et. al.*, 2002).

Da mesma forma que um *grid* de conhecimento, a *Web Semântica* é considerada uma evolução da *Web* atual. Entretanto, estes não são conceitos ortogonais, sendo a sua fusão plenamente possível através da integração de recursos como *Web Services*, RDF e ontologias aos *grids* existentes. Para NEWHOUSE *et. al* (2002, p. 5), “O desenvolvimento de serviços orientados a ontologias permitirão que serviços compatíveis construam de forma autônoma os grandes e complexos ambientes computacionais distribuídos que constituirão o *grid*”.

2.3.1.3 Gestão do Conhecimento

A importância da gestão do conhecimento para o sucesso de qualquer empreendimento é um consenso entre os estudiosos da área, em razão da grande quantidade de informação disponível e aos avanços das tecnologias de informação e comunicação.

É primordial que uma pessoa em seu ambiente de trabalho saiba tirar proveito dos benefícios das ferramentas automatizadas que visam agilizar o processo de geração, armazenamento, busca, seleção e manutenção de informações, durante o processo de tomada de decisão e gerência de uma organização.

Apesar destas atividades terem sido facilitadas com o advento da *Web* e sua estrutura distribuída, muitos aspectos ainda são falhos e fazem com que os usuários percam um tempo considerável na realização de tarefas que poderiam ser executadas automaticamente, caso a forma com que as informações são organizadas e armazenadas fosse melhorada. De acordo com FENSEL *et. al.* (2001), as ferramentas existentes no mercado para gestão do conhecimento apresentam as seguintes fraquezas:

- Busca de Informação: os algoritmos baseados em palavras-chave utilizados atualmente podem trazer informações irrelevantes ou ignorar alguma informação útil, devido aos termos sinônimos e homônimos;
- Extração de Informação: este processo requer a participação de um humano, pois os agentes automatizados não possuem o conhecimento necessário para extrair informações de documentos textuais e para relacioná-las com informações obtidas de outras fontes.

- Manutenção de Informações: manter as fontes de dados atualizadas, quando elas se tornam muito extensas, requer um grande esforço e tempo disponível devido à falta de estruturação das mesmas;
- Geração automática de documentos: na forma pela qual os dados são organizados e armazenados atualmente, não é possível criar um sistema capaz de processá-los para gerar informações automaticamente.

Estes e outros problemas relacionados à gestão do conhecimento podem ser mais facilmente tratados se os dados forem armazenados de maneira estruturada e com seu significado semântico associado a ele. As tecnologias utilizadas para o desenvolvimento da *Web Semântica*, como metadados e ontologias, são fundamentais para aprimorar este tipo de aplicação.

2.3.1.4 Educação Baseada na *Web*

São inúmeros e bem conhecidos os benefícios que a *Web* trouxe para a EAD. Como descrito por NAGY (2004, p. 5), “Estas novas tecnologias trouxeram novas oportunidades para o aluno não tradicional, assim como para as instituições de ensino tradicionais. Atualmente, quase todos os desenvolvimentos em tecnologia da informação e comunicação disponíveis estão sendo utilizados na EAD (...)”. Todavia, também são inúmeros os problemas enfrentados, tanto pelos alunos participantes quanto pelas pessoas que oferecem este tipo de curso.

Do ponto de vista do aluno, pode-se destacar a dificuldade que o mesmo tem para encontrar um curso com o conteúdo adequado às suas necessidades. Além, ou em decorrência disto, os alunos sentem-se desmotivados para continuar seus estudos devido à forma com que a informação é apresentada e as atividades são propostas e desenvolvidas.

Analisando sob o ponto de vista de quem oferece o serviço, encontra-se a reutilização de conteúdos e vinculação com outros sistemas de EBW como as maiores dificuldades enfrentadas. Segundo NILSSON (2001), estes e outros problemas existentes na EBW podem ser resolvidos, ou pelo menos amenizados, com o desenvolvimento da *Web Semântica*. Ele ainda destaca que a *Web Semântica* permitirá a construção de agentes inteligentes para auxiliar os alunos a encontrarem e utilizarem recursos de aprendizagem globalmente distribuídos e tornará possível a criação de cursos à distância de forma distribuída e colaborativa.

2.4 Considerações

Através destes exemplos, pôde-se observar que já está ocorrendo outra grande transformação relacionada à forma com que os serviços são acessados e executados. No início do século XXI, a *Web Semântica* mudará a maneira pela qual muitas tarefas são realizadas, da mesma forma que a *Web* fez com que as pessoas revissem seus papéis dentro das organizações e estas dentro da sociedade como um todo no final do século passado.

A necessidade de se desenvolver novas formas para representar as informações disponibilizadas na *Web* é evidente. Aliada à possibilidade de desenvolver aplicações com maior autonomia, esta necessidade motiva os pesquisadores a buscar alternativas que permitam a incorporação da *Web Semântica* nas mais variadas áreas do conhecimento, expandindo a estrutura já consolidada da *Web* atual, através da construção de ontologias e anotação dos recursos por meio de metadados.

Os benefícios da *Web Semântica* e suas aplicações previamente apresentadas parecem inalcançáveis para aqueles que ainda não tomaram conhecimento dos projetos que estão em desenvolvimento, ou pelo menos que se pretende desenvolver, para que a *Web Semântica* torne-se uma realidade.

O fato de nenhuma grande aplicação com retorno financeiro significativo ter sido desenvolvida também contribui para que a *Web Semântica* ainda seja vista apenas como um projeto de laboratório, com pouca, ou até mesmo sem aplicação prática. Certamente, as pessoas que pensam desta forma ainda não conhecem os planos de trabalho coordenados pelo W3C e desenvolvidos por pesquisadores do mundo todo, que definem a *Web Semântica* como um projeto de longo prazo a ser desenvolvido em partes.

3. TECNOLOGIAS NECESSÁRIAS PARA A WEB SEMÂNTICA

Este capítulo tem como objetivo apresentar a arquitetura da Web Semântica e descrever as principais tecnologias que a compõem com a finalidade de oferecer uma visão de suas potencialidades e limitações.

3.1 Estrutura da *Web Semântica*

Alcançar os propósitos da *Web Semântica* certamente não é uma tarefa fácil. De acordo com BERNERS-LEE (1999), esta tarefa requer o desenvolvimento de:

- Linguagens para expressar meta-informação sobre documentos que seja processável pelo computador;
- Terminologias padrões sobre os mais diversos domínios;
- Ferramentas e arquiteturas que usem tais linguagens e terminologias para fornecer suporte à busca, acesso, apresentação e manutenção de informações.

Por ser um projeto de longo prazo, a *Web Semântica* foi estruturada em níveis com diferentes funções. Enquanto os níveis inferiores representam a estrutura básica para o transporte de dados, os superiores contemplam definições semânticas que possibilitam o processamento automatizado dos mesmos pelo computador. A figura 2 apresenta a estrutura da *Web Semântica* com as camadas necessárias para que ela se torne realidade e propicie todos os benefícios descritos nos exemplos do capítulo 2.

Além dos níveis da *Web Semântica* que já estão sendo construídos e aperfeiçoados através de incessantes pesquisas, a figura 2 mostra aqueles cuja criação ainda levará algum tempo, tendo em vista que os níveis inferiores devem estar consolidados para que eles possam ser desenvolvidos.

A *Web Semântica* está sendo construída sobre dois grandes pilares, o *Unicode* e o URI. O *Unicode* é um padrão para representar caracteres criado pelo *Unicode Consortium Inc.* Qualquer caractere é identificado por um número de 16 *bits*, possibilitando a

existência de até 65536 caracteres. Este número, que parece ser muito grande quando se considera apenas os idiomas do oeste europeu, é necessário para que se possam representar os símbolos dos alfabetos grego, chinês, japonês, entre outros (UNICODE, 2004).

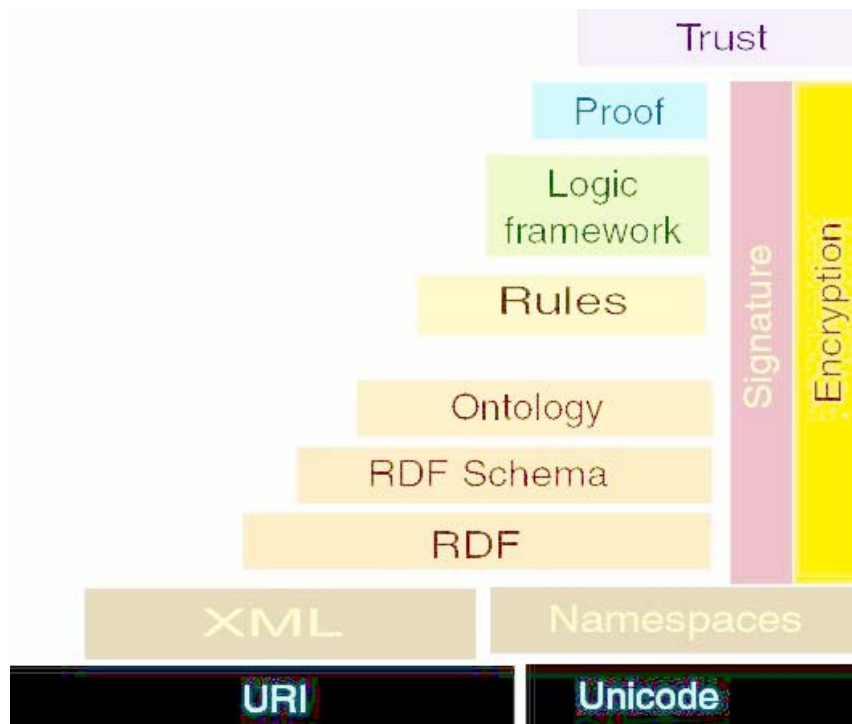


FIGURA 2 - Níveis da *Web Semântica*.
Fonte: BERNERS-LEE (2002)

Muito embora computadores distintos utilizem formatos diferentes para representar o identificador de um símbolo, ele pode ser transferido de uma plataforma para outra, pois não importa a forma pela qual ele é armazenado em uma máquina específica, mas sim o seu significado, que é padrão, tornando o *Unicode* independente de plataforma e sistema operacional.

A independência proporcionada pelo *Unicode* foi fundamental para o desenvolvimento da *Web* e também o é para a *Web Semântica*, pois, desta forma, seus níveis superiores não precisam se preocupar em decodificar dados e transformá-los de um formato de representação para outro antes de enviar ou receber mensagens. Toda informação enviada pela *Web Semântica* deve estar em formato texto, com seus respectivos identificadores numéricos definidos pelo *Unicode*.

O segundo pilar da *Web Semântica* é constituído por Identificadores Uniformes de Recursos. Um URI é formado por um conjunto de caracteres que identifica recursos na *Web* tais como: páginas de texto, imagens, arquivos de áudio, caixas de e-mail e programas. A forma mais comum e conhecida de um URI é a URL (*Uniform Resource*

Locator - Localizador Uniforme de Recursos), que identifica um recurso através de sua localização na rede (W3C, 2001).

Entretanto, é importante registrar informações sobre elementos que não possuem localização de rede ou URL. Outra forma para identificar recursos é através do URN (Nome Uniforme de Recursos), utilizado para definir nomes únicos e persistentes para recursos em um *namespace*, independente de onde ele esteja armazenado. De forma semelhante a uma URL, um URN pode se referir a um elemento acessível via *Web*, como documentos eletrônicos, mas pode, também, referenciar objetos ou pessoas que não estão acessíveis pela rede ou, ainda, conceitos abstratos como “Autor” e “Ano de Publicação” de uma obra.

Com a *Web Semântica*, a tendência é que o URN tenha a mesma importância da URL, pois, agora, além de saber onde está armazenada a informação desejada, precisa-se saber o que ela significa. Assim, quando se pensa em um identificador único de recursos (URI) não se pode pensar apenas em URLs, mas também em URNs que, juntos, possibilitam a identificação de recursos via *Web*.

De acordo com o exposto sobre as tecnologias que servem de base para a *Web Semântica*, percebe-se a preocupação existente com a interoperabilidade, a identificação e o acesso aos recursos. Apesar de ainda ser em um nível bastante primitivo, a interoperabilidade é proporcionada pelo unicode e a identificação e o acesso aos recursos, pelos URIs.

Com as questões básicas resolvidas, faz-se necessário o desenvolvimento dos níveis superiores da *Web Semântica*, agregando a ela as definições sintáticas e semânticas imprescindíveis para atingir o nível de abstração requerido.

3.2 XML

As linguagens para a *Web Semântica* devem fornecer sintaxe e semântica formais para possibilitar o processamento automatizado de seu conteúdo. Também precisam fornecer um vocabulário padrão sobre termos do mundo real para possibilitar que humanos e computadores compartilhem informação e conhecimento. Estas linguagens são necessárias para expressar os conteúdos da *Web Semântica*, que são explorados pelos mecanismos de inferência presentes nos níveis superiores (FENSEL *et al.*, 2003).

O XML é um dos componentes da segunda camada da *Web Semântica* e se constitui na linguagem base da mesma. Ele tem como principal objetivo fornecer meios para

descrever a sintaxe e as definições semânticas básicas dos documentos acessados, tornando-os independentes de aplicação. As definições semânticas mais complexas e os métodos de inferência que manipulam os dados representados nestas linguagens estão presentes nas linguagens das camadas superiores, cuja construção está baseada na linguagem XML.

O XML, assim como o HTML, é uma linguagem de marcação, pois permite a definição de como o conteúdo deve ser interpretado. Entretanto, o XML não se preocupa com a forma de apresentação das informações, sua tarefa é definir seus significados, diferentemente da linguagem precursora que popularizou a *Web*. Para isto, o XML permite a criação de extensões específicas, de acordo com o domínio e a tarefa em questão.

Um documento XML é um arquivo texto que contém dados e metadados. Contudo, apesar de também ser baseado em marcações (*tags*), o XML não pode ser considerado uma linguagem de marcação, já que não traz um conjunto de *tags* pré-definidas. Na verdade, o XML é um conjunto de regras para a criação de linguagens de marcação semântica em um domínio específico (RAY, 2001). A possibilidade de definir o significado das informações através das *tags*, aliada à flexibilidade para criar *tags* de acordo com a necessidade do usuário e o domínio em questão, tornaram o XML a linguagem essencial para a *Web Semântica*.

O XML foi elaborado para ser uma forma simples de enviar documentos pela *Web*. Ele permite a qualquer pessoa projetar seus próprios formatos de documentos, contendo marcações que definam os seus significados, e escrever documentos naquele formato, que podem ser processados automaticamente. Como os arquivos XML são salvos no formato texto e codificados através do Unicode, eles podem ser lidos e manipulados em qualquer plataforma computacional, garantindo, assim, a interoperabilidade requerida pela *Web Semântica*.

Segundo DACONTA, OBRST e SMITH (2003), o sucesso da linguagem XML deve-se a:

- Possibilidade de criar documentos e dados independentes de aplicação;
- Existência de uma sintaxe padrão para definição de metadados;
- Capacidade de definição de uma estrutura padrão para documentos e dados.

Sua sintaxe padrão e aberta contribuiu para que o XML se tornasse uma tecnologia indispensável para a *Web Semântica*, pois, com as descrições verbais do significado dos dados, um documento XML pode ser lido e entendido por todos e não apenas pela pessoa

ou aplicação que o gerou.

Para tornar clara a diferença entre as linguagens HTML e XML, toma-se como exemplo uma página que apresenta informações sobre livros de um determinado assunto, cujo resultado visualizado pelo usuário é mostrado na figura 3. Caso esta página tenha sido desenvolvida com a linguagem HTML, o código gerado é semelhante ao do quadro 1, onde é possível identificar as *tags* responsáveis pela formatação do conteúdo na tela, tais como “font”, “b” e “br”. Além de prejudicar a visibilidade do documento, este tipo de marcação não agrega nenhuma informação sobre o significado do conteúdo da página.



FIGURA 3 - Exemplo de uma página para apresentação de livros.

QUADRO 1 - Código HTML da figura 3.

```

1. <html>
2. <head>
3.   <title>Livros sobre Web Semântica</title>
4. </head>
5. <body>
6.   <font face="Arial"><b>Livros sobre Web Semântica</b><br>
7.   <font size="2">Michael Daconta</font><br>
8.   <b><font color="#0000FF">The Semantic Web</font></b>
9.   <font size="2">2003<br>
10.  Dieter Fensel</font><br>
11. <b><font color="#0000FF">Spinning The Semantic Web</font></b>
12. <font size="2">2003</font> <br>
13. </body>
14. </html>

```

Com os dados organizados desta forma, a única alternativa para uma pessoa que deseja desenvolver um sistema de busca de materiais sobre *Web Semântica* é, após um

grande esforço de programação, elaborar um algoritmo de busca em arquivos por palavras-chave. Esta pessoa pode ter como retorno o endereço desta página (e de muitas outras desnecessárias), mas precisa analisar pessoalmente cada uma delas, já que o sistema não é capaz de processar de forma mais eficiente este tipo de arquivo, pois não sabe o significado de palavras como “Fensel”, “Daconta” e “Spinning”.

A mesma página da figura 3 pode ser gerada a partir de um documento XML, com código semelhante ao do quadro 2. Neste exemplo, podem ser observadas as semelhanças e diferenças entre o HTML e o XML. Inicialmente, como semelhança, cabe destacar o fato de ambos serem baseados em *tags*. Entretanto, as diferenças ficam bastante claras quando são analisados os conteúdos destas *tags*. Enquanto na linguagem HTML elas são pré-definidas e têm o papel de indicar o tipo de fonte utilizada, tamanho de letra, posicionamento e outras questões visuais, no XML seu objetivo é descrever o significado da informação e são definidas pelo próprio usuário.

QUADRO 2 - Código XML da figura 3.

```

1.<?xml version="1.0"?>
2.<acervo>
3.  <assunto>Livros sobre Web Semântica</assunto>
4.  <livro ID="1">
5.    <autor>Michael Daconta</autor>
6.    <titulo>The Semantic Web</titulo>
7.    <ano>2003</ano>
8.  </livro>
9.  <livro ID="2">
10.   <autor>Dieter Fensel</autor>
11.   <titulo>Spinning The Semantic Web</titulo>
12.   <ano>2003</ano>
13. </livro>
14. </acervo>

```

Retomando o exemplo anterior, em que uma pessoa pretende desenvolver um programa para buscar informações sobre *Web Semântica*, percebe-se que a complexidade de programação é sensivelmente reduzida, já que para cada informação apresentada na página existe uma *tag* indicando o seu significado.

Desta forma, para este simples arquivo é possível criar um sistema que efetue buscas por títulos de livros, autores e datas de publicação com um grau de autonomia e precisão muito maior do que é possível fazer com arquivos como o apresentado no quadro 1, pois,

neste caso, as palavras “Fensel”, “Daconta” e “*Spinning*” estão associadas às *tags* que definem o seu significado.

É importante destacar que, por não ser uma linguagem voltada à apresentação de dados, o XML por si só não é capaz de transformar o código do quadro 2 na *homepage* mostrada na figura 3. Para tanto, faz-se necessário o uso de folhas de estilo para formatar a apresentação das informações. Contudo, vale destacar que isto não afeta a clareza do documento, considerando-se que todas as definições ligadas à apresentação ficam isoladas no arquivo correspondente à folha de estilo. Outra vantagem de organizar os dados em um documento XML diz respeito ao fato de eles poderem ser apresentados de forma diferente, bastando que se troque a folha de estilo, sem alterar nenhuma informação no documento original. Maiores informações sobre a utilização de XML com folhas de estilo podem ser obtidas em RAY (2001) e HOLZNER (2001).

3.2.1 Estrutura de um Documento XML

Conforme apresentado no quadro 2, os documentos XML são formados por conteúdos e por *tags* criadas pela pessoa que gerou o arquivo. O processo de criação de um documento XML, contudo, não é tão flexível quanto parece, pois existem regras que devem ser respeitadas para que o mesmo seja considerado bem formado e válido. De acordo com HOLZNER (2001), para que um documento seja considerado bem formado ele deve:

- Conter um prólogo;
- Conter um elemento raiz;
- Opcionalmente conter uma miscelânea de dados;
- E respeitar as regras sintáticas definidas na especificação XML 1.0.

No quadro 3, apresenta-se a constituição de um documento XML.

QUADRO 3 - Exemplo de um documento XML

```
1: <?xml version="1.0"?>
2: <livro ID="1">
3:   <titulo>The Semantic Web</titulo>
4:   <autor> Michael Daconta</autor>
5:   <ano>2003</ano>
6: </livro>
```


O prólogo deve conter, pelo menos, a declaração de que aquele arquivo é um documento XML, para que os processadores possam verificar o tipo dos dados armazenados no arquivo. Isto é feito através da *tag* `<?xml version="1.0"?>`, presente na linha 1, que, além de indicar o tipo do documento, confirma que ele segue os padrões definidos na especificação XML 1.0 (a única disponível até o momento).

O elemento raiz é aquele que engloba todos os demais elementos do documento e indica de uma forma geral o que ele contém. No exemplo do quadro 3, o elemento raiz é definido pela *tag* `<livro>`, cuja abertura é feita na linha 2 e o fechamento na linha 6, indicando que o documento contém dados sobre um livro. É importante destacar que nenhum elemento pode ser colocado fora do elemento raiz, caso isto aconteça, o documento não é mais considerado bem formado.

O elemento raiz pode, e geralmente deve, conter outros elementos e atributos que descrevem o conteúdo do arquivo. No quadro 3, o livro possui o atributo ID, cujo valor é determinado na própria *tag* de abertura deste elemento, e três outros elementos que o descrevem: título, autor e ano de publicação. O conteúdo destas informações é definido entre as *tags* de abertura e fechamento `<título>` e `</título>`, `<autor>` e `</autor>` e `<ano>` e `</ano>`, respectivamente.

Respeitar as regras sintáticas, entre outras coisas, significa criar *tags* válidas e utilizá-las de forma correta, ou seja, toda *tag* de abertura deve possuir sua correspondente para fechamento (com exceção de elementos definidos como vazios). Pode-se perceber que o documento do quadro 3 respeita este aspecto, pois todos os elementos possuem as *tags* de abertura e de fechamento.

O XML, ao contrário do HTML, não é complacente com erros na criação do documento. Um analisador XML não aceita que uma *tag* aberta não seja fechada ou vice-versa, pois como o seu objetivo é permitir a estruturação de informações para serem processadas de forma automatizada, não se pode tolerar erros ou ambigüidades no documento, como ocorre em páginas HTML criadas para serem lidas por humanos.

Apesar de simples, o XML constitui-se em poderosa ferramenta para representar, estruturar e definir o significado de conteúdos. Embora as tarefas de criar e analisar documentos XML pareçam árduas, e realmente são, quando executadas através de um editor de textos simples, atualmente elas são facilitadas por editores com funções que automatizam o processo. A figura 4 apresenta os dados armazenados no documento do quadro 2, utilizando o software *Stylus Studio*.

Element	#text	id	titulo	autor	ano	editora
assunto	Web Semântica					
livro		1	The Semantic Web	Michael Daconta	2003	Wiley
livro		2	Sppining The Semantic Web	Dieter Fensel et al	2003	MITPress

FIGURA 4 - Visualização dos dados de um arquivo XML através do Stylus Studio

Além de possibilitar a visualização dos dados já existentes no arquivo, ele permite a inserção de novas informações de forma semelhante a um banco de dados, gerando o código XML correspondente ao novo conteúdo armazenado. Na parte inferior esquerda da tela é possível identificar os botões “Grid”, que está habilitado, “Tree”, “Schema” e “Text”. Através deles, o usuário pode alternar a forma de visualização dos dados, de acordo com o seu interesse.

Verificar se um documento XML é bem formado constitui apenas uma das etapas de análise, pois nem sempre um documento bem formado é válido para ser processado por alguma aplicação. Pode ser tomado como exemplo o código do quadro 3, que apesar de bem formado não é válido para uma aplicação que faz busca por livros a partir do seu título, autor, ano de publicação ou editora, pois, neste caso, falta no documento a informação da editora responsável pela publicação do livro.

Para evitar que aplicações tenham problemas para processar diferentes documentos XML e precisem se preocupar em analisá-los previamente para verificar se eles atendem a determinadas especificações, criou-se o conceito de validação de documentos.

3.2.2 Validando Documentos XML

A validação é um passo posterior à verificação da boa formação do arquivo. Ela, que nem sempre é executada, serve para comparar o arquivo em questão com um modelo previamente estabelecido. Desta forma, as aplicações que utilizam os dados do documento XML não precisam verificar se ele possui todos os campos necessários e se eles estão estruturados corretamente. Basta que as aplicações também conheçam o modelo utilizado no desenvolvimento do documento e façam uso de um *parser* para validação de

documentos XML (comparação entre o modelo e o documento), reduzindo o trabalho do programador.

Para validar um documento XML, pode-se fazer uso de uma Definição de Tipo de Documento (DTD – *Data Type Definition*) ou de um esquema XML. Estes recursos são fundamentais para que diferentes organizações possam trocar informações via documentos XML, por permitirem a criação de modelos comuns, utilizados com o intuito de padronizar o conteúdo trocado.

Tomando como base, mais uma vez, um documento utilizado para descrever livros sobre um determinado assunto, verifica-se que tanto o código do quadro 2 quanto o do quadro 4 são bem formados.

Todavia, eles apresentam algumas diferenças como a ordem dos campos (título e autor), a data de publicação, que é especificada em um documento através da *tag* <data> e no outro é feito pela *tag* <ano>, e, por fim, a editora responsável pela publicação, que não é descrita em um dos documentos.

Supondo que o documento apresentado no quadro 2 seja utilizado por uma biblioteca para descrever seus livros e o documento do quadro 4 o padrão adotado por outra biblioteca, precisa-se de um trabalho de programação extra para que o sistema de uma delas entenda os dados gerados pelo sistema da outra. Visando eliminar este problema criou-se o conceito de modelo de documento e validação de documentos.

QUADRO 4 - Outro exemplo de documento XML.

```
1.<?xml version="1.0"?>
2.<acervo>
3.  <assunto>Livros sobre Web Semântica</assunto>
4.  <livro id="1">
5.    <titulo>The Semantic Web</titulo>
6.    <autor>Michael Daconta</autor>
7.    <data>2003</data>
8.    <editora>Wiley</editora>
9.  </livro>
10. <livro id="2">
11.  <titulo>Spinning The Semantic Web</titulo>
12.  <autor>Dieter Fensel</autor>
13.  <data>2003</data>
14.  <editora>MIT Press</editora>
15. </livro>
16.</acervo>
```

De acordo com RAY (2001, p. 148), um modelo de documento “define os documentos que podem ser produzidos com uma linguagem (...) o modelo de documento determina quais documentos estão de acordo com a linguagem”.

Os DTDs constituem-se na primeira forma utilizada para construção de modelos e são criados através de regras declarativas. Os esquemas XML, segundo HOLZNER (2001), foram criados por um comitê do W3C, após ouvir intensas reclamações sobre a dificuldade dos DTDs, e são definidos através de exemplos. Muito embora a dificuldade de uso não tenha sido reduzida, os esquemas XML ganharam muito em representatividade devido à inclusão de características como: definição dos tipos de dados dos conteúdos; mecanismos de herança de outros esquemas; especificação do número mínimo e máximo de vezes que um elemento pode aparecer, entre muitas outras, tornando-se a forma mais utilizada para validar documentos.

3.2.3 Esquemas XML (XML Schemas)

Um esquema XML é semelhante a um esquema de um banco de dados, pois através dele é possível definir o nome e o tipo de dados das colunas, além da forma como estes elementos devem estar organizados dentro do documento. Os esquemas podem ser utilizados para gerar e validar documentos de acordo com sua sintaxe, conforme ilustra a figura 5.

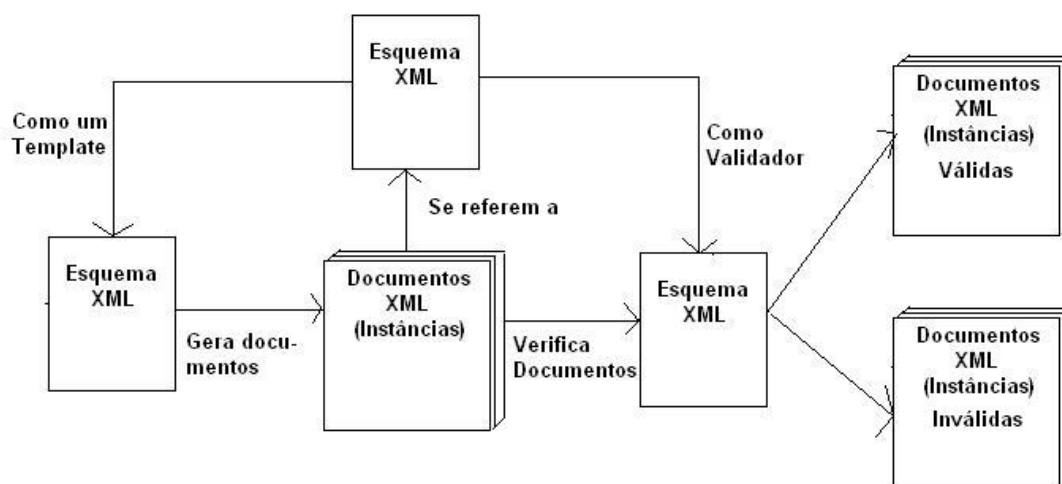


FIGURA 5 - Relação entre um esquema e documentos XML
Fonte: DACONTA, OBRST e SMITH (2003)

Nesta figura, pode-se observar que um único esquema XML pode ser utilizado de várias maneiras. Ele pode servir, por exemplo, como um *template*, através do qual

aplicações geram documentos seguindo as regras por ele definidas. Estes documentos podem, posteriormente, ser validados por outras aplicações que utilizam seus dados, com base nas regras definidas no mesmo esquema utilizado na geração dos documentos no início do processo.

A sintaxe de um esquema XML é igual à de um documento XML, o que permite que um mesmo *parser* seja utilizado para validá-los. O quadro 5 apresenta os componentes básicos de um esquema XML.

QUADRO 5 - Componentes básicos de um esquema XML

<pre> 1. <?xml version="1.0"?> 2. <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"> 3. </xsd:schema> </pre>

Na linha 1, encontra-se a *tag* responsável por definir que este é um documento XML. A linha 2 indica que este documento contém a definição de um esquema XML, de acordo com as definições padrões feitas no *namespace*⁶ identificado pelo URI especificado, o qual contém a definição de termos como *string*, *integer*, *long*, *boolean*, *element*, *complexType*, *attribute* e *sequence*, utilizados durante a criação de um esquema XML. Por fim, a linha 3 indica o término deste esquema.

O quadro 6 apresenta um esquema XML utilizado para validar documentos semelhantes ao do quadro 4. Além das linhas básicas apresentadas no quadro 5, este código utiliza os seguintes elementos para definir a sintaxe dos documentos:

- *xsd:element* - indica a definição de um novo elemento. Este pode ser de um tipo simples ou complexo e conter outros elementos e atributos. No documento do quadro 6, o elemento *xsd:element* foi utilizado na linha 4 para definir o elemento raiz, neste caso *Acervo*, e os elementos *Assunto*, *Livro*, *Titulo*, *Ano*, *Autor* e *Editora* nas linhas 7, 8, 11, 12, 13 e 14 respectivamente. Os elementos simples têm seu tipo definido pelo atributo *type*, como é feito em *Assunto* (linha 7), *Titulo* (linha 11), *Ano* (linha 12), *Autor* (linha 13) e *Editora* (linha 14).
- *xsd:complexType* e *xsd:sequence* – definem que um elemento é composto por outros elementos, como feito neste caso para o *Acervo* (linhas 5 e 6), que possui um assunto e um conjunto de elementos do tipo *Livro*, um tipo também complexo, composto por um título, autor, ano de publicação e

editor (linhas 9 e 10).

- `xsd:attribute` – utilizado para definir os atributos de um elemento. Neste caso, indica que todo livro deve possuir um identificador, conforme especificado na linha 16.

QUADRO 6 - Esquema XML

```

1. <?xml version="1.0"?>
2. <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
3.           elementFormDefault="qualified">
4. <xsd:element name="Acervo">
5. <xsd:complexType>
6. <xsd:sequence>
7.   <xsd:element name="assunto" minOccurs="1" type="xsd:string"/>
8.   <xsd:element name="livro" maxOccurs="unbounded">
9.     <xsd:complexType>
10.      <xsd:sequence>
11.        <xsd:element name="titulo" type="xsd:string"/>
12.        <xsd:element name="autor" type="xsd:string"/>
13.        <xsd:element name="ano" type="xsd:integer"/>
14.        <xsd:element name="editora" type="xsd:string"/>
15.      </xsd:sequence>
16.      <xsd:attribute name="id" use="required" type="xsd:integer"/>
17.    </xsd:complexType>
18.  </xsd:element>
19. </xsd:sequence>
20. </xsd:complexType>
21. </xsd:element>
22.</xsd:schema>

```

O código do quadro 6 mostra que todo documento do tipo acervo deve possuir um e apenas um assunto seguido de um conjunto de livros. Os livros, por sua vez, possuem um atributo ID (identificador) e os elementos *Título*, *Autor*, *Ano* e *Editora*. O número de livros cadastrados não é limitado pelo documento.

O código apresentado no quadro 7, apesar de bem formado, é considerado inválido por violar duas regras do esquema apresentado. Primeiro porque o elemento *Acervo* possui dois elementos *Assunto* (um para cada livro, definidos nas linhas 3 e 9) e o esquema permite que seja definido apenas um assunto para todos os livros. Além disso, não foi

⁶ A definição e criação de *namespaces* é abordada na seção 3.2.4.

especificado neste documento o elemento *Editora* que, segundo o esquema, é parte obrigatória da descrição de um *Livro*.

QUADRO 7 - Código XML inválido

```
1.<?xml version="1.0"?>
2.<acervo>
3.  <assunto>Livros sobre Web Semântica</assunto>
4.  <livro id="1">
5.    <titulo>The Semantic Web</titulo>
6.    <autor>Michael Daconta</autor>
7.    <data>2003</data>
8.  </livro>
9.  <assunto>Livros sobre XML</assunto>
10. <livro id="2">
11.   <titulo>Spinning The Semantic Web</titulo>
12.   <autor>Dieter Fensel</autor>
13.   <data>2003</data>
14. </livro>
15.</acervo>
```

Além destes, muitos outros elementos, atributos e tipos de dados simples podem ser utilizados para criar esquemas XML com um grau de complexidade, e conseqüentemente de detalhamento, bem maior que o previamente apresentado. Uma lista completa com estas informações pode ser obtida em HOLZNER (2001) e as recomendações do W3C estão definidas nas seguintes referências FALLSIDE e WALMSLEY (2004), THOMPSON *et al* (2004) e BIRON e MALHOTRA (2004).

3.2.4 Namespaces

A liberdade oferecida pela linguagem XML em relação à nomenclatura das *tags* é, ao mesmo tempo, útil e perigosa. Útil, por permitir a cada pessoa que projeta um documento XML criar o nome de suas próprias *tags* e, perigosa, devido aos conflitos que podem ocorrer quando se deseja utilizar elementos ou atributos definidos em diferentes modelos em um mesmo documento. É possível, e até provável, em se tratando de documentos relacionados a um mesmo assunto, que eles tenham elementos ou atributos com o mesmo nome, porém, com construções e significados diferentes. Portanto, é importante que os documentos tenham nomes universais, cujo escopo vai além do conteúdo do próprio

documento.

Para evitar este tipo de conflito desenvolveu-se o conceito de *namespace* que, segundo DACONTA, OBRST e SMITH (2003), caracteriza-se como um mecanismo simples para criar nomes únicos para os elementos e atributos de um documento XML. Este é um requisito fundamental para o desenvolvimento da *Web Semântica*, pois se torna imprescindível a união de informações de diferentes documentos para se chegar aos resultados desejados. Assim, justifica-se a importância dos *namespaces*, que aparecem como uma das tecnologias base, ao lado do XML, para permitir que as informações sejam corretamente representadas, trocadas, unidas e processadas de forma autônoma pelos computadores.

Tecnicamente, um *namespace* é uma coleção de nomes, identificado por um URI, que é utilizado em documentos XML como tipos de elementos e nomes de atributos (BRAY, HOLLANDER e LAYMAN, 1999).

Como é impossível nomear um elemento ou atributo qualquer e ter a certeza de que nenhuma outra pessoa no mundo irá utilizar o mesmo nome para outro elemento, optou-se por criar um prefixo único para os elementos. Entretanto, até aí o problema apenas foi passado do nome do elemento para o nome do prefixo, considerando-se que ninguém poderia garantir a inexistência de dois prefixos idênticos, com elementos ou atributos iguais. Para contornar este problema estão sendo utilizados URIs válidos como prefixos, que indicam onde está o *namespace* com definição daquele elemento.

As definições feitas em *namespaces* aparecem em documentos XML como nomes qualificados (QNome). Um QNome é constituído de duas partes: o prefixo do *namespace* e a parte local, separados pelo símbolo “:”. O prefixo do *namespace* pode ser o próprio URI referenciado ou uma abreviatura criada no documento onde ele será utilizado.

Para criar o esquema XML do quadro 6 foi usado um *namespace* definido em <http://www.w3.org/2001/XMLSchema>, que contém os elementos utilizados na construção de um esquema XML. Esta associação é feita na linha 2, por meio da palavra reservada *xmlns*. Esta palavra é responsável por associar o *namespace* a uma sigla local, utilizada como prefixo para as *tags* presentes no documento e que já foram definidas no *namespace*. No exemplo apresentado, o prefixo local utilizado para identificar o *namespace* foi “xsd”, o qual foi usado posteriormente em todas as *tags* que referenciam elementos do *namespace* especificado pela URI.

O código `<xsd:complexType>` encontrado no quadro 6 representa um exemplo de *QNome*. A sigla *xsd* identifica o *namespace* e *complexType* o elemento definido neste

namespace. Outra forma de representar este *QNome* é usando a URI correspondente a ele, sem abreviá-la, como segue: *http://www.w3.or/2001/XMLSchema:complexType*.

3.2.5 Considerações sobre XML para a Web Semântica

O XML é a peça fundamental para a representação do conhecimento na *Web Semântica*, tendo em vista as características que permitem o intercâmbio de dados entre aplicações executadas em diferentes plataformas.

Outro ponto que torna o XML indispensável é a liberdade oferecida durante a confecção dos documentos, permitindo a cada pessoa construir suas próprias *tags*. Com este recurso é possível definir o significado dos conteúdos presentes em um documento, processo fundamental para o desenvolvimento de uma estrutura que possua informações processáveis pelos computadores.

Contudo, ainda são muitas as limitações desta linguagem que devem ser eliminadas ou pelo menos atenuadas para que a *Web Semântica* possa atingir o nível de autonomia esperado. Uma delas reside no fato de ser apenas possível anexar palavras aos dados, ou seja, os metadados de um documento XML se constituem de palavras isoladas. Ainda não há como definir sentenças, nem relacionamentos mais complexos sobre estas entidades, devido à ausência de uma sintaxe bem definida e padronizada para a sua construção.

Além disso, nem todas as pessoas utilizam o mesmo termo para definir uma informação, causando, assim, um novo problema. Tomando como exemplo o documento para descrição dos livros, verifica-se que diferentes pessoas podem criar esquemas distintos para definir os mesmos dados, como as *tags* que indicam a data de publicação do livro, ora chamada de *Ano* ora de *Data*. Este fato, facilmente contornado por um humano, pode causar problemas para as aplicações que venham a processar estes documentos.

Com o intuito de resolver tais problemas, novas linguagens e formas para representação do conhecimento foram desenvolvidas e alocadas em níveis superiores da *Web Semântica* para torná-la mais robusta, suprimindo as limitações do XML.

3.3 Resource Description Framework

A camada da *Web Semântica* imediatamente acima da linguagem XML e dos *namespaces* é formada pelo *Framework* para Descrição de Recursos (RDF - *Resource Description Framework*). Segundo BERNERS-LEE (1998a, p.2) “Quando se olha uma possível formulação de uma *Web* universal de declarações semânticas, o princípio de

projeto mínimo requer que ela seja baseada em um modelo comum de grande generalidade. Apenas quando o modelo comum é geral, é que se pode mapear qualquer aplicação para o modelo; e este modelo geral é o RDF”. Esta afirmação ajuda a entender a importância que o RDF possui dentro da estrutura da *Web Semântica*.

Enquanto na *Web* atual os usuários podem fazer ligações entre quaisquer recursos de forma que o computador consiga entender, com a *Web Semântica* será possível “dizer” o que quiser sobre qualquer entidade de uma forma que também será entendida pelo computador (NILSSON, 2001).

Para que isso se torne possível, são necessários um sistema de identificação que seja facilmente processado por um computador e uma linguagem que também seja processável pelas máquinas para representar as sentenças e possibilitar a sua troca entre elas. O primeiro problema é resolvido pelos URIs apresentados na seção 3.1 e, o segundo, pelo RDF.

Segundo MANOLA e MILLER (2004), “RDF é uma linguagem para representar informações sobre recursos na *World Wide Web*. Ela é particularmente necessária para representar metadados sobre recursos *Web*, tais como o título, autor e data de modificação de uma página, direitos autorais e informações sobre licenças de documentos *Web* ou a disponibilidade de algum recurso compartilhado. Entretanto, ao generalizar o conceito de recurso *Web*, permite-se que o RDF seja usado para representar informações sobre entidades que podem ser identificadas na *Web*, mesmo quando elas não possam ser recuperadas através dela”.

O RDF é uma linguagem construída sobre o XML e utilizada para fazer afirmações sobre entidades e documentos como um todo. Enquanto o XML serve para adicionar metadados a partes de um documento, com o RDF pode-se associar metadados ao documento como uma entidade única (DACONTA, OBRST e SMITH, 2003). Ele é baseado na associação de descrições a entidades (recursos) e tem como função representar o conhecimento sobre o relacionamento entre elas. Cada recurso possui uma propriedade, à qual se atribui um valor, sendo que em RDF o recurso é chamado de sujeito, a propriedade, de predicado e o valor, de objeto. Através do RDF é possível criar sentenças que as aplicações possam facilmente processar, mesmo que elas não consigam entender o seu significado.

Dentre outros motivos que tornaram o RDF uma recomendação do W3C para a *Web Semântica*, KLYNE e CARROLL (2004) destacam:

- A capacidade que ele tem de fornecer informações sobre recursos *Web* e os

sistemas que o utilizam (Metadados), tais como avaliações de conteúdo, descrições de capacidades e preferências de privacidade;

- A possibilidade de atender aplicações que necessitam de modelos de informações abertos em vez de restritos, para atividades como agendamento, descrição de processos organizacionais e anotações de recursos *Web*;
- Que o RDF é capaz de fazer pelas informações processáveis por máquina o que a *Web* fez pelo hipertexto: permitir que dados sejam processados fora do ambiente particular em que foram criados;
- O seu poder para combinar informações de diferentes aplicações para chegar a novas informações, permitindo o processamento do conteúdo *Web* por agentes de *software* e fornecendo uma linguagem única para eles.

Por ser um *framework* comum, assim como o XML, o RDF permite que os desenvolvedores utilizem *parsers* e ferramentas disponíveis para fazer a geração e o processamento de documentos RDF, liberando-os desta tarefa e possibilitando que os mesmos possam dedicar mais tempo ao desenvolvimento das partes específicas de sua aplicação.

Para aquelas pessoas que estão iniciando o estudo sobre *Web* Semântica e metadados, pode não estar bem clara a diferença entre XML e RDF e surgir a seguinte dúvida: “Por que eu devo utilizar RDF e não apenas o XML, já que o objetivo de ambas as linguagens é possibilitar a descrição de metadados?”. Segundo BERNERS-LEE (1998b), existe uma diferença muito grande entre as aplicações ou pessoas que querem apenas consultar documentos e aquelas que desejam extrair o significado dos documentos para então consultá-los. Embora para as aplicações e pessoas da primeira categoria apenas a linguagem XML seja suficiente, para as da segunda, é necessário um poder de descrição maior, que apenas o RDF consegue fornecer.

O mesmo autor destaca que é possível mapear documentos XML para grafos semânticos, mas que esta é uma tarefa perigosa porque:

- O mapeamento é de muitos para um;
- É preciso um esquema para saber o que o mapeamento significa;
- A expressão necessária para consultar uma árvore XML é muito mais complexa que a expressão para consultar uma árvore RDF.

Por fim, BERNERS-LEE (1998b p.4) conclui que “Não seria bom se nós pudéssemos rotular nossos documentos XML para que quando o analisador os lesse, ele fosse capaz de encontrar asserções (triplas) e distinguir seus sujeitos e objetos para deduzir asserções lógicas (...) ? Isto é o que o RDF faz”.

3.3.1 Representação de Dados em RDF

A sintaxe abstrata para criação de documentos RDF é uma recomendação do W3C e pode ser encontrada em KLYNE e CARROLL (2004). Para ilustrar o modelo de representação de dados do RDF, será utilizado um exemplo extraído de SANTACHÈ (2003), para quem o entendimento do RDF passa pelo entendimento de cada uma das letras que compõem sua sigla.

O termo “Descrição” é a base para a representação de informações em RDF. Para os humanos, a forma mais comum para descrever algo é através de uma sentença em um dado idioma. Como exemplo, considerar-se-á a seguinte sentença: “O livro Desvendando XML, publicado pela editora Campus no ano de 2001, foi escrito por Steven Holzner”. Esta sentença, apesar de ser facilmente entendida por qualquer pessoa alfabetizada na Língua Portuguesa, provavelmente causará alguma dificuldade a um programa de computador que queira extrair informações da mesma. Portanto, devem-se buscar estruturas mais formais para representar estas informações, de forma a facilitar a sua manipulação e entendimento pelo computador. Isto pode ser feito através de uma ficha, como a apresentada no quadro 8.

QUADRO 8 - Ficha com informações de um livro.

Livro: Desvendando XML	
Autor	Steven Holzner
Editora	Campus
Ano de Publicação	2001

Esta ficha pode ser vista como uma descrição formada por um recurso, que é o elemento que está sendo fichado, e pelo corpo da descrição, constituído de várias declarações, que são os pares propriedade-valor associados ao recurso. Neste exemplo, existem três declarações, como mostra o quadro 9.

QUADRO 9 - Declarações sobre o livro apresentado no quadro 8

Sujeito ou Recurso	Livro: Desvendando XML		
Predicados ou Propriedades	Autor	Editora	Ano de Publicação
Objetos ou Valores	Steven Holzner	Campus	2001

As triplas (sujeito, predicado, objeto) ou (recurso, propriedade, valor) definem uma declaração RDF e constituem o bloco fundamental para a construção de qualquer documento nesta linguagem. Uma forma simples de visualizar declarações em RDF é através de um Grafo Direcionado Rotulado, onde os sujeitos e objetos são vértices rotulados e os predicados são arestas rotuladas que apontam do sujeito para o objeto (MANOLA e MILLER, 2004). A figura 6 apresenta o grafo correspondente às declarações do quadro 9.

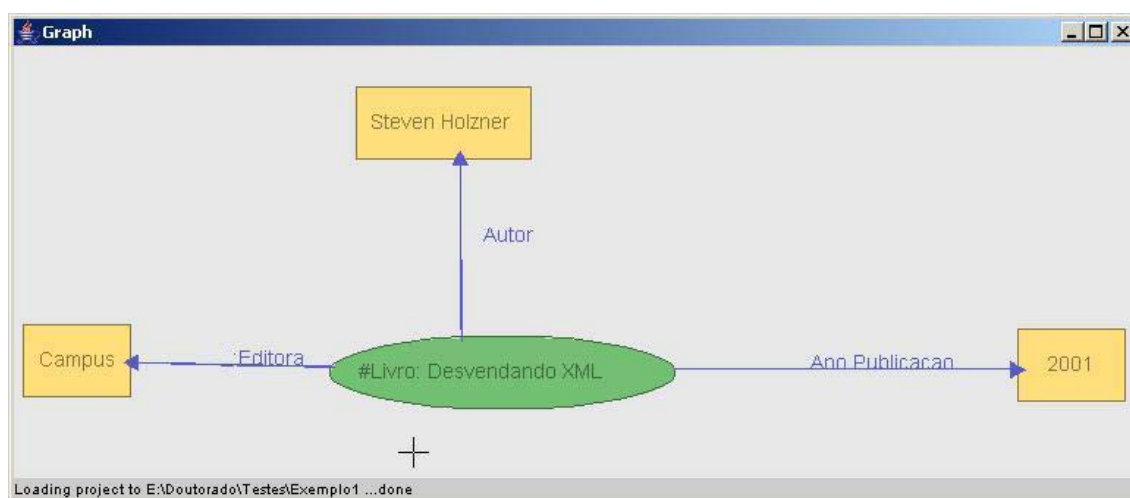


FIGURA 6 - Representação por grafos de declarações em RDF.

Após a compreensão do que é uma declaração, deve-se entender o conceito e o papel de um “Recurso” dentro do RDF. O recurso é o sujeito da sentença, ele indica sobre quem é aquela sentença e o que está sendo descrito pela propriedade e por seu valor.

O uso de literais na definição do sujeito de uma declaração é útil para facilitar o seu entendimento, contudo, ele não o é quando se pensa no objetivo da *Web Semântica*, pois permite que diferentes pessoas usem diferentes nomes para indicar os mesmos recursos. Para evitar problemas como este, todo recurso utilizado em uma declaração RDF deve ter uma identificação única, sendo os URIs a forma mais fácil de fazer isto, como mostra a figura 7.

Devido à forma mais conhecida de um URI ser a URL, as pessoas associam os recursos apenas às entidades que podem ser acessadas via *Web*, como *homepages*, *e-mails*, ou arquivos em FTPs (*File Transfer Protocol* – Protocolo para Transferência de Arquivos). Porém, nem sempre isto é verdadeiro, pois a *homepage* com as informações sobre o livro “Desvendando XML” pode não existir, e uma biblioteca criar um URI da mesma forma.

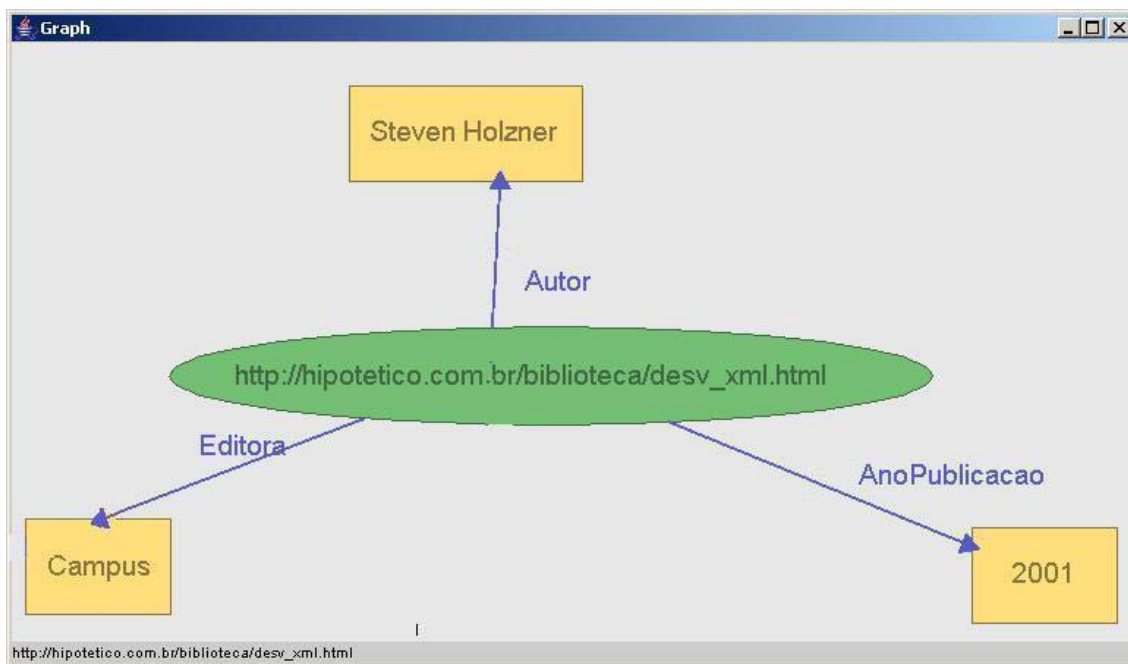


FIGURA 7 - Declaração RDF com URIs representando recursos.

Finalizando o estudo sobre cada um dos componentes da sigla RDF, chega-se ao termo *framework*⁷, que surge da utilização de URIs para identificar predicados e, opcionalmente, objetos, possibilitando a criação de uma estrutura para representação das informações cujas vantagens da unicidade das descrições dos recursos são expandidas para os predicados e objetos. No caso dos predicados, os URIs referenciam propriedades descritas em vocabulários (*namespaces*). Através destes vocabulários, comunidades estabelecem termos padrões para a descrição de recursos relacionados a sua área de interesse (SANTACHÈ, 2003).

Um exemplo é o vocabulário criado pelo *Dublin Core* (DUBLINCORE, 2005), o qual padroniza uma série de predicados que definem, entre outras propriedades, a que caracteriza o autor de um documento, através do URI <http://purl.org/dc/elements/1.1/creator> e sua editora <http://purl.org/dc/elements/1.1/publisher>. A flexibilidade da *Web Semântica* pode ser notada neste momento, ao considerar que ainda não exista um URI que defina o predicado *Ano de publicação*. Este problema pode ser facilmente resolvido, pois a pessoa interessada pode criar seu próprio esquema e definir os termos de seu interesse, sem a necessidade de se submeter a um órgão controlador. Neste caso, pode ser criado um vocabulário identificado por <http://hipotetico.com.br/Exemplo/>, que define o predicado *AnoPublicacao*, através do URI <http://hipotetico.com.br/Exemplo/>

⁷ Uma estrutura que dá forma e suporte para algo (OXFORD, 1995).

AnoPublicacao.

Ao substituir os predicados *Autor*, *Editora* e *Ano de Publicação* da figura 7 pelos correspondentes definidos nos esquemas <http://purl.org/dc/elements/1.1/> e <http://hipotetico.com.br/exemplo/>, chega-se a estrutura apresentada na figura 8.

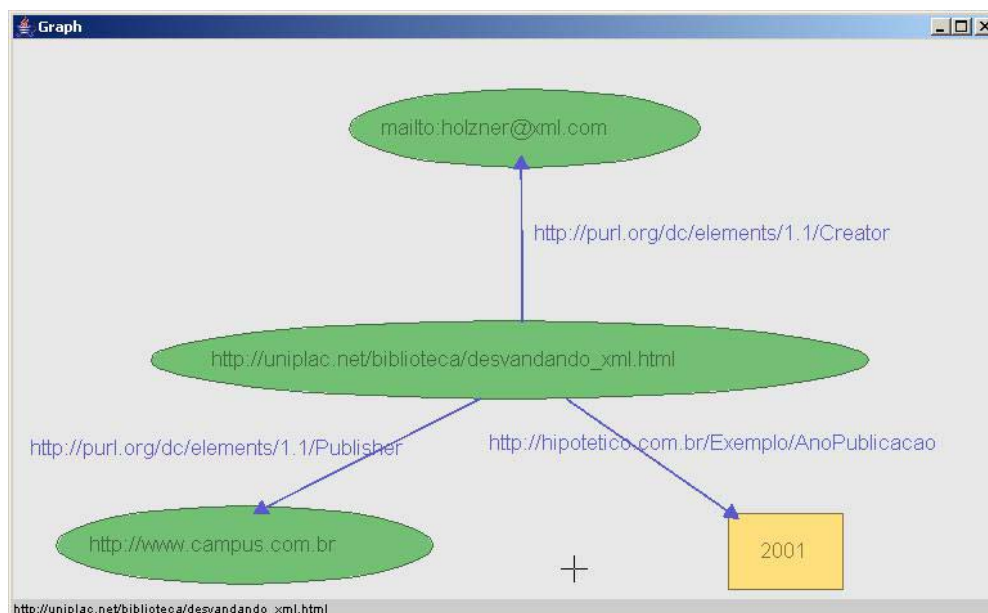


FIGURA 8 - Declaração RDF com URIs representando recursos, predicados e objetos.

De acordo com MANOLA e MILLER (2004), duas vantagens surgem quando se utilizam URIs como predicados para sentenças RDF. A primeira é que eles evitam que o mesmo predicado seja utilizado com significados diferentes. Por exemplo, o termo *nome* pode ser utilizado por uma aplicação para definir o nome de uma pessoa e por outra aplicação para definir o nome de um programa em execução. Um terceiro programa que unisse as duas descrições RDF teria dificuldade para definir o significado do predicado *nome*.

Ao associar um URI ao predicado, cria-se uma identificação “única” para ele. O responsável pela criação da primeira declaração pode estabelecer um vocabulário chamado <http://hipotetico.com.br/Pessoas/>, com termos para descrever pessoas, sendo o predicado *nome* descrito por <http://hipotetico.com.br/Pessoas/Nome>. Já a pessoa que criou o segundo documento pode definir um vocabulário com termos sobre linguagens de programação, tal como <http://hipotetico2.com.br/Program/>, e definir <http://hipotetico2.com.br/Program/Nome>. O programa que fizer a união destas declarações pode, facilmente, identificar que os predicados *nome* têm significados diferentes, mesmo que ele não saiba exatamente qual seja este significado.

Outra vantagem obtida ao utilizar URIs para definir predicados é que eles podem ser tratados como sujeitos de outra declaração, inclusive tendo informações adicionais associadas. “A utilização de URI como sujeitos, predicados e objetos em sentenças RDF possibilita o uso de vocabulários compartilhados na *Web*, desde que as pessoas consigam descobrir vocabulários já desenvolvidos por outras para descrever elementos, refletindo um entendimento compartilhado destes conceitos” (MANOLA e MILLER, 2004, p.15).

A pessoa que criou as declarações da figura 8, por exemplo, ao usar o predicado *Creator* está determinando que a relação existente entre o livro, *Desvendando XML*, e a pessoa cujo e-mail é *holzner@xml.com*, é exatamente a definida pelo *Dublin Core* em <http://purl.org/dc/elements/1.1/Creator>. Qualquer pessoa familiar com este vocabulário, ou que encontre os seus significados na *Web*, sabe claramente o que significa aquela relação, podendo criar um programa que a processe para obter alguma informação. Este mesmo raciocínio é válido para os predicados definidos em *Publisher* e *AnoPublicacao*.

Nesta mesma figura, pode-se observar que não apenas os sujeitos e predicados são definidos como URIs, mas também os objetos, o que permite que os objetos de uma declaração sejam usados como sujeitos em outras declarações, criando uma estrutura de conhecimento que dá sentido ao termo *framework* da sigla RDF. É importante salientar que no modelo RDF apenas os objetos podem ser URIs ou literais, pois tanto os sujeitos quanto os predicados devem, obrigatoriamente, ser definidos através de um URI.

Além disto, vale ressaltar que predicados definidos em diferentes vocabulários são utilizados neste grafo. Além de *Creator* e *Publisher* definidos em <http://purl.org/dc/elements/1.1/>, que está se tornando um vocabulário padrão para definições relacionados a publicações bibliográficas, foi utilizado o predicado *AnoPublicacao*, que pode ter sido definido no vocabulário <http://hipotetico.com.br/Exemplo/>, pela pessoa que está desenvolvendo o sistema. A possibilidade de unir diferentes vocabulários já prontos ou criados pela pessoa interessada é que proporciona ao RDF a flexibilidade que a *Web Semântica* necessita.

Outra forma de representar uma sentença em RDF apresentada por MANOLA e MILLER (2004), é através de uma tripla (sujeito, predicado, objeto). As sentenças da figura 8, transformadas em triplas, apresentam-se conforme descrito no quadro 10.

Contudo, é importante ressaltar que um URI apenas fornece subsídios para a criação de um identificador universal, porém não é uma garantia disto. Por exemplo, ao considerar um endereço de *e-mail* como um URI, subentende-se que ele será um identificador para um único recurso (neste caso uma pessoa), entretanto, este mesmo endereço pode ser

utilizado por duas ou mais pessoas. Este primeiro caso pode ser facilmente resolvido pelo criador do URI, a partir do momento em que ele tem a certeza de que aquele identificador corresponde a um único recurso.

QUADRO 10 - Sentenças RDF apresentadas em triplas

Sentença 1	Sujeito	http://uniplac.net/biblioteca/desvandando_xml.html
	Predicado	http://purl.org/dc/elements/1.1/Publisher
	Objeto	http://www.campus.com.br
Sentença 2	Sujeito	http://uniplac.net/biblioteca/desvandando_xml.html
	Predicado	http://purl.org/dc/elements/1.1/Creator
	Objeto	mailto:holzner@xml.com
Sentença 3	Sujeito	http://uniplac.net/biblioteca/desvandando_xml.html
	Predicado	http://hipotetico.com.br/Exemplo/AnoPublicacao
	Objeto	2001

Um problema um pouco mais complexo reside no fato de existirem dois ou mais identificadores para um mesmo recurso. Por exemplo, uma empresa X cria um identificador para cada um de seus funcionários, da mesma forma que uma universidade Y cria para os seus alunos. Existirão dois identificadores para uma mesma pessoa, caso algum empregado da empresa X estude na universidade Y. Como não existe nenhum mecanismo central de controle para a criação dos identificadores, não há como garantir que cada recurso seja identificado uma única vez. Este problema também pode existir em esquemas que definem predicados, pois diferentes grupos ou pessoas podem usar termos diferentes para definir um mesmo predicado.

Como exemplo, pode-se tomar uma pessoa que não tenha conhecimento do esquema definido pelo *Dublin Core* e queira criar um predicado para definir o autor de um livro. Esta pessoa pode usar o esquema <http://hipotetico.com.br/exemplo>, para definir o predicado <http://hipotetico.com.br/exemplo/autor>. Para uma pessoa que esteja manipulando estes dados, talvez não seja uma tarefa complexa concluir que este predicado tenha o mesmo significado de <http://purl.org/dc/elements/1.1/creator>, mas, para um programa de computador que esteja trabalhando de forma autônoma (que é o principal objetivo da *Web Semântica*), provavelmente é.

A *Web Semântica* buscou as Ontologias na área de Inteligência Artificial para estabelecer as relações entre os diferentes vocabulários, visando contornar este tipo de problema e resolver as ambigüidades existentes entre a criação e identificação de recursos e predicados. Para tornar mais clara a relação entre RDF e os componentes situados logo

acima dele na estrutura da *Web Semântica*, pode-se considerar que:

- O RDF permite afirmar fatos sobre entidades (recursos), através de declarações em triplas, do tipo (sujeito, predicado, objeto);
- O RDF Schema possibilita a criação de vocabulários para definir termos que descrevam recursos e predicados;
- Além de fornecerem mecanismos mais sofisticados para a definição de terminologias padrões, as Ontologias podem ser utilizadas para descrever o relacionamento entre diferentes vocabulários.

3.3.2 Outras Características do RDF

Além da possibilidade de representar conceitos simples, como os vistos nos exemplos anteriores, o RDF fornece algumas capacidades adicionais. A primeira delas é a presença de um nó complementar para representar propriedades estruturadas. Como os relacionamentos descritos em RDF são sempre binários, qualquer relação n-ária deve ser quebrada em vários relacionamentos binários. Em geral, isto é feito pela inclusão de um nó vazio no grafo, como mostra a figura 9, que descreve o relacionamento entre uma pessoa e seu endereço.

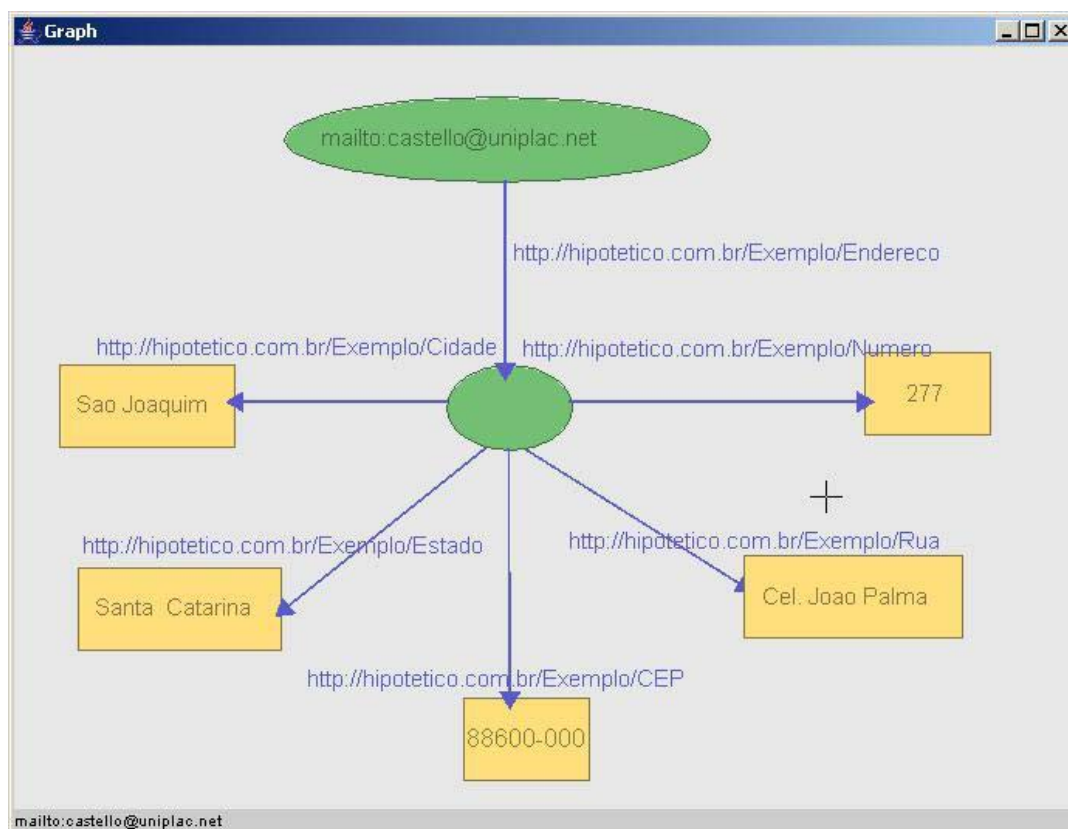


FIGURA 9 - Nó em branco para representar relacionamentos n-ários.

Considerando que o endereço de uma pessoa é constituído de vários dados e o RDF não permite que um único predicado seja associado a mais de um objeto, se faz necessária a inclusão de um nó que represente o endereço de uma pessoa, mesmo que ele não tenha um identificador, o qual será posteriormente associado através de outros predicados aos valores dos objetos propriamente ditos.

Além de serem utilizados para auxiliar na definição de propriedades estruturadas, os nós em branco servem para criar sentenças sobre recursos que não têm um URI, mas podem ser descritos através de relações com outros recursos que tenham um identificador único.

Um exemplo simples é o caso de uma pessoa. Nos exemplos anteriores, ela estava sendo identificada por seu e-mail. Entretanto, sabe-se que uma pessoa é diferente de seu endereço de e-mail, pois apesar de identificar a mesma entidade, as ações que podem ser feitas com uma pessoa e com um endereço de e-mail são completamente distintas. Assim, cria-se um nó que represente a pessoa, com as informações que a identificam associadas a ele, como mostra a figura 10, além de uma definição do tipo *type* (termo definido pelo RDF) para indicar o tipo de recurso descrito.

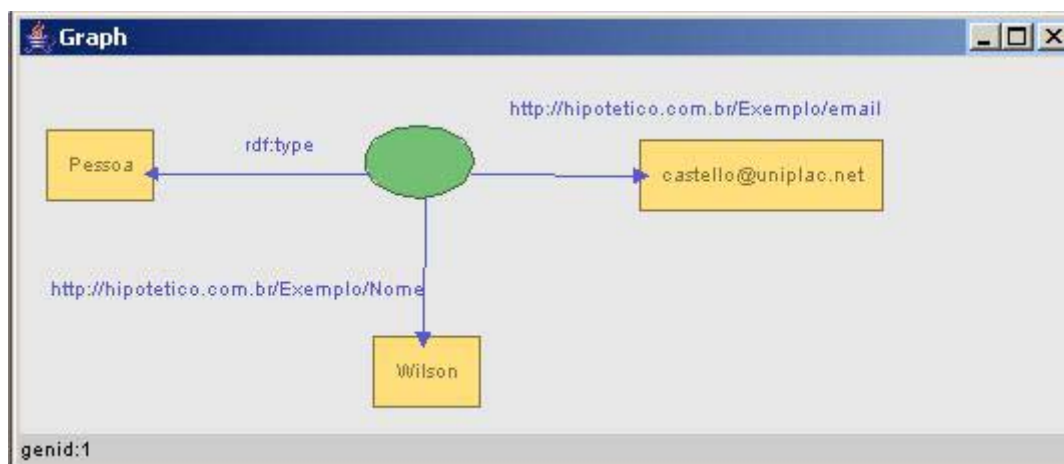


FIGURA 10 - Nó em branco para descrever recursos sem URI.

Outro recurso fornecido pelo RDF é a possibilidade de criar literais tipados, através de definições criadas em esquemas XML, permitindo a associação de tipos como inteiros, reais e datas aos dados, aumentando, assim, o poder de representatividade da linguagem.

Além dos recursos já mencionados, MANOLA e MILLER (2004) apresentam outros também importantes, como o que possibilita a criação de grupos de elementos para, por exemplo, definir o conjunto de autores de uma obra ou os estudantes de uma turma. Isto pode ser feito através dos tipos pré-definidos *Bag*, *Seq*, *Alt* e *List*, sendo o último mais

completo, pois possui propriedades como *first* e *rest*, comuns em outras linguagens de programação que manipulam listas.

Por fim, cabe ressaltar a propriedade de Reificação, que possibilita a criação de sentenças RDF sobre sentenças RDF, para determinar dados como quem criou determinada sentença e quando. Os termos pré-definidos do RDF para isto são: *Statement*, *Subjetc*, *Predicate* e *Object*.

3.3.3 Representação de Sentenças RDF em XML

A sintaxe abstrata baseada em Grafos do RDF, definida em KLYNE e CARROLL (2004), é extremamente útil por facilitar o entendimento das informações. Todavia, este modelo pode apresentar algumas ambigüidades e não é adequado para o envio de informações pela *Web*, tão pouco para o processamento automático das mesmas. Por isto, é necessário criar uma sintaxe para representar os dados de forma que eles possam ser facilmente armazenados, processados e transmitidos de um computador para outro.

Ao considerar as vantagens da linguagem XML, o W3C optou por utilizá-la como base para a construção de documentos RDF, criando a sintaxe normativa chamada RDF/XML, definida em BECKETT (2004), o qual destaca que, para codificar um grafo em XML, deve-se representar os recursos, predicados e objetos em termos XML – nomes de elementos, nomes de atributos, conteúdo dos elementos e valores dos atributos.

Para apresentar a sintaxe RDF/XML, utiliza-se o código do quadro 11, que representa as mesmas informações descritas no grafo da figura 11⁸. A linha 1, conforme descrito na seção 3.2.1, serve para indicar que aquele é um documento XML. A linha 2 indica que o conteúdo encontrado neste documento a partir desta linha até a *tag* `</rdf:RDF>`, presente na linha 12, é composto de declarações RDF.

As linhas 3, 4 e 5, que ainda fazem parte da *tag* de abertura `<rdf:RDF` da linha 2, definem os *namespaces* e os prefixos utilizados para identificá-los neste documento através do atributo *xmlns*. O *namespace* da linha 3 deve estar presente em todos os documentos RDF, pois é ele que define os termos do vocabulário RDF, como *description*, *about* e *RDF* presentes nas linhas 2, 6, 7, 11 e 12. Apesar de ter sido usado o prefixo *rdf*, é importante especificar que a pessoa ou aplicação que gera o documento tem liberdade para associá-lo ao prefixo que desejar. O que não pode ser alterado é o URI `http://www.w3.org/1999/`

⁸ A geração do código do quadro 11 e dos subseqüentes foi realizada automaticamente pela ferramenta Isaviz, com base nos grafos desenvolvidos com o auxílio desta mesma ferramenta.

02/22-rdf-syntax-ns# que indica onde se encontram as definições dos símbolos.

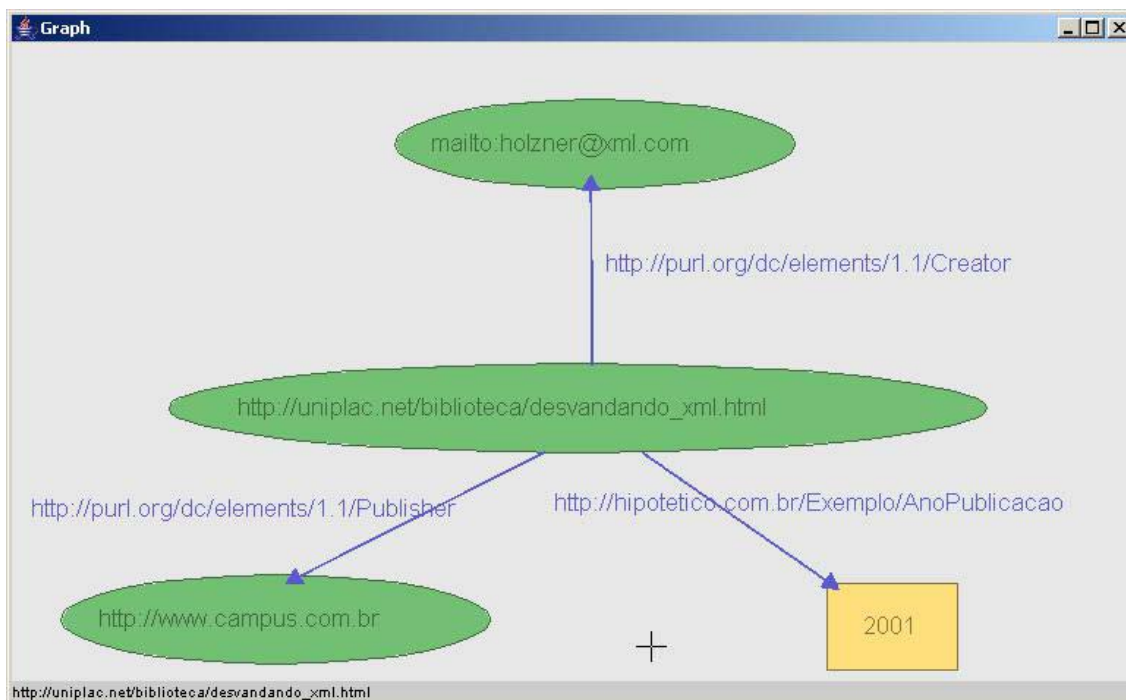


FIGURA 11 - Grafo com informações sobre o livro “Desvendando XML”

QUADRO 11 - Código do grafo da figura 11.

```

1. <?xml version="1.0"?>
2. <rdf:RDF
3.   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4.   xmlns:dc="http://purl.org/dc/elements/1.1/">
5.   xmlns:ex="http://hipotetico.com.br/Exemplo/"
6. <rdf:Description
7. rdf:about="http://uniplac.net/biblioteca/desvendando_xml.html">
8.   <ex:AnoPublicacao> 2001 </ex:AnoPublicacao>
9.   <dc:Publisher rdf:resource="http://www.campus.com.br"/>
10.  <dc:Creator rdf:resource="mailto:holzner+xml.com"/>
11.</rdf:Description>
12.</rdf:RDF>

```

As linhas 4 e 5 definem os outros dois vocabulários utilizados e os seus prefixos, que podem variar de documento para documento. A linha 4 traz a associação do vocabulário do *Dublin Core*, que pode nem sempre ser necessário, e a linha 5 indica que é usado um vocabulário desenvolvido pelo próprio criador do documento.

Nas linhas 6 a 11 estão as informações do grafo propriamente ditas. A linha 6 indica o início da descrição de um recurso, através do termo *Description* definido no *namespace*

rdf. Este é o início de uma declaração RDF. O sujeito é definido nesta mesma *tag* através do atributo *about* (linha 7), que neste caso recebeu como valor o identificador do recurso, ou seja a URL da página do livro.

As linhas 8, 9 e 10 apresentam a descrição dos predicados com seus respectivos objetos. O predicado é especificado através de uma *tag*, que é composta pelo prefixo do *namespace* onde a *tag* foi definida, concatenado ao símbolo “:” e seguido do termo que define o predicado.

Quando o objeto associado ao predicado é um literal, como no caso do *AnoPublicacao*, o seu valor é definido entre a *tag* de abertura do predicado e a sua respectiva de encerramento, como ocorre na linha 8. Mas quando o objeto é um outro recurso, ele é definido dentro da própria *tag* que define o predicado, através do atributo *rdf:resource*, que recebe o URI do recurso, conforme as linhas 9 e 10.

Segundo MANOLA e MILER (2004, p. 32 e 33), “Se o URI deste recurso fosse escrito como um literal entre uma *tag* de início e outra de fim, da mesma forma que os valores literais dos outros elementos, ele estaria dizendo que o valor de *dc:Creator*, por exemplo, é o conjunto de caracteres “<http://www.campus.com.br>” e não o recurso identificado por aquele literal interpretado como um URI”.

A descrição de um objeto é terminada pela *tag* de encerramento `</rdf:Description>` (linha 11). A partir deste ponto, outros recursos podem ser definidos de forma semelhante à definição apresentada.

O código apresentado no quadro 11 é uma forma resumida de descrever as informações do grafo apresentado na figura 11, pois todas as sentenças sobre o mesmo recurso foram agrupadas em uma cláusula `<rdf:Description>`. Entretanto, nem todas as ferramentas o fazem desta forma, criando uma descrição para cada sentença separadamente, como mostra o quadro 12, que apesar de ser organizado de outra forma, apresenta as mesmas informações do código do quadro 11.

A seguir é apresentado o código RDF/XML que descreve algumas das características adicionais da linguagem RDF/XML. O quadro 13 mostra o código correspondente ao grafo da figura 9, descrevendo a sintaxe para representar nós em branco.

Neste código pode-se observar que o recurso identificado pelo e-mail `castello@uniplac.net` está associado, através de um predicado “Endereço”, ao nó cujo identificador é igual a 1 (linha 6). Na seqüência, são descritas as associações deste nó com todos os objetos que o caracterizam (linhas 8 a 14).

QUADRO 12 - Código do grafo da figura 11 sem abreviaturas.

```

1. <?xml version="1.0"?>
2. <rdf:RDF
3.     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4.     xmlns:ex="http://hipotetico.com.br/Exemplo/"
5.     xmlns:dc="http://purl.org/dc/elements/1.1/">
6. <rdf:Description
7.   rdf:about="http://uniplac.net/biblioteca/desvendando_xml.html">
8.     <ex:AnoPublicacao> 2001 </ex:AnoPublicacao>
9. </rdf:Description>
10.<rdf:Description
11  rdf:about="http://uniplac.net/biblioteca/desvendando_xml.html">
12.     <dc:Publisher rdf:resource="http://www.campus.com.br"/>
13. </rdf:Description>
14.<rdf:Description
15  rdf:about="http://uniplac.net/biblioteca/desvendando_xml.html">
16.     <dc:Creator rdf:resource="mailto:holzner@xml.com"/>
17. </rdf:Description>
18.</rdf:RDF>

```

Por não ser o objetivo deste trabalho fazer um estudo exaustivo sobre RDF, não é apresentada a sintaxe para representação de grupos de elementos e reificação através do RDF/XML. As regras e exemplos sobre estes casos podem ser encontrados em BECKETT (2004) e MANOLA e MILLER (2004).

QUADRO 13 - Código RDF/XML para nós em branco.

```

1. <?xml version="1.0"?>
2. <rdf:RDF
3.     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4.     xmlns:ex="http://hipotetico.com.br/Exemplo/">
5.   <rdf:Description rdf:about="mailto:castello@uniplac.net">
6.     <ex:Endereco rdf:nodeID="1">
7.   </rdf:Description>
8.   <rdf:Description rdf:nodeID="1">
9.     <ex:Numero>277</ex:Numero>
10.    <ex:Cidade>Sao Joaquin</ex:Cidade>
11.    <ex:CEP>CEP</ex:CEP>
12.    <ex:Rua>Cel. Joao Palma</ex:Rua>
13.    <ex:Estado>Santa Catarina</ex:Estado>
14.   </rdf:Description>
15. </rdf:RDF>

```

3.3.4 RDF Schema

A linguagem RDF serve para criar sentenças sobre recursos através da utilização de propriedades e valores. Entretanto, ela não padroniza como estes nomes devem ou não ser utilizados, tão pouco descreve o significado de cada um deles. Esta característica remete ao mesmo problema enfrentado na linguagem XML, onde diferentes pessoas podem usar estruturas distintas para descrever uma mesma informação ou estruturas iguais para descrever informações diferentes.

No primeiro caso, a solução encontrada foi a criação dos esquemas XML para que pessoas interessadas pudessem chegar a um consenso e criar uma estrutura padrão para construir os seus documentos. Em RDF, o caminho é o mesmo, definiu-se uma linguagem para descrição de vocabulários, o RDF *Schema* (RDF/S), que é uma extensão semântica do RDF e fornece mecanismos para descrever recursos e os relacionamentos entre eles. A especificação do RDF *Schema* é definida em BRICKLEY e GUHA (2004).

De acordo com MANOLA e MILER (2004), o RDF/S surgiu da necessidade que as comunidades de usuários têm de definir os vocabulários padrões que eles pretendem utilizar nas sentenças. RDF/S não fornece um vocabulário específico para uma aplicação, definindo, por exemplo, elementos do tipo *Pessoa*, *Livro* ou *Autor*. “RDF *Schema* é um simples conjunto de recursos e propriedades RDF padrões que habilitam as pessoas a criar seus próprios vocabulários RDF” (DACONTA, OBRST e SMITH, 2003, p. 104). O vocabulário do RDF *Schema* é definido em <http://www.w3.org/2000/01/rdf-schema#> e é convencionalmente associado ao prefixo *rdfs* nos documentos que usam seus termos.

De certa forma, o RDF/S pode ser associado a uma linguagem de programação orientada a objetos, pois ele representa as informações através de um sistema de classes e propriedades. Porém, as semelhanças param neste ponto, já que o RDF/S não implementa os comportamentos dos objetos.

A seguir, são apresentados os conceitos mais utilizados na criação de esquemas RDF. Outros termos podem ser encontrados em DACONTA, OBRST e SMITH (2003) e BRICKLEY e GUHA (2004).

- *rdfs:class* – define que um determinado recurso é uma estrutura abstrata à qual são associadas propriedades que descreverão suas instâncias;
- *rdfs:subClassOf* – define que as instâncias de uma classe herdam todas as propriedades da outra;
- *rdfs:property* – define que um determinado recurso é uma propriedade;

- *rdfs:range* – define o tipo de valor recebido pelas propriedades;
- *rdfs:domain* - define a que classe uma propriedade está associada;
- *rdfs:literal* – define que a propriedade é definida por um conjunto de caracteres;
- *rdfs:label* – define um rótulo em linguagem natural para a classe, visando facilitar o entendimento da mesma pelos humanos;
- *rdfs:comment* – esta propriedade descritiva serve para adicionar informações textuais às classes e propriedades para auxiliar no entendimento das mesmas por outros usuários do esquema;
- *rdf:about* – define a identificação do recurso que está sendo definido.

Com estes termos, foi construído um exemplo de Esquema RDF para definir os seguintes conceitos: uma classe *Pessoa*, com os atributos *Nome* e *Idade* definidos como sendo do tipo literal; e uma classe *Empregado*, que é subclasse de *Pessoa* e possui o atributo adicional *CarteiraTrabalho*, também do tipo literal. Este esquema pode ser representado através de um grafo como mostra a figura 12.

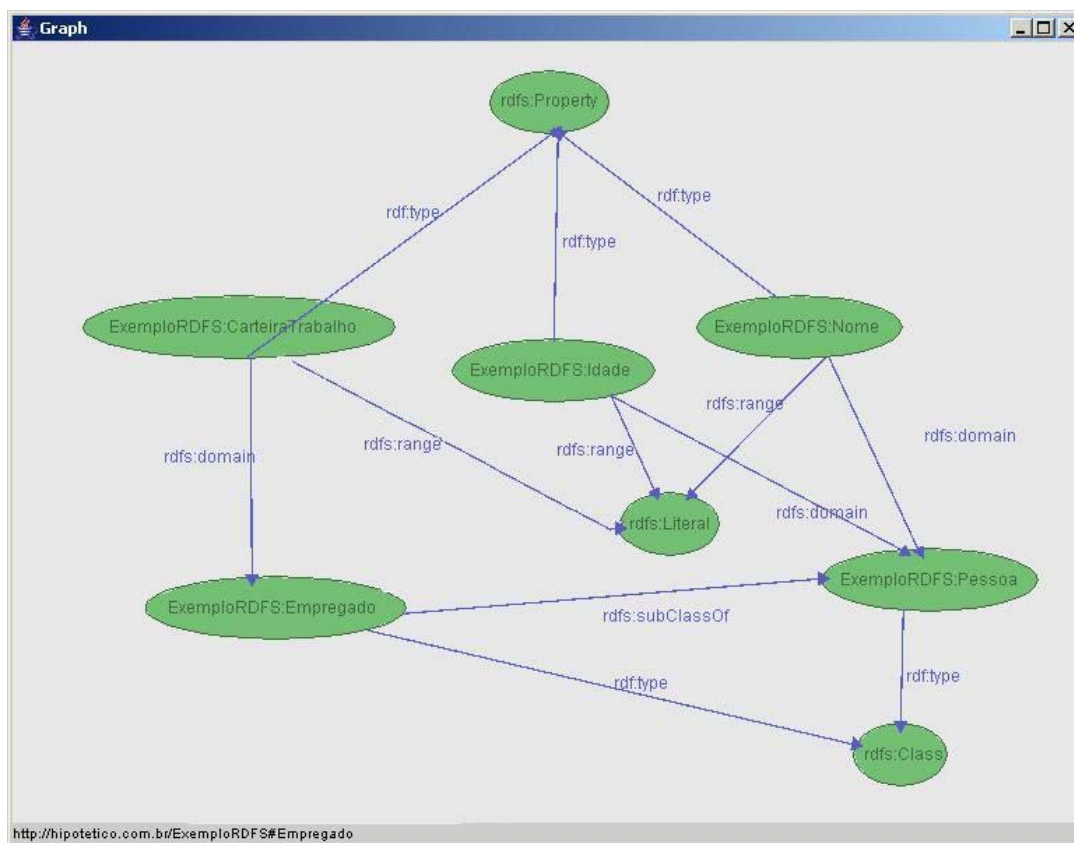


FIGURA 12 - Exemplo de Esquema RDF

Os URIs utilizados neste esquema foram abreviados para facilitar a visualização da figura. O prefixo rdfs foi associado ao *namespace* “<http://www.w3.org/TR/1999/PR-rdf-schema-19990303#>”, que contém as definições básicas para criação de esquemas RDF; e o prefixo ExemploRDFS foi associado a “<http://hipotetico.com.br/ExemploRDFS#>”, que é o namespace que conterá as definições criadas, como mostra o quadro 14, o qual traz o código RDF/XML correspondente ao grafo gerado pela ferramenta para criação de Ontologias e Esquemas RDF, Protégé (PROTEGE, 2005).

QUADRO 14 - Código de um Esquema RDF

```

1. <?xml version='1.0' encoding='UTF-8'?>
2. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.   xmlns:ExemploRDFS="http://hipotetico.com.br/ExemploRDFS#"
4.   xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#">
5.   <rdfs:Class rdf:about="http://hipotetico.com.br/ExemploRDFS#Pessoa">
6.     rdfs:comment="Exemplo - Classe Pessoa" rdfs:label="Pessoa">
7.   <rdfs:subClassOf rdf:resource=
8.     "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource"/>
9.   </rdfs:Class>
10. <rdfs:Class rdf:about="http://hipotetico.com.br/ExemploRDFS#Empregado"
11.   rdfs:comment="Exemplo-Classe Empregado" rdfs:label="Empregado">
12.   <rdfs:subClassOf rdf:resource=
13.     "http://hipotetico.com.br/ExemploRDFS#Pessoa"/>
14. </rdfs:Class>
15. <rdf:Property rdf:about="http://hipotetico.com.br/ExemploRDFS#Nome"
16.   rdfs:comment="Exemplo - Propriedade Nome" rdfs:label="Nome">
17.   <rdfs:domain rdf:resource=
18.     "http://hipotetico.com.br/ExemploRDFS#Pessoa"/>
19.   <rdfs:range rdf:resource=
20.     "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Literal"/>
21. </rdf:Property>
22. <rdf:Property rdf:about="http://hipotetico.com.br/ExemploRDFS#Idade"
23.   rdfs:comment="Exemplo - Propriedade Idade" rdfs:label="Idade">
24.   <rdfs:domain rdf:resource=
25.     "http://hipotetico.com.br/ExemploRDFS#Pessoa"/>
26.   <rdfs:range rdf:resource=
27.     "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Literal"/>
28. </rdf:Property>
29. <rdf:Property rdf:about=
30.   "http://hipotetico.com.br/ExemploRDFS#CarteiraTrabalho"
31.   rdfs:comment="Exemplo - Propriedade CarteiraTrabalho"

```

```

32.   rdfs:label="CarteiraTrabalho" >
33.   <rdfs:domain rdf:resource=
34.   "http://hipotetico.com.br/ExemploRDFS#Empregado"/>
35.   <rdfs:range rdf:resource=
36.   "http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Literal"/>
37. </rdf:Property>
38. </rdf:RDF>

```

Neste código, é possível visualizar que a criação de um esquema XML obedece às mesmas regras de um documento RDF comum. O cabeçalho contém as definições de que aquele é um arquivo RDF/XML e os prefixos associados aos *namespaces* utilizados.

Nas linhas 5 a 9 estão os elementos que definem a classe *Pessoa*, sua descrição e a indicação de que ela é derivada do tipo “recurso”. Já nas linhas 10 a 14 está a declaração da classe “Empregado” e a definição de que a mesma é subclasse de “Pessoa”. As declarações das propriedades estão nas linhas 15 a 21, 22 a 28 e 29 a 36, que definem os tipos de cada uma delas e indicam a que classe elas estão associadas.

Neste ponto, é importante destacar a diferença entre as propriedades de uma linguagem de programação orientada a objetos e de um esquema RDF. Enquanto no primeiro caso as propriedades são criadas dentro de uma classe e pertencem apenas a ela, em um esquema RDF elas são entidades independentes das classes, o que permite que uma mesma propriedade pertença a várias classes simultaneamente, bastando para isto adicionar novos elementos “domain” à propriedade.

O quadro 15 apresenta um documento RDF com instâncias do esquema definido no quadro 14. Este documento possui a definição de um empregado de nome Paulo, 28 anos e carteira de trabalho número 123.456.789 (linhas 5 a 8); e de duas pessoas com o nome de Maria e Pedro, e idade de 30 e 20 anos, respectivamente, nas linhas 9 a 11 e 12 a 14.

3.3.5 Considerações sobre RDF para a Web Semântica

A importância da linguagem RDF para a *Web Semântica* pode ser vista em BERNERS-LEE (1998a e 1998b), quando o autor mostra que ela é a base que possibilita a criação de metadados que podem ser analisados e processados pelos computadores de forma autônoma. Ele destaca, inclusive, as vantagens que a mesma tem sobre outras linguagens para definição de metadados, como o XML quando utilizado de forma isolada. Além disso, esta seção procurou mostrar a importância da criação de esquemas RDF para

que comunidades possam criar vocabulários comuns, expressando um consenso sobre as definições de um determinado assunto.

QUADRO 15 - Instância do Esquema RDF.

```

1. <?xml version='1.0' ?>
2. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3. xmlns:ExemploRDFS="http://hipotetico.com.br/ExemploRDFS#"
4. xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#">
5. <ExemploRDFS:Empregado
6.   rdf:about="ExemploRDFS:RDFS_Instance_0"
7.   ExemploRDFS:CarteiraTrabalho="123.456.789" ExemploRDFS:Idade="28"
8.     ExemploRDFS:Nome="Paulo" />
9. <ExemploRDFS:Pessoa
10.   rdf:about="ExemploRDFS:RDFS_Instance_1"
11.   ExemploRDFS:Idade="30" ExemploRDFS:Nome="Maria" />
12.<ExemploRDFS:Pessoa
13.   rdf:about="ExemploRDFS:RDFS_Instance_2"
14.   ExemploRDFS:Idade="20" ExemploRDFS:Nome="Pedro" />
15.</rdf:RDF>

```

Segundo MANOLA e MILLER (2004), os recursos fornecidos pelo RDF e pelos esquemas RDF são suficientes para o desenvolvimento de uma série de aplicações úteis, mas ainda não o são para o desenvolvimento da *Web Semântica*, a qual requer a existência de linguagens que possibilitem o tratamento da semântica dos recursos pelos computadores.

De acordo com FENSEL *et al.* (2003), este tratamento é possibilitado pelas linguagens para criação de Ontologias, em uma relação de extensão com os esquemas RDF, dando um significado comum a termos como *class*, *property*, *subclassof* e todos os demais termos definidos no *namespace* para criação de esquemas RDF. Dessa forma, as ontologias permitem a representação de informações e significados que ainda não podem ser capturados pelos esquemas RDF, aumentando o nível de representatividade da linguagem e utilizando a mesma sintaxe para a criação dos documentos.

3.4 Ontologias

Ao analisar a colocação de BERNERS-LEE (1999), onde é explicitado que o desenvolvimento da *Web Semântica* requer: Linguagens para expressar meta-informação

sobre documentos que seja processável pelo computador; Terminologias padrões sobre os mais diversos domínios; e Ferramentas e arquiteturas que usem tais linguagens e terminologias para fornecer suporte à busca, acesso, apresentação e manutenção de informações; cabe destacar que XML e RDF possibilitam a associação de meta-informação aos documentos, satisfazendo, desta forma, o primeiro requisito.

O RDF/S e as ontologias são partes fundamentais para o desenvolvimento da *Web Semântica*, por possibilitarem a construção das terminologias padrões sobre os diferentes domínios, satisfazendo o segundo requisito listado por BERNERS-LEE (1999).

O termo Ontologia, utilizado primeiramente para definir um campo da Filosofia, tornou-se uma palavra comum na área de tecnologia da informação, principalmente entre os pesquisadores em IA e *Web Semântica*. Alguns conceitos ajudam a entender o papel de uma ontologia, quando este termo refere-se à engenharia do conhecimento:

- “Uma ontologia é uma especificação explícita dos conceitos em um domínio e as relações entre eles, que fornecem um vocabulário formal para a troca de informações” (NOY *et al.* 2001, p. 60);
- “Uma ontologia define as palavras e conceitos (significados) comuns usados para descrever e representar uma área do conhecimento” (DACONTA, OBRST e SMITH, 2003, p.186);
- Ontologia é uma especificação de uma conceitualização. Por conceitualização entende-se uma visão simplificada e abstrata do mundo que se deseja representar para algum propósito (GRUBER, 1993a, p. 1).

Embora ainda existam algumas dúvidas sobre o que é uma ontologia na engenharia do conhecimento, os seus objetivos nesta área já estão consolidados como sendo uma forma de possibilitar a formalização, reutilização e o compartilhamento do conhecimento. Esses objetivos, que despertaram o interesse dos pesquisadores de IA há vários anos, ganharam maior atenção com o advento da *Web Semântica*, devido ao anseio por novas formas de representação do conhecimento capazes de possibilitar maior grau de autonomia aos sistemas que o manipulam.

Com o intuito de definir de forma mais detalhada o que representa uma ontologia na engenharia do conhecimento e, ao mesmo tempo, mostrar sua relação com termos conhecidos como taxionomia, thesaurus e modelo conceitual, DACONTA, OBRST e SMITH (2003) usam o termo *espectro de ontologia*. Ele engloba os conceitos apresentados na figura 13, que dispõem um conjunto de modelos em um *continuum* de acordo com o

poder de expressividade semântica de cada um deles. Todos estes modelos tratam de representar, classificar e tornar não ambíguo o conteúdo semântico (significado) das informações.

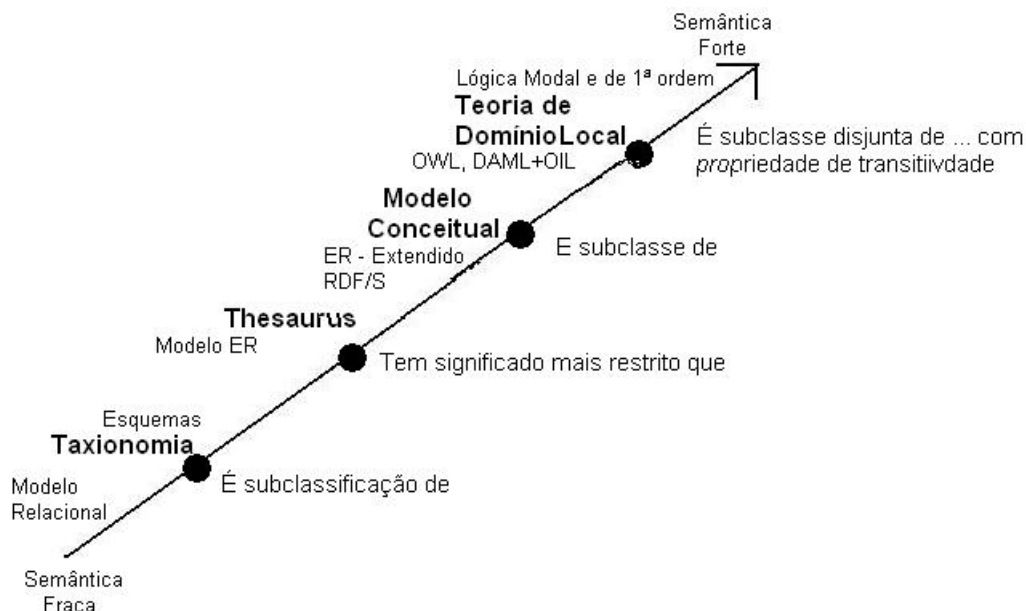


FIGURA 13 - Espectro de Ontologia

Fonte: Adaptado de DACONTA, OBRST e SMITH (2003, p. 157)

Segundo os autores, pode ser considerada como uma ontologia desde uma simples taxionomia, que está no início do *continuum* por possuir um pequeno poder de representação semântica, um *thesaurus*, os modelos conceituais e até as mais elaboradas teorias lógicas, com seus ricos e consistentes mecanismos para representação do conhecimento. Nesta mesma figura, os autores posicionam os modelos para definição de bancos de dados, visando melhor identificar o poder de representação de cada uma das formas apresentadas.

Como a arquitetura da *Web Semântica* (figura 2) apresenta os esquemas (XML e RDF) nas camadas inferiores, o que se convencionou chamar de Ontologia são os modelos criados através das linguagens OWL (*Web Ontology Language*) e DAML+OIL, que são baseados nas teorias lógicas e possuem um poder de representatividade maior que os esquemas das camadas inferiores.

Uma ontologia, de forma semelhante aos esquemas, possui mecanismos que permitem definir:

- Classes (entidades genéricas);
- Instâncias (entidades específicas);
- Relacionamentos entre entidades;

- Propriedades das entidades e seus respectivos valores;
- Funções e Processos envolvendo estas entidades;
- Restrições e Regras aplicadas às entidades.

Entretanto, KLEIN *et al.* (2003) aponta algumas diferenças cruciais entre estes tipos de linguagens. Em relação a uma linguagem para criação de ontologias e os esquemas XML, o autor destaca que apesar de ambos possuírem o objetivo de possibilitar o intercâmbio de dados, a principal diferença está no nível em que eles atuam. Enquanto uma ontologia serve para especificar teorias sobre um determinado domínio, os esquemas XML são utilizados para definir restrições de integridade para documentos e dados estruturados;

Para melhor expor a diferença entre o papel das ontologias e esquemas XML, a figura 14 compara-os aos papéis de um modelo entidade-relacionamento e relacional no processo de desenvolvimento de um banco de dados.

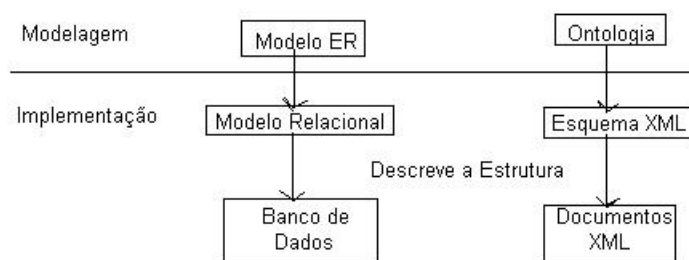


FIGURA 14 - Relação entre ontologias e esquemas XML
 Fonte: Adaptado de KLEIN *et al.*(2003, p. 113)

Nesta figura, é possível perceber que as linguagens são complementares, já que a ontologia é utilizada em um nível de abstração maior, para modelar as informações, enquanto os esquemas XML servem para dar uma forma processável pelo computador às informações descritas na ontologia.

Com relação ao RDF/S, KLEIN *et al.* (2003) destaca que eles possuem os mesmos objetivos das ontologias, ficando como diferença significativa o poder de representatividade da linguagem. As linguagens para definição de ontologias possuem mais primitivas de modelagem e uma semântica formal melhor definida, tornando o processo de raciocínio sobre elas mais fácil do que em RDF/S.

Dentre outras limitações do RDF/S que são resolvidas pelas ontologias, ANTONIOU e HARMELEN (2004) destacam a possibilidade de definir:

- Classes disjuntas;
- Restrições de cardinalidade;

- Propriedades transitivas, inversas e simétricas;

Da mesma forma que os esquemas RDF expandem a linguagem RDF por meio das definições semânticas de termos como “Class” e “SubClassOf”, as ontologias podem ser escritas através de esquemas RDF, desde que seja definida a semântica de alguns termos específicos. É importante destacar que, apesar de em alguns momentos estas linguagens se completarem e em outros se sobreporem, seu maior objetivo é permitir que os benefícios das ontologias na modelagem de um domínio possam ser aplicados a recursos *on-line*, possibilitando, assim, a construção da *Web Semântica*.

3.4.1 Aplicações de Ontologias

A pesquisa sobre ontologias tornou-se um campo bastante amplo desde que este termo foi integrado ao vocabulário das pessoas envolvidas com tecnologia da informação. Ela é utilizada em pesquisas que vão desde a engenharia e representação do conhecimento, passando pelo projeto de bancos de dados e análise orientada a objetos, chegando à modelagem, integração e recuperação de informação (GUARINO, 1998). É certo que a *Web Semântica* não aparece na lista acima devido à data em que o documento foi escrito, mas ela representa, hoje, uma das áreas que mais tem se beneficiado dos recursos das ontologias.

Considerando o sucesso da *Web*, a fase de transição para a *Web Semântica* e a importância das ontologias neste processo, McGUINNESS (2003) apresenta algumas formas pelas quais as ontologias podem ser utilizadas, iniciando pelas mais simples:

- Fornecem um vocabulário controlado – com as pessoas utilizando o mesmo conjunto de termos, mesmo que com sentidos diferentes, dá-se um passo em direção à interoperabilidade;
- São capazes de organizar e dar suporte à navegação em um *website*, bem como auxiliar no processo de busca de informações;
- Podem ser usadas para criar uma estrutura “guarda-chuva”, a partir da qual o conteúdo é estendido.

Já as ontologias mais estruturadas podem ser aplicadas para:

- Checagem de Consistência – as informações obtidas em uma busca podem ser associadas às definições de uma ontologia para verificar se os valores

retornados são válidos ou não;

- Complemento de Informações – após obter um pequeno conjunto de informações sobre determinada entidade, as ontologias podem ser utilizadas para gerar novas informações com base nas previamente obtidas;
- Busca estruturada, comparativa e personalizada – com os formalismos existentes nas ontologias é possível executar buscas mais precisas que as feitas atualmente através de palavras-chave;
- Interoperabilidade completa - com a criação e a aceitação das ontologias pelas comunidades, será possível desenvolver sistemas heterogêneos que façam uso das mesmas para trocar informações sem a necessidade de intervenção dos humanos.

Um dos aspectos a serem considerados no desenvolvimento de uma ontologia é a escolha da linguagem para a sua construção, que deve possuir os recursos necessários para possibilitar a representatividade semântica desejada. Atualmente, as linguagens mais comuns para a criação de ontologias, de acordo com DACONTA, OBRST e SMITH (2003), são RDF *Schema*, DAML+OIL (DARPA *Agent Markup Language+Ontology Inference Layer* – Linguagem para Marcação de Agentes DARPA + Camada de Inferência de Ontologias) e OWL.

Na seqüência, são apresentadas as principais características da OWL, por ser a linguagem padrão definida pelo W3C para a construção de ontologias para a *Web Semântica*.

3.4.2 *Web Ontology Language*

OWL é uma linguagem para definir e instanciar ontologias *Web* recomendada pelo W3C para o desenvolvimento da *Web Semântica* (SMITH, WELTY e McGUINNESS, 2004). “OWL deve ser utilizada quando a informação contida em documentos precisa ser processada por aplicações, em contraposição às situações onde o conteúdo apenas deve ser mostrado aos humanos. OWL pode ser utilizada para representar explicitamente o conteúdo e a semântica de termos em vocabulários e os relacionamentos entre estes termos” (McGUINNESS e HARMELEN, 2004).

Além da OWL, outra linguagem bastante difundida para a construção de ontologias é a DAML+OIL, que teve origem na fusão das linguagens DAML, desenvolvida pelo

DARPA, e OIL, desenvolvida por pesquisadores da união européia. Estas linguagens serviram como base para a OWL, a qual deve possuir todas as características de suas antecessoras, além de novos recursos exclusivos (DACONTA, OBRST e SMITH, 2003).

A OWL fornece três sublinguagens com níveis de representatividade diferentes, são elas (McGUINNESS e HARMELEN, 2004):

- OWL *Lite*: Fornece os mecanismos básicos para classificação dos termos em hierarquias e para a definição de restrições sobre eles;
- OWL DL: possibilita uma representatividade completa da OWL, pois inclui todas as construções, além de garantir completude (todas as conclusões são computáveis) e decidibilidade (todas as computações terminarão em um tempo finito);
- OWL FULL: possui os mesmos construtores da OWL DL, mas oferece uma sintaxe mais flexível, o que não garante que as ontologias desenvolvidas através de seus recursos sejam completas e decidíveis.

A seguir, são apresentados alguns termos utilizados para construção de ontologias com a OWL. Maiores detalhes podem ser vistos em McGUINNESS e HARMELEN (2004), SMITH, WELTY e McGUINNESS (2004) e DEAN e SCHREIBER (2004).

- *owl:ontology*: Define o cabeçalho com algumas informações sobre a ontologia;
- *owl:imports*: Indica as ontologias que devem ser utilizadas juntamente com a que está sendo desenvolvida;
- *owl:class*: Define uma classe na ontologia;
- *owl:datatypeProperty*: Define uma propriedade que tem como valor um literal;
- *owl:objectProperty*: Define uma propriedade que tem como valor um objeto ou uma classe;
- *owl:functionalProperty*: Define que uma propriedade é funcional, desta forma, cada instância pode ter apenas um valor atribuído a esta propriedade;
- *owl:symmetricProperty*: Define que uma propriedade é simétrica; ou seja, se o par (X,Y) é válido para um determinado predicado, automaticamente o par (Y,X) também o será;
- *owl:transitiveProperty*: Define que uma propriedade é transitiva; se os pares

(X,Y) e (Y,Z) são válidos para um predicado, o par (X,Z) também será;

- *owl:inverseOf*: Define uma propriedade inversa em relação à outra propriedade;
- *owl:cardinality*: Define a cardinalidade das propriedades;
- *owl:disjointWith*: Indica que duas classes são disjuntas; as classes associadas por este termo não possuem instâncias em comum;
- *owl:intersectionOf*: Determina que o conjunto de elementos de uma classe é formado pela intersecção dos elementos de outras duas ou mais classes;
- *owl:unionOf*: Determina que o conjunto de elementos de uma classe é formado pela união dos elementos de outras duas ou mais classes;

A modelagem apresentada na figura 15 apresenta as classes de uma ontologia sobre publicações científicas, são elas:

- *Publicação*: possui propriedades como o título da publicação, ano, título e autor do artigo, local e área de concentração. Esta classe é especializada em duas outras, *Periódico* e *Anais_Evento*, que apresentam propriedades específicas de cada tipo de publicação.
- Além das três classes principais e suas propriedades textuais, foram criadas as classes *Área_Concentracao*, *Editora* e *Autor*, cujas instâncias definem as propriedades da classe *Publicação*.

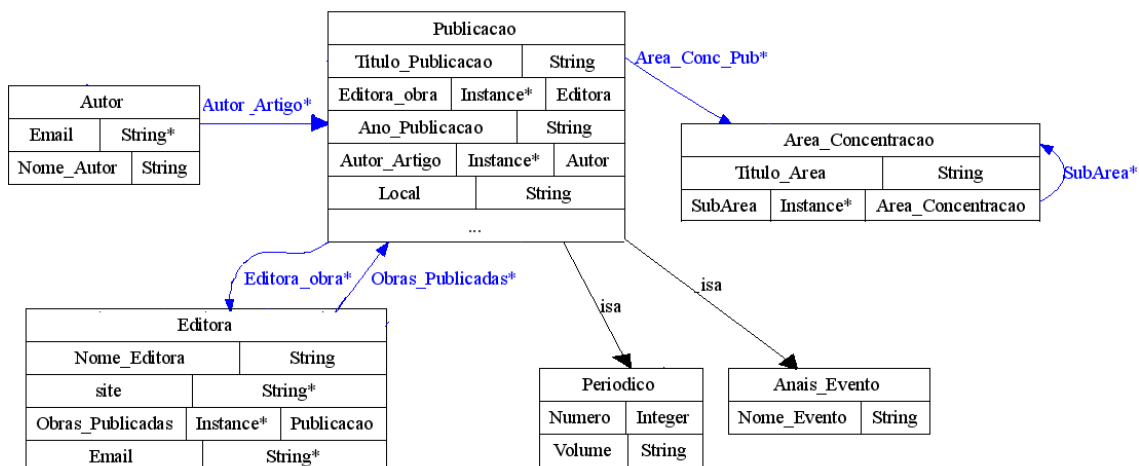


FIGURA 15 - Ontologia para publicações científicas

O código desta ontologia foi dividido em partes, as quais são descritas separadamente nos quadros 16, 17, 18 e 19 para facilitar o entendimento.

Com base no código do quadro 16, pode-se perceber que uma ontologia OWL também possui construções da linguagem RDF/XML. Suas instruções possuem funções semelhantes às já vistas nos exemplos anteriores, que é indicar que aquele é um documento escrito em XML/RDF e de definir os *namespaces* que são utilizados na ontologia.

QUADRO 16 - Ontologia para o modelo da figura 15 (Início).

```

1. <?xml version="1.0"?>
2. <rdf:RDF
3.     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4.     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
5.     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
6.     xmlns:owl="http://www.w3.org/2002/07/owl#"
7.     xml:base=""http://hipotetico.com.br/OntoEx.owl" >

```

Além dos *namespaces* já conhecidos, merece destaque a linha 6, responsável por associar o prefixo owl ao URI `http://www.w3.org/2002/07/owl#`, que contém as definições dos termos específicos para construção de ontologias; e a linha 7, que define onde serão armazenadas as definições desta ontologia.

Após a definição dos *namespaces*, é criado o cabeçalho que contém informações para controle de versão e comentários gerais, como mostram as linhas de código do quadro 17.

QUADRO 17 - Ontologia para o modelo da figura 15 (Cabeçalho)

```

8. <owl:Ontology rdf:about="">
9.<owl:versionInfo rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
10.     1.0
11. </owl:versionInfo>
12. <rdfs:label>
13.     Ontologia de Exemplo sobre Publicações Científicas.
14. </rdfs:label>
15.</owl:Ontology>

```

O quadro 18 apresenta as instruções responsáveis pela criação de duas das classes existentes na ontologia, *Anais_Evento* (linhas 16 a 23) e *Publicação* (24 a 28). A cada uma delas é associada uma breve descrição e é indicado que a classe *Anais_Evento* é uma subclasse de *Publicação*, conforme comandos das linhas 20 a 22. Os comandos para criação das demais classes são semelhantes aos descritos no quadro 18 e foram omitidos

para facilitar a legibilidade.

QUADRO 18 - Ontologia para o modelo da figura 15 (Definição das Classes)

```

16.<owl:Class rdf:ID="Anais_Evento">
17.  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
18.    Subclasse que herda as caracteristica da classe publicacao e a
19.    especializa para anais de eventos </rdfs:comment>
20.  <rdfs:subClassOf>
21.    <owl:Class rdf:ID="Publicacao"/>
22.  </rdfs:subClassOf>
23.</owl:Class>
24.<owl:Class rdf:about="#Publicacao">
25.  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
26.    Classe que engloba todos os tipos de publicações científicas.
27.  </rdfs:comment>
28.</owl:Class>

```

O quadro 19 apresenta a definição das propriedades da ontologia, bem como a associação das mesmas às classes. Neste ponto, surgem as diferenças mais significativas entre as ontologias e os esquemas RDF.

Inicialmente, ressalta-se a definição das propriedades, que em um esquema RDF é feita apenas através do comando `<rdf:Property>`. Já em OWL, existem dois comandos para realizar esta tarefa:

- *owl:ObjectProperty* - utilizado nas linhas 29 e 33 para definir as propriedades *Area_Conc_Pub* e *Subarea*, que recebem como valor uma instância da classe *Area_Concetracao*;
- *owl:DatatypeProperty* - para definir que o valor da propriedade é um literal (definição da propriedade *Titulo_Area*, na linha 39).

Na definição de *Titulo_Area* há a indicação de que aquela é uma propriedade funcional (linhas 41 e 42), portanto, cada área de concentração pode possuir um único título. Também foi utilizado o predicado *UnionOf* (linhas 47 a 50), para definir que o domínio da propriedade *Email* é constituído de instâncias das classes *Autor* e *Editora*.

Outro recurso da linguagem OWL que pode ser visto neste exemplo é a propriedade inversa (*owl:inverseOf*). Ao definir a propriedade *Editora_obra* na classe *Publicação* (linhas 54 a 60), que recebe um objeto do tipo *Editora*, criou-se a propriedade inversa *Obras_Publicadas* (linhas 61 a 65), que define que uma editora pode estar associada a

várias instâncias da classe *Publicação*.

QUADRO 19 - Ontologia para o modelo da figura 15 (Definição das Propriedades)

```

29. <owl:ObjectProperty rdf:ID="Area_Conc_Pub">
30.   <rdfs:range rdf:resource="#Area_Concentracao"/>
31.   <rdfs:domain rdf:resource="#Publicacao"/>
32. </owl:ObjectProperty>
33. <owl:ObjectProperty rdf:ID="Subarea">
34.   <rdfs:domain rdf:resource="#Area_Concentracao"/>
35.   <rdfs:type rdf:resource=
36.     "http://www.w3.org/2002/07/owl#TransitiveProperty"/>
37.   <rdfs:range rdf:resource="#Area_Concentracao"/>
38. </owl:ObjectProperty>
39. <owl:DatatypeProperty rdf:ID="Titulo_Area">
40. <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
41. <rdfs:type rdf:resource=
42.   "http://www.w3.org/2002/07/owl#FunctionalProperty"/>
43. <rdfs:domain rdf:resource="#Area_Concentracao"/>
44. </owl:DatatypeProperty>
45. <owl:DatatypeProperty rdf:ID="Email">
46. <rdfs:domain> <owl:Class>
47.   <owl:unionOf rdf:parseType="Collection">
48.     <owl:Class rdf:about="#Autor"/>
49.     <owl:Class rdf:about="#Editora"/>
50.   </owl:unionOf>
51. </owl:Class> </rdfs:domain>
52. <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
53. </owl:DatatypeProperty>
54. <owl:ObjectProperty rdf:ID="Editora_obra">
55.   <owl:inverseOf>
56.     <owl:InverseFunctionalProperty rdf:ID="Obras_Publicadas"/>
57.   </owl:inverseOf>
58.   <rdfs:range rdf:resource="#Editora"/>
59.   <rdfs:domain rdf:resource="#Publicacao"/>
60. </owl:ObjectProperty>
61. <owl:InverseObjectProperty rdf:about="#Obras_Publicadas">
62.   <owl:inverseOf rdf:resource="#Editora_obra"/>
63.   <rdfs:domain rdf:resource="#Editora"/>
64.   <rdfs:range rdf:resource="#Publicacao"/>
65. </owl:InverseObjectProperty>

```

Outros recursos, como propriedades transitivas, foram utilizados na construção desta ontologia, porém não foram descritos neste exemplo já que seu objetivo é apenas destacar a superioridade das ontologias sobre o RDF/S, em relação ao seu poder de representatividade, e não descrever todos os seus recursos.

Instâncias das classes expostas na figura 15 podem ser criadas para representar informações sobre as publicações, de acordo com a estrutura proposta na ontologia. A figura 16 expõe algumas instâncias da classe *Area_Concentracao*, que padronizam a nomenclatura utilizada para indicar a que área do conhecimento uma publicação está vinculada.

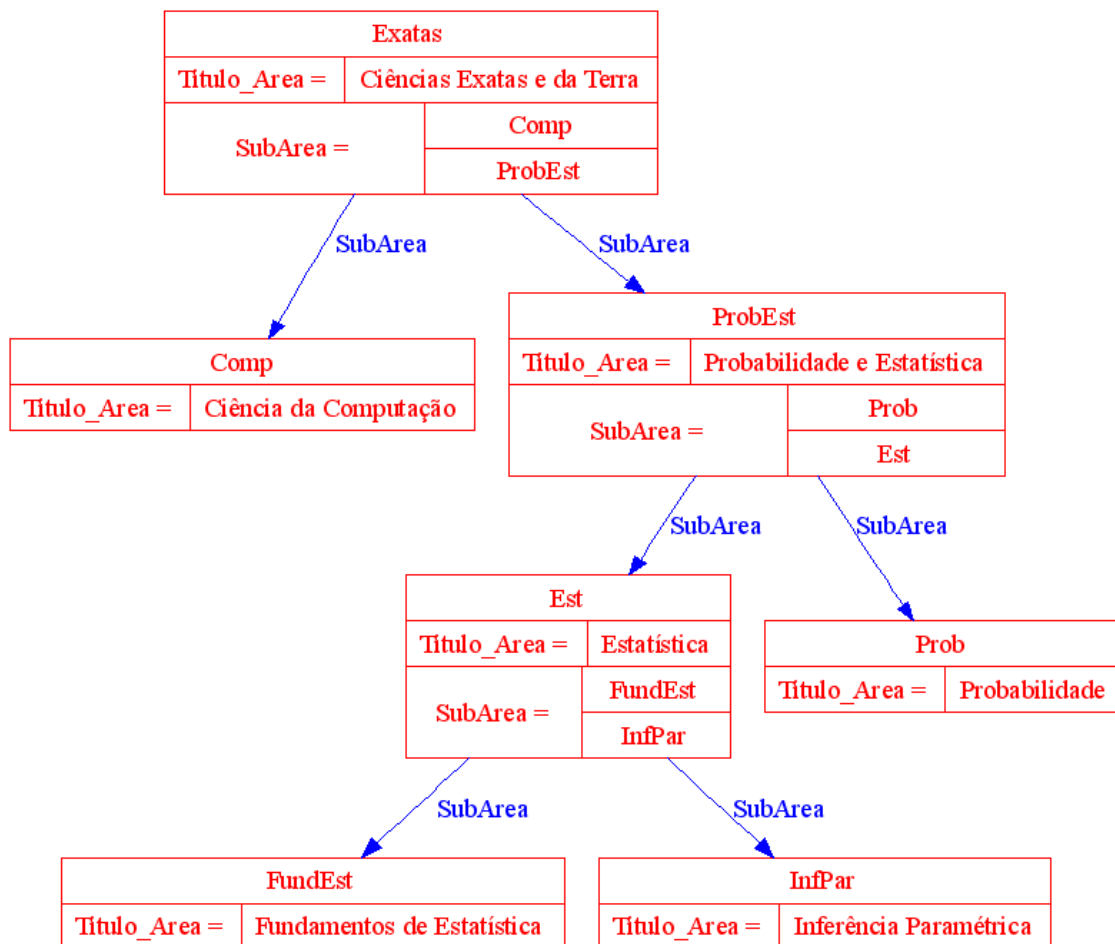


FIGURA 16 - Instâncias da classe *Area_Conc_Pub*

Neste exemplo, pode-se perceber a vantagem de utilizar ontologias para representar informações. Todas as publicações que receberem um dos termos da classe *Area_Concentracao*, definidos na ontologia para descrever a sua área de concentração, podem ser encontradas por mecanismos de buscas baseados nesta mesma ontologia. Isto

possibilita que resultados mais precisos sejam gerados para o usuário e evita que materiais sem utilidade sejam apresentados, em função da falta de estruturação da linguagem natural e das páginas existentes na *Web*.

No quadro 20 pode ser visto parte do código OWL referente às instâncias apresentadas na figura 16. A linha 1 indica o início da definição da área de concentração denominada *Exatas*, cujo título é *Ciências Exatas e da Terra* (linhas 2 a 4).

A partir da linha 5 estão listadas as suas subáreas, cuja construção é semelhante à da área de *Exatas*, iniciando com a definição do identificador da área, seguindo com o seu título e terminado com as subáreas. Esta semelhança pode ser vista a partir da linha 6, onde se inicia a definição da instância *ProbEst*, e das linhas 11 e 16, onde estão definidos os identificadores das instâncias *Prob* e *Est*.

QUADRO 20 - Código OWL com instâncias apresentadas na figura 16

```

1.<Area_Concentracao rdf:ID="Exatas">
2. <Titulo_Area rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
3.   Ciências Exatas e da Terra
4. </Titulo_Area>
5. <SubArea>
6.   <Area_Concentracao rdf:ID="ProbEst">
7.     <Titulo_Area rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
8.       Probabilidade e Estatística
9.     </Titulo_Area>
10.    <SubArea>
11.      <Area_Concentracao rdf:ID="Prob">
12.        <Titulo_Area ....>
13.          Probabilidade
14.        </Titulo_Area>
15.      </Area_Concentracao>
16.      <Area_Concentracao rdf:ID="Est">
17.        <Titulo_Area ....>
18.          Estatística
19.        </Titulo_Area>
20.      </Area_Concentracao>
21.    </SubArea>
22.  </Area_Concentracao>
23.</SubArea>
24.</Area_Concentracao>

```


3.4.3 Construção de Ontologias

Somente uma linguagem robusta e completa não é suficiente para que ontologias úteis possam ser desenvolvidas. É importante tomar alguns cuidados durante o projeto das mesmas para atingir os resultados esperados que, segundo GRUBER (1993b), são:

- Clareza – Uma ontologia deve comunicar efetivamente o significado pretendido para seus termos, através de definições objetivas e bem documentadas;
- Coerência – Os axiomas devem ser logicamente consistentes e as inferências geradas não podem contradizer a documentação em linguagem natural;
- Extensível – O conteúdo de uma ontologia deve ser projetado de forma que ela possa ser atualizada monotonicamente, ou seja, novos termos poderão ser definidos sem que os já existentes precisem ser revistos;
- Limiar de Codificação Mínimo – A conceitualização deve ser especificada no nível do conhecimento, sem depender de uma linguagem específica;
- Compromisso Ontológico Mínimo – Uma ontologia deve definir apenas os termos extremamente necessários para que as informações possam ser compartilhadas. As demais devem ser especializadas por cada um dos interessados.

Todos os aspectos destacados por GRUBER (1993b) influenciam diretamente no momento em que uma ontologia é compartilhada e reutilizada. É de fundamental importância que toda ontologia tenha suas definições claramente descritas para que outras pessoas possam reutilizá-las. Além disto, o projeto deve prever a inclusão de novas informações sem que haja necessidade de alterar o que já está implementado. As dificuldades em reutilizar ontologias são destacadas em SMITH, WELTY e McGUINNESS (2004) com a seguinte assertiva: “Se você for capaz de encontrar ontologias que têm sido utilizadas e refinadas extensivamente, faz sentido adotá-la (... mas...) será desafiador unir uma coleção de ontologias”.

Algumas instruções OWL auxiliam a tarefa de união de ontologias, são elas:

- *owl:equivalentClass* – define que uma classe definida em uma ontologia possui as mesmas características de outra classe já existente em outra ontologia;

- *owl:equivalentProperty* – funciona de forma semelhante ao comando anterior, mas sua abrangência é apenas para uma propriedade;
- *owl:sameAs* – define a equivalência entre instâncias;
- *owl:differentFrom* – possui efeito oposto ao comando *sameAs*.

Entretanto, não bastam apenas comandos para que ontologias possam ser criadas, compartilhadas e reutilizadas. O desenvolvimento da Web Semântica requer que as comunidades realizem esforços comuns com o objetivo de padronizar suas terminologias, para que aplicações mais complexas possam ser desenvolvidas.

3.4.4 Regras para a Web Semântica

Os assuntos abordados até o momento no presente capítulo constituem os níveis inferiores e intermediários da Web Semântica e são relacionados à representação do conhecimento, seja ele sobre os conteúdos disponíveis na Web ou sobre conceitos de qualquer domínio do conhecimento.

As linguagens utilizadas para representar o conhecimento na Web Semântica (RDF, RDF/S e OWL) podem ser vistas como especializações da lógica de predicados que definem um subconjunto desta lógica, adequado para a construção de linguagens baseadas em *tags*. Pode-se dizer que OWL corresponde, grosseiramente, à lógica descritiva. Outro subconjunto existente desta lógica é um sistema de regras, também conhecido como cláusulas ou sentenças de Horn (ANTONIOU e HARMELEN, 2004).

A implementação computacional deste subconjunto da lógica de predicados torna possível a construção de programas capazes de raciocinar sobre fatos já conhecidos para inferir novos conhecimentos.

Com o objetivo de tornar mais representativas as linguagens utilizadas na Web Semântica, está sendo desenvolvida a linguagem SWRL (*Semantic Web Rule Language* – Linguagem de Regras para a Web Semântica), que é uma combinação das sublinguagens OWL-DL e OWL-Lite com a sublinguagem Unary/Binary Datalog RuleML (RULEML, 2005). A SWRL estende o conjunto de axiomas existentes na linguagem OWL para incluir cláusulas de Horn, possibilitando a construção de regras sobre o conhecimento descrito na ontologia através das classes, propriedades e instâncias. Esta linguagem é a base para construir os níveis da Web Semântica acima das ontologias, conforme figura 2, que são as regras e o *framework* lógico.

A sintaxe abstrata desta linguagem, que expande a sintaxe abstrata da linguagem OWL definida em PATEL-SCHNEIDER, HAYES e HORROCKS (2004), é definida juntamente com sua sintaxe baseada na linguagem XML, em (HORROCKS *et. al.*, 2004)

3.4.5 Considerações sobre Ontologias para a Web Semântica

A importância das ontologias para a representação do conhecimento fica clara quando se verifica a extensiva lista de projetos de pesquisa e publicações existentes sobre o assunto. Este é um nível fundamental da *Web Semântica*, pois ele é o responsável por fornecer os meios para que vocabulários padronizados possam ser definidos, servindo como base para que inferências complexas sejam realizadas e sistemas computacionais autônomos, desenvolvidos.

As ontologias representam o ponto mais elevado já atingido em termos de representação, compartilhamento e reutilização do conhecimento. Entretanto, para que a *Web Semântica* aconteça realmente (pelo menos até um certo ponto), é preciso construir programas capazes de trocar dados via *Web* e tomar decisões sem a intervenção constante dos humanos. São estas as características dos *web services* apresentados na próxima seção.

3.5 Web Services

Web service é uma tecnologia que se contrapõem às barreiras impostas pelas soluções proprietárias à interoperabilidade e integração de aplicações. São definidos por DACONTA, OBRST e SMITH (2003, p. 58) como “aplicações de *software* que podem ser descobertas, descritas e acessadas através do protocolo XML e de outros protocolos *Web* padrões em *intranets*, *extranets* e na Internet”.

Esta definição resume as principais características de um *web service*. Inicialmente, cita-se que um *web service* é uma aplicação de *software*, o que significa que ele nada mais é que um sistema computacional que executa determinadas funções. Na seqüência, aparece um ponto importante sobre os *web services*, o fato de eles serem descobertos, descritos e acessados através de protocolos padrões. É esta característica que os tornam ferramenta indispensável para a integração e a interoperabilidade entre sistemas, pois como é utilizado o padrão XML para a construção das mensagens e o protocolo HTTP para o envio das mesmas, qualquer *web service* é capaz de estabelecer comunicação com os demais para solicitar serviços ou responder a requisições.

A parte final da definição apresentada mostra que os *web services* não precisam ser

necessariamente públicos. Atualmente, existem muitos deles dedicados a executar tarefas nas redes fechadas das corporações, proporcionando uma economia significativa na execução de seus processos.

3.5.1 Componentes Básicos de um Web Service

Como a proposta dos *web services* envolve o conceito de autonomia na execução de suas tarefas, é importante a existência de uma estrutura capaz de possibilitar que isto aconteça. Na figura 17 podem ser observados os passos envolvidos na execução de um *web service*.

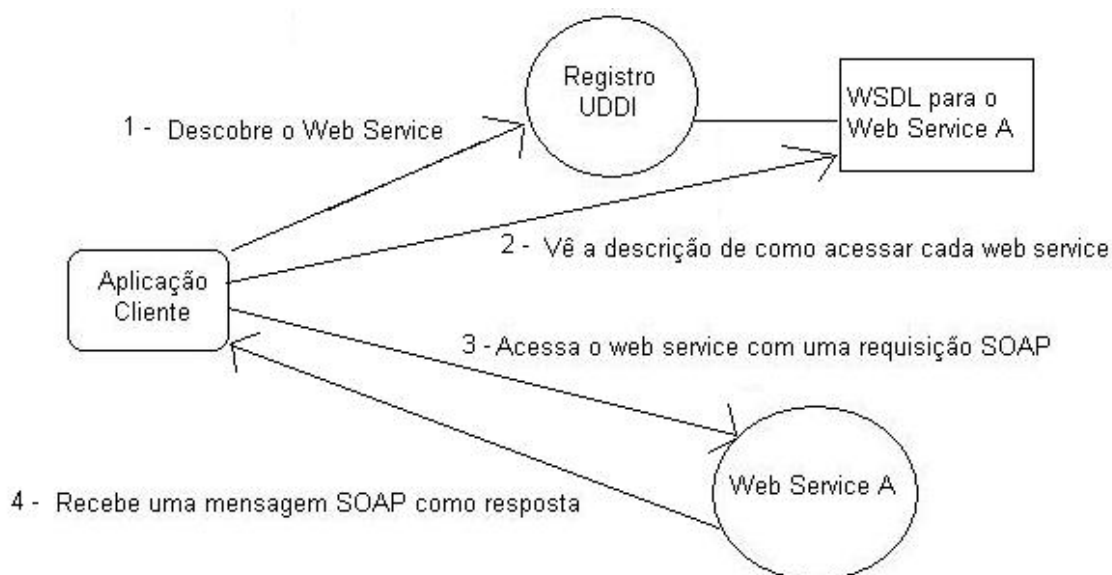


FIGURA 17 - Passos para acessar um *web service*

O processo é iniciado pela aplicação cliente, que deseja encontrar um *web service* que implemente as funções por ele desejadas. Caso a pessoa que está desenvolvendo o aplicativo saiba onde está tal *web service* e conheça as suas funções, ela facilmente pode fazer com que a aplicação cliente se comunique com ele. Entretanto, uma situação comum é quando se procura por *web services* apenas com base em suas características. Neste caso, se faz necessário o uso de um registro de *web services*. Uma das formas de registro de *web services* mais comuns é o UDDI (*Universal Description, Discovery, and Integration* – Descrição, Descoberta e Integração Universal), que funciona de forma semelhante a uma lista telefônica.

As organizações disponibilizam informações públicas sobre os seus *web services* no UDDI, que é pesquisado pelos interessados em encontrar estes sistemas para integrá-los às

suas aplicações. O processo de busca por um *web service* é ilustrado no passo 1, da figura 17. Ao encontrar o serviço desejado, o cliente tem acesso ao seu arquivo WSDL (*Web Service Definition Language* – Linguagem para Definição de *Web Services*) através do passo 2. A WSDL define a forma pela qual são descritas as informações operacionais sobre o serviço como: localização, o que ele faz e como pode ser acessado. Um arquivo WSDL é escrito em XML, o que o torna facilmente processável por qualquer aplicação. De posse do arquivo WSDL, a aplicação cliente é capaz de gerar a mensagem SOAP (*Simple Object Access Protocol* – Protocolo de Acesso a Objetos Simples) com a requisição a ser enviada ao *web service* (passo 3).

O SOAP é um protocolo, cujas mensagens também são escritas em XML, que define como as requisições feitas a um *web service* (passo 3) e as respostas que o mesmo envia ao cliente (passo 4) devem ser construídas. Dentre outras informações, uma mensagem SOAP contém o método que está sendo chamado, os parâmetros enviados e os valores obtidos como resposta.

3.5.2 Considerações Sobre Web Services para a Web Semântica

Esta rápida revisão sobre *web services* tem por objetivo mostrar a importância desta tecnologia para que a *Web Semântica* atinja o nível de autonomia desejado. O desenvolvimento de protocolos padrões para descrição e troca de mensagens entre os sistemas faz com que eles se tornem uma peça importante na construção de ferramentas que executem tarefas e tomem decisões sem a intervenção de um humano.

De todo o processo apresentado, pode-se considerar como mais rudimentar a primeira etapa, em que se faz a busca por um *web service*. Neste caso, ou a pessoa interessada já deve saber onde encontrar o serviço que necessita ou precisa realizar buscas por palavras-chave em catálogos de *web services*, de forma semelhante às pesquisas realizadas nos mecanismos de busca atuais. Seguindo o ensejo da *Web Semântica*, estudos estão sendo realizados sobre *web services* semânticos, que são o resultado da aplicação de ontologias no processo de descrição dos serviços para que buscas mais precisas possam ser efetuadas tanto por usuários humanos quanto por outros programas de computador (McILRAITH, SON e ZENG, 2001) e (BATTLE *et al.*, 2005).

3.6 Considerações

Por ser uma área bastante recente, a *Web Semântica* ainda desperta muita curiosidade

e ao mesmo tempo receio, mesmo nas pessoas ligadas à computação. Ainda não está bem clara a fronteira entre o que é, e o que não é possível fazer com as tecnologias existentes.

Com este capítulo, procurou-se apresentar a *Web Semântica* através da descrição das tecnologias que a suportam. Com o desenvolvimento do mesmo, pôde-se ter uma visão mais precisa desta fronteira, o que foi bastante útil para definir de forma mais clara os objetivos desta tese, principalmente no que tange as suas limitações.

A partir do estudo realizado, verificou-se que é possível minimizar vários problemas relacionados à EBW, no que tange a construção e disponibilização dos conteúdos, através da vinculação destas duas áreas. O estudo que levou à identificação dos problemas da EBW possibilitou a redação do capítulo 4, o qual descreve as principais características e os problemas existentes na EBW.

4. EDUCAÇÃO BASEADA NA *WEB*

Neste capítulo são apresentados conceitos relativos à Educação Baseada na *Web* (EBW). Inicia-se com uma breve revisão histórica da Educação a Distância (EAD). Em seguida, descrevem-se as tecnologias que foram, ou ainda estão sendo, incorporadas a ela. São abordados os conceitos de Sistemas Tutores Inteligentes, Hipermissão Adaptativa, Sistemas Gerenciadores de Aprendizagem e Objetos de Aprendizagem, os quais contribuíram para o surgimento de termos como Aprendizagem Mediada pelo Computador e Aprendizagem ou Educação Baseada na *Web*. Também são descritos padrões de metadados propostos com o objetivo de possibilitar a interoperabilidade entre sistemas de EBW, bem como modelos e sistemas desenvolvidos utilizando as tecnologias citadas, cujo objetivo vem ao encontro do assunto abordado nesta tese. Para finalizar o capítulo, apresenta-se um resumo sobre alguns trabalhos relacionados às ciências da aprendizagem, relevantes para a concepção do modelo e do sistema desenvolvido.

4.1 Histórico e Conceitos

O marco inicial da educação a distância não está claramente definido e provoca controvérsias entre os pesquisadores da área. Embora alguns autores remontem o surgimento da imprensa no século XV, como berço da EAD, de acordo com BITTENCOURT (2003), foi através das ferrovias e do correio nos EUA e na Europa, no século XIX, que as universidades e outras instituições de ensino puderam prover educação longe de suas sedes.

Este modelo foi rapidamente aceito pela grande parcela da população que estava à margem da educação e do acesso à informação nas grandes cidades. O processo de ensino-aprendizagem não presencial se consolidou ao longo do século XX, apresentando diferentes formas de suporte à transmissão destas informações. Livros, cartilhas e guias impressos constituem a primeira forma de disseminação de informação para a EAD. A televisão e o rádio foram inseridos neste processo na década de 70, assim como as fitas de áudio e vídeo, na década de 80 (LITWIN, 2001).

Nos anos 90, a Internet revolucionou o modo de fazer a Educação a Distância, com o surgimento da EBW. Através de seus recursos, não apenas a transmissão de conhecimento é possível, mas, também, um processo muito mais amplo que possibilita uma melhor formação do aluno, minimizando um problema crucial da EAD que era a falta de interatividade.

Dentre outros fatores que contribuíram para o desenvolvimento da EBW, MAIA (2001) destaca:

- A disseminação das tecnologias de comunicação e informação;
- A redução dos custos dos equipamentos de informática;
- A demanda crescente por educação continuada;
- O aumento da competitividade entre universidades, causado pela saturação do mercado;
- O hábito da aprendizagem por descoberta;
- O reconhecimento por parte da legislação brasileira da certificação de cursos de EAD, sejam eles oferecidos via *Web* ou não.

Os fatores acima descritos são esforços que se originaram em diferentes seguimentos e convergem para o objetivo comum de criar métodos cada vez mais práticos e eficientes de educação. Do governo vem o apoio através das leis que amparam a EAD como um todo, e, especificamente, a EBW; o mercado de trabalho contribui exigindo capacitação dos candidatos a novas vagas e formação continuada de seus funcionários; a área tecnológica, através do desenvolvimento de pesquisas, facilita o acesso aos recursos de informação e comunicação a um número cada vez maior de pessoas; as universidades, embaladas pela competitividade, desenvolvem projetos de EBW, visando atender o maior número possível de alunos; e, por fim, estes alunos sentem-se motivados pela característica de liberdade espaço-temporal que a EBW oferece.

Apesar do grande número de alunos frequentando cursos de graduação e pós-graduação a distância no Brasil, foi a educação corporativa a distância que teve os índices de evolução mais significativos nos últimos anos. Segundo MAIA (2001), o termo educação corporativa surge como um avanço às estratégias de treinamento de profissionais, em um momento em que as organizações não investem mais no aperfeiçoamento de seus profissionais de forma aleatória, mas sim visando à melhoria dos resultados em todas as áreas onde o investimento é realizado.

Ratificando sua posição, MAIA (2001) cita que existem duas mil universidades corporativas nos EUA, onde já surgiu uma discussão a respeito da sobreposição ou até mesmo substituição do ensino superior tradicional pelas universidades corporativas. No Brasil, são destacadas algumas empresas que tiveram sucesso com seus programas à distância tais como, Accor, Bank Boston, Motorola, Mc-Donald's, Visa, Alcoa, Siemens, Carrefour e Telemar.

Mesmo com tanto sucesso e números cada vez mais expressivos, a EBW é ainda vista com alguma desconfiança por muitos, em especial quando o assunto é a sua eficácia no processo de aprendizagem. Esta angústia é bem retratada quando LITWIN (2001, p. 14) diz: “Se os materiais utilizados substituem as aulas convencionais e estas nunca são suficientes para assegurar o êxito da aprendizagem, é difícil que um material o assegure”.

Corroborando esta afirmação, GADOTTI (2000, p. 255) diz que “as vantagens do ensino via Internet desaparecem diante do velho estudante que espera lições prontas. Ele precisa construí-las ou reconstruí-las (...) exigindo dele grande autonomia. Ele precisa saber o que quer e estar motivado para o curso escolhido”.

Partindo deste problema, professores e pesquisadores realizam estudos visando à criação de mecanismos que motivem e permitam que os alunos possam aprender de forma reflexiva e não apenas treine-os a repetir conteúdos decorados e a resolver problemas de forma automática.

Em um rápido resgate na área da psicologia da aprendizagem são encontradas duas correntes bastante conhecidas e difundidas no século XX. O Condutivismo, que teve seu maior expoente em Burrhus Frederick Skinner, defende que o aprendizado é mnemônico e estruturado a partir de práticas de repetição por parte dos alunos.

De acordo com seus seguidores, o conteúdo deve ser dividido em módulos seqüenciais e a passagem de um módulo para outro é condicionada à capacidade de resposta do aluno para questões elaboradas sobre o conteúdo do módulo visto. Muito utilizado nos anos 50 e 60, o condutivismo era bastante adequado para o ensino pelo computador, tendo em vista o seu caráter linear (MAIA, 2003).

Posteriormente, surgiu o Construtivismo, tendo Jean Piaget e Lev Semyonovich Vygotsky como seus idealizadores. Em linhas gerais, eles defendem que o aluno constrói o seu próprio conhecimento através de suas experiências com o ambiente e, mais especificamente no trabalho de Vygotsky, através das experiências vividas com um mediador (MAIA, 2003). O processo seqüencial e repetitivo do Condutivismo perdeu força perante as idéias construtivistas.

Assim como o Construtivismo é uma evolução do Condutivismo, o hipertexto é uma evolução dos sistemas de acesso linear. Ele, juntamente com outras técnicas computacionais, como a Inteligência Artificial e a Programação Multimídia, possibilitaram que o ensino via computador e, mais especificamente, a EBW fossem completamente remodelados, de modo a facilitar o processo de interação entre o aluno e o sistema, o aluno e o instrutor e entre os próprios alunos, com o intuito de motivá-los e facilitar o seu processo de aprendizagem.

Entretanto, a tarefa de ensino e motivação à distância é bastante complexa, pois como destaca LITWIN (2001, p. 24), “Em todos os processos e nas práticas de ensino e de aprendizagem (...) deverão participar pessoas com diferentes níveis de compreensão a respeito de um determinado campo de conhecimento ou de uma habilidade a ser ensinada”. Estes diferentes níveis de conhecimento exigem que cada pessoa seja tratada de forma diferenciada pelo sistema, no que diz respeito ao conteúdo que será abordado, bem como à forma de apresentação do mesmo. Esta diferenciação inclui a ordem de apresentação, a velocidade, os recursos utilizados (texto, imagens, vídeos, sons) e a forma de avaliação.

Com o objetivo de melhorar a interação entre os alunos e os *softwares* educacionais, sejam eles executados localmente ou à distância, foram criados os sistemas de educação adaptativa, que podem ser divididos em dois grandes grupos: os Sistemas Tutores Inteligentes e os Sistemas de Hipermídia Adaptativa.

4.2 Sistemas Tutores Inteligentes

Os Sistemas Tutores Inteligentes (STI) constituem uma área de pesquisa que objetiva criar sistemas de ensino com habilidades e conhecimentos semelhantes aos dos humanos. Sua principal meta é oferecer ao aluno um ambiente personalizado, tanto na forma quanto no conteúdo, superando uma das grandes dificuldades do ensino, seja pelos métodos tradicionais ou através de recursos computacionais, que é o tratamento igual para pessoas com conhecimentos e dificuldades diferentes. Isto fica claro quando SELF (1999) diz que “Sistemas Tutores Inteligentes são sistemas de ensino baseados em computador que tentam se adaptar às necessidades dos alunos”.

De acordo com SELF (1990, p. 2), “Historicamente, a pesquisa em STI começou dentro da Inteligência Artificial, mas os pesquisadores da IA retiraram-se da arena assim que começaram a perceber a necessidade de trabalhos mais fundamentados sobre modelos mentais, entendimento da linguagem, representação do conhecimento, etc., possibilitando

que outros entrassem em um campo intrinsecamente interdisciplinar”.

A capacidade de tratar cada um dos alunos que utilizam o sistema de uma forma diferenciada é o aspecto mais importante e que caracteriza um STI, pois, apesar das muitas dúvidas ainda existentes sobre o processo da aprendizagem humana, uma das certezas que já se tem é que as pessoas aprendem de maneiras diversas.

Os STI são programas que, ao interagirem com o aluno, modificam as suas bases de conhecimento (aprendem), percebem as intervenções do aluno e possuem a capacidade de adaptar as estratégias de ensino de acordo com o desenrolar do diálogo. Este tipo de programa de ensino permite que o conteúdo seja gerado sob demanda, além de possibilitar que a interação seja iniciada pelo aluno, através de perguntas, dividindo, assim, com ele o controle sobre a aprendizagem (MURRAY, 1999).

Segundo VICARI e GIRAFFA (2003), os STI foram projetados como uma tentativa de fazer com que os programas educacionais deixassem de ser meros “viradores de páginas” e se tornassem um elemento mais ativo no processo de ensino-aprendizagem. Para as autoras, a abordagem dos STI, mesmo que timidamente, vem ao encontro das modernas teorias de educação, que estão trocando o modelo centrado no professor, por um modelo centrado no aluno. Algumas diferenças entre um STI e os programas de instrução assistida pelo computador (CAI) criados desde a década de 50 são apresentadas no quadro 21.

Todas essas diferenças devem-se ao fato dos STI separarem três questões importantes: o que deve ser ensinado; para quem será ensinado e como será ensinado.

QUADRO 21 - Diferenças entre STI e CAI

Aspecto	CAI	STI
Origem	Educação	Ciência da Computação
Bases Teóricas	Skinner (behaviorista)	Psicologia Cognitivista
Estruturação e Funções	Uma única estrutura algorítmicamente pré-definida, onde o aluno não influi na seqüenciação.	Estrutura subdivida em módulos, cuja seqüenciação se dá em função das respostas dos alunos.
Estruturação do Conhecimento	Algorítmica	Heurística
Modelagem do Aluno	Avaliam a última resposta	Tentam avaliar todas as respostas do aluno durante a interação
Modalidades	Tutorial, exercício e prática, simulação e jogos educativos.	Socrático, ambiente interativo, diálogo bidirecional e guia.

Fonte: (VICARI e GIRAFFA, 2003).

A resposta da primeira questão define o tipo de conteúdo a ser ensinado pelo sistema

(Módulo ou Modelo de Domínio); a resposta à segunda, define para que tipo de aluno este conteúdo será ensinado (Módulo ou Modelo do Aluno) e, por fim, a resposta para a terceira, indica a maneira pela qual o conteúdo deve ser apresentado (Módulo ou Modelo do Tutor).

Estas respostas, segundo SELF (1999), formaram a base para a pesquisa em STI já na década de 70, e são apresentados por VICARI e GIRAFFA (2003) como a arquitetura clássica de um tutor inteligente, conforme figura 18.

A separação destes módulos possibilitou o avanço da área de STI ao separar o domínio de sua forma de manipulação, permitindo, assim, que as estratégias de ensino fossem relacionadas às informações de cada aluno. As seções a seguir descrevem de forma resumida cada um dos módulos de um STI.

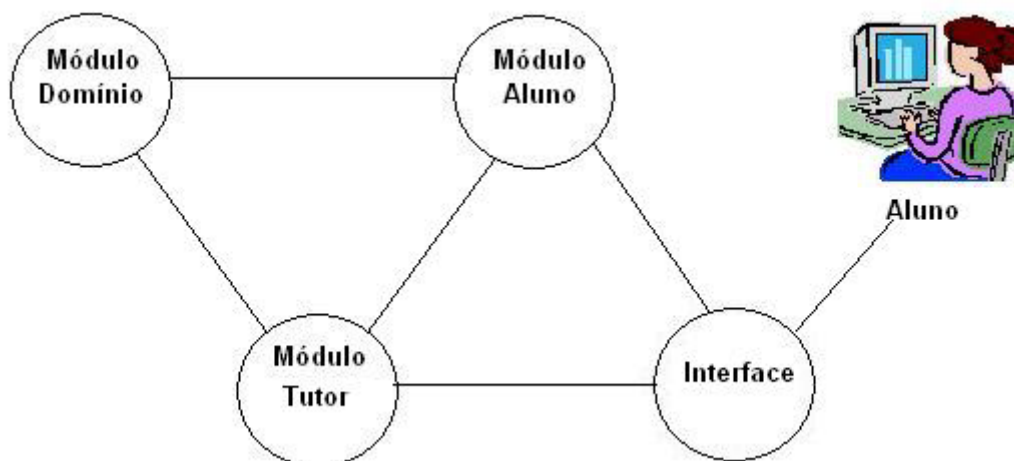


FIGURA 18 - Arquitetura clássica de um STI
Fonte: VICARI e GIRAFFA (2003, p. 169)

4.2.1 Modelo de Domínio

O modelo de domínio é constituído pelo material que descreve o conteúdo a ser apresentado aos alunos. Este conteúdo pode ser composto por textos, imagens, sons, exemplos, dicas, exercícios e desafios e, segundo VICARI e GIRAFFA (2003), podem ser armazenados em redes semânticas, *frames*, *scripts*, regras de produção, páginas HTML entre outros.

Todavia, esta definição de modelo de domínio não contempla todas as informações necessárias para a aprendizagem segundo os construtivistas, o que provoca o desdém dos mesmos em relação aos STI, por considerarem que o conhecimento não pode ser objetivamente definido e nem estaticamente representado.

As críticas construtivistas fizeram com que os pesquisadores em STI ampliassem a

função do modelo do domínio. Nos trabalhos mais recentes, este modelo, denominado modelo da situação, não deve contemplar apenas fatos, princípios e regras organizados hierarquicamente, ele também precisa conter a natureza das situações, contextos e interações que possam levar o aluno a construir o conhecimento sobre o assunto desejado (SELF, 1999).

4.2.2 Modelo do Aluno

“Neste modelo são armazenadas as características individuais do aluno (conhecimento individual sobre o domínio, por exemplo)” (GOULART e GIRAFFA, 2001, p. 5). Ele é constituído por dados estáticos e dinâmicos, que são de fundamental importância para o tutor poder comprovar hipóteses a respeito do conhecimento e do comportamento do aluno. A partir dos dados contidos neste modelo, o STI define a melhor estratégia de ensino para aquele aluno. Ter um modelo de aluno bem definido implica em uma constante atualização do mesmo, à medida que o sistema avalia a interação do aluno com o STI.

Entretanto, há uma grande dificuldade na construção deste modelo que se dá, em grande parte, por não se conhecer como a informação é processada no cérebro humano (VICARI e GIRAFFA, 2003).

As formas mais comuns para representar o modelo do aluno são:

- Modelo *Overlay*: define que o conhecimento do aluno é representado como um subconjunto do modelo de domínio. O objetivo do STI, neste caso, é fazer com que o modelo do aluno chegue o mais próximo possível do modelo de domínio;
- Modelo Estereótipo: nesta forma de representação são criados estereótipos, como por exemplo “iniciante”, “intermediário” e “avançado”. A cada aluno é associado um determinado estereótipo em relação a uma parte do modelo de domínio. Por exemplo, um aluno pode ser considerado “intermediário” em um determinado assunto e “iniciante” em outro. Para cada estereótipo existe uma forma de ensino relacionada no modelo do tutor.

Este módulo, de acordo com SELF (1999), é uma das características que definem um STI, pois ele permite que o sistema trate cada aluno de forma diferenciada. Sua relevância, segundo o autor, é comprovada pelo fato do termo “Modelagem do Usuário”, criado

originalmente pelos pesquisadores em STI, estar sendo amplamente utilizado em outras áreas, como a hipermídia adaptativa.

Também devido às críticas dos construtivistas, ampliou-se a abrangência do modelo do aluno que, atualmente, passou a ser chamado de modelo de interação e não deve mais contemplar apenas o conhecimento que um aluno possui. Ele também deve representar como o processo de aprendizagem acontece ao longo do tempo, através das interações entre as estruturas cognitivas do aluno, suas ações e o contexto em que elas ocorrem (SELF, 1999).

4.2.3 Modelo do Tutor

O modelo de tutor deve conter as estratégias de ensino que definem o conhecimento sobre como ensinar e como apresentar com sucesso determinado tópico a um aluno, a partir das informações de diagnóstico, monitoração e análise obtidas de seu modelo. Uma estratégia de ensino deve definir:

- Quando interromper? Que razões justificam interromper o curso de raciocínio e aprendizado do aluno?
- O que dizer? Esta questão desdobra-se em:
 - Seleção do(s) tópico(s) a ser(em) apresentado(s);
 - Ordenação destes tópicos, quando houver mais de um.
- Como dizer? Que é, provavelmente, a questão mais difícil de ser respondida devido à falta de teorias pedagógicas suficientemente detalhadas.

Assim como nos anteriores, o construtivismo teve grande influência no modelo do tutor, que agora não é mais responsável apenas por selecionar o conteúdo e as estratégias para sua apresentação. Este modelo passou a ser responsável por conduzir o aluno, com base em seus objetivos, através de espaços de interação proporcionados pelo sistema e foi denominado de modelo de permissões⁹ (SELF, 1999).

Como resultado das idéias construtivistas sobre a arquitetura clássica de um STI, SELF (1999) propôs uma nova configuração, onde os antigos modelos de domínio, aluno e tutor são vistos, respectivamente, como subconjuntos dos modelos de situação, interação e permissões, como mostra a figura 19.

⁹ A tradução do termo foi retirada de GOULART e GIRAFFA (2001).

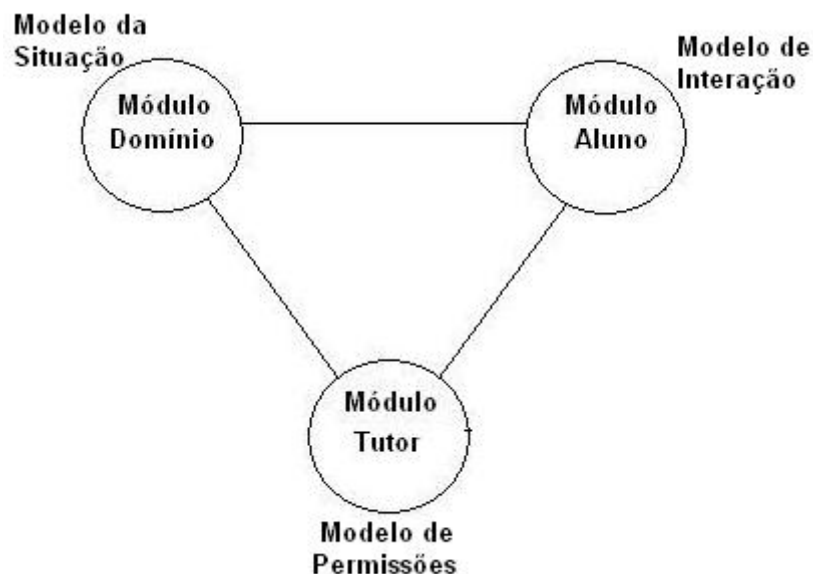


FIGURA 19 - Expansão da arquitetura clássica de um STI, por SELF (1999)
 Fonte: (GOULART e GIRAFFA) 2001

4.2.4 Classificação de Sistemas Tutores Inteligentes

A construção de um STI é uma tarefa que demanda tempo e recursos, por exigir o envolvimento de uma equipe interdisciplinar. Com o intuito de facilitar este processo, muitas ferramentas de autoria têm sido disponibilizadas para que o desenvolvimento de um STI possa ser mais rápido, menos oneroso e realizado por pessoas sem conhecimento em programação de computadores.

Uma classificação dos tipos de STI que podem ser desenvolvidos com as ferramentas existentes é apresentada por MURRAY (1999)¹⁰, que os divide em sete categorias. Abaixo, são relacionadas apenas as cinco categorias consideradas mais relevantes para o desenvolvimento desta tese. São elas:

- Planejamento e seqüenciamento de currículos: tem como principal característica a divisão do conteúdo em unidades instrucionais, que são organizadas hierarquicamente e possuem relacionamentos do tipo “é pré-requisito para” ou “faz parte de”. O papel dos tutores desta categoria consiste apenas em definir a melhor seqüência de apresentação das informações, sendo apropriados para ensinar conteúdos declarativos, conceituais e não para assuntos que envolvem o desenvolvimento da habilidade de resolver problemas;

¹⁰ Apesar do trabalho já ter sido publicado há sete anos, não foi encontrado nenhum avanço significativo na literatura que o tornasse obsoleto.

- Estratégias de tutoria: são similares aos STI da categoria anterior na forma de representar o conteúdo. Entretanto, diferem na profundidade das interações instrucionais que, nestes, vão além da definição da seqüência de apresentação do conteúdo, estabelecendo, também, quando e como apresentar explicações, resumos, exemplos, analogias, exercícios e dicas. Apesar de aumentar a interação entre o aluno e o STI, os usuários de sistemas desta classe ainda aprendem muito mais lendo e pensando do que fazendo;
- Simulação de dispositivos e treinamento em equipamentos: a principal diferença entre esta categoria e as anteriores é que nesta os alunos “aprendem fazendo”, em geral, através de simulações, o que implica em uma interface muito mais preparada para apresentar as informações ao aluno e capturar o retorno do mesmo. Como estes sistemas, em geral, são aplicados em treinamentos de recursos específicos, em que o processo de construção do conhecimento é mais fácil, seu modelo de tutor não é tão robusto como o dos sistemas das categorias anteriores;
- Sistemas especialistas e tutores cognitivos: nesta classe os alunos resolvem problemas e subproblemas visando um objetivo enquanto o seu conhecimento é comparado ao conhecimento do especialista. Quando eles divergem, o tutor interfere no processo explicando os conceitos não compreendidos. Este tipo de STI tem uma representação bastante aprofundada do modelo do domínio, mas peca na personalização por ter um modelo de aluno e de tutor muito superficial;
- Hipermídia Inteligente/Adaptativa: as funções deste tipo de STI se sobrepõem às dos tutores das duas primeiras categorias (Planejamento e seqüenciamento de currículos e Estratégias de Tutoria), com a diferença de que eles precisam gerenciar os *hyperlinks* e facilitar a navegação do aluno pelo espaço instrucional.

Como descrito nos itens acima, os STI têm um enfoque maior em alguns de seus módulos, dependendo da categoria em que se enquadram. Entretanto, todos precisam possuir os modelos do aluno, tutor e domínio para serem considerados STI.

4.3 Hipermissão Adaptativa

A hipermissão adaptativa (HA) é uma área de pesquisa relativamente nova, formada pela união da hipermissão com as técnicas de modelagem do usuário. Ela surgiu devido às limitações das páginas de hipermissão estáticas, que apresentavam o mesmo conteúdo a todas as pessoas.

Os Sistemas de Hipermissão Adaptativa (SHA) constroem um modelo dos objetivos, preferências e conhecimentos de cada usuário, a ser utilizado durante as interações com o mesmo, para que o sistema possa se adaptar às suas necessidades. Eles podem ser aplicados em qualquer área do conhecimento, em que o sistema seja utilizado por pessoas com diferentes objetivos e conhecimentos e onde o hiperespaço¹¹ seja razoavelmente grande.

De acordo com BRUSILOVSKY (2001), a pesquisa em HA pode ser dividida em duas épocas: até 1996 e após 1996. Os primeiros SHA surgiram no início da década de noventa, quando as áreas de hipermissão e modelagem do usuário atingiram um certo grau de maturidade. Nessa época, as aplicações desenvolvidas estavam confinadas em laboratórios e visavam apenas demonstrar que uma determinada técnica era viável.

A partir de 1996, devido à expansão da Internet e ao embasamento teórico obtido nos primeiros anos de pesquisa, a HA passou a ser utilizada em aplicações reais nas mais diversas áreas, as quais, segundo BRUSILOVSKY (2001), podem ser divididas em seis grandes grupos:

- Hipermissão Educacional;
- Sistemas de Informação *on-line*;
- Sistemas de ajuda *on-line*;
- Sistemas de recuperação de informação em hipermissão;
- Hipermissão Institucional;
- Sistema para gerenciamento de visões personalizadas em espaços de informação.

É, sem dúvida, na hipermissão educacional que estão as maiores pesquisas atualmente, devido ao poder que ela apresenta na personalização das interfaces, tornando-se uma peça importante na construção de STIs utilizados via *Web*.

¹¹ Define-se hiperespaço como o espaço de informações em sistemas de hipermissão

A figura 20 apresenta como ocorre o processo de adaptação, tendo como base o modelo do usuário. Inicialmente, o sistema coleta dados sobre o usuário através de questionários ou da análise de sua interação. Em seguida, são processadas estas informações (modelagem do usuário), gerando um modelo que descreve as características daquele usuário. Por fim, as informações constantes no modelo do usuário são processadas pelo sistema para definir a forma de apresentação dos dados (adaptação).

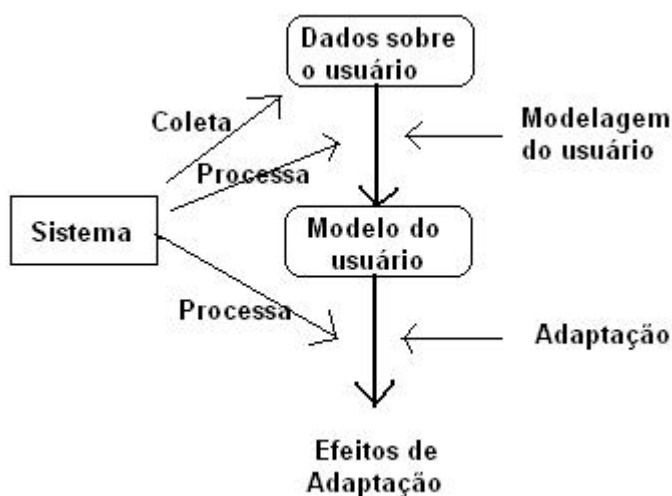


FIGURA 20 - Modelo de adaptação com base na modelagem do usuário.
Fonte: BRUSILOKSV (1996)

A adaptação nos sistemas hipermídia, segundo OLIVEIRA e FERNANDES (2002), ajuda a resolver quatro problemas:

- Quebra de fluxo conceitual: nas estruturas com muitos *links*, nem sempre é possível garantir que o usuário tenha passado por todos os nós que possuam uma relação de pré-requisito com o atual, dificultando o entendimento dos conceitos apresentados;
- Desorientação: ocorre nas estruturas complexas de *links*, onde o usuário tem dificuldade para saber como chegou a um determinado nó, como voltar ou como chegar onde deseja;
- Sobrecarga cognitiva: está relacionada à sobrecarga de informação apresentada ao usuário, seja ela em termos de conteúdos ou de *links*.
- Quebra de fluxo narrativo: nem sempre a passagem de um nó para outro apresenta um fluxo suave como esperam os usuários;

Segundo BRUSILOVSKY (1996), para resolver problemas como os previamente citados, os sistemas adaptativos atuam em duas grandes áreas, a Apresentação Adaptativa e a Navegação Adaptativa. Em BRUSILOVSKY (1996) são apresentadas as tecnologias que dão suporte a estas duas grandes áreas, as quais são revistas e atualizadas em BRUSILOVSKY (2001), dando origem à taxionomia apresentada na figura 21.

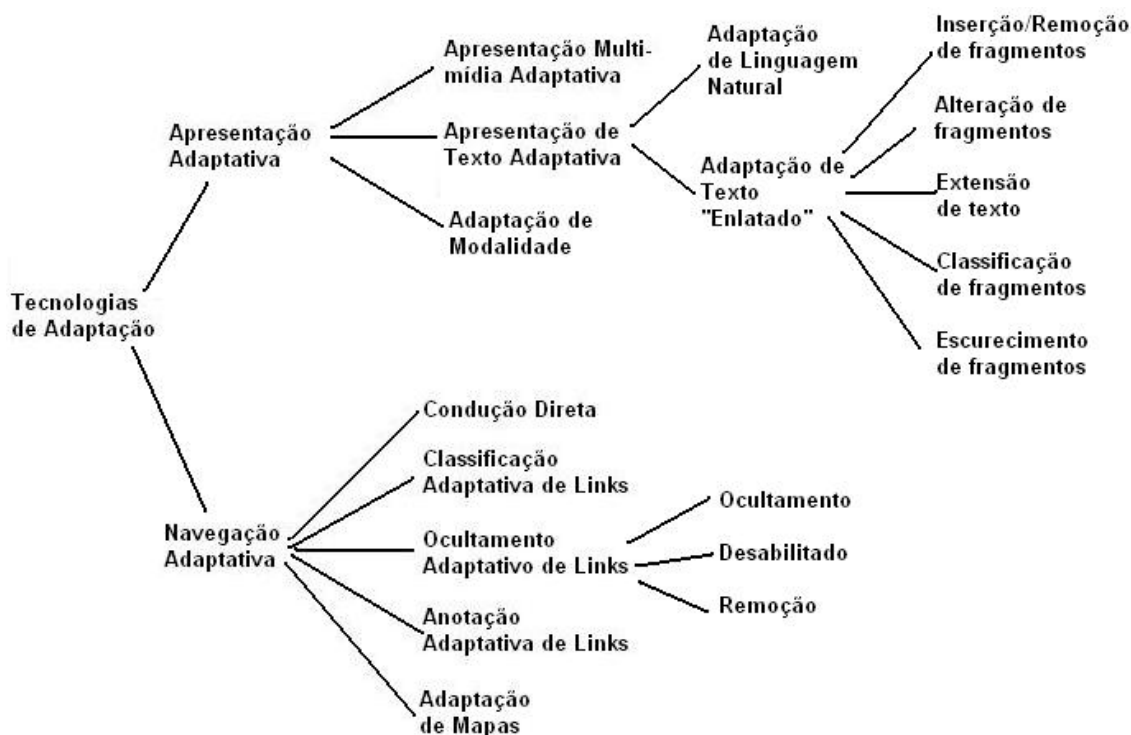


FIGURA 21 - Tecnologias de adaptação em Hipermídia Adaptativa
Fonte: Adaptado de BRUSILOVSK (2001)

4.3.1 Apresentação Adaptativa

As tecnologias relacionadas à apresentação adaptativa visam personalizar a forma pela qual a interface do sistema mostra o conteúdo, de acordo com o conhecimento, os objetivos e outras características do usuário. Um sistema educacional, por exemplo, pode mostrar uma página com mais detalhes a um usuário com maior experiência no assunto, enquanto mostra outra mais simples e com explicações adicionais a um iniciante.

Como pode ser visto na figura 21, a apresentação adaptativa divide-se em três linhas: apresentação adaptativa de multimídia, apresentação adaptativa de texto e adaptação de modalidade, sendo a adaptação de texto a mais conhecida e utilizada.

Dentre outros métodos que possibilitam a apresentação adaptativa, BRUSILOVSKY (1996) cita:

- Explicações adicionais: seu objetivo é apresentar ou esconder fragmentos de informação que são ou não relevantes ao nível de conhecimento do usuário;
- Explicações de pré-requisitos: a idéia deste método é fazer com que o usuário tenha acesso a todos os pré-requisitos que ainda não foram bem compreendidos por ele para o entendimento de um conteúdo;
- Explicações comparativas: quando um conceito similar ao que está sendo apresentado já é conhecido pelo usuário, o sistema apresenta as semelhanças e diferenças entre eles para facilitar o entendimento;
- Explicações variantes: este método parte do princípio que apenas apresentar, ocultar ou alterar a ordem de apresentação dos textos não é suficiente. Por isto, ele trabalha com diferentes explicações sobre um mesmo assunto e o usuário tem acesso àquela que o sistema julgar mais adequada ao seu perfil;
- Classificação: o objetivo é organizar os fragmentos de código sobre um conceito, onde as informações consideradas mais relevantes, de acordo com o modelo do usuário, são colocadas na frente.

4.3.2 Navegação Adaptativa

A navegação adaptativa propicia uma orientação ao usuário do sistema, seja em nível local ou global. Além disto, ela possibilita que eles tenham acesso a visões diferentes do sistema, de acordo com as suas características pessoais. Os métodos destacados por BRUSILOVSKY (1996) que possibilitam a navegação adaptativa são:

- Condução global: para aplicar este método o sistema, primeiramente, deve conhecer o objetivo do usuário. Em seguida, o sistema guia o usuário pelo caminho mais curto que pode levá-lo a este objetivo, sempre, é claro, considerando o conhecimento que ele já possui;
- Condução local: é uma forma mais modesta de fazer a condução do aluno se comparada ao método anterior, porque ela determina apenas o próximo nó que deve ser visitado de acordo com o conhecimento do usuário, mas sem considerar um objetivo maior;
- Orientação global: o objetivo deste método é fazer com que o usuário tenha uma visão geral do hiperespaço e de sua posição dentro dele. Isto pode ser proporcionado pela apresentação de mapas, ou através de tecnologias de ocultação e explanação;

- Orientação local: também visa auxiliar o usuário a entender o hiperespaço, porém se concentra apenas naquelas partes que são acessíveis do ponto onde ele se encontra;
- Gerência de visões personalizadas: tem como meta proteger o usuário da quantidade excessiva de informações, apresentando a ele apenas as relevantes as suas características, interesses e conhecimentos.

Algumas das técnicas utilizadas para implementar os métodos de apresentação e navegação adaptativa são apresentadas em BRUSILOVSKY (1996), juntamente com uma série de trabalhos que aplicam cada uma delas.

4.3.3 Modelagem do Usuário em Hiperídia Adaptativa

Como mostra a figura 20, as tecnologias de adaptação previamente apresentadas baseiam-se em um modelo do usuário para possibilitar a adaptação do sistema as suas necessidades. A construção deste modelo é feita de forma semelhante a outras áreas que também fazem uso deste recurso, como os STI.

As informações consideradas na construção de um modelo do usuário podem ser divididas em três categorias:

- Dados do usuário: seus objetivos, conhecimentos, experiências, interesses e características pessoais (fatores de personalidade, fatores cognitivos e estilos de aprendizagem).
- Dados de utilização: a forma pela qual o usuário interage com o sistema;
- Dados do ambiente: este indicador, que surgiu devido à diversidade da *Web*, deve levar em consideração fatores como plataforma do cliente, velocidade do *link* de acesso, localização, entre outros.

Na grande maioria dos SHA, a tarefa de definir as informações do modelo do usuário é dividida entre ele e o sistema. Entretanto, um dos objetivos das pesquisas realizadas nesta área é a criação de mecanismos que possibilitem a construção de forma automática de modelos cada vez mais precisos, requerendo, assim, o mínimo de intervenções diretas do usuário.

Esta tarefa, segundo BRUSILOVSKY (2001), é facilitada em sistemas educacionais, pois o modelo do usuário é integrado ao modelo do aluno, que contém várias informações

sobre ele e é, constantemente, atualizado com base nas respostas do aluno aos exercícios, testes e outras atividades.

4.4 Sistemas Gerenciadores de Aprendizagem e Objetos de Aprendizagem

O crescente número de usuários da *Web*, principalmente na segunda metade dos anos 90, fez com que ela fosse rapidamente integrada aos antigos sistemas de EAD baseados em materiais impressos e fitas de áudio e vídeo, substituindo-os e tornando-se uma das formas mais comuns para disponibilizar o conteúdo aos participantes.

Devido à euforia desta nova possibilidade, os primeiros sistemas de aprendizagem baseados na *Web* foram desenvolvidos sem um planejamento adequado, e eram adaptações diretas dos programas instrucionais até então distribuídos em CD-ROMs. Neste período, era necessário criar uma estrutura completa para cada novo curso, devido à impossibilidade de reutilizar recursos e serviços em diferentes cursos.

Com o passar do tempo, surgiu a necessidade de melhorar o processo de construção e gerência dos sistemas de aprendizagem baseados na *Web*, bem como construir estruturas capazes de executar funções básicas que pudessem ser utilizadas em diferentes cursos. Diante destes desafios e do amadurecimento das equipes responsáveis pela construção dos sistemas, surgiu o conceito de Sistemas Gerenciadores de Aprendizagem (LMS – *Learning Management System*).

LMS são sistemas especializados com tecnologias para o ensino e aprendizagem baseados na Internet e na *Web*, que visam fornecer educação e treinamento seguindo o paradigma de educação a distância (AVGERIOU, 2003). Eles possibilitam que os professores desenvolvam o material instrucional, os exercícios e testes, além de facilitarem a comunicação com os alunos. Já para os alunos, um LMS deve possibilitar o acesso ao material disponibilizado e fornecer ferramentas de comunicação com seus colegas e com os professores e monitores.

O surgimento dos LMS propiciou um desenvolvimento ainda mais rápido da EBW, por não serem concebidos para um curso ou área específica. O que eles oferecem é apenas um arcabouço que deve ser preenchido com o conteúdo a ser ministrado, facilitando, assim, a sua reutilização. O sucesso dos LMS é destacado por NAGY (2004), quando afirma que o setor de tecnologia da EBW é dominado pelos fornecedores destes sistemas.

A busca por tecnologias capazes de facilitar o processo de desenvolvimento de ambientes e conteúdos de EBW fez com que fosse desenvolvido, paralelamente aos LMS,

o conceito de objetos de aprendizagem (LO – *Learning Objects*). Segundo LTSC (2004), LO são definidos como entidades, digitais ou não, que podem ser utilizadas, reutilizadas e referenciadas durante a aprendizagem auxiliada por computador. Um LO pode ser constituído de conteúdos instrucionais (texto ou multimídia), objetivos de aprendizagem, *softwares* instrucionais, pessoas, ferramentas, entre outras entidades referenciadas durante o processo de ensino-aprendizagem.

De acordo com LONGMIRE (2000), algumas características são comuns a todos os LO, independente de sua aplicação, são elas:

- Modularidade: um LO é uma pequena parte de um curso, que pode ser formada por outros LO;
- Flexibilidade - Como os LO são construídos para possuírem início, meio e fim bem definidos, eles são flexíveis e podem ser reutilizados sem nenhum tipo de manutenção;
- Aumento de significância do conhecimento – com a reutilização de um LO várias vezes e em diferentes situações, seu conteúdo é consolidado naturalmente à medida que o tempo passa;
- Interoperabilidade – a criação de padrões para armazenar LO permite a sua reutilização em diferentes sistemas de aprendizagem ou até em outros tipos de aplicações;
- Indexação e busca - estas tarefas também são facilitadas com a padronização das descrições dos LO.

Sem dúvida, a grande vantagem dos LO é a sua possibilidade de reutilização, pois uma vez criados e disponibilizados, eles podem ser utilizados por qualquer LMS que tenha suporte à busca e vinculação destes recursos. Desta forma, a criação de um curso à distância tornou-se uma tarefa mais fácil, uma vez que o professor não precisa preparar todo o material, bastando integrar ao seu curso os LO que o interessam.

4.5 Problemas da Educação Baseada na Web

Analisando as tecnologias apresentadas nas seções anteriores, fica evidente a presença de dois grupos distintos de pesquisadores, com objetivos diferentes, sobre a EBW. Enquanto o propósito dos pesquisadores do primeiro grupo é tornar mais produtiva a aprendizagem dos estudantes através dos STI e SHA, o segundo grupo visa facilitar a

construção e a disponibilização de cursos e materiais educacionais por meio dos LMS e LO. Esta seção discute alguns dos principais problemas relacionados às tecnologias desenvolvidas pelos dois grupos.

Os benefícios derivados da pesquisa em STI para o ensino através do computador são muitos, apesar de todas as dúvidas e contradições existentes sobre o processo de aprendizagem. Entretanto, ainda há um vasto campo a ser explorado em dois caminhos diferentes: o primeiro está relacionado ao custo e às dificuldades para se desenvolver um STI; e o segundo, à melhoria da qualidade do ensino.

Sua principal contribuição para a EBW é a arquitetura dividida em modelo do domínio, modelo do aluno e modelo do tutor. Contudo, uma grande dificuldade permanece, mesmo após esta divisão, porque os modelos do aluno e do tutor ainda são fortemente dependentes do modelo do domínio. Esta dependência dificulta a reutilização de cursos inteiros ou de suas partes, estratégias de ensino e informações sobre o estudante, tornando a construção de um STI lenta e cara.

Outra questão importante, é que profissionais ligados à educação não acreditam na eficiência de STI, e de qualquer outro sistema de EBW, pois, segundo eles, falta uma fundamentação mais consistente nas teorias de aprendizagem. A falta de suporte nestas teorias já é um problema grave quando os STI são criados por pessoas que passam meses estudando para modelarem adequadamente o domínio, as estratégias de tutoria e as informações sobre os alunos. Porém, ele é ainda pior quando ferramentas computacionais sem um embasamento adequado nas teorias já citadas são empregadas na construção dos STI, reutilizando conteúdos e estratégias, com o objetivo de reduzir o custo e o tempo de desenvolvimento.

Alguns problemas existentes nos STI e em seu processo de desenvolvimento são listados por MIZOGUCHI e BOURDEAU (2000):

- Existe uma grande lacuna conceitual entre autores e ferramentas de autoria;
- As ferramentas de autoria existentes não são inteligentes nem amigáveis;
- O Conhecimento e os componentes existentes em STI são raramente compartilhados e reutilizados;
- A maioria dos STI não considera os resultados das pesquisas nas ciências da aprendizagem.

Com relação aos SHA, vale ressaltar que eles sofrem dos mesmos problemas dos

STI, pois as técnicas adaptativas e os modelos do usuário são muito dependentes do domínio em questão. Além disto, a inicialização do modelo do aluno ou usuário ainda apresenta muitas dificuldades, pois o sistema não dispõe de informações suficientes sobre ele, o que o obriga a solicitá-las diretamente, através de um questionário.

De acordo com as informações apresentadas, está claro que STI e SHA ainda apresentam muitas limitações. Enquanto a falta de embasamento nas ciências da aprendizagem parece ser o principal obstáculo a ser superado do ponto de vista educacional, a dificuldade para reutilizar informações durante a construção dos sistemas é o principal aspecto a ser tratado do ponto de vista computacional.

Em ambos os casos, a *Web Semântica* vem se consolidando como uma alternativa viável e promissora, por facilitar o processo de desenvolvimento, além de maximizar o aprendizado do aluno, através de melhorias no processo de modelagem do aluno, do tutor e do domínio.

Facilitar o processo de desenvolvimento de sistemas de EBW adaptativos requer o uso de ferramentas de autoria, as quais já existem em grande número atualmente, mas ainda apresentam muitas limitações. Limitações, estas, que podem ser reduzidas quando se usam ontologias e taxionomias para representar os modelos de domínio, tutor e aluno. Já para maximizar a aprendizagem, é necessário que os sistemas tenham mais alternativas para representar o conhecimento dos seus modelos. Uma proposta que visa aumentar a capacidade destes sistemas é a modelagem de cada um de seus módulos através de um ou mais agentes inteligentes, como mostra GOULART e GIRAFFA (2001).

Alguns avanços já foram obtidos com estas pesquisas, dando sentido à seguinte pergunta: “Os resultados não serão ainda melhores se os módulos de um sistema de EBW, além de serem modelados como agentes, tiverem a capacidade de interagir com outros agentes para trocar informações de forma autônoma?”. A resposta para esta pergunta passa pelo desenvolvimento da *Web Semântica*, mais especificamente de metadados, ontologias e *web services* voltados à educação, que possibilitem a vinculação dos STI e SHA com os LMS e LO, os quais individualmente também apresentam várias limitações, para facilitar a construção dos cursos.

O surgimento dos LMS e LO, ao mesmo tempo em que tornou possível disponibilizar e gerenciar vários cursos simultaneamente e facilitou a construção de materiais educacionais, tornou a situação mais complicada em alguns aspectos. Entre outros se destacam a falta de suporte pedagógico para a construção dos cursos e a falta de adaptação dos conteúdos às necessidades do estudante.

Em geral, os estudantes sentem-se frustrados e desmotivados pelos LMS. Contudo, apesar destes problemas, eles são claramente dominantes em relação aos STI e SHA em números de sistemas desenvolvidos e utilizados em larga escala, o que justifica as pesquisas que objetivam melhorar a fundamentação nas teorias de aprendizagem dos LMS e torná-los capazes de adaptar os conteúdos de acordo com as características e necessidades de cada aluno.

Os problemas e implicações da forma padrão de apresentação do conteúdo aos alunos, os altos custos de desenvolvimento e a baixa reutilização de material, são amplamente discutidos e documentados nos anais do *workshop* “Em Direção aos Sistemas Gerenciadores de Aprendizagem Inteligentes”, realizado em conjunto com a conferência sobre Inteligência Artificial na Educação (ILMS, 2003). Neste *workshop* são propostas soluções para estes problemas baseadas na fusão dos LMS com STI e SHA, criando os Sistemas Gerenciadores de Aprendizagem Inteligentes (ILMS – *Intelligent Learning Management System*) (SIMIC, GASEVIC, DEVEDZIC, 2004) e (BRUSILOVSKY, 2003).

4.6 Padrões de Metadados para Educação Baseada na Web

A união entre STI, HA, LMS e LO ainda não teve tempo suficiente para consolidar suas idéias e mostrar seus resultados e já está sendo perpassada pelas novas tendências criadas com a *Web* Semântica. Muitos trabalhos têm sido dedicados a esta área de pesquisa como pode ser visto em ET&S (2004).

A sigla LO (*Learning Object*) já recebeu mais uma letra, dando origem à sigla LOM (*Learning Object Metadata* - Metadados para Objetos de Aprendizagem), que é definida por LTSC (2004) como um padrão que especifica a sintaxe e a semântica dos atributos necessários para descrever completa e adequadamente um objeto de aprendizagem.

Por meio dos metadados pretende-se resolver alguns problemas relacionados à reutilização de conteúdo, que ainda permanecem mesmo após o surgimento dos LMS e LO. Por exemplo, um professor que deseja construir um curso a partir de materiais já existentes, deve saber quais LO pretende utilizar e onde eles estão armazenados, porque o modo de estruturação das informações na *Web* não permite a execução de buscas precisas. Através dos metadados associados aos LO, segundo KOPER (2004), é possível criar estruturas realmente adaptáveis e reutilizáveis devido as suas representações semânticas.

Com o passar do tempo, diversos padrões foram construídos por diferentes organizações com o objetivo de descrever informações referentes ao processo educacional

baseado na *Web*. Além de descrever LO individualmente, estes padrões definem estruturas de metadados para atividades de aprendizagem, objetivos de aprendizagem, informações sobre alunos e até para cursos inteiros, inclusive com regras que definem diferentes formas de navegação e seqüenciamento do conteúdo, com vistas à construção de cursos adaptativos baseados em tais padrões. A seguir, são apresentadas as principais características dos padrões utilizados nesta tese.

4.6.1 O Padrão IEEE 1484.12.1 sobre Metadados para Objetos de Aprendizagem

Padrões para descrever LO estão sendo criados para que eles possam ser mais facilmente encontrados, utilizados e reutilizados através da *Web*. Um destes padrões é o 1484.12.1 IEEE *Standard for Learning Object Metadata* (LOM) (LTSC, 2002), que especifica um esquema conceitual com a definição da estrutura de uma instância de metadados para um LO isoladamente. Neste padrão, os elementos que descrevem um LO são divididos em nove categorias. A seguir é feito um breve relato dos objetivos de cada uma delas, com a descrição de seus principais elementos.

4.6.1.1 Geral (General)

Os elementos desta categoria são responsáveis por identificar o objeto de aprendizagem. Entre outras informações, eles trazem seu título; o esquema de catálogo utilizado para cadastrar o objeto (por exemplo, ISBN, ou um *namespace*) e o seu valor; a linguagem (idioma) utilizada pelo LO para apresentar suas informações; uma descrição e as palavras-chave do conteúdo; sua estrutura e nível de agregação.

Alguns elementos desta e das demais categorias, como título, descrição e palavras-chave, podem receber quaisquer valores digitados pelo usuário, pois o padrão não define um conjunto de valores possíveis para eles. Outros elementos, porém, possuem um conjunto de valores possíveis já especificados pelo próprio padrão, como é o caso do elemento que representa a estrutura do LO (*structure*), cujos valores possíveis são:

- *Atomic* - indica que aquele LO é indivisível;
- *Collection* – quando o LO é formado por um conjunto de LO sem nenhum relacionamento entre eles;
- *Networked* – um conjunto de LO com relacionamentos não especificados;
- *Hierarchical* – um conjunto de LO cujos relacionamentos podem ser representados por uma árvore;

- *Linear* – um conjunto de LO completamente ordenados (relacionamentos de próximo e anterior).

Outro exemplo de elemento com vocabulário definido é o nível de agregação (AggregationLevel) de um LO, que recebe um valor entre 1 e 4. O número 1 indica que aquele é um LO simples, como por exemplo uma figura, uma *homepage* ou parte dela. O nível 2 é composto por vários componentes do nível 1, assim como LO de nível 3 são formados por vários LO de nível 2 e, por fim, um objeto de nível 4, que em geral representa um curso ou algo que valha um certificado, engloba vários LO de nível 3.

4.6.1.2 Ciclo de Vida (Life Cycle)

Esta categoria fornece informações sobre o estado de desenvolvimento do LO. Dentre outras, destacam-se o *status* do LO (indisponível, rascunho, revisado ou final), as pessoas que contribuíram para o desenvolvimento do mesmo e as alterações realizadas.

Estas informações podem ser utilizadas por uma pessoa ou sistema que esteja buscando LO para montar um curso, que busque priorizar aqueles que possuam um estado de desenvolvimento mais avançado e tenham recebido contribuições positivas

4.6.1.3 Meta Metadados (Meta-Metadata)

Esta categoria visa descrever dados sobre os metadados do LO, tais como: quem criou determinado metadado e quando isto foi feito.

4.6.1.4 Técnicas (Technical)

A quarta categoria contempla descrições técnicas do LO como Formato (Ex.: vídeo/mpeg, text/html, image/gif, etc.), tamanho em *bytes*, localização (em geral uma URL), sistemas operacionais suportados e tempo de duração (aplicável para vídeos e sons).

4.6.1.5 Educacionais (Educational)

Nesta categoria encontram-se as informações mais importantes para as pessoas envolvidas na qualidade do processo educacional. Devido à relevância de seus elementos, eles são descritos separadamente.

- Tipo de Interatividade (*Interactivity Type*) – determina o modo predominante de aprendizagem suportado pelo LO. Seus valores são:

Active, ou aprender fazendo, neste caso o LO faz com que o aluno participe do processo de aprendizagem interagindo com simulações, respondendo perguntas e resolvendo problemas; *Expositive*, quando o aluno deve absorver as informações apenas através de leituras; e *Mixed*, quando o LO envolve tanto a aprendizagem ativa quanto a expositiva;

- Tipo de Recurso de Aprendizagem (*Learning Resource Type*) – determina o tipo do recurso dentre uma das seguintes opções: exercício, simulação, questionário, diagrama, figura, grafo, índice, slide, tabela, texto narrativo, exame, experimento, definição de problema, auto-avaliação e aula. Como um mesmo LO pode possuir diferentes tipos de recursos, esta propriedade pode receber uma lista de valores ordenados crescentemente, de acordo com a sua representatividade;
- Nível de Interatividade (*Interactivity Level*) – representa o grau de interatividade que um LO fornece a um aluno, seus valores são: muito baixo, baixo, médio, alto e muito alto. É importante observar que esta propriedade deve estar relacionada ao tipo de interatividade, pois não faz sentido definir que um LO do tipo *expositivo* tenha um nível de interatividade *muito alto*;
- Densidade Semântica (*Semantic Density*) – indica o grau de concisão do LO e seus valores são iguais aos do nível de interatividade. Quanto mais resumido for o material e mais informações ele contiver, maior será a sua densidade semântica. Deve-se tomar cuidado para não confundir densidade semântica com dificuldade;
- Usuário Final Pretendido (*Intended End User Role*) – define para que tipo de usuário o LO foi desenvolvido. Seus valores podem ser: professor, autor, aluno e gerente;
- Contexto (*Context*) – é responsável por indicar o principal ambiente onde o LO deve ser aplicado. Pode receber os seguintes valores: escola, educação superior, treinamento e outros;
- Faixa Etária (*Typical Age Range*) – este elemento estabelece a faixa etária dos usuários pretendidos para o LO. Pode ser feito pela definição das idades mínima e máxima (“12 – 15”), apenas da mínima (“18-”) ou por descrição

textual (“pt”¹², “Apenas para Adultos”);

- Dificuldade (*Difficulty*) – determina o nível de dificuldade do material. Pode receber um dos seguintes valores: muito fácil, fácil, médio, difícil e muito difícil;
- Tempo Típico de Aprendizagem (*Typical Learning Time*) – define o tempo aproximado que um aluno pertencente à audiência típica do LO (Usuário Final Pretendido) deve levar para realizar todas as atividades;
- Descrição (*Description*) – apresenta comentários sobre como o LO deve ser utilizado;
- Linguagem (*Language*) – define o idioma utilizado para representar o conteúdo do LO.

4.6.1.6 Direitos (Rights)

Esta categoria especifica os metadados que definem as informações sobre os direitos autorais do LO. Basicamente, ele determina se a utilização do LO é cobrada ou não e se existe alguma restrição de uso para o mesmo.

4.6.1.7 Relações (Relation)

Define as relações que o LO descrito possui com outros LO. Esta propriedade é formada por uma lista que contém o recurso relacionado e o tipo de relacionamento (por exemplo: *é parte de*, *tem parte*, *é versão de*, *tem versão*, *referencia*, *é referenciado por*, *é baseado em*, *é base para*, *requer* e *é requerido por*).

4.6.1.8 Anotação

A cada LO é possível adicionar uma lista com até 30 anotações sobre o mesmo. Cada anotação é composta da entidade (pessoa ou organização) que a criou, data e descrição.

4.6.1.9 Classificação

Os elementos desta categoria definem onde o LO se encaixa dentro de um sistema particular de classificação. Para isto, deve ser especificado o propósito pelo qual o LO está sendo classificado, um caminho para a taxionomia onde está a classificação, seu nome,

¹² “pt” indica que a descrição é feita em português.

número de entrada e, por fim, uma descrição e as palavras-chave relacionando o LO com aquela taxionomia.

Além do padrão apresentado, outros têm sido desenvolvidos com o objetivo de definir metadados para LO. Dentre eles, vale ressaltar o IMS *Learning Resource Meta-data* (IMS, 2001b), cujas especificações são completamente baseadas no LOM e apresentam poucas alterações, como a inclusão de alguns elementos não existentes no padrão que lhe deu origem.

4.6.2 O Padrão IEEE 1484.2 Informações Públicas e Privadas de Alunos (PAPI)

As informações que representam as características dos alunos que participam de cursos através da *Web* também estão sendo padronizadas, assim como as informações dos LO acessados por eles. Com isto, pretende-se construir modelos padrões de aluno que possam ser trocados entre diferentes ambientes de EBW.

Dois padrões foram propostos com este objetivo, são eles: IEEE 1484.2, Padrão para Tecnologias de Aprendizagem – Informações Públicas e Privadas de Alunos (PAPI) (LTSC, 2000), e Pacote de Informações de Alunos (LIP) (IMS, 2001a). A seguir, são apresentadas, apenas de forma resumida, as categorias do padrão PAPI, pois não faz parte dos objetivos desta tese construir um modelo de aluno padrão.

O modelo PAPI prevê que as informações de um aluno sejam divididas em seis grandes categorias, como mostra a figura 22.

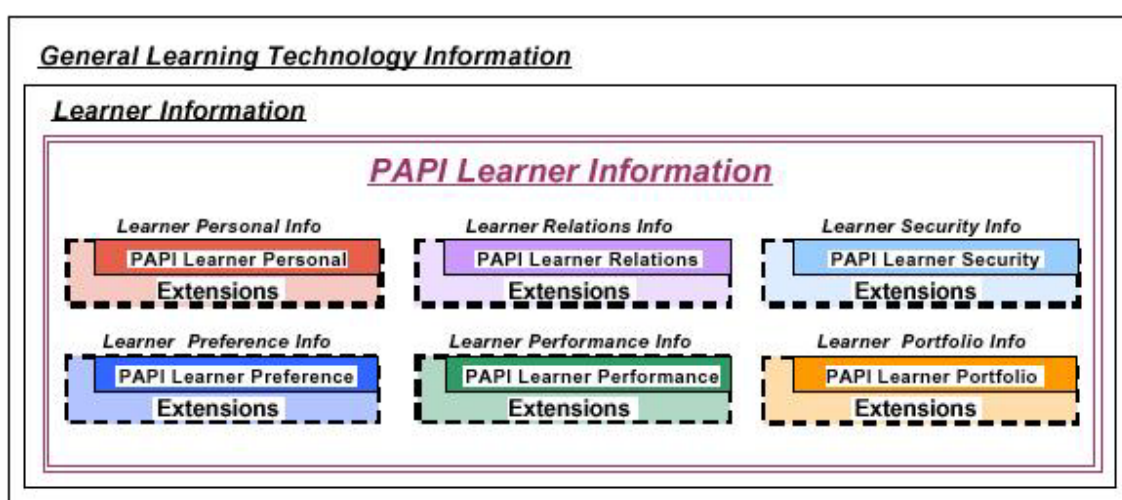


FIGURA 22 - Modelo conceitual do PAPI
Fonte: (LTSC, 2000)

A seguir é descrita cada uma das categorias, juntamente com alguns exemplos das

informações pertencentes a elas:

- Informações Pessoais do Aluno: Este grupo não está relacionado diretamente às medidas de desempenho do aluno. Ele contempla dados como nome, telefone, e-mail, endereço, entre outras informações que são utilizadas mais pelos módulos administrativos do sistema do que pelos módulos educacionais;
- Informações sobre as Relações do Aluno: Define uma lista com relações entre o aluno e outros usuários de ambientes de EBW. A lista é composta pelo nome da pessoa com quem o aluno possui uma relação e o tipo desta relação que pode ser: colega, é professor, professor de, é instrutor, instrutor de e pertence a (quando o aluno pertence a um grupo);
- Informações de Segurança do Aluno: Armazena as credenciais do aluno, com o seu contexto de utilização (por exemplo, ambiente de ensino X), o seu tipo (Senha) e o seu valor (ABC123);
- Informações de Preferências do Aluno: Descreve as preferências do aluno, com o objetivo de facilitar a sua interação com o sistema. Dentre outras informações, ele define o tipo de mídia preferida para as lições, juntamente com a prioridade desta preferência;
- Informações do Desempenho do Aluno: Relaciona a história do aluno, listando, por exemplo, seus certificados, seu trabalho atual e seus objetivos. Como as informações referentes ao trabalho atual e aos objetivos podem mudar com certa frequência, são associadas a elas datas de validade;
- Informações sobre o *Portfólio* do Aluno: Armazena um conjunto de referências para os trabalhos feitos pelo aluno, indicando o tipo de mídia em que foi desenvolvido e se o aluno recebeu algum certificado por ele.

4.6.3 O Padrão SCORM

Além dos padrões apresentados, vários outros vêm sendo desenvolvidos com o objetivo de definir metadados para componentes distintos envolvidos na EBW, que não os LO isoladamente e os alunos. Dentre eles estão o IMS *Content Packing* (IMS, 2004), o IMS *Simple Sequencing* (IMS, 2003a) e o IMS *Learning Design* (IMS, 2003b), que estabelecem, respectivamente, como os LO devem ser organizados para formar um curso baseado na *Web*, as regras a serem executadas para definir a ordem de apresentação do

conteúdo e o processo instrucional como um todo, referenciando as definições de vários outros padrões.

Com objetivos semelhantes, foi desenvolvido um padrão bastante conhecido: o SCORM (*Sharable Content Object Reference Model* – Modelo de Referência para Objetos de Conteúdo Compartilháveis) (ADL, 2004a), cuja força reside no fato de ser baseado nos padrões definidos pelo IEEE, como o LOM, e pelo IMS, como o IMS *Content Packing* e IMS *Simple Sequencing*. De acordo com ADL (2004a, p. 1-7), “o SCORM referencia especificações, padrões e guias desenvolvidos por outras organizações que são adaptados e integrados uns aos outros, para formar um modelo mais completo e fácil de implementar”.

As especificações do padrão SCORM estão divididas em três grandes grupos: Modelo de Agregação do Conteúdo (CAM – *Content Aggregation Model*) (ADL, 2004b); Ambiente em Tempo de Execução (RTE – *Run Time Enviroment*) (ADL, 2004c); e Sequenciamento e Navegação (SN – *Sequencing and Navigation*) (ADL, 2004d), conforme mostra a figura 23.



FIGURA 23 - Conjunto de especificações SCORM
(Fonte: ADL, 2004a)

O Modelo de Agregação do Conteúdo define:

- Os componentes utilizados durante o processo de aprendizagem (LO);
- Como os LO devem ser organizados (estruturados) para constituírem uma

atividade de aprendizagem, a qual pode representar uma lição, capítulo, curso completo ou qualquer denominação que se deseje atribuir àquele conjunto de LO;

- Como os componentes devem ser agrupados, formando os Pacotes de Conteúdos (*Content Packages*) para facilitar a troca de informações entre diferentes LMS;
- Como os componentes devem ser descritos, para facilitar a busca e a indexação dos mesmos;
- As regras para seqüenciamento do conteúdo.

Todas estas informações são armazenadas em um arquivo XML, denominado *imsmanifest*. O formato deste arquivo também é descrito e especificado pelo CAM, que tem suas especificações baseadas no padrão IMS *Content Packing*. Um curso completo é constituído pelos arquivos com seus recursos (LO), além do arquivo *imsmanifest.xml* reunidos em um arquivo do tipo *zip*. O quadro 22 apresenta os principais componentes de um arquivo *imsmanifest.xml*.

QUADRO 22 - Componentes principais de um arquivo *imsmanifest.xml*

```

1. <?xml version="1.0" encoding="UTF-8" ?>
2. <manifest xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"
3.   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4.   xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_v1p3" ....
5.   identifier="Curso2" version="1.3">
6.   <organizations default="Organizacao1">
7.     <organization identifier="Organizacao1">
8.       ....
9.       <imsss:sequencing>
10.        <imsss:controlMode flow="true" />
11.      </imsss:sequencing>
12.    </organization>
13.  </organizations>
14.  <resources> ....
15. </resources>
16. </manifest>

```

Nas linhas 1 a 5 estão as declarações iniciais, responsáveis por estabelecer que aquele arquivo é compatível com o padrão SCORM e por determinar os *namespaces* que

definem os elementos utilizados. A partir da linha 6, encontram-se as informações sobre curso propriamente ditas, que se dividem em duas partes principais: a definição de suas estruturas, realizadas através dos elementos `<organizations>` e `<organization>`; e a relação dos LO utilizados, através do elemento `<resources>`.

A tag `<organizations>` engloba as diferentes estruturas para organização e apresentação do conteúdo que o curso pode conter. Ela é formada por um ou mais elementos do tipo `<organization>`, cuja função é definir uma estrutura de apresentação e navegação do conteúdo. Dentro deste elemento, são dispostas hierarquicamente as atividades de aprendizagem, como é visto no quadro 23, e as regras de apresentação do conteúdo.

O código apresentado no quadro 23 refere-se às atividades de aprendizagem que compõem apenas uma das estruturas de um curso. Estas atividades são representadas no arquivo `imsmanifest.xml` por elementos do tipo `<item>`, que são alocados dentro do elemento `<organization>`. Cada atividade de aprendizagem pode conter subatividades de aprendizagem ou uma referência para um recurso, que indica os LO que devem ser apresentados. Devido a uma restrição do CAM, uma atividade não pode conter subatividades e estar associada a um recurso ao mesmo tempo.

QUADRO 23 - Estrutura das atividades de aprendizagem e regras de sequenciamento

```

1. <organization identifier="Organizacaol">
2.   <title>Pedagogico: Iniciante - Tecnico: Domestico</title>
3.   <item identifier="Atv1" identifierref="RECAtv1">
4.     <title>Conceitos Iniciais de Estatistica</title>
5.   </item>
6.   <item identifier="Atv2">
7.     <title>Tipos de Medidas Numericas</title>
8.     <item identifier="Atv3" identifierref="RECAtv3">
9.       <title>Principais Medidas de Posicao</title>
10.    </item>
11.    <item identifier="Atv4" identifierref="RECAtv4">
12.      <title>Principais Medidas de Dispersao</title>
13.    </item>
14.  </item>
15.  <imsss:sequencing>
16.    <imsss:controlMode flow="true"/>
17.  </imsss:sequencing>
18. </item>
19. </organization>

```

A linha 2 (quadro 23) indica o título atribuído àquela estrutura específica do curso. Como já citado, um mesmo curso pode conter várias estruturas diferentes, as quais também recebem títulos diferentes. A linha 3 indica a criação da primeira atividade, com o título *Conceitos Iniciais de Estatística* (linha 4), que está associada a um recurso composto por dois LO, como mostra o quadro 24, linha 2.

Na linha 6 está definida a segunda atividade, intitulada *Tipos de Medidas Numéricas* (linha 7). Esta atividade possui duas subatividades, que são definidas nas linhas 8 a 10 e 11 a 13. Por este motivo, esta atividade não está associada a nenhum recurso.

Entre as linhas 15 a 17, encontram-se as *tags* que definem a forma de sequenciamento e navegação do conteúdo. Neste exemplo, foi utilizado apenas um elemento, que permite que o conteúdo seja apresentado em seqüência, através dos botões anterior e próximo. Mecanismos mais elaborados para o sequenciamento e a navegação do conteúdo podem ser definidos utilizando as variáveis e as regras definidas no SN. As informações deste padrão são traduzidas em elementos definidos no CAM e incorporadas ao arquivo *imsmanifest* do curso, para que possam ser interpretadas pelo LMS.

QUADRO 24 - Recursos utilizados no curso

```

1.<resources>
2.<resource identifier="RECAtv1" adlcp:scormType="sco"
3.           type="webcontent" href="index.html">
4.   <file href="index.html"/>
5.   <file href="introducao.html"/>
6. </resource>
7. <resource identifier="RECAtv3" adlcp:scormType="sco"
8.           type="webcontent" href="med_posicao.html">
9.   <file href="med_posicao.html"/>
10.  <file href="med_posicao.html"/>
11.  <file href="med_posicao.html"/>
12. </resource>
13. ....
14.</resources>

```

O quadro 24 mostra como os recursos (LO) estão dispostos no arquivo *imsmanifest.xml*. Na linha 1, está a *tag* que determina o início da declaração dos recursos. Na linha 2, encontra-se a identificação do primeiro recurso, que é utilizada para associá-lo às atividades de aprendizagem, como pode ser visto no quadro 23, linha 3, por meio da propriedade *identifierref*. A propriedade *href* (linha 3) identifica o LO principal, ou seja, o

primeiro a ser apresentado dentre os listados nas linhas 4 e 5. Na linha 6, o elemento *resource* é fechado e na seqüência são definidos os demais recursos, de maneira semelhante.

Algumas informações do CAM são utilizadas pelo Ambiente de Tempo de Execução (RTE), para definir como o conteúdo dos LO deve ser processado pelos LMS. As especificações do RTE definem os requisitos que ambientes de EBW ou LMS devem possuir, para utilizarem a API SCORM e manipular cursos construídos de acordo com este padrão. O RTE também apresenta métodos para conduzir e intermediar as comunicações entre o aluno e o LMS, incluindo dados sobre a situação do curso, como os materiais que estão sendo apresentados para o estudante, e informações sobre o seu progresso durante o curso, sua pontuação em atividades, limites de tempo para execução de tarefas e outras informações que possam vir a ser de interesse dos instrutores responsáveis.

O estudo e a descrição dos elementos do SCORM têm como objetivo o entendimento de um padrão que já está sendo adotado por muitas pessoas e empresas ligadas ao desenvolvimento de sistemas educacionais baseados na *Web*. Estas informações foram úteis para atingir o objetivo deste trabalho, uma vez que elas são utilizadas na construção dos cursos gerados pela ferramenta desenvolvida.

4.7 Propostas de Ambientes Educacionais Baseados na *Web* Semântica

O objetivo desta seção é apresentar algumas das propostas de sistemas educacionais que utilizam recursos da *Web* Semântica encontradas durante a revisão bibliográfica, para que se possa destacar seus pontos fracos e fortes, visando subsidiar a concepção da arquitetura desenvolvida neste trabalho.

4.7.1 Multitutor

A arquitetura de um ILMS denominado Multitutor é apresentada na figura 24 (SIMIC, GASEVIC e DEVEDZIC, 2004).

Este sistema prevê dois usuários principais, o professor e o aluno, que acessam o conteúdo armazenado no servidor *Web* através de um navegador comum. O Multitutor possui três módulos fundamentais:

- Especialista: através dele o professor cria os conteúdos do curso;
- Modelo do aluno: armazena os dados pessoais do aluno, seus resultados e histórico de utilização do sistema, para que o módulo tutor possa definir a

melhor estratégia de ensino para cada um deles;

- Tutor: armazena as estratégias pedagógicas. É ele que analisa os dados vindos do modelo do aluno e decide qual conteúdo deve ser apresentado e de que maneira.

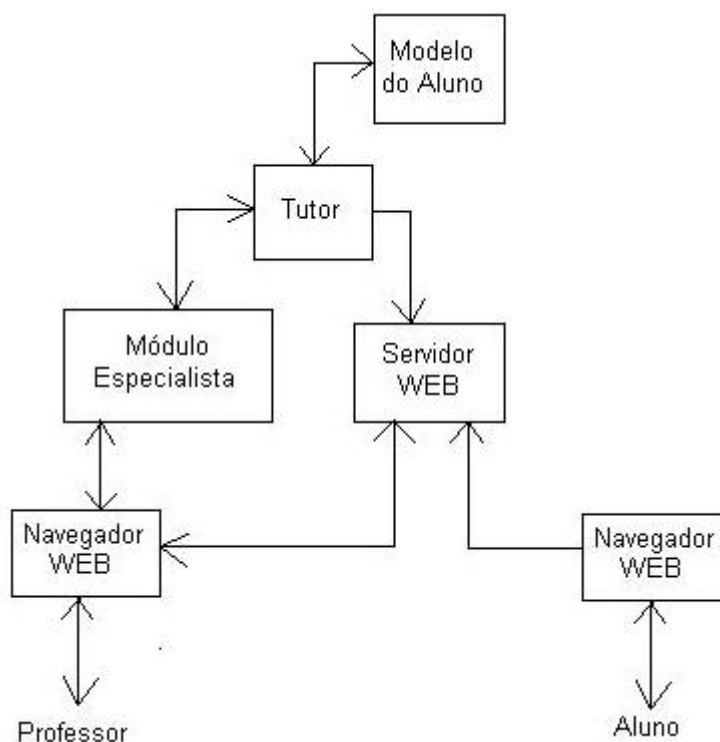


FIGURA 24 - Arquitetura do Multitutor
 Fonte: Adaptado de SIMIC, GASEVIC e DEVEDZIC (2004)

A arquitetura apresentada na figura 24, a priori, não apresenta grandes novidades em relação ao que já é conhecido sobre STI. Seu diferencial está na forma como cada um dos módulos é implementado, com base em metadados e ontologias. O sistema prevê uma ontologia para o curso e outra para o modelo do aluno. Um curso é dividido em capítulos, que por sua vez é dividido em várias lições. A cada lição é associado um LO, que possui um conceito, uma explicação e vários testes e conteúdos de aprendizagem. Esta ontologia é implementada através de um esquema XML, com os metadados necessários para possibilitar a reutilização dos cursos, capítulos e lições. Esta arquitetura configura-se como um exemplar dos STI baseados na pedagogia tradicional, com uma rígida estrutura para organização e apresentação do conteúdo, onde a passagem para uma nova lição está condicionada ao sucesso obtido na avaliação da lição anterior.

A ontologia do aluno, também implementada através de um esquema XML, inclui seus dados básicos, estereótipo, habilidades reais e habilidades estimadas pelo sistema.

Através destas informações, o módulo tutor determina os próximos conteúdos a serem apresentados ao aluno e a forma pela qual isto será feito.

A possibilidade de definir o modelo do aluno e o conteúdo dos cursos através de ontologias está entre as principais virtudes do Multitutor, pois isto permite a associação de metadados com o intuito de facilitar a reutilização de suas informações.

Contudo, algumas características importantes não estão presentes neste sistema. Dentre elas, destaca-se a falta de interoperabilidade com outros sistemas, pois apenas através dele é que os professores podem criar diferentes cursos, reutilizando os LO. Outro aspecto que pode ser melhorado é a linguagem utilizada para representar as ontologias, pois como visto no capítulo 3, os esquemas XML possuem baixa representatividade semântica, ficando muito aquém de linguagens como DAML+OIL e OWL. Por fim, destaca-se a rigidez na estrutura do curso criado, organizada em capítulos e lições, que força o usuário a adaptar os conteúdos para se enquadrarem na estrutura definida na ontologia.

4.7.2 INES – Servidores Educacionais Inteligentes

Uma arquitetura mais completa, porém ainda em fase inicial de desenvolvimento, é apresentada por DEVEDZIC (2004). Segundo este autor, a EBW deve ser completamente distribuída, como mostra a figura 25.

Esta figura mostra que a intenção do autor é oferecer uma estrutura distribuída, onde os LO estão armazenados em diferentes servidores e são acessados, de forma transparente, tanto pelos professores que os utilizam na construção de um curso quanto pelos alunos que os acessam para aprender determinado conteúdo. Esta transparência deve ser garantida pelos agentes pedagógicos, que fazem a localização do servidor que possui o LO desejado, e pelos próprios servidores, que devem ser capazes de procurar em outros servidores materiais que complementem as informações a serem apresentadas.

Outra característica existente é que os servidores educacionais, além de servirem como repositórios de LO, devem atuar como STI, possuindo juntamente com o conhecimento do domínio, o conhecimento pedagógico, que aliado ao modelo do aluno definido em seu agente pedagógico, serve para determinar a melhor ordem e forma de apresentação do conteúdo.

Cada servidor educacional deve possuir uma arquitetura semelhante à apresentada na figura 26, a qual mostra que o servidor proposto oferece serviços de apresentação do

conteúdo (aprendizagem), realização de exercícios e testes (avaliação), suporte à consulta de materiais complementares (referências) e organização de turmas e comunicação entre seus participantes (colaboração). Todos estes serviços estão relacionados aos conteúdos educacionais, que envolvem as informações sobre o domínio que está sendo estudado e as questões pedagógicas sobre as melhores formas para o seu seqüenciamento e apresentação.

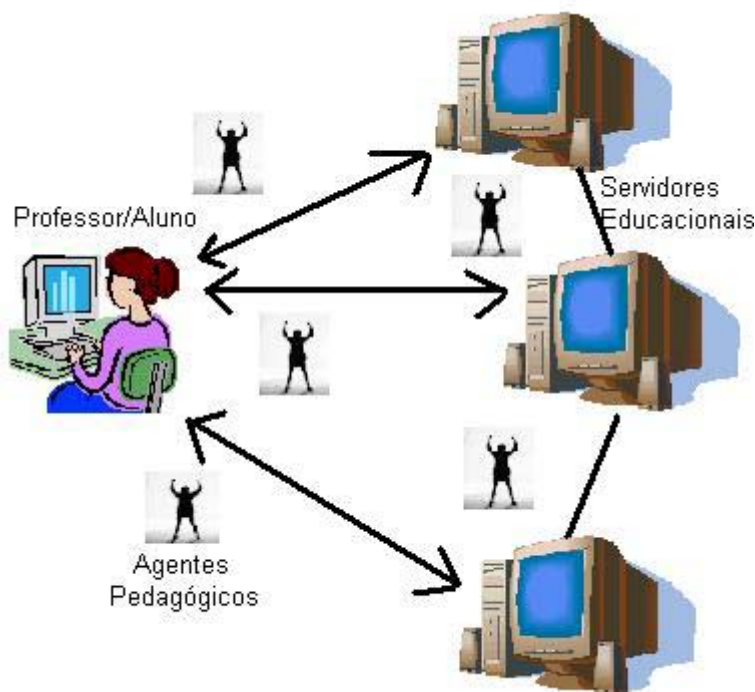


FIGURA 25 - Arquitetura desejada dos sistemas educacionais baseados na *Web*
Fonte: (DEVEDZIC, 2004)

A diferença entre a proposta de DEVEDZIC (2004) e um STI convencional reside nas ligações existentes entre o servidor e um conjunto de ontologias (O-1, O-2, O-n), além do uso dos agentes pedagógicos.

Através das ontologias, o servidor educacional pode delegar tarefas a outros servidores, bem como buscar informações a fim de executar suas próprias tarefas. Desta forma, vários servidores, devidamente descritos e interligados através de ontologias, poderão interagir em prol de um objetivo comum, que é o de maximizar a aprendizagem dos alunos.

Segundo o autor desta proposta, são muitos os benefícios que podem ser alcançados com a sua implementação, mas também fica claro que muito ainda precisa ser pesquisado em áreas como *web services* inteligentes, redes sociais, ontologias e mineração na *Web* até que ela possa ser completamente desenvolvida.

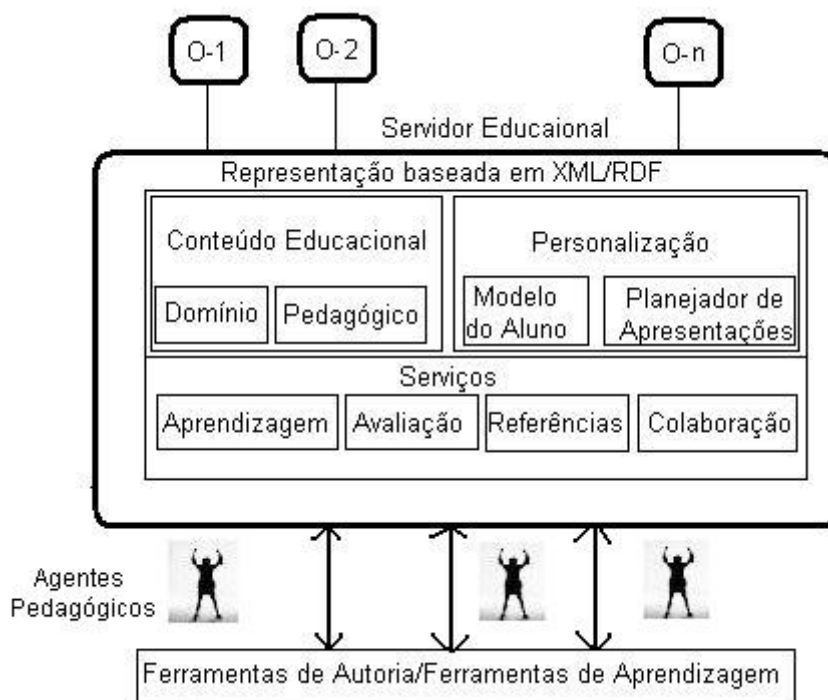


FIGURA 26 - Arquitetura interna de um servidor educacional

4.7.3 OntAWare

O sistema proposto por HOLOHAN *et. al.* (2005), denominado OntAWare, fornece um conjunto de ferramentas que suportam a autoria, a gerência e a disponibilização de conteúdos instrucionais. Ele possui um módulo que recebe como entrada uma ontologia OWL, selecionada a partir de uma lista pré-existente de ontologias, e gera como saída vários cursos estáticos.

Após selecionar a ontologia, o usuário deve escolher qual conceito (representado por uma classe da ontologia escolhida) é o objetivo de aprendizagem principal, para que o sistema possa gerar os LO com base no conceito selecionado e em seus subconceitos. O curso produzido inclui uma lição construída automaticamente a partir das informações da ontologia (arquivo HTML com a seqüência de *links* que apontam para os LO) e pelos próprios LO, que podem ser arquivos HTML, PPT, PDF etc. O sistema também gera automaticamente LO com exercícios do tipo “Qual dos seguintes (não) é um exemplo do conceito X?”.

Além desta funcionalidade, o OntAWare gera cursos que podem ser adaptados dinamicamente a partir das necessidades do usuário. Mas diferentemente do curso estático, que pode ser executado em qualquer navegador *Web*, o curso dinâmico requer um sistema

proprietário. Por isto, foi construído um ambiente para apresentação do conteúdo, baseado na ontologia, o qual monitora os passos do estudante para guiá-lo em sua navegação. O mecanismo utilizado é bastante simples, pois se baseia apenas no fato do aluno ter acessado um LO ou não, para indicar se ele ainda precisa passar por ele. O curso gerado pode ser exportado em um arquivo compatível com o CAM do padrão SCORM para facilitar a interoperabilidade, o que não elimina a necessidade de utilização do sistema proprietário para disponibilizar o curso.

Como trabalhos futuros, HOLOHAN *et. al.* (2005) propõem a melhoria na adaptação do conteúdo, incluindo outras variáveis em seu processo, e a vinculação do sistema proprietário para disponibilizar o curso ao Ambiente de Tempo de Execução (RTE) do padrão SCORM, visando aumentar a interoperabilidade ao tornar os cursos gerados independentes do sistema proprietário de disponibilização.

Outra questão não contemplada neste trabalho é um suporte pedagógico na elaboração do curso. Percebe-se que termos como conceito e objetivos são utilizados para referenciar a mesma entidade, o que prejudica a qualidade do curso resultante. Além disto, existe uma simplificação do processo avaliativo que pode comprometer a aprendizagem, devido à forma pela qual o sistema gera LO com questões baseadas apenas no conteúdo, sem que objetivos de aprendizagem tenham sido definidos.

4.7.4 *OntoAIMS*

O *OntoAIMS* (DENAUX, DIMITROVA e AROYO, 2005) é um sistema desenvolvido com o objetivo de integrar, por meio de ontologias, a modelagem de usuário interativa com a gerência de conteúdos instrucionais para resolver problemas relativos à adaptação de conteúdo.

O sistema possui uma ontologia de domínio que define os conceitos e suas relações, um modelo de recursos que descreve os documentos no repositório, os quais estão associados aos conceitos da ontologia de domínio, e um modelo de tarefas do curso que representa a sua estrutura.

A adaptação do conteúdo é alcançada graças ao modelo do usuário que cobre as características e preferências do aluno. Este sistema possui mecanismos bem mais elaborados para representar as informações do usuário e para possibilitar a adaptação do conteúdo que o *OntAWARE*, pois ele volta-se mais para a construção e utilização do modelo do aluno do que para questões pedagógicas e por isto não contempla muitos aspectos desta

área e não possui recursos para auxiliar na construção da estrutura do curso e na definição dos conceitos abordados.

4.7.5 MOT

Outra ferramenta que visa facilitar a construção de materiais adaptativos para a EBW é apresentada por POWER *et. al.* (2005). O sistema denominado MOT permite que usuários criem mapas conceituais sobre diferentes domínios e lições que apresentem os assuntos definidos nestes mapas. Ao gerar um curso, o MOT possibilita que o usuário selecione os conceitos e as lições desejadas, bem como defina as estratégias de adaptação utilizadas.

Inicialmente, o MOT foi projetado para trabalhar isoladamente, portanto os cursos gerados só podiam ser acessados através do próprio programa. Atualmente, está sendo implementado um módulo capaz de converter as informações do formato proprietário para o padrão SCORM, visando facilitar a interoperabilidade e possibilitar que os cursos gerados sejam executados em qualquer LMS compatível com este padrão.

Entretanto, o trabalho está em fase inicial e muitas características adaptativas presentes no projeto original foram suprimidas para facilitar a sua implementação, devido às restrições do padrão SCORM. Por fim, vale destacar que nenhuma referência às ciências da aprendizagem é feita neste trabalho, o que torna o trabalho de construção de um curso uma tarefa puramente computacional, sem nenhuma preocupação com os resultados que o mesmo produzirá ao ser acessado pelos alunos.

4.8 Teorias Educacionais

Analisando as arquiteturas e ferramentas propostas para integrar os recursos da *Web Semântica* à EBW, pode-se perceber que, em sua grande maioria, elas não contemplam as teorias das ciências da aprendizagem. Mesmo quando o fazem, é de forma superficial, sendo os aspectos pedagógicos preteridos em relação aos computacionais. Por isto, optou-se por realizar um estudo mais detalhado da área para subsidiar a construção das ontologias e do sistema desenvolvido, que são resumidamente apresentados nesta seção.

Considerando que o propósito fundamental de qualquer programa educacional, inclusive os de EBW, é promover a aprendizagem, e que esta é alcançada através da instrução, é fundamental que este termo seja claramente definido. Segundo GAGNÉ (1980, p.2), “A palavra instrução pode ser definida como o conjunto de eventos planejados para

iniciar, ativar e manter a aprendizagem no aluno”. Estes eventos devem ser, em primeiro lugar, planejados e, posteriormente, entregues ao aluno, para que produzam os efeitos necessários.

Para que os eventos instrucionais possam ser adequadamente planejados para um aluno ou para um conjunto de alunos, é preciso ter em mente que objetivos pretende-se que eles atinjam após a aprendizagem. Por isto, é importante entender as diferentes classes de objetivos e as operações mentais requeridas por cada uma delas antes de planejar os eventos instrucionais.

Um trabalho relevante relacionado à classificação de objetivos de aprendizagem que foi utilizado para subsidiar o desenvolvimento desta tese é o de BLOOM *et. al.* (1976). Sua principal contribuição reside em uma taxionomia que classifica os objetivos de aprendizagem em ordem crescente de complexidade, com base nas operações mentais que eles requerem, independente do domínio do conhecimento. Esta taxionomia está dividida em seis categorias principais, a saber:

- Conhecimento - é a categoria que menos exige do aluno, pois basta que ele memorize algumas informações, mesmo que não consiga compreender completamente o seu significado ou aplicá-las para resolver um problema. Objetivos de aprendizagem pertencentes a esta categoria são úteis nos momentos iniciais da aprendizagem, quando a pessoa deve, primeiramente, armazenar informações que serão recuperadas posteriormente para desenvolver processos mais complexos. Alguns verbos que refletem este nível de conhecimento são: citar, nomear, listar, reproduzir, repetir e apresentar;
- Compreensão - nos objetivos desta categoria não basta que o aluno memorize informações, é necessário, também, que ele compreenda o significado e a importância das mesmas. Alguns verbos que refletem este nível de conhecimento são: explicar, relacionar, traduzir, transformar, descrever e associar;
- Aplicação – nesta categoria é preciso, além de compreender determinados conceitos, generalizá-los a fim de aplicá-los a novas situações. Alguns verbos que refletem este nível de conhecimento são: aplicar, prever, demonstrar, preparar, resolver e modelar;
- Análise – esta categoria enfoca o desdobramento de um conceito em suas

partes constitutivas e a percepção de suas inter-relações e de seus modos de organização. Por esta razão, para analisar determinado conceito é preciso previamente compreendê-lo como um todo e ser capaz de generalizar a sua aplicação. Esta etapa é fundamental para que se chegue a última categoria da taxionomia, que é a capacidade de avaliação ou julgamento. Alguns verbos que refletem este nível de conhecimento são: examinar, determinar, categorizar, diagnosticar, dividir, classificar e organizar;

- Síntese – esta categoria possibilita o conhecimento sobre como unir partes já conhecidas e estudadas em um todo com características não percebidas anteriormente. Esta categoria é responsável por dotar as pessoas com um comportamento criador. Alguns verbos que refletem este nível de conhecimento são: combinar, conceber, integrar, produzir, inventar, projetar, reorganizar, formular etc.
- Avaliação – é a capacidade de julgamento sobre valores, idéias, materiais, métodos, enfim, tudo o que pode ser apresentado a uma pessoa com um determinado propósito. Foi disposta no último estágio da taxionomia, porque é preciso *conhecer, compreender, aplicar, analisar e sintetizar* para ter a capacidade de *avaliar*. Alguns verbos que refletem este nível de conhecimento são: concluir, avaliar, criticar, justificar, deduzir, recomendar e comparar.

Ao criar os objetivos para o seu curso, o professor deve classificá-los em uma destas categorias antes de planejar os eventos instrucionais, pois isto lhe servirá como um guia, já que objetivos de uma mesma categoria podem ser alcançados através de atividades semelhantes, independente do assunto em questão.

Diferentes maneiras de organizar a instrução (planejar os eventos) têm sido propostas por vários autores, mesmo para objetivos da mesma categoria, sendo cada uma delas baseada em uma teoria ou modelo de aprendizagem. Esta diversidade de modelos ocorre porque ainda são muitas as dúvidas existentes sobre como a informação é processada no cérebro humano e, conseqüentemente, sobre como o ser humano aprende.

Segundo SASKATCHEWAN EDUCATION (1991), “as estratégias instrucionais determinam a abordagem que um professor pode utilizar para atingir os objetivos de aprendizagem”. Segundo o autor, cinco estratégias diferentes podem ser utilizadas, são elas:

- Instrução direta – é a estratégia mais centrada no professor. Ela está baseada no método tradicional de apresentação do conteúdo, exercícios, resolução de dúvidas e avaliação;
- Instrução indireta – é uma estratégia mais focada no aluno, onde termos como observação, investigação e resolução de problemas, denotam sua característica de liberdade para que o aluno possa construir seu próprio conhecimento;
- Instrução interativa – é uma evolução da estratégia anterior, em que a característica de liberdade permanece, mas o conhecimento é construído coletivamente a partir de discussões entre os grupos de alunos;
- Aprendizagem Experimental – também é uma estratégia centrada no aluno, mas orientada a atividades (experimentos). A aprendizagem ocorre em um ciclo de cinco etapas que envolvem *Experimentar, Compartilhar, Analisar, Inferir e Aplicar*;
- Estudo independente – esta abordagem propicia que o aluno seja responsável pelo planejamento e pela execução das atividades que o levarão a atingir seus objetivos;

Vários autores desenvolveram teorias sobre como organizar os eventos e as atividades instrucionais, dependendo da estratégia instrucional utilizada. A seguir são apresentadas as fases que a instrução deve possuir para levar a uma efetiva aprendizagem, de acordo com a proposta de Gagné (GAGNÉ, 1985), que tem por base a teoria de aprendizagem denominada “Teoria de Processamento de Informação”. As idéias de Gagné são mais adequadas para a instrução direta, mas podem ser aplicadas também a outras estratégias de ensino, com alguns ajustes nas atividades oferecidas aos alunos em cada uma das fases ou eventos instrucionais.

Segundo Gagné, o processo instrucional pode ser dividido em nove fases, que após sua realização, levam o aluno a aprender efetivamente um conteúdo. A seguir, faz-se um breve relato de cada uma delas:

- Obtendo a atenção - É fundamental que o aluno esteja motivado para aprender o conteúdo, por isto é preciso gerar expectativas sobre a *recompensa* que ele terá após atingir uma meta. É importante frisar que a *recompensa* não precisa, ou não deve, focar apenas os benefícios diretos ao

aluno, como a atribuição de notas. Ela deve ser mais abrangente, como, por exemplo, mostrar ao aluno o que ele será capaz de fazer após aprender aquele conteúdo;

- Apresentação dos objetivos - Nesta fase o aluno deve ser estimulado a prestar atenção naquilo que é relevante, devem ficar bem claros os objetivos de aprendizagem que se espera que o aluno atinja após concluir o processo instrucional;
- Recuperação de Pré-requisitos – Estando informado dos objetivos de aprendizagem, o aluno deve receber estímulos que o façam lembrar de assuntos relacionados já vistos e conhecimentos que ele já possui que possam ser associados aos novos conhecimentos, com o objetivo de facilitar a sua assimilação;
- Apresentação do material – Nesta fase, ocorre o momento essencial da aprendizagem, quando a informação entra na memória do aluno, após a apresentação do conteúdo a ser aprendido;
- Guiar a aprendizagem - nesta fase deve-se ajudar o aluno a adquirir capacidades particulares, especificadas nos objetivos;
- Observar a execução – o aluno deve ser estimulado a desenvolver alguma ação que o faça mostrar se adquiriu a capacidade desejada. É importante que esta fase esteja relacionada com os objetivos de aprendizagem;
- Fornecer um retorno sobre a execução – nesta fase, que é fundamental para a aprendizagem, o aluno deve ser informado se a atividade desenvolvida por ele atingiu os objetivos esperados ou não. Quanto mais detalhada e informativa for esta descrição, mais fácil é para o aluno identificar os pontos onde precisa melhorar;
- Avaliando o desempenho – o objetivo desta fase é verificar se o aluno atingiu os objetivos desejados. Mais uma vez é importante que esta avaliação esteja em consonância com os objetivos propostos;
- Melhorando a retenção e a transferência – instrutores não podem esperar que os alunos sejam capazes de generalizar as informações aprendidas por si só, é importante que o processo instrucional preveja atividades que auxiliem nesta tarefa.

As informações constantes nos trabalhos apresentados embasaram a elaboração dos mecanismos capazes de auxiliar os usuários durante a construção de cursos ou atividades, fornecendo um suporte pedagógico a eles.

4.9 Considerações

Através do estudo realizado para o desenvolvimento deste capítulo, pôde-se ter uma idéia da importância da EAD e da EBW para a sociedade moderna. Na seqüência, foi possível identificar as principais linhas de pesquisa que se desenvolveram a sua volta, como STI, HA, LMS e LO, suas limitações e como elas estão convergindo para um ponto comum, visando melhorar o processo de desenvolvimento de materiais e a sua utilização pelos alunos.

Com o estudo de trabalhos correlatos foi possível conhecer o estado da arte desta linha de pesquisa, validar o objetivo desta tese e definir de forma mais clara os requisitos da arquitetura e da ferramenta desenvolvida. O conhecimento obtido através dos estudos sobre teorias de aprendizagem foi fundamental para estruturar as informações que fornecem um auxílio pedagógico durante a construção de cursos para a EBW, que é uma das principais limitações encontradas nas ferramentas estudadas.

5. DESCRIÇÃO DA ARQUITETURA PROPOSTA

Neste capítulo apresenta-se a modelagem das ontologias e do sistema desenvolvido, detalhando-se seus requisitos e arquitetura.

5.1 Definição dos Requisitos

Os estudos realizados sobre a EBW possibilitaram a identificação de uma relação de problemas que ainda desafiam os pesquisadores e merecem atenção para que melhores resultados possam ser alcançados. Estes problemas dizem respeito a dois aspectos distintos, o computacional e o educacional.

Os principais problemas enfrentados na EBW sob a ótica computacional são:

1. Pouco material é reutilizado durante a construção de novas atividades instrucionais, tornando esta tarefa cara e onerosa;
2. Faltam ontologias para descrever formalmente domínios do conhecimento, com o objetivo de facilitar a reutilização de conteúdos e atividades instrucionais, bem como a interoperabilidade entre sistemas;
3. Para que algum material possa ser reutilizado é preciso saber onde ele se encontra, pois a *Web* atual não permite realizar buscas precisas;
4. Mesmo com a criação de padrões como o LOM, ainda não foram resolvidos todos os problemas de busca de LO, pois muitos de seus atributos podem ser associados a um texto qualquer;
5. Embora já existam alguns padrões que possibilitam descrever qualquer componente do processo instrucional, eles ainda não foram tão aceitos quanto os padrões que descrevem LO. Por isto, muitos sistemas utilizam padrões próprios para representar tais componentes, dificultando ainda mais a reutilização e a interoperabilidade entre os sistemas.

Do ponto de vista educacional, os problemas que se destacam são:

6. A maioria das ferramentas de autoria não é fundamentada nas ciências instrucionais e da aprendizagem;
7. As técnicas adaptativas e pedagógicas são vinculadas ao domínio do conhecimento, o que dificulta a sua reutilização;
8. Devido ao problema listado no item anterior, a maioria dos sistemas de EBW não é capaz de adaptar os conteúdos às necessidades e características de cada usuário;
9. Devido à falta de interoperabilidade entre os sistemas, o conhecimento que eles possuem é restrito a uma base fechada e proprietária.

Soluções para todos estes problemas estão sendo pesquisadas, envolvendo estudos nas áreas de Engenharia *Web*, *Web Semântica*, Inteligência Artificial, Sistemas Tutores Inteligentes, Hipermídia Adaptativa, Ciências da Aprendizagem entre outras. Esta tese apresenta uma arquitetura criada com o objetivo de minimizá-los e envolve conceitos de *Web Semântica*, Hipermídia Adaptativa e Ciências da Aprendizagem. O quadro 25 descreve os requisitos que devem ser atendidos por esta arquitetura, para atingir a solução parcial ou, pelo menos, a minimização dos problemas previamente descritos.

QUADRO 25 - Requisitos da arquitetura desenvolvida

	Descrição do Requisito	Problema
1	Deve possibilitar a utilização de recursos da <i>Web Semântica</i> , como ontologias e conteúdos educacionais descritos por metadados, para aumentar a capacidade de reutilização dos componentes usados na construção de cursos para a EBW.	1, 2, 3 e 4.
2	Deve fornecer terminologias padrões para descrever o domínio do conhecimento (ontologia de domínio), para que pessoas compartilhem e reutilizem informações durante a construção dos cursos.	2
3	A ontologia de domínio deve conter uma estrutura que possa ser reutilizada em outros domínios, sem afetar os demais módulos da arquitetura.	7
4	Deve auxiliar a estruturar as atividades instrucionais, com base no conhecimento existente nas ciências instrucionais e da aprendizagem.	6
5	As informações oriundas das ciências instrucionais e da aprendizagem devem estar formalmente definidas através de ontologias (ontologia pedagógica).	6 e 7
6	Devem ser definidas informações que descrevam as características dos alunos para que se possa auxiliar na adaptação dos conteúdos.	8
7	As informações sobre as características dos estudantes também devem estar definidas em uma ontologia (ontologia do estudante).	7
8	As ontologias de domínio, pedagógica e do estudante devem ser expressas em OWL, para facilitar a interoperabilidade.	1
9	Além de reutilizar qualquer componente do processo instrucional (LO isolados, atividades de aprendizagem completas ou objetivos de aprendizagem), deve-se tornar possível a construção ou descrição de tais componentes.	4 e 9
10	A atividade instrucional gerada (curso) deve ser representada utilizando um padrão de metadados para componentes educacionais, para que ela possa ser executada em qualquer LMS que suporte o padrão escolhido.	5

O problema 8, adaptação do conteúdo às características do usuário, é abordado apenas em partes pois não foram elaborados os mecanismos para coleta das informações do aluno e adaptação durante a sua interação com o sistema. Como o foco do trabalho está relacionado à construção do curso, ele é gerado com diferentes estruturas que podem ser aplicadas a diferentes tipos de alunos, porém, a verificação de qual é a estrutura mais adequada para cada aluno fica a cargo do LMS utilizado para disponibilizar o curso.

Os requisitos listados no quadro 25 foram utilizados também para a validação da arquitetura elaborada. Como eles representam soluções para os problemas da EBW, identificados no levantamento bibliográfico, satisfazê-los significa atingir o objetivo proposto, que é modelar uma arquitetura para ambientes de aprendizagem adaptativos fundamentados em teorias das ciências instrucionais, que permitam criar materiais didáticos utilizando descrições semânticas.

5.2 Arquitetura do Sistema

A figura 27 apresenta uma visão geral da arquitetura proposta para atender aos requisitos listados no quadro 25. Nessa figura, percebe-se a presença de quatro diferentes tipos de usuários: o administrador, o criador do curso, o tutor e o aluno. O termo criador do curso é utilizado para referenciar professores, projetistas instrucionais ou qualquer outra pessoa que esteja criando um curso para uso próprio ou para disponibilizar a outras pessoas, incluindo os próprios alunos e administradores do sistema.

O administrador indicado na figura, que é responsável pela construção e atualização das ontologias de domínio, pedagógica e do estudante, também representa mais de uma pessoa, pois é inviável que um único indivíduo seja capaz de realizar todas estas tarefas.

O ideal é que um grupo de pessoas, cada uma em sua especialidade, atue como administrador. Especialistas em diferentes assuntos devem construir e atualizar as ontologias de domínio da sua área de atuação; profissionais da educação devem manter a ontologia pedagógica atualizada com as novas descobertas sobre o processo de ensino-aprendizagem; por fim, pesquisadores sobre sistemas adaptativos e modelagem do usuário devem ficar responsáveis pela definição das características da ontologia do estudante.

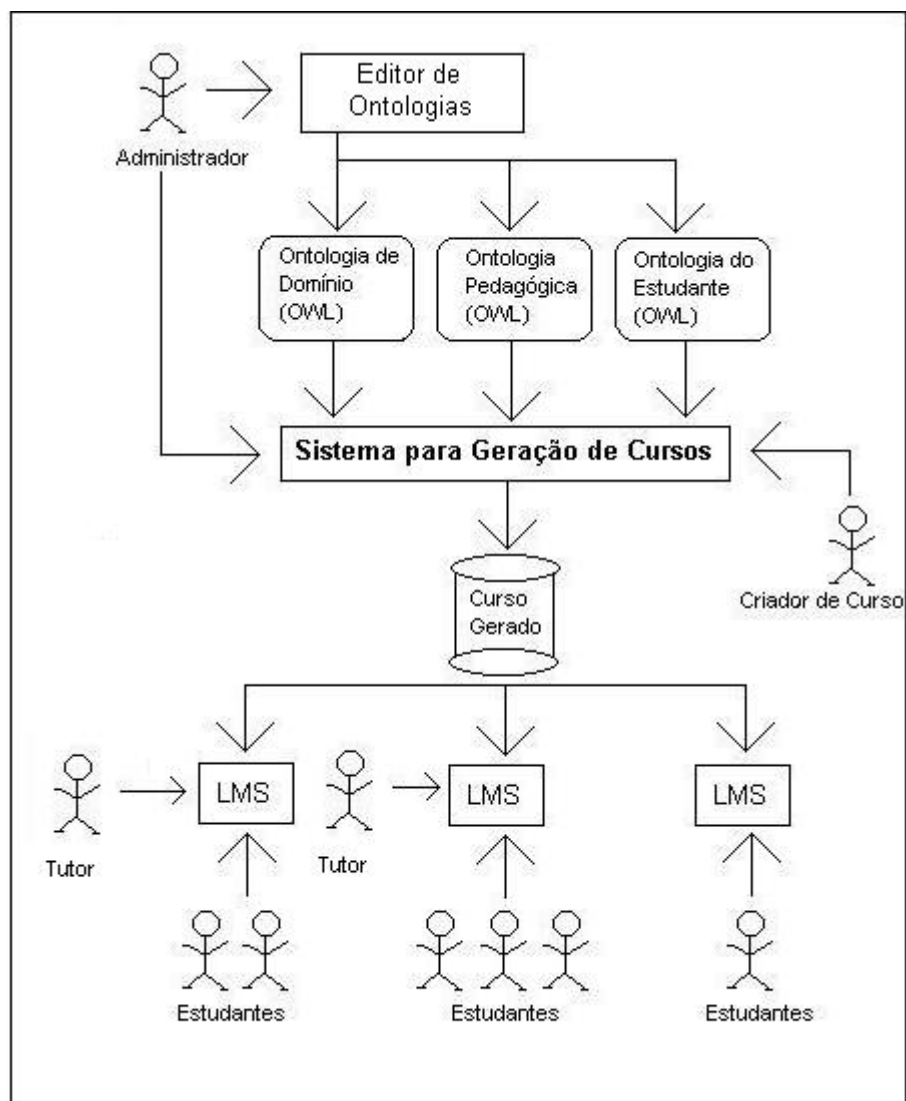


FIGURA 27 - Visão geral da arquitetura proposta

O tutor representa as pessoas responsáveis por acompanhar as atividades desenvolvidas pelos estudantes e auxiliá-los sempre que necessário. Em alguns casos, é possível que o criador do curso e o tutor sejam a mesma pessoa. Entretanto, a presença desta figura não é obrigatória, como pode ser visto na figura 27, onde estudantes acessam cursos disponíveis em LMS, sem um tutor.

Considerando o objetivo de possibilitar a interoperabilidade entre sistemas e a reutilização de informações, optou-se por utilizar a linguagem OWL para representar as ontologias, devido às vantagens já expostas no capítulo 3. Considerando a existência de ferramentas completas para o desenvolvimento de ontologias que suportam esta linguagem, também se percebeu que não seria de grande valia construir uma ferramenta para gerá-las, decidindo-se, assim, utilizar alguma já existente.

Para realizar tal tarefa, foi empregado o Protégé, que é uma ferramenta de *software*

integrada, usada por engenheiros de conhecimento e especialistas para desenvolver sistemas baseados em conhecimento (PROTEGE, 2005). O Protégé permite que o usuário raciocine sobre um determinado domínio em um nível conceitual, sem se preocupar com questões sintáticas. Foram as várias facilidades oferecidas por esta ferramenta para a edição das ontologias e o fato dela suportar a linguagem OWL que levaram a sua escolha para utilização neste trabalho.

O fato de não criar um sistema e uma linguagem própria para construir e armazenar as ontologias permite que outros sistemas sejam construídos com base em suas definições. Além disso, é possível que diferentes sistemas que façam uso dessas ontologias possam trocar informações para reutilizar componentes instrucionais durante a construção ou disponibilização de cursos, com um suporte nas ciências instrucionais.

No capítulo 6 é apresentado um sistema computacional que utiliza as ontologias já referidas para gerar um curso descrito de acordo com um padrão de metadados, o que possibilita que ele seja acessado pelos alunos através de qualquer LMS com suporte ao padrão escolhido. Isso reduz a complexidade de desenvolvimento, pois não é preciso construir toda a estrutura administrativa do curso nem os mecanismos para apresentação do conteúdo, já que elas estão implementadas no próprio LMS. Além disso, o fato de utilizar algum padrão de metadados para representar o curso gerado, faz com que ele possa ser mais facilmente reutilizado, no todo ou em partes, por outras pessoas, sem que elas precisem possuir o ambiente específico para a disponibilização dos conteúdos reaproveitados.

5.3 Modelagem das Ontologias

Nesta seção são descritas as ontologias desenvolvidas para representar as informações envolvidas na construção de cursos e materiais instrucionais para a EBW.

Além da representação visual da estrutura de classes e de algumas instâncias, esta seção apresenta uma descrição formal de cada ontologia. Para tanto, utilizou-se a sintaxe abstrata da linguagem OWL, descrita em PATEL-SCHNEIDER e HORROCKS (2004), que permite a definição de várias informações em uma única estrutura, ao invés de dividi-las em vários axiomas, como nas linguagens baseadas em descrições lógicas. O quadro 26 apresenta os termos utilizados para a representação das ontologias juntamente com o seu significado e um exemplo de utilização.

QUADRO 26 - Termos utilizados para representar as ontologias

Termo	Exemplo	Significado
Class	Class (X)	O termo Class é utilizado para definir uma classe com o nome passado como parâmetro, neste caso, X.
SubClassOf	SubClassOf (Y)	Define que uma classe é subclasse de Y
DisjointClasses	DisjointClasses (X, Y)	Indica que uma instância não pode pertencer às classes X e Y, simultaneamente.
Domain	Domain (X)	Define o domínio da propriedade X, ou seja, a classe ou conjunto de classes a que a propriedade X está vinculada
Range	Range (X)	Define o tipo de valor que uma propriedade pode receber. O parâmetro X pode representar uma classe no caso de uma ObjectProperty ou um tipo de dado no caso de um DatatypeProperty
ObjectProperty	ObjectProperty (X domain (Y) range (Z))	Define uma propriedade X, que está associada à classe Y e cujos valores são instâncias da classe Z.
DatatypeProperty	ObjectProperty (X domain (Y) range (Z))	Define uma propriedade X, que está associada à classe Y e cujos valores são do tipo Z
MinCardinality	MinCardinality (0)	Determina a cardinalidade mínima de uma propriedade, ou seja, quantos valores, no mínimo, devem estar associados a ela. Neste exemplo, a propriedade pode não possuir nenhum valor
MaxCardinality	MaxCardinality (10)	Determina a cardinalidade máxima de uma propriedade, ou seja, quantos valores, no máximo, podem estar associados a ela, neste exemplo, 10.
Transitive	Transitive	Indica que uma propriedade é transitiva, ou seja, se a propriedade X relaciona A e B, e B e C, esta mesma propriedade também é válida para A e C.
Functional	Functional	Indica que uma propriedade pode receber apenas um único valor
Inverse	Inverse (Y)	Indica que a propriedade que está sendo declarada, por exemplo X, é inversa da propriedade Y. Neste caso, se X é válido para as instâncias A e B, então Y é para B e A.
Value	Value (X, Y)	Determina que o valor de uma propriedade X é igual a Y para uma determinada instância.
Individual	Individual (X type Y)	Define uma instância da classe Y que é identificada pelo elemento X.
Namespace ¹³	Namespace (X prefix Y)	Indica que o namespace X é representado pelo prefixo Y na definição de classes, propriedades e instâncias. Quando o prefixo Y é omitido, indica que aquele é o namespace usado para identificar a ontologia que está sendo criada.

¹³ Este termo não é definido na sintaxe abstrata da linguagem OWL, mas foi incorporado à descrição das ontologias para facilitar o entendimento.

5.3.1 Ontologia LOM

A seção 4.6.1, sobre o padrão LOM, destacou que ele não define os valores possíveis para alguns de seus atributos, os quais aceitam qualquer informação de um determinado tipo. Entretanto, outros atributos devem receber um valor dentre um conjunto pré-estabelecido pelo próprio padrão. Para determinar que valores são esses, criou-se uma ontologia auxiliar, denominada LOM, com classes e instâncias que representam os valores que podem ser atribuídos às propriedades dos LO, a fim de descrevê-los. A ontologia foi construída com base no IEEE *Learning Object Metadata RDF Binding* (NILSSON, 2002), que fornece uma representação em RDF para o referido padrão.

A estrutura da ontologia segue as nove categorias do padrão LOM, embora implemente os atributos de apenas três delas. Por isso, como pode ser visto na figura 28, ela possui uma classe LOM, que agrega todos os tipos de metadados, e três subclasses para representar separadamente os metadados das categorias: *General*, *Educational* e *Technical*. Cada categoria também possui um conjunto de subclasses, que representam os metadados de um LO.

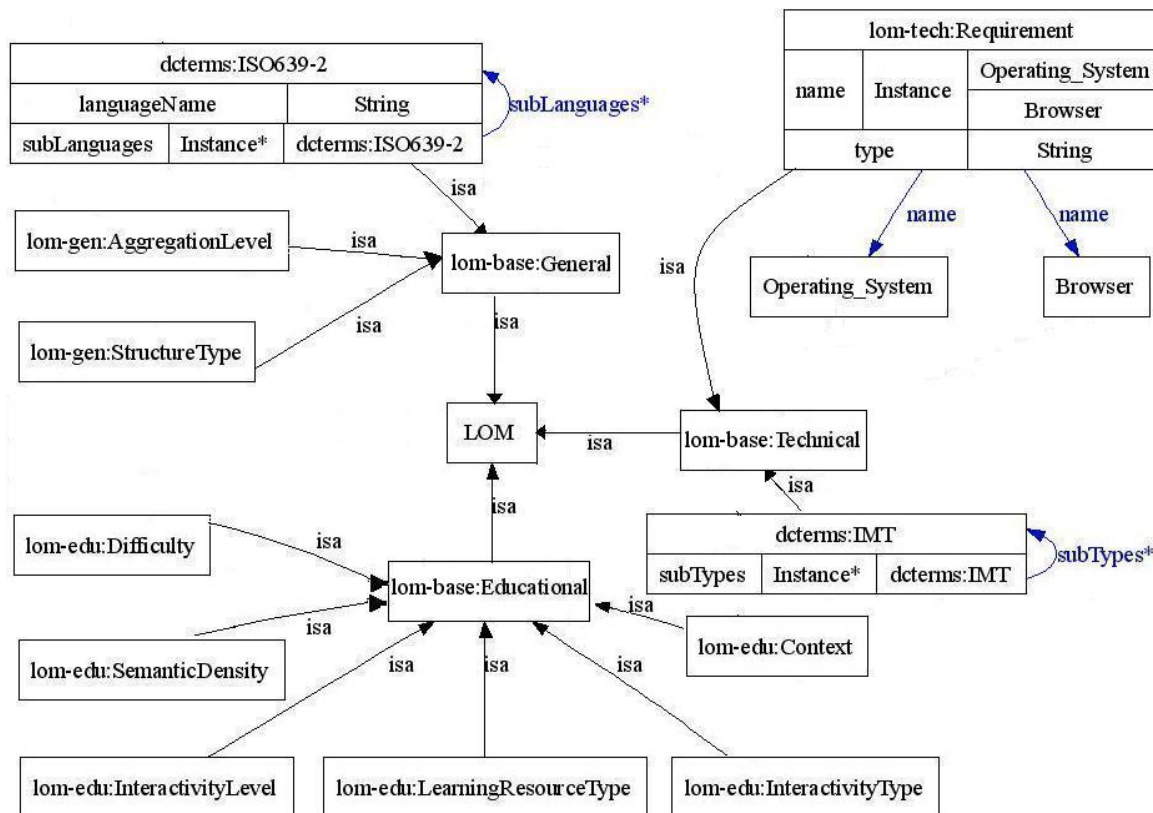


FIGURA 28 - Estrutura de classes da ontologia auxiliar LOM

O quadro 27 descreve, através da sintaxe abstrata da linguagem OWL (quadro 26), a hierarquia de classes da ontologia LOM (figura 28). Na linha 1 está a definição do *namespace* que armazena os termos definidos nessa ontologia; as linhas 2 a 9 definem os prefixos dos demais *namespaces* utilizados na construção dessa ontologia

QUADRO 27 - Classes da ontologia LOM

```

1. Namespace (http://localhost/ontologias/lom.owl#)
2. Namespace (http://ltsc.ieee.org/2002/09/lom-base# prefix lom-base)
3. Namespace (http://ltsc.ieee.org/2002/09/lom-educational#
4.     prefix lom-edu)
5. Namespace (http://ltsc.ieee.org/2002/09/lom-technical#
6.     prefix lom-tech)
7. Namespace (http://ltsc.ieee.org/2002/09/lom-general#
8.     prefix lom-gen)
9. Namespace (http://purl.org/dc/terms/ prefix dcterms)
10. Class (LOM)
11. Class (lom-base14:Educational SubClassOf (LOM))
12. Class (lom-base:Technical SubClassOf (LOM))
13. Class (lom-base:General SubClassOf (LOM))
14. Class (lom-edu:Context SubClassOf (lom:Educational))
15. Class (lom-edu:Difficulty SubClassOf (lom:Educational))
16. Class (lom-edu:InteractivityLevel SubClassOf (lom:Educational))
17. Class (lom-edu:InteractivityType SubClassOf (lom:Educational))
18. Class (lom-edu:LearningResourceType SubClassOf (lom:Educational))
19. Class (lom-edu:SemanticDensity SubClassOf (lom:Educational))
20. Class (dcterms:ISO639-2 SubClassOf (lom:General))
21. Class (lom-gen:AggregationLevel SubClassOf (lom:General))
22. Class (lom-gen:StructureType SubClassOf (lom:General))
23. Class (RequirementType SubClassOf (lom:Technical))
24. Class (dcterms:IMT SubClassOf (lom:Technical))
25. Class (lom-tech:Requirement SubClassOf (lom:Technical))
26. Class (Browser)
27. Class (Operating_System)

```

A partir da linha 10, estão as definições das classes, propriedades e instâncias da ontologia, que inicia com a definição da classe LOM, cuja função é apenas agrupar as subclasses que representam as categorias do padrão LOM. Nas linhas 11 a 13 são definidas

¹⁴ Os prefixos utilizados antes do símbolo “:” nos nomes de classes, propriedades e instâncias indicam o *namespace* onde esta classe, instância ou propriedade está definida.

suas três subclasses, que abrangem, respectivamente, os atributos das categorias *General*, *Educational* e *Technical* de um LO. Nas linhas 14 a 27 são definidas as classes que efetivamente representam os atributos de um LO, juntamente com a identificação de suas superclasses, para indicar a que categoria pertence o atributo representado por aquela classe.

A nomenclatura das classes é baseada nos *namespaces* que estabelecem os nomes das propriedades do padrão LOM. Por exemplo, as instâncias da classe *dcterms:ISO639-2* representam os idiomas que podem ser utilizados para construir um LO, os quais são definidos no *namespace* de mesmo nome. As instâncias da classe *dcterms:IMT* representam os tipos de arquivos que podem ser utilizados para armazenar os LO (*text*, *html*, *jpg*, *application* etc.). Os significados dos nomes das demais classes e os tipos de informações que elas representam nos LO podem ser vistos na seção sobre o padrão LOM (4.6.1), com exceção das classes *Browser* e *Operating_System* (linhas 26 e 27) que foram criadas para descrever, respectivamente, os diferentes tipos de navegador e sistema operacional utilizados na definição dos requisitos para execução de um LO, através da classe *lom-tech:Requirement* (linha 25).

Nem todas as propriedades das três categorias foram incorporadas à ontologia. Algumas delas, porque não são relevantes para o desenvolvimento desta tese e, outras, porque podem receber qualquer valor, como a idade dos participantes, tornando desnecessária a sua representação através de classes e instâncias.

O quadro 28 apresenta a definição das propriedades da ontologia LOM, que são responsáveis por armazenar as informações de cada instância, bem como relacioná-las às instâncias de outras classes.

QUADRO 28 - Propriedades da ontologia LOM

1.	<code>ObjectProperty(subLanguages domain(dcterms:ISO639-2)</code>
2.	<code>range(dcterms:ISO639-2) Transitive minCardinality(0))</code>
3.	<code>DatatypeProperty(languageName domain(dcterms:ISO639-2)</code>
4.	<code>range(string) Functional)</code>
5.	<code>ObjectProperty(subTypes domain(dcterms:IMT) range(dcterms:IMT)</code>
6.	<code>Transitive minCardinality(0))</code>
7.	<code>ObjectProperty(name domain(lom-tech:Requirement)</code>
8.	<code>range(Operating_System) range(Browser) Functional))</code>
9.	<code>DatatypeProperty(type domain(lom-tech:Requirement)</code>
10.	<code>range(string) Functional)</code>

Poucas propriedades são necessárias porque o objetivo desta ontologia é apenas definir os termos do padrão LOM, o que pode ser feito através dos nomes das classes e instâncias. Entre elas estão as propriedades que relacionam uma linguagem as suas sublinguagens (linhas 1 e 2), o nome por extenso de uma linguagem (linhas 3 e 4), os subtipos de um arquivo (linhas 5 e 6) e os tipos de restrições para utilização de um LO, juntamente com seus possíveis valores (linhas 7 a 10).

A figura 29 mostra as instâncias da classe *lom-edu:InteractivityType*, que são responsáveis por determinar quais valores podem ser associados à propriedade *lom-edu:interactivityType*. Esta propriedade determina se o LO é *Expositivo*, o aluno apenas lê, se é *Ativo*, o aluno interage com o LO, respondendo perguntas, realizando experimentos etc., ou se ele é *Misto*.

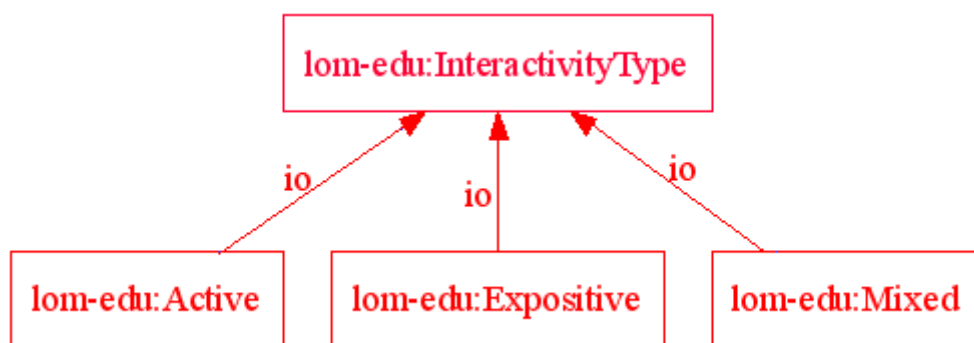


FIGURA 29 - Instâncias da classe *lom-edu:InteractivityType*

Com o objetivo de simplificar a visualização das instâncias, o quadro 29 apresenta informações semelhantes às da figura 29, porém, com base na sintaxe abstrata da linguagem OWL. As linhas 1 a 13 apresentam a criação de instâncias de classes que não possuem propriedades, por isso basta definir o nome da instância e a que classe ela pertence. Conforme já explicitado, enquanto o nome da classe determina a propriedade do padrão LOM que ela descreve, o nome da instância estabelece os possíveis valores desta propriedade.

Nas linhas 1 a 3 estão definidas as mesmas instâncias apresentadas na figura 29, cujos valores são utilizados para indicar o tipo de interação propiciado pelo LO. As linhas 4 a 7 determinam as instâncias utilizadas para descrever o contexto de aplicação de um LO, que pode ser: *Educação Superior*, *Escola*, *Treinamento* ou *Outros*. Já nas linhas 8 a 12, estão as instâncias que designam os termos utilizados para determinar o grau de dificuldade de um LO.

QUADRO 29 - Instâncias da ontologia LOM

1.	Individual(lom-edu:Active type lom-edu:InteractivityType)
2.	Individual(lom-edu:Expositive type lom-edu:InteractivityType)
3.	Individual(lom-edu:Mixed type lom-edu:InteractivityType)
4.	Individual(lom-edu:higher_education type lom-edu:Context)
5.	Individual(lom-edu:school type lom-edu:Context)
6.	Individual(lom-edu:training type lom-edu:Context)
7.	Individual(lom-edu:other type lom-edu:Context)
8.	Individual(lom-edu:VeryEasy type lom-edu:Difficulty)
9.	Individual(lom-edu:Easy type lom-edu:Difficulty)
10.	Individual(lom-edu:MediumDifficult type lom-edu:Difficulty)
11.	Individual(lom-edu:Difficult type lom-edu:Difficulty)
12.	Individual(lom-edu:VeryDifficult type lom-edu:Difficulty)
13.	Individual (zip type (dcterms:IMT))
14.	Individual(en type dcterms:ISO639-2 value(languageName English)
15.	value(subLanguages en-AU) value(subLanguages en-US)
16.	value(subLanguages en-GB))
17.	Individual(en-US type dcterms:ISO639-2
18.	value(languageName American English))
19.	Individual (audio type (dcterms:IMT) value (subTypes mpeg)
20.	value (subTypes x-wav) values (subTypes midi))

Neste quadro pode-se perceber a utilização das propriedades definidas no quadro 28, entre as linhas 14 e 20. Eles são utilizados para definir o nome por extenso de um idioma (linha 14 e 18), bem como os idiomas derivados do mesmo (linhas 15 e 16). Por fim, uma outra propriedade é utilizada na linha 20 para definir os tipos de arquivos que são subtipos de um arquivo de áudio, declarado na linha 19.

Todas as demais classes apresentadas no quadro 27 possuem instâncias, a exemplo das apresentadas no quadro 29, que estabelecem os possíveis valores dos atributos do padrão LOM. Para tomar conhecimento do tipo de informação determinada por cada classe ou dos nomes das instâncias que representam seus possíveis valores, basta consultar a seção 4.6.1 ou o próprio padrão LOM, já que todas as informações seguem suas recomendações.

Esta ontologia foi criada para facilitar manutenções futuras, caso o padrão LOM venha a sofrer mudanças com a inserção, alteração ou exclusão de possíveis propriedades ou valores. Caso isto ocorra, basta criar, eliminar ou alterar as classes e instâncias desta ontologia para que a mudança tenha efeito nos demais módulos da arquitetura desenvolvida. Por ser uma ontologia auxiliar, que visa definir os termos utilizados nas

outras ontologias, não existem outras propriedades a não ser o nome das classes e das instâncias, nem outros relacionamentos a não ser os apresentados no quadro 28.

5.3.2 Ontologia Pedagógica

A ontologia pedagógica foi construída com o objetivo de representar o conhecimento sobre o processo de aprendizagem. A partir de suas informações, pretende-se auxiliar as pessoas a construírem e a estruturarem as atividades dos cursos de forma a facilitar a aprendizagem dos alunos e fazer com que eles atinjam os objetivos estabelecidos. Suas informações são oriundas das propostas de GAGNÉ (1985), BLOOM *et. al.* (1976) e SASKATCHEWAN EDUCATION (1991), apresentadas na seção 4.8. A figura 30 apresenta as classes e propriedades desta ontologia, as quais também são definidas através da sintaxe abstrata da linguagem OWL, no quadro 30.

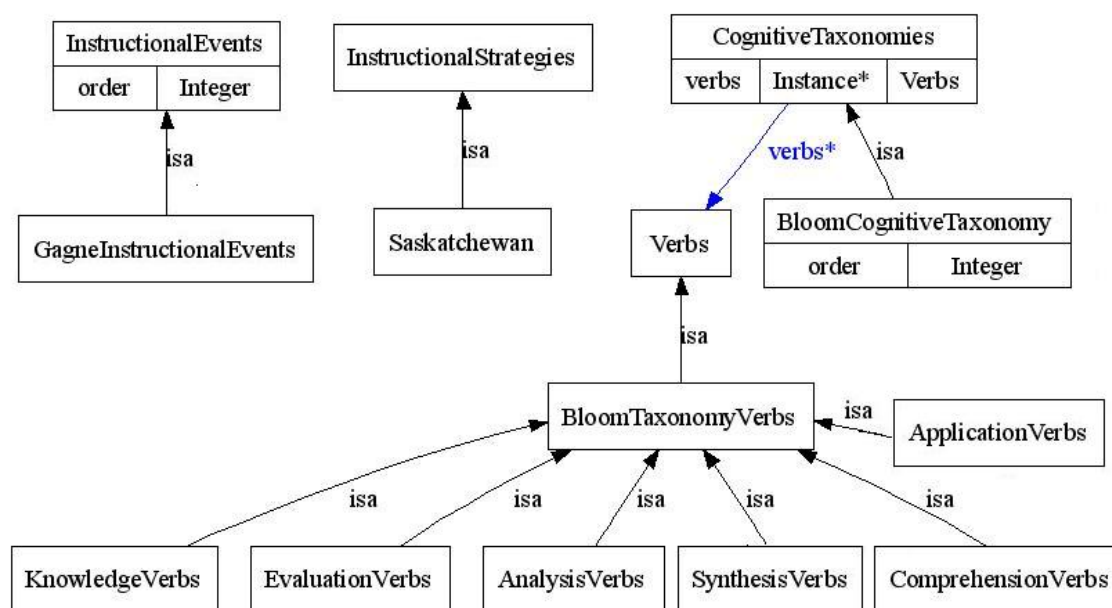


FIGURA 30 - Classes da ontologia pedagógica

Esta ontologia apresenta definições em três aspectos do processo instrucional. A primeira, refere-se à estratégia instrucional utilizada (classes *InstructionalStrategies* e *Saskatchewan*), em seguida, está a definição das fases que devem ser desenvolvidas durante o processo instrucional (*InstructionalEvents* e *GagneInstructionalEvents*) e, por fim, tem-se a categorização dos níveis de operações mentais (*CognitiveTaxonomies* e *BloomCognitiveTaxionomy*), juntamente com os verbos associados a cada uma delas (*Verbs*, *BloomTaxionomyVerbs* e suas subclasses), que indicam ações que os alunos devem realizar para demonstrar que atingiram um nível de conhecimento em determinado assunto.

QUADRO 30 - Classes da ontologia pedagógica

1.	Namespace (http://localhost/ontologias/pedagogical.owl#)
2.	Class (CognitiveTaxonomies)
3.	Class (BloomCognitiveTaxonomy SubClassOf (CognitiveTaxonomies))
4.	Class (InstructionalEvents)
5.	Class (GagneInstructionalEvents SubClassOf (InstructionalEvents))
6.	Class (InstructionalStrategies)
7.	Class (Saskatchewan SubClassOf (InstructionalStrategies))
8.	Class (Verbs)
9.	Class (BloomTaxonomyVerbs SubClassOf (Verbs))
10.	Class (KnowledgeVerbs SubClassOf (BloomTaxonomyVerbs))
11.	Class (ComprehensionVerbs SubClassOf (BloomTaxonomyVerbs))
12.	Class (ApplicationVerbs SubClassOf (BloomTaxonomyVerbs))
13.	Class (AnalysisVerbs SubClassOf (BloomTaxonomyVerbs))
14.	Class (SynthesisVerbs SubClassOf (BloomTaxonomyVerbs))
15.	Class (EvaluationVerbs SubClassOf (BloomTaxonomyVerbs))
16.	ObjectProperty (verbs domain (CognitiveTaxonomies) range (Verbs))
17.	DatatypeProperty (order domain (CognitiveTaxonomies)
18.	domain (InstructionalEvents) Functional minCardinality (1))

Embora nesta ontologia as classes não estejam relacionadas, na ontologia de domínio elas estão, pois suas instâncias definem os termos utilizados para descrever os objetivos, atividades e objetos de aprendizagem dos cursos desenvolvidos.

A criação de uma classe e apenas uma subclasse para representar as informações na ontologia justifica-se pela facilidade de expansão proporcionada. Por exemplo, se fosse criada apenas a classe *GagneInstructionalEvents*, suas instâncias representariam da mesma forma os eventos instrucionais. Mas neste caso, se posteriormente for criada outra classe baseada em outra teoria instrucional, com instâncias que representem o mesmo tipo de informação, será necessário readequar todos os módulos dos sistemas que utilizam a ontologia, pois, para eles, apenas as instâncias da classe *GagneInstructionalEvents* denotam termos quem definem os tipos de eventos instrucionais.

Da forma que a ontologia foi construída, basta criar uma nova classe, que seja subclasse de *InstructionalEvents*, para que suas instâncias sejam automaticamente reconhecidas como termos que definem os eventos instrucionais. Os sistemas que utilizam a ontologia podem apresentar as duas terminologias e solicitar que o usuário escolha qual das teorias instrucionais (subclasses) deseja utilizar para descrever suas atividades de aprendizagem, ou fazer com que o usuário siga uma delas, se julgar mais adequado.

A classe *Saskatchewan* representa uma forma de classificar as estratégias instrucionais, embasada no trabalho de SASKATCHEWAN EDUCATION (1991). A figura 31 ilustra suas instâncias, que estabelecem os termos que devem ser utilizados para definir a(s) estratégia(s) utilizada(s) em um curso. Também com o objetivo de facilitar a inserção de novas classificações de estratégias instrucionais, foi criada a classe *InstructionalStrategies* para agrupar diferentes classificações e a subclasse *Saskatchewan*, que representa uma classificação específica.

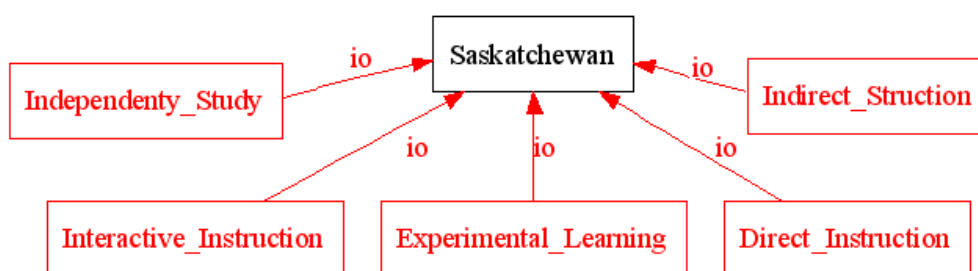


FIGURA 31 - Instâncias da classe *Saskatchewan*

Na ontologia pedagógica não é disposta mais nenhuma informação sobre a classe *Saskatchewan* e suas instâncias, além de uma descrição textual do significado de seus termos, baseada na descrição das estratégias apresentadas em 4.8. Estas descrições podem ser utilizadas pelas pessoas que virem a desenvolver sistemas computacionais que utilizem esta ontologia, para entender o seu significado e definir a melhor forma de utilizá-las.

Conforme já mencionado, o processo instrucional deve cumprir algumas etapas ou eventos para levar os alunos a atingirem seus objetivos, que variam de autor para autor e dependem da estratégia instrucional aplicada. Na seção 4.8 foram descritos tais eventos, com base no trabalho de GAGNÉ (1985). Visando criar uma terminologia padrão para nomear os eventos, foi criada a classe *GagneInstructionalEvents*, cujas instâncias constituem os nomes atribuídos a cada um dos nove eventos, como mostra a figura 32. Além do nome, cada instância contém um atributo denominado *order* para definir a ordem em que os eventos devem acontecer.

Estes termos são utilizados para designar quais eventos de aprendizagem são proporcionados aos alunos por uma atividade de aprendizagem. Através deles, o usuário pode verificar se as atividades de aprendizagem selecionadas para o curso e a ordem que elas estão dispostas são coerentes. Com estas definições, é possível advertir o construtor de um curso que, por exemplo, não inseriu uma atividade de aprendizagem para despertar o interesse dos alunos ou que a tenha inserido apenas no final.

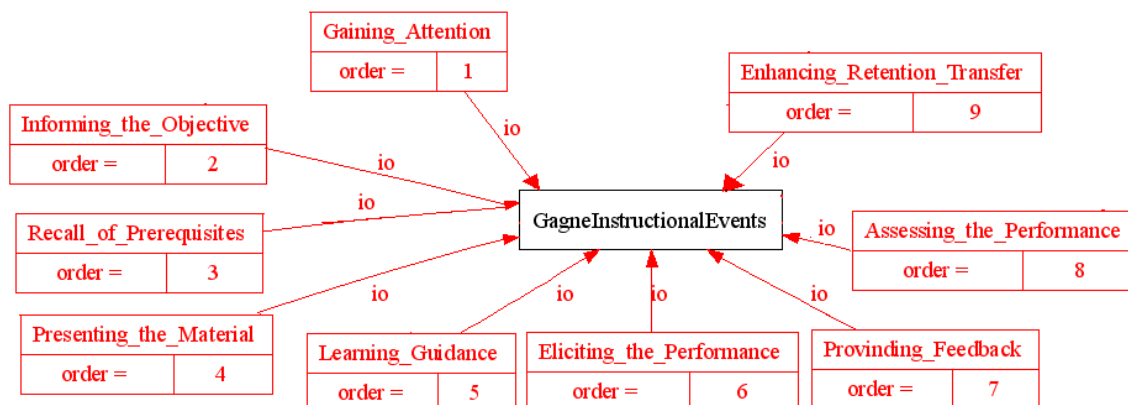


FIGURA 32 - Instâncias da classe *GagneInstructionalEvents*

A advertência não impede, necessariamente, que o usuário crie o curso de tal maneira, pois é possível apenas alertá-lo para que ele inclua outras atividades, se julgar necessário. A escolha de impedir a criação do curso, cujas atividades não possuem os eventos instrucionais estabelecidos na ontologia, ou de apenas avisar ao usuário, não precisa estar formalizada na ontologia. Ela compete às pessoas que implementam os sistemas que fazem uso de suas definições, as quais podem optar por qualquer uma das possibilidades.

O último aspecto contemplado na ontologia pedagógica refere-se à classificação dos diferentes níveis de operações mentais que um aluno pode realizar. Para isso também foram criadas duas classes: *CognitiveTaxonomies* – que engloba diferentes classificações sobre os níveis de operações mentais e a subclasse *BloomCognitiveTaxonomy* – cujas instâncias definem as categorias da classificação proposta por BLOOM *et. al.* (1976), juntamente com a classe *None*, que representa a completa falta de conhecimento sobre o assunto.

Esta classificação é utilizada em dois momentos distintos: para indicar a categoria de um objetivo de aprendizagem, ou seja, para definir o nível das operações mentais (nível de conhecimento) que o aluno deve ser capaz de realizar sobre determinado assunto após atingir o objetivo proposto e para definir o nível de conhecimento que um aluno já possui sobre um conceito e o nível que se deseja que ele atinja após concluir as atividades instrucionais. Também com relação aos objetivos de aprendizagem, as instâncias da classe *BloomCognitiveTaxonomy* (figura 33) são utilizadas para definir o nível de conhecimento mínimo que o aluno precisa possuir sobre um assunto para acompanhar as atividades propostas associadas a ele.

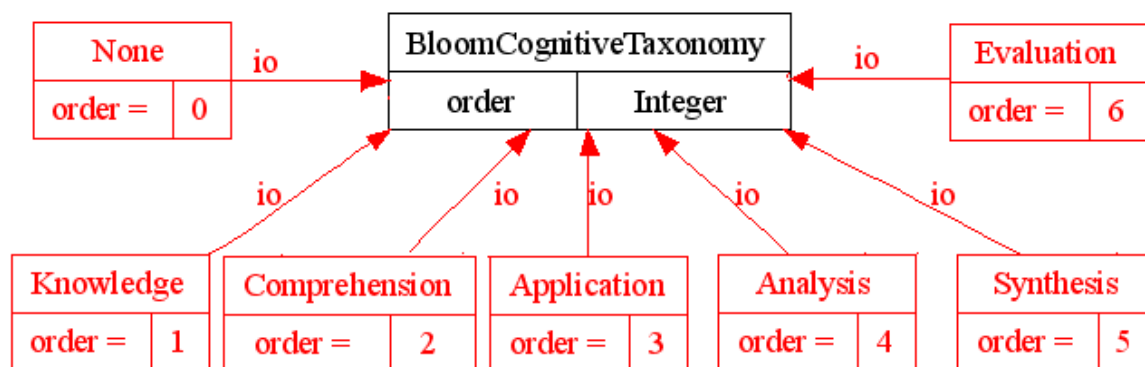


FIGURA 33 - Instâncias da classe *BloomCognitiveTaxonomy*

É importante não confundir esta informação com os pré-requisitos constantes na ontologia de domínio apresentada posteriormente. Enquanto os pré-requisitos definem os conceitos que um aluno deve conhecer para aprender novos conceitos, o nível de conhecimento mínimo estabelece o quanto o aluno deve saber sobre o próprio conceito para ser capaz de realizar operações mentais mais complexas sobre ele.

Por exemplo, uma pessoa que já sabe somar, dividir e comparar valores, mas não conhece a fórmula para calcular a média aritmética, não é capaz de verificar se ela foi aprovada ou não se lhe disserem que “para ser aprovada, a média de suas avaliações deve ser superior a 7”. Embora não exista um outro conceito pré-requisito que a pessoa desconheça, ela não será capaz de resolver o problema, aplicando a fórmula da média aritmética (conhecimento no terceiro nível – *Application*) que ela não conhece (conhecimento no primeiro nível - *Knowledge*).

Desta forma, cruzando as informações pode-se encontrar os objetivos mais adequados para os diferentes tipos de alunos, comparando: o nível de conhecimento que o aluno possui com o nível de conhecimento necessário para acompanhar as atividades que levam a atingir um objetivo de aprendizagem e o nível de conhecimento que se espera que o aluno alcance com a categoria do objetivo de aprendizagem.

O grau de complexidade envolvido em cada um dos níveis é definido através do atributo *order*. Portanto, as instâncias com valores maiores para este atributo indicam as categorias que exigem raciocínios mais complexos do aluno. O valor “0” está associado à instância que determina que *Nenhum* conhecimento se tem ou se deve ter sobre o assunto (*None*).

O exemplo a seguir visa ilustrar uma das formas de aplicação das informações definidas na classe *BloomCognitiveTaxonomy* em sistemas de autoria para EBW. Supondo que uma pessoa esteja construindo um curso para ensinar os alunos a identificar e resolver

problemas utilizando o valor da média aritmética. Ao procurar atividades já prontas para reutilizá-las, o sistema sugere a inclusão de uma atividade que ensina a calcular o valor da média. Baseada apenas no conceito abordado nas atividades, que é o mesmo, a sugestão do programa não está errada. Porém, ao considerar os objetivos e, principalmente, o nível de operação mental exigido do aluno, pode não fazer sentido reutilizar a atividade sugerida, pois se espera que os alunos cheguem ao terceiro nível da taxionomia de Bloom (*Application*) e a atividade proporciona apenas que o aluno chegue ao segundo (*Comprehension*).

Seguindo com o exemplo, podem ser consideradas duas situações distintas: uma em que os alunos ainda não sabem como calcular a média; e outra que eles já possuem este conhecimento. Na primeira situação, deve estar definido no modelo do aluno (ontologia do estudante apresentada posteriormente) que ele não possui *Nenhum* conhecimento sobre o assunto. Neste caso, a atividade sugerida pelo sistema deve ser incorporada ao curso, pois ela serve de base para que o aluno consiga acompanhar outras atividades que devem ser inseridas posteriormente, para fazer com que o aluno chegue ao nível de conhecimento desejado (*Application*).

Já na segunda situação, se estiver corretamente definido no modelo do aluno que ele já sabe calcular a média aritmética, a atividade sugerida deve ser desconsiderada, porque ela proporciona um nível de conhecimento sobre o assunto que o aluno já possui. Eventualmente, ela pode ser inserida apenas para lembrar o que o aluno já sabe sobre o assunto, proporcionando um dos eventos de aprendizagem descrito por Gagné.

Por isso, ao associar as instâncias da classe *BloomCognitiveTaxonomy*, ou genericamente as instâncias das subclasses de *CognitiveTaxonomies*, ao modelo do aluno e às categorias dos objetivos de aprendizagem, que por sua vez são alcançados pela realização de atividades de aprendizagem, é possível construir mecanismos mais elaborados para busca e reutilização de tais componentes.

Juntamente com a taxionomia apresentada, foram criadas a classe *Verbs*, a subclasse *BloomTaxionomyVerbs* e mais seis subclasses de *BloomTaxionomyVerbs*. As instâncias das seis subclasses denotam verbos que representam ações típicas que exigem o nível de operação mental requerido por objetivos de cada categoria. Estes verbos podem ser utilizados, por exemplo, para validar os objetivos de aprendizagem, verificando se em seu título ou descrição não aparece algum verbo que caracterize uma categoria diferente da sua. A figura 34 apresenta as instâncias da classe *KnowledgeVerbs*, que estabelecem os verbos característicos da primeira categoria da taxionomia de Bloom (*Knowledge*).

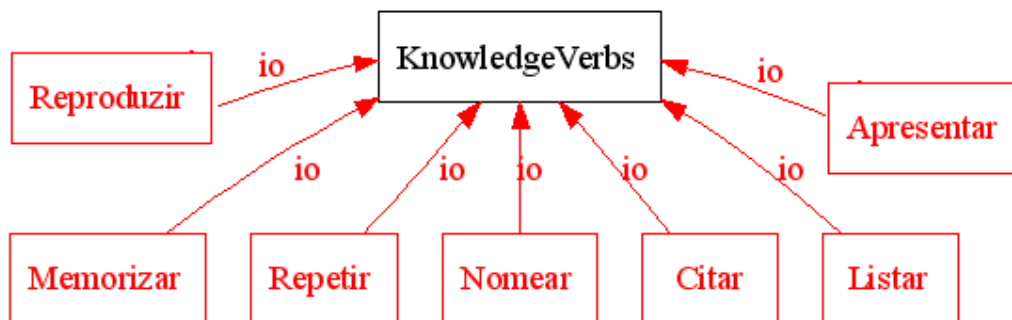


FIGURA 34 - Instâncias da classe *KnowledgeVerbs*

Pelos mesmos motivos expostos na ontologia LOM, as classes desta ontologia não possuem atributos, com exceção do atributo *order*, já citado, constante nas classes *InstructionalEvents* e *BloomCognitiveTaxonomy*, e da propriedade que relaciona os verbos a uma categoria da taxionomia de Bloom.

No quadro 31 são listadas as instâncias das classes da ontologia pedagógica apresentadas nas figuras 31 a 34. A partir da linha 31, estão definidos alguns verbos relacionados a todas as categorias da taxionomia de Bloom. Para simplificar a leitura, os demais verbos foram omitidos já que eles podem ser encontrados em BLOOM *et. al* (1976). O significado do termo definido em cada instância pode ser obtido na seção 4.8 (Teorias Educacionais) ou, diretamente no trabalho dos autores referenciados naquela seção.

5.3.3 Ontologia do Estudante

A ontologia do estudante tem como objetivo definir os termos utilizados para caracterizar os tipos de estudantes que utilizam os sistemas de EBW. Para isso, seus dados são descritos sob duas dimensões, uma pedagógica e outra técnica. Como pode ser visto na figura 35, que apresenta as classes desta ontologia, um perfil de estudante possui um título e está associado a duas instâncias, uma da classe *Pedagogical* e outra da classe *Technical*.

As propriedades destas classes registram as características dos alunos referentes às duas dimensões, sendo a maioria delas baseada no padrão LOM. Assim, como as propriedades que descrevem os LO também são baseadas neste padrão, é possível comparar os dados no momento de selecionar os LO de um curso, visando melhor atender às necessidades do aluno.

QUADRO 31 - Exemplos de Instâncias da ontologia pedagógica

1. Individual(None type BloomCognitiveTaxonomy value(order 0))
2. Individual(Knowledge type BloomCognitiveTaxonomy value(order 1))
3. Individual(Comprehension type BloomCognitiveTaxonomy value(order 2))
4. Individual(Application type BloomCognitiveTaxonomy value(order 3))
5. Individual(Analysis type BloomCognitiveTaxonomy value(order 4))
6. Individual(Synthesis type BloomCognitiveTaxonomy value(order 5))
7. Individual(Evaluation type BloomCognitiveTaxonomy value(order 6))
8. Individual(Gaining_Attention type GagneInstructionalEvents
9. value(order 1))
10. Individual(Informing_the_Objective type GagneInstructionalEvents
11. value(order 2))
12. Individual(Recall_of_Prerequisites type GagneInstructionalEvents
13. value(order 3))
14. Individual(Presenting_the_Material type GagneInstructionalEvents
15. value(order 4))
16. Individual(Learning_Guidance type GagneInstructionalEvents
17. value(order 5))
18. Individual(Eliciting_the_Performance type GagneInstructionalEvents
19. value(order 6))
20. Individual(Providing_Feedback type GagneInstructionalEvents
21. value(order 7))
22. Individual(Assessing_the_Performance type GagneInstructionalEvents
23. value(order 8))
24. Individual(Enhancing_Retention_Transfer type
25. GagneInstructionalEvents value(order 8))
26. Individual(Direct_Instruction type Saskatchewan)
27. Individual(Indirect_Instruction type Saskatchewan)
27. Individual(Interactive_Instruction type Saskatchewan)
29. Individual(Experimental_Learning type Saskatchewan)
30. Individual(Independenty_Study type Saskatchewan)
31. Individual(Apresentar type KnowledgeVerbs)
32. Individual(Citar type KnowledgeVerbs)
33. Individual(Explicar type ComprehensionVerbs)
34. Individual(Descrever type ComprehensionVerbs)
35. Individual(Aplicar type ApplicationVerbs)
36. Individual(Resolver type ApplicationVerbs)
37. Individual(Classificar type AnalysisVerbs)
38. Individual(Organizar type AnalysisVerbs)
39. Individual(Formular type SynthesisVerbs)
40. Individual(Comparar type EvaluationVerbs)

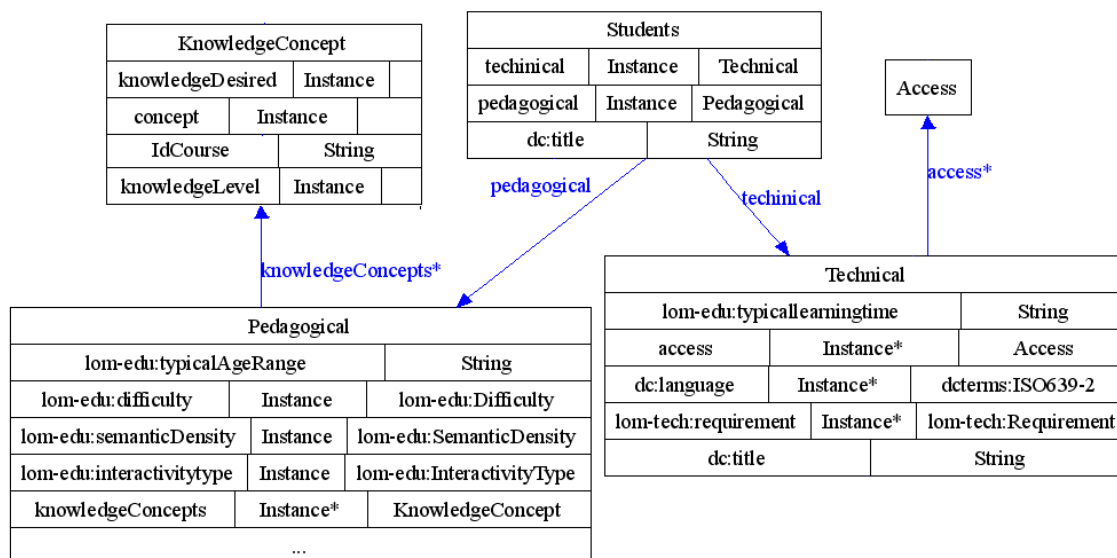


FIGURA 35 - Estrutura de classes da ontologia do estudante.

A separação das características em duas dimensões foi realizada porque em muitos casos os alunos diferem em apenas uma delas, sendo a outra semelhante. Mesmo assim, como são enquadrados em perfis distintos, elaboram-se atividades completamente diferentes para eles. Além disso, as diferentes combinações entre as dimensões técnicas e pedagógicas levam à criação de novos perfis de aluno, sem precisar redefinir todas as suas características, o que também facilita a sua reutilização, por possuírem um nível de granularidade maior.

Supondo que os alunos de um curso presencial precisem realizar em um laboratório, todos ao mesmo tempo, determinada atividade, significa que eles têm as mesmas condições de acesso ao curso, tempo para realização etc. Portanto, basta criar um único perfil técnico para todos os alunos. Entretanto, o professor pode criar dois ou mais perfis pedagógicos, se ele perceber que existem grupos distintos em relação à facilidade para assimilação do conteúdo, conhecimento prévio sobre o assunto, estilo de aprendizagem etc. Os diferentes perfis pedagógicos¹⁵ podem ser associados ao único perfil técnico para criar dois ou mais perfis de aluno, sem a necessidade de redefinir todas as informações do perfil técnico.

O quadro 32 apresenta as classes da ontologia do estudante presentes na figura 35, juntamente com as propriedades e instâncias relacionadas à dimensão técnica de seu perfil.

¹⁵ Quando for utilizado o termo perfil pedagógico ou perfil técnico, faz-se referência a uma das dimensões do perfil do aluno. Já o termo perfil, usado sem especificação, refere-se ao perfil do aluno como um todo, englobando as duas dimensões.

QUADRO 32 - Classes, propriedades e instâncias da dimensão técnica

```

1. Namespace (http://localhost/ontologias/students.owl#)
2. Namespace (http://localhost/ontologias/pedagogical.owl#
3.     prefix pedagogical)
4. Namespace (http://localhost/ontologias/concepts.owl# prefix concepts)
5. Namespace (http://purl.org/dc/elements/1.1/ prefix dc)
6. Class (Students)
7. Class (Pedagogical)
8. Class (Technical)
9. DatatypeProperty (title domain (Students) domain (Pedagogical)
10.     domain (Technical) range (string) Functional)
11. ObjectProperty (pedagogical domain (Students) range (Pedagogical))
12. ObjectProperty (technical domain (Students) range (Technical))
13. ObjectProperty (dc:language domain (Technical) range (dcterms:ISO639-2))
14. DatatypeProperty (lom-edu:typicallearningtime domain (Technical)
15.     range (duration))
16. ObjectProperty (lom-tech:requirement domain (Technical)
17.     range (lom-tech:Requirement))
18. ObjectProperty (Access domain (Technical) range (Access))
19. Class (Access)
20. Individual (ADSL type Access)
21. Individual (Fibra_Otica type Access)
22. Individual (Radio type Access)
23. Individual (Linha_Discada type Access)

```

A linha 1 define o *namespace* utilizado para registrar os termos definidos na ontologia. Nas linhas 2 a 4, os *namespaces* das ontologias pedagógica e de domínio (*Concepts*) são associados aos seus respectivos prefixos. Nas linhas 6 a 8, estão as definições das três classes que representam, respectivamente, os diferentes perfis dos estudantes de um curso e as dimensões pedagógicas e técnicas que compõem estes perfis. Em seguida, nas linhas 9 e 10, está a definição da propriedade título (*title*) existente nas três classes citadas. As linhas 11 e 12 vinculam as classes *Pedagogical* e *Technical* à classe *Students*, enquanto as linhas 13 a 18 determinam as propriedades da classe *Technical*.

Os valores que elas podem assumir estão listados na definição da ontologia LOM e na definição do padrão homônimo, exceto para a classe *Access* (linha 19), que foi criada para padronizar a nomenclatura utilizada para as formas de acesso à Internet, cujas instâncias estão listadas nas linhas 20 a 23.

Embora praticamente todas as propriedades da dimensão técnica, assim como as da

dimensão pedagógica sejam baseadas no padrão LOM, ressalta-se que o seu significado na ontologia do estudante é diferente do referido padrão, onde elas representam as características de um LO. Portanto, as explicações ao lado das propriedades refletem o entendimento que se tem das mesmas e de seus valores nesta ontologia. O seu significado de acordo com o padrão LOM será apresentado em momento oportuno, quando forem descritas as propriedades da classe *LearningObjects* na ontologia de domínio.

- *dc:title* – título ou nome atribuído ao perfil técnico;
- *access* – tipo de acesso à Internet que o aluno utiliza. É a única propriedade não existente no padrão LOM;
- *dc:language* – idioma que o aluno utiliza para ler os materiais;
- *lom-edu:typicallearningtime* - tempo de que o aluno dispõe para realizar as atividades;
- *lom-tech:requirement* – instância definida no padrão LOM que estabelece o sistema operacional e o *browser* requerido para acessar um LO. Esta ontologia armazena o sistema operacional e o *browser* utilizados pelo estudante, para sugerir LO que possam ser acessados através deles.

Como pôde ser visto na descrição das propriedades da classe *Technical*, esta dimensão do perfil foi implementada seguindo o modelo estereótipo. Já a dimensão pedagógica une os modelos de estereótipo e *overlay*, visando melhor representar as características e necessidades dos alunos, como pode ser visto no quadro 33.

Em um primeiro momento, através dos atributos baseados no padrão LOM (linhas 1 a 12), são criados estereótipos para os alunos. Os valores que eles podem assumir são estabelecidos pelo padrão LOM e descritos na ontologia de mesmo nome. Já a interpretação de cada um dos atributos na ontologia do estudante é apresentada a seguir:

- *dc:title* – título ou nome atribuído ao perfil pedagógico;
- *lom-edu:context* – Seus possíveis valores são *Escola*, *Educação Superior*, *Treinamento* ou *Outros* e representam o motivo pelo qual o aluno está fazendo o curso;
- *lom-edu:difficulty* – determina o nível de dificuldade que as atividades propostas aos alunos devem possuir;
- *lom-edu:interactivitytype* – estabelece se os alunos preferem atividades expositivas, atividades com as quais eles possam interagir ou atividades

mistas;

- *lom-edu:semanticDensity* – determina se os conteúdos apresentados devem ser mais condensados, com mais informações em menos tempo ou se o conteúdo deve ser apresentado separadamente;
- *lom-edu:typicalAgeRange* – indica a idade média dos alunos deste perfil.

QUADRO 33 - Classes e Propriedades da dimensão pedagógica

1.	ObjectProperty(lom-edu:context domain(Pedagogical)
2.	range(lom-edu:Context) Functional)
3.	ObjectProperty(lom-edu:difficulty domain(Pedagogical)
4.	range(lom-edu:Difficulty) Functional)
5.	ObjectProperty(lom-edu:interactivitytype domain(Pedagogical)
6.	range(lom-edu:Interactivitytype) Functional)
7.	ObjectProperty(lom-edu:semanticDensity domain(Pedagogical)
8.	range(lom-edu:SemanticDensity) Functional)
9.	DatatypeProperty(lom-edu:typicalAgeRange domain(Pedagogical)
10.	range(string) Functional)
11.	ObjectProperty(lom-edu:semanticDensity domain(Pedagogical)
12.	range(lom-edu:SemanticDensity) Functional)
13.	ObjectProperty(knowledgeConcepts domain(Pedagogical)
14.	range(KnowledgeConcept))
15.	Class(KnowledgeConcept)
16.	ObjectProperty(concept domain(KnowledgeConcept)
17.	range(concepts:Concepts))
18.	ObjectProperty(knowledgeLevel domain(KnowledgeConcept)
19.	range(pedagogical:CognitiveTaxonomies))
20.	ObjectProperty(knowledgeDesired domain(KnowledgeConcept)
21.	range(pedagogical:CognitiveTaxonomies))

Além de criar o estereótipo dos alunos e definir suas características, é possível determinar o nível de conhecimento que os alunos de cada perfil pedagógico possuem sobre os conceitos abordados em um curso, através da propriedade *knowledgeConcept*. Como pode ser visto no quadro 33 (linhas 15 a 21), esta propriedade associa uma ou mais instâncias da classe *KnowledgeConcepts* a uma instância da classe *Pedagogical*. Cada instância da classe *KnowledgeConcepts* possui uma referência para um dos conceitos definidos na ontologia de domínio, a indicação do nível de conhecimento que o aluno possui sobre aquele conceito e o nível que ele deve alcançar após concluir o curso.

As definições destes níveis são baseadas na ontologia pedagógica, através das

instâncias das subclasses de *CognitiveTaxonomies*. Isso facilita a busca e a reutilização de objetivos e atividades de aprendizagem, uma vez que a terminologia utilizada para determinar o nível de conhecimento que o aluno possui e o nível onde quer chegar é a mesma que define o nível de conhecimento prévio que ele deve possuir para acompanhar as atividades que levam a um objetivo de aprendizagem, bem como o nível que ele deve atingir após alcançar tal objetivo.

As propriedades que compõem os perfis pedagógicos e técnicos podem ser utilizadas durante a busca de LO, restringindo os resultados e possibilitando que apenas os que possuem características compatíveis com os usuários sejam incluídos no curso.

5.3.4 Ontologia de Domínio

As informações da ontologia do domínio podem ser divididas em duas dimensões: a primeira, define o mapa conceitual de uma área do conhecimento, com seus conceitos e os relacionamentos entre eles; a segunda, representa os objetivos, atividades e objetos de aprendizagem utilizados na construção dos cursos. A figura 36 mostra as classes desta ontologia, com alguns de seus atributos e relacionamentos.

Estas mesmas classes, bem como os *namespaces* utilizados nesta ontologia são apresentados no quadro 34.

QUADRO 34 - Classes da ontologia de domínio

1.	Namespace (http://localhost/ontologias/concepts.owl#)
2.	Namespace (http://localhost/ontologias/pedagogical.owl#
3.	prefix pedagogical)
4.	Namespace (http://localhost/ontologias/students.owl# prefix students)
5.	Class (Courses)
6.	Class (Organizations)
7.	Class (LearningObjectives)
8.	Class (LearningActivities)
9.	Class (LearningObjects)
10.	Class (Concepts)

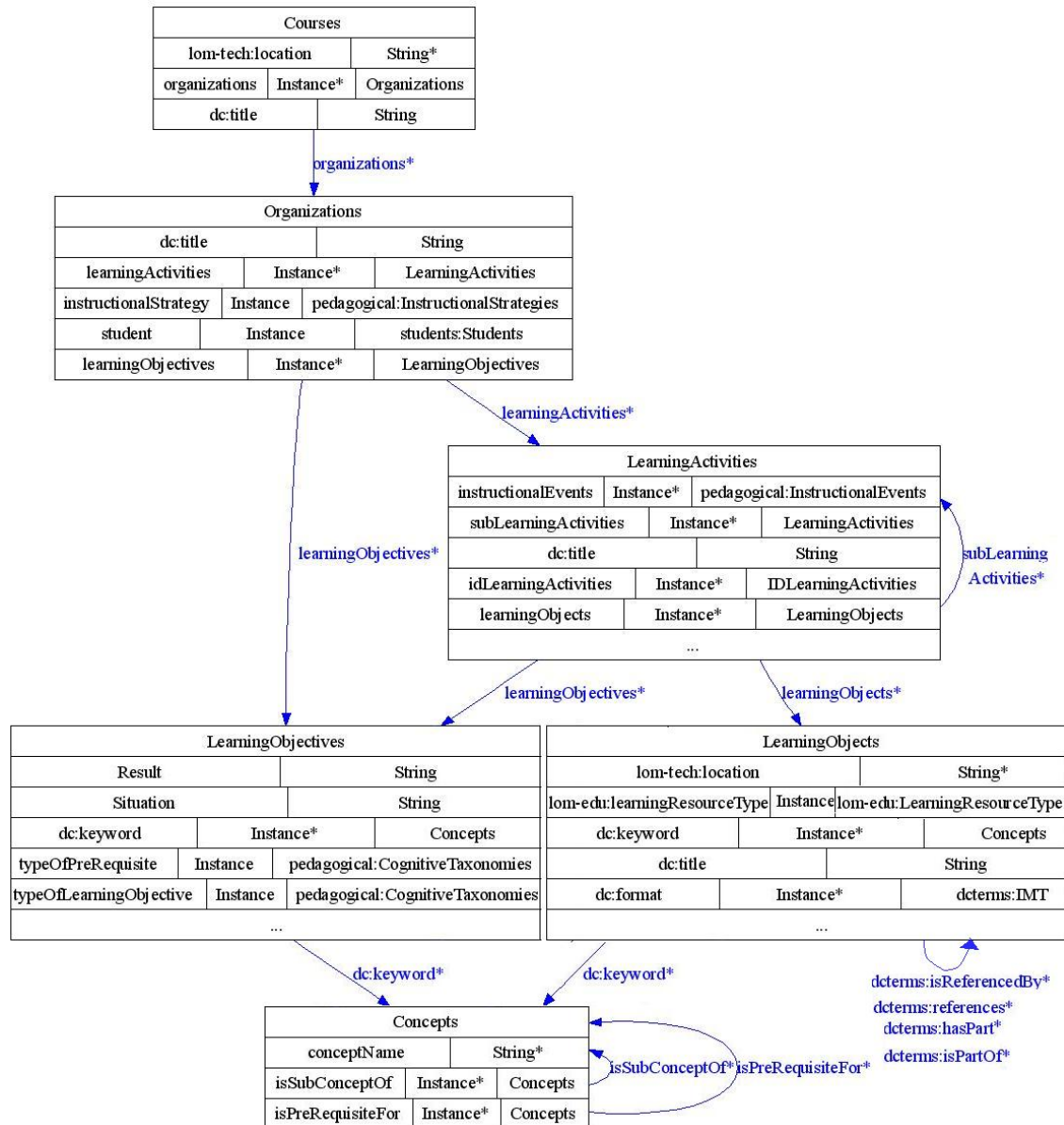


FIGURA 36 - Classes da ontologia de domínio

As instâncias da classe *Concepts* representam os conceitos de um domínio do conhecimento. Cada uma delas possui um nome e pode estar relacionada a outros conceitos de duas maneiras: como pré-requisito (propriedade *isPreRequisiteFor*) ou como subconceito (*isSubConceptOf*), formando o mapa conceitual do domínio. Estes dois tipos de relacionamentos possibilitam a representação das informações necessárias para subsidiar a seleção dos conceitos abordados em um curso, evitando que ele fique inconsistente.

Nas linhas 1 a 6, do quadro 35, são declaradas as duas propriedades citadas da classe *Concepts*, além da propriedade que armazena o seu nome (*conceptName*). Percebe-se que a cardinalidade mínima da propriedade *conceptName* é igual a um, que determina que todo

conceito deve possuir um nome. Como a cardinalidade máxima não é estabelecida, um conceito pode possuir vários nomes, o que permite a associação de vários termos sinônimos a uma única instância.

Tanto a propriedade *isPreRequisiteFor* quanto a *isSubConceptOf* são transitivas (linhas 4 e 6), o que designa que um conceito X é pré-requisito para o conceito Z (*isPreRequisiteFor*), se X for pré-requisito para Y e Y for pré-requisito para Z. O mesmo raciocínio é válido para a propriedade *isSubConceptOf*.

QUADRO 35 - Propriedades e instâncias da classe *Concepts*

1.	DatatypeProperty(conceptName domain(Concepts) range(string)
2.	MinCardinality(1))
3.	ObjectProperty(isPreRequisiteFor domain(Concepts) range(Concepts)
4.	Transitive)
5.	ObjectProperty(isSubConceptOf domain(Concepts) range(Concepts)
6.	Transitive)
7.	Individual(Estatística type Concepts value(conceptName Estatística)
8.	Individual(EstatísticaDescritiva type Concepts
9.	value(conceptName Estatística Descritiva)
10.	value(isSubConcept Estatística_Descritiva))
11.	Individual(Variável type Concepts value (conceptName Variável)
12.	value(isPreRequisiteFor Estatística)

Com as instâncias desta classe (linhas 7 a 12), visa-se criar uma representação padronizada dos assuntos tratados em uma área de conhecimento, para que professores e alunos utilizem os mesmos termos para se referirem a uma mesma entidade, seja ela concreta ou abstrata, evitando as confusões causadas pela falta de estrutura da linguagem natural e pelos termos sinônimos e homônimos. Por exemplo, um sistema não saberia que um LO, cujo assunto é *Estatística Inferencial*, pode ser utilizado em um curso que trata de *Inferência Estatística*, se não for informado (se não estiver definido em algum lugar, como em uma ontologia) que estes dois termos são sinônimos¹⁶.

As instâncias da classe *Concepts* formam a base para a anotação dos componentes utilizados no processo instrucional, como objetivos e objetos de aprendizagem. Por isso, elas são as únicas instâncias que devem ser criadas previamente, para que as pessoas responsáveis pela criação dos cursos utilizem os mesmos termos para definir os conceitos

¹⁶ Não é necessário criar um atributo na classe *Concepts* para indicar quais termos são sinônimos, para isto basta criar outro conceito com o nome desejado e utilizar o predicado *SameAs*, da linguagem OWL, ou então definir como diferentes nomes para uma mesma instância.

abordados. A figura 37 apresenta instâncias criadas para representar conceitos da área de Estatística, além das apresentadas no quadro 35.

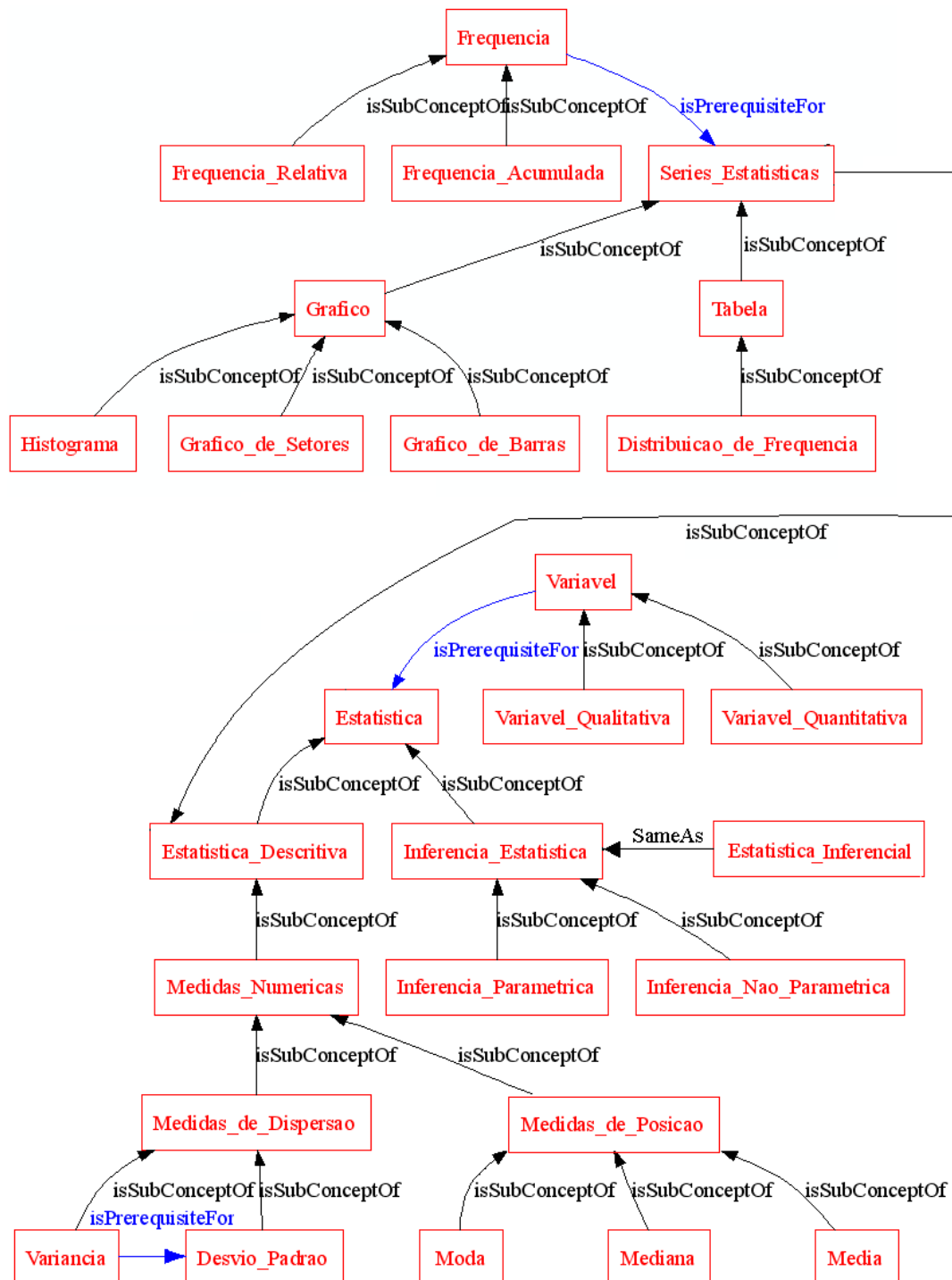


FIGURA 37 - Instâncias da classe *Concepts* para o domínio de Estatística

Considerando que a estrutura de classes da ontologia é o fator mais importante, não foi realizado um estudo aprofundado sobre Estatística para que todos os seus conceitos relevantes fossem incluídos. Cabe ressaltar que é a estrutura das classes que garante o

funcionamento dos sistemas desenvolvidos com base nesta ontologia e permite que mapas conceituais em qualquer domínio do conhecimento sejam criados, utilizados e reutilizados por tais sistemas.

Conforme descrito em 5.2, todas as ontologias foram desenvolvidas utilizando o editor Protégé e salvas no formato OWL para serem, então, utilizadas pelos demais módulos. Por isso, quem desejar utilizar o sistema para geração de cursos em outros domínios do conhecimento, só precisa construir o seu mapa conceitual através do Protégé, ou de qualquer outra ferramenta que possua suporte à linguagem OWL.

Os conceitos definidos na ontologia de domínio são utilizados para definir os assuntos envolvidos nos objetivos de aprendizagem (instâncias da classe *LearningObjectives*) e objetos de aprendizagem (instâncias da classe *LearningObjects*) através do atributo *dc:keyword*, definido pelo *Dublin Core* e incorporado ao padrão LOM.

As instâncias da classe *LearningObjects* representam os menores componentes instrucionais que são agrupados para constituírem os componentes instrucionais mais complexos. Além do atributo *dc:keyword*, os LO possuem vários outros, também definidos no padrão LOM, que descrevem suas características técnicas e pedagógicas.

Estes atributos são utilizados como opções de busca quando se deseja encontrar um LO para utilizá-lo em um curso. Os valores de suas propriedades podem ser comparados às informações constantes nos perfis dos alunos, definidos na ontologia do estudante, que, como já foi explicado, também são baseadas no padrão LOM. O quadro 36 exhibe a declaração das propriedades desta classe, com base na sintaxe abstrata da linguagem OWL.

O significado de cada uma destas propriedades é:

- *dc:title* – título do LO;
- *dc:format* – indica o tipo do LO (texto, imagem, vídeo, programa etc);
- *dcterms:hasPart* – informa que o LO é formado por outros LO;
- *dcterms:isPartOf* – determina que o LO faz parte de outro LO;
- *dcterms:references* – indica que o LO faz referência a outro LO;
- *dcterms:isReferencedBy* – indica que LO é referenciado por outro LO;
- *lom-edu:typicalAgeRange* – determina a faixa etária para a qual o LO é adequado;
- *lom-edu:interactivityType* – define se o LO é expositivo, interativo ou misto;

QUADRO 36 - Propriedades da classe *LearningObjects*

1.	DatatypeProperty(dc:title domain(LearningObjects) range(string)
2.	Functional)
3.	ObjectProperty(dc:keyword domain(LearningObjects) range(Concepts))
4.	ObjectProperty(dc:format domain(LearningObjects) range(dcterms:IMT))
5.	DatatypeProperty(dcterms:extent domain(LearningObjects)
6.	range(integer))
7.	ObjectProperty(dcterms:isPartOf domain(LearningObjects)
8.	range (LearningObjects) Inverse (dcterms:hasPart))
9.	ObjectProperty(dcterms:hasPart domain(LearningObjects)
10.	range (LearningObjects) Inverse (dcterms:isPartOf))
11.	ObjectProperty(dcterms:isReferencedBy domain(LearningObjects)
12.	range (LearningObjects) Inverse (dcterms:references))
13.	ObjectProperty(dcterms:references domain(LearningObjects)
14.	range (LearningObjects) Inverse (dcterms:isReferencedBy))
15.	ObjectProperty(lom-edu:difficulty domain(LearningObjects)
16.	range (lom-edu:Difficulty) Functional))
17.	ObjectProperty(lom-edu:interactivityLevel domain(LearningObjects)
18.	range (lom-edu:InteractivityLevel) Functional)
19.	ObjectProperty(dc:language domain(LearningObjects)
20.	range (dcterms:ISO639-2))
21.	DatatypeProperty(lom-edu:typicalAgeRange domain(LearningObjects)
22.	range (string))
23.	ObjectProperty(lom-edu:interactivityType domain(LearningObjects)
24.	range (lom-edu:InteractivityType) Functional)
25.	ObjectProperty(lom-edu:learningResourceType domain(LearningObjects)
26.	range (lom-edu:LearningResourceType) MinCardinality(1))
27.	ObjectProperty(lom-edu:interactivityType domain(LearningObjects)
28.	range (lom-edu:InteractivityType) Functional)
29.	DatatypeProperty(lom-edu:typicalLearningTime domain(LearningObjects)
30.	range (string) Functional)
31.	ObjectProperty(lom-gen:aggregationLevel domain(LearningObjects)
32.	range (lom-gen:AggregationLevel) Functional)
33.	ObjectProperty(lom-gen:structure domain(LearningObjects)
34.	range (lom-gen:StructureType) Functional)
35.	DatatypeProperty(lom-tech:location domain(LearningObjects)
36.	range (string) MinCardinality(1))
37.	ObjectProperty(lom-edu:semanticDensity domain(LearningObjects)
38.	range(lom-edu:SemanticDensity) Functional)

- *lom-edu:interactivityLevel* – define o nível de interatividade do LO. Seu valor deve ser compatível com o definido em *lom-edu:interactivityType*;
- *lom-edu:learningResourceType* – determina o tipo do LO (exercício, texto, simulação, exame, diagrama, experimento etc.);
- *lom-edu:typicalLearningTime* – estabelece o tempo médio para que os alunos cumpram as atividades;
- *lom-edu:difficulty* – nível de dificuldade do LO;
- *dc:language* – idioma do LO;
- *lom-tech:location* – armazena a localização física do objeto de aprendizagem;
- *dcterms:extent* – tamanho do arquivo em *bytes*;
- *lom-edu:semanticDensity* – densidade ou condensação das informações apresentadas pelo LO. Não se deve confundir-lo com dificuldade;
- *lom-gen:structure* - Estrutura do LO. Seus valores podem indicar que o LO é formado por um único arquivo (*atomic*) ou por um conjunto de outros objetos, que podem ser estruturados de diferentes maneiras (*linear*, *collection*, *networked* e *hierarchical*);
- *lom-gen:aggregationLevel* – Nível de agregação do LO. Está relacionado ao atributo *structure*.

O atributo *lom-tech:location* é fundamental para os propósitos do modelo desenvolvido, pois armazena a localização física do LO. Este atributo possibilita que um LO selecionado possa ser localizado na *Web* e uma cópia sua feita para ser incorporada ao novo curso. Os metadados sobre os objetivos, atividades e objetos de aprendizagem permanecem registrados na ontologia para que qualquer usuário realize buscas a fim de reutilizá-los. Entretanto, os LO propriamente ditos (arquivos com o conteúdo) não ficam armazenados em um repositório e sim dispersos nos vários cursos que os utilizam. Para tanto, é necessário manter a referência para o seu endereço, através do atributo *lom-tech:location*, para que ele possa ser recuperado e reutilizado em novos cursos. O quadro 37 apresenta a declaração de uma instância da classe *LearningObjects*, que é iniciada com a especificação do nome da instância e da classe a que ela pertence (linha 1).

QUADRO 37 - Instância da classe *LearningObjects*

1.	Individual(LO1 type LearningObjects
2.	value (dc:title Introdução)
3.	value (dc:keyword Estatística)
4.	value (dc:keyword Estatística_Descritiva)
5.	value (dc:keyword Inferencia_Estatística)
6.	value (dc:format lom:html)
7.	value (dcterms:extent 2100)
8.	value (lom-edu:difficulty lom-edu:VeryEasy)
9.	value (dc:language pt-BR)
10.	value (lom-edu:typicalAgeRange 14-)
11.	value (lom-edu:interactivityType lom-edu:Active)
12.	value (lom-edu:interactivityLevel lom-edu:VeryLowInteractivityLevel)
13.	value (lom-edu:learningResourceType lom-edu:narrative_text)
14.	value (lom-edu:typicalLearningTime 7m)
15.	value (lom-gen:aggregationLevel lom-gen:AggregationLevel1)
16.	value (lom-gen:structure Atomic)
17.	value (lom-tech:location http://localhost/estatistica/intro.html)
18.	value (lom-edu:semanticDensity lom-edu:VeryLowDensity))

Nas linhas 2 a 18 estão os valores associados a cada uma das propriedades da instância, cujo significado já foi explicado previamente.

As atividades de aprendizagem, que em geral são formadas por um conjunto de LO, representam as informações exibidas aos usuários para que eles atinjam um objetivo de aprendizagem. Como pode ser visto no quadro 38, onde estão declaradas as propriedades vinculadas à classe *LearningActivities*, cada atividade possui: um título (linhas 1 e 2); os eventos instrucionais que ela proporciona (linhas 3 e 4); o nível de conhecimento que o estudante deve possuir para acompanhar a atividade e o nível de conhecimento que ele alcançará após concluí-la (linhas 5 a 8); objetivos de aprendizagem (linhas 9 e 10); objetos de aprendizagem (linhas 11 e 12) e algumas subatividades de aprendizagem (13 e 14).

A divisão recursiva de atividades em subatividades torna desnecessária a criação de definições rígidas na ontologia, do tipo: um curso divide-se em capítulos que, por sua vez, são divididos em módulos, os quais contém várias lições. Dessa maneira, não é necessário que o conteúdo seja subdividido com um número fixo de níveis, pois o usuário pode construir um único nível de atividades com todo o conteúdo ou subdividi-las quantas vezes considerar necessário. Além disso, diferentes pessoas que criam sistemas baseados nesta ontologia podem associar os termos que julgarem mais adequados às atividades de cada

nível e limitar a quantidade de níveis de acordo com as suas preferências.

QUADRO 38 - Propriedades da classe *LearningObjects*

1.	<code>DatatypeProperty(dc:title domain(LearningActivities) range(string)</code>
2.	<code>Functional)</code>
3.	<code>ObjectProperty(instructionalEvents domain(LearningActivities)</code>
4.	<code>range(pedagogical:InstructionalEvents))</code>
5.	<code>ObjectProperty(knowledgeDesired domain(LearningActivities)</code>
6.	<code>range(pedagogical:CognitiveTaxonomies) Functional)</code>
7.	<code>ObjectProperty(knowledgeLevel domain(LearningActivities)</code>
8.	<code>range(pedagogical:CognitiveTaxonomies) Functional)</code>
9.	<code>ObjectProperty(learningObjectives domain(LearningActivities)</code>
10.	<code>range(learningObjectives) MinCardinality(1))</code>
11.	<code>ObjectProperty(learningObjects domain(LearningActivities)</code>
12.	<code>range(LearningObjects))</code>
13.	<code>ObjectProperty(subLearningActivities domain(LearningActivities)</code>
14.	<code>range(LearningActivities) Transitive)</code>

A propriedade *instructionalEvents*, que designa quais etapas do processo instrucional são cobertas por ela, associa as instâncias da classe *LearningActivities* a instâncias de uma das subclasses de *InstructionalEvents* definida na ontologia pedagógica. Como já foi explicado anteriormente, é possível detectar inconsistências na estruturação das atividades, verificando se, em conjunto, elas cumprem todas as etapas previstas no processo instrucional.

A associação das atividades de aprendizagem a um ou mais objetivos de aprendizagem permite que o usuário, ao reutilizar objetivos de aprendizagem de outros cursos, recupere as atividades de aprendizagem vinculadas a ele no primeiro curso, para reutilizá-las também.

O quadro 39 apresenta um exemplo de uma atividade de aprendizagem, cujo título é *Medidas Numéricas* (linha 2), que obtém a atenção e motiva os alunos (*Gaining_Attention*) e apresenta os objetivos das próximas atividades (*Informing_the_Objective*).

Para acompanhar esta atividade, o aluno já deve ter um conhecimento sobre o conceito de medidas numéricas (linha 5) e, ao concluí-la, deverá alcançar o nível de compreensão (linha 6) dos conhecimentos abordados. Os objetivos de aprendizagem (linhas 7 e 8) e os objetos que compõem a atividade (linhas 9 e 10) e subatividades (linha 11) são indicados apenas através do nome das instâncias correspondentes. Os valores para cada uma das propriedades destas instâncias é definido no momento de sua declaração,

como exemplificado no quadro 37, para objetos de aprendizagem, 41, para objetivos de aprendizagem e no próprio quadro 39, para as subatividades.

QUADRO 39 - Instância da classe *LearningActivities*

1.	Individual(LACurso11 type LearningActivities
2.	value(dc:title Medidas Numericas)
3.	value(instructionalEvents pedagogical:Gaining_Attention)
3.	value(instructionalEvents pedagogical:Informing_the_Objective)
5.	value(knowledgeDesired pedagogical:Knowledge)
6.	value(knowledgeLevel pedagogical:Comprehension)
7.	value(learningObjectives OBCurso11)
8.	value(learningObjectives OBCurso12)
9.	value(learningObjectes LOCurso11)
10.	value(learningObjectes LOCurso12)
11.	value(subLearningActivities LA Curso12))

Os objetivos de aprendizagem possuem um título (quadro 40, linha 1), que é uma descrição em linguagem natural utilizada para resumir suas características, e informações que visam facilitar a sua reutilização em diferentes cursos. Entre elas estão os conceitos que o aluno deve aprender ao atingir o objetivo (instâncias da classe *Concepts*, linhas 7 e 8), o nível de conhecimento que ele alcançará sobre estes conceitos (linhas 9 e 10) e o nível de conhecimento que o aluno já deve possuir sobre o conceito para ter condições de acompanhar as atividades que o levam a atingir o objetivo (linhas 11 e 12).

QUADRO 40 - Propriedades da classe *LearningObjectives*

1.	DatatypeProperty(dc:title domain(LearningObjectives) range(string)
2.	Functional)
3.	DatatypeProperty(result domain(LearningObjectives) range(string)
4.	Functional)
5.	DatatypeProperty(situation domain(LearningObjectives) range(string)
6.	Functional)
7.	ObjectProperty(dc:keywords domain(LearningObjectives)
8.	range(Concepts) MinCardinality(1))
9.	ObjectProperty(typeOfLearningObjective domain(LearningObjectives)
10.	range(pedagogical:CognitiveTaxonomies) Functional)
11.	ObjectProperty(typeOfPreRequisite domain(LearningObjectives)
12.	range(pedagogical:CognitiveTaxonomies) Functional)
13.	ObjectProperty(ActionVerb domain(LearningObjectives)
14.	range(pedagogical:Verbs))

Os níveis de conhecimento que o aluno deve possuir e o que ele alcançará são definidos pelas instâncias das subclasses de *CognitiveTaxonomies* na ontologia pedagógica. Como estas mesmas instâncias são utilizadas na ontologia do estudante para definir o nível de conhecimento inicial e o nível que se deseja alcançar em cada conceito do curso para cada perfil de aluno, é fácil encontrar objetivos e atividades de aprendizagem que possam ser reutilizadas nos novos cursos.

Além disso, com o objetivo de facilitar a tarefa de encontrar as atividades de aprendizagem que são adequadas para levar os alunos a atingir um determinado objetivo, foram inseridas na classe *LearningActivities* as propriedades *knowledgeLevel* e *knowledgeDesired*, cujos valores também são definidos através das subclasses de *CognitiveTaxonomies*. Assim, sabe-se o nível de conhecimento que uma atividade pode proporcionar, para verificar se ele é compatível com os objetivos do curso que, por sua vez, também devem ser compatíveis com as expectativas dos alunos.

Além das propriedades já citadas, os objetivos de aprendizagem possuem outras duas, cujas informações são textos desestruturados e têm como função tornar mais claro o seu contexto de aplicação. São elas:

- *Situation* (linhas 3 e 4) – Indica a situação onde o objetivo pode ser aplicado. Por exemplo: Dado um conjunto de vários números inteiros;
- *Result* (linhas 5 e 6) – Estabelece o que se espera que o aluno consiga fazer com os dados definidos na propriedade *Situation*. Por exemplo: Calcular a média entre eles.

A última propriedade que um objetivo de aprendizagem possui o associa a uma das instâncias das subclasses de *Verbs* (linhas 13 e 14), descritas na ontologia pedagógica. Esse verbo está relacionado a uma das categorias que determinam o nível das operações mentais que os alunos devem ser capazes de realizar após atingir o objetivo. Ele pode ser comparado a duas outras informações: a categoria atribuída ao objetivo e o resultado esperado (*Result*) para verificar se não existem inconsistências.

Por exemplo, o verbo *Descrever*, que determina o tipo de resultado esperado (propriedade *Result*), indica que o nível de raciocínio que o aluno deve alcançar neste objetivo está na segunda categoria da taxionomia de Bloom (*Comprehension*). Caso seja indicado que o objetivo pertence a outra categoria ou seja associado a um verbo de outra categoria, é possível detectar a inconsistência para impedir que o usuário cometa este equívoco ou apenas para alertá-lo, de acordo com a preferência de quem desenvolver o

programa que utiliza as ontologias.

No quadro 41 é declarada uma instância da classe *LearningObjectives*, com valores para as propriedades descritas acima.

QUADRO 41 - Instâncias da classe *LearningObjectives*

1.	Individual(OBCurso11 type LearningObjectives
2.	value(dc:title Capacidade para resumir os objetivos da estatística)
3.	value(result Descrever de forma sucinta os objetivos da estatística)
3.	value(situation Mediante a questão: Para que serve a estatística?)
5.	value(dc:keywords Estatística)
6.	value(dc:keywords Inferencia_Estatística)
7.	value(dc:keywords Estatística_Descritiva)
8.	value(typeOfLearningObjective pedagogical:Comprehension)
9.	value(typeOfPreRequisite pedagogical:Nenhum)
10.	value(ActionVerb pedagogical:Descrever))

Conforme visto, uma atividade de aprendizagem é composta por subatividades e objetos de aprendizagem, além de estar associada a um ou mais objetivos de aprendizagem. Estas atividades são unidas e dispostas hierarquicamente para formar as diferentes estruturas de organização e apresentação do conteúdo que um curso pode ter. Um conjunto de atividades de aprendizagem é representado na ontologia como instância da classe *Organizations*, a qual possui as propriedades apresentadas no quadro 42.

QUADRO 42 - Propriedades da classe *Organizations*

1.	DatatypeProperty(dc:title domain(Organizations) range(string)
2.	Functional)
3.	ObjectProperty(instructionalStrategy domain(Organizations)
4.	range(pedagogical:InstructionalStrategies) Functional))
5.	ObjectProperty(learningActivities domain(Organizations)
6.	range(LearningActivities) MinCardinality(1))
7.	ObjectProperty(learningObjectives domain(Organizations)
8.	range(learningObjectives) MinCardinality(1))
9.	ObjectProperty(student domain(Organizations)
10.	range(students:Students) Functional)

Uma instância da classe *Organization* também possui um título (linhas 1 e 2), que define suas características principais, e uma estratégia instrucional (linhas 3 e 4), cujos valores possíveis são definidos pelas instâncias das subclasses de *InstructionalStrategies*, na ontologia pedagógica. Conforme descrito na apresentação da referida ontologia, as

instâncias das subclasses de *InstructionalStrategies*, além do nome, possuem apenas uma descrição textual sobre a estratégia instrucional. Assim, o desenvolvedor do sistema que faz uso da ontologia pode decidir como implementar as diferentes estratégias instrucionais. No capítulo 6, durante a apresentação do sistema desenvolvido com base nestas ontologias, será apresentado como a estratégia instrucional influencia na organização das atividades dos cursos gerados.

Além disso, uma instância da classe *Organization* está associada a atividades de aprendizagem (linhas 5 e 6), objetivos de aprendizagem (linhas 7 e 8) e referenciam também um perfil de aluno definido na ontologia do estudante (linhas 9 e 10), que indica o tipo de aluno para o qual aquela estrutura do curso foi construída. Um curso deve possuir pelo menos uma instância desta classe, mas, neste caso, não é oferecida a possibilidade de adaptação do conteúdo a diferentes tipos de alunos. Quando existem duas ou mais instâncias desta classe, as atividades de aprendizagem podem ser reaproveitadas entre elas, mudando apenas a sua ordem de apresentação ou pode-se selecionar atividades completamente distintas.

O quadro 43 descreve uma instância da classe *Organizations*, com definições para todas as suas propriedades. Tanto as atividades de aprendizagem (linhas 4 a 6) quanto os objetivos (linhas 7 e 8) e o perfil dos alunos (linha 9) são referenciados apenas pelo nome da instância que contém todas as definições sobre eles.

QUADRO 43 - Instâncias da classe *Organizations*

1.	<code>Individual(Org1 type Organizations</code>
2.	<code>value(dc:title Introducao a Estatistica (Basico - Domestico))</code>
3.	<code>value(instructionalStrategy pedagogical:Direct_Instruction)</code>
4.	<code>value(learningActivities LACurso11)</code>
5.	<code>value(learningActivities LACurso12)</code>
6.	<code>value(learningActivities LACurso13)</code>
7.	<code>value(learningObjectives OBCurso11)</code>
8.	<code>value(learningObjectives OBCurso12</code>
9.	<code>value(student Est_Bas_Dom))</code>

Finalmente, chega-se à estrutura mais abrangente da ontologia de domínio, a classe *Course*, a qual representa o curso que está sendo criado como um todo. Seu papel é apenas registrar o título, a URL (endereço onde ele será disponibilizado) e agrupar a(s) instância(s) da classe *organization* que o compõem, como pode ser visto nos quadros 44 e 45, que exibem as propriedades e o exemplo de uma instância da classe *Course*.

QUADRO 44 - Propriedades da classe *Courses*

- | | |
|----|--|
| 1. | DatatypeProperty(dc:title domain(Courses) range(string) |
| 2. | Functional) |
| 3. | DatatypeProperty(lom-tech:location domain(Courses) range(string) |
| 4. | Functional) |
| 5. | ObjectProperty(organizations domain(Courses) |
| 6. | range(Organizations) MinCardinality(1)) |

QUADRO 45 - Instâncias da classe *Courses*

- | | |
|----|---|
| 1. | Individual(Org1 type Courses |
| 2. | value(dc:title Introducao a Estatistica) |
| 3. | value(lom-tech:location http://uniplac.net/estatistica) |
| 4. | value(organizations Org1) |
| 5. | value(organizations Org2)) |

A URL onde o curso será disponibilizado permanece registrada na ontologia para que, posteriormente, outras pessoas possam reutilizar seus LO. Através desse valor, é definido o atributo *lom-tech:location* dos LO utilizados no curso, possibilitando a recuperação e reutilização dos mesmos.

Finalizando a descrição das ontologias desenvolvidas, vale ressaltar o cuidado tomado no seu projeto para facilitar a sua reutilização e expansão, atendendo a um dos requisitos listados por GRUBER (1993b) para o projeto de uma boa ontologia, denominado Compromisso Ontológico Mínimo. Esse requisito estabelece que uma ontologia deve definir apenas os termos extremamente necessários para que as informações possam ser compartilhadas. Os demais devem ser especializados por cada um dos interessados.

5.4 Transformação dos Dados da Ontologia em um Curso

As ontologias expostas neste capítulo se constituem em um importante meio para tornar possível a reutilização de componentes instrucionais em diferentes cursos e a orientação do usuário durante a sua criação, a partir das teorias da aprendizagem. Entretanto, é preciso representar as informações nelas dispostas de uma maneira que algum LMS seja capaz de interpretá-las e tornar o curso disponível aos alunos.

A primeira questão analisada em relação ao formato do curso gerado diz respeito à criação de um padrão próprio para ele ou a utilização de um dos padrões de metadados educacionais existentes. Analisando os prós e contras de cada alternativa, podem ser

destacados os seguintes argumentos a favor da criação de um padrão próprio:

- Podem ser inseridas no padrão todas as características desejadas, com base nas ontologias desenvolvidas, tornando mais simples a conversão das informações nela definidas para um curso que possa ser disponibilizado na *Web*;
- A criação de um padrão próprio implica no desenvolvimento de um sistema para disponibilizar os cursos compatíveis com ele. Como o padrão para representar os cursos e o sistema para disponibilizá-los aos alunos seriam criados pela mesma pessoa, muitas especificidades poderiam ser incorporadas ao sistema, o que tornaria mais fácil a construção dos mecanismos para apresentação adaptativa;

Entretanto, algumas desvantagens da criação de um novo padrão são eminentes, tais como:

- Este padrão pode se tornar tão específico que apenas o seu criador tenha interesse em utilizá-lo;
- É necessário um grande esforço de programação para desenvolver o sistema para disponibilizar os cursos no padrão criado, porque não seriam utilizados LMS já prontos que possuem todas as funcionalidades básicas já implementadas;
- Os cursos gerados precisariam de um sistema de disponibilização próprio para serem executados, o que limitaria a possibilidade de execução dos mesmos, dificultando a sua aceitação por pessoas interessadas nas ontologias e no sistema desenvolvido para gerar cursos para a EBW.

Assim, optou-se por utilizar um padrão já consolidado para representar os cursos gerados a partir das informações das ontologias, com a intenção de que eles possam ser acessados pelo maior número de pessoas possível. Devido às vantagens já apresentadas na seção 4.6.3, optou-se pela utilização do Modelo de Agregação de Conteúdo (CAM) definido no padrão SCORM, que estabelece como as informações devem estar dispostas para constituir um curso. O quadro 26 descreve as relações existentes entre as informações da ontologia de domínio e os elementos do CAM.

QUADRO 46 - Relação entre os elementos da ontologia de domínio e do padrão SCORM

Classe da ontologia de domínio	Significado	Elemento do padrão SCORM
<i>Courses</i>	Cada instância desta classe engloba todas as informações de um curso	Arquivo <i>imsmanifest.xml</i> e os arquivos dos LO.
<i>Organizations</i>	Cada instância representa uma estrutura diferente para organizar e apresentar as atividades de aprendizagem e está associada a um perfil de aluno diferente.	< <i>organization</i> >
<i>LearningObjectives</i>	Cada instância desta classe possui as informações sobre um objetivo de aprendizagem que os alunos precisam alcançar.	< <i>learning_objective</i> > * *O CAM não define este atributo, mas mesmo assim ele foi adicionado, como é explicado no capítulo 6.
<i>LearningActivities</i>	Cada instância desta classe representa uma atividade de aprendizagem, que pode ser tanto uma atividade principal quanto uma subatividade.	< <i>item</i> >
<i>LearningObjects</i>	Cada instância desta classe representa um objeto de aprendizagem, com seus metadados e uma referência para o arquivo com o conteúdo.	< <i>file</i> > e < <i>resource</i> > O primeiro elemento representa um LO isoladamente. O segundo, engloba os vários LO utilizados em uma mesma atividade de aprendizagem.
<i>Concepts</i>	Cada instância desta classe representa um conceito de um domínio do conhecimento.	Ela não possui relação com o padrão SCORM, pois seus valores não são componentes de um curso. Eles servem para indicar o assunto dos componentes previamente descritos.

A partir destas relações, foi construído um sistema que possibilita ao usuário manipular as informações dispostas nas ontologias e gerar um curso de acordo com o CAM. Para simplificar a leitura do próximo capítulo, as informações são referidas com base em seu significado e não através do nome de suas classes. Por exemplo, para tratar de uma estrutura de organização e apresentação do conteúdo armazenada na ontologia, utiliza-se apenas o termo estrutura ou estrutura de apresentação e não uma instância da classe *organization*. Nomes de classes, instâncias e propriedades são utilizados apenas quando se fazem estritamente necessários.

5.5 Considerações

Este capítulo apresentou um resumo dos problemas relativos à EBW, identificados a partir do levantamento bibliográfico, e descreveu as ontologias desenvolvidas com o intuito de formalizar o conhecimento necessário para resolvê-los. Além disso, foram estabelecidos os requisitos que devem ser satisfeitos pela arquitetura desenvolvida, para que a mesma possa ser considerada válida e capaz de melhorar o processo de construção de cursos para a EBW.

Alguns problemas foram encontrados nesta etapa, como a falta de resultados consolidados sobre certas tecnologias da *Web Semântica* e dificuldades para representar computacionalmente as informações do processo instrucional. Mesmo assim, foi possível formular os princípios básicos que possibilitam minimizar parte dos problemas listados, principalmente no que tange à reutilização de materiais e ao suporte fundamentado nas teorias instrucionais durante a estruturação das atividades e elaboração do curso como um todo.

6. ATLAS - SISTEMA BASEADO NA *WEB* SEMÂNTICA PARA GERAÇÃO DE CURSOS

Neste capítulo é apresentado o sistema desenvolvido com o objetivo de gerar cursos para a EBW, denominado Atlas. Seus principais módulos são descritos e a construção de um curso exemplo é exposta, para que os resultados sejam discutidos e validados através da comparação com os requisitos da arquitetura e pela sua execução em um LMS com suporte ao padrão escolhido.

6.1 Caracterização do Sistema

O sistema Atlas é uma ferramenta baseada na *Web* que tem como objetivo facilitar o processo de construção de cursos para a EBW. Sua principal característica é fazer uso dos recursos disponibilizados pela *Web* Semântica, como as ontologias definidas no capítulo 5 e conteúdos educacionais descritos através de metadados, para tornar possível a reutilização dos componentes utilizados na construção destes cursos. Além disto, ele visa auxiliar na organização e estruturação das atividades que compõem os cursos, fundamentando-se, para isso, nas teorias instrucionais e da aprendizagem. O curso resultante é compatível com o CAM definido no padrão SCORM.

A figura 38 apresenta um diagrama de navegação que fornece uma visão geral sobre a seqüência de passos realizados durante a criação de um curso. Neste diagrama, não estão representadas as interfaces do sistema individualmente, mas sim os conjuntos de interfaces que constituem os módulos do sistema, os quais serão explicados separadamente na seção 6.2.

6.2 Módulos do Sistema

Nesta seção são descritas as funcionalidades das principais interfaces de cada um dos módulos presentes na figura 38.

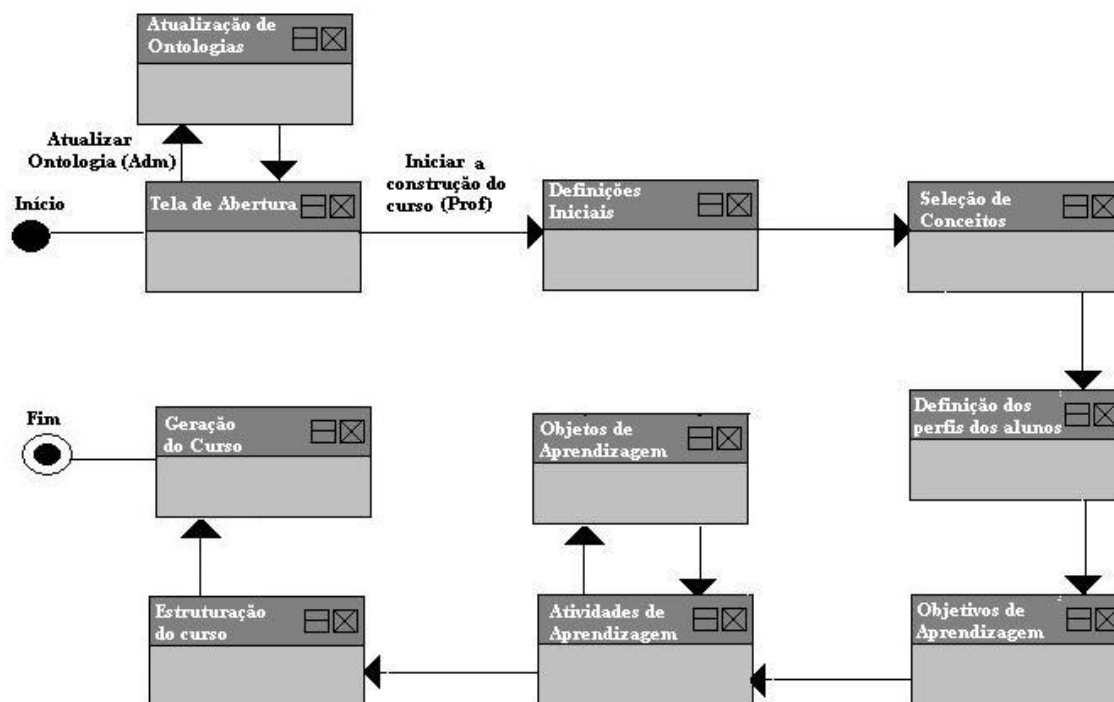


FIGURA 38 - Diagrama de navegação do sistema Atlas

6.2.1 O Módulo do Administrador

Quatro tipos de usuários diferentes são previstos na figura 27, que descreve a arquitetura desenvolvida. Entretanto, apenas dois deles, o criador do curso e o administrador, interagem diretamente com o Atlas, já que os alunos acessam os cursos gerados através de LMS. A figura 39 apresenta a interface de abertura do sistema, que possui dois *links*, um que leva às interfaces do usuário *Administrador* e, outro, do *Professor* que, neste caso, representa qualquer pessoa que utilize o Atlas para criar um curso.

Para que pessoas compartilhem e reutilizem informações em qualquer domínio do conhecimento, é primordial que utilizem terminologias padrões para descrevê-las, as quais podem ser definidas através de ontologias.

O sistema desenvolvido não possibilita a construção de ontologias, mas é capaz de ler ontologias já prontas que descrevem os conceitos de um domínio do conhecimento e suas relações (ontologia de domínio - figura 37), criadas em qualquer editor que suporte a linguagem OWL. Além da ontologia de domínio, o sistema também pode ler as outras três ontologias construídas e descritas no capítulo 5, que representam as informações pedagógicas, informações sobre os alunos e do padrão LOM.

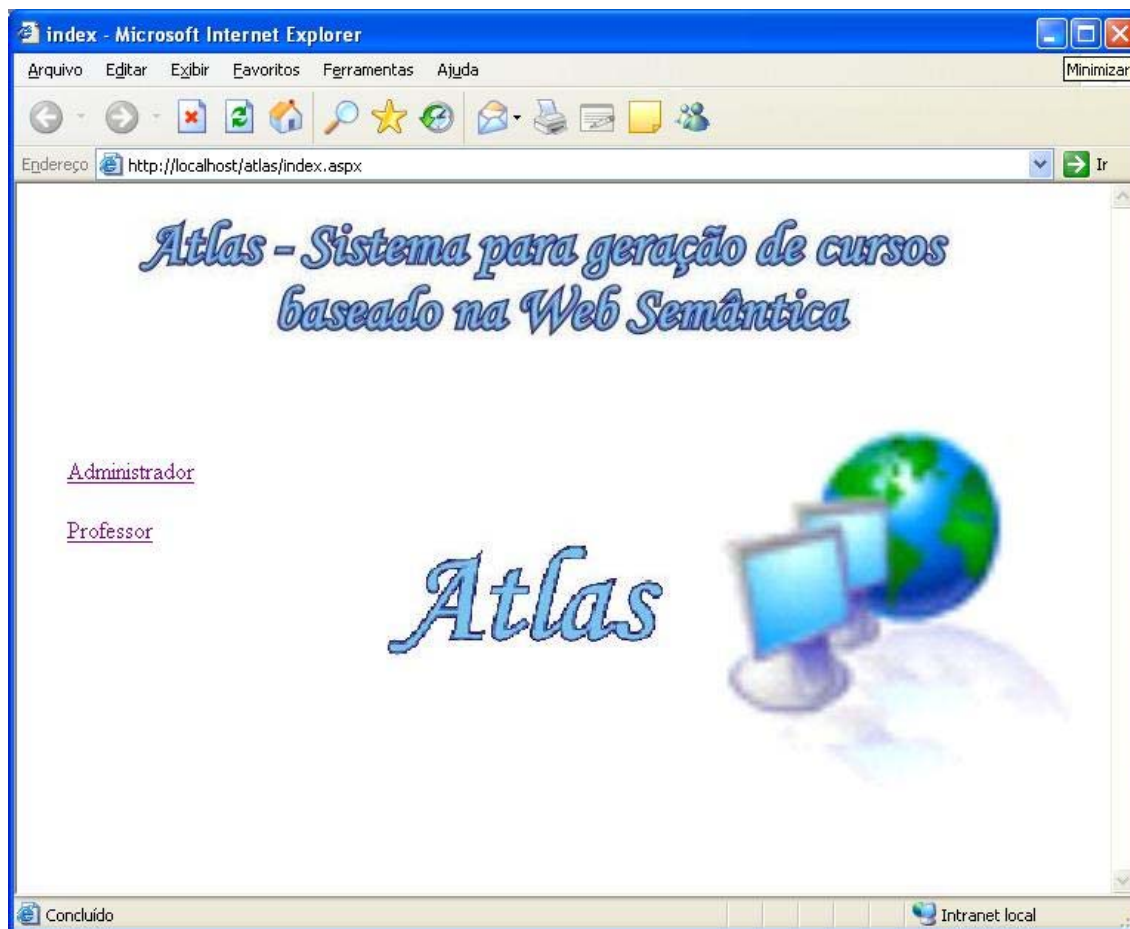


FIGURA 39 - Interface inicial do sistema

A função do administrador do Atlas em termos computacionais é apenas importar estas ontologias na primeira vez em que o sistema for executado ou quando elas forem alteradas, para que as novas informações sejam incorporadas ao sistema. Contudo, sabe-se que durante a construção e atualização das ontologias, o papel do administrador é muito mais abrangente, sendo importante a presença de uma equipe interdisciplinar capaz de definir de maneira consistente as suas informações.

A figura 40 ilustra a interface onde o administrador indica qual das ontologias está sendo atualizada e define o endereço do arquivo OWL, que a contém. Essa é a única interface que o administrador utiliza no sistema. Caso ele deseje construir um curso deve acessar o *link* do *Professor* para ter acesso a todas as outras funcionalidades do sistema, descritas nas próximas seções.

6.2.2 O Módulo para Iniciar a Construção de um Curso

O *link Professor* leva às interfaces para construção do curso que utilizam os dados das ontologias importadas pelo administrador. O primeiro passo consiste em definir o título

do curso e a URL onde ele será disponibilizado, através da interface exibida na figura 41.

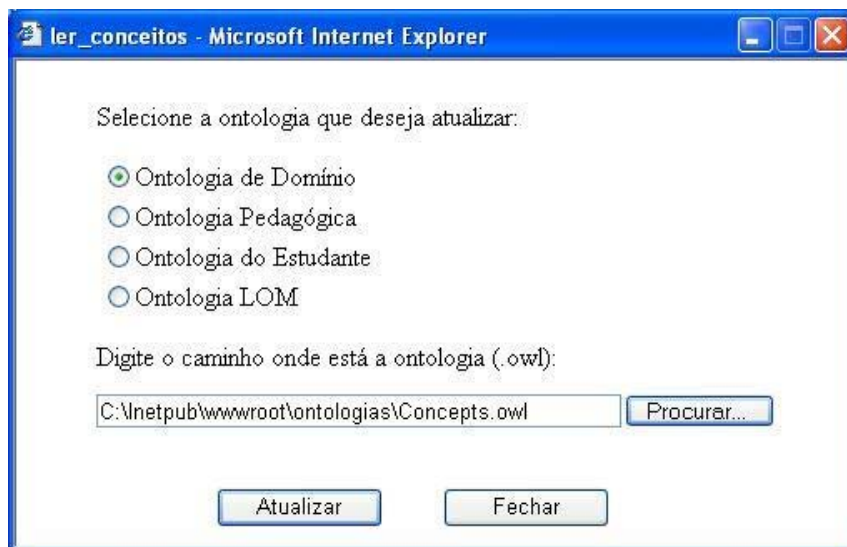


FIGURA 40 - Interface para importação da ontologia

O endereço onde o curso será disponibilizado deve ser corretamente indicado, pois é por meio dele que o sistema registra na ontologia de domínio a localização física dos LO utilizados, através do atributo *lom-tech:location* da classe *LearningObjects*, para que os mesmos possam ser reutilizados posteriormente.



FIGURA 41 - Identificação do curso

6.2.3 O Módulo para Seleção de Conceitos

Após registrar as informações iniciais, o sistema apresenta ao usuário os conceitos sobre o assunto de interesse, importados da ontologia de domínio. Nesse momento, o usuário indica os conceitos que devem ser abordados no curso que está criando. Tais conceitos são utilizados, tanto para buscar componentes instrucionais já prontos que possam ser reutilizados quanto para definir os assuntos dos novos componentes construídos, para que eles também possam ser reutilizados posteriormente.

Como estes conceitos formam a base para anotar os demais componentes instrucionais definidos na ontologia de domínio, é fundamental que eles estejam bem definidos e representem o consenso entre as pessoas que utilizam o sistema para construir os cursos. A figura 42 apresenta a interface que possibilita a execução dessa tarefa.

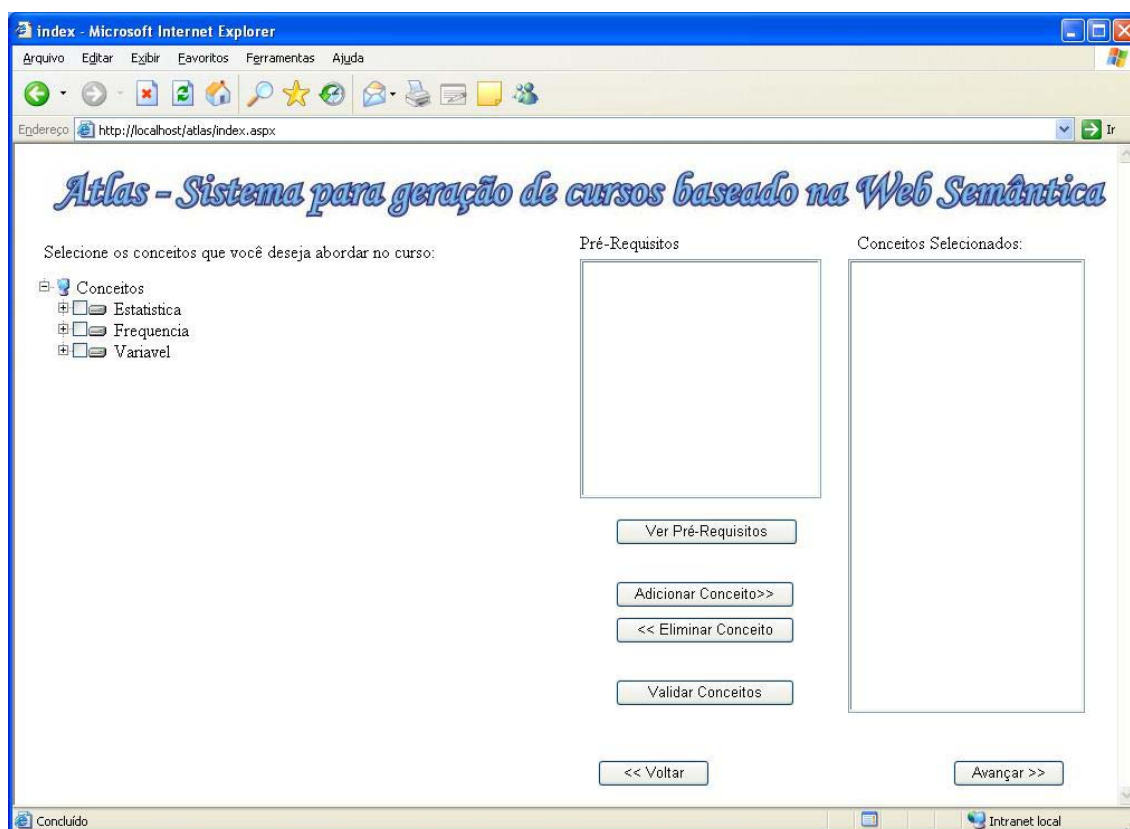


FIGURA 42 - Interface para seleção de conceitos (1)

À esquerda da janela, encontra-se o componente responsável por listar os conceitos existentes na ontologia de domínio. Inicialmente, foram realizados testes visando criar um grafo para apresentar os conceitos e as suas relações, semelhante ao apresentado na figura 37. Contudo, devido às limitações existentes na construção de interfaces para sistemas *Web*, optou-se por utilizar uma representação em árvore.

Cada nó da árvore representa um conceito e os nós filhos representam seus subconceitos. Em um primeiro momento, são apresentados apenas os nós que não são subconceitos de nenhum outro (mais abrangentes), os quais podem ser expandidos para visualizar os demais conceitos.

Considerando a impossibilidade de representar também os relacionamentos do tipo *Pré-requisito* na árvore, criou-se uma função que os apresenta em uma lista. Basta que o usuário indique quais conceitos ele deseja conhecer os pré-requisitos e pressionar o botão *Ver Pré-Requisitos*, para que eles sejam exibidos na lista disposta no centro da janela.

À direita, encontra-se a lista que armazena os conceitos que o usuário selecionou para abordar em seu curso, como pode ser visto na figura 43.

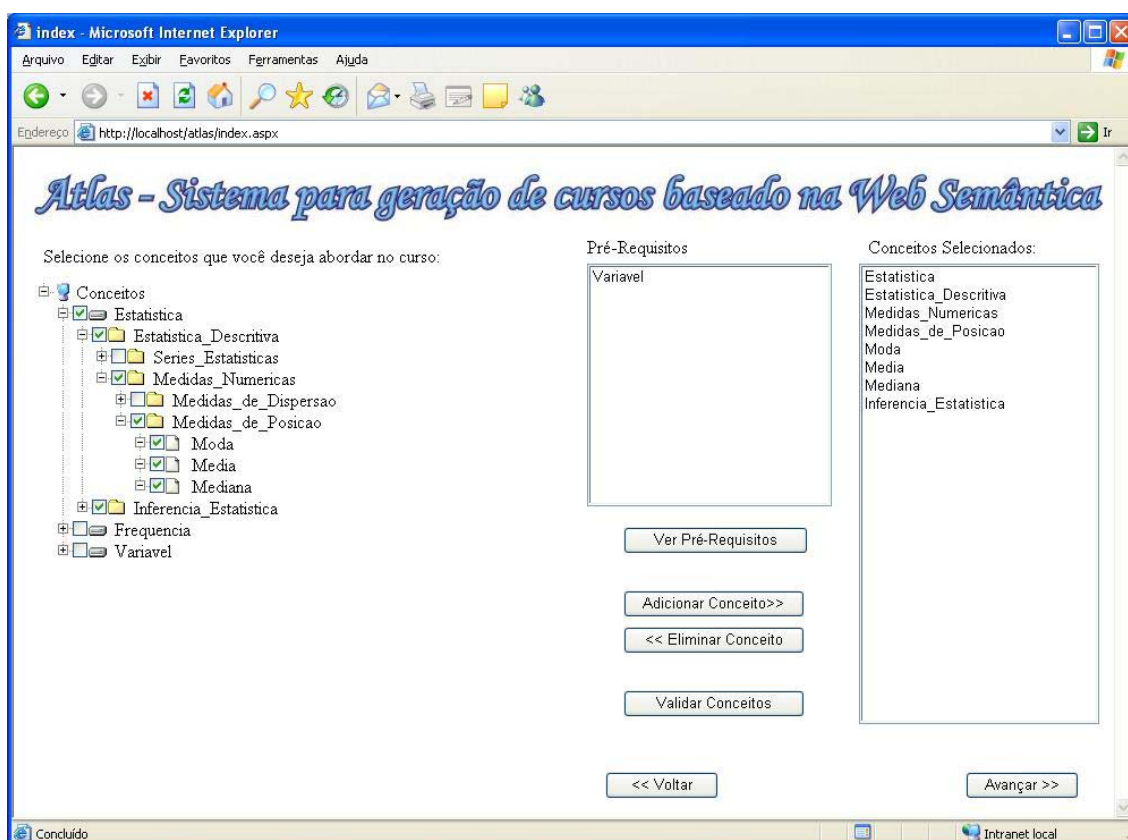


FIGURA 43 - Interface para seleção de conceitos (2)

Como não foi criada nenhuma restrição na ontologia com relação aos conceitos selecionados para um curso, é possível, por exemplo, incluir conceitos mesmo que seus pré-requisitos não sejam adicionados. Por isso, fica a critério do desenvolvedor do sistema que utiliza a ontologia definir o que fazer quando alguns relacionamentos previstos não são mantidos nos conceitos selecionados para o curso. Especificamente no Atlas, optou-se apenas por avisar ao usuário sobre esta situação, para que ele possa incluir os conceitos que

por ventura tenha esquecido ou ignorar o aviso e seguir adiante.

A figura 44 apresenta o resultado da validação dos conceitos listados na figura 43.

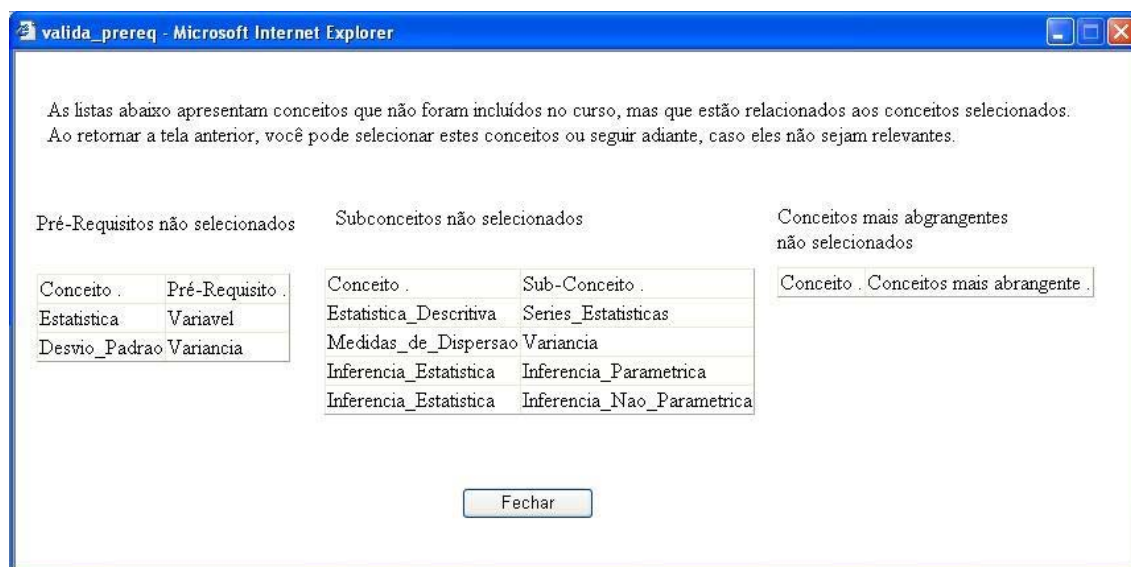


FIGURA 44 - Interface para validação de conceitos

Nesta interface, é possível verificar que o Atlas utiliza os relacionamentos na ontologia para verificar três situações:

- Conceitos que são pré-requisitos e não foram selecionados;
- Quando os subconceitos de um conceito não foram selecionados;
- Quando o conceito que engloba um subconceito não foi selecionado.

No exemplo, foram incluídos os conceitos *Estatística* e *Desvio Padrão*, cujos pré-requisitos, *Variável* e *Variância*, respectivamente, não foram selecionados. Outro problema apontado pelo sistema é que alguns subconceitos de *Estatística Descritiva*, *Medidas de Dispersão* e *Inferência Estatística* também não foram selecionados. Neste caso, nenhum subconceito foi incluído no curso sem que o seu conceito mais abrangente também o fosse. Nesse momento, o usuário pode retornar à interface de seleção de conceitos (figura 43) para incluir os que faltam ou seguir apenas com os conceitos já selecionados.

6.2.4 O módulo para Definição dos Perfis dos Alunos

Após selecionar os conceitos, o usuário é direcionado para as interfaces que lhe possibilitam definir o perfil dos alunos que participarão do curso, tanto em sua dimensão técnica quanto pedagógica. As interfaces para realizar esta tarefa são semelhantes. Ambas possuem uma lista na parte esquerda com os perfis já utilizados em outros cursos,

armazenados na ontologia do estudante, os quais podem ser adicionados na lista da direita, que registra os diferentes perfis dos alunos do curso que está sendo criado.

A figura 45 mostra a interface para seleção de perfis pedagógicos dos alunos. Nela, pode-se perceber que apenas dois tipos de perfis foram utilizados até o momento nos cursos criados, que são denominados de *Básico* e *Avançado*

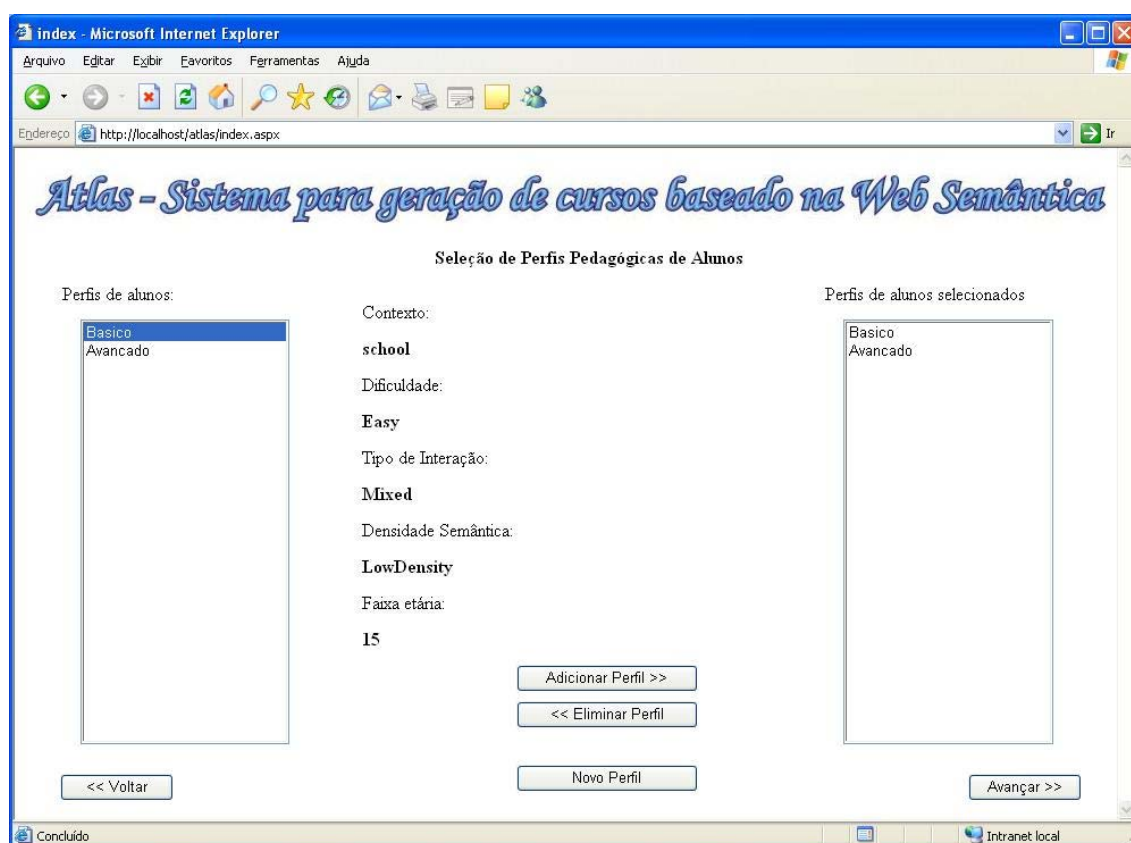



FIGURA 45 - Interface para seleção de perfis pedagógicos

Ao selecionar um dos perfis existentes, são apresentados os valores dos atributos definidos na ontologia do estudante para o usuário na parte central desta figura. Assim, o usuário pode identificar se os alunos do curso que está criando se enquadram naquele perfil ou se é preciso procurar outros ou, ainda, criar um novo. Para criar um perfil com características diferentes dos que já existem, o usuário utiliza a interface apresentada na figura 46. Nela, ele define o título do perfil e os valores para cada uma de suas propriedades. Ao salvar as informações, uma nova instância da classe *Pedagogical* é criada na ontologia do estudante.

Através de interfaces semelhantes às apresentadas, o usuário pode selecionar ou criar os perfis técnicos a serem utilizados no curso. Quando um novo perfil técnico é definido, o usuário deve informar os valores das propriedades da classe *Technical*, que são associados

à nova instância criada na ontologia do estudante.

De acordo com a estrutura da referida ontologia, as características sobre os alunos que freqüentam os cursos são definidas através de estereótipos nas dimensões técnicas e pedagógicas, o que foi implementado no sistema Atlas através das interfaces previamente apresentadas.



The image shows a web browser window titled "cria_perfisalunos2 - Microsoft Internet Exp". The main content area is titled "Cadastro de perfis pedagógicos de alunos:". It contains several form fields and buttons:

- Nome do Perfil:** A text input field containing the word "Avancado".
- Contexto:** A dropdown menu with "school" selected.
- Dificuldade:** A dropdown menu with "Difficult" selected.
- Tipo de Interação:** A dropdown menu with "Active" selected.
- Densidade Semântica:** A dropdown menu with "MediumDensity" selected.
- Faixa Etária:** A text input field containing the number "15".
- At the bottom, there are three buttons: "Salvar Perfil" (highlighted in blue), "Novo Perfil", and "Fechar".

FIGURA 46 - Interface para cadastro de perfis técnicos

As informações do perfil pedagógico são utilizadas para definir os objetivos instrucionais para aquele conjunto de alunos, as atividades de aprendizagem que devem levá-los a atingir tais objetivos, os tipos de objetos de aprendizagem que compõem as atividades, além da estruturação das atividades de aprendizagem, proporcionando a adaptação do conteúdo e da navegação. Já as informações do perfil técnico são utilizadas apenas na seleção do tipo de objeto de aprendizagem empregado para apresentar determinado conteúdo, o que proporcionará a adaptação da apresentação.

Além dos estereótipos criados, é utilizado o modelo de *Overlay* para definir parte da dimensão pedagógica, registrando o nível de conhecimento que os alunos possuem sobre cada conceito abordado no curso, juntamente com o nível de conhecimento que eles

desejam alcançar. Através da interface apresentada na figura 47, o usuário relaciona cada um dos perfis pedagógicos selecionados aos conceitos incluídos no curso. Para isso, ele deve informar o nível de conhecimento que os alunos com aquele perfil possuem sobre o(s) conceito(s) indicado(s), juntamente com o nível de conhecimento que eles devem atingir após concluírem o curso.

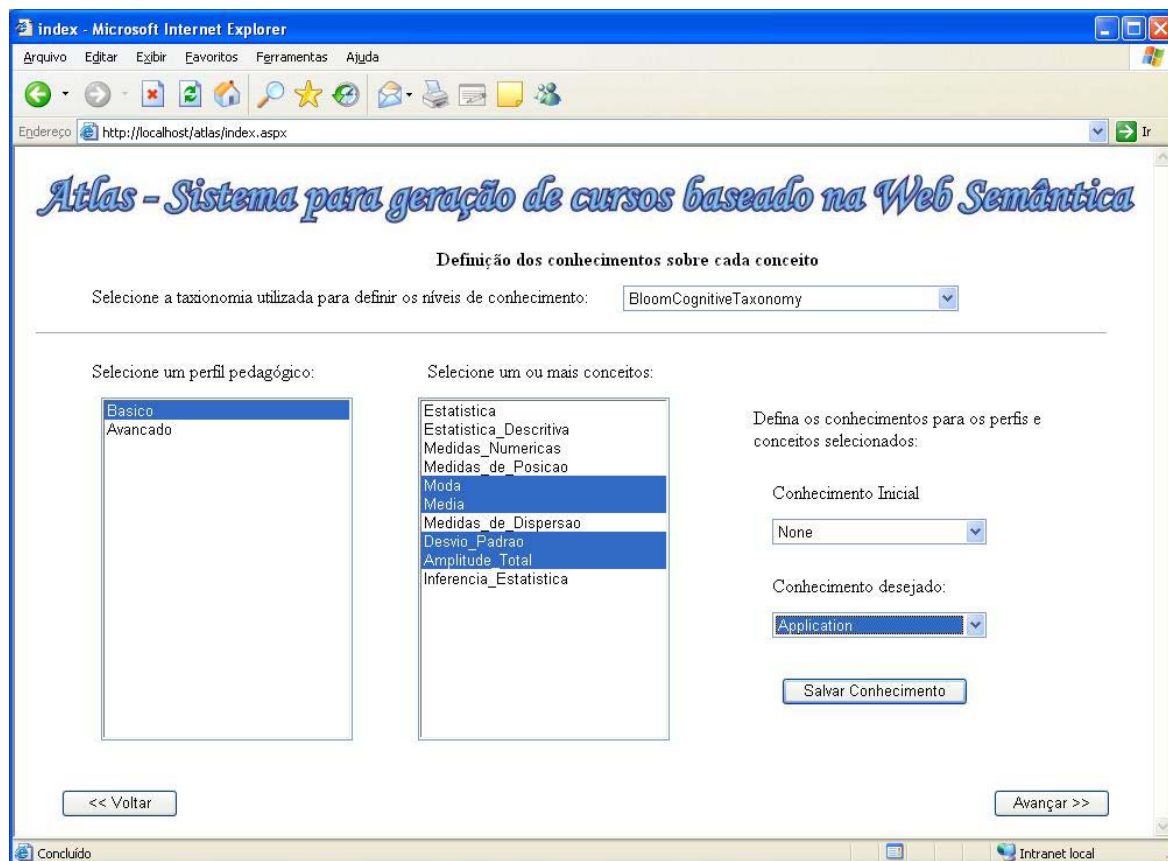


FIGURA 47 - Interface para definição do nível de conhecimento

Na parte superior da janela é apresentada uma lista com as diferentes taxionomias definidas na ontologia pedagógica para classificar os níveis de conhecimento. Assim, o sistema Atlas pode incorporar facilmente novas taxionomias, apenas importando uma nova versão desta ontologia.

Isto é possível porque está definido na ontologia pedagógica que todas as subclasses de *CognitiveTaxonomies* representam taxionomias para classificação de níveis de conhecimento e o Atlas apresenta todas estas subclasses na referida lista. Ao selecionar uma subclasse, o Atlas automaticamente altera as listas que indicam os valores a serem utilizados para descrever os níveis de conhecimento dos alunos, apresentando as instâncias da subclasse selecionada.

Após selecionar uma taxionomia, um dos perfís pedagógicos e um ou mais conceitos,

o usuário define o nível de conhecimento que o aluno possui sobre eles e o nível que pretende alcançar, através das listas dispostas na parte direita da interface, e salva as informações. Ao salvar as informações, novas instâncias da classe *KnowledgeConcepts* são criadas na ontologia do estudante e associadas ao perfil pedagógico correspondente.

Após definir estas informações, são criadas as instâncias da classe *Students*, de forma transparente para o usuário, através da combinação dos diferentes perfis técnicos e pedagógicos. Como exemplo, considera-se que um usuário tenha selecionado, além dos perfis pedagógicos *Básico* e *Avançado*, os perfis técnicos *Doméstico* e *Laboratório*, que refletem os diferentes locais que os alunos possuem para acessar o curso. Combinando estas informações, chega-se a quatro perfis de alunos diferentes, como mostra o quadro 47.

QUADRO 47 - Perfis de estudantes criados

Perfil	Dimensão Pedagógica	Dimensão Técnica
1	Básico	Doméstico
2	Básico	Laboratório
3	Avançado	Doméstico
4	Avançado	Laboratório

Todavia, mesmo gerando os quatro perfis automaticamente, o sistema não obriga que sejam criadas quatro estruturas diferentes para a organização e apresentação do conteúdo, pois em alguns casos existem relações, nas quais nenhum aluno se enquadra.

Após a combinação entre os perfis pedagógicos e técnicos, o usuário deve definir a estratégia instrucional a ser utilizada para os alunos de cada perfil pedagógico, através da interface apresentada na figura 48. Primeiramente, o usuário deve determinar qual classificação de estratégias instrucionais pretende utilizar, selecionando-a na lista disposta na parte superior da janela, a exemplo do que foi feito para definir a taxionomia que define os níveis de conhecimento.

Em seguida, é preciso designar qual estratégia deve ser utilizada para cada perfil pedagógico selecionado. É importante observar que a definição da estratégia instrucional é feita apenas com base no perfil pedagógico. Portanto, é utilizada a mesma estratégia para os alunos dos perfis 1 e 2, assim como é igual a estratégia utilizada com os alunos dos perfis 3 e 4. Porém, vale ressaltar que esta decisão foi tomada durante o projeto do sistema Atlas, pois as ontologias de domínio e do estudante permitem definir uma estratégia para cada combinação de perfil (neste exemplo, quatro diferentes).

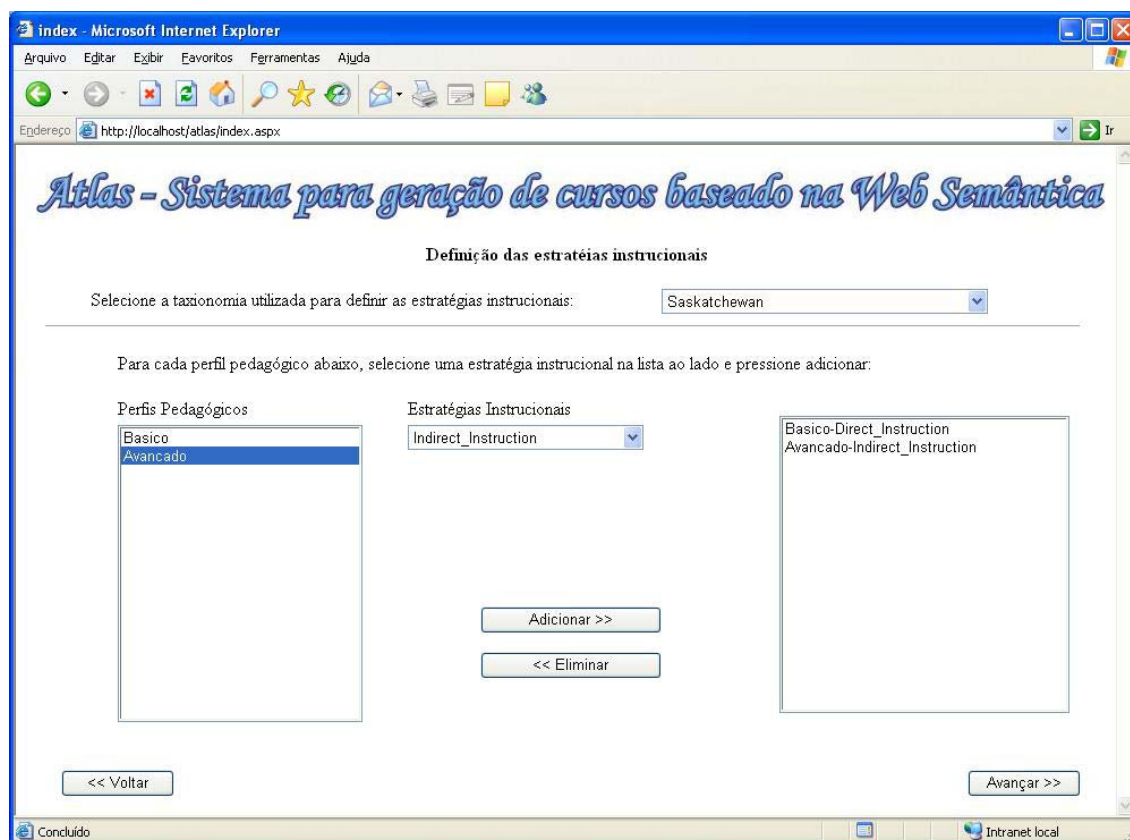


FIGURA 48 - Definição da estratégia instrucional

6.2.5 O Módulo para Definição dos Objetivos de Aprendizagem

A etapa seguinte à definição dos perfis dos alunos consiste no estabelecimento dos objetivos de aprendizagem a serem alcançados por eles, de acordo com os diferentes perfis pedagógicos. A principal interface usada nesta tarefa é apresentada na figura 49.

Na parte superior da janela existem duas listas: na primeira, estão os perfis pedagógicos selecionados para o curso e, na segunda, os conceitos. Quando o usuário seleciona um perfil pedagógico e um conceito, o sistema verifica na ontologia do estudante o nível de conhecimento que ele possui sobre aquele conceito e o nível que pretende alcançar, para informar ao usuário através de uma sentença do tipo: *Os alunos do perfil Basico possuem um conhecimento no nível "None" e devem alcançar um conhecimento no nível "Comprehension" sobre o conceito "Estatística"*.

A mensagem pode ser vista na figura 49, abaixo das listas com os perfis e os conceitos selecionados para o curso. Ela tem como função orientar a pessoa que está construindo o curso, para que sejam incluídos objetivos de aprendizagem compatíveis com o conhecimento que os alunos já possuem e com os resultados que devem ser atingidos.

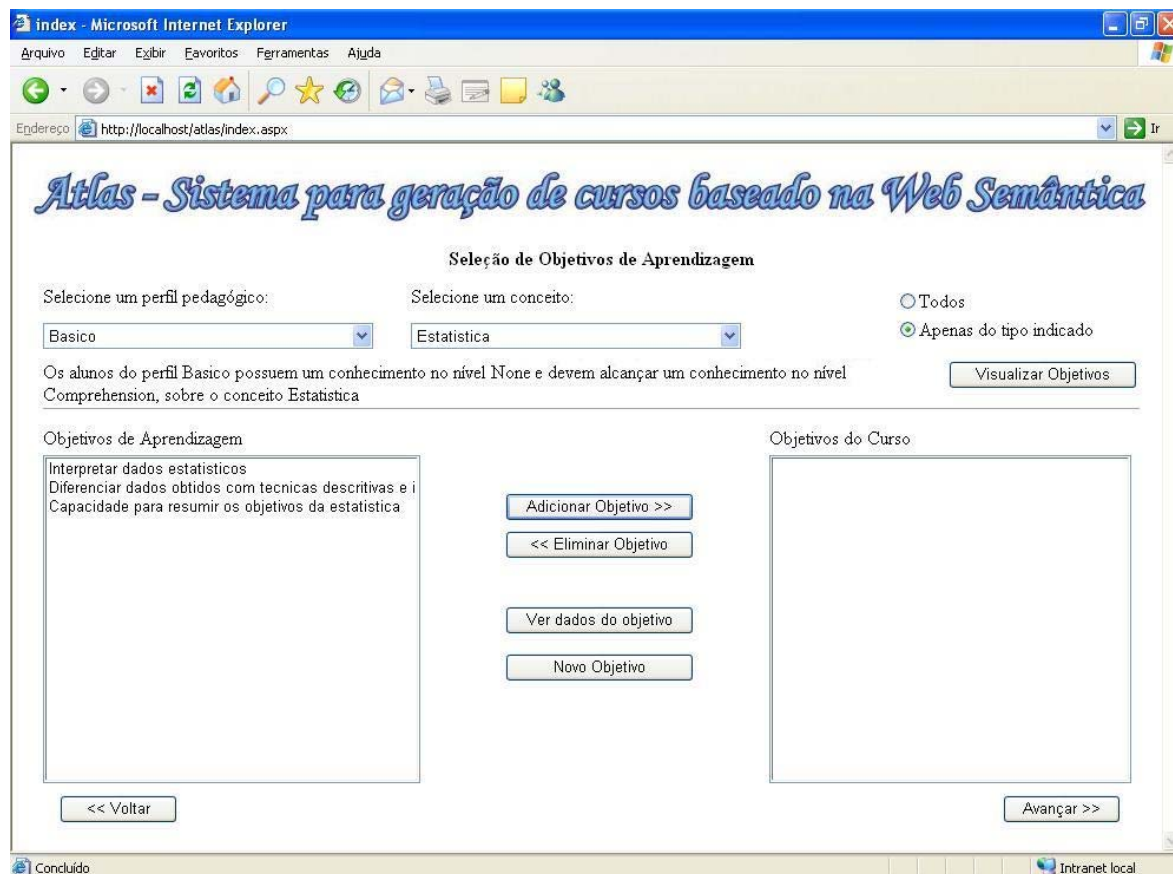


FIGURA 49 - Interface para seleção dos objetivos de aprendizagem (1)

Em seguida, o usuário visualiza os objetivos de aprendizagem já existentes na ontologia de domínio, que envolvem o conceito selecionado. Ele pode optar por visualizar todos eles ou apenas os que levam o aluno a atingir o nível de conhecimento desejado, neste caso, “*Comprehension*”. Esta funcionalidade possibilita a reutilização de objetivos de aprendizagem já definidos, facilita a sua busca, retornando apenas os objetivos associados ao conceito desejado, e auxilia o usuário, informando o nível de conhecimento que o objetivo deve possuir para proporcionar aos alunos o nível de conhecimento desejado.

A flexibilidade proporcionada pela ontologia pedagógica pode ser percebida, pois o usuário pode optar por visualizar todos os objetivos vinculados a um conceito ou apenas aqueles cujo nível de conhecimento é igual à meta estabelecida no perfil do estudante. Além destas, uma terceira opção pode ser implementada em sistemas que utilizam esta ontologia, usando o atributo *order*, da classe *CognitiveTaxonomies*. Em vez de retornar apenas os objetivos que levam ao mesmo nível de conhecimento do aluno, o sistema poderá retornar, além destes, todos os que levam a níveis superiores para que o usuário possa optar por incluí-los, proporcionando uma aprendizagem além da prevista aos alunos.

Entretanto, esta opção não foi implementada no Atlas.

Apesar de informar o nível de conhecimento que um objetivo deve proporcionar aos alunos de um perfil durante a seleção dos objetivos de aprendizagem, o Atlas não impede que o usuário inclua objetivos de outros níveis, a exemplo do que faz na seleção de conceitos. Todavia, também é possível criar um outro sistema baseado na mesma ontologia, que só permita a inclusão de objetivos que levem os alunos a atingirem o nível de conhecimento que eles precisam.

Caso tenha interesse em algum dos objetivos listados, o usuário pode ver suas informações através da interface mostrada na figura 50. Nela, são mostrados os valores das propriedades do objetivo de aprendizagem definidas na ontologia de domínio.

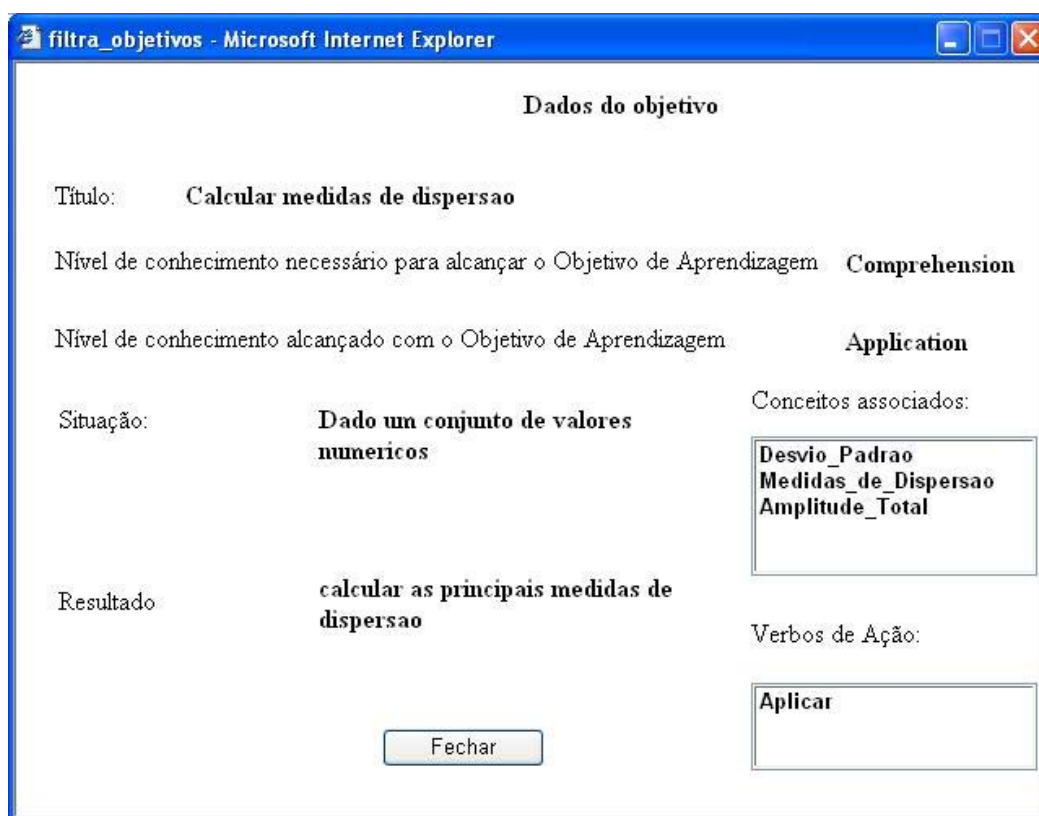


FIGURA 50 - Interface para visualização dos dados de um objetivo de aprendizagem

Retornando a interface apresentada na figura 49, o usuário pode repetir o processo descrito para os demais conceitos abordados no curso, armazenando na lista disposta na parte direita da janela os objetivos dos alunos com o perfil pedagógico selecionado. A figura 51 mostra a mesma interface da figura 49, porém, com um conjunto de objetivos de aprendizagem já selecionados para os alunos com o perfil pedagógico *Básico*.

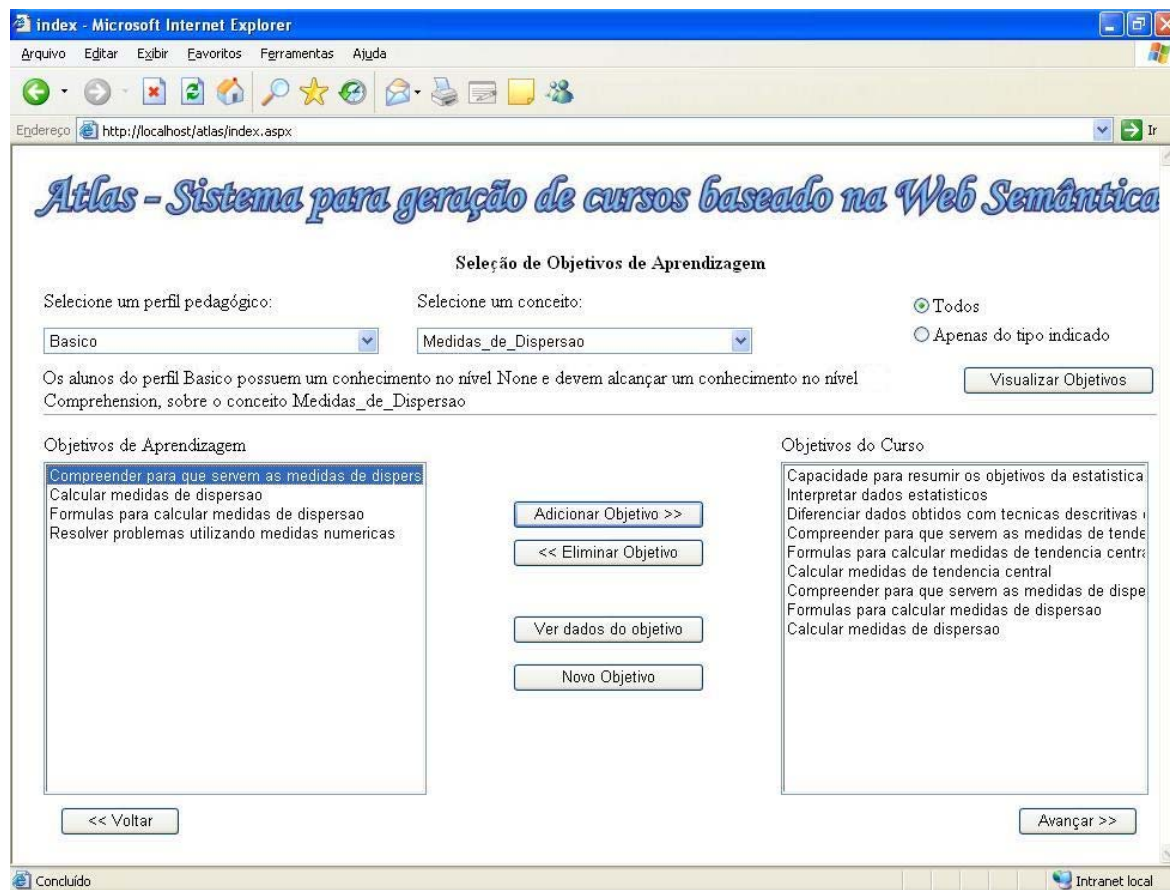


FIGURA 51 - Interface para seleção dos objetivos de aprendizagem (2)

Ao terminar a definição dos objetivos para um perfil pedagógico, o usuário deve selecionar um outro na lista de perfis. O sistema, então, armazena os objetivos do primeiro perfil e os exclui da lista de objetivos do curso, para que os objetivos do novo perfil possam ser selecionados da mesma maneira.

É possível, e até provável, que nem todos os objetivos de aprendizagem necessários já existam e estejam prontos para serem reutilizados. Nesta situação, o usuário pode criar seus próprios objetivos de aprendizagem, através da interface mostrada na figura 52, onde ele deve definir os valores dos atributos da nova instância da classe *LearningObjectives* criada na ontologia de domínio.

Nessa figura, pode-se perceber, além dos campos que correspondem aos atributos da referida classe, a lista onde o usuário indica a taxionomia utilizada para descrever os níveis de conhecimento do objetivo de aprendizagem.

As propriedades *Verbs*, *Situation* e *Result* dos objetivos de aprendizagem não são utilizadas nos módulos de busca do Atlas e servem apenas para que o usuário tenha mais detalhes sobre o objetivo. Entretanto, como já foi explicado na apresentação da ontologia pedagógica, outros sistemas podem usá-las para verificar inconsistências nos objetivos.

The screenshot shows a web browser window titled 'cria_objetivos - Microsoft Internet Explorer'. The main content area is titled 'Cadastro de Objetivos de Aprendizagem'. It contains several form elements:

- Título do Objeto de Aprendizagem:** A text input field containing 'Capacidade para resumir os obje'.
- Classificação de Objetivos de Aprendizagem:** A dropdown menu with 'BloomCognitiveTaxonomy' selected.
- Nível de conhecimento necessário para acompanhar o Objetivo de Aprendizagem:** A dropdown menu with 'Nenhum' selected.
- Nível de conhecimento alcançado com o Objetivo de Aprendizagem:** A dropdown menu with 'Comprehension' selected.
- Verbo de Ação:** A dropdown menu with 'Descrever' selected.
- Conceitos Abordados:** A list box containing 'Estatística', 'Estatística_Descritiva', and 'Inferencia_Estatística'. Below it is a button labeled 'Selecionar Conceitos'.
- Descrição da Situação:** A text area containing 'Mediante a questão: Para que se'.
- Descrição do Resultado Esperado:** A text area containing 'Descricao sucinta dos objetivos'.
- Buttons:** 'Novo Objetivo', 'Salvar Objetivo', and 'Fechar' are located at the bottom of the form.

FIGURA 52 - Interface para criação de novos objetivos de aprendizagem

6.2.6 O Módulo para Definição das Atividades de Aprendizagem

Após selecionar os objetivos de aprendizagem para cada perfil pedagógico inserido no curso, cabe ao usuário definir as atividades de aprendizagem utilizadas para fazer com que os alunos atinjam tais objetivos. Este módulo é composto por várias interfaces que são acessadas simultaneamente às interfaces do módulo para definição dos objetos de aprendizagem descritos posteriormente. Por isso, foi criado um diagrama de navegação para ilustrar as relações entre as várias janelas, como pode ser visto na figura 53.

Este diagrama pode ser visto como a expansão da janela denominada de *Atividades de Aprendizagem*, apresentada na figura 38. Seu ponto de partida é a interface para definição dos objetivos de aprendizagem e o seu término leva ao módulo para a estruturação do curso, como pode ser visto nas figuras 38 e 53.

A interface principal deste módulo (figura 54) é semelhante à utilizada para seleção dos objetivos. Nela, o usuário deve, primeiramente, selecionar um perfil pedagógico na lista posicionada na parte superior esquerda da janela. Ao executar esta tarefa, os objetivos de aprendizagem associados àquele perfil são apresentados na lista situada na parte superior central. Quando o usuário seleciona um dos objetivos de aprendizagem listados, o sistema busca na ontologia de domínio e apresenta na lista posicionada na parte esquerda inferior, todas as atividades relacionadas àquele objetivo.

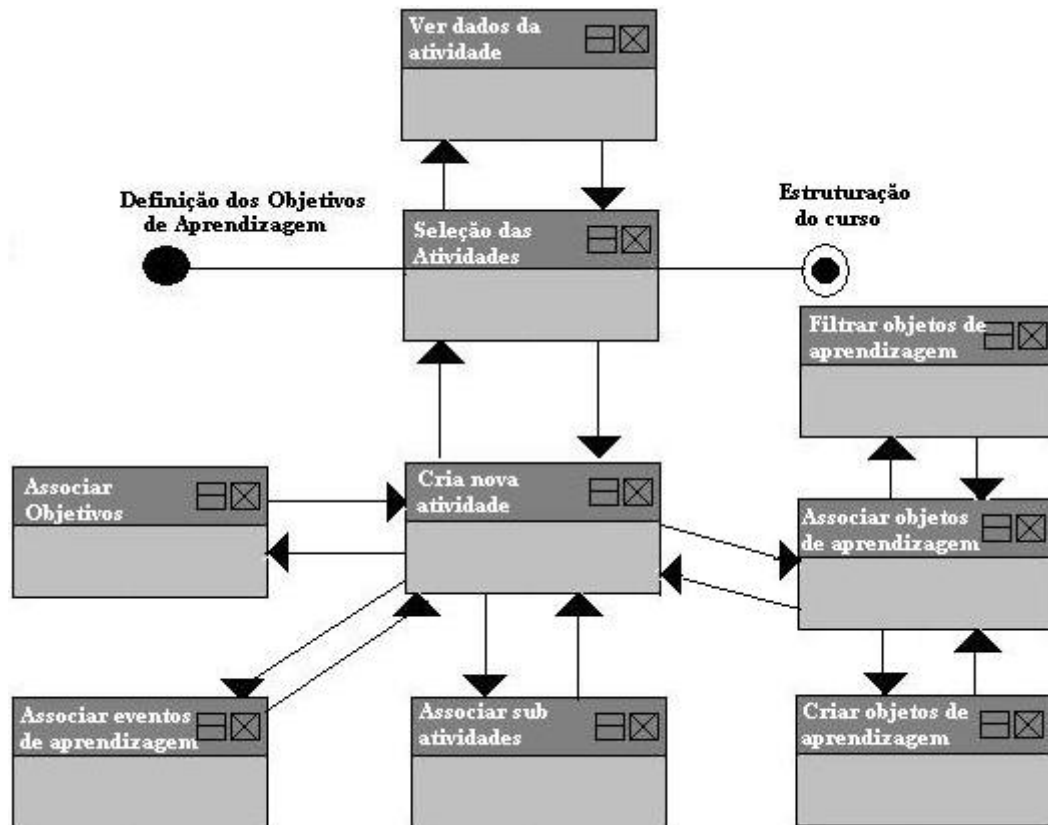


FIGURA 53 - Diagrama de navegação para a definição das atividades e objetos de aprendizagem

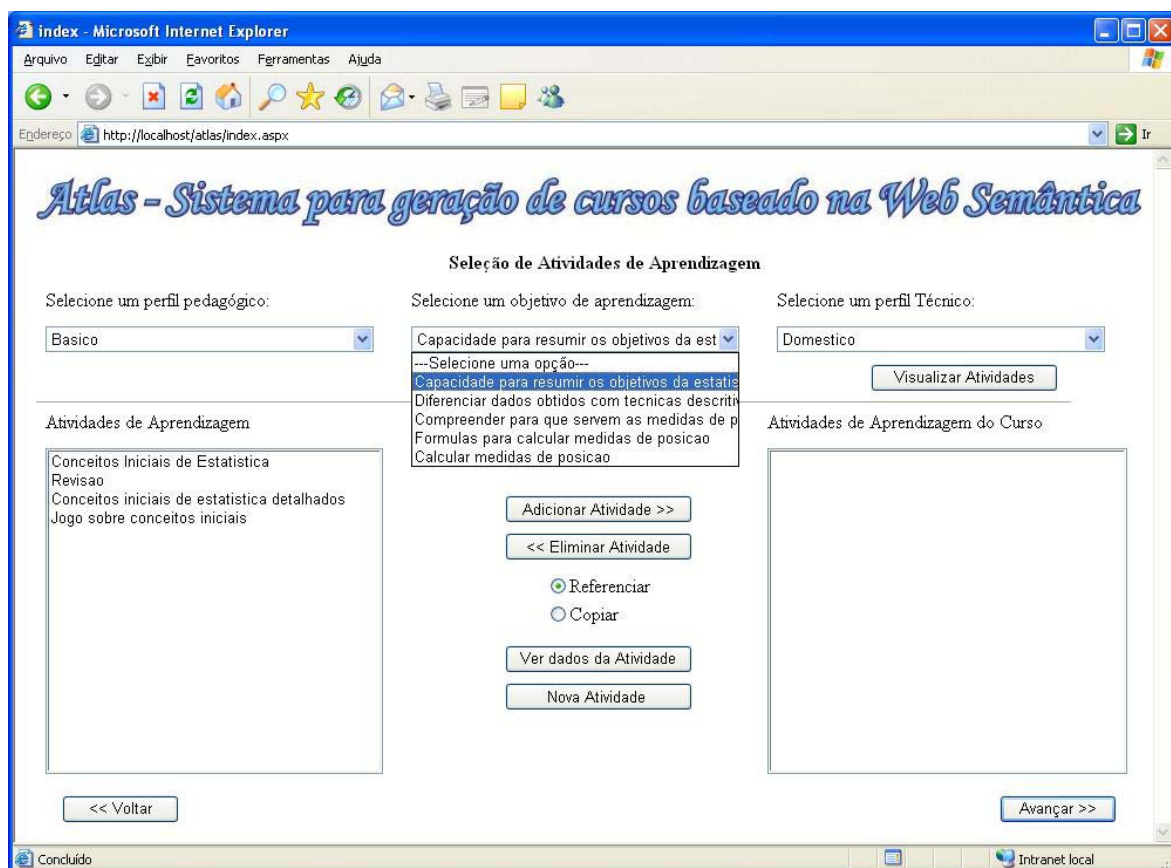


FIGURA 54 - Interface para seleção de atividades de aprendizagem (1)

Também de maneira semelhante ao que é feito durante a seleção de objetivos de aprendizagem, o usuário pode selecionar uma das atividades listadas para visualizar suas informações, através da interface mostrada na figura 55, antes de decidir se ela deve ser incluída no curso ou não.

A principal diferença entre as interfaces para seleção de objetivos e de atividades de aprendizagem reside no fato de que a última também deve contemplar os diferentes perfis técnicos dos alunos, o que não era feito durante a seleção dos objetivos. Isso ocorre porque os alunos que possuem o mesmo perfil pedagógico e perfis técnicos diferentes possuem os mesmos objetivos dentro do curso, mas que podem ser alcançados através de atividades de aprendizagem diferentes.

Assim, além de definir atividades diferentes para alunos com perfis pedagógicos distintos, o usuário deve relacionar atividades diferentes para alunos que possuam o mesmo perfil pedagógico, mas com perfis técnicos diversos. Desta forma, é possível construir atividades que levem em conta as características do aluno em relação ao meio usado para acessar a Internet, o tempo que possui para realizar o curso e outras características definidas na dimensão técnica do seu perfil.

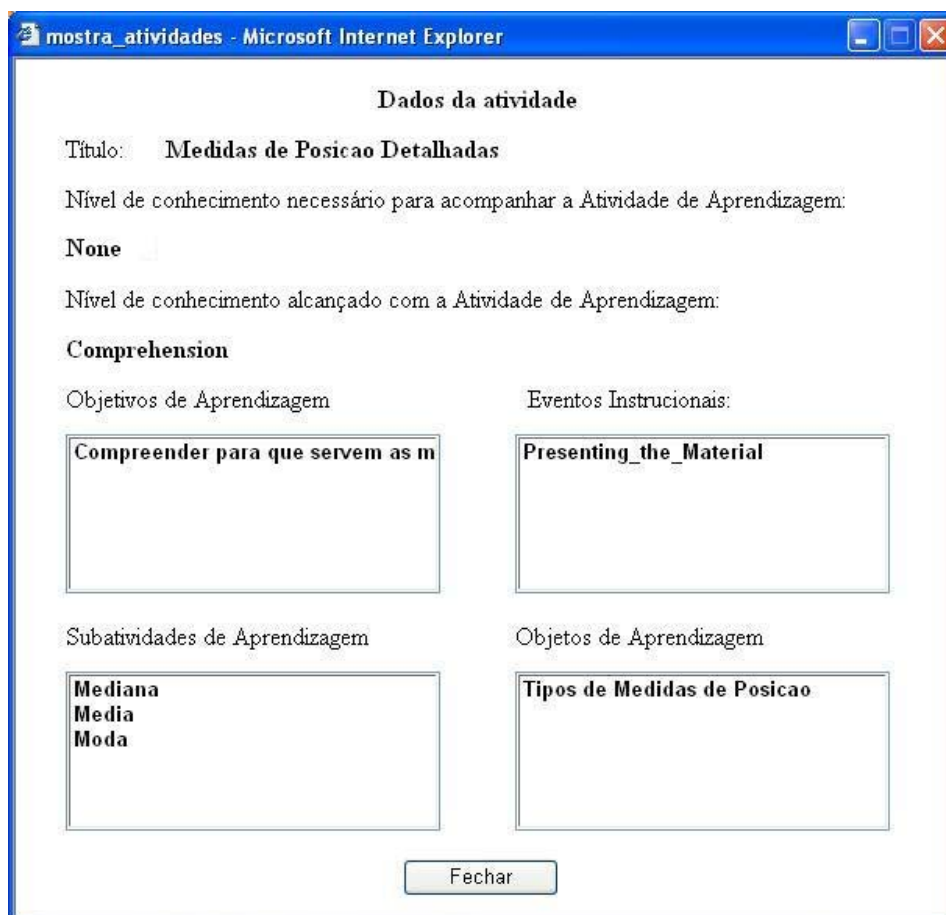


FIGURA 55 - Interface para visualização dos dados de uma atividade de aprendizagem

Ao selecionar a dimensão pedagógica e a dimensão técnica, o usuário define que as atividades adicionadas ao curso a partir daquele momento devem ser incluídas na estrutura associada aos alunos com o perfil indicado. A seleção das atividades deve ser repetida para os diferentes objetivos associados a um perfil de aluno. O resultado deste processo é um conjunto de atividades de aprendizagem que formam a estrutura do curso a ser apresentada aos alunos com aquele perfil, como pode ser visto na figura 56. Esta figura apresenta, na lista posicionada na parte inferior direita, as três atividades a serem realizadas pelos alunos com o perfil pedagógico *Básico* e perfil técnico *Doméstico*, para atingir os objetivos propostos.

Após definir as atividades para todos os objetivos existentes, o usuário deve alterar a dimensão técnica do perfil e repetir os passos apresentados, indicando as atividades de aprendizagem a serem oferecidas aos alunos para que eles atinjam seus objetivos. Da mesma maneira, todas as ações de indicação da dimensão técnica do perfil, seleção dos objetivos e seleção das atividades devem ser repetidas para os demais perfis pedagógicos selecionados para o curso.

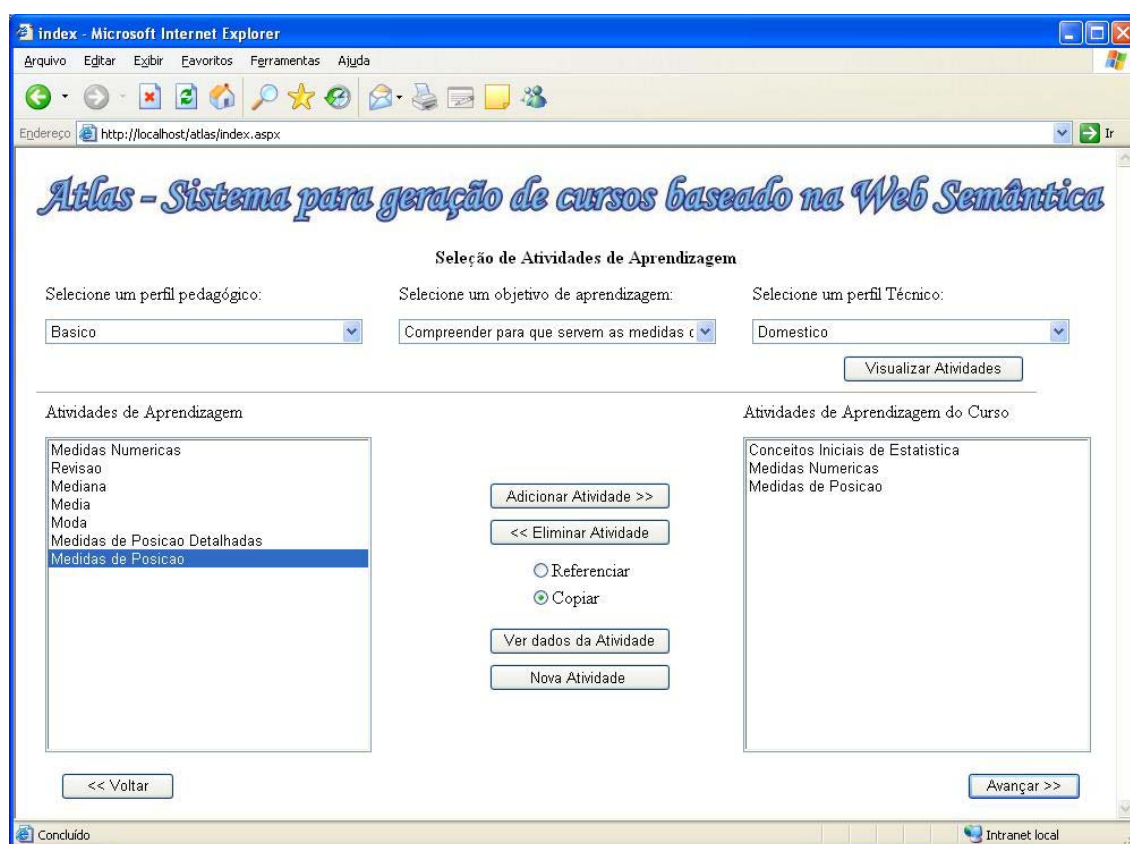


FIGURA 56 - Interface para seleção de atividades de aprendizagem (2)

A figura 57 ilustra a mesma interface apresentada na figura 56, mas com as

atividades de aprendizagem associadas aos alunos com outro perfil. Para visualizar as atividades de alunos com diferentes perfis, o usuário precisa apenas selecionar os perfis desejados nas listas existentes na parte superior da tela, para que o sistema atualize a lista de atividades apresentada na parte inferior direita da interface.

Ao reutilizar uma atividade, o usuário pode optar por copiar ou apenas referenciar seus LO, através dos botões de seleção presentes no centro da janela. Quando é selecionada a opção *copiar*, o Atlas busca os LO (arquivos de conteúdo) através do atributo *lom-tech:location* armazenado na ontologia e efetua uma cópia dos mesmos que é incluída no curso gerado. Já na opção *referenciar*, o Atlas apenas insere o endereço do LO no curso para que ele seja acessado a partir de sua localização original.

A referência direta a LO pode ser perigosa, pois, o responsável por ele pode retirá-lo da Internet ou mudar seu endereço sem saber que ele é referenciado pelo curso. Entretanto, ela torna possível o acesso a LO que requerem muito espaço em disco para serem armazenados ou para referenciar programas que precisem de pré-requisitos que não estão disponíveis no computador onde o LMS está instalado.

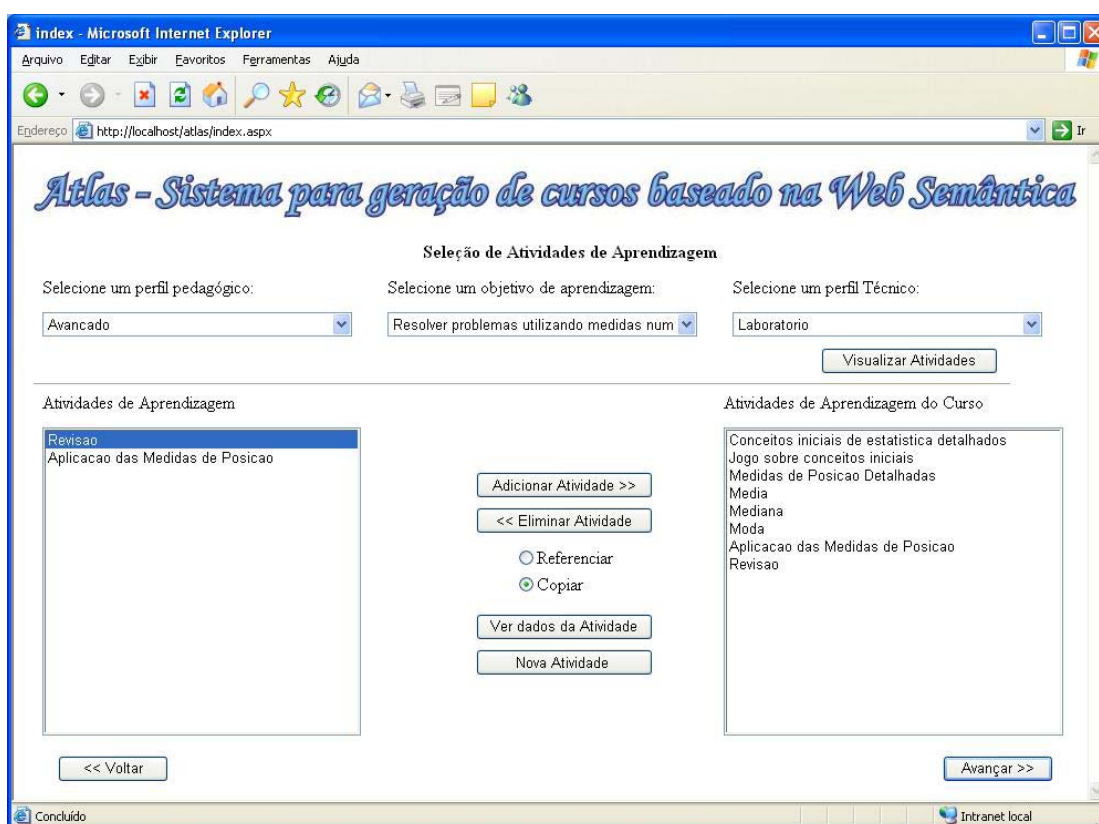


FIGURA 57 - Interface para seleção de atividades de aprendizagem (3)

Além de possibilitar a reutilização de atividades de aprendizagem já existentes, a interface apresentada na figura 54 permite que o usuário acesse a função para criar suas

próprias atividades de aprendizagem (figura 58). Através desta interface, o usuário define os valores que descrevem a atividade de aprendizagem e que são armazenados nas propriedades da instância da classe *LearningActivities*, criada na ontologia de domínio. Informações como título, nível de conhecimento necessário para acompanhar a atividade e nível de conhecimento alcançado após a sua realização são digitados ou selecionados nesta mesma interface.

FIGURA 58 - Interface para criação de atividades de aprendizagem

Entretanto, outras informações são selecionadas a partir de janelas auxiliares, como a apresentada na figura 59, que possibilitam ao usuário selecionar os eventos instrucionais que a atividade proporciona, dentre os definidos na ontologia pedagógica. Interfaces semelhantes a esta são utilizadas para selecionar os objetivos de aprendizagem associados à atividade criada e às subatividades que a compõem. A indicação dos LO que constituem uma atividade é uma tarefa mais complexa, por isso, ela é apresentada separadamente na próxima seção.



FIGURA 59 - Interface para seleção dos Eventos Instrucionais

6.2.7 O Módulo para Definição dos Objetos de Aprendizagem

A partir da interface para criação de atividades de aprendizagem apresentada na figura 58, o usuário deve acessar a interface para selecionar os objetos de aprendizagem que devem compor a atividade criada (figura 60). Nela, existem duas listas principais: a primeira, à esquerda, apresenta os títulos de vários LO já cadastrados no sistema; a segunda, à direita, armazena os LO que o usuário deseja associar à atividade.

A exemplo do que foi feito nas interfaces para seleção de objetivos e atividades de aprendizagem, elaborou-se a interface apresentada na figura 61 para apresentar todas as informações de um LO selecionado na lista da esquerda. Desta maneira, o usuário pode visualizar as suas características antes de decidir se ele deve ou não fazer parte daquela atividade de aprendizagem.

Como a quantidade de LO cadastrados pode se tornar muito grande com o passar do tempo, foi desenvolvida a interface apresentada na figura 62, visando tornar mais fácil a tarefa de encontrar LO com as características desejadas. Ela permite que o usuário estabeleça valores para um ou mais atributos do LO e realiza uma busca na ontologia, retornando apenas os que possuem as características estabelecidas.

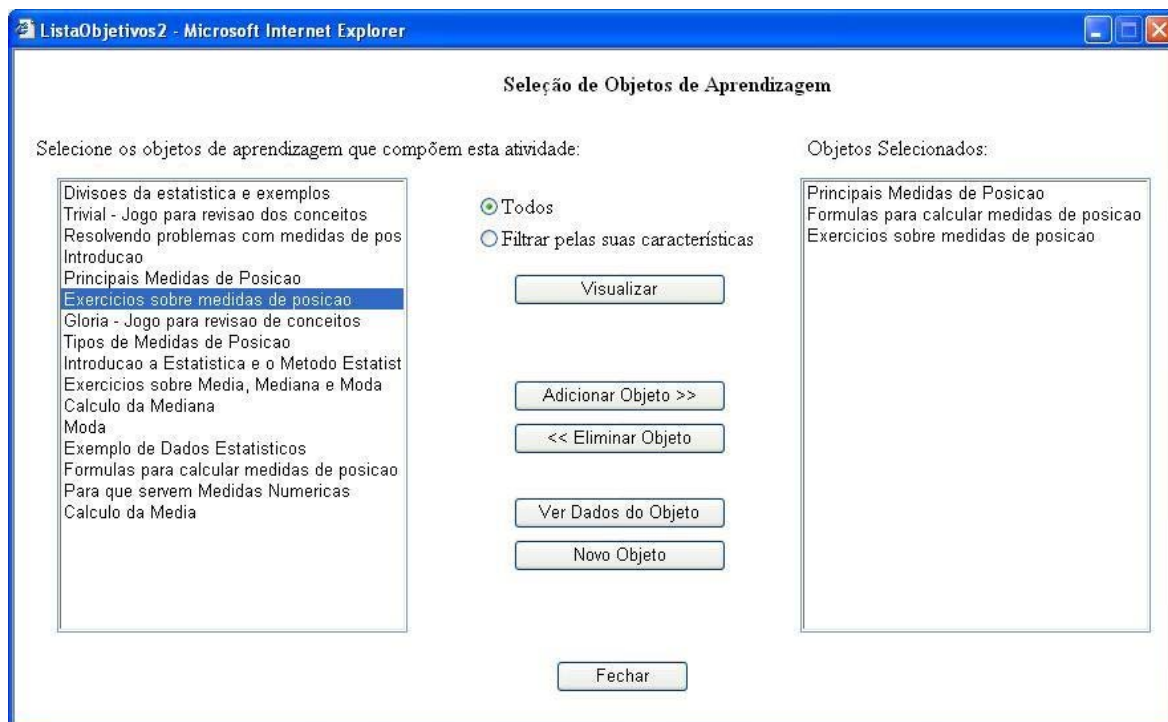


FIGURA 60 - Interface para seleção dos Objetos de Aprendizagem

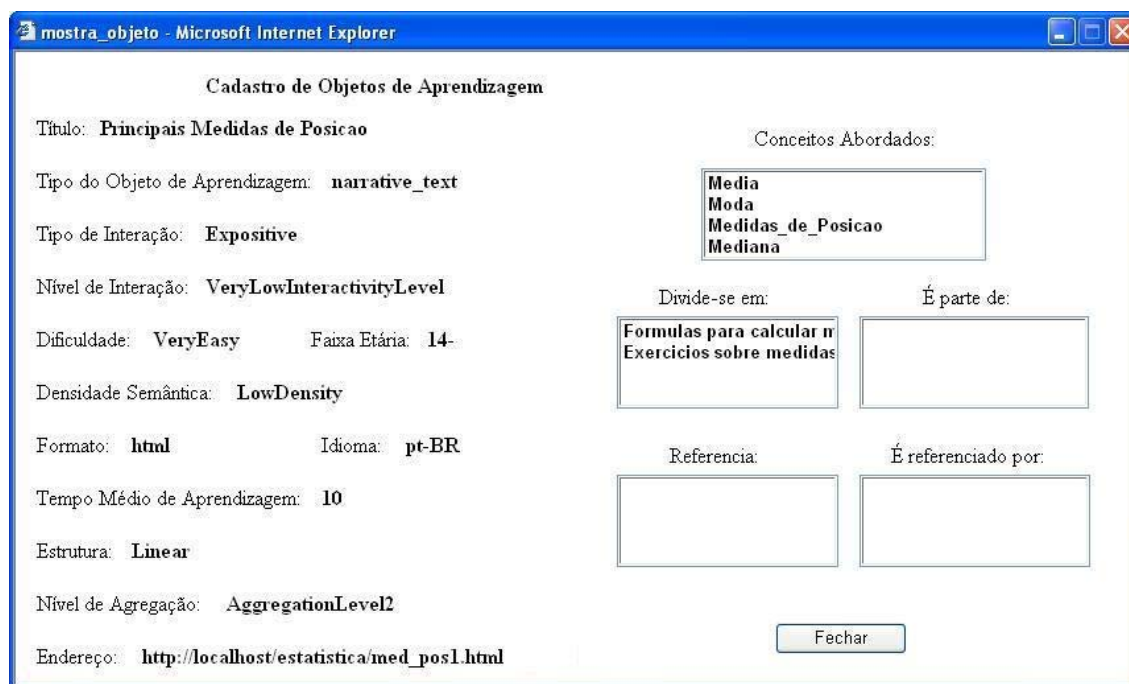


FIGURA 61 - Interface para apresentar as informações de um LO

É neste momento que as informações do perfil do aluno podem ser utilizadas para orientar o usuário na busca e na seleção dos LO que devem compor as diversas atividades de aprendizagem. O usuário pode definir os valores de uma ou mais propriedades dos LO definidas na ontologia de domínio. Ao efetuar a busca, o Atlas utiliza o conectivo lógico “E” para unir as cláusulas elaboradas a partir das propriedades cujos valores foram definidos, sendo as demais ignoradas. Por fim, o usuário pode criar um LO e cadastrá-lo

através da interface apresentada na figura 63, caso ele não encontre os LO com as características que deseja.

FIGURA 62 - Interface para filtrar Objetos de Aprendizagem

O Atlas não possibilita a construção propriamente dita dos LO. Eles podem ser desenvolvidos nas várias ferramentas existentes para criação de *homepages*, caso sejam arquivos HTML, ou qualquer outro sistema, já que os LO podem ser arquivos executáveis, apresentações, bancos de dados, planilhas etc. Porém, esta interface é fundamental, por possibilitar a anotação do LO criado com as propriedades definidas na ontologia de domínio.

Esta anotação é a base de todo processo de reutilização dos componentes instrucionais, pois os LO são as partes básicas que são agrupadas para formar os componentes instrucionais mais complexos, como atividades de aprendizagem, estruturas para organização e apresentação do conteúdo e cursos.

The screenshot shows a web browser window titled 'cria_objetos - Microsoft Internet Explorer'. The main content area is titled 'Cadastro de Objetos de Aprendizagem'. The form includes the following fields and options:

- Título:** Text input field containing 'Tipos de Medidas de Posicao'.
- Tipo do Objeto de Aprendizagem:** Dropdown menu with 'narrative_text' selected.
- Tipo de Interação:** Dropdown menu with 'Expositive' selected.
- Nível de Interação:** Dropdown menu with 'VeryLowInteractivityLevel' selected.
- Dificuldade:** Dropdown menu with 'Easy' selected.
- Faixa Etária:** Text input field containing '14-'.
- Densidade Semântica:** Dropdown menu with 'VeryLowDensity' selected.
- Formato:** Dropdown menu with 'html' selected.
- Idioma:** Dropdown menu with 'pt-BR' selected.
- Tempo Médio de Aprendizagem:** Text input field containing '25'.
- Estrutura:** Dropdown menu with 'Atomic' selected.
- Nível de Agregação:** Dropdown menu with 'AggregationLevel1' selected.
- Conceitos Abordados:** A list box containing 'Medidas_de_Posicao', 'Moda', 'Media', and 'Mediana'. A 'Selecionar' button is located below the list.
- Divide-se em:** An empty text input field.
- É parte de:** An empty text input field.
- Referencia:** An empty text input field.
- É referenciado por:** An empty text input field.
- Arquivo:** A text input field containing 'E:\Doutorado\Cursos\LO\Avancado\med_pos.html' and a 'Procurar...' button.

At the bottom of the form, there are three buttons: 'Novo Objeto', 'Salvar Objeto', and 'Fechar'. Additionally, there are radio buttons for 'Divide-se em', 'É parte de', 'Referencia', and 'É referenciado por', with 'Divide-se em' being selected.

FIGURA 63 - Interface para cadastro de LO

Para cada LO cadastrado, todas as suas informações (metadados) são armazenadas nas propriedades da instância da classe *LearningObjects* criada na ontologia de domínio. Assim, um outro usuário que utilize o sistema pode realizar buscas precisas de LO através das informações constantes na ontologia. Como uma das propriedades armazenadas dos LO é a sua localização física, o Atlas pode recuperá-lo para reutilizá-lo no novo curso, sem precisar manter uma cópia armazenada do seu conteúdo.

Além de manter as informações dos LO na ontologia de domínio, que podem ser utilizadas pelo próprio Atlas ou por outros sistemas baseados nesta ontologia, uma cópia dos metadados é adicionada ao arquivo de conteúdo do LO (desde que ele seja uma página HTML). Embora estas informações não sejam utilizadas em nenhum momento pelo Atlas, elas foram inseridas para auxiliar no desenvolvimento da *Web Semântica*, o que requer conteúdos descritos através de metadados definidos em ontologias. Dessa maneira, mecanismos de busca e outros sistemas envolvidos com a EBW podem encontrar mais facilmente os LO na *Web* e entender o seu significado, mesmo que não utilizem o Atlas.

O quadro 48 mostra o código inserido no arquivo de um LO. Na linha 5 está a associação do *namespace* que referencia a ontologia de domínio ao prefixo *concepts*. Através deste prefixo, nas linhas 6 e 7 é indicado que aqueles metadados descrevem um objeto de aprendizagem definido na referida ontologia com o identificador *LOCurso12*.

Nas linhas seguintes, estão os valores que o LO possui para as propriedades definidas na ontologia. Merecem destaque as linhas 27 a 31, que utilizam a propriedade *dc:keyword* para associar o LO aos conceitos definidos na ontologia, estabelecendo claramente os assuntos tratados por ele.

Os novos LO cadastrados são apresentados ao usuário através da interface exposta na figura 60, para serem incluídos na nova atividade de aprendizagem.

QUADRO 48 - Metadados adicionados ao arquivo do LO *Exemplo de Dados Estatísticos*

```

1. <rdf:RDF
2.   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.   xmlns:dc="http://purl.org/dc/elements/1.1/"
4.   xmlns:lom-edu="http://ltsc.ieee.org/2002/09/lom-educational#"
5.   .....
6.   xmlns:concepts="http://localhost/ontologias/concepts.owl#">
7.   <concepts:LearningObjects
8.     rdf:about="http://localhost/ontologias/concepts.owl#LOCurso12">
9.     <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
10.       Exemplo de Dados Estatísticos
11.     </dc:title>
12.     <lom-edu:ageRange rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
13.       14-
14.     </lom-edu:ageRange>
15.     <dc:format rdf:resource="http://localhost/ontologias/concepts.owl#html"/>
16.     <dc:language rdf:resource=
17.       "http://localhost/ontologias/concepts.owl#pt-BR"/>
18.     <lom-edu:semanticDensity rdf:resource=
19.       "http://ltsc.ieee.org/2002/09/lom-9educational#LowDensity"/>
20.     <lom-tech:location rdf:datatype=
21.       "http://www.w3.org/2001/XMLSchema#anyURI">
22.       http://localhost/estatistica/intro_exemplos.html
23.     </lom-tech:location>
24.     <lom-edu:interactivityType rdf:resource=
25.       "http://ltsc.ieee.org/2002/09/lom-educational#Expositive"/>
26.     <lom-edu:difficulty rdf:resource=
27.       "http://ltsc.ieee.org/2002/09/lom-educational#Easy"/>
28.     <dc:keyword rdf:resource=
29.       "http://localhost/ontologias/concepts.owl#Estatistica"/>
30.     <dc:keyword rdf:resource=
31.       "http://localhost/ontologias/concepts.owl#Estatistica_Descritiva"/>
32.     <dc:keyword rdf:resource=
33.       "http://localhost/ontologias/concepts.owl#Inferencia_Estatistica"/>
34.   </concepts:LearningObject>
35. </rdf:RDF>

```

6.2.8 O Módulo para Definição das Estruturas do Curso

Após selecionar as atividades de aprendizagem que levam os alunos de cada perfil a atingir os seus objetivos, o usuário deve determinar a ordem em que elas devem ser apresentadas ao usuário e a sua hierarquia, através da interface apresentada na figura 64.

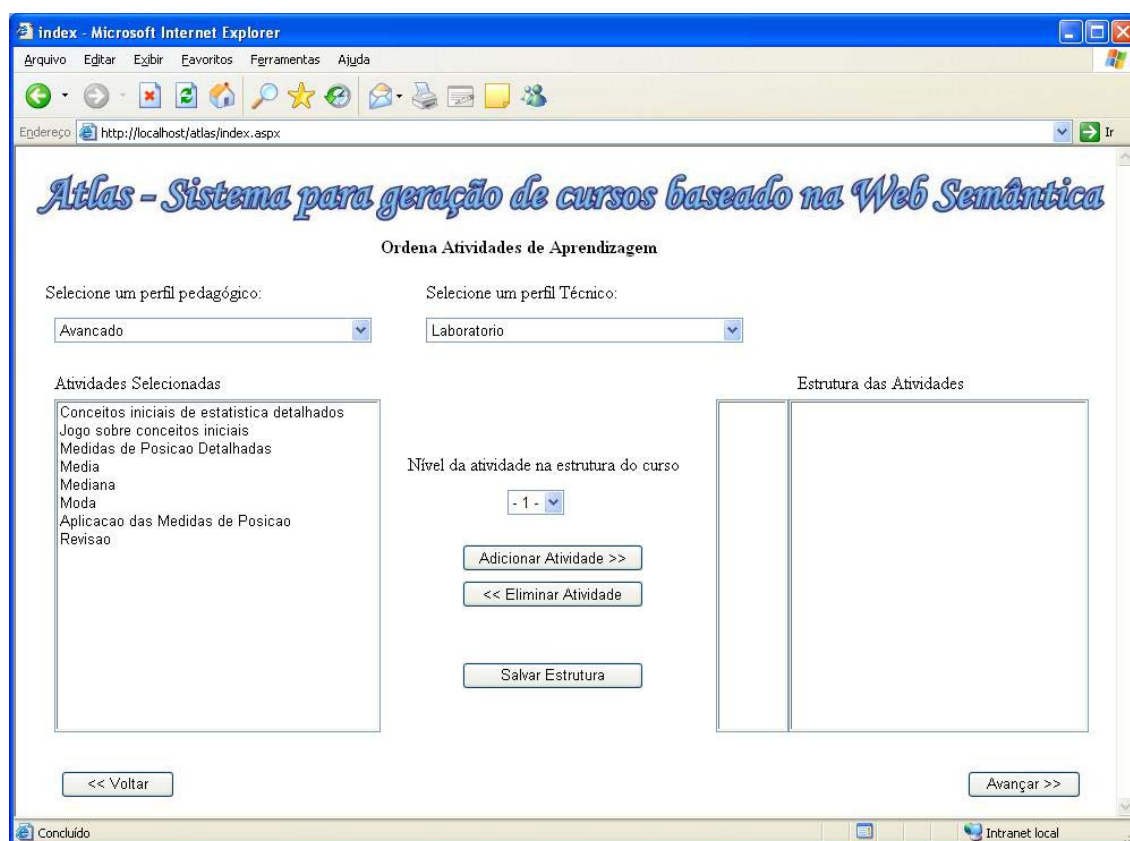


FIGURA 64 - Interface para estruturar as Atividades de Aprendizagem (1)

Embora a ontologia de domínio estabeleça a relação entre atividades e subatividades de aprendizagem, podem ocorrer casos em que o usuário deseja criar uma estrutura de apresentação diferente. Por isso, o Atlas permite que ele defina a hierarquia e a ordem das atividades, que podem ser iguais ou diferentes das existentes na ontologia.

Nesta interface, o usuário deve selecionar um dos perfis pedagógicos e um dos perfis técnicos selecionados para o curso. O sistema apresenta na lista à esquerda as atividades a serem realizadas pelos alunos com estes perfis para que sejam indicadas a ordem de apresentação e a sua hierarquia.

Para definir a ordem, o usuário deve selecionar na lista da esquerda a atividade que ele deseja apresentar primeiro aos alunos e passá-la para a lista da direita. Repetindo este mesmo processo para as demais atividades, o usuário está ordenando-as, como pode ser visto nas figuras 65 e 66, que apresentam em momentos distintos o processo de ordenação

e hierarquização das atividades de aprendizagem para um mesmo perfil de aluno. Para hierarquizar as atividades, basta que o usuário altere o nível em que a atividade selecionada deve ser inserida. Na figura 65 pode-se observar que as três primeiras atividades estão no primeiro nível, pois nenhuma delas é subatividade de outra. Já a atividade *Média*, que é incluída em seguida, foi definida como de nível 2, fazendo com que o sistema a colocasse como uma subatividade da atividade *Medidas de Posição Detalhadas*.

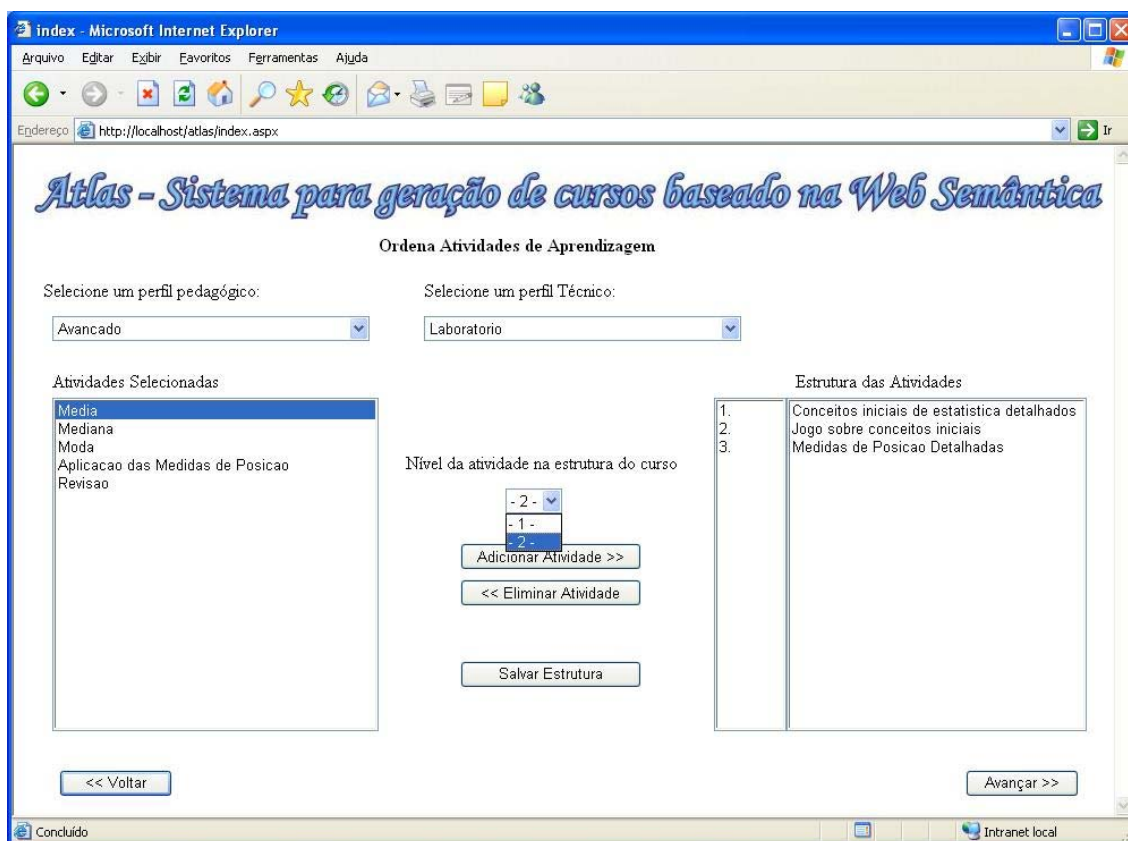


FIGURA 65 - Interface para estruturar as Atividades de Aprendizagem (2)

O mesmo ocorreu com as atividades *Mediana*, *Moda* e *Aplicação das medidas de posição*, que foram inseridas como subatividades de *Medidas de posição detalhadas*.

Embora a ontologia de domínio não apresente limites para a quantidade de níveis das subatividades, na implementação do sistema Atlas isto foi limitado a cinco. Na figura 65 aparecem apenas os níveis 1 e 2 como valores possíveis, porque é feito um controle sobre os níveis possíveis em que uma subatividade pode ser inserida. Desta maneira, evitam-se inconsistências do tipo: uma atividade de nível 1 possui uma subatividade de nível 3, sem nenhuma atividade de nível 2 entre elas. Por exemplo, atividade de nível 1, 3 – *Medidas de Posição Detalhadas*, possui a subatividade de nível 3, 3.1.1 – *Média*, sem nenhuma subatividade de nível 2, com a numeração 3.1. Como no exemplo citado, a última atividade

inserida é de nível 1, após ela só pode ser inserida uma nova atividade de nível 1 ou uma subatividade de nível 2. Supondo que a última atividade seja de nível 2, após ela podem ser inseridas novas atividades de nível 1 e 2 ou uma subatividade de nível 3.

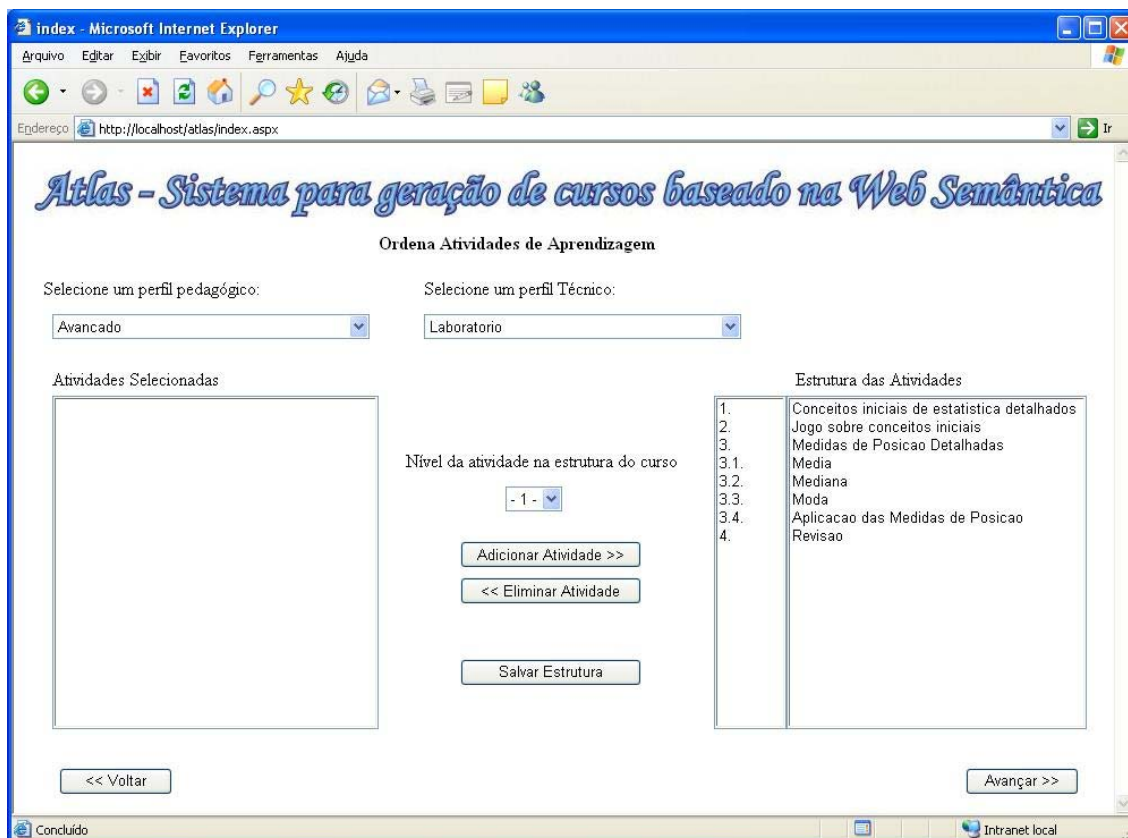


FIGURA 66 - Interface para estruturar as Atividades de Aprendizagem (3)

Após definir todas as informações necessárias, o usuário deve acessar o último módulo do sistema, que não possui nenhuma interface, para gerar o curso.

6.3 Construção do Curso Exemplo

Nesta seção são apresentadas as informações utilizadas na construção de um curso de Introdução à Estatística através do sistema Atlas, a partir das quais foram geradas as estruturas do curso apresentado na seção 6.4. Com esta seção, visa-se esclarecer como as informações definidas pelo usuário através dos módulos descritos na seção anterior são utilizadas na construção de um curso, de acordo com as relações existentes entre as informações das ontologias e o CAM, dispostas no quadro 46. As interfaces não são apresentadas, tão pouco é discutido se um determinado componente instrucional foi criado durante o desenvolvimento deste curso ou reaproveitado, pois isso não altera a estrutura do curso gerado.

Considerando que a ontologia sobre o domínio de estatística já havia sido importada pelo administrador, o processo é iniciado com a definição do título e endereço para disponibilização do curso, que são:

Título: Introdução à Estatística

Endereço: <http://uniplac.net/~castello/estatistica>

Em seguida, é feita a seleção dos conceitos a partir das informações disponibilizadas na ontologia. São eles:

- Estatística;
- Estatística Descritiva;
- Inferência Estatística;
- Medidas Numéricas;
- Medidas de Posição;
- Média;
- Moda;
- Mediana.

Após, as dimensões *técnica* e *pedagógica* do perfil dos alunos que participarão do curso são definidas. Neste exemplo, eles foram enquadrados em duas categorias, nas duas dimensões, como pode ser visto nos quadros 49 e 50.

QUADRO 49 - Perfis técnicos selecionados

Nome Perfil	Características
1 – Doméstico	Tipo de Acesso = Linha Discada SO utilizado = msWindows Browser = internet Explorer Idioma = pt-BR Tempo para realizar o curso= 600
2 – Laboratório	Tipo de Acesso = ADSL SO utilizado = msWindows Browser = internet Explorer Idioma = pt-BR Tempo para realizar o curso= 480

A dimensão pedagógica teve uma influência mais significativa na definição dos objetivos, atividades e objetos de aprendizagem do curso. Isto se justifica porque os alunos com o perfil *Básico* caracterizam-se como aqueles que desenvolvem apenas as atividades obrigatórias para aprender o conteúdo, sem procurar relações com outros assuntos nem atividades complementares que possam reforçar a sua aprendizagem. Desafios propostos

pelo professor não os motivam e, em geral, são vistos apenas como atividades inúteis. Eles preferem apenas ler, a realizar atividades que ponham à prova o seu conhecimento.

QUADRO 50 - Perfis pedagógicos selecionados

Nome Perfil	Características
1 – Básico	Contexto = Escola Dificuldade = Fácil Tipo de Interação = Expositiva Densidade Semântica = Baixa Faixa Etária = 15
2 – Avançado	Contexto = Escola Dificuldade = Média Tipo de Interação = Mista Densidade Semântica = Média Faixa Etária = 15

Já os alunos com perfil pedagógico *Avançado* sentem-se motivados com exercícios e problemas extras passados pelo professor. Eles testam os seus conhecimentos sempre que possível, realizando outras atividades que não as obrigatórias e procurando relações com outros assuntos de seu conhecimento. A diferença entre as características destes alunos reflete-se em seu modelo nos diferentes valores atribuídos às propriedades *Nível de Dificuldade*, *Densidade Semântica* e *Tipo de Interatividade*.

A dimensão técnica, neste caso, diferiu apenas nos atributos *tipo de acesso à Internet* e *tempo para realização do curso*. O segundo atributo serve apenas para comparar se a soma dos tempos destinados a cada objeto de aprendizagem (definido na propriedade *lom-edu:typicallearningtime*) não é superior ao tempo de que o aluno dispõe para realizar o curso. Neste caso, ele não foi relevante, pois a soma dos valores é inferior em todos os perfis. O primeiro atributo foi responsável pela não inserção de algumas atividades nas estruturas destinadas aos alunos com perfil técnico *Doméstico*, devido ao tempo excessivo que eles levariam para acessar os arquivos através de uma conexão à Internet lenta.

Ainda em relação aos diferentes perfis pedagógicos, além de definir os estereótipos descritos no quadro 50 é preciso informar o nível de conhecimento que os alunos de cada perfil possuem sobre os conceitos selecionados e o nível que eles devem atingir após a conclusão do curso. No quadro 51 é feito o cruzamento entre os conceitos e os perfis pedagógicos selecionados para o curso. Nele, é possível perceber que: os alunos não possuem nenhum conhecimento sobre estatística; espera-se que os alunos do perfil básico apenas compreendam os conceitos selecionados; os alunos do perfil avançado, além de compreenderem alguns conceitos, devem ser capazes de aplicar outros como média, mediana e moda.

QUADRO 51 - Perfis pedagógicos x conceitos

Perfil Pedagógico	Conceito	Nível de Conhecimento – Pré-Requisito	Nível de Conhecimento a ser atingido
Básico	Estatística	Nenhum ¹⁷	Compreensão
Básico	Estatística Descritiva	Nenhum	Compreensão
Básico	Inferência Estatística	Nenhum	Compreensão
Básico	Medidas Numéricas	Nenhum	Compreensão
Básico	Medidas Posição	Nenhum	Compreensão
Básico	Média	Nenhum	Compreensão
Básico	Moda	Nenhum	Compreensão
Básico	Mediana	Nenhum	Compreensão
Avançado	Estatística	Nenhum	Compreensão
Avançado	Estatística Descritiva	Nenhum	Compreensão
Avançado	Inferência Estatística	Nenhum	Compreensão
Avançado	Medidas Numéricas	Nenhum	Compreensão
Avançado	Medidas Posição	Nenhum	Aplicação
Avançado	Média	Nenhum	Aplicação
Avançado	Moda	Nenhum	Aplicação
Avançado	Mediana	Nenhum	Aplicação

Com relação às estratégias de aprendizagem, optou-se por utilizar *Instrução Direta* para os alunos com o perfil *Básico* e *Instrução Indireta* para os alunos com perfil *Avançado*. Esta escolha justifica-se porque os alunos com perfil pedagógico *Avançado* são mais motivados e possuem mais condições para controlarem o seu processo de aprendizagem.

Na etapa seguinte, são definidos os objetivos de aprendizagem que devem ser alcançados pelos alunos de cada um dos perfis pedagógicos. No quadro 52 estão os objetivos a serem alcançados pelos alunos com o perfil pedagógico *Básico*, com todas as suas características. Os alunos com o perfil *Avançado* também devem atingir estes mesmos objetivos, além do listado no quadro 53. Não é necessário criar um conjunto de objetivos comum a todos os perfis, para então definir os específicos. Caso seja de interesse do usuário, podem ser criados conjuntos de objetivos completamente distintos, sem nenhuma interseção entre eles.

¹⁷ Os termos definidos na ontologia pedagógica, através das instâncias das classes *BloomCognitiveTaxonomy* e *InstructionalEvents*, que são utilizados para indicar as categorias dos objetivos de aprendizagem, níveis de conhecimento e eventos do processo instrucional foram traduzidos para a Língua Portuguesa, para facilitar o entendimento do exemplo

QUADRO 52 - Objetivos comuns entre os alunos com perfil básico e avançado

Título do Objetivo	Nível de Conhecimento – Pré-Requisito	Nível de Conhecimento a ser atingido	Conceitos
1 - Capacidade para resumir os objetivos da estatística	Nenhum	Compreensão	Estatística Estatística Descritiva Inferência Estatística
2 - Diferenciar dados obtidos com técnicas descritivas e inferenciais	Nenhum	Compreensão	Estatística Estatística Descritiva Inferência Estatística Medidas Numéricas
3 - Fórmulas para calcular medidas de posição	Nenhum	Conhecimento	Medidas de Posição Média Moda Mediana
4 - Compreender para que servem as medidas de posição	Nenhum	Compreensão	Medidas de Posição Média Moda Mediana
5 - Calcular medidas de posição	Compreensão	Compreensão	Medidas de Posição Média Moda Mediana

QUADRO 53 - Objetivo específico para os alunos com perfil avançado

Título do Objetivo	Nível de Conhecimento – Pré-Requisito	Nível de Conhecimento a ser atingido	Conceitos
6 - Resolver problemas utilizando medidas de Posição	Compreensão	Aplicação	Medidas de Posição Média Moda Mediana

Após determinar os objetivos de aprendizagem para os diferentes perfis pedagógicos, deve-se criar ou reutilizar atividades de aprendizagem que possibilitem o alcance dos mesmos. Neste curso de exemplo, como foram selecionados dois perfis pedagógicos e dois técnicos, é possível criar quatro estruturas diferentes para organizar e apresentar o conteúdo. Por isso, podem ser criados quatro tipos diferentes de atividades para cada objetivo, uma para cada combinação de perfil mostrada no quadro 54.

QUADRO 54 - Combinações de perfis possíveis no curso de Introdução à Estatística

Perfil	Perfil Pedagógico	Perfil Didático
1	Básico	Doméstico
2	Básico	Laboratório
3	Avançado	Doméstico
4	Avançado	Laboratório

Nos quadros 55 a 58 são apresentadas as atividades de aprendizagem criadas para levar os alunos dos quatro perfis a atingirem os seus objetivos, juntamente com os LO que

as constituem. Além disso, a numeração ao lado das atividades de aprendizagem indica a sua ordem e hierarquia de apresentação.

Como já foi citado, o perfil pedagógico foi mais preponderante no momento da definição das atividades de aprendizagem, por isso, poucas são utilizadas para os alunos do perfil *Básico* e *Avançado* simultaneamente. Já o perfil técnico não teve uma influência tão forte, sendo várias das atividades utilizadas para o perfil *Doméstico* reaproveitadas para os alunos com perfil *Laboratório*, que possuem o mesmo perfil pedagógico. No quadro 55 é possível perceber que apenas 3 atividades foram propostas para os alunos com perfil pedagógico *Básico* e técnico *Doméstico*.

QUADRO 55 - Atividades de Aprendizagem para o perfil 1 (Básico – Doméstico)

Título da Atividade	Objetivos de Aprendizagem	Sub-Atividades e Objetos de Aprendizagem¹⁸	Nível de Conhecimento Pré-Requisito e Desejado	Eventos Instrucionais
1. Conceitos Iniciais de Estatística	1 e 2	1 - Introdução; 2 - Exemplos de dados estatísticos;	Nenhum Compreensão	- Obtendo a atenção; - Apresentando os objetivos; - Recuperação de Pré-requisitos; - Apresentando o material; - Melhorando a retenção e a transferência;
2. Medidas Numéricas	3, 4 e 5	1 – Para que servem Medidas Numéricas;	Nenhum Compreensão	- Obtendo a atenção; - Apresentando os objetivos;
2.1 Medidas de Posição	3, 4 e 5	1 – Principais med. de posição; 2 – Fórmulas para calcular med. de posição; 3 – Exercícios sobre med. posição;	Nenhum Compreensão	- Apresentando o material; - Avaliando o desempenho;

Para cada atividade estão listados os objetivos de aprendizagem associados, o nível de conhecimento que os alunos devem possuir para acompanhá-la, o nível que devem atingir após o seu término e os eventos instrucionais que ela propicia aos alunos. O nível de conhecimento foi um fator preponderante na escolha das atividades para propiciar o

¹⁸ Os elementos marcados com * representam subatividades, os demais são objetos de aprendizagem.

alcance de cada objetivo. Os eventos instrucionais foram levados em conta, embora a idéia proposta por Gagné não tenha sido respeitada em muitos momentos, por faltarem LO que propiciassem o acontecimento de alguns eventos.

Os LO que constituem as atividades foram retirados de cursos de estatística encontrados na Internet e devidamente descritos com base nas ontologias de domínio, pedagógica e LOM. A utilização de objetos já existentes dificultou bastante a estruturação do curso, já que eles não foram elaborados com as características exatas, de acordo com as ontologias. Isso mostra tanto a falta de embasamento nas teorias instrucionais dos cursos disponíveis atualmente quanto a dificuldade que deve ser enfrentada para que os materiais já existentes na *Web* sejam anotados com base em ontologias, para facilitar a busca e reutilização dos mesmos.

Os LO utilizados nas atividades do perfil 1 são expositivos, respeitando a preferência definida em sua dimensão pedagógica, à exceção dos exercícios que foram inseridos para proporcionar os dois últimos eventos instrucionais propostos por Gagné, sobre *Medidas de Posição*. As atividades são compatíveis com a faixa etária dos alunos e apresentam apenas os conceitos necessários, sem relacioná-los a outros assuntos para seguir a preferência por baixa densidade semântica, também expressa no perfil pedagógico do aluno. Os exemplos e exercícios propostos não possuem alto grau de complexidade, cobrindo apenas os aspectos básicos do assunto. Além disso, todos os conteúdos são escritos em português e podem ser acessados a partir de computadores com qualquer *browser* e sistema operacional e não requerem acesso rápido à Internet.

Para os alunos do perfil 2, cuja variação ocorre apenas na dimensão técnica, as atividades propostas foram basicamente as mesmas, como pode ser visto no quadro 56. Apenas uma atividade de revisão a mais foi proposta, para propiciar que os alunos vivenciem os dois últimos eventos instrucionais sobre todos os conceitos vistos. Esta atividade é composta por um jogo que exige uma alta largura de banda e não foi incluída na estrutura dos alunos do perfil 1, porque eles acessam o curso através de linha telefônica, que é uma forma lenta de conexão à Internet.

Pelos motivos já expostos, as atividades para os perfis 3 e 4, listadas nos quadros 57 e 58, são completamente distintas das previstas para os alunos dos perfis 1 e 2. Elas são compostas por LO que, apesar de abordarem os mesmos conceitos utilizados para os alunos dos perfis 1 e 2, o fazem com maiores detalhes, estabelecendo relações com outros assuntos e, na medida do possível, levando o aluno a interagir, resolvendo exercícios ou através de jogos educativos.

QUADRO 56 - Atividades de Aprendizagem para o perfil 2 (Básico – Laboratório)

Título da Atividade	Objetivos de Aprendizagem	Sub-Atividades e Objetos de Aprendizagem	Nível de Conhecimento Pré-Requisito e Desejado	Eventos Instrucionais
1. Conceitos Iniciais	1 e 2	1 - Introdução; 2 - Exemplos de dados estatísticos;	Nenhum Compreensão	- Obtendo a atenção; - Apresentando os objetivos; - Recuperação de Pré-requisitos; - Apresentando o material; - Melhorando a retenção e a transferência;
2. Medidas Numéricas	3, 4 e 5	1 – Para que servem Medidas Numéricas;	Nenhum Compreensão	- Obtendo a atenção; - Apresentando os objetivos;
2.1. Medidas de Posição	3, 4 e 5	1 – Principais med. de posição; 2 – Fórmulas de Cálculo; 3 – Exercícios;	Nenhum Compreensão	- Apresentando os objetivos; - Avaliando o desempenho;
3. Revisão sobre conceitos iniciais de estatística	1 a 5	1 – Jogo “Glória”	Compreensão Aplicação	- Avaliando o desempenho; - Melhorando a retenção e a transferência;

QUADRO 57 - Atividades de Aprendizagem para o perfil 3 (Avançado – Doméstico)

Título da Atividade	Objetivos de Aprendizagem	Sub-Atividades e Objetos de Aprendizagem	Nível de Conhecimento Pré-Requisito e Desejado	Eventos Instrucionais
1. Conceitos Iniciais de Estatística Detalhados	1 e 2	1 – Introdução e Método Est; 2 - Exemplos de dados estatísticos;	Nenhum Compreensão	- Obtendo a atenção; - Apresentando os objetivos; - Recuperação de Pré-requisitos; - Apresentando o material; - Melhorando a retenção e a transferência;
2. Medidas de Posição Detalhadas	4	1- Tipos de Medidas de Posição *1 – Média *2 – Mediana *3 - Moda	Nenhum Compreensão	- Apresentando o material;
2.1. Média	3 e 4	1 - Cálculo da Média	Nenhum Compreensão	- Apresentando o material;
2.2 Mediana	3 e 4	1 – Cálculo da Mediana	Nenhum Compreensão	- Apresentando o material;

2.3. Moda	3 e 4	1 – Cálculo da Moda	Nenhum Compreensão	- Apresentando o material;
3. Aplicação de medidas de posição	5 e 6	1 – Exercícios sobre Média, Mediana e Moda 2 – Resolvendo problemas com medidas de posição	Compreensão Aplicação	- Avaliando o desempenho; - Melhorando a retenção e a transferência;

QUADRO 58 - Atividades de Aprendizagem para o perfil 4 (Avançado – Laboratório)

Título da Atividade	Objetivos de Aprendizagem	Sub-Atividades e Objetos de Aprendizagem	Nível de Conhecimento Pré-Requisito e Desejado	Eventos Instrucionais
1. Conceitos Iniciais Detalhados	1 e 2	1 – Introdução e Método Est; 2 - Exemplos de dados estatísticos;	Nenhum Compreensão	- Obtendo a atenção; - Apresentando os objetivos; - Recuperação de Pré-requisitos; - Apresentando o material; - Melhorando retenção e transferência;
1.1. Jogo sobre conceitos iniciais	1	1 – Jogo Estatística Trivial	Compreensão Aplicação	- Avaliando o desempenho - Melhorando retenção e transferência;
2. Medidas de Posição Detalhadas	4	1 - Tipos de Medidas de Posição *1 – Média *2 – Mediana *3 - Moda	Nenhum Compreensão	- Apresentando o material;
2.1 Média	3 e 4	1 - Cálculo da Média	Nenhum Compreensão	- Apresentando o material;
2.2 Mediana	3 e 4	1 – Cálculo da Mediana	Nenhum Compreensão	- Apresentando o material;
2.3 Moda	3 e 4	1 – Cálculo da Moda	Nenhum Compreensão	- Apresentando o material;
3. Aplicação de medidas de posição	5 e 6	1 – Exercícios sobre Média, Mediana e Moda 2 – Resolvendo problemas com méd. de posição	Compreensão Aplicação	- Avaliando o desempenho; - Melhorando a retenção e a transferência;
4. Revisão sobre conceitos iniciais de estatística	1 a 6	1 – Jogo “Glória”	Compreensão Aplicação	- Avaliando o desempenho - Melhorando a retenção e a transferência

Entre os perfis 3 e 4, as diferenças são pequenas, pois ambos possuem a mesma dimensão pedagógica. Mais uma vez, apenas a inclusão de jogos extras para os alunos com perfil técnico *Laboratório* diferenciou as atividades propostas.

6.4 Geração e Disponibilização do Curso

Nesta seção é apresentado o curso gerado através da explanação do arquivo *imsmanifest* que contém a sua estrutura e da sua disponibilização em um LMS compatível com o padrão SCORM. Para testar se o curso gerado é realmente compatível com as definições do padrão SCORM, utilizou-se um LMS desenvolvido pela própria *Advanced Distributed Learning* (ADL), organização responsável pelo padrão SCORM. Apesar de simples, este LMS implementa os requisitos básicos necessários para executar um curso que adota o referido padrão, executando todas as funções definidas em ADL (2004c). A interface principal deste LMS é apresentada na figura 67.

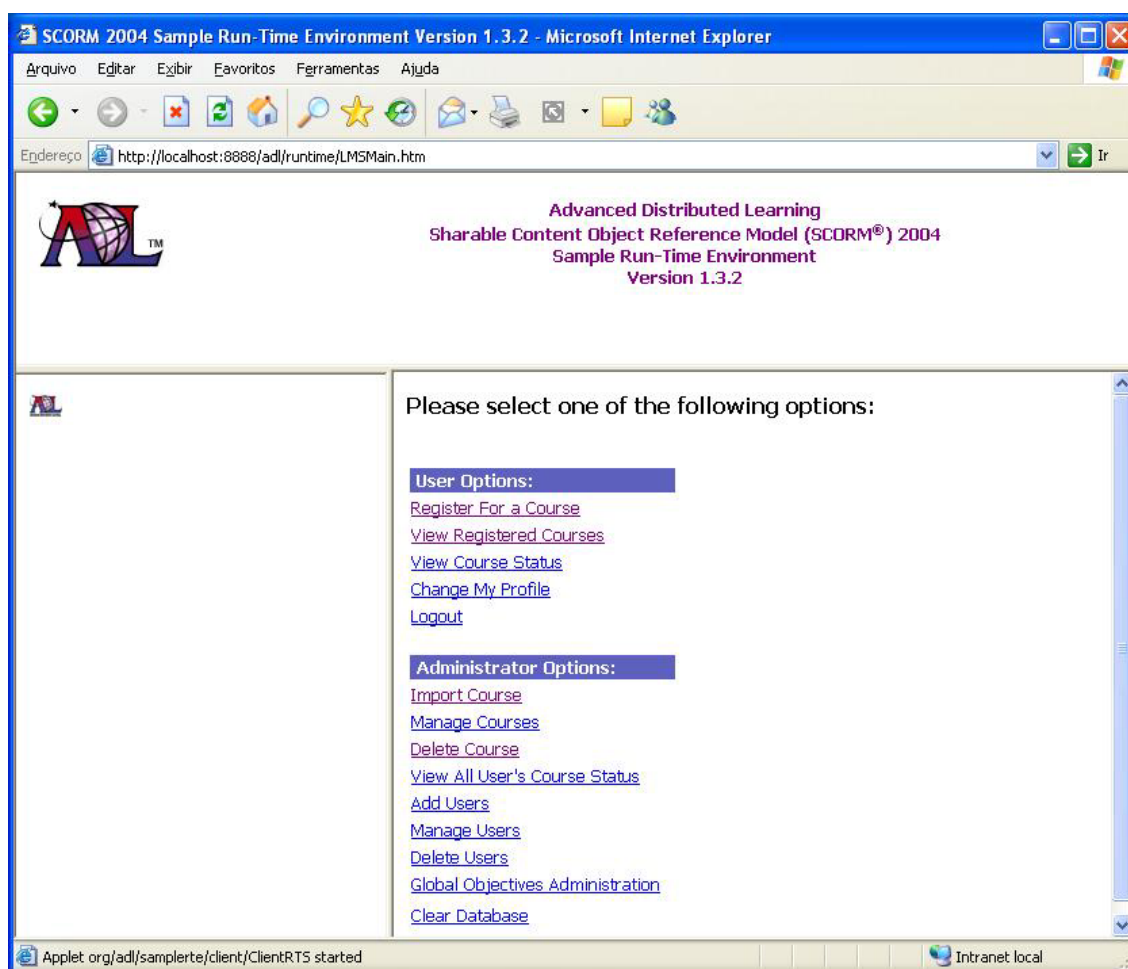


FIGURA 67 - Interface principal do LMS compatível com o padrão SCORM

A partir desta interface, o administrador do sistema acessa várias funcionalidades

relacionadas à gerência de cursos e usuários. Para realizar os testes foram utilizados apenas os recursos para:

- importação do curso – importa um curso e verifica, simultaneamente, se ele é compatível com o Modelo de Agregação de Conteúdo (CAM) do padrão SCORM;
- registro no curso – inscreve um aluno em um curso;
- visualização do curso – possibilita que um aluno acesse o curso.

De acordo com o CAM, um curso é constituído pelo arquivo *imsmanifest*, que contém as definições da estrutura do curso e as referências para os LO. O quadro 59 apresenta os componentes principais do *imsmanifest* gerado pelo Atlas, com o curso criado a partir das informações apresentadas na seção 6.3. Este quadro serve como um sumário do que o arquivo contém. Os conteúdos internos de seus componentes, representados pelas linhas preenchidas com pontos, são detalhados posteriormente em quadros separados.

Entre as linhas 1 e 6, está o cabeçalho do arquivo. Ele designa que este é um arquivo XML compatível com o padrão SCORM e indica os *namespaces* utilizados para definir os elementos existentes no arquivo. Os *namespaces* apresentados, e também os omitidos, possuem as definições dos elementos que podem ser inseridos em um arquivo baseado no CAM. A única exceção é o *namespace* associado na linha 4, desenvolvido com o intuito de registrar informações sobre os objetivos de aprendizagem de cada atividade, o que não é previsto no padrão SCORM.

Através deste *namespace*, o Atlas pode incluir o(s) objetivo(s) de aprendizagem associado(s) a cada atividade do curso, através da *tag metadata*. Como pode ser visto no quadro 60, são inseridos o título e a referência para a instância na ontologia de domínio que define o objetivo de aprendizagem, caso alguém deseje obter maiores informações sobre ele.

Embora os LMS compatíveis com o padrão SCORM não reconheçam este elemento e não o utilizem na apresentação do curso, ele pode contribuir com mecanismos de busca, pois referencia a ontologia de domínio onde se encontra uma definição completa dos dados do objetivo de aprendizagem.

QUADRO 59 - Componentes principais do arquivo *imsmanifest* gerado pelo Atlas

```

1. <?xml version="1.0" encoding="UTF-8"?>
2. <manifest xmlns="http://www.imsglobal.org/xsd/imsdp_v1p1"
3.   xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_v1p3"
4.   xmlns:obj="http://localhost/concepts/objetivos"
5.   ....
6.   identifier="Exemplo" version = "1.3">
7.   <metadata>
8.     <schema>ADL SCORM</schema>
9.     <schemaversion>CAM 1.3</schemaversion>
10.  </metadata>
11.  <organizations default="Organizacao1">
12.    <organization identifier="Organizacao1">
13.      .....
14.    </organization>
15.    <organization identifier="Organizacao2">
16.      .....
17.    </organization>
18.    <organization identifier="Organizacao3">
19.      .....
20.    </organization>
21.    <organization identifier="Organizacao4">
22.      .....
23.    </organization>
24.  </organizations>
25.  <resources>
26.    <resource>...</resource>
27.    .....
28.  </resources>
29.</manifest>

```

Após o cabeçalho, estão as definições dos metadados, seguidas pelas duas partes principais do arquivo, demarcadas pelos seguintes elementos:

- *organizations* (linhas 11 a 24) – que engloba as diferentes estruturas para apresentação do conteúdo que o curso possui;
- *resources* (linhas 25 a 28) – conjunto de LO que são apresentados aos alunos durante a execução das atividades de aprendizagem;

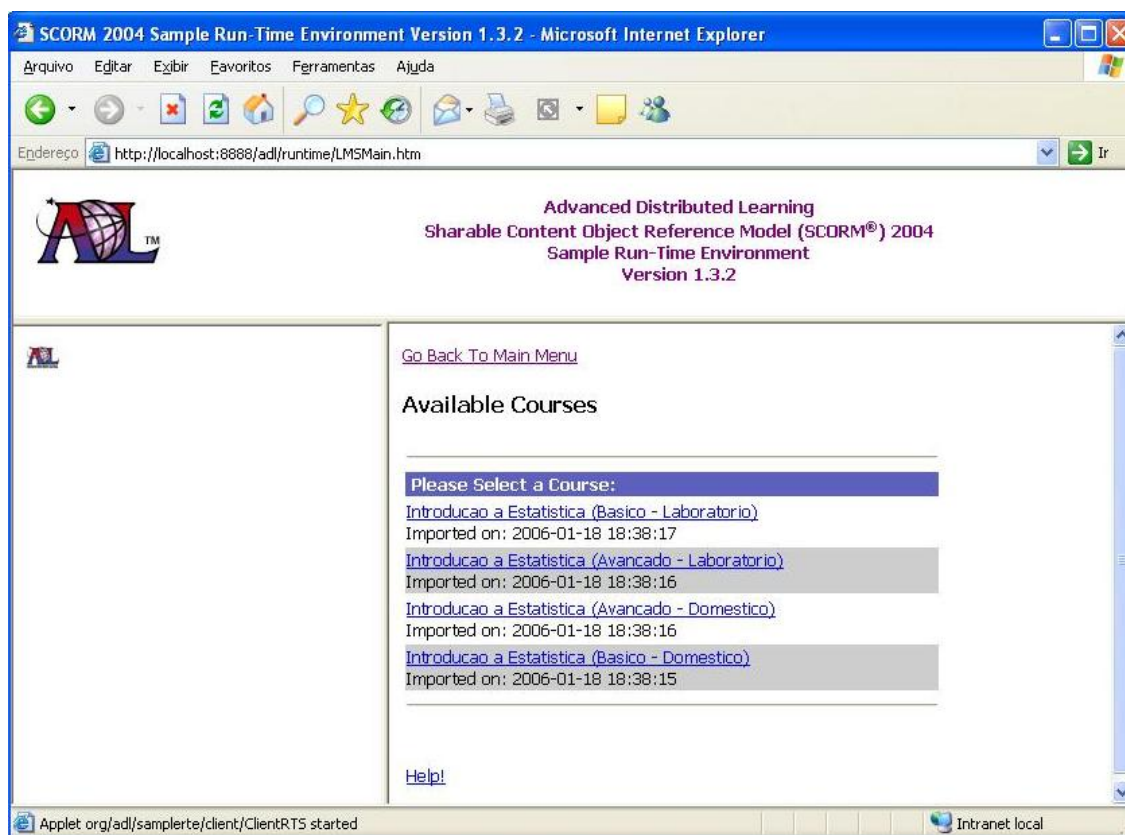
QUADRO 60 - Objetivos de aprendizagem de uma atividade

```

1. <metadata>
2. <obj:LearningObjectives>
3. <obj:LearningObjective>
4. <obj:Titulo>
5. Capacidade para resumir os objetivos da estatística
6. </obj:Titulo>
7. <obj:Indentificador>
8. http://localhost/ontologias/concepts.owl#OBCurso11
9. </obj:Indentificador>
10. </obj:LearningObjective>
11. </obj:LearningObjectives>
12. </metadata>

```

Cada elemento do tipo *organization* define uma estrutura diferente para as atividades de aprendizagem. No quadro 59 existem quatro elementos deste tipo, um para cada tipo de perfil de aluno, que podem ser vistos na figura 68, após a sua importação pelo LMS.

**FIGURA 68** - Interface com os cursos disponíveis após a importação do curso gerado

Embora o Atlas tenha gerado um único curso com quatro estruturas diferentes para apresentar o conteúdo, o LMS utilizado interpreta como quatro cursos diferentes. Isso

ocorre porque ele não possui mecanismos capazes de identificar em qual perfil o aluno se enquadra. Assim, fica sob a responsabilidade do professor ou do próprio aluno identificar qual dos cursos deve acessar. Esta é uma das desvantagens de utilizar um padrão já existente, com LMS já implementados. Entretanto, isto pode ser superado com a inserção de LO que auxiliem a definir o perfil do aluno previamente e possam direcioná-lo à estrutura mais adequada.

O quadro 61 apresenta o código da organização 1, que corresponde às atividades de aprendizagem selecionadas para os alunos com o perfil pedagógico *Básico* e o perfil técnico *Doméstico*, listadas no quadro 55.

QUADRO 61 - Estrutura das atividades de aprendizagem para o perfil 1.

```

1. <organization identifier="Organizacao1">
2.   <title>Introducao a Estatistica (Basico - Domestico)</title>
3.   <item identifier="Atv111">
4.     <title>Conceitos Iniciais de Estatistica</title>
5.     <item identifier="CAtv111" identifierref="RECAtv11">
6.       <title>Conteudo sobre Conceitos Iniciais de Estatistica</title>
7.     </item>
8.   </item>
9.   <item identifier="Atv121">
10.    <title>Medidas Numericas</title>
11.    <item identifier="CAtv121" identifierref="RECAtv12">
12.      <title>Conteudo sobre Medidas Numericas</title>
13.    </item>
14.    <item identifier="Atv131">
15.      <title>Medidas de Posicao</title>
16.      <item identifier="CAtv131" identifierref="RECAtv13">
17.        <title>Conteudo sobre Medidas de Posicao</title>
18.      </item>
19.      <imsss:sequencing>
20.        <imsss:controlMode flow="true" choice="false"/>
21.      </imsss:sequencing>
22.    </item>
23.    <imsss:sequencing>
24.      <imsss:controlMode flow="true" choice="false"/>
25.    </imsss:sequencing>
26.  </item>
27.</organization>

```

Nas linhas 1 e 2 deste quadro, estão definidos o identificador e o título da estrutura. Para facilitar a visualização no LMS, o título de cada estrutura é formado pelo título do curso concatenado à descrição do perfil do aluno associado àquela estrutura. A partir da linha 3, estão dispostas as atividades de aprendizagem, com seus identificadores, títulos e subatividades, de acordo com a estrutura definida para os alunos com este perfil.

Foi necessário criar um artifício para representar as atividades de aprendizagem existentes na ontologia de domínio através de elementos *item*. Isto foi preciso porque o CAM não permite que uma mesma atividade de aprendizagem (elemento *item*) possua subatividades e esteja associada a LO simultaneamente, o que é permitido na ontologia de domínio. Por isso, foi associada uma subatividade a cada item criado, que referencia seus LO (linhas 5 a 7). Assim, como a atividade de aprendizagem principal não vincula nenhum LO diretamente, é possível associar a ela todas as suas subatividades definidas na ontologia. Para representar que uma atividade é subatividade de outra, basta adicionar o elemento *<item>* (linha 14) que a inicia, antes do elemento *</item>* (linha 26) que encerra a atividade principal.

Outro elemento importante mostrado no quadro 61 é o responsável por indicar as possíveis formas de navegação (linhas 19 e 21 ou 23 e 25). Os atributos *flow* e *choice* representam duas características importantes. O primeiro, quando recebe o valor *true*, determina que o LMS deve oferecer algum mecanismo que oriente o aluno a realizar as atividades em seqüência. Já o parâmetro *choice*, quando definido como *true*, indica que o usuário pode acessar outras atividades que não a imediatamente anterior ou posterior à atividade em que ele se encontra.

O valor destes atributos no curso gerado pelo Atlas depende da estratégia instrucional escolhida para o perfil pedagógico, como é apresentado no quadro 62.

QUADRO 62 - Estratégias instrucionais e forma de navegação

Estratégia Instrucional	Flow	Choice	Explicação
Instrução Direta	true	false	Neste caso, o sistema deve guiar o aluno e não permitir que ele acesse atividades fora de ordem
Instrução Indireta	true	true	Neste caso, o sistema deve guiar o aluno, mas pode permitir que ele acesse atividades fora de ordem
Instrução Interativa, Aprendizagem Experimental e Estudo Independente	false	true	Nestes casos, o sistema não deve guiar o aluno e deve permitir que ele acesse as atividades na ordem que desejar

O LMS utilizado para testar o sistema, por ser apenas uma ferramenta para teste e sem fins comerciais, não possui muitos recursos de navegação diferentes. Ele guia o aluno durante as atividades de aprendizagem através dos botões *Previous* e *Continue*, como pode ser visto na figura 69, que ilustra a execução da estrutura definida para os alunos do perfil 1 (pedagógico=*Básico* e técnico=*Doméstico*).

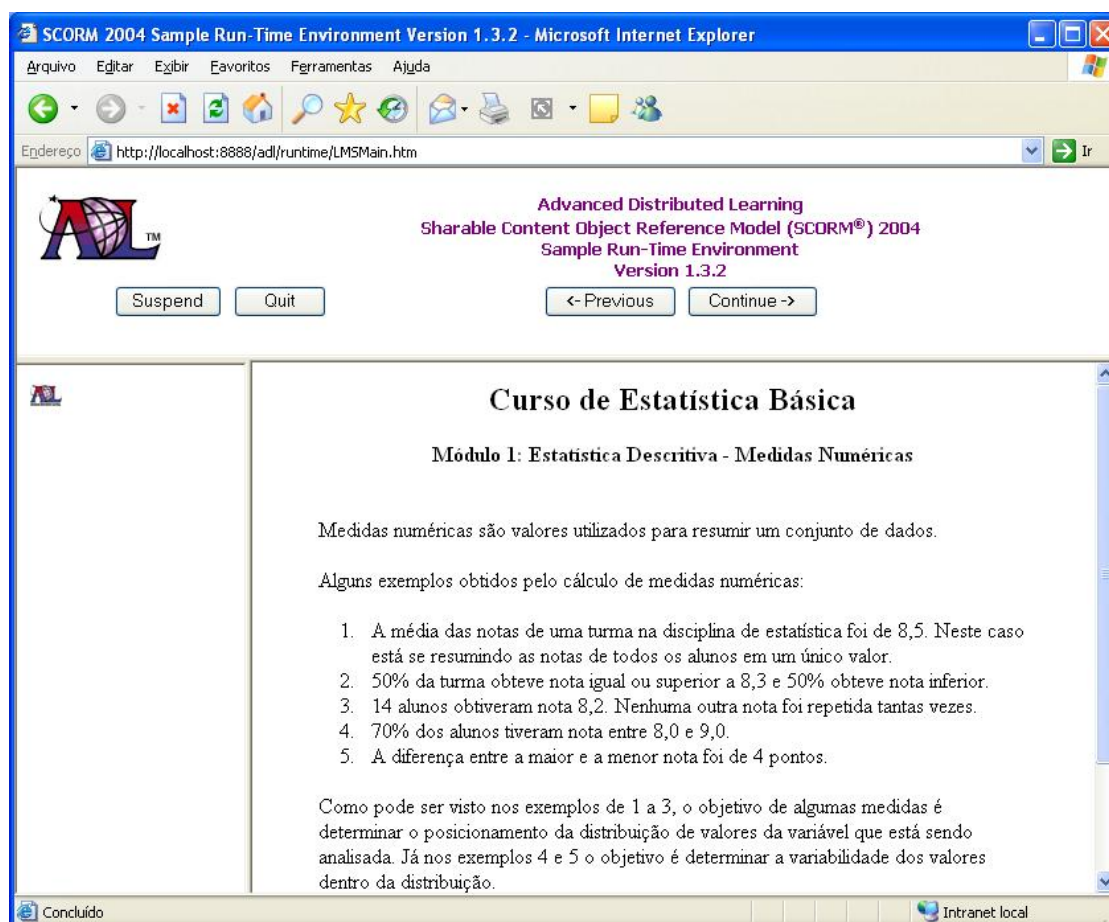


FIGURA 69 - Apresentação do curso gerado para o perfil 1

Durante a criação do curso, determinou-se que a estratégia a ser utilizada para os alunos com perfil pedagógico *Básico* é *Instrução Direta*. Por isso, além de guiar o aluno através das atividades, o LMS não permite que ele acesse outras atividades que não a imediatamente anterior ou posterior à atividade em que ele se encontra, ocultando o menu com as demais atividades.

Já para os alunos com perfil pedagógico *Avançado*, foi designada a estratégia de *Instrução Indireta*. Esta, apesar de guiar o aluno, permite que ele navegue pelas atividades em qualquer seqüência. A figura 70 mostra a execução do curso para os alunos com perfil 4. Nela, podem ser observados os botões *Previous* e *Continue* e o menu de atividades à esquerda, a partir do qual o aluno pode acessar qualquer atividade em qualquer tempo.

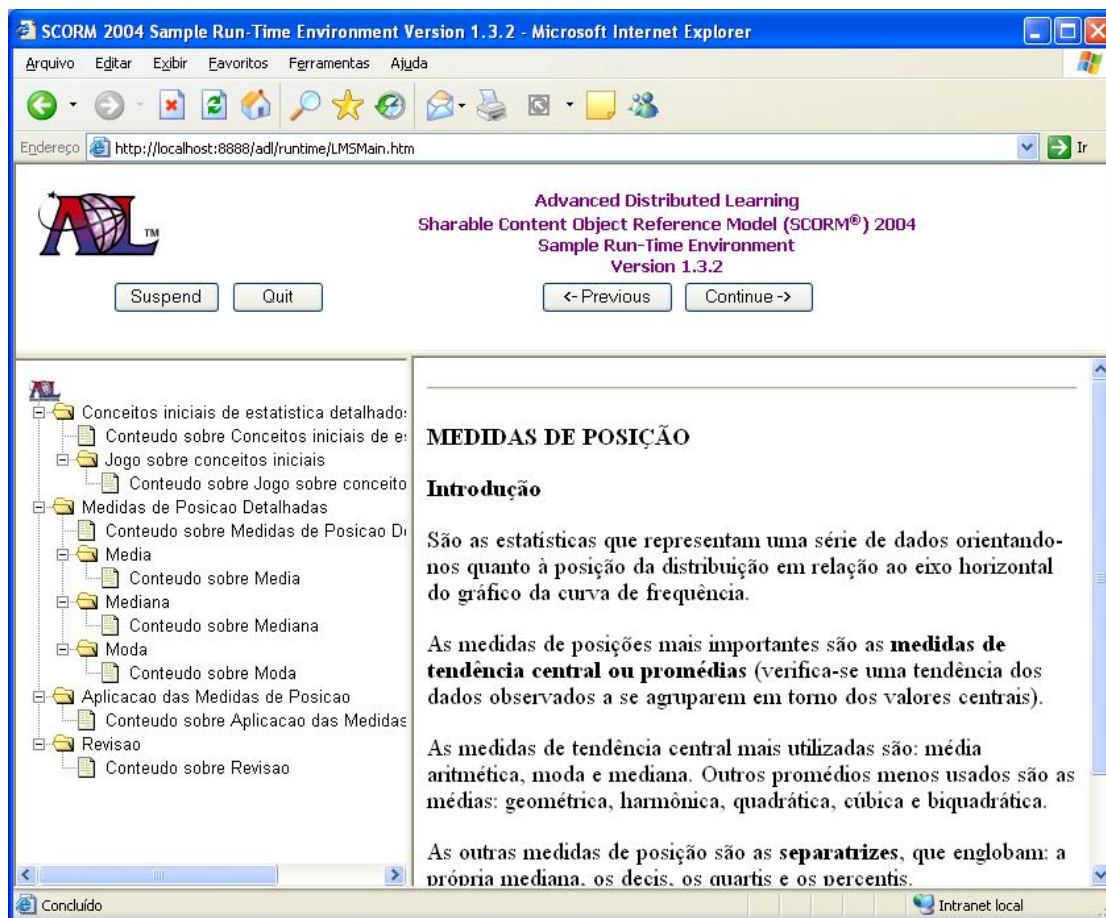


FIGURA 70 - Apresentação do curso gerado para o perfil 4

Com o intuito de mostrar as diferenças entre as formas de navegação de acordo com a estratégia instrucional escolhida, foi construído um curso com as mesmas atividades definidas para o perfil 4, porém, com a estratégia de *Estudo Independente*. A figura 71 apresenta a execução do curso gerado, onde é possível ver que o LMS apresenta o menu para o aluno acessar as atividades, mas não apresenta os botões que indicam a ordem de execução das mesmas.

O segundo componente de um arquivo *imsmanifest*, de acordo com a estrutura exposta no quadro 59, é composto pelos recursos, que nada mais são que os conjuntos de LO apresentados durante a execução das atividades de aprendizagem. O quadro 63 mostra dois dos recursos disponibilizados no curso gerado. Nas linhas 1 e 6 estão os elementos que iniciam os recursos e definem seu identificador, seu tipo e qual arquivo deve ser mostrado primeiro, dentre os agrupados no recurso através do elemento *file*.

Ao selecionar as atividades de aprendizagem (figura 54), o usuário pode escolher se ele deseja copiar ou apenas referenciar os LO em seu curso. No exemplo do quadro 63, o usuário optou por copiar, por isso, apenas o nome do arquivo é definido, já que ele foi copiado e é importado juntamente com o *imsmanifest*. O quadro 64 apresenta os mesmos

recursos, ao serem gerados pelo Atlas quando o usuário optou apenas por referenciar os LO. Neste quadro, pode-se observar que os LO são referenciados através de seu endereço completo, já que eles devem ser acessados em sua localização original.

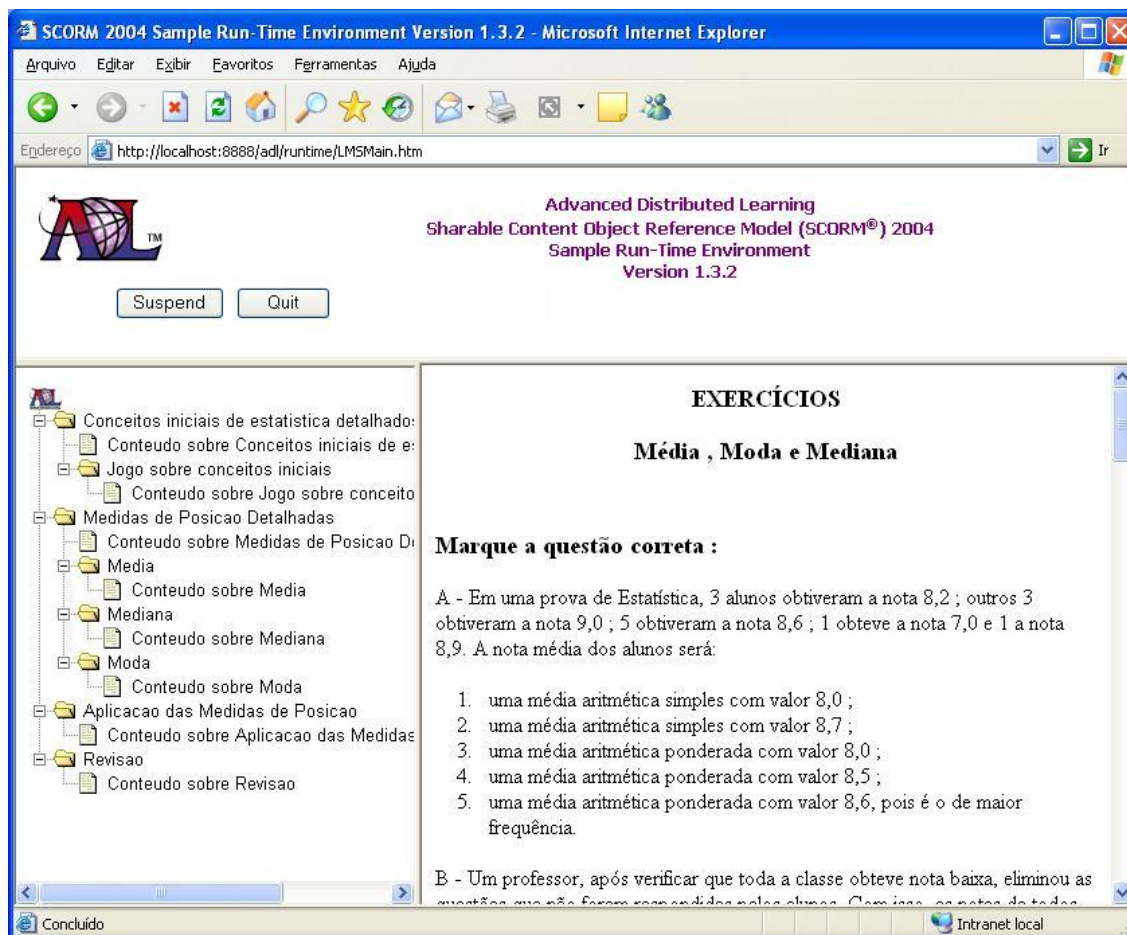


FIGURA 71 - Apresentação do curso gerado para o perfil 4 com estratégia de estudo independente

QUADRO 63 - Definição dos recursos do curso gerado com LO copiados

```

1. <resource identifier="RECAtv11" adlcp:scormType="sco"
2.           type="webcontent" href="intro.html">
3.   <file href="intro.html"/>
4.   <file href="exemplos.html"/>
5. </resource>
6. <resource identifier="RECAtv27" adlcp:scormType="sco"
7.           type="webcontent" href="exercicios_mmm.html">
8.   <file href="exercicios_mmm.html"/>
9.   <file href="problemas_mmm.html"/>
10. </resource>

```

QUADRO 64 - Definição dos recursos do curso gerado com LO referenciados

```

1. <resource identifier="RECAtv11" adlcp:scormType="sco"
2.   type="webcontent" href="http://localhost/estatistica/intro.html">
3.     <file href="http://localhost/estatistica/intro.html"/>
4.     <file href="http://localhost/estatistica/exemplos.html"/>
5. </resource>
6. <resource identifier="RECAtv27" adlcp:scormType="sco"
7. type="webcontent" href="http://uniplac.net/est/exercicios_mmm.html">
8.   <file href="http://uniplac.net/est/exercicios_mmm.html"/>
9.   <file href="http://uniplac.net/est/problemas_mmm.html"/>
10. </resource>

```

6.5 Comparação dos Requisitos com a Arquitetura Desenvolvida

Nesta seção são resgatados os principais problemas da EBW identificados e os requisitos da arquitetura desenvolvida que objetivam resolvê-los, para que eles sejam comparados aos resultados gerados pelo sistema.

Os primeiros problemas listados no capítulo cinco, que se referem à dimensão computacional, podem ser resumidos na dificuldade para reutilizar componentes instrucionais. Os requisitos propostos para minimizar tais problemas estabelecem que:

- Deve ser possível utilizar recursos da *Web Semântica* para aumentar a capacidade de reutilização dos componentes usados na construção de cursos para a EBW;
- Devem ser criadas terminologias padrões para descrever domínios do conhecimento, para que pessoas compartilhem e reutilizem informações durante a construção dos cursos;
- Além de reutilizar qualquer componente do processo instrucional (LO isolados, atividades de aprendizagem completas ou objetivos de aprendizagem), deve-se tornar possível a construção ou descrição de tais componentes.

Todos estes requisitos são atendidos através da ontologia de domínio. Nela é possível representar os conceitos de um domínio do conhecimento para que as pessoas utilizem uma terminologia padrão para referenciar os assuntos ensinados. Além disso, esta ontologia propicia a reutilização de componentes instrucionais mais abrangentes, como objetivos e atividades de aprendizagem, além dos LO, quando os termos padrões são utilizados para descrever os assuntos abordados por eles.

Outros requisitos ainda relativos à dimensão computacional são:

- As ontologias devem ser expressas em OWL, para facilitar a interoperabilidade com outros sistemas;
- Os cursos gerados devem ser representados através de um padrão de metadados para componentes educacionais, para que eles possam ser executados em qualquer LMS que suporte o padrão escolhido.

O primeiro requisito foi atendido ao utilizar a ferramenta Protégé para construir as ontologias e pela criação de um módulo que as importam no sistema Atlas. Como é utilizada uma linguagem com sintaxe aberta e bem documentada, é possível que outras pessoas construam sistemas baseados nas ontologias, completamente diferentes do Atlas. Mesmo assim, como os dois sistemas utilizam a mesma linguagem para representar as instâncias criadas, é possível reutilizar em um sistema um objetivo, atividade ou objeto de aprendizagem criado no outro.

O segundo requisito foi implementado através do mapeamento das instâncias da ontologia para elementos definidos no CAM, do padrão SCORM. Desta maneira, os cursos gerados podem ser acessados por um número consideravelmente maior de pessoas, do que se precisasse de um sistema proprietário para a sua disponibilização. Além disso, a ontologia de domínio foi construída de maneira independente do padrão SCORM, como pôde ser visto no fato dela considerar objetivos de aprendizagem que não são tratados neste padrão. Por isso, é possível elaborar mapeamentos para outros padrões de metadados, como o IMS *Learning Design*, e implementar ferramentas que utilizem as mesmas ontologias e reutilizem as mesmas informações, mas gerem cursos em outros formatos.

Os problemas educacionais podem ser resumidos na falta de adaptação do conteúdo às características e necessidades dos alunos e na falta de fundamentação dos cursos nas teorias instrucionais e da aprendizagem. Os requisitos que visam à solução destes problemas são:

- A arquitetura deve possuir uma estrutura que possa ser reutilizada para outros domínios, sem afetar os demais módulos da arquitetura.
- Ela deve auxiliar a estruturar as atividades instrucionais, com base no conhecimento existente nas ciências instrucionais e de aprendizagem.
- Devem ser definidas informações que descrevam as características dos alunos para que se possa auxiliar na adaptação dos conteúdos.

As ontologias pedagógica e do estudante constituem a base necessária para atender a estes requisitos. As informações nelas disponibilizadas definem aspectos sobre o processo instrucional de maneira genérica e podem ser utilizadas na construção de cursos sobre qualquer assunto definido na ontologia de domínio. Além disso, elas podem ser utilizadas para auxiliar as pessoas que constroem os cursos, durante a definição das atividades a serem apresentadas aos diferentes tipos de alunos, de acordo com as informações armazenadas em seu perfil na ontologia do estudante.

A arquitetura não prevê como estas informações devem ser usadas na adaptação de um curso, porque não foi implementado um sistema para realizar tal tarefa. Conforme citado, optou-se por utilizar um padrão já existente para tornar mais fácil o acesso aos cursos gerados. Mesmo assim, as informações descritas nas ontologias pedagógica e do estudante se constituem em uma base que pode ser usada pelos LMS para propiciar a adaptação. Também é possível, futuramente, construir sistemas específicos com mecanismos adaptativos baseados nestas informações, ou até mesmo propor expansões nos LMS que suportam os padrões existentes, para que elas sejam melhores utilizadas.

6.6 Considerações

Neste capítulo, pôde-se observar o sistema construído com base nas ontologias apresentadas no capítulo 5. Sua implementação ajudou a verificar se as informações dispostas nas ontologias estavam coerentes e contribuíam para que os objetivos propostos fossem alcançados. Por isso, algumas alterações nas ontologias precisaram ser feitas, enquanto o sistema era implementado.

As regras para representar as informações das ontologias no padrão SCORM, após alguns ajustes, também se mostraram corretas. Isto pôde ser visto na seção 6.4, onde os componentes instrucionais apresentados na seção 6.3 foram transformados em um curso que foi corretamente executado por um LMS que adota o referido padrão.

Outro aspecto destacado neste capítulo, é a flexibilidade proporcionada pelas ontologias às pessoas que pretendem construir sistemas baseados nas mesmas. Em vários momentos, explicou-se como o Atlas age mediante uma situação, mas mostrou-se que é possível construir sistemas que se comportem de maneira diferente na mesma situação.

Por fim, ressalta-se que as informações aqui apresentadas foram fundamentais para a comparação com os requisitos definidos para a arquitetura, para que ela pudesse ser validada.

7. CONCLUSÕES E TRABALHOS FUTUROS

7.1 Conclusões

Nesta tese foram apresentados os principais problemas enfrentados pela comunidade envolvida com a EBW. Ressaltou-se a divisão existente entre os pesquisadores da área em dois grupos: um interessado em melhorar a aprendizagem dos alunos e outro em reduzir os custos e o esforço necessários para construir sistemas e cursos para a EBW. Com base nisso, foram propostos os objetivos que, resumidamente, focam a aplicação das tecnologias da *Web Semântica* para promover melhorias na EBW, a partir da fusão das técnicas oriundas dos dois grupos.

O trabalho inicial, realizado através de uma extensa revisão bibliográfica sobre o assunto, possibilitou identificar as limitações relacionadas às tecnologias desenvolvidas pelos grupos citados, como STI, SHA, LMS e LO. Esta etapa, completada com um estudo empírico sobre trabalhos correlatos, propiciou a elaboração de alguns objetivos a serem atingidos que, inicialmente, focavam diversos aspectos relacionados à EBW e necessitavam de recursos tecnológicos até então desconhecidos pelo autor.

A partir deste momento, foi iniciado o estudo sobre *Web Semântica* para compreender a sua estrutura e conhecer as tecnologias já desenvolvidas. Esta etapa foi fundamental para delimitar a pesquisa a ser desenvolvida e focar seus objetivos, até então um pouco dispersos, pois ficou mais claro o que é e o que “ainda” não é possível fazer com a *Web Semântica*.

A principal contribuição desta tese reside nas ontologias desenvolvidas que possibilitam a construção de cursos adaptativos completos ou de suas partes, de forma mais rápida e econômica, através da reutilização de componentes instrucionais devidamente descritos pelos metadados definidos nas ontologias. Além disso, estas mesmas ontologias propiciam um suporte baseado nas teorias instrucionais e da aprendizagem, independente do domínio de conhecimento, que auxilia as pessoas que elaboram os cursos a definirem seus objetivos, conteúdos e estrutura de apresentação.

Alguns pontos positivos podem ser destacados, tais como: a estrutura da ontologia de domínio permite a construção de mapas conceituais em qualquer domínio do conhecimento, o que não altera a sua relação com as demais ontologias nem com os sistemas que utilizam-na; esta mesma ontologia possibilita a reutilização de componentes instrucionais mais complexos, como objetivos e atividades de aprendizagem, e não apenas de objetos de aprendizagem isolados; as informações constantes na ontologia pedagógica auxiliam os usuários na construção de cursos, também independentemente do domínio do conhecimento, o que é uma das grandes limitações das ferramentas atuais; a estrutura criada para a ontologia pedagógica permite facilmente a inclusão de informações oriundas de outras teorias instrucionais e da aprendizagem, para contemplar a diversidade de correntes existentes na área educacional; as relações existentes entre as ontologias de domínio, pedagógica e do estudante facilitam a construção de componentes instrucionais mais adequados ao perfil de cada estudante; a independência entre as ontologias e o padrão utilizado para representar o curso gerado permite que as mesmas informações sejam utilizadas para criar cursos em diferentes padrões; ao escolher o padrão SCORM para representar os cursos gerados pelo Atlas, permite-se que eles possam ser acessados por um número maior de pessoas, do que se fosse utilizado um padrão próprio; sistemas completamente distintos poderão interagir e reutilizar informações entre eles, por utilizarem a mesma estrutura (definida na ontologia de domínio) e uma linguagem comum (OWL) para armazenarem suas informações.

Contudo, alguns pontos negativos surgem com a arquitetura proposta: a flexibilidade para representar diversos domínios tolhe algumas possibilidades de auxílio pedagógico e adaptação dos cursos gerados; a partir do momento em que outras teorias forem incorporadas na ontologia pedagógica, pode ocorrer que componentes instrucionais sejam descritos de diferentes maneiras e não sejam reutilizados porque não existe uma estrutura capaz de relacionar as informações das diferentes teorias dispostas na referida ontologia; como a arquitetura desenvolvida não possui um repositório para armazenar LO e guarda apenas a sua referência, é possível que com o passar do tempo os sistemas que utilizam suas informações sugiram LO que não podem ser mais encontrados a partir da referência existente na ontologia; por representar as informações de forma independente do padrão utilizado para representar o curso gerado e por fornecer acesso aos mesmos através de LMS já existentes, surgem muitas limitações na adaptação em tempo real do conteúdo.

Com relação aos objetivos propostos, pode-se concluir que eles foram alcançados através das ontologias desenvolvidas e do sistema implementado que faz uso de suas

informações. Finalizando, destaca-se que ainda é longo o caminho a ser trilhado pelos educadores, tanto os que utilizam métodos tradicionais quanto os que fazem uso da tecnologia, até que se tenha certeza sobre quais são os métodos mais eficientes para levar os alunos a aprender. Esta dúvida reflete-se em todas as pesquisas relacionadas à EBW, que ainda precisa conviver com a desconfiança de muitos na eficácia de seus métodos. Do ponto de vista computacional, os problemas são mais simples, pois as possibilidades criadas pela *Web Semântica* e apresentadas neste trabalho vêm se consolidando como uma nova forma de conceber os materiais utilizados durante o processo instrucional.

7.2 Trabalhos Futuros

Por serem áreas de pesquisa relativamente novas, a EBW e a *Web Semântica* possuem muitos pontos a serem estudados individualmente. Entretanto, os apresentados a seguir concentram-se nas possibilidades geradas pela união das duas áreas.

Como a estrutura da ontologia de domínio é bastante generalista, foi criado um mapa conceitual na área de estatística apenas para validar a arquitetura proposta. Mas, para que ela realmente seja útil, se faz necessário um esforço das comunidades envolvidas nos mais diversos domínios do conhecimento com o intuito de construir mapas conceituais que representem um consenso entre os interessados.

O estudo de outras teorias instrucionais e da aprendizagem, e a sua conseqüente formalização através de ontologias, pode tornar os sistemas mais eficientes para orientar os usuários na construção de cursos ou de seus componentes, facilitando a aprendizagem dos alunos. Além disso, é importante expandir a estrutura desta ontologia para que ela seja capaz de mapear as relações entre as informações de diferentes teorias, que podem ser utilizadas para descrever um mesmo elemento.

A ontologia do estudante contempla apenas as informações que descrevem os estudantes, sem relacioná-las às técnicas adaptativas. Um trabalho semelhante ao realizado nas ontologias pedagógica e de domínio pode levar à construção de uma ontologia com técnicas adaptativas genéricas, que possam ser reutilizadas nos diferentes cursos criados. Além disso, a proposição de melhorias no padrão SCORM pode proporcionar, também, melhores condições na adaptação dos cursos gerados com base nestas ontologias.

Com relação ao sistema Atlas, várias funcionalidades podem ser adicionadas a ele durante a seleção de objetivos, atividades e objetos de aprendizagem, através da integração de um Sistema Especialista, tornando este processo mais automatizado. Em um primeiro

momento, as regras podem ser estabelecidas separadamente no Sistema Especialista, porém, posteriormente, elas podem ser incluídas nas próprias ontologias, através da linguagem SWRL, desde que não prejudiquem a flexibilidade existente nem diminuam a facilidade para a sua reutilização.

8. REFERÊNCIAS BIBLIOGRÁFICAS

ADL. Advanced Distributed Learning. **Sharable Content Object Reference Model – SCORM[®] Overview**. 2004a. Disponível em: <<http://www.adlnet.org/index.cfm?fuseaction=DownFile&libid=648&bc=false>>. Acesso em: 07 mar. 2005.

ADL. Advanced Distributed Learning. **Sharable Content Object Reference Model – SCORM[®] Content Aggregation Model**. v. 1.3.1. 2004b. Disponível em: <<http://www.adlnet.org/index.cfm?fuseaction=DownFile&libid=648&bc=false>>. Acesso em: 07 mar. 2005.

ADL. Advanced Distributed Learning. **Sharable Content Object Reference Model – SCORM[®] Run-Time Environment**. v. 1.3.1. 2004c. Disponível em: <<http://www.adlnet.org/index.cfm?fuseaction=DownFile&libid=648&bc=false>>. Acesso em: 07 mar. 2005.

ADL. Advanced Distributed Learning. **Sharable Content Object Reference Model – SCORM[®] Sequencing and Navigation**. v.1.3.1. 2004d. Disponível em: <<http://www.adlnet.org/index.cfm?fuseaction=DownFile&libid=648&bc=false>>. Acesso em: 07 mar. 2005.

ANTONIOU, G e HARMELEN, F. van. **A Semantic Web Primer**. Cambridge: MIT Press, 2004. 238 p.

AVGERIOU, P. *et al.* Towards a Pattern Language for Learning Management Systems. **Educational Technology & Society**, v. 6, n. 2, p.11 – 24, abr. 2003.

BATTLE, S. *et al.* **Semantic Web Service Framework (SWSL) Overview** – W3C Member Submission. 2005. Disponível em: <<http://www.w3.org/Submission/SWSF>>. Acesso em: 10 out. 2005.

BECKETT, D. **RDF/XML Syntax Specification (Revised)** - W3C Recommendation 2004. Disponível em: <<http://www.w3.org/TR/rdf-syntax-grammar/>>. Acesso: 18 out. 2004.

BERNERS-LEE, T. **The World Wide Web: A Very Short Personal History**. Apresenta uma visão pessoal do autor sobre o desenvolvimento da *web*. Disponível em: <<http://www.w3.org/People/Berners-Lee/ShortHistory.html>> Acesso: 06 maio 2004.

BERNERS-LEE, T. Foreword. In: FENSEL, D; HENDLER, J.; LIEBERMAN, H.; e WAHLSTER, W. **Spinning the Semantic Web: Bringing the World Wide Web to its Full Potential**. Cambridge: MIT Press, 2003.

- BERNERS-LEE, T. **The Semantic Web**. Academic Discussion: Japan Prize Commemorative Lecture. 2002. Disponível em:
<<http://www.w3.org/2002/04/Japan/Lecture.html>. Acesso em: 01 nov. 2004.
- BERNERS-LEE, T.; HENDLER, J.; e LASSILA, O. **The Semantic Web. Scientific American**. maio. 2001.
- BERNERS-LEE, T. **Weaving the Web**. London: Orion Business, 1999.
- BERNERS-LEE, T. **Semantic Web Road Map**, 1998a. Apresenta um plano em alto nível para a arquitetura da *Web Semântica*. Disponível em:
<<http://www.w3.org/DesignIssues/Semantic.html>>. Acesso em: 21 abr. 2004.
- BERNERS-LEE, T. **Why RDF Model is Different form the XML Model**, 1998b. Apresenta um comparativo entre as linguagens XML e RDF, destacando a importância de cada uma delas para a *Web Semântica*. Disponível em:
<<http://www.w3.org/DesignIssues/RDF-XML.html>>. Acesso: 27 out. 2004.
- BERNERS-LEE, T. **Information Management: A Proposal**, 1990. Apresenta o projeto de um espaço global de hipertexto que mais tarde tornou-se a *Web*. Disponível em:
<<http://www.w3.org/History/1989/proposal.html>>. Acesso em: 13 jul. 2004.
- BIRON, P. V.; MALHOTRA, A. **XML Schema Part 2: Datatypes - W3C Recommendation**. 2004. 2 ed. Disponível em: <<http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>>. Acesso em: 01 nov. 2004
- BITTENCOURT, D. F. **Estratégia e Tomada de Decisões para Educação a Distância**. Baseado em Universidade Nacional Autônoma do México. Tubarão: UNISUL, 2003. 197p.
- BLOOM, B. S.; *et. al.* **Taxionomia de Objetivos Educacionais**. v. 1. Porto Alegre: Globo. 1976.
- BRAY, T.; HOLLANDER, D.; LAYMAN, A. **Namespaces in XML – W3C Recommendation**. 1999. Disponível em: <<http://www.w3.org/TR/1999/REC-xml-names-19990114/>>. Acesso em: 01 nov. 2004.
- BRICKLEY, D.; GUHA, R. V. **RDF Vocabulary Description Language 1.0: RDF Schema - W3C Recommendation**. 2004. Disponível em:
<<http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>> . Acesso em: 20 out. 2004.
- BRUSILOVSKY, P. A Distributed Architecture for Adaptive and Intelligent Learning Management Systems. In: Proceedings of TOWARDS INTELLIGENT LEARNING MANAGEMENT SYSTEMS, 2003, Sydney.
- BRUSILOVSKY, P. Adaptive Hypermedia. **User Modelling and User-Adapted Interaction**. v 11. n 1-2. p. 87 – 110, mar. 2001.
- BRUSILOVSKY, P. Adaptive and Intelligent Technologies for Web-based Education. **Künstliche Intelligenz**. v. 4. 1999. p 19 – 25.
- BRUSILOVSKY, P. Methods and techniques of Adaptive Hypermedia. **User Modelling and User-Adapted Interaction**. v. 6 n. 2 - 3. p. 87 – 129, 1996.

CALVO, R. A. User Scenarios for the Design and Implementation of iLMS. In: Proceedings of TOWARDS INTELLIGENT LEARNING MANAGEMENT SYSTEMS, 2003, Sydney.

CHKLOVSKI T.; PANTEL, P. Large-Scale Extraction of Fine-Grained Semantic Relations between Verbs. IN: Proceedings of KDD WORKSHOP ON MINING FOR AND FROM SEMANTIC WEB, 2004, Seattle (EUA).

COMER, D. E. **Interligação em Rede com TCP/IP**: Princípios, Protocolos e Arquitetura. v.1. 3.ed. Rio de Janeiro: Campus, 1998. 672 p.

CONNOLLY, D. **A little History of the World Wide Web**: from 1945 to 1995, 2000. Apresenta um breve histórico com os fatos marcantes para o desenvolvimento da *Web* entre 1945 e 1995. Disponível em: <<http://www.w3.org/History.html>>. Acesso em: 06 maio 2004.

DACONTA, M. C.; OBRST, L. J.; SMITH, K. T. **The Semantic Web**. A Guide to the Future of XML, Web Services, and Knowledge Management. Indianápolis(EUA): Wiley Publishing Inc, 2003. 281 p.

DEAN, M.; SCHREIBER, G. **OWL – Web Ontology Language Reference**: W3C Recommendation. 2004. Disponível em: <<http://www.w3.org/TR/2004/REC-owl-ref-20040210/>>. Acesso em: 10 jan. 2005.

DENAUX, R; DIMITROVA, V. e AROYO, L. Integrating Open User Modeling and Learning Content Management for the Semantic Web. In: Proceedings of INTERNATIONAL CONFERENCE ON USER MODELING, 10. Edimburgo (Escócia), 2005.

DEVEDZIC, V. Web Intelligence and Artificial Intelligence in Education. **Educational Technology & Society**. v.7 n. 4. p. 29 – 39. out. 2004.

DODIG-CRNKOVIC, G. **Scientific Methods in Computer Science**. In: Proceedings of CONFERENCE FOR THE PROMOTION OF RESEARCH IN IT AT NEW UNIVERSITIES AND AT UNIVERSITY COLLEGES IN SWEDEN. 2002, Skovde (Suécia).

DUBLINCORE. **The Dublin Core Metadata Initiative**. Disponível em: <<http://dublincore.org/>>. Acesso em: 15 mar. 2005.

E-COMMERCEORG. E-Commerce.org. Tudo sobre comércio Eletrônico. **Dados Estatísticos sobre a Internet e o Comércio Eletrônico**. 2005. Apresenta dados sobre os valores movimentados nas transações eletrônicas entre empresas no Brasil, e nas vendas diretas aos clientes através da Web. Disponível em: <<http://www.e-commerce.org.br/STATS.htm>>. Acesso em: 06 jan. 2006.

ET&S. **Educational Technology & Society**. v. 7, n. 4. 2004.

EUZENAT, J.(Editor). **Research Challenges and Perspectives of the Semantic Web**. Report of the European Commission and US National Science Foundation Strategic Workshop. Sophia Antipolis (França), 2001.

- FALLSIDE, D. C.; WALMSLEY, P. XML Schema Part 0: Primer -W3C Recommendation. 2004. 2 ed. Disponível em: <<http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/>>. Acesso: 01 nov. 2004.
- FENSEL, D *et al.* Introduction. In: FENSEL, D; HENDLER, J.; LIEBERMAN, H. e WAHLSTER, W. **Spinning the Semantic Web: Bringing the World Wide Web to its Full Potential**. Cambridge: MIT Press, 2003. Cap. 1.
- FENSEL, D.; *et al.* **On-To-Knowledge: Semantic Web Enabled Knowledge Management**. Submetido para IEEE Computers. 2001. Disponível: <http://www.ontoknowledge.org>. Acesso em: 10 set. 2004.
- GADOTTI, M. **Perspectivas Atuais da Educação**. Porto Alegre: Artmed. 2000. 296 p.
- GAGNÉ, R. M. **The Conditions of Learning and Theory of Instruction**. New York: Holt, Rinehart & Winston. 1985.
- GAGNÉ, R. M. **Princípios Essenciais da Aprendizagem para o Ensino**. Porto Alegre: Globo. 1980.
- GOULART, R. R. V; GIRAFFA, L. M. M. **Arquitetura de Sistemas Tutores Inteligentes**. Série Technical Reports, 11. Porto Alegre: Campus Global - FACIN – PUCRS, 2001. 31 p. Disponível em: <<http://www.inf.pucrs.br/tr/tr011.pdf>>. Acesso em: 12 nov. 2004.
- GRUBER, T. R. A Translation Approach to Portable Ontologies. **Knowledge Acquisition**. v.5. n. 2, p. 199 – 220, 1993a.
- GRUBER, T. R. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In GUARINO, N.;POLI, R.; **Formal Ontology in Conceptual Analysis and Knowledge Representation**. Kluwer Academic Publishers, 1993b.
- GUARINO, N. Formal Ontology and Information System. IN. FORMAL ONTOLOGY AND INFORMATION SYSTEM. 1998. Trento (Itália). **Proceedings....** Amsterdam: IOS Press. p. 3 – 15.
- HENDLER, J.; BERNERS-LEE, T.; MILLER, E.. Integrating Applications on the Semantic Web. **Journal of the Institute of Electrical Engineers of Japan**. v. 122 n.10, out. 2002. p. 676 - 680.
- HOLOHAN, E. *et al.* Adaptive E-Learning Content Generation based on Semantic Web Technology. In: Proceedings of INTERNATIONAL WORKSHOP ON APPLICATIONS OF SEMANTIC WEB TECHNOLOGIES FOR E-LEARNING (SW-EL). International Conference on Artificial Intelligence in Education. Amsterdam, 2005.
- HOLZNER, S. **Desvendando XML**. Trad. Daniel Vieira. Rio de Janeiro: Campus, 2001. 858 p.
- HORROCKS, I. *et al.* **SWRL: A Semantic Web Rule Language Combining OWL e RuleML - W3C Recommendation**. 2004. Disponível em: <<http://www.w3.org/Submissions/2004/SUBM-SWRL-20040521/>>. Acesso em: 19 abr. 2005.

ILMS. Towards Intelligent Learning Management Systems. Conference: Artificial Intelligence in Education. Sydney. 2003.

IMS. Global Learning Consortium, Inc. **IMS Content Packing Information Model.** v 1.1.4. 2004. Disponível em: <http://www.imsglobal.org/content/packaging/cpv1p1p4/imscp_infov1p1p4.html>. Acesso em: 07 mar. 2004.

IMS. Global Learning Consortium, Inc. **IMS Simple Sequencing Information and Behaviour Model.** v 1.0. 2003a. Disponível em: <http://www.imsglobal.org/simplesequencing/ssv1p0/imsss_infov1p0.html>. Acesso em: 07 mar. 2004.

IMS. Global Learning Consortium, Inc. **IMS Learning Design - Information Model.** V. 1.0. 2003b. Disponível em: <<http://www.imsglobal.org/specificationdownload.cfm>>. Acesso em: 15 mar. 2005.

IMS. Global Learning Consortium, Inc. **IMS Learner Information Package - Information Model.** v. 1.0. 2001a. Disponível em: <<http://www.imsglobal.org/profiles/lipinfo01.html>>. Acesso em: 13 jan. 2005.

IMS. Global Learning Consortium, Inc. **IMS Learning Resource Meta-data Information Model.** v 1.2. 2001b. Disponível em: <<http://www.imsglobal.org/metadata/index.html>>. Acesso em: 21 dez. 2004.

INTERNET World Stats. **Internet Usage Statistics – The Big Picture.** 2005. Apresenta dados sobre o número de usuários da Internet em cada continente ou país. Disponível em: <<http://www.internetworldstats.com/stats.htm>>. Acesso em: 05 jan. 2005.

JURAFSKY, D.; MARTIN, J. H. Introduction. In: _____. **Speech and Language Processing: A Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.** New Jersey: Prentice Hall, 2000. cap 1.

KLEIN, M; BROEKSTRA, J.; FENSEL, D.; HARMELEN, F. Van; HORROKS, I. Ontologies and Schema Languages on the Web. In: FENSEL, D; HENDLER, J.; LIEBERMAN, H.; e WAHLSTER, W. **Spinning the Semantic Web: Bringing the World Wide Web to its Full Potential.** Cambridge: MIT Press, 2003. Cap. 4.

KLYNE, G., CARROLL J. J. (2004). **Resource Description Framework (RDF): Concepts and Abstract Syntax - W3C Recommendation.** 2004. Disponível em: <[http://www.w3.org/TR/Resource Description Framework \(RDF\) Concepts and Abstract Syntax.htm/](http://www.w3.org/TR/Resource%20Description%20Framework%20(RDF)%20Concepts%20and%20Abstract%20Syntax.htm/)>. Acesso: 25 out. 2004.

KOPER, R. Use of the Semantic Web to Solve Some Basic Problems in Education: Increase Flexible, Distributed Lifelong Learning, Decrease Teacher's Workload. **Journal of Interactive Media in Education**, v. 6, p.1 – 23, maio. 2004.

KUROSE F. J.; ROSS W. K. **Redes de Computadores e a Internet: uma nova abordagem.** São Paulo: Addison Wesley, 2003. 348 p.

LITWIN, E. (Org). **Educação a Distância.** Temas para o Debate de uma Nova Agenda Educativa. Porto Alegre: Artmed, 2001. 100 p.

- LONGMIRE, W. A Primer on Learning Objects. **Learning Circuits on-line** . 2000. <<http://www.learningcircuits.org>>. Acesso : 20 mar. 2005.
- LTSC. **Learning Technology Standards Committee**. Institute of Electrical and Electronic Engineers. 2004. Disponível em: <<http://ltsc.ieee.org/wg12/index.html>>. Acesso : 11 jan. 2005.
- LTSC. **IEEE Standard for Learning Object Metadata**. Learning Technology Standards Committee. Institute of Electrical and Electronic Engineers. 2002. Disponível em: <<http://standards.ieee.org/>>. Acesso em: 11 jan. 2005.
- LTSC. **IEEE Draft Standard for Learning Technology – Public and Private Information (PAPI) for Learners (PAPI Learner)**. Learning Technology Standards Committee. Institute of Electrical and Electronic Engineers. 2000. Disponível em: <<http://ltsc.ieee.org/wg12/>>. Acesso em: 14 jan 2005.
- MAIA, C. (Org). **ead.br:Experiências Inovadoras em Educação a Distância no Brasil – reflexões atuais, em tempo real**. São Paulo: Anhembi Morumbi, 2003. 123 p.
- MAIA, C. **Guia Brasileiro de Educação a Distância 2000/2001**. São Paulo: Esfera, 2001. 176 p.
- MANOLA, F.; MILLER, E. **RDF Primer- W3C Recommendation**. 2004. Disponível em: <<http://www.w3.org/TR/rdf-primer/>>. Acesso em: 10 out. 2004.
- McGUINNESS, D. L.; HARMELEN F. **OWL – Web Ontology Language Overview: W3C Recommendation**. 2004. Disponível em: <<http://www.w3.org/TR/2004/REC-owl-features-20040210/>>. Acesso: 10 jan. 2005.
- McGUINNESS, D. L. Ontologies come of Age. In: FENSEL, D; HENDLER, J.; LIEBERMAN, H.; e WAHLSTER, W. **Spinning the Semantic Web: Bringing the World Wide Web to its Full Potential**. Cambridge: MIT Press, 2003. Cap. 6.
- McILRAITH, S. A.; SON, T. C.; ZENG, H. Semantic Web Services. **IEEE Intelligent Systems**. p 46 – 53, mar. 2001.
- MIZOGUCHI, R. e BOURDEAU, J. Using Ontological Engineering to Overcome Common AI-ED Problems. **International Journal of Artificial Intelligence in Education**. v. 11, n. 2, p. 107 – 121, 2000.
- MOORE, M. G., KEARSLEY, G. **Distance Education: a systems view**. Belmont (EUA): Wadsworth Publishing Company, 1996.
- MURRAY, T. Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art. **International Journal of Artificial Intelligence in Education**.n. 10. p. 98 – 129, 1999.
- NAGY , A. **e-Learning**. E-Content Report 6. ACTeN – Antecipating Content Technology Need. Disponível em: <<http://www.acten.net>>. Acesso em: 08 ago. 2004.
- NEWHOUSE, S. *et. al.* **Laying the Foundations for the Semantic Grid**. Submetido para Imperial College AISB Convention. London, 2002. Disponível em: <<http://www-icpc.doc.ic.ac.uk/components/papers/AISB2002.pdf>>. Acesso em: 18 mar. 2004.

NILSSON, M. **IEEE Learning Object Metadata RDF Binding**. 2002. Disponível em: <<http://kmr.nada.kth.se/el/ims/metadata.html>>. Acesso: 15 set. 2004.

NILSSON, M. **The Semantic Web: How RDF will change learning technology standards**. Apresenta algumas características básicas da linguagem RDF e de que forma ela pode auxiliar no desenvolvimento de sistemas educacionais. Estocolmo. 2001. Disponível em: <<http://www.cetis.ac.uk/content/20010927172953/printarticle>>. Acesso em: 23 abr. 2004.

NILSSON, N. **J. Artificial Intelligence: A New Synthesis**. San Francisco (EUA): Morgan Kaufmann, 1998. 513p.

NOY, N.F. *et al.* Creating Semantic Web contents with Protege-2000. **IEEE Intelligent Systems**. v. 16. n. 2. p 60 – 71, mar. 2001.

OLIVEIRA, J. M. P.; FERNANDES, C. T. Arquitetura de Adaptação em Sistemas Hipermídia Adaptativos Educacionais. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 13, São Leopoldo, 2002. **Anais ...** São Leopoldo: UNISINOS, 2002.

PATEL-SCHNEIDER, P. F., HAYES, P. e HORROCKS, I. **OWL Web Ontology Language Semantics and Abstract Syntax - W3C Recommendation**. 2004. Disponível em: <<http://www.w3.org/TR/2004/REC-owl-semantics-20040210/>>. Acesso: 10 jan. 2005.

PATEL-SCHNEIDER e HORROCKS, I. **OWL Web Ontology Language Semantics and Abstract Syntax Section 2. Abstract Syntax - W3C Recommendation**. 2004. Disponível em: <<http://www.w3.org/TR/2004/syntax.html>>. Acesso: 10 jan. 2005.

POWER, G. *et al.* Goal Oriented Personalisation with SCORM. In: Proceedings of IEEE INTERNATIONAL CONFERENCE ON ADVANCED LEARNING TECHNOLOGIES, 5. Koahsiung (Taiwan), 2005.

PROTEGE. **Welcome to Protégé**. Disponível em: <<http://protege.stanford.edu>>. Acesso em: 07 nov. 2005.

RAY, E. T. **Aprendendo XML**. Trad. Daniel Vieira. Rio de Janeiro: Campus, 2001. 372 p.

ROURE, D. D. **The Semantic Grid Vision**. Apresenta a visão que a comunidade do semantic grid têm sobre a próxima geração da Internet. Disponível em: <<http://www.semanticgrid.org/vision.html>>. Acesso: 21 set. 2004.

RULEML. **The Rule Markup Initiative**. Disponível em: <<http://www.ruleml.org/>>. Acesso em: 12 nov. 2005.

RUSSEL, S.; NORVIG. P. **Artificial Intelligence: A Modern Approach**. 2 ed. New Jersey (EUA): Prentice Hall, 2003.

SANTACHÈ, A. Aplicações Educacionais na *Web* – O papel de RDF e Metadados. In: Anais do SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 14, Rio de Janeiro, 2003.

SASKATCHEWAN EDUCATION. Instructional Models, Strategies, Methods, And Skills In: **Instructional Approaches: A Framework for Professional Practice**. 1991. Cap 2. Disponível em: <<http://www.sasklearning.gov.sk.ca/docs/policy/approach/index.html>>. Acesso em: 10 jun. 2005.

SELF, J. The defining characteristics of intelligent tutoring systems research: ITSs care, precisaly. **International Journal of Artificial Intelligence in Education**. v. 10. p. 350 – 364, 1999.

SELF, J. Theoretical Foundations for Intelligent Tutoring Systems. **International Journal of Artificial Intelligence in Education**. v. 1 n. 4. p. 3 – 14, 1990.

SEMANTIC WEB.org. **Welcome**. Apresenta uma visão geral sobre o que é *Web Semântica* e sobre o projeto liderado pela SemanticWeb.org. Disponível em: <<http://www.semanticweb.org/introduction>> Acesso em: 06 maio 2004.

SILVA, E. L. da e MENEZES, E. M. **Metodologia da Pesquisa e Elaboração de Dissertação**. 4 ed. Florianópolis: UFSC, 2005. 138 p.

SILVESTRE, P. E-commerce faturou R\$ 2,5 bilhões em 2005 no país. **Plantão INFO**. 2006. Apresenta dados sobre os valores movimentados através do comércio eletrônico no Brasil. Disponível em: <<http://info.abril.com.br/aberto/infonews/012006/04012006-1.shl>>. Acesso em: 06 jan. 2006.

SIMIC,G.; GASEVIC, D.; DEVEDZIC, V. Semantic Web and Intelligent Learning Management Systems. In: Proceedings of INTERNATIONAL WORKSHOP ON APPLICATIONS OF SEMANTIC WEB FOR E-LEARNING. Maceió, 2004.

SMITH, M. K.; WELTY, C.; McGUINNESS, D. L. **OWL Web Ontology Language Guide - W3C Recommendation**. 2004. Disponível em: <<http://www.w3.org/TR/2004/REC-owl-guide-20040210/>>. Acesso em: 10 jan. 2005.

SW-EL. **International Workshop on Applications of Semantic Web Technologies for E-Learning**. Maceió, 2004.

TANENBAUM, A. S. **Redes de Computadores**. Tradução da 3 ed. Rio de Janeiro: Campus, 1997. 923 p.

THOMPSON, H. S. *et al.* **XML Schema Part 1: Structures - W3C Recommendation**. 2004. Disponível em: <<http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>>. Acesso em: 01 nov. 2004.

UNICODE. **What is Unicode**. Apresenta informações sobre o que é e quais são os objetivos do Unicode. Disponível em: <<http://www.unicode.org/standard/WhatIsUnicode.html>>. Acesso em: 01 nov. 2004.

VICARI, R. M. GIRAFFA, L. M. M. Fundamentos dos Sistemas Tutores Inteligentes. In: BARONE, D. **Sociedades Artificiais: A Nova Fronteira da Inteligência das Máquinas**. Porto Alegre: Artmed, 2003. Cap. 7.

W3C. **URIs, URNs, and URNs: Clarification and Recommendations 1.0**. 2001. Disponível em: <<http://www.w3.org/TR/uri-clarification>>. Acesso em: 2 out. 2004.