

UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE

ROOSEWELT LINS SILVA

**REPRESENTAÇÃO E AGREGAÇÃO DE CONTEÚDOS EM REPOSITÓRIO
DE OBJETOS DE APRENDIZAGEM**

São Luís
2007

ROOSEWELT LINS SILVA

**REPRESENTAÇÃO E AGREGAÇÃO DE CONTEÚDOS EM REPOSITÓRIO
DE OBJETOS DE APRENDIZAGEM**

Dissertação apresentada ao
Programa de Pós-Graduação em
Engenharia de Eletricidade da
Universidade Federal do Maranhão,
para obtenção do grau de Mestre em
Engenharia de Eletricidade, área de
concentração Ciências da
Computação.

São Luís
2007

ROOSEWELT LINS SILVA

**REPRESENTAÇÃO E AGREGAÇÃO DE CONTEÚDOS EM REPOSITÓRIO
DE OBJETOS DE APRENDIZAGEM**

Dissertação apresentada ao
Programa de Pós-Graduação em
Engenharia de Eletricidade da
Universidade Federal do Maranhão,
para obtenção do grau de Mestre em
Engenharia de Eletricidade, área de
concentração Ciências da
Computação.

Aprovada em ____/____/____

BANCA EXAMINADORA

Prof. Dr. Sofiane Labidi
Orientador

Prof. Dr. Ademir Martins
Co-Orientador

Prof. Dr. Zair Abdelouahab
Examinador Interno

Prof. Dr. Rubens Nascimento Melo
Examinador Externo

*Este trabalho é dedicado às iniciativas de promoção
do compartilhamento da informação e
conhecimento na Web. Sem estas iniciativas, o
estudo não poderia ser realizado*

AGRADECIMENTOS

Aos professores da UFMA que abriram oportunidades para desenvolver pesquisas e atuar no âmbito da computação: Sofiane Labidi, Osvaldo Saavedra, Yuri Catunda, Carlos Brandão e Maria da Guia. Ao Departamento de Engenharia de Eletricidade, aos colegas do Laboratório de Sistemas Inteligentes da UFMA pelo rico aprendizado e a André Santos (CEFET-MA) pela força.

Ao CNPq por todas as bolsas concedidas. À FAPEMA pelo financiamento desta pesquisa e as iniciativas de desenvolvimento da Ciência e Tecnologia no estado do Maranhão.

Aos colegas professores do Departamento de Ciências da Informação da UFC pela receptividade e oportunidades oferecidas para o desenvolvimento acadêmico e estímulo à docência em Tecnologias da Informação. Em especial à Prof^a Virgínia Bentes e integrantes do grupo de pesquisa em Representação da Informação da UFC.

Aos meus grandes amigos que fiz durante a vida acadêmica nos encontros de estudantes e que hoje formam uma grande família: Ediane (PE), Renatha James (CE), Gustavo Costa (RN), Moreno Barros (RJ) e pessoal da UFF, Alessandra (RN), Raline (RN), Maria Luzia (RN), Soninha (PI), Nancy Di Paula (PI), Alex Saraiva (RJ) e pessoal da UniRio, Ana Helena (MA), Rosilene Cardoso (MA), Tati Dias, Selminha e o pessoal de MG e todos os outros amigos de diversos estados e instituições de ensino deste grande país. Aos irmãos: Rodrigo Galvão (PE), Lúcio (PE), Maurício (PE), Gustavo Hen (PB), Rodrigo Peruca (PE), André Cavalcante (CE), Cauê Araújo (CE), Jonathas Silva (CE), Júlio Rei (PB) e Leonardo Junior (MA).

A Bruno Rogens pelas discussões sociológicas referentes aos Softwares Livres, Copyleft e Filosofia Colaborativa. A Hélio Craveiro pelas reflexões sobre educação e TV Digital, assim como a ajuda fornecida. Aos demais amigos da UFMA de diversos cursos que forneceram uma visão mais abrangente da universidade e da cultura maranhense.

À Gerlandy Leão pela paciência, companheirismo, compreensão e carinho, sobretudo.

A Beth Araújo uma segunda mãe que me recebeu de braços abertos e carinho em Fortaleza, assim como Maria do Rosário, grande avó.

À minha querida e amada mãe Telma Silva e irmã Madhava Assis Silva pelos laços familiares, amor, apoio e compreensão. À minha tia Maricelma Silva pelo carinho e preocupação. Em especial ao meu amado avô José de Ribamar Silva que está descansando em paz, ao meu pai Barbosa Filho pelo aprendizado tecnológico e familiar e a todos os demais familiares.

Aos colegas do Projeto Casa Brasil - CE, onde pude conviver com ações em prol do Software Livre e Inclusão Digital.

*“Por de trás de algo que se esconde há
sempre uma grande mina de
conhecimentos e sentimentos”*

Chico Science

RESUMO

A educação mediada por tecnologia é uma ferramenta cada vez mais utilizada em ambientes acadêmicos e corporativos. Com o avanço da Web, diversos ambientes de ensino-aprendizagem possibilitaram a produção e disponibilização de conteúdos multimídias para o uso de aprendizes e educadores. Todavia o acesso a estes conteúdos ainda é um dos principais problemas para o uso e compartilhamento entre diferentes aplicações. A representação de documentos na Web Semântica é uma técnica relacionada ao uso de metadados para descrever recursos, sendo uma solução para o problema de acesso a conteúdos na Web. No cenário da educação baseada na Web, diversos padrões de metadados têm sido propostos para proporcionar o compartilhamento de recursos de aprendizagem de forma distribuída. Acredita-se que o uso das ontologias permitirá uma melhor conceituação e representação do domínio, possibilitando desta forma uma formalização dos esquemas de metadados para gerenciamento de objetos de aprendizagem. Apresenta-se um Modelo de Agregação e Representação de Conteúdo para conceituação de um Repositório Semântico de Objetos de Aprendizagem. O Modelo de Agregação faz uso do padrão LOM (Learning Object Metadata) para descrever e agregar conteúdos educacionais. O Modelo de Representação de Conteúdos é um Esquema de Classificação baseado no padrão SKOS (Simple Knowledge Organisation Systems) destinado à especificação de Sistemas de Organização do Conhecimento na Web Semântica. Utilizou-se a metodologia METHONTOLOGY, linguagem OWL (Web Ontology Language) para construção da ontologia e o uso do *framework* Jena destinado à manipulação de modelo ontológico. Desta forma, discutem-se pressupostos associados à representação do conhecimento na Web, tecnologias educacionais, perspectivas e desafios para o desenvolvimento da nova geração da Web.

Palavras-chave: Representação do Conhecimento. Engenharia de Ontologias. Web Semântica. Repositório de Objetos de Aprendizagem. Sistemas de Gerenciamento de Aprendizagem. Metadados.

ABSTRACT

The education mediated by technology is a tool used in academic and corporative environments. With advance of the Web, diverse environments of teaching and learning make possible the production and distribution of multimedia contents for the use of learners and teachers. However the contents access still is one of the main problems for the use and sharing between different applications. The document representation in Semantic Web is related to the use of metadata to describe resources. In Web-based education, divers standards have been considered to provide sharing learning resources in distributed form. One believes that ontology use allows one better conceptualization and domain representation, making possible the formalization of the metadata schema for learning object management. One presents an Aggregation and Representation Content Model for conceptualization of a Semantic Learning Object Repository. The Aggregation Model makes use of standard LOM (Learning Object Metadata) to describe and add educational contents. The Content Representation Model is a Classification Schema based on SKOS (Simple Knowledge Organisation Systems) standard destined the specification of knowledge organisation systems in the Semantics Web. It was use OWL language (Web Ontology Language) for ontology construction and framework Jena for manipulation of the ontological model. In such a way, it argues concepts associates the educational technologies, perspectives and challenges for knowledge representation on the Web, and for the development of new generation of the Web.

Key-words: Knowledge Representation. Ontology Engineering. Semantic Web. Learning Object Repository. Learning Management System. Metadata.

LISTA DE FIGURAS

Figura 1 – Arquitetura da Web Semântica	34
Figura 2 – Documento XML	37
Figura 3 – Documento XML com um Esquema DTD	38
Figura 4 – Esquema e Instâncias	39
Figura 5 – XML Schema	40
Figura 6 – Utilização do esquema XSD no documento XML	40
Figura 7 – Folha de Estilo XSL	41
Figura 8 – Apresentação de um Documento XML transformado em HTML	42
Figura 9 – Uma tripla representada em grafo	45
Figura 10 – Relações de subclasses OWL e RDF-RDFS	52
Figura 11 – Esquema de Classificação em Engenharia de Eletricidade	70
Figura 12 – Exemplo em Skos-Core do Esquema de Classificação	71
Figura 13 – Grafo Skos-Core representando o conceito Ciências da Computação	72
Figura 14 – Atividades de desenvolvimento proposta pela METHONTOLOGY	73
Figura 15 – Arquitetura do Repositório de Objetos de Aprendizagem	77
Figura 16 – Tarefas da Atividade de Conceituação	78
Figura 17 – Organização de Conteúdo	81
Figura 18 – Taxonomia de Conceitos	87
Figura 19 – Diagramas de relações binárias	88
Figura 20 – Classes e Subclasses	97
Figura 21 – Representação em OWL da Classe Pacote de Conteúdo	98
Figura 22 – Propriedades do Domínio	99
Figura 23 – Exemplo de um Simples Diagrama de Classes	100
Figura 24 – Propriedades de Objeto no Domínio	101
Figura 25 – Propriedade Inversa	109
Figura 26 – Propriedade Funcional de um Objeto	103
Figura 27 – Propriedade Funcional de um Tipo de Dado	103
Figura 28 – Propriedade Simétrica	104
Figura 29 – Axioma para Coleção Ordenada	105
Figura 30 – Axioma para coleção ordenada	106
Figura 31 – Exemplo de Axioma	106
Figura 32 – Criação de um modelo em RDF utilizando JENA	108
Figura 33 – Modelo que gera um documento RDF	108
Figura 34 – Documento RDF gerado pelo Jena	109
Figura 35 – Simples Consulta	109
Figura 36 – Resultado de uma Simples Consulta	110

LISTA DE TABELAS

Tabela 1 – Partes de um documento RDF	46
Tabela 2 – Partes de um documento OWL	54
Tabela 3 – Glossário de termos	79
Tabela 4 - Dicionário de Conceitos	91
Tabela 5 - Detalhamento das Relações Binárias	92
Tabela 6 - Atributos de Instância	93
Tabela 7 - Atributos de Classes	93

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivos	15
1.2	Justificativa	16
1.3	Metodologia	16
1.4	Estrutura da Dissertação	18
2	TECNOLOGIAS EDUCACIONAIS BASEADA NA WEB	20
2.1	Sistemas de Gerenciamento de Aprendizagem	20
2.2	Objetos de Aprendizagem	24
2.3	Especificações para Objetos de Aprendizagem	29
3	WEBSEMÂNTICA E ONTOLOGIAS	30
3.1	Linguagens	36
3.1.1	XML (eXtensible Markup Language)	37
3.1.2	RDF (Resource Description Framework)	43
3.1.3	OWL (Web Ontology Language)	51
3.2	Engenharia de Ontologias	57
3.3	Ferramentas	61
4	CLASSIFICAÇÃO E ORGANIZAÇÃO DO CONHECIMENTO	62
4.1	Fundamentos da Classificação e Organização do Conhecimento	62
4.1.1	Instrumentos para organização do conhecimento	69
5	DESENVOLVIMENTO DO MODELO DE AGREGAÇÃO E REPRESENTAÇÃO DE CONTEÚDOS	73
5.1	Especificação e Conceituação da Ontologia	75
6	FORMALIZAÇÃO E IMPLEMENTAÇÃO	96
6.1	Classes	96
6.2	Relacionamento e Propriedades das Classes	98
6.3	Axiomas e Restrições	105
6.4	Utilização do Framework Jena	107
7	CONCLUSÃO	111
8	TRABALHOS FUTUROS	113
	REFERÊNCIAS	114
	APÊNDICE	122

1 INTRODUÇÃO

Aprendizes e educadores estão cada vez mais utilizando a Internet como recurso de comunicação e compartilhamento de informação. O movimento de compartilhamento da informação emerge na necessidade de integração da produção do conhecimento e a possibilidade de acesso aberto a conteúdos educacionais. A introdução das tecnologias no ambiente educacional favorece novos processos interativos de ensino-aprendizagem através da ruptura de fronteiras geográficas e o uso de recursos multimídias. Esta interatividade estimula novas formas de pensamento através de ambientes de rede que interligam culturas, conectam significados e provêem intercâmbio do conhecimento.

No cenário da educação contemporânea, a incorporação das tecnologias em práticas pedagógicas têm sido um dos fatores mais relevantes para mediação do processo de ensino-aprendizagem. A cultura centrada no uso de tecnologias da informação e comunicação proporcionou inovações na esfera educacional. Aprendizes e professores, todos envolvidos nesse processo, passam a contar com diversas ferramentas para produzir e difundir conteúdos didáticos. Dentre estas ferramentas, convém ressaltar a Internet, em especial a World Wide Web. A Web é um ambiente distribuído e descentralizado que disponibiliza divergentes conteúdos na forma de hipertexto e documentos multimídias. Estes divergentes conteúdos são formatos de arquivos e tipos de dados que formam a heterogeneidade da Web. Tal heterogeneidade é o fator principal para a dificuldade no acesso e organização da informação, pois os dados na Web são desestruturados. Diante deste contexto, a solução encontrada para atribuir uma organização mais sistêmica e lógica na Web é a Web Semântica. A Web Semântica é uma nova geração de aplicações que introduz diversos conceitos da representação do conhecimento e de linguagens extensíveis para marcação de documentos.

De acordo com Santacruz-Valencia, Aedo e Kloos (2005), as tendências de desenvolvimento da Web Semântica aplicada à educação se concentram em três áreas: a informática educacional, projeto instrucional e

os sistemas de bibliotecas. As iniciativas resultantes da combinação entre estas áreas deram origem a novas áreas de interesse como o desenvolvimento de metadados educacionais, Mapas de Tópicos e recentemente investigações pedagógicas sobre os metadados gerando a relação de objetos de aprendizagem com a Web Semântica.

Os objetos de aprendizagem são desenvolvidos e armazenados em diferentes lugares na Web, assim as especificações para metadados de objetos de aprendizagem permitem reusar os conteúdos em diferentes sistemas de gerenciamento de aprendizagem. Os sistemas de aprendizagem atuais são essencialmente centralizados, com objetos de aprendizagem gerenciados em um simples lugar e em um simples sistema. Entretanto, com o crescimento dos repositórios de objetos de aprendizagem na Web Semântica, este modelo pode ceder um modelo distribuído de processamento. Neste caso, os Sistemas de aprendizagem passam a gerenciar objetos de aprendizagem distribuídos (MOHAN; BROOKS, 2003).

Diante deste quadro convém enfocar os Objetos de Aprendizagem, e o seu gerenciamento no cenário da Web Semântica. Esta abordagem é voltada exclusivamente para o compartilhamento e reuso inteligente de conteúdos na educação mediada pela Web. Através de um Repositório de Objetos de Aprendizagem, os conteúdos educacionais podem ser criados, classificados, empacotados e exportados para auxiliar em um processo de ensino/aprendizagem em ambiente Web. O Repositório Semântico de Objetos de Aprendizagem é baseado no Modelo de Agregação de Conteúdo da especificação SCORM (Sharable Content Object Reference Model) (ADL, 2006) e no vocabulário SKOS (MILES; MATHEWS; WILSON, 2006). O SCORM baseia-se no padrão de Metadados para Objetos de Aprendizagem (Learning Object Metadata) proposto pelo IEEE Learning Technology Standards Committee (IEEE, 2001) e no padrão para empacotamento de conteúdos IMS Global Learning Consortium (IMS, 2006). Estes padrões juntos servem para descrição, agregação e empacotamento de Objetos de Aprendizagem.

O SKOS (Simple Knowledge Organisation System) é uma especificação proposta pela W3C (2006a) para criação de Sistemas de Organização do Conhecimento (Knowledge Organisation System - SKO). O padrão SKOS provê um modelo para esquema de conceitos, tal como,

tesauros, taxonomias, cabeçalho de assuntos, folksonomias, esquemas de classificação e glossários. Com SKOS elaboram-se terminologias e vocabulários controlados para classificação de documentos na Web. Um SKO pode ser utilizado para classificar os conteúdos educacionais na Web, permitindo uma organização semântica dos objetos de aprendizagem por área do conhecimento.

Desta forma, todos estes padrões serão representados em ontologias para conceituação de um modelo de repositório destinado a criar, organizar e compartilhar conteúdos educacionais, ou seja, os objetos de aprendizagem (aulas, lições e avaliações).

1.1 Objetivos

1.1.1 Geral

Esta dissertação tem como objetivo central propor um modelo ontológico de representação e agregação de conteúdos educacionais utilizando as especificações para objetos de aprendizagem, linguagens e tecnologias da Web Semântica.

1.1.2 Específicos

Os objetivos específicos da presente pesquisa são:

- Apresentar os principais fundamentos referentes aos objetos de aprendizagem reutilizáveis;
- Identificar as linguagens, ferramentas e técnicas para Web Semântica e criação de ontologias;
- Desenvolver a ontologia para representar o modelo de agregação de conteúdos SCORM e um esquema de classificação baseado no padrão SKOS para organização de objetos de aprendizagem;

1.2 Justificativa

Esta pesquisa é parte integrante do projeto NetClass vinculado ao Laboratório de Sistemas Inteligentes da Universidade Federal do Maranhão. O NetClass é um Sistema de Gerenciamento de Aprendizagem que utiliza técnicas provenientes da Inteligência Artificial Distribuída. Através de uma abordagem Multiagente, o NetClass fornece um ambiente colaborativo de ensino/aprendizagem, no qual entidades humanas e artificiais interagem em ambiente hipermídia visando a promoção do aprendizado. Entretanto, o NetClass não fornece suporte a criação e acesso a objetos de aprendizagem padronizados, o que dificulta o reuso e recuperação de conteúdos distribuídos. Esta ausência de interoperabilidade é o fator crucial para o interesse e realização da presente pesquisa. Este estudo é também motivado pelos seguintes pontos considerados imprescindíveis: a adequação do Netclass às tecnologias e padrões da informática educacional; novas perspectivas referentes ao desenvolvimento de aplicações para Web Semântica, principalmente no domínio educacional e necessidade de se modelar um repositório de objetos de aprendizagem de acordo com a nova geração da Web e padrões de metadados.

Portanto, acredita-se que o uso de ontologias é de extrema importância para representação dos conceitos relacionados a objetos de aprendizagem, permitindo assim, o compartilhamento e reuso de modelos de metadados para recursos educacionais. Desta forma, evidencia-se a relevância do presente trabalho para as tecnologias voltadas à educação e para a promoção da nova geração de aplicações Web.

1.3 Metodologia

Para a compreensão do objeto de estudo realizou-se primeiramente uma pesquisa documental, através do levantamento bibliográfico dos conceitos referentes a sistemas educacionais baseados na web, fundamentos e tecnologias da web semântica, sistemas de organização do conhecimento, especificações de metadados e engenharia de ontologias. Após a fundamentação teórica, definiu-se uma metodologia,

ferramentas e linguagens para conceituação, formalização e implementação dos modelos de metadados para objetos de aprendizagem.

O repositório é baseado no modelo de referência SCORM para criação de objetos de aprendizagem. O SCORM é dividido em 4 partes: Visão Geral (Overview), Modelo de Agregação de Conteúdo (Content Aggregation Model), Ambiente de Execução (The SCORM Runtime Environment) e Sequenciamento e Navegação (Sequencing & Navigation). Esta pesquisa ocupou-se apenas com o Modelo de Agregação de Conteúdo, pelo fato deste livro conter todas as especificações necessárias para organização do conteúdo dos objetos de aprendizagem, ou seja, a forma que os LOs são organizados, empacotados e representados. É nesta parte que os metadados são apresentados, elementos fundamentais para catalogação, consulta e intercâmbio dos recursos educacionais.

O repositório também faz uso do padrão SKOS para criação de um esquema de classificação destinado à organização dos objetos de aprendizagem. Este esquema fornecerá uma estrutura de navegação (browsing) e recuperação dos conteúdos por áreas do conhecimento aos usuários e outras aplicações. A especificação SKOS Core possui um guia (SKOS Core Guide) que enfatiza os objetivos principais do padrão e um modelo para expressar uma estrutura básica e esquema de conceitos. (W3C, 2006a).

O desenvolvimento das ontologias foi guiado pela metodologia METHONTOLOGY (CORCHO, 2001; GOMEZ-PEREZ, 1999), que visa sistematizar as etapas de construção de sistemas baseados em conhecimento. Alguns estudos referentes às metodologias para desenvolvimento de ontologias têm sido apresentados por Sure; Studer (2003); Corcho (2001). Após a análise da literatura, optou-se pela Metodologia METHONTOLOGY para representação dos conceitos do Repositório de Objetos de Aprendizagem, devido aos seguintes fatores:

- Possui uma abordagem madura, comparada às metodologias de engenharia de software e engenharia do conhecimento;
- É recomendada pela Foundation for Intelligent Physical Agents (FIPA), permitindo desta forma, interoperabilidade

através de aplicações baseadas em agentes inteligentes;

- Utilizada na construção de diversas ontologias em domínios diferentes;
- Possui uma descrição detalhada das etapas de desenvolvimento;
- É Independente de aplicação, assim pode-se modelar qualquer domínio;
- Possui tarefas que enfatizam o reuso de outras ontologias.

Estes motivos reforçam uma concepção que esta metodologia é ideal para modelar o repositório de objetos de aprendizagem, pois permite desenvolver várias aplicações em diversos domínios, sendo assim uma metodologia consolidada.

Neste prisma, criou-se o modelo ontológico seguindo os componentes da METHONTOLOGY: Conceitos, Relações, Instâncias, Constantes, Atributos, Axiomas Formais e Regras. Todavia, a metodologia foi aplicada de acordo com as especificidades do domínio, realizando assim algumas adequações nas tarefas propostas pela METHONTOLOGY. Realizou-se reuso de algumas ontologias criadas para o padrão LOM e propostas de mapeamento do modelo SCORM para ontologias. Após conceituação do modelo, utilizou-se a ferramenta Protégé para formalização do domínio através da linguagem OWL. Utilizou-se também o *framework* Jena para manipulação das ontologias em OWL e a ferramenta RACER, que realiza raciocínios para validação e verificação da consistência da ontologia.

1.4 Estrutura da Dissertação

A dissertação está organizada em duas partes. A primeira parte refere-se aos fundamentos teóricos do objeto de estudo, é composta pelos capítulos 1, 2, 3 e 4. O capítulo 1 apresenta uma breve introdução, objetivos, justificativa e métodos da pesquisa. O capítulo 2 é destinado a discutir os conceitos sobre tecnologias educacionais baseada na Web, apresentando definições e características dos Sistemas de Gerenciamento de Aprendizagem e Objetos de Aprendizagem Reutilizáveis. O capítulo 3

apresenta a Web Semântica e fundamentos sobre desenvolvimento de ontologias. O capítulo 4 enfoca os fundamentos da classificação e organização do conhecimento, sob um ponto de vista histórico e teórico, apresentando ainda aplicações no contexto da Web Semântica.

A segunda parte refere-se ao desenvolvimento do modelo ontológico, apresentando as etapas e atividades de especificação, conceituação e implementação e considerações finais. Esta segunda parte é composta pelos capítulos 5, 6, 7, 8. No capítulo 5 é apresentada a conceituação da ontologia, ou seja, a captura e descrição dos componentes (classes, propriedades, relacionamentos, atributos e axiomas). No capítulo 6 é apresentado a formalização e implementação do modelo ontológico. O capítulo 7 aborda perspectivas para trabalhos futuros e apresenta ainda alguns limites do presente trabalho. Finalmente, o capítulo 8 discute as conclusões e alguns resultados parciais da pesquisa, enfocando as experiências obtidas no manuseio de ferramentas e linguagens para formalização de ontologias e as concepções acerca das novas aplicações para gerenciamento da aprendizagem na Web.

2 TECNOLOGIAS EDUCACIONAIS BASEADA NA WEB

Neste capítulo serão abordados alguns fundamentos relevantes sobre Sistemas de Gerenciamento de Aprendizagem e Objetos de Aprendizagem, apresentando algumas definições, características, requisitos e especificações para desenvolvimento de aplicações, e alguns exemplos de aplicações existentes.

2.1 Sistemas de Gerenciamento de Aprendizagem

Para que o processo educacional seja viabilizado com uso de tecnologias, devem ser consideradas as principais ferramentas existentes no contexto da Informática na Educação. Blogs, Softwares Educacionais, Ambientes de Ensino à Distância e Jogos Educativos, são alguns exemplos de aplicações relevantes para sustentação do ensino baseado em tecnologias da informação e comunicação. Dentre estas ferramentas, será focado os Ambientes de Ensino à Distância, mas precisamente os Sistemas de Gerenciamento de Aprendizagem, apresentando algumas definições, características e tecnologias associadas.

Sistemas de Gerenciamento de Aprendizagem (Learning Management Systems - LMS) são sistemas especializados de Tecnologia de Aprendizagem, baseado no estado da arte da Internet e tecnologias da Web que provê educação e treinamento seguindo o paradigma da educação à distância e aberta. O projeto e implementação de um LMS incorpora uma variedade de componentes administrativos, organizacionais, instrucionais e tecnológicos (IEEE LTSC, 2001a). Os LMSs são destinado a organizar e prover acesso a serviços de aprendizagem *on-line* para estudantes, professores e administradores. Estes serviços usualmente incluem controle de acesso, provisão de conteúdo de aprendizagem, ferramentas de comunicação, e organização de grupos de usuários. Outro termo que sempre é usado como sinônimo de LMS é Ambiente Virtual de Aprendizagem (BROISIN et al, 2005).

Conforme Rapuano e Zoino (2005), o conceito de Sistema de Gerenciamento de Aprendizagem, inclui um sistema de autoria para provê

uma estrutura que pode ser usada rapidamente para criar, modificar, e gerenciar conteúdos para um longo alcance de aprendizagem adequado às necessidades de rápidas mudanças de requisitos de negócios. Alguns destes sistemas de softwares também provêem suporte à interação entre os participantes do espaço de aprendizagem, um mecanismo para fornecimento de materiais de cursos na Web, componentes administrativos que permitem instrutores registrar a trajetória de estudantes, monitorar seu progresso, e componentes colaborativos como *chats*, fórum, e-mail, mural etc.

Assim, pode-se definir que um Sistema de Gerenciamento de Aprendizagem é um termo utilizado para conceituar uma aplicação que permite administrar recursos existentes em um ambiente de ensino mediado por computador baseado na Web. Através do LMS, pode-se gerenciar usuários, cadastrar cursos, realizar avaliações, inserir conteúdos, interagir entre os participantes do sistema através de fóruns e *chats*. Funciona como ambiente de ligação entre os conteúdos de um determinado curso, interação entre grupo de aprendizes e tutores, além de funcionar como uma ferramenta de manutenção de todo o sistema de ensino: gerenciar notas, formar grupos, adicionar cursos, cadastrar notícias etc.

Observam-se alguns pontos em comum existente nas definições de acordo com a literatura. Um ponto central é o princípio da colaboração e cooperação para promoção do aprendizado em um ambiente computadorizado e a noção de compartilhamento de conteúdos como conceito fundamentado na interatividade no processo de ensino-aprendizagem e reuso do conhecimento. Portanto, o LMS pode ser utilizado em diversos contextos organizacionais: no ensino semi-presencial em instituições de ensino e para treinamento em ambientes corporativos e órgãos públicos.

Estudos realizados por Avgeriou et al (2003) têm proposto uma definição de padrões de projeto para o desenvolvimento de LMS tomando como base as vantagens de padrões de projetos de sistemas hipermídia, já que os LMSs são essencialmente aplicações hipermídia que necessitam de modelos para desenvolver interfaces, navegação e conteúdo. Assim a definição de requisitos é fundamental para a especificação de projetos

instrucionais utilizando as tecnologias multimídias em um ambiente de rede distribuído. Os autores apontam para a problemática de se desenvolver um sistema para aprendizagem, devido à complexidade dos sistemas hipermídias e as variáveis existentes no manuseio da ferramenta por parte dos usuários, em especial os aprendizes, foco do sistema de ensino baseado em computadores.

Segundo Avgeriou et al, (2003) os usuários de LMSs podem ser classificados como: aprendizes, instrutores e administradores.

- Os aprendizes usam o sistema para participar do processo educacional. De fato, os aprendizes são os usuários focais do LMS, assim, estes sistemas são desenvolvidos para satisfazer algumas de suas necessidades e resolver seus problemas;
- Os instrutores, sendo os professores e seus assistentes usam o sistema em para treinar, supervisionar, assistir e avaliar os estudantes (isto é, notificar assuntos importantes em uma mural de notícias, empregar discussões em um fórum eletrônico, comunicar e trocar mensagens pessoais com os estudantes, coletar, acessar e retorno de entregas).
- Os administradores do sistema, responsáveis pelo suporte a todos os usuários do sistema e protege seu próprio status operacional.

Através destas considerações, um projeto de LMS deve tomar como referencial as entidades básicas para desenvolvimento de LMS: tutores, alunos e administradores, sendo estes os principais usuários do sistema e cada qual com seu próprio ambiente. Cada qual com suas atividades específicas para promoção do aprendizado baseado na Web. Conforme Mc cmarck; Jones apud Avgeriou, et al (2003) um LMS oferece serviços para satisfazer as necessidades específicas instrucionais e ou automatizar (parcialmente ou totalmente) projetos instrucionais. O LMS pode suportar desenvolvimento e execução de quatro tarefas básicas via uma simples, amigável e uniforme interface de usuário:

- Distribuição de informação, isto é anúncios de tipos de datas, calendário, glossário, etc;
- Gerenciamento de material de aprendizagem, isto é, customização de interface de usuário para as necessidades do instrutor, atualizando, material de aprendizagem;
- Oferta de facilidades de comunicação múltiplas, isto é, comunicação assíncrona e síncrona;
- Gerenciamento de classes, isto é mantendo estudantes on-line, trajetórias de aprendizes, gerenciamento de perfis de líderes.

Avgeriou et al. (2003) considera as questões apresentadas, como problemas básicos para projetos de LMS a serem resolvidos, desta forma os autores utilizam uma metodologia em cima dessas problemáticas. Outro fator a ser considerado refere-se ao compartilhamento de conteúdos entre diferentes LMSs. Conforme Motelet e Bailoan (2004) algumas iniciativas especificam estruturas padronizadas baseada em XML para descrever diferentes níveis de LMS como metadados para projetos de aprendizagem, pacotes de conteúdos, sequenciamento de conteúdos, perfis de estudantes e projetos de aprendizagem. Padrões como AICC, IMS, SCORM, LOM, tentam garantir compatibilidade *plug-and-play* e habilita interoperabilidade, acessibilidade, reusabilidade de conteúdos de aprendizagem baseada na Web. Estes padrões serão explicados na seção 2.3, que abordará as principais especificações para conteúdos de aprendizagem.

Mesmo com todas essas iniciativas de padronização, algumas limitações no desenvolvimento da nova geração de LMS são visíveis. Sessink et al (2003) ressalta que estes padrões não atentam para alguns requisitos funcionais, nem mesmo a mais nova versão do SCORM 1.3. Os requisitos identificados pelos autores são: adaptabilidade, recuperação de histórico e estado, comparação de resultados, trajetória para pesquisa pedagógica, uma base de dados de referência compartilhada e um cenário de problema de base de dados.

Adaptabilidade refere-se a sistemas adaptativos que fornecem a informação para usuário e permite guiar o usuário em um espaço de informação para apresentar o mais relevante material, capturando um

modelo de objetivos de usuários, interesses e preferências. Recuperação de histórico e estado refere-se a recuperação dos estados e o histórico de projetos de aprendizagem. Comparação de resultados permite que os usuários comparem seus resultados com de outros usuários, criando assim um ambiente de competitividade. Trajetória para pesquisa pedagógica refere-se ao desenvolvimento de navegação interna de objetos de aprendizagem, serve para acompanhar a trajetória de aprendizes no sistema. Base de dados de referência compartilhada é criar bases de conhecimento para atribuir referências, definições e explicações adicionais aos objetos de aprendizagem. Cenário de problema de base de dados é a criação de uma base de dados que possa fornecer um mecanismo de busca compartilhada de objetos de aprendizagem (SESSINK et al, 2003). Percebe-se desta forma, que projetos de LMSs devem necessariamente atender a estes requisitos, principalmente o suporte a objetos de aprendizagem. Os aspectos conceituais e padrões técnicos referentes aos Objetos de Aprendizagem serão descritos na próxima seção

2.2 Objetos de Aprendizagem

Nesta seção serão abordados conceitos relevantes acerca dos Learning Objects (LO) ou Objetos de Aprendizagem, assim como serão apresentadas definições, abordagens e características incorporadas à criação e distribuição de conteúdos educacionais na Web.

Na literatura existem diversas definições de LO, não havendo um conceito aceito universalmente. Objetos de Aprendizagem, conforme o IEEE Learning Technology Standards Committee (LTSC) (2006) é definido como:

[...] qualquer entidade digital ou não digital que possa ser usada, re-usada ou referenciada durante a aprendizagem suportada por tecnologia. Exemplos de aprendizagem suportadas por tecnologias incluem sistemas de treinamento baseado em computadores, ambientes interativos de aprendizagem, sistemas inteligentes de instrução auxiliada por computador, sistemas de educação à distância, e ambientes colaborativos de aprendizagem. Exemplos de Objetos de Aprendizagem incluem conteúdo multimídia, conteúdo instrucional, objetivos de aprendizagem, ferramentas de software e software instrucional, e pessoas, organizações, ou eventos referenciados durante a aprendizagem suportada por tecnologia.

Cisco citado por Muzio, Heins e Mundell (2006) define Objeto de Aprendizagem com um granular, pedaço reusável de informação que são independentes de mídia. Os autores mencionam também os termos Objetos Educacionais, Objetos de Conteúdo, Componentes de Treinamento, Pedacos e pepita (nuggets). South e Monson apud Muzio, Heins e Mundell (2006) usam o termo Objeto de mídia, definindo tal objeto como mídia digital que é designada e/ou usada para propósitos instrucionais, variando de mapas e cartas às demonstrações vídeo e às simulações interativas.

O objeto de aprendizagem é uma maneira nova de pensar sobre conteúdo de aprendizagem. Tradicionalmente, o conteúdo vem em diversos pedaços enquanto os objetos de aprendizagem são unidades muito menores da aprendizagem, variando tipicamente de 2 a 15 minutos (WISCONSIN ONLINE RESOURCE CENTER apud POLSANI, 2003).

L'Allier citado por Polsani, (2003) afirma que um objeto de aprendizagem é definido como a menor experiência estrutural independente que contém um objetivo, uma atividade de aprendizagem e uma avaliação. O mesmo explica que o “objetivo” é um elemento de um componente estrutural do objeto de aprendizagem que é uma indicação que descreve o resultado baseado em critério pretendido de uma atividade de aprendizagem. A “atividade de aprendizagem” é um elemento de um componente estrutural do objeto de aprendizagem que ensina um objetivo. A “avaliação” é um elemento de um componente estrutural do objeto de aprendizagem que determina o objetivo a ser alcançado.

Merrill (2006) usa o termo objetos do conhecimento e introduz também o termo componente instrucional. Esta definição é centrada nos modelos mentais, ou seja, nas estruturas cognitivas que referencia as inter-relações entre os componentes de conhecimento. Portanto, o autor parte do princípio sustentado por Gagné que os componentes de conhecimento são os fatos, conceitos, passos (procedimentos) e princípios (GAGNÉ apud MERRIL, 2006).

O projeto ARIADNE usa o termo “documentos pedagógicos” (ARIADNE Apud WILEY, 2002). O projeto NSF-funded Educational Software Components of Tomorrow (ESCOT) usa o termo “componentes de software educacional” (ESCOT apud WILEY, 2002), enquanto a iniciativa Multimedia

Educational Resource for Learning and On-Line Teaching – (MERLOT) conceitua como “materiais de aprendizagem online” (MERLOT Apud WILEY, 2002). A Apple Learning Interchange conceitua simplesmente como “recursos” (ALI apud WILEY, 2002).

A definição proposta por Wiley (2002) é baseada no princípio da reusabilidade. Para o autor, objeto digital é qualquer recurso digital que possa ser reusado para dar suporte à aprendizagem ou qualquer recurso, grande ou pequeno, que possa ser entregue sob demanda através da rede. Pequenos recursos digitais reusáveis incluem imagens, fotos, vídeos pré-gravados, ou pedaços de áudio, pequenos pedaços de texto, animações, e aplicações Web entregue sob demanda, como, Calculadoras Java. Exemplos de grandes recursos digitais reusáveis incluem páginas Web que combinam texto, imagens e outras mídias ou aplicações para entrega completa de experiências, tais como eventos instrucionais completos. A idéia central dos Objetos de Aprendizagem é quebrar o conteúdo instrucional em pequenos pedaços que possam ser reutilizados em diferentes ambientes de aprendizagem.

As definições da IEEE-LTSC, Wiley (2002), Cisco baseia-se na reusabilidade da aprendizagem. O reuso de componentes de software é um fundamental conceito no paradigma orientado a objetos, desta forma os LOs são sustentado nesta filosofia adaptada para os projetos instrucionais. Wisconsin Online Resource Center e Wiley (2002), sustentam a idéia de pedaço de conteúdo, devido ao fato da modularidade existente no próprio paradigma da orientação a objetos. Através da divisão em módulos, os recursos de aprendizagem podem ser utilizados, reutilizados e compartilhados no intuito de atingir objetivos pedagógicos propostos por projetos instrucionais em contextos diferenciados. As demais definições expostas apenas acrescentam as ideais de “componentes” e/ou “materiais” de aprendizagem assistido por computador. Qualquer arquivo digital pode ser considerado um LO, desde que esteja vinculado a uma forma e uma relação de contextualização educacional, um objeto digital ou mídia isolada não pode ser considerado um LO.

Longmire (2006) apresenta alguns argumentos para a promoção do reuso de LO:

- **Flexibilidade:** Se material é designado para ser usado em múltiplos contextos, este pode ser reusado muito mais facilmente do que o ser reescrito para cada contexto novo.
- **Facilidade para Atualização, busca e gerenciamento de conteúdo:** *tags* de metadados facilitam uma atualização rápida, busca e gerenciamento do conteúdo por filtragem e seleção apenas do conteúdo relevante para um dado propósito.
- **Customização:** Quando as necessidades individuais ou organizacionais requerem uma customização de conteúdo, o objeto de aprendizagem facilita uma abordagem *just-in-time* para customização. Os objetos de aprendizagem modulares maximiza o potencial do software que personaliza o conteúdo permitindo a entrega e o recombinação do material no nível de granularidade desejado.
- **Interoperabilidade:** A abordagem de objetos permite que as organizações ajustem as especificações respeitando o projeto, desenvolvimento, e apresentação de objetos de aprendizagem baseados nas necessidades organizacionais, ao segurar a interoperabilidade com outros sistemas de aprendizagem e contextos.
- **Facilitação da aprendizagem baseada na competência:** A abordagem da competência baseada na aprendizagem foca a aprendizagem como intersecção das habilidades, do conhecimento, e das atitudes dentro de modelos de competência. Apesar desta abordagem tem conquistado grandes interesses entre os educadores e patrões no ambiente corporativo, o desafio constante é a implementação de uma competência baseada na aprendizagem é a falta de conteúdo apropriado para que seja suficientemente modular para ser verdadeiramente adaptativo. Colocar etiquetas nos objetos de aprendizagem granular permite uma abordagem competência baseada no casamento dos metadados do objeto com as competências individuais

- **Aumento do valor de um Conhecimento:** O valor do conteúdo é aumentado cada vez que é mais reutilizado. Isto reflete em melhorias significativas ao longo do tempo.

Abordagens, paradigmas e teorias associadas a projetos educacionais com base nos LOs permitem um referencial teórico sobre os principais conceitos para a compressão e modelagem do domínio. Uma abordagem conceitual dos princípios e fundamentos de um LO é proposta por Polsani (2003), como base na reusabilidade. Para Polsani (2003) a reusabilidade é um princípio que serve como fundamento para definição de LO. Um LO pode aproveitar-se da flexibilidade, escalabilidade e adaptabilidade oferecida pela tecnologia da informação apenas quando o objeto é predisposto para o reuso por múltiplos desenvolvedores em vários contextos instrucionais. Para conseguir uma reusabilidade genuína o desenvolvimento e operação de LOs devem ser considerados mutuamente processos exclusivos. A separação entre criação e distribuição requererá que os LOs não favoreçam uma ou outra metodologia instrutiva. Um dos maiores problemas que causam confusão a respeito dos LOs é a influência excessiva de teorias de projetos instrucionais.

Como a terminologia e a prática do projeto instrucional sugerem, é necessária a geração efetiva de *frameworks* instrucionais para conseguir objetivos gerais de aprendizagem. Conseqüentemente teorias de projetos instrucionais, podem oferecer uma grande contribuição para LOs, mas não podem ser o ponto de partida para o processo de desenvolvimento. Uma clara compressão da necessidade de separar o desenvolvimento do objeto e o uso instrucional dos LOs pode ser visto se considerar o primeiro como uma estratégia e o segundo como tática. Estratégia é uma base plana para concluir um objetivo geral, neste caso aprendizagem para exploração do conhecimento. Tácticas por outro lado são ações concretas ou séries de medidas de participantes que são requeridas e apoio em realizações de objetivos estratégicos. LOs são assets estratégicos que são dispostos e predispostos e cuidadosamente designado e especialmente executado em situações instrucionais para facilitar a aprendizagem efetiva (POLSANI, 2003).

A reusabilidade é realizada através da separação da criação do objeto e seu uso, desta forma, facilita a livre troca de assets de LO entre os desenvolvedores, organizações e instituições. É com este intercâmbio articulado que tal reusabilidade dos LOs adquirem valor; sem este seu valor permaneceria como uma simples imagem digital ou uma página Web. Com base em todos estes princípios fundamentais, Polsani (2003) define LO como uma unidade independente e permanente de um conteúdo de aprendizagem que é predisposto para reuso em múltiplos contextos instrucionais. Esta definição de Polsani tem a reusabilidade como cerne dos LOs em estratégias instrucionais em diferentes contextos educacionais. Silicia; García (2003) avaliam o conceito de objetos reusáveis de aprendizagem acrescentando algumas características essenciais para subsidiar o uso concretos destes recursos em diversos contextos.

A característica para “predisposição do reuso” deve ser mais analisada para produzir propriedades que são mais concretas. Especificações sempre se referem a: 1) durabilidade; 2) interoperabilidade; 3) acessibilidade; e 4) reusabilidade. A primeira das três características é essencialmente de natureza técnica. “Durabilidade” e “interoperabilidade” são características relacionadas à independência de plataformas de software e hardware, no qual pode ser obtida pelas linguagens Web e convenções. A terceira característica, “acessibilidade”, é entendida neste contexto como a capacidade de ser pesquisado e localizado no qual é realizado pela presença de um apropriado registro recuperável de metadado (SILICIA; GARCÍA, 2003).

Silicia e García (2003) concluem então que a reusabilidade é uma essencial das mais importantes características dos LOs. Entretanto, visto que a reusabilidade oferece perspectivas de uso em cenários futuros. Os autores colocam ainda que a reusabilidade deve ser mensurada de acordo com o grau de adequação em cada contexto específico. Portanto, mesmo com todas essas definições, pode-se verificar que todos os autores abordam a importância da reutilização dos LOs em contextos diversos.

2.3 Especificações para Objetos de Aprendizagem

Para a promoção da reusabilidade a padronização é um fator essencial, neste caso, os objetos de aprendizagem possuem padrões de metadados que especificam um esquema comum para compartilhamento

de recursos armazenados em diferentes repositórios na web. Conforme Polsani (2003), nos últimos sete anos houve esforços consideráveis no campo de aprendizagem mediada por computadores para a padronização de elementos de metadados para facilitar um método comum para identificar, buscar e recuperar objetos de aprendizagem. Recentemente, um consenso emergiu entre as várias corporações que lideram estes esforços, incluindo IEEE Learning Technology Standards Committee (LTSC) Learning Object Metadata Working Group, IMS Global Learning Consortium, Inc., e Dublin Core Metadata Initiative. Atualmente estas organizações têm se preocupado em desenvolver especificações no intuito de prover a integração de diferentes tipos de conteúdos e sistemas. Iniciativas como a LOM, IMS e SCORM são propostas de padrões empregados para descrever conteúdo e funcionalidades de materiais de ensino na Web.

O Metadado de Objetos de Aprendizagem (LOM) tem como proposta facilitar a pesquisa, evolução, aquisição e uso de objetos de aprendizagem, por aprendizes ou instrutores. O modelo de dados LOM tem alcançado um nível de estabilidade e reconhecimento internacional requisitado para a implementação em grande escala de infraestruturas de ambientes de aprendizagem (IEEE, 2006). O LOM define elementos de metadados como: Título, Palavra-chave, Classificação, Ciclo de Vida etc. A especificação IMS define um modelo para empacotamento, apresentação e navegação de conteúdos de aprendizagem. O padrão Dublin Core criado pela Dublin Core Metadata Initiative (DCMI), que se dedica a promover a adoção generalizada de padrões de metadados interoperáveis e desenvolver vocabulários de metadados especializados para a descrição de recursos para poder criar sistemas de busca de informação mais inteligentes. Este padrão apresenta elementos como: Autor, Título, identificador, Descrição, Data etc. para marcação de documentos na Web. Estes padrões podem ser expressos em linguagem XML e linguagens derivadas que possuem a mesma sintaxe, como RDF.

Estes objetos em ambientes de ensino a distância, ficam armazenados em um repositório no qual devem ser representados para agregar os pacotes de conteúdo. Yang, Yu e Chen (2004) propõem o conceito de Sistema de Gerenciamento de Conteúdo de Repositório (CRMS,

Content Repository Management System), como uma coleção de materiais de ensino com os objetos de aprendizagem e agregação de conteúdo baseado em uma aplicação Web, implementado para professores criarem e organizarem seus materiais de ensino. A representação dos conteúdos dos objetos de aprendizagem dá-se através do uso de padrões de metadados para possibilitar a classificação e a consulta por professores e aprendizes. Allert; Richter e Nejdl (2005, p.12) destacam alguns valores pedagógicos destas tecnologias, colocando que um aspecto importante sobre os objetos de aprendizagem é que são caracterizados por teorias de aprendizagem, paradigmas educacionais (como construtivismo e cognitivismo) e princípios instrucionais e estruturas conceituais.

Nesta perspectiva, convém enfocar o padrão SCORM (Shareable Content Reference Model), sendo este um conjunto unificado de padrões e especificações para conteúdo, tecnologias e serviços para educação à distância especificados pela ADL (Advanced Distributed Learning). Conforme Rouyet; Martín (2005), SCORM é um conjunto de padrões técnicos que permitem aos sistemas de aprendizagem baseado na Web, encontrar, importar, compartilhar, reusar e exportar conteúdos de informação de maneira normalizada. Assim, a proposta do SCORM é ser um padrão internacional para desenvolvimento de cursos para Educação à Distância, sendo este desenvolvido a partir da união dos padrões para implementação de repositórios de objetos de aprendizagem em nível mundial, como o AICC (Aviation Industry CBT Comitee), o IMS (IMS Global Learning Consortium, Inc.) e o ARIADNE (Aliance of Remote Instructional Authoring & Distribution).

3 WEBSEMÂNTICA E ONTOLOGIAS

A criação da Web em 1994 causou diversos avanços na produção e publicação de documentos em diversos formatos. Através da HTML (Hypertext Markup Language), a linguagem padrão para edição de documentos de hipertexto, qualquer pessoa poderia criar um documento contendo texto, gráficos, multimídia e vínculos a outros documentos, bastava apenas ter conhecimentos básicos da linguagem. Para publicação destes documentos, o usuário necessitava de um espaço em disco em um servidor e enviar via um padrão de protocolo, por exemplo, FTP, para seu documento ser visualizado em qualquer parte do mundo.

A Web é baseada ainda no HTML, no qual descreve como a informação é mostrada e colocada em uma página Web para que os seres humanos leiam. De fato, Web tornou-se como um meio para a exposição da informação diretamente aos seres humanos; não houve nenhuma ênfase real em estabelecer a potencialidade para a compreensão e processamento da informação baseada na Web por máquina. O HTML não é capaz de ser explorado diretamente por técnicas da recuperação de informação; então o processamento da informação na Web é amplamente restrito a buscas manuais por palavras-chave (ALESSO; SMITH, 2006, p.13).

Este modelo de publicação permanece até os dias atuais, mesmo com o advento das linguagens que executam ao lado do servidor destinadas a criação de páginas dinâmicas e integração com banco de dados, o paradigma de desenvolvimento de Web sites ainda é baseado na apresentação gráfica de conteúdos. Conforme Alesso; Smith (2006), a World Wide Web tem sua origem na idéia do hipertexto, a Web é focalizada nos dados textuais e enriquecida através de ilustrações e materiais audiovisuais. O paradigma do *status quo* da Web é centrado na interação do cliente-servidor, que é um relacionamento fundamental assimétrico entre os fornecedores de conteúdo de hipertexto da Web (o servidor) e usuários que essencialmente lêem textos ou preenchem formulários (os clientes).

A Web Semântica foi inicialmente proposta por Tim Berners-Lee, criador da Web e atual diretor da World Wide Web Consortium (W3C). A Web Semântica é uma nova geração da Web baseada em uma estrutura que permitirá atribuir significado a páginas Web. Berners-Lee; Hendler; Lassila (2001) acreditam que o uso dos padrões das tecnologias eXtensible Markup

Language (XML), Resource Description Framework (RDF) e técnicas para representação do conhecimento possibilita uma melhoria significativa na indexação de conteúdos na rede mundial de computadores.

A nova Web permite ainda que as informações criadas possam ser interpretadas por máquinas, contrariamente ao modelo atual destinado a leitura por humanos. Para tal, utiliza-se uma marcação semântica expressa em documentos XML e regras de inferência para atribuição de uma camada lógica que descreve propriedades dos objetos em vocabulários e permite a consulta em linguagem natural. Assim, muitas aplicações terão uma completa interoperabilidade de dados, acesso a documentos em diversos formatos e independente de plataforma através de protocolos comum. Esta arquitetura define um ambiente onde os agentes de software vasculham página por página possam prontamente realizar tarefas sofisticadas aos usuários (BERNERS-LEE; HENDLER; LASSILA, 2001).

A Web Semântica oferece aos usuários a habilidade ao trabalho de compartilhado do conhecimento construindo representações significativas novas na Web. A pesquisa em Web Semântica desenvolveu-se das tradições linguagens de Inteligência Artificial e ontologias e oferece processamento automatizado de metadado compreensível por máquina. Agentes na Web Semântica podem utilizar metadados, ontologias, e lógica para realizar suas tarefas. Agentes são pedaços de software que trabalham autonomamente e proativamente na Web executando certas tarefas. Em muitos casos, agentes coletam e organizam informação. Agentes recebem algumas tarefas para executar, procurar informação e recursos Web e comunicar com outros agentes, para realizar certas tarefas (ALESSO; SMITH, 2006).

O novo modelo arquitetural da Web Semântica possui diversas camadas provendo um *framework* que permite fornecer, compartilhar e reutilizar dados na grande rede. Esta arquitetura abrange as linguagens, o uso das ontologias para representação do conhecimento e ferramentas para desenvolvimento de aplicações. A figura 1 demonstra a visão da W3C e as camadas que formam a Web Semântica, em seguida é explicada cada camada.

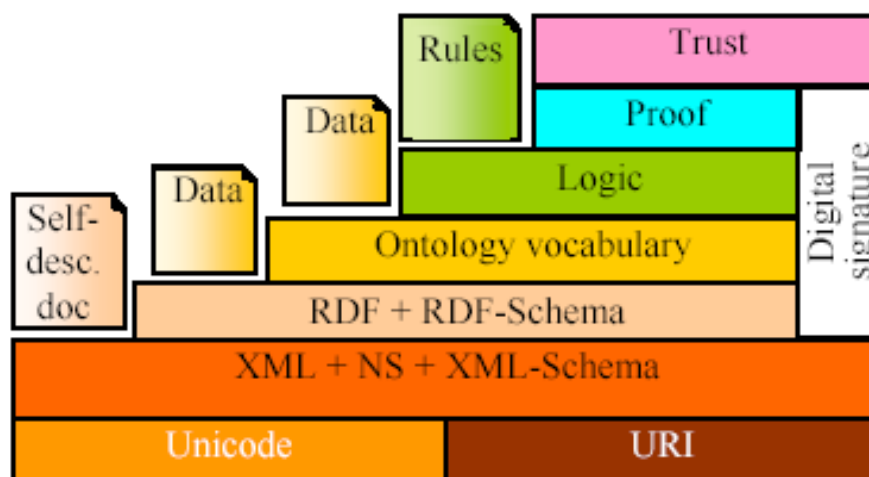


Figura 1: Arquitetura da Web Semântica

Fonte: W3C

As camadas Unicode, URI, XML e NS (Name Spaces) são camadas estruturais e sintáticas. Unicode refere-se a caracteres universais que podem ser visualizados de forma padronizada em diferentes idiomas. O Unicode fornece um único número para cada caractere, independente de plataforma, software, língua. Este padrão tem sido adotado por diversas organizações do setor de informática tais como a Apple, HP, IBM, JustSystem, Microsoft, Oracle, SAP, Sun, Sybase, Unisys e muitos outros. O Unicode é necessário para padrões modernos tais como o XML, Java, ECMAScript (JavaScript), LDAP, CORBA 3.0, WML, etc. É também suportado por muitos sistemas operacionais, todos os *browsers* modernos e muitos outros produtos (UNICODE, 2006).

URIs (Uniform Resource Identifiers) são para nomear, referenciar e localizar recursos na Web. Conforme a W3C (2006), uma URI é uma seqüência compacta de caracteres que identificam recursos na Web: documentos, imagens, arquivos para *download*, serviços, caixas postais eletrônicas, e outros recursos disponíveis e acessados via métodos HTTP. Os endereços comuns existentes na Web são as URIs, como por exemplo: <http://www.ufma.br/> que faz referência ao espaço em um servidor dedicado a armazenar e disponibilizar conteúdo referente a UFMA.

XML *namespaces* são mecanismos simples para criação de nomes únicos e globalmente originais para os elementos e atributos de uma linguagem de marcação. Isto é importante por duas razões: Para evitar o

conflito em significado de nomes idênticos em linguagens de marcação diferentes e permitir que estas sejam misturadas junto sem ambigüidade (DACONTA; OBRST; SMITH, 2003), Segundo a W3C (2006b), um XML namespace é identificado por uma referência URI, os nomes do elemento e do atributo podem ser colocados em um namespace de XML usando os mecanismos descritos nesta especificação. Pode-se atribuir um nome para um elemento criando-se assim um elemento próprio, definido pelo desenvolvedor. Os *namespaces* são escritos usando o formato URI, utilizando o protocolo HTTP (TESCH Jr, 2002).

Exemplo de *namespaces*:

xmlns = "http://www.netclass.ufma.br/MeusObjetos"

Esquemas XML (XML Schema) são documentos que contêm as informações sobre como devem ser estruturados os documentos XML, que se associam a ele. Nesta camada, são definidos os tipos e atributos que validarão os dados contidos no documento XML. Existem dois tipos de esquemas: DTD (Document Type Definition) e XML Schemas. Uma DTD pode incluir um conjunto de entidades, definição de atributos, declarações em marcações (TESCH JR, 2002). Os XML Schemas expressam vocabulários compartilhados e permitem que as máquinas realizem regras feitas por pessoas. Fornecem meios para definir a estrutura, o conteúdo e a semântica dos documentos XML com mais detalhes. O XML Schema foi aprovado como uma recomendação da W3C em 2 de maio 2001 e uma segunda edição que incorpora muitas erratas foi publicada em 28 de outubro 2004 (W3C, 2006b).

O Resource Description Framework (RDF) e a OWL Web Ontology Language (OWL) são algumas especificações recomendadas pela W3C para representação de ontologias. O RDF é usado para representação de informações e troca de conhecimentos na Web. O RDFS (RDF Schema) permite a criação de vocabulários básicos em RDF e nesse nível existe a noção de classe, hierarquia e relações, também chamadas de propriedades. A OWL é usada para publicar e compartilhar conjunto de termos, as ontologias, para suportar buscas avançadas, agentes de software e gerenciamento do conhecimento (W3C, 2005). Na seção 3.1 estas

linguagens e sua importância para representação de ontologias será melhor abordada.

O vocabulário de Ontologia é destinado a armazenar os modelos de domínios através de representação do conhecimento. Permite a criação de hierarquia de classes e subclasses e suas relações semânticas. As ontologias consistem tipicamente de definição de conceitos relevantes para um domínio, suas relações, e axiomas sobre estes conceitos. Várias linguagens e sistemas são definidos para representação de ontologias (PANTELEYEV. et al, 2002, p.2).

A camada Logic possibilita definir regras para o tratamento da informação descrita nas camadas anteriores. Proof verifica a consistência dos dados acessíveis na Web Semântica. Digital Signature garante a procedência de um documento ou definição. Trust corresponde à última camada que visa permitir a criação de agentes de software que trabalhem apenas com dados confiáveis na Web Semântica.

Nos tópicos seguintes, serão apresentadas de forma mais aprofundada, as linguagens, as técnicas para construção de ontologias e as ferramentas de apoio ao desenvolvimento de aplicações para Web Semântica.

3.1 Linguagens

As linguagens para desenvolvimento da Web Semântica são basicamente destinadas à apresentação de conteúdos através de marcações (*tags*) e validados com esquemas que definem os elementos e os atributos em uma estrutura hierárquica de um documento. Na Web tradicional, a linguagem para marcação de documentos, a HTML, existe *tags* predefinidas, o que prejudica a definição conceitual de elementos em um documento, por não permitir a personalização, servindo apenas para a apresentação do conteúdo no navegador Web. A linguagem padrão da Web é limitada à descrição estrutural do documento, não permitindo a marcação semântica do conteúdo do documento. Na HTML, descrevem-se os dados contidos no documento através de elementos úteis para marcação. Cada elemento representa um tipo de marcação, que pode ser textual (títulos,

parágrafos, cabeçalhos), mídia, vínculos. Desta forma, o propósito da HTML é apenas apresentar o conteúdo, não permitindo construir seus próprios elementos destinados à marcação de algum conceito do mundo real. Devido a esta problemática, criou-se a XML (eXtensible Markup Language) para suprir esta necessidade de marcação semântica de elementos contidos em um documento para Web.

As principais linguagens utilizadas para Web Semântica são todas derivadas da linguagem a XML como, por exemplo, XSL, Xpath, Xquery, RDF, OWL e outras. XML é uma recomendação da W3C para gerar linguagens de marcação para necessidades especiais, ou seja, o desenvolvedor pode criar seus próprios *tags* de acordo com os conceitos relativos ao conteúdo do documento.

3.1.1 XML (eXtensible Markup Language)

Conforme a W3C (2006), XML é um simples e muito flexível formato de texto, derivado da SGML (ISO 8879). Originalmente projetado para adequar aos desafios da publicação eletrônica em grande escala, XML também tem um papel importante na troca de uma grande variedade de dados na Web e em outros locais. Para escrever documentos XML é necessário levar em consideração quatro aspectos básicos: a sintaxe, os elementos, atributos e sua validação. Exemplo de documento XML.

```
1 <?xml version="1.0" encoding="iso-8859-1"?>
2 <curso>
3   <area>Ciência da Computação</area>
4   <titulo>Programação em Java</titulo>
5   <professor>Jaime Moura</professor>
6   <modulo>Introdução à Orientação a Objetos</modulo>
7   <aula>A orientação a objetos</aula>
8 </curso>
```

Figura 2: Documento XML

De acordo com o código apresentado na figura 2, XML possui uma sintaxe similar à linguagem HTML, possui a flexibilidade de criação de elementos particulares para marcação de dados. Neste exemplo, criou-se um simples documento para representar um curso na Web. Como foi

explanado na seção anterior, geralmente um documento XML está associado a um esquema, neste caso, diz-se que o documento XML é um documento válido. A figura 3 mostra um código de marcação do exemplo anterior validado por um esquema DTD (DOCUMENT TYPE DEFINITION):

```

1 <?xml version="1.0" encoding="iso-8859-1"?>
2 <!DOCTYPE curso [
3   <!ELEMENT curso (area,titulo,professor,modulo,aula)>
4   <!ELEMENT area      (#PCDATA)>
5   <!ELEMENT titulo    (#PCDATA)>
6   <!ELEMENT professor  (#PCDATA)>
7   <!ELEMENT modulo     (#PCDATA)>
8   <!ELEMENT aula      (#PCDATA)>
9 ]>
10 <curso>
11   <area>Ciência da Computação</area>
12   <titulo>Programação em Java</titulo>
13   <professor>Jaime Moura</professor>
14   <modulo>Introdução à Orientação a Objetos</modulo>
15   <aula>A orientação a objetos</aula>
16 </curso>

```

Figura 3: Documento XML com um esquema DTD

Na linha 1 do código XML declara-se o prólogo XML. Na linha 2 especifica-se o tipo de documento, neste caso “curso”. Na linha 3 declara-se todos os elementos (“area”, “titulo”, “professor”, “modulo”, “aula”) referentes ao elemento “curso”. Na linha 4, define-se o elemento do tipo #PCDATA (W3C SCHOOLS, 2006). Pode-se perceber que as DTDs não são criadas totalmente com a sintaxe XML

XML Schema permite restringir um vocabulário específico e uma estrutura hierárquica específica em documentos XML. O XML Schema é análogo a um esquema de base de dados, que define os nomes da coluna e os tipos de dados em tabelas de base de dados. (DACONTA; OBRST; SMITH, 2003, p.37). Existem dois tipos de documentos: um documento esquema (ou documento de definição) e documentos de múltiplas instâncias de acordo com o esquema. A diferença é que a definição do esquema é um modelo (*template*) e cada instância é uma encarnação desse modelo. Isto demonstra também os dois papéis que um esquema pode proporcionar: um modelo que gere instâncias de um tipo de documento e um esquema para validar e assegurar a exatidão dos documentos. Tanto os esquemas do documento quanto suas instâncias usam sintaxe XML (*tags*, elementos e

atributos). Isto é um dos fatores primordiais para substituir as DTDs, que não usam sintaxe XML. A figura 4 ilustra a forma como os esquemas XML e suas instâncias são validadas.

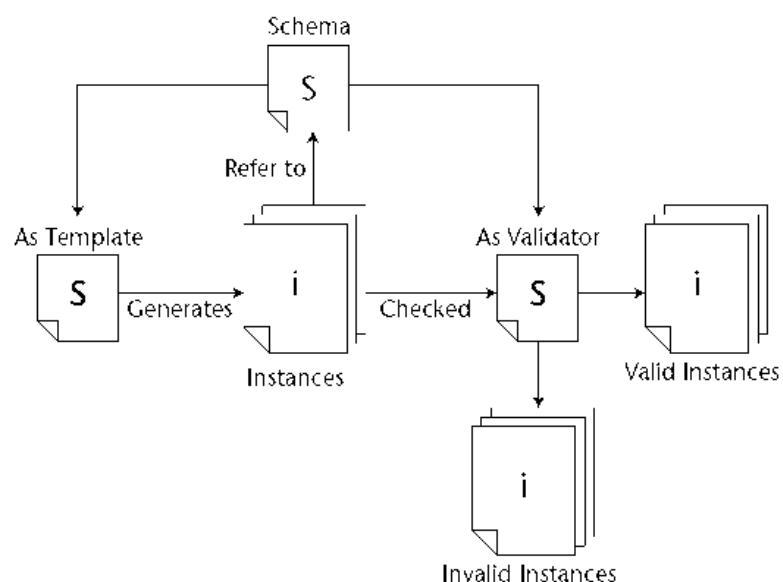


Figura 4: Esquema e Instâncias (DACONTA; OBRST; SMITH, 2003)

XML Schema permite a validação de instâncias para assegurar a exatidão dos campos de valores e estruturas de documentos na hora da criação. A exatidão dos campos é checada; por exemplo, verifica se a quantidade tipada é um inteiro ou um tipo de moeda como decimal. A estrutura do documento é checada para coisas como elemento e nome de atributos requeridos. Todos os documentos XML podem ser checados para validar antes de serem transferidos para outro documento ou sistema.

Na figura 5, é apresentado um exemplo de XML Schema.

```

1 <?xml version="1.0"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
3 targetNamespace="http://www.netclass.ufma.br"
4 xmlns="http://www.netclass.ufma.br"
5 elementFormDefault="qualified">
6 <xs:element name="curso">
7   <xs:complexType>
8     <xs:sequence>
9       <xs:element name="area" type="xs:string"/>
10      <xs:element name="titulo" type="xs:string"/>
11      <xs:element name="professor" type="xs:string"/>
12      <xs:element name="modulo" type="xs:string"/>
13      <xs:element name="aula" type="xs:string"/>
14    </xs:sequence>
15  </xs:complexType>
16 </xs:element>
17 </xs:schema>

```

Figura 5: XML Schema

De acordo com o esquema da figura 5, os elementos definidos no documento são do tipo *string* (xs:string), declara-se um elemento raiz (linha 6) e demais elementos relacionados. Um exemplo de utilização do esquema XSD no documento XML é demonstrado na figura 6.

```

1 <?xml version="1.0" encoding="iso-8859-1"?>
2 <curso xmlns="http://www.netclass.ufma.br"
3 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4 xsi:schemaLocation="http://www.netclass.ufma.br/aula3.xsd">
5   <area>Ciência da Computação</area>
6   <titulo>Programação em Java</titulo>
7   <professor>Jaime Moura</professor>
8   <modulo>Introdução à Orientação a Objetos</modulo>
9   <aula>A orientação a objetos</aula>
10 </curso>

```

Figura 6: Utilização do esquema XSD no documento XML

Em XML podem-se definir diversos tipos de dados, inclusive os complexos, possibilitando desta forma um maior controle das estruturas que compõe os documentos, bem como o intercâmbio e flexibilidade na Web. Outras tecnologias são associadas ao XML, pois a sua extensibilidade permite construir outras linguagens derivadas. Outros padrões também são úteis para apresentação de conteúdo e consultas em árvores de documentos XML baseado em sistemas de arquivos. Para apresentação e transformação de documentos existem a XSL e XSLT, respectivamente. A XSL (eXtensible Stylesheet Language) é uma linguagem utilizada para criação de folhas de estilo para apresentação de documentos XML,

enquanto a XSLT (eXtensible Stylesheet Language Transformation) é uma linguagem para transformar documentos XML em documentos XHTML ou em outros documentos XML.

XSLT, A Linguagem para Transformação, é uma recomendação oficial do Consórcio World Wide Web (W3C). Provê uma flexível, poderosa linguagem para transformação de documentos XML e outras coisas. Estas coisas podem ser documentos HTML, outros documentos XML, arquivos no formato de documentos portáteis (PDF), arquivos, arquivos em Gráfico Vetor Escalável (SVG), arquivos em Linguagem para Modelagem de Realidade Virtual (VRML), código Java, arquivos de texto, arquivo JPEG, ou qualquer coisa que você quiser. Você pode escrever uma folha de estilo XSLT para definir as regras para transformação em um documento XML, e o processador XSLT faz este trabalho (TIDWELL, p.07, 2001).

Pode-se utilizar também CSS (Cascading Style Sheets) para apresentação de conteúdos de documentos XML, criando folhas de estilo similares as usadas em páginas Web feitas em HTML e XHTML. XSLT usa XPath (XML Path Language) para encontrar a informação em um documento XML. XPath é usado para navegar nos elementos e os atributos em documentos XML (W3SCHOOLS, 2006). Um exemplo de uma folha de estilo em XSL para um documento XML é apresentado na seguinte figura 7.

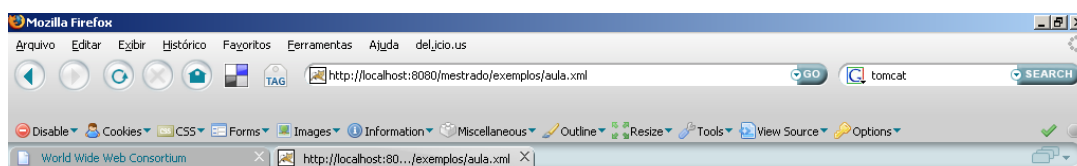
```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <xsl:stylesheet version="1.0"
3 xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
4 <xsl:template match="/">
5   <html>
6     <body>
7       <h1>Cursos Disponiveis</h1>
8       <table border="1">
9         <xsl:for-each select="curso">
10          <tr><h2>Área:<xsl:value-of select="area"/> </h2></tr>
11          <tr>
12            <th>Titulo</th>
13            <td><xsl:value-of select="titulo"/></td>
14          </tr>
15          <tr>
16            <th>Professor</th>
17            <td><xsl:value-of select="professor"/></td>
18          </tr>
19          <tr>
20            <th>Módulo</th>
21            <td><xsl:value-of select="modulo"/></td>
22          </tr>
23          <tr>
24            <th>Aula</th>
25            <td><xsl:value-of select="aula"/></td>
26          </tr>
27        </xsl:for-each>
28      </table>
29    </body>
30  </html>
31 </xsl:template>
32 </xsl:stylesheet>

```

Figura 7: Folha de estilo XSL

O atributo *match* (linha 4) é usado para associar um modelo com um documento XML. O valor do atributo *match* é uma expressão de XPath (isto é *match="/"* define o documento com um todo) (W3SCHOOLS, 2006). Basta adicionar a linha `<?xml-stylesheet type="text/xsl" href="aula4.xsl"?>` para referenciar o arquivo XSL chamado “aula4” no documento XML apresentado na figura 2. A renderização deste documento XML em um navegador comum é mostrada na seguinte figura:



Cursos Disponíveis

Área: Ciência da Computação	
Título	Programação em Java
Professor	Jaime Moura
Módulo	Introdução à Orientação a Objetos
Aula	A orientação a objetos



Figura 8: Apresentação de um documento XML transformado em HTML.

Como visto no exemplo acima, os dados do documento XML foram dispostos em tabelas HTML, aplicando uma folha de estilo que permite apresentar dados tabulares, criando-se assim uma interface mais apresentável ao usuário final. Pode-se ainda atribuir cores, vínculos e demais elementos da linguagem XHTML.

Atualmente as principais linguagens do mercado fornecem suporte a XML, o que permite gerar, interpretar e manipular documentos XML, realizando desta forma a interoperabilidade entre diversas aplicações. Na área de Desenvolvimento Web, linguagens como JSP, PHP, ASP e Perl

possuem bibliotecas para transformação em XSLT e suporte fundamental para criar interfaces para acessar Web Services e fornecer recursos com RSS, ATOM e outras aplicações usadas para em blogs, portais de notícias e repositórios para compartilhar e acessar conteúdos.

3.1.2 RDF (Resource Description Framework)

Resource Description Framework (RDF) é uma linguagem que tem como propósito principal representar informação na Web. RDF utiliza a sintaxe, XML, URIs, um vocabulário e um modelo baseado em grafos desenvolvido e recomendado pela W3C desde fevereiro de 1999 (W3C, 2006). RDF prover um simples, porém poderoso modelo para descrever informações, todavia, a semântica (o significado da informação) é descrito usando RDF Schema (RDFS). O propósito de RDFS é prover um vocabulário XML que pode expressar classes e suas subclasses e seus respectivos relacionamentos, ou seja, propriedades associadas a suas classes (ALESSO; SMITH, 2006).

RDF é uma infra-estrutura que permite codificação, troca e reuso de metadados estruturados e fornece ainda, interoperabilidade através de mecanismos que suportam uma convenção comum de semântica, sintaxe e estrutura (MILLER, 1998). Os vocabulários é um conjunto das propriedades, ou elementos dos metadados, definidos por comunidades de descrição de recursos. A habilidade de padronizar a declaração de vocabulários é incentivar o reuso e a extensão da semântica entre comunidades diferentes de informação. Por exemplo, a iniciativa Dublin Core, que consiste em uma comunidade internacional para descrição de recurso que focaliza a descrição simples de recurso para a descoberta, tem adotado RDF.

RDF é projetado para suportar este tipo de modularidade semântica criando uma infra-estrutura que suporte a combinação de registros de atributos distribuídos. Assim, um registro central não é requerido. Isto permite comunidades declararem os vocabulários no qual podem ser reusados, estendidos e/ou refinados à aplicação para endereçar aplicações ou requisitos descritivos de domínios específicos (MILLER, 1998).

O RDF é um formato para dados que usa um modelo relacional simples que permite dados estruturado ou semi-estruturado ser incorporado, exportado, e compartilhado entre diferentes aplicações. RDF é atualmente construído sob um simples modelo de dados que suporta o processamento da informação em larga-escala. Um documento RDF pode delinear relações precisas entre itens do vocabulário para construir uma gramática de representação. As declarações em diferentes documentos RDF podem ser combinadas para prover muito mais informação e uma poderosa e flexível estrutura de consulta (ALESSO; SMITH, 2006).

Com RDF, os recursos (fontes de dados, páginas Web, documentos multimídias etc.) podem ser representados através dos metadados e identificados por URIs na Web. Estes metadados são designados a serem legíveis por computadores visando assim uma maior semântica na representação de dados, já que na Web é predominantemente composta por dados não-estruturados. Acredita-se que desta forma, pode-se implementar mecanismos mais inteligentes de representação e recuperação de conteúdos na Web.

A linguagem RDF pode ser aplicada em diversos domínios. Pode ser aplicado para descrever propriedades de itens de um *shopping*, tal como preço e disponibilidade; descrever programação de uma lista de eventos na Web; descrever informações sobre páginas Web, tal como conteúdo, autor, data de criação e modificação; descrever e avaliar conteúdo de fotografias; descrever conteúdo para ferramentas de busca e descrever bibliotecas eletrônicas (W3SCHOOLS, 2006).

O modelo RDF é baseado em declarações feitas sobre os recursos que podem ser qualquer coisa associada a URI. O modelo básico RDF produz uma tripla: recurso, propriedade e valor. O recurso é o sujeito que pode ser uma pessoa, um lugar, ou coisa descrita. O recurso pode ser qualquer coisa em um modelo de dados (documentos, usuários, produtos, etc.) e é unicamente identificado por uma URI. A propriedade é o predicado, que pode ser um nome, cidade, título, cor, forma, característica de um sujeito e é também identificada unicamente por uma URI. O valor referente ao objeto é especificado para uma propriedade e pode ser algum tipo de dado RDF válido (ALESSO; SMITH, 2006).

Um recurso (sujeito) tem uma propriedade (predicado) com um valor (objeto) específico. A tripla pode ser representada graficamente em grafos rotulados com nós e arcos. O arco é dirigido do recurso (sujeito) ao valor (objeto), este tipo de grafo na comunidade de IA é conhecido como rede semântica (ALESSO; SMITH, 2006). O exemplo: "O curso tem o módulo Redes Neurais", é uma tripla e pode ser expresso em uma fórmula lógica $P(x,y)$, onde o predicado binário P indica o objeto x para o objeto y , ou seja, *tem módulo (o curso, Redes Neurais)*. Esta tripla pode ser representado graficamente em forma de grafo, com mostra a figura 9 (ALESSO; SMITH, 2006).

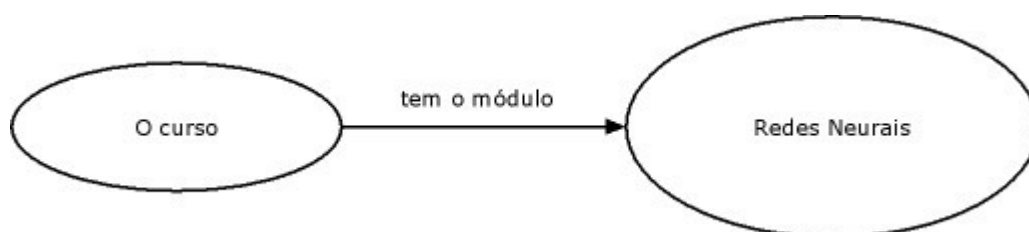


Figura 9: Uma tripla representada em grafo

No exemplo exposto, apresentou-se um simples grafo, com apenas um único relacionamento (propriedade), porém em RDF, pode-se associar diversas propriedades interconectados de acordo com atributo-valor de um ou mais recursos. Estas propriedades são componentes fundamentais para atribuir significados entre os sujeitos e predicados para um modelo de dados.

Tanto RDF quanto RDF Schema (RDFS) usam sintaxe XML. RDF funciona como um excelente complemento a XML, provendo significado na descrição das relações entre recursos em propriedades e valores, codificando triplas em documentos XML e permitindo interoperabilidade entre plataformas que possuem validadores XML. Os elementos básicos de um documento RDF, cabeçalho, sintaxe XML, elemento raiz, *namespaces*, a tripla e o elemento final são demonstrados na tabela 1. (ALESSO; SMITH, 2006).

Partes do Documento	Documento RDF
Cabeçalho-Declaração	da <?xml version="1.0" ?>
Sintaxe XML	
Tag do elemento Raiz	<rdf:RDF
Namespaces XML para rdf e	xmlns:rdf=http://www.w3c.org/1999/02/222-rdf-syntax-
dc, assim como, as URLs onde	ns#xmlns:dc="http://purl.org/dc/elements/1.1/"
são definidas.	
Introdução da tripla (sujeito,	<rdf:Description rdf:about" SUJEITO ">
predicado, objeto) com o	<dc: PREDICADO >" OBJETO "</dc: PREDICADO >
código.	</rdf:Description>
Fim do elemento raiz indica o	</rdf:RDF>
fim do documento RDF	

Tabela 1: Partes de um documento RDF (ALESSO; SMITH, 2006).

Diferente de DTDs ou XML Schemas que fornecem restrições sobre a estrutura de um documento XML, um esquema RDF fornece informações sobre a interpretação das declaração em um modelo de dados RDF. Os conceitos intrínsecos da definição de esquemas RDF são demonstrados na lista abaixo:

- Classes
 - o rdfs:Resource
 - o rdfs:Class
 - o rdfs:Literal
 - o rdfs:Datatype
 - o rdf:XMLLiteral
 - o rdf:Property
- Properties
 - o rdfs:range
 - o rdfs:domain
 - o rdf:type
 - o rdfs:subClassOf
 - o rdfs:subPropertyOf
 - o rdfs:label
 - o rdfs:comment
 - o rdfs:seeAlso

Conforme a W3C (2006), todas as coisas são chamadas de recursos e suas instâncias de classe `rdfs:Resource`. Esta é a classe de tudo. Todas classes restantes são subclasses desta classe. O elemento `rdfs:Resource` é uma instância de `rdfs:Class`.

O `rdfs:Class` refere-se a classe de todas as demais classes. Classes marcam um conjunto de recursos, e possui uma propriedade `rdf:type` (instâncias tem propriedade `rdf:type` para uma classe). Visto que todos os conjuntos de recursos tem uma definição de propriedade `rdf:type` referente a `rdfs:Class` e todos as propriedades recomendadas pela W3C tem `rdf:type` referente a `rdf:Property`. Classes são estruturadas em um subconjunto hierárquico denotados pela propriedade `rdfs:subClassOf` (CHAMPIN, 2006). Uma classe define um grupo de coisas relacionadas a um conjunto associado de propriedades. É um sinônimo para um tipo de conceito ou categoria. Trabalha conjuntamente com `rdf:Property`, `rdfs:range`, e `rdfs:domain` para assinar propriedades para a classe. Requer uma URI com um identificador de atributo `rdf:about` (DACONTA; OBRST; SMITH, 2003).

A classe `rdfs:Literal` é uma classe tal como as *strings* e inteiros. As *strings* textuais são exemplos de literais RDF. Literais podem ser tipadas. A literal tipada é uma instância de uma classe de tipo de dado. A especificação da W3C não define literais simples (W3C, 2006). Representa ainda um valor constante representado como uma *string* de caractere. A revisão da sintaxe RDF/XML tem recentemente adicionado literais para RDF assim pode-se especificar qualquer tipo em XML Schema (como inteiro ou *float*) (DACONTA; OBRST; SMITH, 2003).

O elemento `rdf:Datatype` serve para especificação de tipos de dados. Todas as instâncias de `rdfs:Datatype` corresponde a um modelo RDF de um tipo de dado descrito na especificação de Conceitos RDF da W3C (2006). Segundo o vocabulário de descrição da linguagem RDF Schema da W3C (2006), `rdfs:Datatype` é tanto uma “instância de” quanto uma “subclasse de” do elemento `rdfs:Class` e cada instância de `rdfs:Datatype` é uma subclasse de `rdfs:Literal`. A classe `rdf:XMLLiteral` é a classe dos valores literais da XML. É uma instância de `rdfs:Datatype` e uma subclasse de `rdfs:Literal` (W3C, 2006). `rdfs:XMLLiteral`.

O elemento `rdf:Property` é a classe das propriedades da linguagem RDF e uma instância de `rdfs:Class` (W3C, 2006). Define ainda uma propriedade para a classe e um alcance de valores que podem representá-la e é usada em conjunção com as propriedades `rdfs:domain` e `rdfs:range` (DACONTA; OBRST; SMITH, 2003). De acordo com Champin (2006), as propriedades são recursos usados como predicados de triplas; a semântica da tripla depende claramente da propriedade usada como predicado. RDF considera propriedades como classes de objeto, parecido com as linguagens de modelagem orientadas a objetos, onde propriedades são atributos de classes. Embora o conceito de classe exista em RDF, propriedades podem ser definidas e usadas independentemente das classes.

É importante para entender a diferença chave entre a modelagem de classes em RDFS e a modelagem de classes na programação orientada a objetos. RDFS emprega uma abordagem *bottom-up* para modelagem das classes, enquanto OOP emprega uma abordagem *top-down*. Em orientação a objetos, define-se propriedades após criar as classes. Em RDFS, definem-se as propriedades e declaram-se as classes que fazem parte. Assim em RDFS, parte-se das propriedades para as classes. (DACONTA; OBRST; SMITH, 2003).

A especificação de Conceitos e Sintaxe Abstrata da linguagem RDF descreve o conceito de propriedade RDF como uma relação entre recursos de sujeito e recursos de objetos. Esta especificação define o conceito de sub-propriedade. A propriedade `rdfs:subPropertyOf` pode ser usada para declarar que uma propriedade é sub-propriedade de outra. Se uma propriedade *P* é sub-propriedade da propriedade *P'*, quando todos os pares de recursos no qual são relacionados por *P* são também relacionados por *P'*.

O termo super-propriedade é frequentemente usado como o inverso de sub-propriedade. Se uma propriedade *P'* é uma super-propriedade da propriedade *P*, quando todos os pares de recursos no qual são relacionados por *P* são também relacionados por *P'*. Esta especificação não define uma propriedade superior que é a super-propriedade de todas as propriedades (CHAMPIN, 2006).

Conforme Daconta; Obrst e Smith (2003), a propriedade `subPropertyOf` é uma característica vai além das línguas comuns de OOP

como Java e C# que oferecem somente a herança de classe. Um exemplo deste seria declarar uma propriedade chamada o “fim de semana,” que pode ser uma `subPropertyOf` “semana.”. O atributo `rdfs:label` define um rótulo legível por humanos para a classe. Isto é importante para que as aplicações indiquem o nome da classe nas aplicações (DACONTA; OBRST; SMITH, 2003).

O elemento `rdfs:subclassOf` especifica que a classe é um especialização de uma classe existente. Este segue o mesmo modelo da herança biológica, onde uma classe filha pode herdar propriedades de uma classe pai. A idéia de especialização é que a subclasse adiciona algumas características únicas para o conceito geral. Então, abaixo da hierarquia da classe é referida a especialização, enquanto acima a hierarquia da classe é referida a generalização (DACONTA; OBRST; SMITH, 2003).

A propriedade `rdfs:domain` define formalmente a classe que uma propriedade pertence. O valor da propriedade deve ser uma classe previamente definida. A propriedade `rdfs:range` define um conjunto legal para uma propriedade. O valor deste atributo pode ser uma classe previamente definida. A propriedade `rdf:type` é uma propriedade padrão para definir que um sujeito RDF é um tipo definido em um esquema RDF. Por exemplo, uma pessoa de uma equipe com o ID 865 é um tipo de funcionário e pode ser escrito como (DACONTA; OBRST; SMITH, 2003):

```
<rdf:Description rdf:about= "http://www.mybiz.com/staff/ID/865">
  <rdf:type rdf:resource = "&example_chp5;Employee">
```

Outra propriedade útil é `rdfs:seeAlso`, que permite referenciar um recurso que possa fornecer a informação RDF adicional sobre o recurso atual. `rdfs:isDefinedBy` é uma propriedade para definir o *namespace* de um sujeito, na prática, o *namespace* pode apontar para um documento do RDF Schema. A `rdfs:comment` é uma propriedade útil para acrescentar uma informação descritiva adicional para explicar as classes e as propriedades a outros usuários do esquema. Como na programação, os comentários são essenciais para promover a compreensão e a adoção.

Pode-se representar diversos domínios com RDF, desde que exista uma descrição das entidades, relacionamentos e modelos de dados,

explicitando assim elementos fundamentais para o compartilhamento em ambiente distribuído de recursos na Web. Desta forma, evidencia-se o poder de extensibilidade desta linguagem, assim como também o fornecimento de interoperabilidade de conhecimento entre aplicações. No domínio educacional a rede Edutella fornece metadados em RDF distribuídos em uma rede P2P (Peer-to-Peer). Estudos referentes à utilização de XQuery para recuperação de metadados de objetos de aprendizagem compatíveis com o modelo RDF da Edutella têm sido proposto por Qu e Nejdl (2003). Aplicação Friend of a Friend (FOAF) define um vocabulário para Web descrever páginas pessoais de pessoas, grupos, companhias e outras coisas (FOAF, 2007). O padrão Dublin Core (2006) também possui um vocabulário mapeado para RDF.

A linguagem RDF como uma camada conceitual da Web Semântica é crucial para representação e integração de recursos. Como foi demonstrado, RDF possibilita formalizar estruturas conceituais atribuindo semântica a recursos existentes na Web e RDFS funciona como uma extensão desta funcionalidade, permitindo inclusive especificar ontologias. Mesmo com a possibilidade de se formalizar classes de recursos, restrições de classes e relacionamentos e detecção de violação de restrições, RDFS ainda apresenta alguns limites para uma representação mais rica de domínios.

RDF e RDF Schema permitem representação do conhecimento em uma ontologia. As primitivas principais para modelagem de RDF-RDFS são relacionadas com a organização de vocabulários em hierarquias de classes. Todavia, algumas características são necessárias no escopo das propriedades, classes disjuntas, combinações booleanas de classes, cardinalidade e restrições. Então, uma linguagem de ontologia deve ser mais rica que RDF Schema, assim são necessárias características adicionais. Uma linguagem de ontologia deve ser rica e expressiva para suportar eficientes raciocínios. (ALESSO; SMITH, 2006, p. 109).

Desta forma, apresenta-se a OWL (Web Ontology Language) como uma proposta de linguagem para especificação formal de ontologias, permitindo uma descrição mais rica de conceitos em um determinado domínio. OWL adiciona mais vocabulários para descrever classes,

propriedades, relacionamentos entre classes, cardinalidade, igualdade, e uma maior variedade de tipos de propriedades, características de propriedades e classes numeradas.

3.1.3 OWL (Web Ontology Language)

A OWL foi desenvolvida pela W3C através da extensão do vocabulário RDF, e é proveniente da linguagem de ontologias DAML (DARPA Agent Markup Language) e OIL (Ontology Interchange Language). De acordo com Alesso; Smith (2006), uma ontologia descreve formalmente uma lista de termos e suas relações a fim de representar uma área do conhecimento. Os termos representam conceitos importantes, tal como classes de objetos. Além de relações de subclasses, ontologias podem incluir informações, tal como propriedades, restrições de valores, declarações separadas e especificações de relações lógicas entre objetos. Assim, as ontologias podem ser usadas por ferramentas automatizadas para serviços avançados tais como buscas mais precisas, agentes de software inteligentes e gestão do conhecimento. (W3C, 2007).

OWL pode ser usado para descrever as classes e relações entre classes que são inerentes em documentos Web e aplicações. Pode-se formalizar um domínio definido por classes e propriedades daquelas classes, definir indivíduos e declarar propriedades sobre elas, e raciocinar sobre estas classes através de semântica e regras (ALESSO; SMITH, 2006). A W3C define três diferentes sub-linguagens da OWL: OWL lite, OWL DL e OWL Full.

A OWL Lite suporta uma hierarquia de classificação e restrições simples, é facilmente de ser compreendida e é menos expressiva que as demais sub-linguagens (ALESSO; CRAIG, 2006). Ela apresenta apenas algumas características que a OWL como um todo oferece e também impõe algumas limitações aos recursos oferecidos. Um exemplo é o uso de cardinalidade, que apesar de estar disponível é restrita a obter somente os valores 0 ou 1 (OWL, 2004).

A OWL DL é uma sub-linguagem da OWL Full que tem a forma de construtores da OWL e RDF. Suporta uma poderosa expressividade, permitindo computar automaticamente uma classificação hierárquica e

checar inconsistência na ontologia. OWL DL permite raciocínio automatizado utilizando Lógica Descritiva, um fragmento da Lógica de Primeira Ordem (ALESSO; SMITH, 2006). Esta linguagem oferece todos os recursos apresentados pela OWL Lite e também todos os outros mecanismos que a linguagem OWL disponibiliza. Entretanto, ainda impõe algumas restrições quanto ao uso destes recursos, como uma classe não pode ser instância e propriedade ao mesmo tempo.

A OWL Full usa todas as primitivas e permite a combinação com RDF e RDFS. Suporta máxima expressividade, pois além de oferecer todos os vocabulários disponíveis pela linguagem OWL, não impõe restrição ao uso dos recursos (OWL, 2006). Esta sub-linguagem não impõe restrição aos valores das classes, às propriedades, às instâncias e aos dados (OWL, 2006). Desta forma, existe a possibilidade de uma classe ser tratada ao mesmo tempo como instância de uma classe. A vantagem da OWL Full é que ela é totalmente compatível com a sintaxe e semântica RDF. Qualquer documento RDF válido é também um documento OWL Full válido. Uma conclusão RDF-RDFS válida é também uma conclusão OWL Full válida. A desvantagem da OWL Full é que não foi definido ainda, se provê suporte completo (ou eficiente) a raciocínio.

Todas as sub-linguagens da OWL usam RDF para declarar sua sintaxe e instâncias. A figura 5 apresentada por Alesso; Smith (2006) ilustra as relações de subclasses OWL e RDF-RDFS:

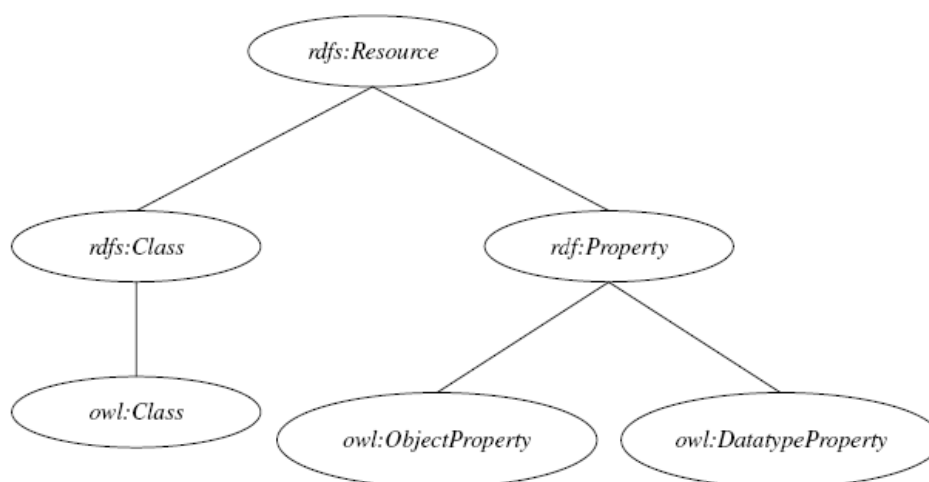


Figura 10: Relações de subclasses OWL e RDF-RDFS (ALESSO; SMITH, 2006)

Um documento OWL identifica os seguintes elementos de uma ontologia (ALESSO; SMITH, 2006):

- **Hierarquia de Classes:** Define relações de classes-subclasses;
- **Sinônimo:** Identifica classes equivalentes e propriedades equivalentes;
- **Associação de classes:** Mapeia uma ou mais classes de uma ou mais classes, através do uso de propriedades (i.e, domain/range);
- **Metadado de Propriedade:** Contém o metadado das propriedades;
- **Definição de classes:** Especifica a composição das classes.

A Web Ontology Language, define as classes e propriedades, assim como seu relacionamento, similar a RDF Schema, por exemplo, uma owl:Class categoriza elementos que são suas classes. O documento de referência da OWL provê uma especificação formal da OWL. Diferente de RDF, o vocabulário OWL é completamente amplo. Como RDF, OWL faz uso dos elementos da RDFS. Entretanto, OWL tem vários conceitos únicos, tal como combinações booleanas de expressões de classes e restrições de propriedades no qual acrescenta uma camada de raciocínio para aplicações (ALESSO; SMITH, 2006).

Com já foi explicitado, muitos dos elementos de uma ontologia em OWL são classes, propriedades, instâncias de classes, e relações entre estas instâncias. A sintaxe da OWL é construída sob RDF e RDFS, portanto é baseada na sintaxe XML. Entretanto, por causa da sintaxe RDF não ser muito legível, outras formas de sintaxe para OWL foram propostas: uma sintaxe XML básica RDF, uma sintaxe abstrata, e uma sintaxe gráfica baseada em UML (Universal Modeling Language). Um documento OWL contém uma ontologia OWL e é um documento RDF com elementos, *tags* e *namespaces*. Tal documento inicia com um cabeçalho que identifica um elemento raiz como um elemento rdf:RDF, no qual também especifica um

número de *namespaces* (ALESSO; SMITH, 2006) A tabela 2 é apresenta partes de um documento OWL.

Partes do Documento	Documento OWL
Cabeçalho-Declaração da	<?xml version="1.0" encoding="UTF-8"?>
Sintaxe XML	
Tag do elemento Raiz	<rdf:RDF
Namespaces	xmlns:iq = "http://www.Web-iq.com"> xmlns:owl = "http://www.w3.org/2002/07/owl#" xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs = "http://www.w3.org/1999/02/22-rdf-schema#" xmlns:dc = "http://purl.org/dc/elements/1.1/" xmlns:xsd = "http://www.w3.org/2000/1/XMLSchema#">
Classes e Propriedades OWL	<owl:Ontology rdf:about = "http://www.amazon.com"> <owl:versionInfo> \$ID: Overview.html </owl:versionInfo> <dc:creator> Douglas R. Hofstadter </dc:creator> <dc:title> G
Fim da OWL	</owl:Ontology>
Fim da RDF	</rdf:RDF>

Tabela 2: Partes de um documento OWL (ALESSO; SMITH, 2006)

De acordo com a tabela 2, depois do cabeçalho, o documento inclui a declaração do elemento RDF raiz e logo em seguida declara-se os *namespaces* referentes a OWL, RDF, RDFS, Dublin Core e XML Esquema. Conforme Alesso; Craig, (2006), as classes OWL definem entidades através de propriedades. Uma ontologia começa com a declaração do elemento owl:Ontology, no qual inclui elementos do padrão de metadados Dublin Core (DC) para prover o título, criador e outras informações, desde que a ontologia seja publicada como um recurso.

As classes OWL são interpretadas com um conjunto que contém indivíduos. É descrita usando uma descrição formal (matemática) que declara precisamente os requisitos para membros de uma classe e é definida pelo elemento owl:Classe e a subclasse é definida por rdfs:subClassOf (ALESSO; SMITH, 2006).

```
<owl:Class rdf:ID="Computer">
  <rdfs:subClassOf rdf:resource="#laptop"/>
</owl:Class>
```

Uma propriedade OWL é muito similar a propriedade RDFS, ambos usam `rdfs:domain` e `rdfs:range`, mas OWL acrescenta algumas restrições. Propriedades permitem declarar fatos gerais sobre os membros das classes e específicas fatos sobre indivíduos. Uma propriedade é uma relação binária. Existem dois tipos de propriedades OWL: propriedades de objetos que relaciona objeto a outros objetos e as propriedades de tipos de dados que relaciona objetos a valores de tipos de dados. Uma propriedade pode ser definida como uma especialização (sub-propriedade) de uma propriedade existente. OWL usa alguns dos vários tipos de dados de Esquemas XML, contém também restrições de propriedades, representada pelo elemento `owl:Restriction` que contém um elemento `owl:onProperty` e uma ou mais declarações de restrições. Uma restrição pode ser um tipo de valor (ALESSO; SMITH, 2006). OWL permite ainda realizar combinações booleanas, tal como união, intersecção, e complemento de classes que são elementos poderosos que podem ser colocados arbitrariamente.

Em OWL, as instâncias das classes são igualmente declaradas como RDF:

```
<rdf:Description rdf:ID="123456">
  <rdfs:type rdf:resource="computer"/>
</rdf:Description>
```

Diversas aplicações podem ser contempladas com OWL: portais Web, coleções multimídias, gerenciamento de Web sites corporativos, documentação de projetos, agentes e serviços e computação ubíqua (W3C, 2007).

Portais Web podem definir uma ontologia para a comunidade que permite mais organização semântica. A ontologia pode prover uma terminologia para descrever conteúdos e termos sobre outros termos. Estas definições permitem inferências que possibilitam usuários obterem resultados de buscas que são impossíveis nos sistemas de recuperação tradicionais (ALESSO; SMITH, 2006, p. 120).

Segundo a W3C (2007), um Portal Web é um Web Site que provê conteúdo informacional em um tópico comum, por exemplo, uma cidade específica ou um domínio de interesse. Um Portal Web permite indivíduos que têm interesses em um tópico, receber notícias, procurar e conversar com outros indivíduos. Construir uma comunidade e procurar *links* em outros recursos Web de interesses comuns (W3C, 2007).

A OWL também pode ser usada para prover anotações semânticas para coleções de imagens, áudio e vídeo. Ontologias multimídia podem ser de dois tipos: especificada por mídia ou especificada por conteúdo. Ontologias especificadas por mídia podem ter taxonomias de diferentes tipos de mídias e descrever propriedades de diferentes mídias. Por exemplo, vídeo pode incluir propriedades para identificar a duração de um clipe e cenas. Ontologias especificadas por conteúdo podem descrever o assunto do recurso, tal como as configurações ou participantes. Tal ontologia pode ser reusada por outros documentos no mesmo domínio, desde que seja para especificar a mídia (ALESSO; SMITH, 2006, p. 120).

Grandes corporações possuem numerosas páginas Web relacionadas a coisas como impressos, oferta de produtos, estudos de casos, procedimentos corporativos, produtos internos, *white papers*, e descrições de processos. Ontologias podem ser usadas para indexar estes documentos e prover melhor significado na recuperação. Embora, muitas grandes corporações possuem uma taxonomia para organizar suas informações, esta é sempre insuficiente. Uma simples ontologia é muitas vezes limitada por causa das categorias constituintes são granulares de um domínio. Assim a habilidade de trabalhar simultaneamente com múltiplas ontologias incrementará uma descrição mais rica de um domínio. Uma proposta de utilização de ontologias especificadas na linguagem OWL para representar padrões de projetos em desenvolvimento de aplicações Web tem sido apresentada em Kamthan e Pai (2006). Os autores apresentam uma proposta de utilização da OWL para representar o conhecimento de experiências futuras de especialistas para solucionar problemas em um domínio.

3. 2 Engenharia de Ontologias

Esta seção pretende apresentar uma revisão dos principais fundamentos referentes à construção de ontologias e suas implicações na Web Semântica. Tais fundamentos referem-se às linguagens, ferramentas e metodologias que sustentam a aprendizagem, a captura, a especificação e a formalização do conhecimento.

Pesquisas no campo da Engenharia do Conhecimento têm sido aplicadas em diversos contextos computacionais para capturar, representar e codificar o conhecimento fundamental em domínios particulares. Esta disciplina permite desenvolver Sistemas Baseados em Conhecimento que incorporam conceitos advindos de especialistas de domínios, propondo um conjunto de técnicas para representação do conhecimento. Neste prisma, destacam-se as ontologias como formalismos para construção de vocabulários semânticos para representação de conceitos intrínsecos em algum domínio.

Define-se ontologia como uma especificação formal de uma conceituação explícita e compartilhada. Esta conceituação refere-se a um modelo abstrato de alguns fenômenos no mundo identificados por conceitos relevantes. Explícito significa que os tipos de conceitos usados, e as restrições no seu uso são explicitamente definidos. Formal refere-se ao fato das ontologias serem legíveis por máquinas e compartilhada reflete a noção que as ontologias capturam o conhecimento consensual, isto é, não é privada a alguns indivíduos, mas aceita por um grupo (GRUBER; STUDER apud PÉREZ-GÓMEZ, 1999).

Desde o final dos anos noventa, as ontologias transformaram-se em tópico popular de pesquisa. Estas vêm sendo estudadas por diversas comunidades de pesquisa em Inteligência Artificial, incluindo a Engenharia de Conhecimento, Processamento da linguagem natural e Representação do conhecimento. Recentemente o uso de ontologias tornou-se difundido nos campos tais como: integração inteligente da informação, sistemas de informação cooperativos, recuperação de informação, comércio eletrônico, e gestão do conhecimento. As ontologias estão tornando-se popular devido a sua promessa de compartilhamento e entendimento comum que pode ser

comunicado entre pessoas e aplicações. O uso de ontologias e ferramentas de apoio oferece uma oportunidade de melhorar significativamente as potencialidades de gestão do conhecimento nas grandes organizações (DAVIES; FENSEL; HARMELEN, 2003).

A representação de ontologias é realizada através de linguagens simbólicas que permitem criar esquemas conceituais. Conforme Ding et al. (2005), as linguagens apropriadas para ontologia são necessárias para realizar ontologias explícitas com base em três aspectos importantes: conceituação, vocabulário e axiomas. A conceituação é necessária para criar um modelo apropriado de referência para representar o conhecimento factual, tal como definir as entidades e as relações em um domínio, e afirmar relações entre o vocabulário de entidades. Além da semântica, a linguagem deve também cobrir a sintaxe tal como a atribuição do símbolo (isto é, atribuindo símbolos aos conceitos) e as gramáticas (isto é, colocando em série de conceitos na representação explícita). Axiomas capturam a semântica para a inferência, regras e restrições necessárias ao conhecimento factual. Por exemplo, pode-se usar regras para gerar novos fatos de conhecimento existente, e validar a consistência do conhecimento.

A construção de ontologias compreende o estudo dos domínios de conhecimento, captura, organização e técnicas de estruturação desse conhecimento, portanto um projeto apoiado por ontologias é organizado em etapas, procedimentos e atividades definidas visando atingir os resultados esperados. Devedzić (2002) aponta para a similaridade entre a engenharia de software e engenharia de ontologias, colocando que as qualidades desejáveis para as ontologias, tais como: a decomposição, extensibilidade, manutenibilidade, modularidade, definição de interfaces e universalidade, são também desejáveis para componentes de software interoperável, ou classes uniformes de objetos em projeto orientados a objetos. O campo de estudo da Engenharia de Ontologias cobre conceitos de tópicos básicos como a representação do conhecimento e fundamentos teóricos; projeto; aplicação; desenvolvimento; compartilhamento e reuso do conhecimento.

Algumas metodologias para Engenharia de Ontologias têm sido propostas na literatura, norteando a prática de construção e validação de ontologias, assim como um guia para processos de análise e projeto de

sistema. Uma revisão realizada por Corcho; Fernández-Lopes; Gómez-Pérez (2002), apresentam um estudo comparativo das principais metodologias, ferramentas e linguagens disponíveis pra construção de ontologias. Uma metodologia apropriada para a Gestão do Conhecimento baseado na metodologia CommonKADS é proposta em Sure; Studer (2003). Os autores apresentam em Ontologias processos dirigidos a ontologias e expõe ainda trabalhos relacionados como as metodologias Skeletal, KACTUS e METHONTOLOGY.

Um processo de Engenharia de Ontologias Distribuídas que faz uso de redes Peer-to-Peer system (P2P), chamada DILIGENT é proposto por Pinto (2006) et al. Esta metodologia tem como eixo fundamental especificar um *framework* para engenharia de ontologias colaborativas e descentralizadas para uma organização virtual. Novas abordagens como o Gerenciamento de Conteúdo Distribuído (Distributed Knowledge Management) por Bonifácio (2006) baseia-se no princípio de que a multiplicidade e heterogeneidade de perspectivas nas organizações complexas não podem ser visualizadas como um obstáculo para a exploração do conhecimento. Assim, o autor apresenta uma arquitetura Peer-to-Peer, chamada KEx que permite representar coleções de nós de conhecimento para buscar e prover documentos com base na Web Semântica. Outras pesquisas que convêm serem destacadas são os sistemas para recuperação e indexação de conteúdos multimídias descritas na obra editada por Stamou e Kolliasa (2005).

Davies, Fensel e Harmelen (2003), descrevem uma arquitetura baseada na Web Semântica de gerencia do conhecimento e uma suíte de ferramentas inovadoras para processamento semântico da informação. Estas ferramentas são dirigidas à aquisição de ontologias e integração de grandes quantidades de dados, tecnologia de processamento de linguagem natural, desenvolvimento um repositório de esquemas de metadados e consultas e navegação semanticamente enriquecidas através de motores de busca de conhecimento. Esta tecnologia foi usada para facilitar o compartilhamento e reuso do conhecimento em uma organização virtual. Portanto, verifica-se a relevância das ontologias empregadas na criação de aplicações de gestão do conhecimento para a Web.

Estudos recentes sobre análise e projeto de aplicações Web têm sido destacados por Suh (2005) e Kappel et al (2003) com diversos relatos de abordagens e experimentos desenvolvidos com os fundamentos sistematizados da Engenharia de Web. A Engenharia de Web apoiada por ontologias é mais uma extensão do novo paradigma de desenvolvimento de sistemas inteligentes para Web. Behrendt; Arora (2003) afirmam que o desenvolvimento de aplicações com componentes semânticos é bem diferente de uma aplicação Web tradicional, pois integra agentes, representação do conhecimento, modelagem do conhecimento, anotação semântica, armazenagem e consulta de conhecimento. A expansão de uma arquitetura convencional para desenvolvimento de um método para aplicações da Web Semântica deve seguir os seguintes passos:

- Adoção do sistema para agentes de softwares;
- Seleção de uma linguagem de representação do conhecimento com suporte a ferramenta, incluindo adaptação de ferramentas para ambientes concretos;
- Criação de uma aplicação específica de ontologia (caso necessário);
- Anotação manual e/ou automática: manual do espaço informacional que cobrirá o desenvolvimento da aplicação Web e os esquemas conceituais para classificação. Automática é o raciocínio realizado por agentes inteligentes que classificam e fornecem conteúdo.

Projetos desta natureza como o SEAL Maedche et al (2003) apresentam a abordagem para construção de portais de gerenciamento do conhecimento baseado na Web Semântica. Ontoportal Woukeu et al (2003) apresenta um portal educacional baseado na Web Semântica para criar metadados e interligar com vários recursos educacionais (tutoriais, avaliações, cursos, etc.) para aprendizagem e ensino.

Desta forma, a Engenharia de Ontologias vem sendo amplamente aplicada como metodologia de desenvolvimento de um Repositório de Objetos de Aprendizagem para Web Semântica e também Sistemas de

Gerenciamento de Aprendizagem. Estas aplicações educacionais podem fornecer um sistema de Gerenciamento de Conteúdo, com recursos de interoperabilidade entre sistemas de ensino e repositório de teses, dissertações, objetos de aprendizagem e bibliotecas digitais. Atualmente a nova geração de aplicações Web deve fornecer recursos como: *feeds* RSS para compartilhamento de notícias, conferências, por exemplo; uma Interface aberta, através de uma API (Application Program Interface) que pode ser utilizada por outras aplicações; customização de *layouts* segundo os padrões Web; modularidade; um motor de inferência e um esquema de navegação destinado a consultas semânticas de toda a base de conhecimento existente no sistema.

3.3 Ferramentas

O desenvolvimento de aplicações baseadas na Web Semântica, em especial a modelagem do conhecimento é uma tarefa demasiadamente complexa, devido a grande dificuldade de se especificar ontologias. Para tal problemática, existem diversas ferramentas que automatizam o processo de desenvolvimento. Estas ferramentas vão desde editores de XML e linguagens derivadas a editores e *frameworks* para processamento de ontologias. Além de automatizar o processo de desenvolvimento, permite também documentar domínios através da apresentação de diagramas e grafos.

Para criação de ontologias existem diversas ferramentas, basicamente são implementadas em Java e a maioria são *open-source*. Os editores de ontologias permitem formalizar os componentes e gerar representações em RDF e OWLOs principais editores são: Protégé (2006), OntoEdit (2006), Ontobroker (2006). WebODE (2006). Existe também o *framework* Jena (2006) que permite manipular documentos RDF e OWL.

4 CLASSIFICAÇÃO E ORGANIZAÇÃO DO CONHECIMENTO

Nesta sessão serão apresentados alguns aspectos teóricos que permitem uma visão epistemológica da classificação e organização do conhecimento. Algumas classificações clássicas propostas por filósofos são apresentadas, evidenciando a complexidade de organização do conhecimento. Serão abordadas também as teorias contemporâneas para classificação do conhecimento e suas implicações no cenário da Web Semântica, assim como os principais padrões e recursos para representação de conceitos. Esta fundamentação é necessária para a compreensão da especificação conceitual do Modelo de Representação de Conteúdos que propõe um Sistema de Organização do Conhecimento - neste caso um Esquema de Classificação – para organização dos LOs.

4.1 Fundamentos da Classificação e Organização do Conhecimento

Em um processo gradativo, o homem atingiu o espírito científico, criando desta forma uma maturidade racional na interpretação e explicação dos fatos. Com a multiplicidade de fenômenos e fatos existentes no universo, a necessidade de explicá-los e investigá-los com diferentes procedimentos metodológicos e tecnologia empregada, houve a necessidade de delimitar o objeto de estudo, surge então a classificação das ciências. A história da ciência prova que os filósofos foram os primeiros a classificarem o conhecimento. Para Vickery (1980, p. 89), os filósofos gregos foram os primeiros a elaborar a classificação do conhecimento. A primeira classificação sistemática do conhecimento foi feita por Aristóteles. Ciências Teóricas, Ciências práticas e Ciências Poéticas. Estas por sua vez têm definições específicas conforme seu campo de atuação. As ciências teóricas buscam constatar a verdade, são a Matemática, a Física e a Teologia. As Ciências Práticas determinam as regras que dirigem os nossos atos perante a sociedade, são a Moral ou a Ética, a Economia e a Política. As Ciências Poéticas indicam os meios a usar obras exteriores, são a Retórica, a Poética e a Dialética.

Na obra *A República*, Platão, descreveu as ciências com instrumento da cultura, dividindo o conhecimento em Física, Ética e Lógica. Porfírio, filósofo grego do século IV, apresentou uma classificação dicotômica, a fim de explicar o processo de divisão de classes, denominada “Árvore de Porfírio”, que também é conhecida como “Árvore de Remeé”, por consequência de sua divulgação feita pelo filósofo Pierre de la Remeé, no século XVI. Matianus Capella escreveu a obra literária: “*De Nuptiis Philologiae et Mercurii*”, dividindo o conhecimento humano em Gramática, Dialética, Retórica, Geometria, Astronomia e Música.

Cassiodoro, filósofo e educador seguiu a classificação de Capella, agrupando as ciências em dois grandes grupos: *Trivium* e *Quadrivium*, chamada as setes artes literárias. No grupo Trivium (Artes ou Ciências Sermoniais) estão a Gramática, Dialética e Retórica. O grupo Quadrivium (Ciências Reais) é composto pelas áreas Geometria, Aritmética, Astronomia e Música. Este sistema também influenciou a classificação de Konrad Gesner, botânico e bibliófilo, sendo um dos grandes contribuintes para história da classificação.

Francis Bacon, filósofo empirista teve também uma importante contribuição para os sistemas de classificação. Publicou *Partitiones Scientiarum*, baseada também no trivium e quadrivium de Cassiodoro. Bacon dividiu as ciências de acordo com as faculdades intelectuais: a Memória, a Imaginação e a Razão, dando origem respectivamente a História, Poesia e Filosofia. Este sistema influenciou na classificação dos livros da Library of Congress (LC), que dispõe os assuntos de acordo com suas próprias necessidades.

Várias outras classificações bibliográficas sofreram influência de Bacon, como a de Harris apud Piedade (1977) que inverteu o esquema e adotou três grupos gerais, que serviram de base posteriormente para o surgimento do Sistema Decimal de Melvil Dewey. No século XIX, a classificação das ciências estavam sendo recuperadas pelo francês Descartes, os ingleses Hobbes, Locke e o alemão Leibniz. (PIEADADE, 1977). Com o cunho mais positivista mais século XIX, as classificações, fundamentaram-se na natureza dos fenômenos. Surgem as obras de

Bentham, Coleridge, Hegel, Ampère, Comte, Stuart Mill, Spencer, Wundt (PIEADA, 1977).

Augusto Comte estabeleceu a classificação das ciências quanto ao grau de complexidade, sob uma ordem crescente. A matemática é menos geral do que a Astronomia, embora a Astronomia requer aplicação matemáticas, mas a Matemática não necessita de outra ciência. Os fenômenos físicos, estudados pela Física, são mais complexos do que os astronômicos. O autor distinguiu seis ciências fundamentais classificadas na seguinte ordem: matemática, astronomia, física, química, biologia, sociologia e a moral. Sua classificação é do mais simples ao mais complexo, do mais abstrato ao mais concreto. Esta classificação corresponde à ordem histórica do surgimento das ciências positivas, as ciências de base mais complexas, assim como as ciências de base mais concretas, dependem intrinsecamente das mais abstratas.

Existe também a classificação de Rudolf Carnap, que divide as ciências em formais e factuais. As ciências formais (Lógica e Matemática) contêm apenas enunciados analíticos, cujo enunciado depende unicamente do significado de seus termos ou de sua estrutura lógica. As factuais (física, biologia, sociologia, psicologia e história), são ciências de conteúdo empírico, contêm enunciados analíticos e sintéticos, cuja verdade não depende só do significado de seus termos, mas igualmente dos fatos a que se referem (LAKATOS e MARIONI, 1991).

Outra classificação que merece ser destacada é a de Mário Bunge, que parte dos mesmos pressupostos de divisão do conhecimento de Rudolf Carnap, todavia as ciências factuais são divididas em Ciências Naturais e Culturais. As Naturais são: Física, Química, Biologia, Psicologia Individual. As Culturais são: Psicologia social, Sociologia, Economia, Ciência Política, História Material e História das Idéias. Já a classificação proposta por Wundt divide a ciência em Formais e reais. Sendo que as Ciências Formais são representadas pela matemática e a lógica. As ciências Reais são subdivididas em Ciências da Natureza e Ciências do Espírito. Para o autor as Ciências da Natureza são ainda divididas por Fenomenológicas, Genéticas e Sistemáticas. A química, a Física e a Fisiologia são exemplos de Ciências Fenomenológicas. Cosmologia, Geologia, Embriologia e Filogênese são

pertence a classe das Ciências Genéticas. Mineralogia, Zoologia e Botânica são exemplos de Ciências Sistemáticas propostas por Wundth.

Estas classificações estão voltadas à categorização do conhecimento científico como um todo. São oriundas da necessidade de compreensão dos fenômenos, fatos e comportamentos humanos pelos filósofos e cientistas. As mesmas não foram criadas para organização dos registros do conhecimento (livros, documentos iconográficos, revistas etc.), entretanto, estas classificações filosóficas é a base para a compreensão dos esquemas que mais tarde serviriam para organização de toda a produção intelectual. Assim, muitos enciclopedistas, bibliotecários e biólogos passaram a criar seus próprios mecanismos para classificação do conhecimento e organizar a informação para a recuperação a posteriori. Atualmente a classificação, representação e organização do conhecimento é objeto de estudo da Ciência da Informação e Ciência da Computação e áreas como a Terminologia. Esquema de Classificação da ACM (Association for Computing Machinery) e IEEE (Institute of Electrical and Electronics Engineers) foram criados para organização de artigos e documentos técnicos em suas bibliotecas digitais. Tesouros do Museu Getty e o vocabulário AGROVOC Thesaurus (2007) da Agricultural Information Management Standards e muitos outros são exemplos de esquemas de conceitos para organização do conhecimento na Web baseado em princípios da análise faceta.

A classificação como uma linguagem documentária permite a criação de instrumentos para leitura, condensação e representação de conteúdos existentes em documentos. Campos (2004) discute a problemática representacional, comparando os mecanismos de abstração presentes nas teorias da ciência da informação e da computação, que permitem a representação de conhecimento em domínios diversos, em uma proposta de encontrar um núcleo comum de conceitos imprescindíveis ao ato de modelar o conhecimento. Conforme a autora a ciência da informação possui fundamentos ligadas à representação de sistemas de conceitos, como a teoria da classificação facetada Ranganathan e a teoria do conceito de Dahlberg, que possibilitam a representação de domínios de conhecimento. Na ciência da computação, investigam-se os modelos

representacionais associados à modelagem orientada a objetos e a inteligência artificial, através da representação formal dos domínios de conhecimento, neste caso as ontologias apresentadas por Guarino, Gruber e Sowa.

Na ciência da computação, modelos de objetos e de dados são exemplos de representação no nível epistemológico. Contudo, para representar conhecimento, tais representações são ainda pobres. Esses formalismos estabelecem apenas significados particulares de estruturação. É necessário, portanto, introduzir a noção de nível ontológico (Guarino, 1994). A ontologia formal, como o próprio nome indica, é um formalismo classificado no nível ontológico, pois sistematiza conhecimento pretendendo a formalização de definições axiomáticas.

Campos (2004) enfoca a representação do conhecimento sob o ponto de vista da ciência da informação, afirmando que:

Na ciência da informação, a teoria da classificação pode ser considerada em um nível de transição entre o nível epistemológico e ontológico: apesar de não pretender chegar à definição dos conceitos de um dado domínio, ela possui um formalismo que possibilita a representação do conhecimento. Já as teorias do conceito e da terminologia podem ser classificadas como de um nível ontológico propriamente, pois permitem a sistematização de conhecimentos e possuem diretrizes para a elaboração de definições.

Desta forma, pode-se colocar que as teorias da classificação e do conceito estão voltadas à organização do conhecimento no campo da ciência da informação, e já a modelagem orientada a objetos está voltada à engenharia de software e já a representação do conhecimento é voltada à inteligência artificial no campo da ciência da computação.

Na Organização do Conhecimento, a Teoria da Classificação Facetada é o referencial para o desenvolvimento de muitos sistemas de classificação modernos. Esta teoria proposta é representante de um modelo que se utiliza do método dedutivo para classificar o conhecimento dentro de um contexto. Desta forma, possui mecanismos de representação para trabalhar com metaníveis conceituais, como, por exemplo, as categorias. É a partir delas que os conceitos são ordenados para formar classes de conceitos. O princípio de análise de faceta (subdivisões de assuntos em suas partes componentes) e a síntese (recombinação destas partes para classificar um documento adequadamente) (CAMPOS, 2004). A Classificação

Facetada compreende princípios e técnicas para os sistemas de organização e recuperação da informação. Um sistema facetado reconhece aspectos em um único assunto, e tenta sintetizar estes aspectos de maneira que os descreva de forma mais adequada. A relação entre assuntos, feita através do símbolo de dois pontos, seria infinita, provando que o conhecimento pode ser multidimensional e que essas relações podem tomar rumos diferentes, dependendo da síntese entre vários conceitos múltiplos (VICKERY, 1980).

Desde o final da década de 1980, diversos pesquisadores começaram a estudar a possibilidade de se utilizarem as teorias da classificação na organização conceitual de sistemas de hipertexto, principalmente a classificação facetada. Li et. al. (2005) propõe a utilização de Classificação Decimal de Dewey tradicionalmente utilizada para classificar documentos em bibliotecas, na construção de uma sistemas de organização de documentos na Web. Ingwersen e Wormell (1992) também destacam a análise facetada, para o desenvolvimento dos sistemas de recuperação da informação. Os autores examinam essa abordagem do Conhecimento e sua relevância para o acesso intelectual nas bibliotecas. Concluem que os princípios da organização do conhecimento e a metodologia facetada, por ter uma abordagem estruturada, poderiam ser utilizadas como ponto de partida na modelagem de uma estrutura cognitiva e no trabalho de análise de uma determinada área do conhecimento, antes do processo de projeto de um sistema de recuperação da informação.

Além desses, a exposta teoria introduz o conceito de categorias para representar um dado domínio de conhecimento. Esta teoria pode ser definida como um movimento para discutir a geração do conhecimento. O processo de relacionar objetos e fatos é um processo classificatório dentro da teoria da classificação.

Embora todas estas disciplinas permitam modelar o conhecimento, cada qual possui sua função dentro da sua área. A representação do conhecimento e a orientação a objetos podem ser utilizadas para modelar um software capaz de processar o conhecimento, desta forma, pode-se desenvolver uma ferramenta semântica para gerenciamento de tesauros,

taxonomias e esquemas de classificação utilizando os pressupostos da representação do conhecimento e orientação a objetos.

Como observado, as taxonomias, esquemas de classificação e outros sistemas de organização do conhecimento são amplamente utilizados na Web, no entanto, muitas aplicações ainda possuem um modelo estático de navegação e indexação de documentos. O desenvolvimento da tecnologia Web e a introdução de abordagens colaborativas para gerenciamento e compartilhamento de conteúdo criaram novas aplicações e também perspectivas para a organização do conhecimento dinâmico na Web. As atuais aplicações da chamada Web 2.0 permitem uma nova forma de anotação de recursos como links, vídeos, fotos, sites de relacionamentos sociais etc.

Em contraste aos sistemas formais de classificação em bibliotecas no qual são criados por profissionais especializados, existe uma forma popular para usuários de informação e mídia digital categorizar recursos com palavras-chaves livremente escolhidas. Palavras-chave selecionadas frequentemente para o uso individual tornam-se compartilhadas por toda uma comunidade. Por exemplo, o Flickr.com ativamente pequenos e posts frequentemente mais usados. Este sistema de indexação é conhecido como “folksonomias” (Li et. al, 2005).

Segundo, Li et. al.(2005) tanto os sistemas de classificação formal quanto folksonomias informais tem seus pontos fracos. Os autores constatarem que as classificações formais têm problemas de escalabilidade por que a especialidade técnica para classificação torna impraticável classificar de forma exata a vasta quantia de conteúdo que é produzido e usado em um ambiente *on-line*. Além disso, pessoas de diferentes domínios desenvolvem seu próprio sistema de classificação especializado assim existem múltiplas formas de classificar os mesmos recursos. Por outro lado, folksonomias, não tem um controle de vocabulário; a *tag* significa uma intenção individual qualquer do usuário e este pode aplicar a mesma *tag* de formas diferentes. Conforme Mathes apud Li et. al. (2005), esta ambigüidade é composta por uma falta de controle de sinônimos em diferentes *tags* freqüentemente usadas para o mesmo conceito Todos esses pontos fracos criam grandes barreiras para interoperabilidade de sistemas, apresentando um desafio crescente: (LI, etl al.)

No tópico a ser seguido apresentar-se-á algumas considerações acerca de recurso, padrões e tecnologias para sistemas de organização do conhecimento e aplicabilidade destes na nova geração da Web.

4.1.1 Instrumentos para Organização do Conhecimento na Web

Os tesauros e sistemas de classificação sempre estiveram ligados à organização da informação bibliográfica e ao tratamento de documentos físicos, todavia, recentemente seu uso na Web representa esquemas de controle terminológico para criação de sistemas de navegação, estratégias de recuperação da informação e muitos outros recursos dinâmicos. As iniciativas de padronização Zthes e SKOS Core, são propostas para especificar sistemas e metadados para tesauros Web e outros sistemas de organização do conhecimento.

O SKOS-Core está na versão 1.0 e pode ser definido com um esquema em RDF para representar tesauros e tipos similares do sistema da organização do conhecimento (KOS). O SKOS-Core pode ser usado para organização do conhecimento existente e construção de conceitos simples para a Web Semântica (Miles; Rogers; Beckett, 2006). O Guia SKOS Core e o vocabulário de especificação SKOS Core são recomendações do Consórcio Web (W3C) para promover a Web Semântica.

O Zthes (ZTHES, 2006), é um modelo abstrato para representar e buscar conteúdos em tesauros com hierarquias semânticas de termos e é descrito pela norma ISO 2788. Este modelo pode ser implementado usando os protocolos Z39.50 e SRW (Search/Retrieve Web Service). Através de um DTD (Document Type Definition), declara-se a estrutura de dados necessária para montar os esquemas em um documento XML. A Norma ISO-2788 cobre especificações para construção de tesauros, definindo os tipos de termos como: Termos Preferenciais (TP), Termos Não Preferenciais (TNP) e Relações entre TP e TNP.

Glossários, Taxonomias, Tesauros e Esquemas de Classificação são destacados por Miles; Matthes; Wilson (2006) como exemplos de sistemas que podem ser amplamente expressados em grafos RDF utilizando o vocabulário SKOS Core, os autores demonstram ainda relações com

metadados Dublin Core, no intuito de combinar os dois padrões. O Museu de Arte e Arquitetura Getty tem os seus conceitos expressados com esquemas em SKOS em um tesaurus Web para organização do conhecimento (SÁNCHEZ-ALONSO; GARCIA-BARRIOCANAL, 2006). Tesouro AGROVOC também possui suporte a Skos-Core, mantido pela Agricultural Information Management Standards (2007). AGROVOC é um vocabulário multilíngue, estruturado e controlado projetado para cobrir terminologias de assuntos em agricultura, pesca, florestas, alimentação e domínios relacionados (2007).

SKOS define uma estrutura para terminologias em um esquema de conceitos (skos:ConceptScheme) que possui relações semânticas (skos:semanticRelation). Esta relação semântica possui um termo geral (skos:broader), termo específico (skos:narrower) e termo relacionado (skos:related). Todo conceito (skos:Concept) está subordinado a um conceito topo (skos:hasTopConcept). Um simples Esquema de Classificação utilizado para organização de Teses e Dissertações em Engenharia de Eletricidade de Eletricidade, por exemplo, poderia ser apresentado graficamente como na figura 11.

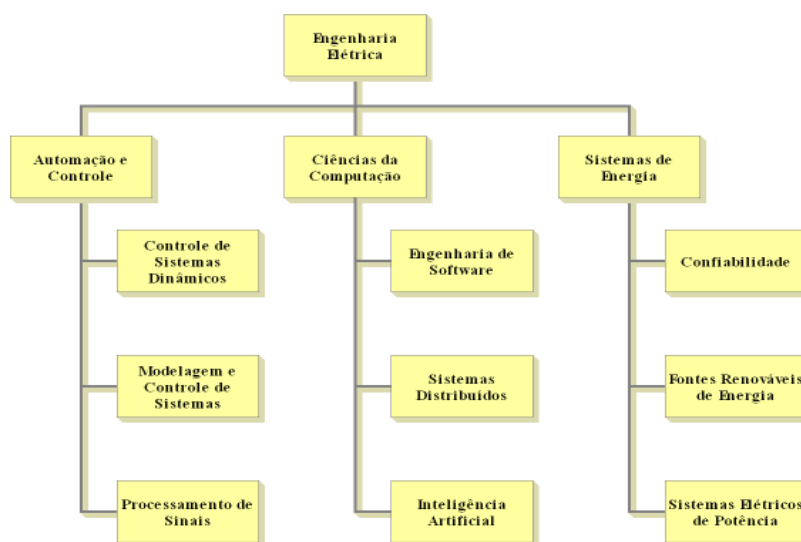


Figura 11: Esquema de Classificação em Engenharia de Eletricidade

De acordo com a figura 11 o termo Engenharia Elétrica é termo geral, Automação e Controle; Sistemas de Energia; Ciências da

Computação são termos específicos. Em Skos-Core este esquema poderia ser representado conforme a figura 12.

```

1 <rdf:RDF>
2   <skos:ConceptScheme rdf:about="http://netclass.ufma.br/Classification">
3     <dc:title>Esquema de Classificação</dc:title>
4     <dc:creator>Roosevelt Lins</dc:creator>
5     <dc:subject>Engenharia Elétrica; Computação; Sistemas de Potência, Automação e Controle</dc:subject>
6     <dc:description>Esquema de Classificação para organização do Conhecimento em Engenharia Elétrica</dc:description>
7     <dc:publisher>Roosevelt Lins</dc:publisher>
8     <dc:date>2006-10-20</dc:date>
9     <dc:language>pt-br</dc:language>
10    <skos:hasTopConcept rdf:resource="http://netclass.ufma.br/Classification/Engenharia"/>
11  </skos:ConceptScheme>
12  <skos:Concept rdf:about="http://netclass.ufma.br/Classification/Engenharia_Eletrica">
13    <skos:prefLabel>Engenharia Elétrica</skos:prefLabel>
14    <skos:inScheme rdf:resource="http://netclass.ufma.br/Classification"/>
15    <skos:narrower rdf:resource="http://netclass.ufma.br/Classification/Automacao_Controla"/>
16    <skos:narrower rdf:resource="http://netclass.ufma.br/Classification/Ciencias_Computacao"/>
17    <skos:narrower rdf:resource="http://www.profinfo.ufma.br/Sistemas_Energia"/>
18  </skos:Concept>
19 </rdf:RDF>

```

Figura 12: Exemplo em Skos-Core do Esquema de Classificação

Na linha 1 do documento RDF da figura 12 é definido o Esquema de Conceito, neste caso o esquema cobre conceitos da Engenharia Elétrica que possui um conceito topo que é Engenharia. Da linha 3 a 9 são declarados os elementos Dublin Core. Da linha 12 a 14 são declaradas as propriedades do conceito e da linha 15 a 18 são declaradas as relações entre o conceito. Um Conceito deve possuir um conceito principal, assim como um esquema de conceitos deve possuir também um conceito principal no Esquema de Classificação. O conceito principal pode ser o termo mais geral de um sistema de classificação, como por exemplo, o conceito Engenharia de Software pode ser um conceito principal em um Esquema de Conceitos em Ciência da Computação. Os conceitos Engenharia de Requisitos e Engenharia de Domínio podem ser conceitos subordinados à Engenharia de Software. O conceito Ciências da Computação e suas respectivas relações é exemplificado em um simples grafo representado na figura 13.

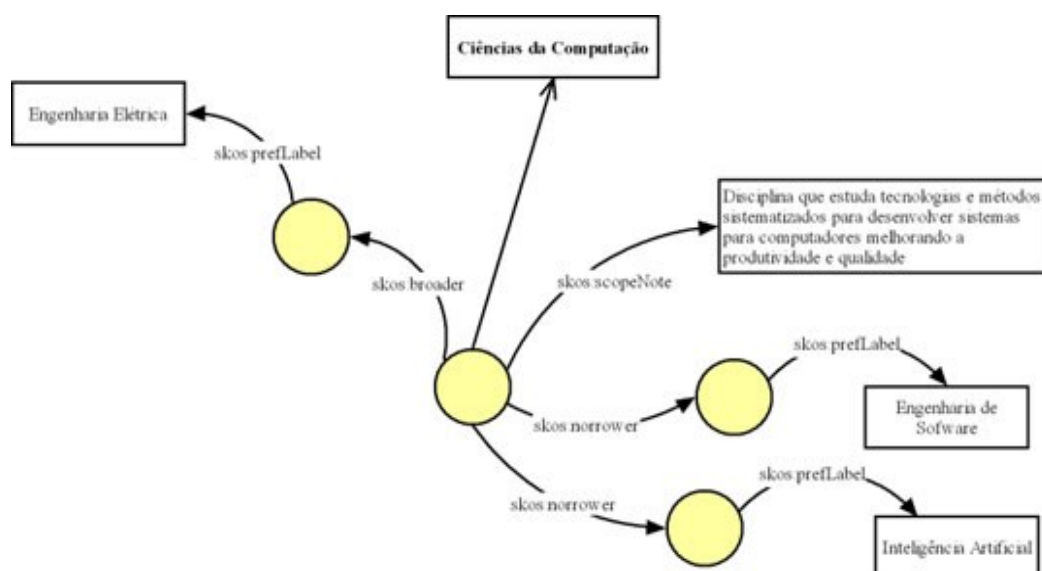


Figura 13: Grafo Skos-Core representando o conceito Ciências da Computação

Trabalhos como o TML – Thesaural Markup Language (Lee et al. 1999) define uma linguagem genérica de marcação em XML para tesouros que baseiam-se também na norma ISO-2788. A linguagem TML visa mapear os tesouros para ontologias, Conforme (Lee et. al., 1999), um tesouro TML apresenta-se separado em duas partes básicas: Ontologia que nos quais são definidos os tipos de termos e relações que estarão presentes no tesouro e suas Instâncias, que consistem nos termos e as relações entre eles.

Para construir esquemas de organização do conhecimento de forma automatizada e colaborativa baseada nas especificações Zthes e SKOS-Core, pode-se utilizar a ferramenta Tematres (2006), que consiste em um software aberto, sob licença GPL, escrito em PHP e base de dados em MySQL e com diversas funcionalidades úteis para gerenciamento de terminologias baseado em uma aplicação Web. Esta ferramenta permite o gerenciamento via termos via formulários e possui exportação dos metadados nos padrões Skos-Core e Zthes. Como ambos os padrões possuem sintaxe XML, qualquer aplicação pode acessar estes esquemas de metadados, tornando-os totalmente interoperável.

5 DESENVOLVIMENTO DO MODELO DE AGREGAÇÃO E REPRESENTAÇÃO DE CONTEÚDOS

Como foi destacada na seção 1.3, esta pesquisa faz uso da metodologia METHONTOLOGY para desenvolvimento do modelo ontológico. A METHONTOLOGY é um *framework* que permite à construção de ontologias no nível do conhecimento, inclui a identificação do processo de desenvolvimento da ontologia, um ciclo de vida baseado no desenvolvimento de protótipos e técnicas específicas para cada atividade. O processo de desenvolvimento de ontologia identifica as tarefas que serão executadas na construção da ontologia (GOMEZ-PEREZ, 1999). Esta metodologia auxilia o desenvolvimento através da especificação, conceituação, formalização, implementação e manutenção de ontologia (CORCHO et al, 2002).

A METHONTOLOGY, está dividida em 3 atividades fundamentais: atividades de gerenciamento, atividades orientadas ao desenvolvimento e atividades de suporte. Cada atividade e suas respectivas sub-atividades são demonstradas na figura 1 e explicadas em seguida.

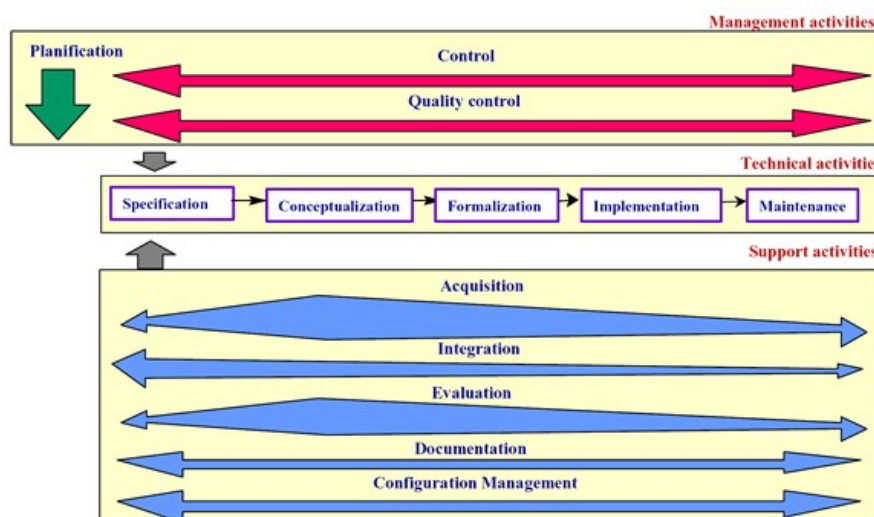


Figura 14: Atividades de desenvolvimento proposta pela METHONTOLOGY (CORCHO et al., 2002)

Atividades de gerenciamento:

1. **Planejamento:** identificação de tarefas a serem desempenhadas, como estas tarefas devem ser organizadas, quanto tempo e quais recursos elas devem consumir até serem completadas. Esta atividade é essencial quando se pretende fazer reuso de ontologias existentes;
2. **Controle:** atividade que garante que as tarefas planejadas na fase anterior sejam executadas completamente;
3. **Garantia de qualidade:** atividade que assegura que os produtos resultantes das atividades (ontologia, software, documentação) sejam satisfatórios.

Atividades de desenvolvimento:

1. **Especificação:** atividades que definem porque a ontologia será construída, que uso será feito dela e quem serão seus usuários finais;
2. **Conceituação:** atividade que organiza e converte a visão observada informalmente do domínio em uma especificação semi-informal usando um conjunto de representações intermediárias baseada em notação gráfica e tabular que pode ser entendida por especialistas de domínio e desenvolvedores de ontologia. O resultado da atividade de conceituação é uma ontologia de modelo conceitual.
3. **Formalização:** transforma o modelo conceitual da atividade de conceituação em um modelo formal ou semicomputável;
4. **Implementação:** constrói modelos computáveis numa linguagem computacional (Ontolígua, RDF Schema, OWL etc.). Ferramentas implementam automaticamente os modelos conceituais em variadas linguagens de ontologia;
5. **Manutenção:** atividades de atualização e correção da ontologia, se necessário.

Atividades de suporte

1. **Aquisição de conhecimento:** atividades de aquisição de conhecimento sobre um determinado domínio;
2. **Avaliação:** atividades de julgamento técnico das ontologias, dos ambientes de software associados e da documentação produzida, usando frames de referência;
3. **Integração:** atividades essenciais quando há reuso de ontologias existentes;
4. **Documentação:** atividades de detalhamento claro e exaustivo das fases de desenvolvimento.
5. **Gerenciamento de configuração:** atividade que permite realizar configurações para integração.

Para automatização do processo de aquisição, formalização e validação das ontologias utilizou-se a ferramenta de Protege 3.2 versão estável com o *plugin* OWL. O editor Protégé possui um *plugin* para exportar as ontologias em XMI (XML Metadata Interchange), que permite gerar automaticamente modelos UML (Unified Modeling Language), assim uma ferramenta para modelagem orientada a objetos pode importar o modelo ontológico e apresentar em diagramas com notação UML. Entretanto, até o presente momento, este recurso só pode ser utilizado se o desenvolvedor optar por construir sua ontologia utilizando *frames*, pois o *plugin* não oferece suporte para geração automática de ontologias em OWL para XMI. No início do desenvolvimento do modelo, optou-se pela abordagem em *frames*, mas durante a conceituação necessitou-se especificar propriedades mais complexas devido ao modelo SKOS, o que levou a utilizar a abordagem baseada em OWL. Desta forma, as classes da ontologia foram enriquecidas e representadas de acordo com a complexidade do domínio.

5.1 Especificação e Conceituação da Ontologia

A presente ontologia visa representar os conceitos acerca de um Repositório de Objetos de Aprendizagem. O eixo central do repositório é o Modelo de Agregação de Conteúdo, documentação do SCORM destinada à organização, empacotamento e compartilhamento de recursos de

aprendizagem. Outro padrão utilizado no Repositório para definição de um modelo de Representação de Conteúdos, foi o vocabulário SKOS-Core, destinado a criação de Sistemas de Organização do Conhecimento para Web Semântica. Neste caso, optou-se por um Sistema de Classificação como um sistema de organização do conhecimento para anotação dos conteúdos relativos aos objetos de aprendizagem. Desta forma, foi desenvolvido um modelo ontológico destinado ao gerenciamento de conteúdo de repositório de objetos de aprendizagem.

Como a idéia central é representar os padrões distintos de metadados, SKOS e LOM, convencionou-se que existem duas categorias de metadados: metadados estruturais e metadados conceituais. Os metadados estruturais descrevem os pacotes de conteúdo, funcionando como um recurso para catalogação de documentos. Metadado conceitual é a representação temática, ou seja, o conteúdo do objeto de aprendizagem, neste caso o Esquema de Classificação auxilia na marcação manual do recurso no repositório. O SKOS é um exemplo de metadado conceitual. Metadado estrutural consiste em um esquema de elementos para descrever a estrutura física de arquivos, dados e todo tipo de recurso na Web. O LOM, Dublin Core e IMS são exemplos de metadados estruturais.

Geralmente os Repositórios de Objetos de Aprendizagem são aplicações independentes que podem ser integrados aos Sistemas de Gerenciamento de Aprendizagem via uma API. O Repositório de Objetos de Aprendizagem funciona apenas para criar, organizar e disponibilizar conteúdos educacionais, cabendo ao LMS o gerenciamento de usuários, gerenciamento de avaliações, formação de grupos de aprendizes e outras atividades referentes à educação baseada na Web. O gerenciamento de conteúdo educacional (aulas, lições e avaliações) é feito pelo professor, ator responsável pela criação, anotação e disponibilização dos objetos de aprendizagem destinados aos aprendizes. No Repositório Semântico de Objetos de Aprendizagem, o modelo integra o LMS, usuários e agentes externos. O Repositório de Objetos de Aprendizagem tem a função de criar um objeto compartilhável de conteúdo (SCO), criar metadados para a descrição dos LOs, organizar e disponibilizar conteúdos educacionais. A arquitetura proposta neste modelo de repositório é ilustrada na figura 15.

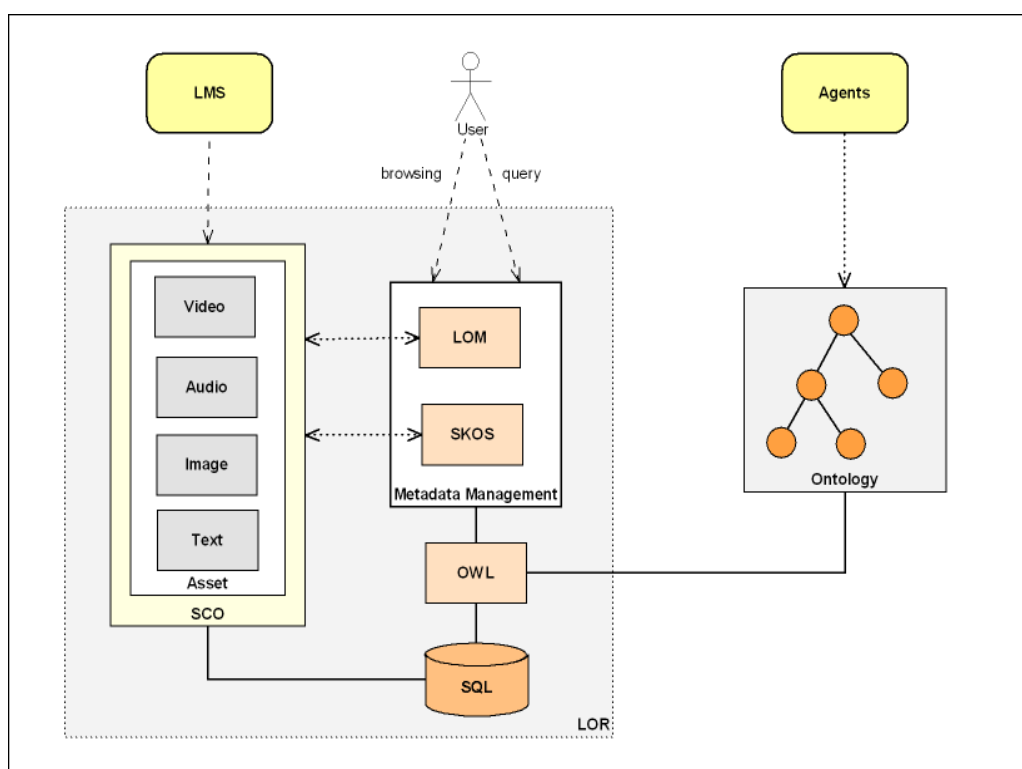


Figura 15: Arquitetura do Repositório de Objetos de Aprendizagem

De acordo com a figura 15, o LMS é um ator externo ao LOR e que acessa os SCOs através de uma API destinada a interpretação do LOM. Os dados referentes a um ou mais SCOs serão armazenados em uma base de dados relacional e também como instâncias em ontologias OWL. Com métodos implementados em ambiente Java utilizando a API Jena, pode-se realizar a transformação e apresentação em XML e XHTML às aplicações externas e aos usuários, respectivamente. A utilização de um Esquema de Classificação é necessária para o fornecimento de uma interface de navegação e consulta através de formulários aos usuários do sistema. A representação do modelo de conteúdos do repositório em OWL é fundamental para que agentes inteligentes existentes na Web possam recuperar e filtrar recursos educacionais de forma distribuída.

A atividade conceituação da METHONTOLOGY inclui tarefas para especificação dos componentes (conceitos, atributos, relações, constantes, axiomas, regras e instâncias), como demonstra a figura 16.

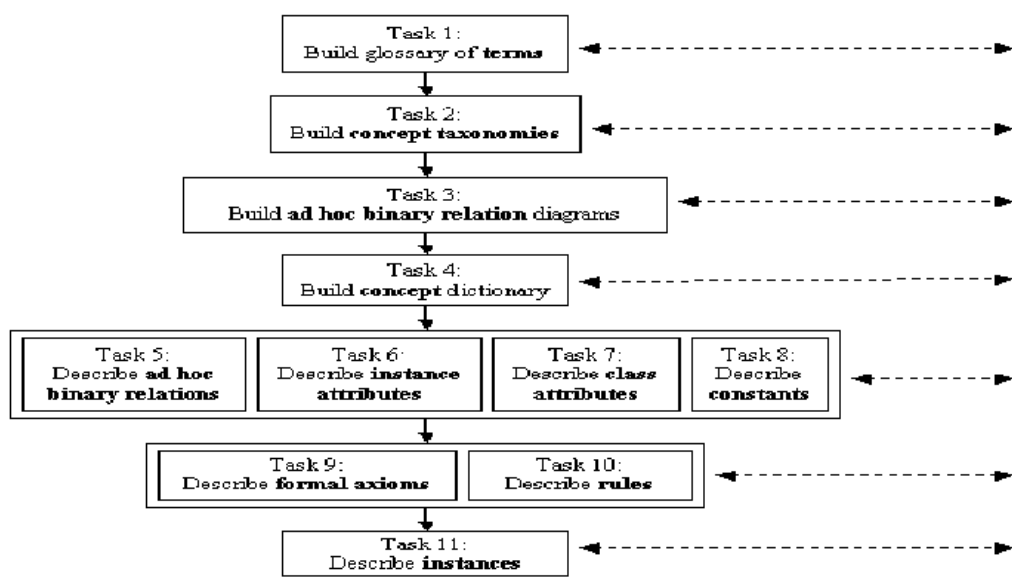


Figura 16: Tarefas da Atividade de Conceituação (CORCHO et al., 2002)

A primeira tarefa da atividade de conceituação refere-se à construção de um glossário de termos que inclui relevantes termos para o domínio (conceitos, instâncias, atributos, relações entre conceitos, etc.). O glossário é descrito em linguagem natural e apresenta ainda sinônimos e acrônimos. Estes termos foram extraídos do livro Modelo de Agregação de Conteúdos da especificação SCORM (ADL, 2006) e das propostas de ontologias para objetos de aprendizagem apresentadas por Ripamonti; Peraboni (2005), Quin; Hernandez (2004) e o vocabulário Skos-Core. Ripamonti; Peraboni (2005) apresentam um sistema para compartilhar objetos de aprendizagem no domínio da Ciência da Computação, através de uma abordagem ontológica para representação das especificações SCORM e LOM. Quin; Hernandez (2004) definem uma representação ontológica para o padrão IEEE LOM. Desta forma a análise destas ontologias permitiu o reuso dos termos, conceitos e algumas propriedades para o desenvolvimento do Repositório Semântico de Objetos de Aprendizagem. A tabela 3 apresenta uma parte do glossário utilizado para a representação do Repositório de Objetos de Aprendizagem. O glossário completo encontra-se na tabela no Apêndice A.

Nome	Sinônimo	Acrônimo	Descrição	Tipo
Objetos de Aprendizagem	Objetos Educacionais	LO	Recurso digital que pode ser reusado para dar suporte à aprendizagem	Conceito
É descrito por	-	-	Relacionamento entre os conceitos objeto de aprendizagem e metadados	Relação
Título	-	-	Nome dado ao objeto de aprendizagem. Atributo do conceito Metadado de Objeto de Aprendizagem	Atributo

Tabela 3: Glossário de termos

A segunda tarefa é caracterizada pela construção de uma taxonomia de conceitos. Os conceitos existentes no glossário de termos são os componentes fundamentais para a definição da taxonomia no domínio. Os conceitos essenciais no domínio são:

- Usuário (User)
- Repositório de Objetos de Aprendizagem (Learning Object Repository)
- Objeto de Aprendizagem (Learning Object)
- Componentes de Modelo de Conteúdo (Content Model Component)
- Atividade (Activity)
- Organização de Conteúdo (Content Organization)
- Recurso (Resource)
- Asset
- Conteúdo de Objeto Compartilhado (Sharable Content Object)
- Pacote de Conteúdo (Content Package)
- Pacote de Agregação de Conteúdo (Content Aggregation Package)
- Pacote de Recurso (Resource Package)

- Metadados de Objetos de Aprendizagem (Learning Object Metadata)
- Agregação de Conteúdo (Content Aggregation)
- Contribuidor (Contribute)
- Sequenciamento e Apresentação
- Sistema de Gerenciamento de Aprendizagem (Learning Management System)
- Esquema de Classificação (Classification Schema)
- Propriedade Coletável
- Coleção
- Conceito
- Esquema de Conceitos
- Coleção Ordenada

Usuário refere-se a todas as entidades humanas que acessam o sistema, ou seja, professores, aprendizes, coordenadores, monitores e administradores. Este conceito é relevante para demonstrar as interatividades dos usuários com o sistema, todavia, neste estudo não será apresentado o comportamento entre o sistema e os usuários, pois o foco é o mapeamento dos metadados de objetos de aprendizagem para ontologias.

Repositório de Objetos de Aprendizagem é um conceito que representa a aplicação com um todo, ou seja, um sistema baseado na Web para gerenciamento de conteúdo educacional que fornece recursos como importar, criar, exportar, armazenar e recuperar objetos de aprendizagem.

O conceito de Objetos de Aprendizagem é centrado na definição no princípio de reusabilidade definido por Wiley (2002). Para o autor, objeto digital é qualquer recurso digital que possa ser reusado para dar suporte à aprendizagem ou qualquer recurso, grande ou pequeno que possa ser entregue sob demanda através da Internet. Desta forma, um objeto de aprendizagem é caracterizado como uma unidade digital que pode ser criado e disponibilizado em ambientes de ensino mediado por computador. Uma animação, arquivos de textos, imagem, áudio pode ser um objetos de

aprendizagem, desde que esteja vinculado às estratégias educacionais em um determinado contexto de ensino.

O Modelo de Conteúdo define como os recursos de aprendizagem compartilhados são agregados em alto nível de unidades de instrução. Conforme a norma SCORM (ADL, 2006), os Componentes do Modelo de Conteúdo são formados por Organização de Conteúdos, Recursos e Atividades. Recursos podem ser Assets e Objetos de Conteúdo Compartilhável. Asset é o conteúdo de aprendizado na forma mais básica (como texto, imagem, vídeo, áudio, página Web, objeto de avaliação, etc). Já o Objetos de Conteúdo Compartilhável (SCO) é o conjunto de um ou mais Assets e representa o menor nível de granularidade de um recurso de aprendizagem. Sistema de Gerenciamento de Aprendizagem (LMS) realiza a comunicação com o Repositório de Objetos de Aprendizagem através dos SCOs. No intuito de prover reusabilidade, um SCO é independente do contexto de aprendizagem. Organização de Conteúdo é um mapa que representa o uso planejado do conteúdo através de unidades estruturadas de instrução (Atividades). O mapa mostra como uma Atividade é relacionada com uma outra. A figura 17 apresenta a Organização de Conteúdo.

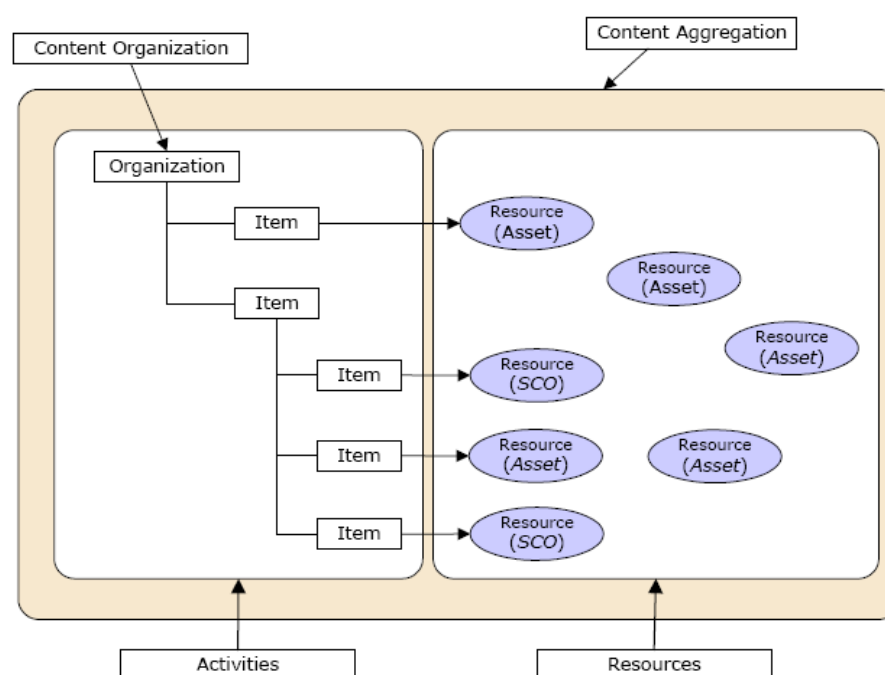


Figura 17: Organização de Conteúdo (ADL, 2006)

As Atividades representadas na Organização de Conteúdo podem ser consistidas por outras Atividades (sub-Atividades), no qual estas podem consistir por outras atividades. Não há nenhum limite do número de níveis aninhados nas atividades. Apesar das taxonomias de aprendizagem poderem ser associadas a níveis hierárquicos de atividades, (por exemplo, curso, capítulo, módulo, etc.), esta não é uma exigência. As atividades que dependem de outras atividades terão um recurso de aprendizagem associado (recurso de SCO ou recurso de Asset) que é usado para executar a atividade. Atividades que consistem de outras Atividades são também chamadas de Clusters no livro Sequenciamento e Navegação da especificação SCORM, que apresenta mais detalhadamente como o comportamento das Atividades e Clusters são definidas. Desta forma, Recursos, Organização de Conteúdos e Atividades são subclasses do conceito Componentes de Modelo de Conteúdo, enquanto Assets e SCOs são subclasses de Recursos.

A norma SCORM faz uso da especificação IMS (Content Packaging Specification) para definir uma estrutura padronizada para compartilhamento de conteúdo. O propósito do Pacote de Conteúdo é prover uma forma padronizada de troca de conteúdos de aprendizagem entre diferentes sistemas ou ferramentas. O Pacote de Conteúdo também provê um local para descrever a estrutura (ou organização) e o comportamento desejado da coleção de conteúdos de aprendizagem. Pacotes de conteúdo são suportados para serem usados para mover conteúdos de aprendizagem ou coleções de conteúdos de aprendizagem entre LMSs, ferramentas de desenvolvimento e repositório de conteúdos. A especificação IMS para Empacotamento de Conteúdo provê um formato comum de “input/output” que qualquer sistema pode suportar. Este formato possui uma estrutura e comportamento do conjunto (curso, módulo) de objetos de aprendizagem facilitando a interoperabilidade. O pacote de conteúdo define ainda um arquivo manifesto (imsmanifest.xml), que descreve o pacote e funciona como orientação para empacotar o arquivo de manifesto e os arquivos físicos, ou seja, empacotar os recursos (arquivos GIF, JPEG, PNG, HTML, Java Script, Flash, PDF, MP3 etc.).

Os componentes do Pacote de Conteúdos são: Pacote de Recurso (Resource Packages) e Pacote de Agregação de Conteúdo (Content Aggregation Packages). Pacote de Recurso é um mecanismo para empacotar Assets e SCOs que não prover alguma organização, contexto de aprendizagem ou taxonomia curricular. Empacotar recursos de aprendizagem é uma forma comum para intercâmbio. O Pacote de Recurso pode ser usado para mover os SCOS e Assets de sistema para sistema, desde que não haja nenhuma organização definida em um pacote de conteúdo, pois nenhuma estrutura lógica deve ser definida. Visto que nenhuma estrutura é definida, este tipo de pacote não pode ser entregue por um LMS ao aprendiz. Em outras palavras, o Pacote do Recurso é meramente uma coleção de recursos de aprendizagem que podem ser transferidos entre sistemas de aprendizagem, sem nenhuma organização lógica. O Pacote de Agregação de Conteúdo (Content Aggregation Package) é um processo de agregação de recursos (SCOs e Assets) em uma unidade de instrução (curso, módulo, capítulo etc.). É o propósito essencial para entrega de conteúdo ao usuário final (ADL, 2006).

Metadado de Objeto de Aprendizagem (Learning Object Metadata - LOM) é a tecnologia que serve para descrição interna dos objetos de aprendizagem no repositório. É padronizado pelo IEEE Learning Technology Standards Committee (LTSC) e sua função é ser um mecanismo para descrever e catalogar os objetos, facilitando a busca e seleção. Descrever também em uma nomenclatura comum, os recursos de aprendizado (Asset, SCOs, Content Aggregation). O padrão LOM define uma lista de diversos elementos para descrição de objetos de aprendizagem, entretanto neste trabalho serão apenas enfocados os nove elementos principais. Os elementos são conceitos que compõe as subclasses de “Metadado de Objetos de Aprendizagem”, sendo estes listados a seguir:

- Geral
- Ciclo de vida
- Meta Metadado
- Técnico
- Educacional

- Direitos
- Relação
- Anotação
- Classificação

A categoria Geral é usada para descrever informações gerais sobre os componentes do modelo de conteúdo. Ciclo de vida é usado para descrever atributos relacionados ao histórico ou o estado atual de um objeto de aprendizagem e quais os componentes que tem afetado o objeto de aprendizagem durante sua evolução. A categoria Meta Metadado pode ser usada para descrever informações sobre o próprio registro de metadados. A categoria Técnica pode ser usada para descrever os requisitos técnicos e as características do modelo de conteúdo. A categoria “Educação” é usada para descrever características pedagógicas dos objetos de aprendizagem. O elemento “Direitos” do padrão LOM é usado para descrever a propriedade intelectual e as condições de uso dos objetos de aprendizagem. A categoria “Relação” descreve o relacionamento entre um objeto de aprendizagem e outros objetos de aprendizagem qualquer. Para definir relacionamentos múltiplos, pode haver múltiplas instâncias desta categoria. Se houver mais de um objeto de aprendizagem alvo, então cada alvo terá uma instância nova de relacionamento. Anotação é uma categoria que pode ser usada para realizar comentários sobre o uso do objeto de aprendizagem e informações sobre quando e por quem o comentário foi criado. A categoria “Classificação” pode ser usada para descrever onde o objeto de aprendizagem é incluído com um sistema de classificação particular.

Classes como “Entidade”, “Palavras-chave”, “Requisito”, “Contato” e “Contribuidor” foram criadas por Qin; Hernandez (2004) para serem reutilizadas como elementos em algumas classes. Contribuidor é um elemento que pode ser usado para descrever quais entidades (pessoas e organizações) que tem contribuído para determinar o modelo de componente de conteúdo durante o ciclo de vida (criação, edição, revisão, publicação, etc.). O conceito “Contribuidor” permite capturar todos os indivíduos ou organizações envolvidas. O conceito “Palavras-chave” é usado para descrever um conteúdo de objeto de aprendizagem, formato ou

estrutura. Requisitos é a capacidade técnica necessária para usar o objeto de aprendizagem. Se houver requisitos múltiplos, todos serão requeridos.

O conceito “Sequenciamento e Apresentação” especifica estratégias de seqüências de um determinado curso. É baseado no padrão IMS que define um conjunto de seqüências de informação para atividades de aprendizagem. Já o conceito “Sistemas de Gerenciamento de Aprendizagem” refere-se a uma aplicação destinada a gerenciamento de ambientes de ensino baseado na Web. Esta aplicação acessa o Repositório de Objetos de Aprendizagem através dos SCOs.

O conceito “Esquema de Classificação” é um Sistema de Organização do Conhecimento para classificação de conteúdos dos objetos de aprendizagem (SILVA et. al.). Consiste em uma estrutura hierárquica de áreas do conhecimento e respectivos termos para representação dos objetos de aprendizagem. Este esquema faz uso da especificação W3C SKOS (Simple Knowledge Organisation Systems), que define um vocabulário padrão para a criação de Tesouros, esquemas de classificação, glossários, vocabulários controlados e outros sistemas de organização do conhecimento. Possui as seguintes subclasses:

- Propriedade Coletável
- Coleção
- Conceito
- Esquema de Conceitos
- Coleção Ordenada

Propriedade Coletável é uma classe que pode ser usada como Coleção. A classe “Coleção” representa uma coleção de conceitos por significado. Coleções nomeadas podem ser usadas com propriedades de relações semânticas coletáveis, por exemplo skos:narrower, quando desejamos que uma coleção de conceitos seja apresentada sob uma "etiqueta" em uma hierarquia. O Skos-Core especifica uma estrutura de relações semânticas: “broader”, “narrower”, “related” e relações hierárquicas: “broader/narrower”.

Conceito, na especificação SKOS, representa o termo, ou a área de conhecimento, elemento fundamental para classificar o conteúdo. Conforme a definição de Agüerra (2004), conceito é uma unidade de pensamento que pode ser definida ou descrita. Um conceito pode ter uma série de etiquetas associadas, onde cada etiqueta é uma palavra, frase ou símbolo que pode ser utilizado pra referenciar a esse conceito. Esquema de Conceitos é um conjunto de conceitos, opcionalmente incluindo declarações sobre relações semânticas entre esses conceitos.

Tesauros, esquemas de classificação, listas de cabeçalhos de assuntos, taxonomias, terminologias, glossários e outros tipos de vocabulário controlado são exemplos de esquemas de conceitos. Neste caso, o esquema de conceitos proposto é um Sistema de Classificação. A classe “Coleção Ordenada” é uma coleção ordenada de conceitos, onde o agrupamento e a ordenação são significativos. Coleções ordenadas podem ser usadas com propriedades de relações semânticas coletáveis, onde o usuário quer que um conjunto de conceitos seja apresentado em uma ordem específica, e opcionalmente sob uma "etiqueta" (W3C, 2007).

A partir destes pressupostos fundamentais para especificação conceitual do LOR, pode-se construir a taxonomia. Os conceitos foram organizados em uma estrutura hierárquica, formando assim as classes e sub-classes do domínio, fazendo uso do mecanismo de herança. A figura 18 ilustra a taxonomia de conceitos apresentada em uma hierarquia de classes em OWL.



Figura 18: Taxonomia de Conceitos

A terceira etapa da METHONTOLOGY consiste em construir os diagramas de relações binárias entre os conceitos explicitados na taxonomia. Uma vez que a taxonomia e as relações binárias têm sido geradas, o engenheiro do conhecimento, deve agora especificar quais as propriedades (atributos) e relações que descrevem cada conceito da taxonomia em um dicionário de conceitos. O dicionário de conceitos é a quarta tarefa da metodologia e deve conter todos os conceitos do domínio, suas relações, suas instâncias e atributos de instância. As relações específicas para cada conceito são aquelas cujo domínio é o conceito. Como exemplo, o conceito Objeto de Aprendizagem que possui duas relações: “descrito por” e “armazenado por”.

A figura 19 é apresentada a seguir, ilustrando graficamente os conceitos e suas relações de acordo com a atividade de construção de diagramas de relações binárias.

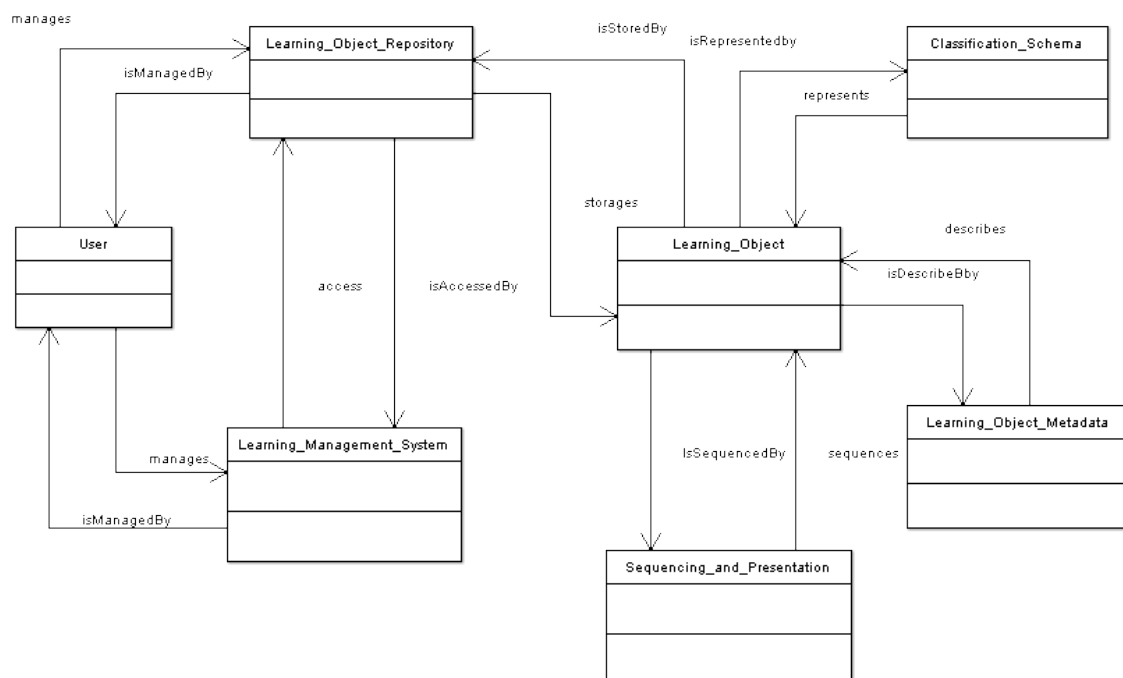


Figura 19: Diagramas de relações binárias

Conforme mostra a figura 19, Usuários, Repositório de Objetos de Aprendizagem (LOR), Sistema de Gerenciamento de Aprendizagem (LMS), Objetos de Aprendizagem (LO), Metadados dos Objetos de Aprendizagem (LOM), Esquema de Classificação, Sequenciamento e Apresentação, são as principais classes que formam a ontologia. A classe usuário possui o relacionamento “gerencia” o Repositório de Objetos de Aprendizagem e Sistema de Gerenciamento de Aprendizagem. O Sistema de Gerenciamento de Aprendizagem “acessa” o Repositório de Objetos de Aprendizagem, já que o LMS se comunica com este através de uma API para fornecer o pacote de conteúdo que contém os objetos de aprendizagem ao usuário. Ao mesmo tempo em que o LMS acessa os LOs, estes são acessados pelo LMS, desta forma o relacionamento é inverso, expresso como uma propriedade inversa: LOs “são acessados pelo” LMS. O LOR armazena todos os conteúdos e pacotes de conteúdos educacionais destinados à aprendizagem, ou seja, os Objetos de Aprendizagem (pacotes de conteúdo

com 1 ou mais Assets), desta forma o relacionamento é: LOR “armazena” LO. Os LOs são descritos pelo LOM. O Esquema de Classificação é destinado à representação temática (classificação) de conteúdos por área do conhecimento. A tabela 4 mostra uma parte do dicionário de conceitos.

Nome do Conceito	Instância	Atributos de Instância	Atributos de Classe	Relações
Usuários		Identificador do Usuário, nome do usuário.		Gerencia
Sistema de Gerenciamento de Aprendizagem				Acessa, é gerenciado por,
Repositório de Objetos de Aprendizagem				Acessado por, Armazena,
Objetos de Aprendizagem				Armazenado por, Descrito por, Representado por, seqüenciado por.
Componente do Modelo de Conteúdo		Título do CMC, Identificador do CMC.		
Organização de Conteúdo		Estrutura de organização de conteúdo		Tem atividade agregada
Pacote de Agregação de Conteúdo				Tem organização de conteúdo agregado
Pacote de Recurso				Tem recurso agregado
Recurso		Tipo de recurso, arquivo do recurso		
Metadado de Objetos de Aprendizagem	-	-	-	Descreve
Entidade		Email da Entidade, identificador da Entidade, Nome da Entidade, Web Site da Entidade.	Contato da Entidade	
Sequenciamento e Apresentação				Sequencia
Contribuidor		Data de Contribuição	Entidade do Contribuidor	
Contribuidor do Metadado		Papel do Contribuidor de Metadados: criador e validador		
Contribuidor do		Papel do Contribuidor		

Objeto de Aprendizagem		de Objeto de Aprendizagem: autor, publicador, anônimo, iniciador, finalizador, validador, editor, designer gráfico, implementador técnico, provedor de conteúdo, validados técnico, escritor de <i>script</i> , projetista instrucional, especialista do assunto.		
Palavra-chave		Identificador da palavra-chave, categoria da palavra- chave,	Palavra-chave	
Geral		Nível de Agregação, Catálogo, Cobertura, Descrição, Entrada, Identificador, Língua, Estrutura, Título		
Ciclo de Vida		Status, versão	Contribuidor	
Meta-Metadados	-	Meta Contribuidor, meta_entry, Meta- esquema, meta identificador, meta catálogo, meta linguagem.	Contribuidor do Metadado	-
Técnico		Outros requisitos de plataforma Duração Formato Localização Tamanho Notas de instalação	Requisitos Técnicos	
Educacional		Tipo de Interatividade, Contexto, Alcance típico de idade (typicalAgeRange), Tempo típico de aprendizagem, Nível de Interatividade, Dificuldade, Densidade Semântica, Tipo de, Recurso de Aprendizagem, Papel pretendido ao usuário final, Descrição		
Direitos		Direitos autorais e Restrições Custo, Descrição dos Direitos		
Relação		Tipo de Relação,	Relação do Recurso	
Anotação		Data da anotação Descrição da Anotação	Anotação da Entidade	
Classificação		Descrição da classe, Propósito da Classe	Palavra-chave da classe,	Representa

			caminho (path) da taxonomia da classe	
Contato		Cidade, Fax, Telefone, Rua, CEP, Estado		
Esquema de Classificação				Representa
Propriedade Coletável				
Coleção				membro
Conceito		Termo preferido		No esquema, relação semântica: Conceito específico Conceito geral, Conceito Relacionado
Esquema de Conceitos				Tem conceito principal,

Tabela 4: Dicionário de Conceitos

A quinta tarefa é destinada à definição das relações binárias detalhadamente incluídas no dicionário de conceitos. Devem-se especificar as cardinalidades, relações inversas. Após esta atividade, é necessário definir os detalhes os atributos de instâncias incluídos no dicionário de conceitos e os atributos das classes, assim como também definir as constantes.

Nome da Relação	Conceito	Cardinalidade	Conceito Alvo	Relação inversa
Gerencia	Usuário	N	Sistemas de Gerenciamento de Aprendizagem, Repositório de Objetos de Aprendizagem	Gerenciado por
Gerenciado por	Sistemas de Gerenciamento de Aprendizagem, Repositório de Objetos de Aprendizagem	N	Usuário	Gerencia
Acessado por	Repositório de Objetos de Aprendizagem	N	Sistemas de Gerenciamento de Aprendizagem	Acessa
Acessa	Sistemas de Gerenciamento de Aprendizagem	N	Repositório de Objetos de Aprendizagem	Acessado por

Descrito por	Objetos de Aprendizagem	N	Metadados de Objetos de Aprendizagem	Descreve
Descreve	Metadados de Objetos de Aprendizagem	N	Objetos de Aprendizagem	Descrito por
Representa do por	Objetos de Aprendizagem	N	Esquema de Classificação	Representa
Representa	Esquema de Classificação	N	Objetos de Aprendizagem	Representado por
Armazenado por	Objetos de Aprendizagem	N	Repositório de Objetos de Aprendizagem	Armazena
Armazena	Repositório de Objetos de Aprendizagem	N	Objetos de Aprendizagem	Armazenado por
Seqüenciado por	Objetos de Aprendizagem	N	Sequenciamento e Apresentação	Seqüência
Seqüência	Sequenciamento e Apresentação	N	Objetos de Aprendizagem	Seqüenciado por
Tem conceito principal	Esquema de Conceito	N	Conceito	-
Tem recurso agregado	Pacote de Recurso	N	Recurso	-
Tem atividade agregada	Organização de Conteúdo	N	Atividade	-
Tem Organização de Conteúdo Agregada	Pacote de Organização de Conteúdo	N	Organização de Conteúdo	-
No esquema	Conceito	N	Esquema de Conceito	-
Tem Asset	Objeto Compartilhado de Conteúdo		Asset	-

Tabela 5: Detalhamento das Relações Binárias

O próximo passo é descrever em detalhes dos atributos de instância que já foram incluídos no dicionário de conceitos. Atributos de instâncias são aqueles atributos que descrevem as instâncias do conceito e os valores que podem ser diferentes para cada instância do conceito. Para cada atributo de instância, o engenheiro do conhecimento deve especificar o nome do atributo, o conceito, o tipo de valor e o valor de alcance (no caso valores numéricos), mínima e máxima cardinalidade (CORCHO et. al., 2002). A tabela 6 mostra os Atributos de Instância definidos para o modelo.

Nome do Atributo	Nome do Conceito	Tipo de Valor	Valor de Alcance	Cardinalidade
Membro	Coleção	Entidade		(0,1)
Anotação da Entidade	Anotação	Entidade		(0,1)
Palavra-chave da Classe	Classificação	Esquema de Classificação		(0:40)
Caminho da Classe da Taxonomia	Classificação	Entidade		(0,15)
Entidade de Contribuição	Contribuição	Entidade		(0,40)
Contato de Entidade	Entidade	Contato		(0,1)
Palavra-chave do LO	Geral	Palavra-chave		(0,10)
Contribuição do LO	Ciclo de Vida	Contribuição		(0,30)
Contribuição do Metadado	Meta-Metadado	Contribuição		(0,10)
Requisito Técnico	Técnico	Requisito		(0,40)

Tabela 6: Atributos de Instância

Após a definição dos atributos de instância é necessário detalhar todos os atributos das classes. Os atributos são apresentados na tabela 7, destacando o conceito no qual está relacionado, tipo de valor e cardinalidade. A tabela completa com os atributos encontram-se no apêndice B.

Nome do Atributo	Nome do Conceito	Tipo de Valor	Cardinalidade	Valores
Termo Preferido	Conceito	String	(1,1)	
Identificador_CMC	Componente do Modelo de Conteúdo	String	(1,1)	
Título_CMC	Componente do Modelo de Conteúdo	String	(1,1)	
Estrutura de Organização de Conteúdo	Organização de Conteúdo	String	(1,1)	

Tabela 7: Atributos de Classes

A METHONTOLOGY contempla também a descrição de constantes definidas no glossário de termos, entretanto no modelo de domínio proposto

não foi identificado constantes. Após a definição dos conceitos, propriedades e atributos e o refinamento dos mesmos através da descrição detalhada, o Engenheiro do Conhecimento deve especificar os axiomas formais do domínio. Desta forma algumas restrições e expressões lógicas foram detectadas e aplicadas na ontologia. Os seguintes axiomas definidos são apresentados em linguagem natural:

- Todas as instâncias de “objetos de aprendizagem” devem ser representadas pelo “metadado de objeto de aprendizagem”;
- Um objeto compartilhado de conteúdo deve possuir no mínimo 1 asset;
- Todo objeto de aprendizagem deve ser classificado em pelo menos 1 conceito;
- Um objeto de Aprendizagem não pode ser classificado em mais de um conceito principal (esquema de classificação), ou seja, não pode ser classificado em duas áreas do conhecimento;
- Um LO pode ser apenas um pacote de conteúdo ou um componente de modelo de conteúdo;
- Um conceito pode ser um membro de mais de um esquema de conceitos.
- Todos os valores da propriedade “membro” são representados como uma união das classes “conceito” e “coleção”.
- A propriedade “Lista de Membros” tem exatamente um valor
- Todos os valores pertencentes à propriedade “no esquema” são um tipo de “esquema de conceito”.
- Todos os valores pertencentes à propriedade “relação semântica” são um tipo “conceito”.
- Todos os “conceitos principais” são “conceitos”.

Em OWL os axiomas são definidos através de propriedades das chamadas classes complexas. Assim, OWL possui construtores para criação

de restrições, quantificadores universais e existenciais que permitem definir axiomas, que geralmente são representados em lógica de descrição.

Através da aplicação da METHONTOLOGY para desenvolvimento da ontologia, pode-se constatar que esta metodologia sistematizou a identificação dos termos, organização de conceitos e definição. As etapas e tarefas desta metodologia permitem que especialistas de domínio e desenvolvedores possam ser guiados, realizando planejamento e gerenciamento em alto nível, evitando assim dificuldades para construção de modelos ontológicos. A METHONTOLOGY facilitou a organização de todos os termos extraídos do modelo de referência SCORM e das especificações LOM e SKOS, caso uma metodologia não fosse aplicada, as atividades ligadas à especificação dos conceitos e propriedades poderia ser uma tarefa mais trabalhosa e complexa.

6 FORMALIZAÇÃO E IMPLEMENTAÇÃO

Nesta seção será apresentada a formalização da ontologia em linguagem semi-computável, neste caso, a OWL. Através da ferramenta Protégé, foi possível formalizar e implementar de modo automatizado os componentes da ontologia do modelo de representação e agregação de conteúdos, seguindo as etapas da METHONTOLOGY. Após esta formalização e implementação, verificou-se a consistência da ontologia utilizando o raciocinador Racer Pro (2007) para validação do modelo. Vale ressaltar que na METHONTOLOGY, “implementar” significa definir construir o modelo ontológico com uso de uma linguagem semi-computável, neste caso utilizou-se a OWL. Através do uso da API Jena pode-se realizar a manipulação da ontologia através de construtores Java. Todavia, apenas será mostrado como Java permite criar, consultar e manipular ontologias em RDF e OWL.

6.1 Classes

De acordo com o Dicionário de Conceitos, produto da primeira tarefa da METHONTOLOGY, os conceitos foram organizados em uma taxonomia e adicionados no editor de ontologia respeitando a sua estrutura hierárquica. Após a formalização dos componentes capturados para o modelo de taxonomias no Protégé, verificou-se a consistência da ontologia. Esta tarefa é imprescindível para detectar potenciais ambigüidades e erros na especificação do domínio. Desta forma, utilizou-se o Racer-Pro para checar a consistência. A figura 20 demonstra as classes e subclasses representadas.

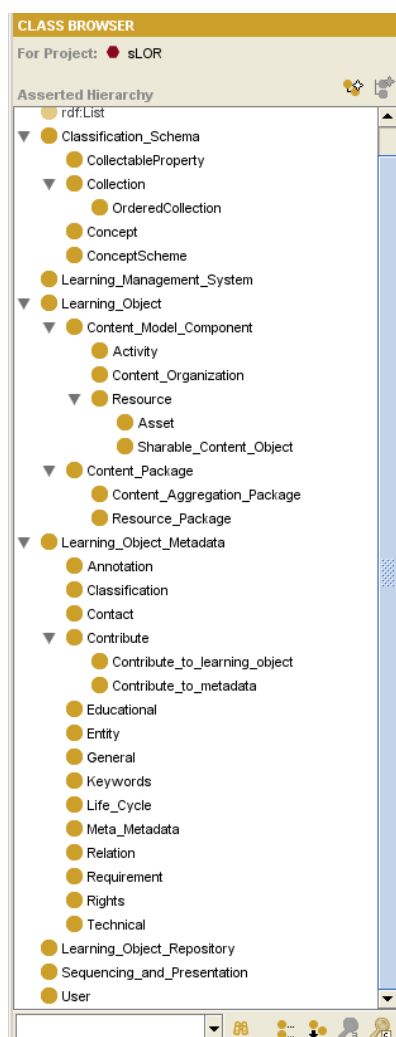


Figura 20: Classes e Subclasses

Na figura 20 visualiza-se a classe “Pacote de Conteúdo” e a sua subclasse “Pacote de Recurso”, o Protégé permite intuitivamente criar estas classes e subclasses que geralmente são representadas pela seguinte expressão lógica:

$$\forall x [\text{Resource_Package}(x) \rightarrow \text{Content_Package}(x)]$$

Desta forma tem-se em linguagem natural: “Todo Pacote de Recurso é um Pacote de Conteúdo”. A figura 21 apresenta em OWL tal expressão.

```

1 <owl:Class rdf:ID="Resource_Package">
2   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
3     Defines a mechanism for packaging Assets and SCOs provides a common medium for exchange.
4   </rdfs:comment>
5   <rdfs:subClassOf rdf:resource="#Content_Package"/>
6 </owl:Class>

```

Figura 21: Representação em OWL da Classe Pacote de Conteúdo

Este relacionamento taxonômico é expresso em OWL através dos elementos `Owl:Class` e `rdfs:subClassOf`, enquanto o elemento `rdfs:comment` é destinado a escrever um comentário do tipo string.

6.2 Relacionamento e Propriedades das Classes

No Protégé OWL existem dois tipos de propriedades: objetos, tipo de dado. Com as propriedades de objeto pode-se criar propriedades que permitem definir relacionamento entre classes e propriedades que são instâncias de classes. Propriedades “Tipos de dados” são variáveis para representar atributos no domínio especificando o seu valor, ou seja, um elemento do tipo *string*, booleano, inteiro, ponto flutuante, etc. Propriedade de tipo de dado pode ser também uma anotação, que serve para atribuir notas com o uso de elementos pré-definidas em RDF, como por exemplo, o elemento “`pref:Label`”, que permite adicionar um termo preferido a um rótulo legível por humanos. Na METHONTOLOGY as propriedades do tipo “objeto” e propriedades “tipos de dados” são denominadas de atributos de classe e atributos de instâncias, respectivamente.

No editor Protégé cria-se primeiramente a propriedade, depois é necessário especificar suas respectivas classes. Se for uma propriedade do tipo objeto definem-se as classes principais (Domain) e as classes de alcance (Range). Se a propriedade for Tipo de Dado é necessário definir também as classes principais e neste caso o *Range* é o tipo de valor aceito (*string*, inteiro etc.). Pode-se também criar sub-propriedades para o domínio, assim as propriedades são organizadas em uma estrutura hierárquica permitindo enriquecer a modelagem de atributos. A figura 22 apresenta todas as propriedades do domínio.

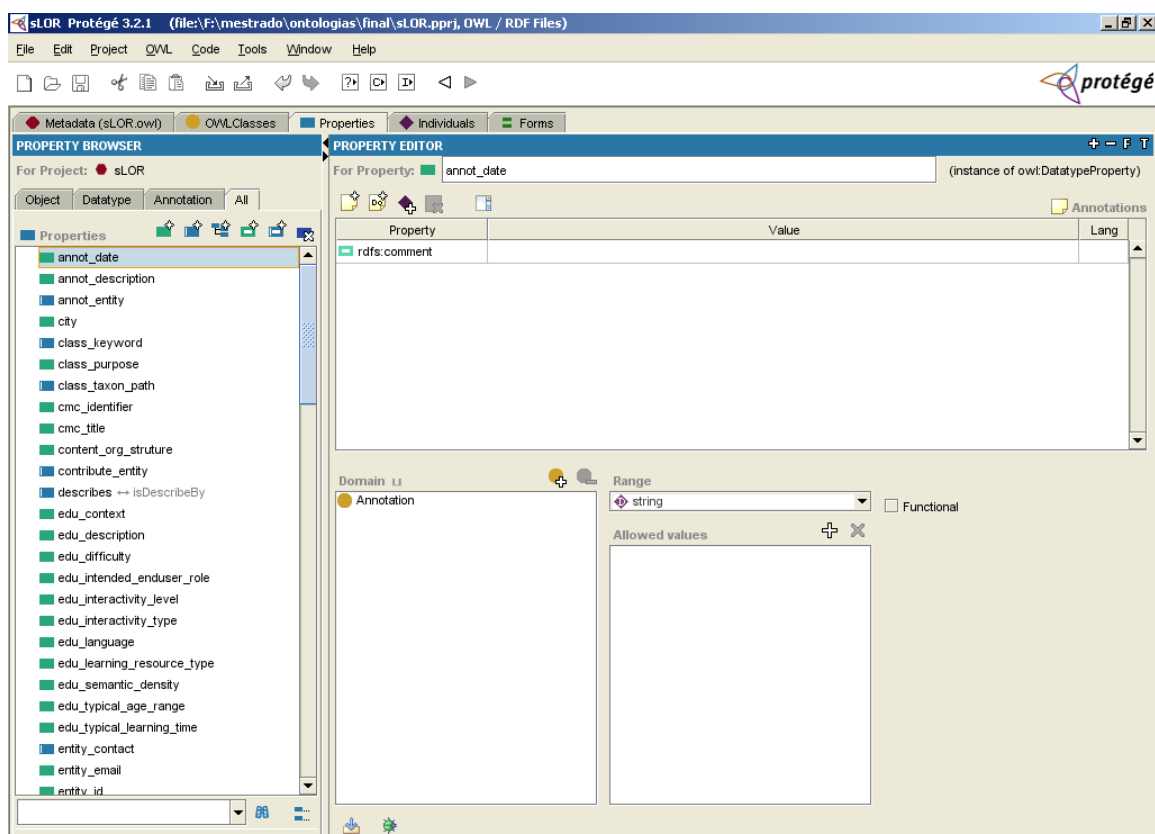


Figura 22: Propriedades do Domínio

No Protégé OWL, as propriedades de objeto são amostradas na cor azul, enquanto as propriedades de tipo de dado e anotação são amostradas na cor verde. O exemplo do conceito LOM e as subclasses “Geral” e “Educativo” e seus respectivos atributos e alguns métodos são apresentados no diagrama em UML na figura 23.

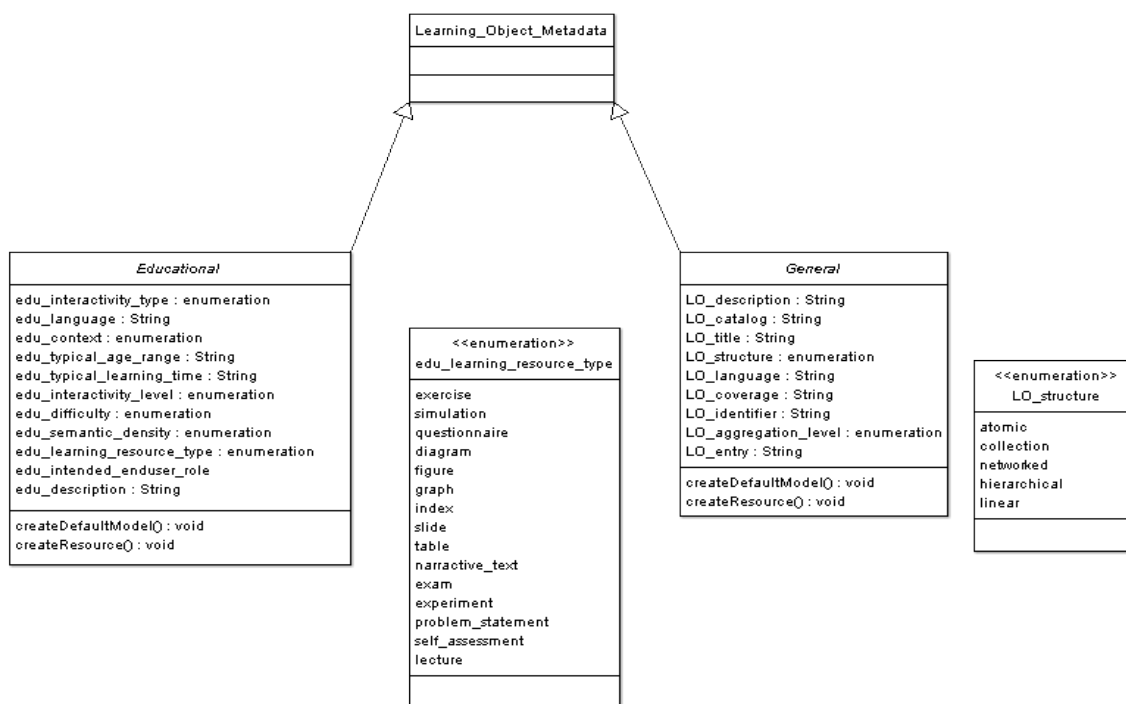


Figura 23: Exemplo de um Simples Diagrama de Classes

Em OWL pode-se especificar valores referentes aos atributos, como no caso do atributo “edu_learning_resource_type” que é um atributo do tipo *string* destinado a definir os tipos de recursos do elemento “Educativo” do LOM. Como pode ser visualizado na figura 23, em UML este atributo “edu_learning_resource_type” e “LO_structure” são declarados como “enumeration” que especifica um tipo enumerado e seus respectivos valores através da aplicação de um estereótipo. As operações “createDefaultModel” são destinadas à criação do modelo e “createResource” é destinado a criação do recursos RDF. Essas operações são implementadas utilizando métodos da biblioteca Jena.

Os relacionamentos têm o papel de relacionar uma classe à outra classe. As relações binárias em OWL são criadas através de propriedades. A especificação através da propriedade do tipo “Objeto” permitiu a formalização de diversos relacionamentos no domínio. O exemplo de relacionamento “é representado por” pode ser visualizado na Figura 24.

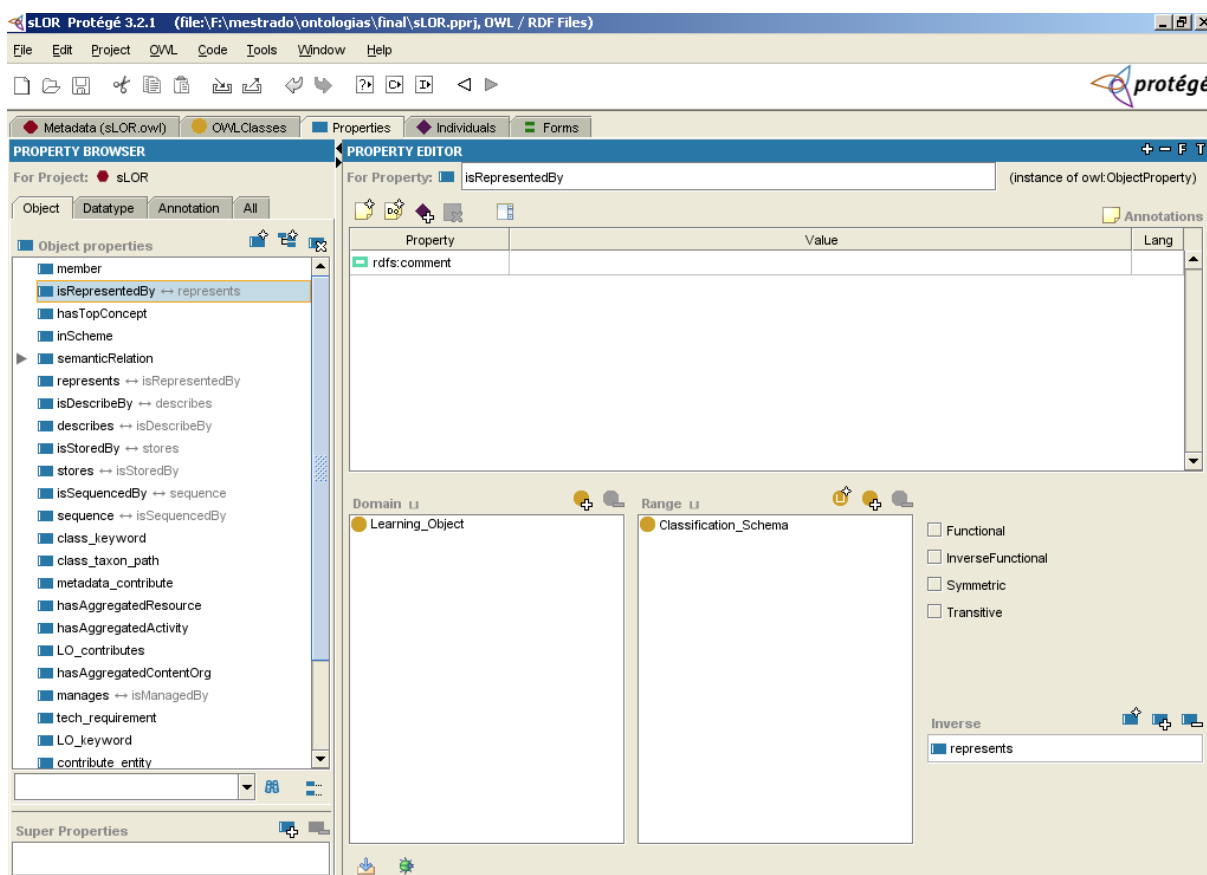


Figura 24: Propriedades de Objeto no Domínio

O relacionamento “é representado por”, ou “isRepresentedBy(x,y)” explicita que a classe Objeto de Aprendizagem (x) é representada por Esquema de Classificação (y). De acordo com Horridge et. al. (2004), em OWL cada propriedade do tipo “objeto” pode ter uma propriedade inversa correspondente. Se uma propriedade liga uma instância x com uma instância y, então a propriedade inversa ligará y com x. Este tipo de relacionamento pode ser representado da seguinte forma:

InverseOf: se $P1(x,y)$ então $P2(y,x)$

O exemplo citado na figura 24 ilustra um relacionamento do tipo inverso, se Objeto de Aprendizagem (x) é representado por (P1) Esquema de Classificação (y), conclui-se que Esquema de Classificação (y) representa (P2) Objeto de Aprendizagem (y). A figura 25 apresenta a propriedade inversa do objeto “é representado por” em OWL.

```

1  <owl:ObjectProperty rdf:about="#isRepresentedBy">
2    <rdfs:range rdf:resource="#Classification_Schema"/>
3    <rdfs:domain rdf:resource="#Learning_Object"/>
4    <owl:inverseOf rdf:resource="#represents"/>
5  </owl:ObjectProperty>
6
7  ...
8
9  <owl:ObjectProperty rdf:ID="represents">
10    <owl:inverseOf>
11      <owl:ObjectProperty rdf:about="#isRepresentedBy"/>
12    </owl:inverseOf>
13    <rdfs:range rdf:resource="#Learning_Object"/>
14    <rdfs:domain rdf:resource="#Classification_Schema"/>
15  </owl:ObjectProperty>

```

Figura 25: Propriedade Inversa

A linha 1 da figura 25, é declarado a propriedade do objeto, neste caso, “é representado por”, logo em seguida nas linhas 2 e 3 são definidas as classes de alcance e o domínio (Super-classe) no qual o relacionamento pertence, respectivamente. Na linha 9 inicia a mesma estrutura, porém é declarado o relacionamento inverso “representa”.

As propriedades de objeto no vocabulário OWL possuem características adicionais que permitem realizar inferência nas classes. As propriedades são:

- Funcional;
- Funcional Inversa;
- Simétrica;
- Transitiva.

As propriedades funcionais definem que um dado indivíduo é relacionado com mais de uma classe através da propriedade, ou seja, mais de uma classe pode ter o mesmo valor. A representação é feita através da seguinte forma:

FunctionalProperty: se $P(x,y)$ e $P(x,z)$ então $y=z$

A declaração da Propriedade Funcional em OWL é apresentada na figura 26:

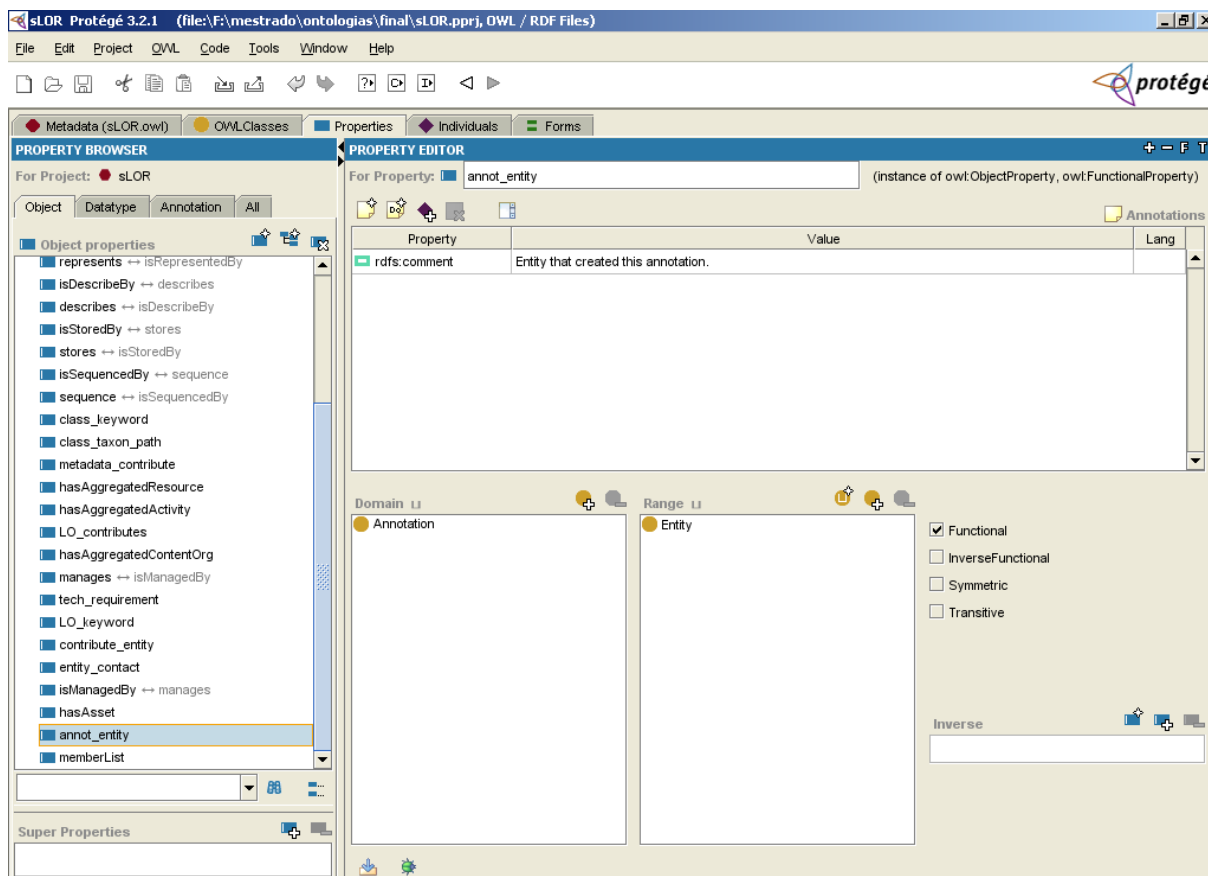


Figura 26: Propriedade Funcional de um Objeto

A propriedade “Anotação de Entidade” é uma instância da classe “Entidade” e pertence ao domínio “Anotação” desta forma, todos os valores (indivíduos) pertencem as duas classes.

Uma propriedade funcional, pode ser também uma propriedade de valor simples, ou seja, um tipo de dado que deve possuir pelo menos um valor. Este exemplo pode ser visualizado na figura 27.

```

1 <owl:FunctionalProperty rdf:ID="LO_title">
2   <rdfs:domain rdf:resource="#General"/>
3   <rdfs:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
4   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
5   <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
6     >Name given to this learning object.</rdfs:comment>
7 </owl:FunctionalProperty>

```

Figura 27: Propriedade Funcional de um Tipo de Dado

De acordo com a figura 27, a propriedade “Título do Objeto de Aprendizagem” (LO_title) é um tipo de dado (linha 3) e possui valor do tipo *String* (linha 5). Esta propriedade é funcional (linha 1) por que todo objeto de aprendizagem deve possuir pelo menos 1 título. Até o presente momento desta pesquisa não foi identificada uma relação Funcional Inversa.

Uma propriedade Simétrica é quando uma propriedade “P” relaciona um indivíduo “a” ao indivíduo “b” e o indivíduo “b” é também relacionado ao indivíduo via a propriedade “P” (HORRIDGE et. al., 2004). Esta característica de relacionamento pode ser expresso da seguinte forma: Se $P(x,y)$ então $P(y,x)$. A figura 28 apresenta o tipo de relacionamento simétrico para a propriedade “é associado”.

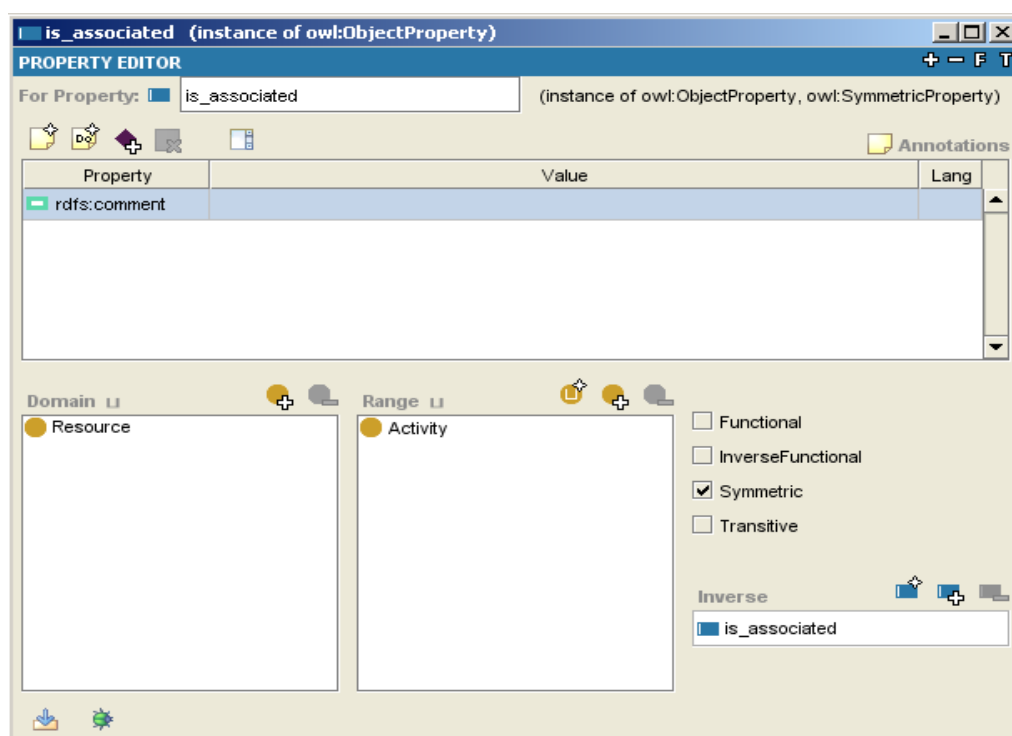


Figura 28: Propriedade Simétrica

O exemplo na figura 28 representa a relação “é associado” entre as classes Recurso e Atividade, pois um Atividade possui um Recurso associado assim como um Recurso pode está associado a uma ou mais Atividades. Por enquanto não foi detectada nenhuma Propriedade Transitiva para o presente modelo.

6.3 Axiomas e Restrições

No modelo proposto efetuou-se a formalização de alguns axiomas que foram apresentados na sessão 5.1 utilizando o Protégé. A figura 26 apresenta o axioma referente a classe Coleção Ordenada do Esquema de Classificação.

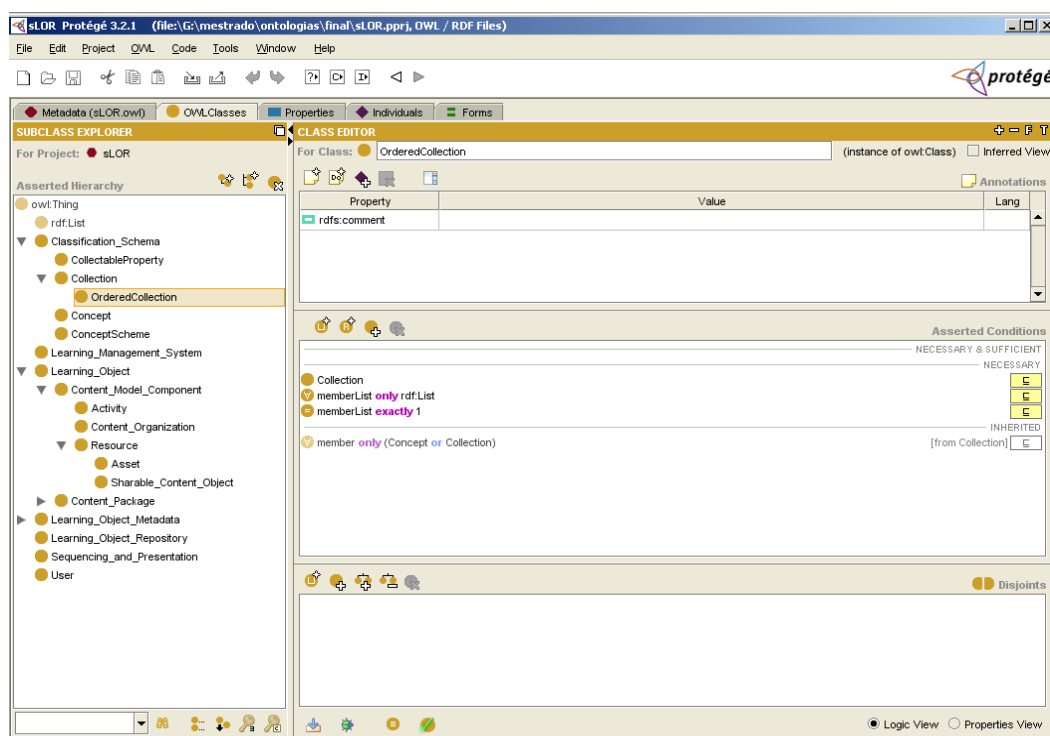


Figura 29: Axioma para Coleção Ordenada

Na figura 29 a propriedade “Lista de Membros” tem exatamente um valor, expressando assim uma restrição para a classe “Coleção Ordenada”. Todos os valores da propriedade “Membro” é uma união entre as classes “Conceito” e “Coleção” para o Esquema de Classificação, esta regra é mostrada na figura 30.

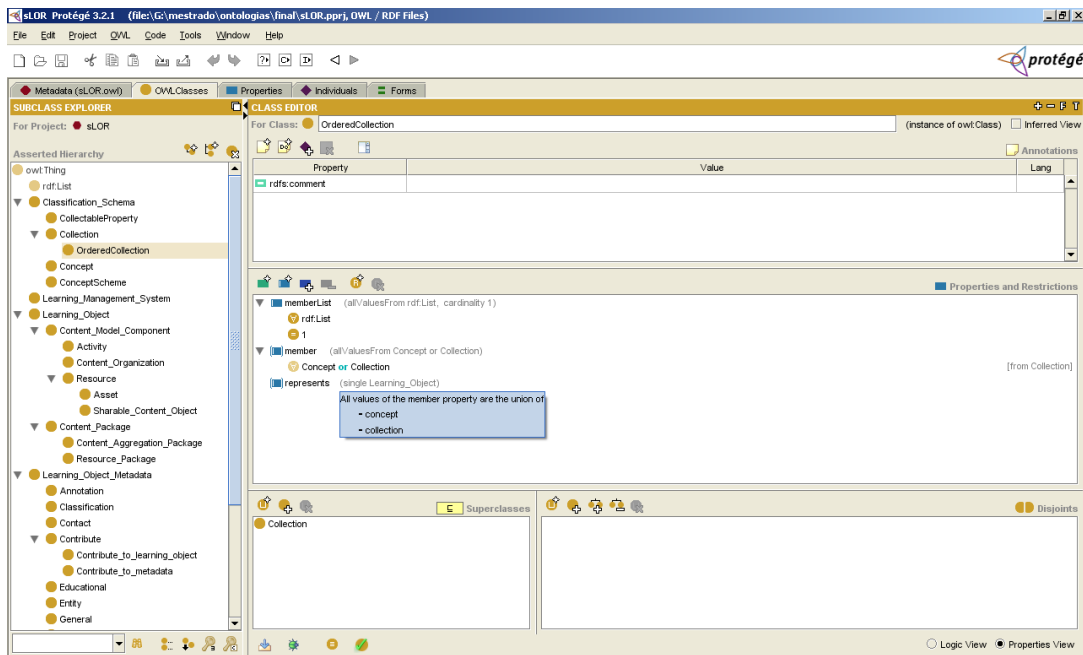


Figura 30: Axioma para Coleção Ordenada 2

Outro axioma referente à classe Objeto de Aprendizagem é ilustrado na figura 31.

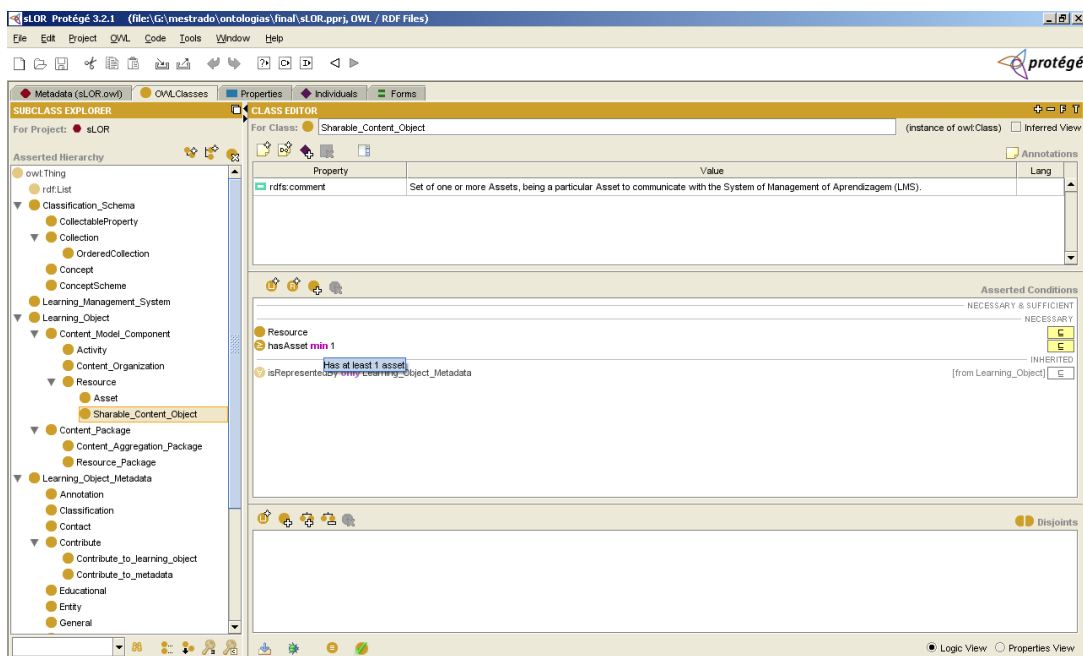


Figura 31: Exemplo de Axioma

Na figura 31 apresenta-se uma condição necessária. Um Objeto Compartilhado de Aprendizagem (SCO) deve ter necessariamente pelo

menos 1 Asset, ou seja, um SCO é um conjunto de 1 ou mais Assets, portanto deve possuir 1 Asset, podendo ser um arquivo de áudio, vídeo, texto, hipertexto ou qualquer outro formato digital.

6.4 Utilização do Framework Jena

Nesta seção será descrito o uso do Framework Jena para desenvolvimento de aplicações para Web Semântica. Jena possui uma API para RDF e uma API para OWL que fornece mecanismos para instanciar, exportar e realizar transações com banco de dados relacionais. Jena possui construtores em Java que permitem realizar a manipulação de ontologias escritas em RDF e OWL, além de fornecer suporte à SPARQL, que funciona como um protocolo e linguagem destinada a consultas em documentos RDF. Será apenas realizada uma simples apresentação do Jena, pois o Repositório não foi implementado até o presente momento.

A classe de `ModelFactory` do Jena é o meio de criar tipos diferentes de modelos e `ModelFactory.createDefaultModel()` é o método para chamar este modelo. Este método retorna uma instância de `Model`, no qual pode ser usado para criar um recurso. Depois que os recursos são criados, podem ser feitas declarações sobre eles e serem adicionadas ao modelo (MCCARTHY, 2007).

Em Jena um sujeito de uma declaração é sempre um `Resource`, o predicado é representado por `Property`, e o objeto é também um outro `Resource` ou valor literal. Literais são representados em Jena pelo tipo `Literal`. Todos estes tipos são compartilhados em uma interface comum, `RDFNode`. As instâncias são criadas como o método `Model.createProperty()` (MCCARTHY, 2007).

A maneira mais simples de adicionar declarações a um modelo é chamando `Resource.addProperty()`. Este método cria uma declaração no modelo com `Resource` como seu sujeito. O método faz exame de dois parâmetros, de uma `Property` que representam o predicado da declaração, e do objeto da declaração. O método `addProperty()` é sobrecarregado: Uma sobrecarga pega um `RDFNode` como o objeto, assim um `Resource` ou um `Literal` podem ser usados. Há também as sobrecargas da conveniência

que pegam de um literal representado por uma primitiva em Java ou uma String (MCCARTHY, 2007).

Um simples exemplo de um modelo em RDF apresentado em um grafo na seção 3.1.2, pode ser escrito conforme a figura 32.

```
package jena;
import com.hp.hpl.jena.rdf.model.*;

public class CourseModel extends Object {

    static String courseURI = "http://netclass.ufma.br/Course/IA/Module/Neural_Networks";
    static String hasModule = "Redes Neurais";

    public static void main (String args[]) {
        Model model = ModelFactory.createDefaultModel();
        Resource Course = model.createResource(courseURI);
        Course.addProperty(RDF.type, hasModule);
    }
}
```

Figura 32: Criação de um Modelo em RDF utilizando Jena

A classe apresentada na figura 32, cria apenas um modelo na memória (O curso tem módulo Redes Neurais), todavia com Jena pode-se criar um modelo e apresentá-lo em um documento RDF. A figura 33 demonstra uma implementação que escreve o modelo em RDF.

```
package jena;
import com.hp.hpl.jena.rdf.model.*;

public class CourseModel2 extends Object {

    static String hasArea = "Ciência da Computação";
    static String hasTitle = "Introdução a Redes Neurais";
    static String hasInstructor = "Jaime Moura";
    static String hasModule = "Redes Neurais";
    static String hasAula = "Roosevelt Lins";
    static String courseURI = "http://netclass.ufma.br/Course/IA/Module/Neural_Networks";

    public static void main (String args[]) {

        String courseURI = "http://netclass.ufma.br/Course/IA/Module/Neural_Networks";
        String hasArea = "Ciência da Computação";
        String hasTitle = "Introdução a Redes Neurais";
        String hasInstructor = "Jaime Moura";
        String hasModule = "Redes Neurais";
        String hasAula = "Conexões Sinápticas";

        Model model = ModelFactory.createDefaultModel();

        Resource Roosevelt
        = model.createResource(courseURI)
            .addProperty(RDF.type, hasArea)
            .addProperty(RDF.type, hasTitle)
            .addProperty(RDF.type, hasInstructor)
            .addProperty(RDF.type, hasModule)
            .addProperty(RDF.type, hasModule);

        model.write(System.out);
    }
}
```

Figura 33: Modelo que gera um documento RDF

A execução do programa gera um simples documento em RDF, mostrado na figura 34.

```

1 <rdf:RDF
2   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
3   <rdf:Description rdf:about="http://netclass.ufma.br/Course/IA/Module/Neural_Networks">
4     <rdf:type>Conexões Sinápticas</rdf:type>
5     <rdf:type>Redes Neurais</rdf:type>
6     <rdf:type>Jaime Moura</rdf:type>
7     <rdf:type>Introdução a Redes Neurais</rdf:type>
8     <rdf:type>Ciência da Computação</rdf:type>
9   </rdf:Description>
10 </rdf:RDF>

```

Figura 34: Documento RDF gerado pelo Jena

Pode-se realizar consultas em grafos RDF através da linguagem SPARQL, similar a linguagem SQL (Structured Query Language) usada em banco de dados relacionais. SPARQL é orientada a dados, pois é destinada apenas a consultar informações em modelos RDF, não permitindo inferência em suas consultas. Todavia, o modelo Jena pode ser utilizado para prover raciocínios sobre demanda em modelos OWL (JENA, 2007).

Existem quatro formas de realizar consultas: SELECT, CONSTRUCT, DESCRIBE e ASK. SELECT Retorna uma tabela de resultados. CONSTRUCT retorna um grafo RDF, baseado em um *template* de uma consulta. DESCRIBE retorna um grafo RDF baseado em qual processador de consulta é configurador para retorno. ASK é uma consulta booleana. Uma simples consulta é realizada de acordo com

```

1 SELECT ?x
2 WHERE
3   { ?x <http://www.w3.org/2001/vcard-rdf/3.0#FN> "John Smith" }

```

Figura 35: Simples Consulta (JENA, 2007)

Executando esta consulta em linha de comando o seguinte resultado é obtido.

```

-----
| x                                     |
=====
| <http://somewhere/JohnSmith/> |
-----

```

Figura 36: Resultado de uma Simples Consulta (JENA, 2007)

De acordo com a figura 30, é criado o comando `SELECT` para selecionar um determinado recurso e a cláusula `WHERE` especifica a URI onde está localizado o recurso. Jena permite ainda diversas formas de armazenar as instâncias em banco de dados, permitindo desta forma persistência dos dados.

7 CONCLUSÃO

A Web Semântica está em pleno processo de maturação, mesmo com diversas ferramentas, linguagens e padrões, desenvolver ontologias e sistemas inteligentes para Web ainda é uma tarefa complexa. Esta complexidade está diretamente ligada à própria proposta da ontologia que é fornecer uma especificação formal e compartilhada. Para que esta especificação seja compartilhada deve-se existir um consenso, pois um modelo de conhecimento pode ser aplicado e compartilhado em um contexto específico, mas não em todos os contextos, já que contextos diferentes existem implicações sociais e culturais. A solução é reutilizar ontologias já criadas, adaptando-as de acordo com as necessidades e especificidades de cada contexto.

Outro fator que convém ressaltar refere-se à extensibilidade das ontologias, ou seja, uma ontologia nunca está finalizada por completo, conforme for o aperfeiçoamento da ciência e descobrimento de novas especificidades semânticas em determinados domínios, as ontologias devem ser ampliadas e adaptadas. Ainda existem poucas ferramentas que permitem o desenvolvimento e modelagem de aplicações para Web Semântica, até o presente momento a linguagem Java através da API Jena é a mais adequada para o desenvolvedor criar sistemas. Paralelamente a este ponto, existem ainda poucos repositórios que disponibilizam ontologias em OWL, ocasionando dificuldades no reuso de modelos.

Os padrões de metadados são ainda elementos que não possuem semântica, mesmo sendo representados em XML, ainda existem diversas problemáticas para se atribuir significado nestes documentos. Desta forma, o mapeamento de metadados em ontologias tem sido a solução ideal para provê semântica. Assim diversas iniciativas de metadados tem sido criadas para catalogar e indexar recursos na web. No que diz respeito à indexação de conteúdos na Web, a teoria da classificação sustenta a prática de organização do conhecimento na Web. Sob o ponto de vista filosófico, os principais preceitos da classificação auxiliam a compressão da complexa atividade de representar classes e terminologias fundamentais para a elaboração dos conceitos para esquema de classificação. Pode-se observar

que este campo necessita de estudos que possam apresentar soluções reais para organização do conhecimento na Web. Softwares para sistemas de organização do conhecimento Web devem fornecer suporte a metadados, visando o compartilhamento e reuso de esquemas de classificação para organização de conteúdos na Web. Em Repositório de Objetos de Aprendizagem estas estruturas taxonômicas permitem um controle e sistematização terminológica relevante para o processamento temático de documentos, ocasionando melhorias para organização e recuperação da informação.

Para classificação dos LO, percebeu-se que as ontologias são poderosas para representação de conceitos, já que servem para modelagem de sistemas baseados em conhecimento e criar os a esquemas de classificação e catalogação. Os tesauros e outros esquemas utilizados pra indexação de conteúdos podem ser enriquecidos com as ontologias, já que pode-se implementar um modelo conceitual através da abordagem ontológica para estruturar taxonomias e definir relações semânticas.

Outro fator importante é as ontologias serviram para compreensão do modelo de referência SCORM, fundamental para que outros desenvolvedores implementem seus repositório de objetos de aprendizagem com base em especificações internacionais. Esta dissertação foi apenas o primeiro passo para formalização dos conceitos acerca do domínio objetos de aprendizagem. Desta forma, será investigado mais pressupostos sobre aprendizagem baseada em objetos de aprendizagem e o acréscimo de novas classes, relacionamentos, propriedades e axiomas.

8 TRABALHOS FUTUROS

Esta pesquisa limitou-se a compreensão dos principais padrões de metadados para objetos de aprendizagem e sua representação semântica. Desta forma realizou-se a modelagem ontológica dos padrões LOM e Skos-Core, para descrição e representação de objetos de aprendizagem para Web Semântica. Assim em estudos futuros será acrescentada a documentação referente ao modelo IMS de Sequenciamento e Navegação da especificação SCORM na ontologia do Repositório de Objetos de Aprendizagem. Poderá ser definido também, mais axiomas e regras para realizar inferências no modelo ontológico. Outro ponto relevante é a criação de métodos para processamento das instâncias, criando desta forma um Sistema para Gerenciamento de Conteúdo de Repositórios que permitirá criar, importar, exportar e disseminar LOs, interagindo desta forma com outros LMSs e LORs existentes na Web padronizados conforme as normas para metadados educacionais.

Estudos no campo da Recuperação e Filtragem de Informação podem ser introduzidos nesta pesquisa, através de propostas de sistemas semânticos que possibilitem o acesso à conteúdos com ferramentas de busca inteligentes. Uma ferramenta desta natureza pode ser um agente de busca destinado a consulta, filtragem e mineração do conhecimento contido nos modelos ontológicos de metadados. Para tal, é necessário um modelo de usuário para que agentes inteligentes possam entregar LOs de acordo com seu perfil cognitivo, provendo deste modo sistemas adaptativos que tornam as aplicações da educação baseada na Web mais usáveis e acessíveis.

REFERÊNCIAS

ADL. SCORM Specification V1.3. Disponível em: <http://www.adlnet.org>. Acesso em: 25 jan. 2006.

AGÜERRA, José Ramón Pérez. Automatización de tesauros y su utilización en la Web semántica. Disponível em: <http://eprints.rclis.org/archive/00004176/01/automatizacion.pdf> . Acesso em 22 jun. 2006.

AROYO, Lora; POKRAEV, Stanislav; BRUSSEE, Rogier. Preparing SCORM for the Semantic Web. In **Proc. of the Int. Conf. on Ontologies, Databases and Applications of Semantics** (ODBASE03), Catania, Italy, 2003.

BENNACER, Nacéra; BOURDA, Yolaine; DOAN, Bich-lien. Formalizing for Queryng Learning Objects Using OWL. **IEEE International Conference on Advanced Learning Technologies (ICALT'04)**, p. 195, v. 0, 2004

ALESSO, H. Peter; SMITH, Craig F. **THINKING ON THE WEB**: Berners-Lee, Gödel, and Turing. Chichester: John Wiley & Sons, 2006.

AVGERIOU, P. et al. (2003). Towards a Pattern Language for Learning Management Systems. *Educational Technology & Society*, 6(2), 11-24, Disponível: <http://ifets.ieee.org/periodical/6-2/2.html>. Acesso em: 12 nov. 2006.

BARBOSA, Alice Príncipe. Classificações facetadas. **Ciência da Informação**. Rio de Janeiro, v.1, n. 2, p. 73-81, 1972.

BERNERS-LEE, Tim; HENDLER, James; LASSILA, Ora. **The Semantic Web. Scientific American**: Feature Article, May 2001. Acesso em: 22 dez 2006. Disponível em: <http://www.scientificamerican.com/2001/0501issue/0501bemers-lee.html>

BEHRENDT, Wernher, ARORA, Nitin. The semantic Web: the network of meanings in the network of documents. In: **Web Engineering**: the discipline of systematic development of Web Applications, 2003.

BONIFACIO, Matteo. A Peer-to-Peer Solution for Distributed Knowledge Management In: **SemanticWeb and Peer-to-Peer Decentralized Management and Exchange of Knowledge and Information**. Springer-Verlag, 2006.

BROISIN, J. et al., Bridging the gap between learning management systems and learning object repositories: exploiting learning context information. In: **Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/ E-Learning on Telecommunications Workshop**. AICT/SAPIR/ELETE 2005. Proceedings, vol.00, p. 478- 483, jul.17-20, 2005.

CAMPOS, Maria Luiza de Almeida. Modelização de domínios de conhecimento: uma investigação de princípios fundamentais. **Ciência da Informação**. Brasília, v33, n.1, 2004.

CORCHO, Oscar; FERNÁNDEZ-LÓPEZ, Mariano; GÓMEZ-PÉREZ; Asunción Methodologies, tools and languages for building ontologies: where is their meeting point? **Data Knowl. Eng.** v.46, n.1, p.41-64.2003.
CHAMPIN, Pierre-Antoine. **RDF Tutorial**. Disponível em: <http://bat710.univ-lyon1.fr/~champin/rdf-tutorial/>. Acesso em 06 dez. 2006.

DACONTA, Michael C.; OBRST, Leo J.; SMITH, Kevin T. **The Semantic Web: a Guide to the Future of XML, Web Services, and Knowledge Management**. 2006.

DACONTA, Michael. Formal **Taxonomies for the U.S. Government**. Disponível em: <http://www.xml.com/pub/a/2005/01/26/formtax.html>. Acesso em 12 jan. 2007.

DACONTA, Michael C.; OBRST, Leo J.; SMITH Kevin T. **The Semantic Web: a Guide to the Future of XML, Web Services, and Knowledge Management**. Chichester: John Wiley & Sons: 2003.

DAVIES, John; FENSEL, Dieter; HARMELEN, Frank van. **Towards the semantic Web: Ontology-driven Knowledge Management**. Chichester: John Wiley & Sons, 2003.

DEVEDZIĆ, V. Understanding ontological engineering. **Communications ACM 45**, 4.Apr. 2002, 136-144. Disponível em: <http://doi.acm.org/10.1145/505248.506002>. Acesso: 15 nov 2006.

DING, Li et at. Using Ontologies in the Semantic Web: a Survey. In: **Ontologies in the Context of Information Systems**. Springer, 2005. Disponível em: http://ebiquity.umbc.edu/_file_directory_/papers/209.pdf. Acesso em 30 set. 2006.

FERNÁNDEZ-LÓPEZ, M.; GÓMEZ-PÉREZ, A.; JURISTO, N. Methontology: from ontological art towards ontological engineering. Proc. Symposium on Ontological Engineering of AAAI. p.33-40. 1997.

FOAF. **FOAF Vocabulary Specification**. Disponível em: <http://xmlns.com/foaf/0.1/> Acesso em: 10 fev. 2007.

GAŠEVIĆ, D; HATALA, M. Ontology mappings to improve learning resource search. British Journal of Educational Technology, Special issue on Advances of Semantic Web for E-learning: Expanding learning frontiers, v.37, n. 3, 375–389, 2006.

HORRIDGE et. al. A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools Edition 1.0. Disponível em:

<http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf>. Acesso em 10 out. 2006.

IEEE Learning Technology Standards **Committee (LTSC) (2001) Draft Standard for Learning Object Metadata Version 6.1**. Disponível em: <http://ltsc.ieee.org/doc/> . Acesso em: 15 nov 2006.

IMS Global Learning Consortium Inc. Content Packaging Specification. Disponível em: <http://www.imsglobal.org/overview.cfm>. Acesso em out. 2006.

INGWERSEN, Peter; WORMELL, Irene. Ranganathan in the perspective of advance information retrieval. **Libri**, v.42, n.3, p.184-201, July/Sept. 1992.

JIMÉNEZ, Antonio García. **Instrumentos de Representación del Conocimiento: Tesauros Versus Ontologías**. Disponível em: <http://www.um.es/fccd/anales/ad07/ad0706.pdf>. Acesso em 12 jan. 2007
JENA. **A Semantic Web Framework for Java**. Disponível em: <http://jena.sourceforge.net/>. Acesso em 20 out. 2006.

KAMTHAN, Pankaj; PAI, Hsueh-leng. Representation of Web Application Patterns in OWL. In: MONASH, David Taniar; WENNY, Johanna. **Web Semantics and Ontology** Rahayu.p. 41-64, Idea Group. 2006.

Li et. al. Ontology Mappings to Enhance Interoperability of Knowledge Domain Taxonomies. In **Proc. of the 3rd Annual Scientific Conference - LORNET Research Network** (I2LOR-05), Canada, November 2005.

LIMA, Gercina Ângela Borém. A análise facetada na modelagem conceitual de sistemas de hipertexto: uma revisão de literatura. **Perspectivas em Ciência da Informação**, Belo Horizonte, v. 7, n. 2, p. 189-196, jul./dez. 2002

LONGMIRE, W. A Primer On Learning Objects. American Society for Training & Development. Virginia. USA. 2001. Disponível em: <http://www.learningcircuits.org/2000/mar2000/Longmire.htm>. Acesso em: 15 nov. 2006.

MCCARTHY, Philip. Introduction to Jena. Disponível em: <http://www-128.ibm.com/developerworks/java/library/j-jena/>. Acesso em: 20 mar. 2007.

MERRIL, M. David, **Knowledge Objects and Mental-Models** Disponível em: <http://id2.usu.edu/Papers/KOMM.PDF>. Acesso em out.2006.

MILES, Alistair J.; ROGERS, Nikki; BECKETT, Dave. **Migrating Thesauri to the Semantic Web**. Disponível em: <http://www.w3.org/2001/sw/Europe/reports/pdf/8.8.pdf>. Acesso em: 20 jun 2006.

MILES, Alistair. **Quick Guide to Publishing a Thesaurus on the Semantic Web**. Disponível em: <http://www.w3.org/TR/2005/WD-swbp-thesaurus-pubguide-20050517/>. Acesso em: 20 jun 2006.

MILES, Alistair; MATHEWS, Brian; WILSON, Michael, Brickley, Dan. **SKOS Core 1.0 guide**. Disponível em: <http://www.w3.org/2001/sw/Europe/reports/thes/1.0/guide/20040504/>. Acessado em: 20 nov. 2006.

MILES, Matthes; Wilson. **SKOS Core: Simple Knowledge Organisation for the Web**. Disponível em: <http://epubs.cclrc.ac.uk/bitstream/675/dc2005skospapersubmission1.pdf>. Acesso em: 22 dez. 2006.

MOHAN, Permanand; BROOKS, Christopher. **Learning Object on the Semantic Web. Third IEEE International Conference on Advanced Learning Technologies (ICALT'03)**, p. 195, v. 0, 2003.

MOREIRA, Alexandra. Uso de ontologias em sistemas de informação computacionais. **Perspectivas em Ciência da Informação**, Belo Horizonte, v.7, n.1, p.39-48, jan./jun. 2002.

MOTELET, Olivier; BALOIAN, Nelson. Introducing learning management systems standards. In: Classroom. **IEEE International Conference on Advanced Learning Technologies ((ICALT'04)**, Proceedings. p.738- 740, 30 ago/set, 2004.

MUZIO, Jeanette; HEINS, Tanya; MUNDELL, Roger. **Experiences with Reusable eLearning Objects: from Theory to Practice**. Victoria, Canadá. 2001.

MILLER, Eric. **An Introduction to the Resource Description Framework D-Lib Magazine**. May 1998. Disponível em: <http://www.dlib.org/dlib/may98/miller/05miller.html>. Acesso em: 02 dez. 2006.

PAULSEN, M. F. (2003). Experiences with Learning Management Systems in 113 European Institutions. **Educational Technology & Society**, v. 6 n. 4, 134-148, Disponível em: http://ifets.ieee.org/periodical/6_4/13.pdf Acesso em: 12 nov. 2006.

PÉREZ-GÓMEZ, Assunción. **Ontological Engineering**: a state of the art. 2(3):33-43, Autumn 1999.

PIEADADE, M. Requião. **Introdução à teoria da classificação**. 2.ed. Rio de Janeiro: Interciência, 1983.

PINTO, H. Sofia et al. Distributed Engineering of Ontologies (DILIGENT). In: Staab, Steffen; Stuckenschmidt, Heiner (Eds.) **SemanticWeb and Peer-to-Peer Decentralized Management and Exchange of Knowledge and Information**. Springer-Verlag, 2006.

PANTELEYEV, M.G. et al. Intelligent Educational Environments Based on the semantic technologies. In **Proc. of the 2002 IEEE International Conference on Artificial Intelligence Systems**, IEEE Computer Society Press, 2002.

POLSANI, Pithamber R. Use and Abuse of Reusable Learning Objects Learning Technology Center, University of Arizona, USA. **Journal of Digital Information**, Volume 3 Issue 4 Article No. 164, 2003-02-19.

QU, Changtao; NEJDL, Wolfgang. Searching SCORM Metadata in a RDF-Based Elearning P2P Network Using XQuery and Query by Example, In **Proceedings ICAIT 2003**, p.81-85.

QIN, J.; HERNANDEZ, N. Ontological representation of learning objects: building interoperable vocabulary and structures. In: **Proceedings of WWW2004**, New York, May 17-22, 2004, 348-349. New York: ACM Press.

RAPUANO, Sergio; Zoino, Francesco. A Learning Management System Including Laboratory Experiments on Measurement Instrumentation. **Instrumentation and Measurement Technology Conference**, 2005. IMTC 2005. Proceedings of the IEEE. v.2, n. p. 1227- 1232. maio16-19, 2005.

RIPAMONTI L.A., PERABONI, C. Hyperkrep . Academic Communities: an ontology-based approach to content portability. Proc. of the In. **Conf. Open Culture: accessing and sharing knowledge: scholarly production and education in the digital age**, Milan, June 27-29 2005, Italy.

RDF Thesaurus Specification (draft). Acesso em : 20 setembro 2006.
Disponível em:
http://www.ilrt.bristol.ac.uk/publications/researchreport/r1011/report_html?ilrtyear=00

ROUYET, Juan Ignacio; MARTÍN, Víctor. SCORM and Dublin Core: a comparative study of the metadata. Disponível em:
http://www.cc.uah.es/spdece/papers/Rouyet_Final.pdf. Acesso em 10 jan. 2005.

SANTACRUZ-VALENCIA, L.P.; AEDO I; KLOOS C. Delgado. Objetos de aprendizaje: tendencias dentro de la web semántica. Disponível em:
<http://www.rediris.es/rediris/boletin/66-67/ponencia18.pdf>. Acesso em: 20 jun. 2005.

SESSINKI et al. Author-defined storage in the next generation learning management systems. **Third IEEE International Conference on Advanced Learning Technologies** (ICALT'03), p. 57- 61, v. 0, 2003.

SICÍLIA, Miguel-Angel; GARCÍA, Elena. On the Concepts of Usability and Reusability of Learning Objects. **The International Review of Research in Open and Distance Learning**, v.4, n.2 (2003).

SILVA, Roosevelt Lins; BENTES, Virgínia. LABIDI, Sofiane. Metadata Ontological for Semantic Learning Object Repository. In: **MTRS'07 - International Conference Metadata and Semantics Proceedings**. Springer, 2007.

SILVA, Roosevelt Lins; LABIDI, Sofiane. Representation and Content Aggregation in Learning Object Semantic Web-Based Repository. In: **ICBL - International Conference on Interactive Computer aided Blended Learning**, 2007, Florianópolis. ICBL - International Conference on Interactive Computer aided Blended Learning Proceedings, 2007.

SILVA, Roosevelt Lins et. al. Ontological Classification in Intelligent Learning Management System. In: Vicente P. Guerrero-Bote. (Org.). **Current Research in Information Sciences and Technologies - InSciT2006 Proceedings**. Badajoz: Open Institute of Knowledge, 2006.

SILVA, Roosevelt Lins. Repositório Semântico de Objetos de Aprendizagem. In: **A dimensão social da Biblioteca Digital na organização e acesso ao conhecimento**: aspectos teóricos e aplicados. São Paulo: CRUESP/Bibliotecas/IBICT. 2005.

STOJANOVIC, Ljiljana; STAAB, Steffen; STUDER, Rudi. Elearning based on the Semantic Web”, **WebNet 2001 - World Conference on the WWW and the Internet**.

STAMOU, Giorgos; KOLLIAS, Stefanos. **Multimedia Content and the SemanticWeb**: methods, standards and tools, 2006.

SURE, York; STUDER, Rudi. A Methodology for Ontology-based Knowledge Management. In: DAVIES, John; FENSEL, Dieter; HARMELEN, Frank van (org.). **Towards the semantic Web**: Ontology-driven Knowledge Management. Chichester: John Wiley & Sons, 2003, p.33-45.

TESCH JÚNIOR, José Roberto. **XML Schema**. Florianópolis: Visual Books, 2002.

TIDWELL, Doug. **XSLT**: Mastering XML Transformations. Sebastopol: Oreilly, 2001.

THE OTK TOOL REPOSITORY. Disponível em: <http://www.ontoknowledge.org/tools/ontoedit.shtml>. Acesso em 15 nov 2006.

THE PROTEGÉ. Disponível em: <http://protege.stanford.edu/>. Acesso em: 15 nov 2006.

UNICODE. **O que é Unicode?** Disponível em: <http://www.unicode.org/standard/translations/portuguese.html>. Acesso em: 28 nov. 2006.

W3C. **RDF/XML Syntax Specification** (Revised). Disponível em: <http://www.w3.org/TR/rdf-syntax-grammar/>. Acesso em 02 dez. 2006.

W3C. Learning about URIs. Disponível em: <http://www.w3.org/Addressing/#background>. Acesso em: 28 nov. 2006.

W3C. Namespaces in XML 1.0 (Second Edition). Disponível em: <http://www.w3.org/TR/REC-xml-names/>. Acesso em: 30 nov. 2006.

W3C. XML Schema. Disponível em: <http://www.w3.org/XML/Schema> Acesso em: 01 dez. 2006.

W3C SCHOOLS. Introduction to DTD. Disponível em: http://www.w3schools.com/dtd/dtd_intro.asp. Acesso em: 01 dez. 2006.

W3C SCHOOLS. Introduction to XSL. Disponível em: <http://www.w3schools.com/xsl/>. Acesso em: 05 dez. 2006.

W3C. Semantic Web. Disponível em: <http://www.w3.org/2001/sw/>. Acesso em: 20 jun. 2005.

W3C. Web Ontology Language (OWL). Disponível em: <http://www.w3.org/2004/OWL/>. Acesso em: 22 dez. 2006.

W3 SCHOOLS. RDF Tutorial. Disponível em: <http://www.w3schools.com/rdf/default.asp>. Acesso em: 02 dez. 2006.

W3C. RDF Vocabulary Description Language 1.0: RDF Schema. Disponível em: <http://www.w3.org/TR/rdf-schema/>. Acesso em: 02 dez. 2006.

W3C. SKOS Core Guide. Disponível em: <http://www.w3.org/TR/2005/WD-swpb-skos-core-guide-20051102/> Acesso em: 20 out. 2006a.

WATANABE, H.; KOGA, S.; KATO, K. Development of Learning Management System and SCO Presentation Program Based on SCORM. In: **Proceedings of the IEEE international Conference on Advanced Learning Technologies (Icalt'04)** - v.00, aug/set. IEEE Computer Society, Washington, DC, 870-871. 2004.

WEBODE PERFORMANCE BENCHMARK SUITE. Disponível em: <http://kw.dia.fi.upm.es/wpbs/> . Acesso em: 15 nov 2006.

WILEY, D. A. **Connecting learning objects to instructional design theory: A definition, a metaphor and a taxonomy**. Disponível em: <http://reusability.org/read/chapters/wiley.doc>. Publicado em 2003. Acesso em: 16 maio 2006.

WOUKEU, A. et al. Ontological Hypermedia in Education: A framework for building web-based educational portals. In: **Proceedings of ED-MEDIA**

2003-World Conference on Educational Multimedia: Hypermedia & Telecommunications, Honolulu: 2003.

APÊNDICE

APÊNDICE A – Glossário de Termos

Nome	Sinônimo	Acrônimo	Descrição	Tipo
Repositório de Objetos de Aprendizagem	Repositório de Objetos Educacionais, Biblioteca de Objetos de Aprendizagem,	LOR	Repositório destinado a armazenar recursos digitais que podem ser utilizados pro usuários para fins educacionais	Conceito
Descrito por	-	-	Relacionamento entre os conceitos objetos de aprendizagem e metadados	Relação
Usuários			Entidade responsável pelo gerenciamento do sistema	Conceito
Login Usuário			Nome do usuário para autenticação	
Senha de Usuário			Senha do usuário para acesso.	
Nome de Usuário			Nome do Usuário	
Email do Usuário			Email do usuário	
Formação do Usuário			Nível de formação do usuário	
Objetos de Aprendizagem		LO	Recurso digital que possa ser reusado para dar suporte à aprendizagem ou qualquer recurso, grande ou pequeno que possa ser entregue sob demanda através da internet.	Conceito
Gerenciado por	Administrado por		Relacionamento ente a classe usuário e Sistema de Gerenciamento de Aprendizagem e Repositório de Objetos de Aprendizagem	Relação
Componente de Modelo de Conteúdo		CMC	Nomenclatura que define os componentes de conteúdo das experiências de aprendizagem	Conceito
Título do Modelo de Conteúdo			Título referente ao componente do Modelo de Conteúdo	Atributo

Identificador do Modelo de Conteúdo			Identificador referente ao componente de Modelo de Conteúdo	Atributo
Atividade			Unidades de aprendizagem com conteúdo estruturado e planejado	Conceito
Organização de Conteúdo			Mapa que representa o uso planejado do conteúdo através de unidades estruturadas de instrução (Atividades)	Conceito
Recurso			Unidade digital que pode ser um SCO (conjunto de Assets) e ou um único Asset	Conceito
Tipo de Recurso			Informação sobre o tipo de recurso	Atributo
Arquivo do Recurso			Define qual ou quais arquivos compõe o recurso	Atributo
Conteúdo de Objeto Compartilhado		SCO	Conjunto de um ou mais Assets, sendo um Asset particular para se comunicar com o Sistema de Gerenciamento de Aprendizagem (LMS)	Conceito
Asset			Mídia de representação eletrônica, tais como o texto, imagens, som, objetos da avaliação ou qualquer outra parte de dados que podem ser fornecidos pela web e apresentados a um aprendiz	Conceito
Pacote de Conteúdo			Define como representar a Estrutura de Conteúdo e como agregar atividades a recursos de aprendizagem para movimentar entre diferentes ambientes	Conceito
Pacote de Agregação de Conteúdo			Processo de agregação de recursos (SCOs e Assets) em uma unidade de instrução (curso, módulo, capítulo etc.).	Conceito
Pacote de Recurso			Mecanismo para empacotar Assets e SCOs que não prover alguma organização lógica, contexto de aprendizagem ou taxonomia curricular. Usado para mover os SCOS e Assets de sistema para sistema	Conceito
Seqüenciamento e Apresentação			Especifica estratégias de seqüências. Define um conjunto de seqüências de informação para atividades.	
Sistema de Gerenciamento de		LMS	Aplicação destinada a gerenciamento de ambientes de ensino baseado na web	Conceito

Aprendizagem				
Metadado de Objetos de Aprendizagem	Metadado	LOM	Mecanismo para descrever instâncias específicas dos componentes do modelo de conteúdo	Conceito
Contribuidor			Elemento que pode ser usado para descrever quais entidades (pessoas e organizações) que tem contribuído para determinar o modelo de componente de conteúdo durante o ciclo de vida (criação, edição, revisão, publicação, etc.). O elemento contribuidor permite capturar todos indivíduos ou organizações envolvidas.	Conceito
Data			Data referente a contribuição	Atributo
Entidade			Responsável pela contribuição	Atributo
Papel			Papel atribuído ao responsável pela contribuição	Atributo
Geral			Descreve informações gerais sobre o modelo de conteúdo	Conceito
Nível de Agregação			A granularidade funcional destes objetos de aprendizagem. 1= é o maior nível de agregação.	Atributo
Catálogo			O nome ou designador da identificação ou esquema de catalogação para esta entrada. Um esquema de namespace.	Atributo
Cobertura			A época, a cultura, a geografia ou região, no qual o objeto de aprendizagem é aplicado.	Atributo
Descrição			Uma descrição textual do conteúdo do objeto de aprendizagem.	Atributo
Entrada			O valor do identificador com a identificação ou esquema de catálogo que designa ou identifica o objeto de aprendizagem. Um namespace do tipo String	Atributo
Identificador		ID	Um rótulo único globalmente que identifica um elemento (objeto de aprendizagem ou metadado).	Atributo
Palavra-chave	Frase		Uma palavra-chave ou frase descreve o tópico do objeto de aprendizagem.	Atributo
Língua	Idioma		A linguagem humana primária ou linguagem usada para o objeto de aprendizagem se comunicar com o usuário.	Atributo
Estrutura			Fundamentos da estrutura organizacional do objeto de	Atributo

			aprendizagem.	
Título	-	-	Nome dado ao objeto de aprendizagem. Atributo relacionado ao conceito Metadados de Objetos de Aprendizagem (LOM)	Atributo
Ciclo de Vida			Histórico ou o estado atual de um objeto de aprendizagem e quais componentes que tem afetado o objeto de aprendizagem durante sua evolução.	Conceito
Status			Estado final ou condição do objeto de aprendizagem	Atributo
Versão			Edição do objeto de aprendizagem	Atributo
Meta-metadados			Descreve informações sobre os metadados	Conceito
Esquema			O nome ou versão da especificação usada para criar a instância do metadado	Atributo
Técnico			Requisitos e características técnicas dos objetos de aprendizagem	Conceito
Requisitos Técnicos			Capacidade técnica necessária para usar o objeto de aprendizagem. Se houver requisitos múltiplos, todos serão requeridos, isto é, o conector lógico é E.	Conceito
Outros requisitos de plataforma			Informação sobre requisitos de software e hardware	Atributo
Duração			Tempo tomado para um objeto de aprendizagem contínuo Este elemento é útil para simulações, filmes e animações	Atributo
Formato			Tipos de dados técnicos de (todos os componentes) objetos de aprendizagem. Este elemento de dado pode ser usado para identificar o software necessário para acessar o objeto de aprendizagem.	Atributo
Localização			Uma string é usada para acessar o objeto de aprendizagem. Pode ser uma localização (isto é, uma URL), ou um método que determine a localização (uma URI)	Atributo
Tamanho			O tamanho do objeto de aprendizagem em bytes	Atributo
Notas de instalação			Descrição de como instalar o objeto de aprendizagem.	Atributo
Educacional			O tipo de característica educacional ou pedagógica do objeto de aprendizagem	Conceito

Tipo de Interatividade			Modo predominante da aprendizagem suportada por objetos de aprendizagem	Atributo
Contexto			O ambiente principal no qual a aprendizagem e o uso do objeto de aprendizagem é planejado	Atributo
Alcance típico de idade (typicalAgeRange)			Elemento que representa a idade do usuário final	Atributo
Tempo típico de aprendizagem	Duração		Tempo aproximado ou típico de duração do objeto de aprendizagem	Atributo
Nível de Interatividade			O grau de interatividade caracterizado para o objeto de aprendizagem. Interatividade neste contexto refere-se ao grau na aparência e comportamento do objeto de aprendizagem	Atributo
Dificuldade			Representa o grau de dificuldade do objeto de aprendizagem	Atributo
Densidade Semântica			O grau de concisão de um objeto de aprendizagem. A densidade semântica do objeto de aprendizagem pode ser estimada nos termos de seu tamanho, extensão, ou no caso de recursos auto-cronometrado, tais como o áudio ou duração de vídeo.	Atributo
Tipo de Recurso de Aprendizagem			Tipo específico de objeto de aprendizagem. O tipo o mais dominante será primeiro.	Atributo
Papel pretendido ao usuário final			Usuário (s) principal no qual o objeto de aprendizagem está designado.	Atributo
Descrição			Comentários de como os objetos de aprendizagem podem ser utilizados.	Atributo
Direitos			Descrever a propriedade intelectual e as condições de uso dos objetos de aprendizagem.	Conceito
Direitos autorais e Restrições			O direito autoral ou outras restrições aplicadas ao uso de objetos de aprendizagem	
Custo			Quando é necessário pagar para usar o objeto de aprendizagem	Atributo

Descrição de Direitos			Comentário sobre as condições de uso do objeto de aprendizagem	
Relação			Descreve o relacionamento entre um objeto de aprendizagem e outros objetos de aprendizagem,	Conceito
Tipo de Relação			Natureza do relacionamento entre a aprendizagem e o objeto de aprendizagem alvo identificado pela relação do recurso.	Atributo
Relação do Recurso			A referência do objeto de aprendizagem alvo	Atributo
Anotação			Comentários sobre uso do objeto de aprendizagem e informações sobre quando por quem o comentário foi criado.	Conceito
Data da anotação			Data que a anotação foi criada	Atributo
Entidade de Anotação			Entidade que criou a anotação.	Atributo
Descrição da Anotação			O conteúdo da anotação	Atributo
Classificação			Descreve onde o objeto de aprendizagem é incluído com um sistema de classificação particular	Conceito
Descrição da Classe			Descrição do objeto de aprendizagem relativo ao expressado. Classificação. Proposta para uma classificação específica, tal como disciplina, idéia, nível de habilidade, objetivos educacionais, etc.	Atributo
Caminho da Taxonomia da Classe			Um caminho taxonômico em uma classificação específica. Cada nível sucedido é um refinamento na definição do nível precedente.	Atributo
Palavra-chave da classe	Descritor		Palavra-chave ou frase descritiva do objeto de aprendizagem relativo ao expressado. Classificação. Propósito da classificação específica, tal como acessibilidade, segurança, os mais relevantes primeiro.	Atributo
Propósito da Classe			O propósito da classificação do objeto de aprendizagem	Atributo
Contato			Contato referente aos criadores do objeto de aprendizagem	Conceito
Cidade			A cidade referente	Atributo
Fax			Número de fax de contato	Atributo
Telefone			Número do telefone de contato	Atributo

Rua			Rua referente ao endereço de contato	Atributo
CEP			CEP referente ao endereço de contato	Atributo
Estado			Estado referente ao endereço de contato	Atributo
Esquema de Classificação			Sistema de Organização do Conhecimento para classificação dos conteúdos dos objetos de aprendizagem Estrutura hierárquica de áreas do conhecimento e respectivos termos para representação dos objetos de aprendizagem	Conceito
Propriedade Coletável			Uma propriedade que pode ser usada com skos:Collection. As seguintes regras se aplicam a essa propriedade: [(?x ?p ?c) (?c skos:member ?y) (?p rdf:type skos:CollectableProperty) implies (?x ?p ?y)]	Conceito
Coleção			Uma coleção de conceitos por significado. Coleções nomeadas podem ser usadas com propriedades de relações semânticas coletáveis, por exemplo, skos:narrower, quando desejamos que uma coleção de conceitos seja apresentada sob uma "etiqueta" em uma hierarquia.	Conceito
Conceito			Um determinado termo, que possui um significado.	Conceito
Esquema de Conceitos			Um conjunto de conceitos, opcionalmente incluindo declarações sobre relações semânticas entre esses conceitos.	Conceito
Coleção Ordenada			Uma coleção ordenada de conceitos, onde o agrupamento e a ordenação são significativos. Coleções ordenadas podem ser usadas com propriedades de relações semânticas coletáveis, onde você quer que um conjunto de conceitos seja apresentado em uma ordem específica, e opcionalmente sob uma "etiqueta".	Conceito
Título do Esquema de Conceito			Elemento do padrão Dublin Core destinado a nomear o esquema de classificação	Atributo
Criador do Esquema de Conceito			Elemento do padrão Dublin Core destinado a indicar a autoria do esquema de classificação	Atributo

Conceito principal			Um conceito principal em um esquema de conceito.	Atributo
Termo preferido			O termo lexical preferido para um recurso, em uma dada linguagem. Dois conceitos em um mesmo esquema de conceitos não podem ter o mesmo valor para skos:prefLabel em uma dada linguagem.	Atributo
Conceito específico			Um conceito de significação mais específica. Conceitos específicos são tipicamente representados como descendentes em uma hierarquia de conceitos (árvore).	Atributo
Conceito geral			Um conceito de significação mais genérica. Conceitos genéricos são tipicamente representados como parentes em uma hierarquia de conceitos (árvore).	Atributo
No esquema			Um esquema de conceito em que o conceito está incluído. Um conceito pode ser um membro de mais de um esquema de conceitos.	Atributo

APÊNDICE B – Atributos de Classe

Nome do Atributo	Nome do Conceito	Tipo de Valor	Cardinalidade	Valores
Termo Preferido	Conceito	String	(1,1)	
Identificador_CMC	Componente do Modelo de Conteúdo	String	(1,1)	
Título_CMC	Componente do Modelo de Conteúdo	String	(1,1)	
Estrutura de Organização de	Organização de Conteúdo	String	(1,1)	

Conteúdo				
Arquivo do Recurso	Recurso	String	(0,1)	
Tipo de Recurso	Recurso	String	(0,1)	
Asset	Recurso		(0,1)	
Descrição da Anotação	Anotação	String	(0,1)	
Data de Anotação	Anotação	String	(0,1)	
Rua	Contato	String	(0,1)	
Cidade	Contato	String	(0,1)	
Fax	Contato	String	(0:*)	
Fone	Contato	String	(0:*)	
CEP	Contato	String	(0,1)	
Estado	Contato	String	(0,1)	
Papel para Objeto de Aprendizagem	Contribuição para o Objeto de Aprendizagem	String	(1,1)	
Tipo de Interatividade	Educacional	String	(0,1)	
Contexto	Educacional	String	(0,10)	
Alcance típico de idade	Educacional	String	(0,5)	
Tempo típico de aprendizagem	Educacional	String	(0,1)	
Nível de Interatividade	Educacional	String	(0,1)	
Dificuldade	Educacional	String	(0,1)	
Densidade Semântica	Educacional	String	(0,1)	
Tipo de Recurso de Aprendizagem	Educacional	String	(0,10)	
Papel pretendido ao usuário final	Educacional	String	(0,10)	
Descrição	Educacional	String	(0,10)	
Cobertura do LO	Geral	String	(0,10)	
Catálogo do LO	Geral	String	(0,1)	
Descrição do LO	Geral	String	(0,1)	
Nível de Agregação	Geral	String	(0,1)	
Entrada do LO	Geral	String	(0,1)	
Estrutura do LO	Geral	String	(0,1)	
Identificador do LO	Geral	Inteiro	(0,10)	
Título do LO	Geral	String	(0,1)	
Língua do LO	Geral	String	(0,10)	

ID das palavras-chave	Palavras-chave	Inteiro	(1,1)	
Categoria das Palavras-chave	Palavras-chave	String	(0,3)	
Palavra das Palavras-chave	Palavras-chave	String	(0,1)	
Versão do LO	Ciclo de Vida	String	(0,1)	
Status do LO	Ciclo de Vida	String	(0,1)	
Entrada do Metadado	Meta-metadados	String	(0,1)	
Identificador do Metadado	Meta-metadados	Inteiro	(0,10)	
Linguagem do Metadado	Meta-metadados	String	(0,1)	
Esquema do Metadado	Meta-metadados	String	(0,10)	
Catálogo do Metadado	Meta-metadados	String	(0,1)	
Tipo de Relação	Relação	String	(0,1)	
Direitos Autorais ou outras restrições	Direitos	Booleano	(0,1)	
Descrição do Direito	Direitos	Booleano	(0,1)	
Formato Técnico	Técnico	String	(0,40)	
Localização Técnica	Técnico	String	(0,10)	
Outras Plataformas requeridas	Técnico	String	(0,1)	
Duração Técnica	Técnico	String	(0,1)	

Tamanho Técnico	Técnico	String	(0,1)	
Observações de Instalação	Técnico	String	(0,1)	
Nome do usuário	Usuário	String	(0,1)	
Email do Usuário	Usuário	String	(1,1)	
Login do Usuário	Usuário	String	(1,1)	
Senha do Usuário	Usuário	String	(1,1)	
Email da Entidade	Entidade	String	(0,1)	
Identificador da Entidade	Entidade	Inteiro	(1,1)	
Nome da Entidade	Entidade	String	(1,1)	
Web Site da Entidade	Entidade	String	(0,1)	
Propósito de Classificação	Classificação	String	(0,1)	