

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

Régis Alessandro Glonvezynski

**MODELO DE ANOTAÇÃO DE DOCUMENTOS
PARA A CODIFICAÇÃO DO CONTEÚDO
SEMÂNTICO NO PROCESSO DE AUTORIA**

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação

Orientador: Prof. Dr. Fernando Ostuni Gauthier

Florianópolis, agosto de 2005.

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

MODELO DE ANOTAÇÃO DE DOCUMENTOS PARA A CODIFICAÇÃO DO CONTEÚDO SEMÂNTICO NO PROCESSO DE AUTORIA

Régis Alessandro Glonvezynski

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

Prof. Raul Sidnei Wazlawick, Dr.

Banca Examinadora

Prof. Fernando Ostuni Gauthier, Dr. (orientador)

Prof. Nilson Ribeiro Modro, Dr.

Prof. Renato de Mello, Ph.D.

Prof. Rogério Cid Bastos, Dr.

Ofereço esta dissertação aos meus pais.

Agradecimentos

A Deus, sempre presente em minha vida.

Aos meus pais, Mario e Maria Regina e ao meu irmão Junior, pelo carinho e apoio nas horas mais difíceis. Vocês são a razão do meu esforço.

Ao meu mentor espiritual Heitor por me acompanhar em todos os momentos de minha vida.

Aos amigos Adamo, Renato, Rafael e Leonardo pelo apoio e consultoria.

Ao professor Gauthier, pelos ensinamentos e pela dedicação, paciência e compreensão que demonstrou ao decorrer deste tempo de pesquisa.

Às grandes amizades conquistadas em Florianópolis, que espero levar para o resto de minha vida. Ao Zago, Carol, Ana, André, Fabian, Beto, Barreto, Neves, Cássia, Ivana, Ana Beatriz, Camila, Athos, Adriano, Michel, Marcelle, Fabiana e Lilia.

Aos meus velhos amigos, Bertei, Ricardo, Cassiano, Kiko, Marco, Rodrigo e Audrei.

A todos aqueles que, direta ou indiretamente, participaram da realização deste trabalho.

SUMÁRIO

LISTA DE FIGURAS.....	viii
LISTA DE SIGLAS	xi
RESUMO.....	12
ABSTRACT	13
1 Introdução	14
1.1 Motivação	15
1.2 Objetivos.....	15
1.2.1 Objetivo geral	15
1.2.2 Objetivos específicos	15
1.3 Metodologia empregada	16
1.4 Organização do texto	16
2 Web Semântica	18
2.1 Arquitetura da <i>Web Semântica</i>	19
2.2 A camada Unicode e URI.....	20
2.3 A camada XML	20
2.3.1 Origem da Linguagem XML	21
2.3.2 Conceitos básicos	21
2.3.3 Linguagens para esquemas	24
2.4 A camada RDF	26
2.5 A camada de ontologias.....	28
2.5.1 Critérios para construção de ontologias.....	29
2.5.2 Tipos de ontologias.....	30
2.5.3 Profundidade ontológica.....	31
2.5.4 Objetos de uma ontologia	31
2.5.5 Etapas na construção de uma ontologia.....	32
2.5.6 Linguagens para representação de ontologias	33
2.6 As camadas de Lógica, Prova e Confiança.....	33
2.7 A linguagem OWL	34
2.7.1 As três sub-linguagens de OWL.....	34
2.7.2 Contexto e evolução	35

2.7.3 Estrutura da linguagem OWL.....	36
2.7.3.1 Namespaces	36
2.7.3.2 <i>Headers</i>	37
2.7.3.3 Classes	38
2.7.3.4 Indivíduos	39
2.7.3.5 Propriedades	39
2.8 Considerações Finais	40
3. Anotação semântica de documentos	41
3.1 Considerações a respeito de anotação semântica de documentos	41
3.2 Ferramentas de Anotação Semântica de Documentos.....	43
3.2.1 <i>OntoMat Annotizer</i>	43
3.2.2 <i>SemanticWord</i>	44
3.2.3 <i>Annotea</i>	45
3.2.4 O trabalho proposto frente aos trabalhos correlatos	46
3.3 Considerações Finais	47
4 Modelo para anotação semântica de documentos	49
4.2 Arquitetura do modelo de anotação.....	49
4.2.1 Interface com o Usuário	51
4.2.2 Módulo de Anotação	52
4.2.3 Módulo de Manipulação de Ontologias	53
4.2.4 Módulo de Interface com Pacote de Aplicativos para Escritório	53
4.2.5 Repositório de Ontologias	54
4.2.6 Repositório de Documentos	54
4.3 Considerações Finais	55
5 Ferramenta implementada	56
5.1 Tecnologias empregadas.....	56
5.1.1 A Linguagem Java.....	56
5.1.2 API para manipulação de ontologias.....	57
5.1.2.1 API Jena.....	58
5.1.3 API de acesso ao OO	64
5.1.3.1 API Java UNO	65
5.1.4 OpenOffice.org (OO).....	67

5.2 A ferramenta.....	68
5.3 Testes de validação.....	71
5.3.1 Serviço de Validação do W3C	72
5.3.2 Aplicação de testes na ferramenta implementada.....	73
5.4 Considerações Finais	73
6 Conclusões e trabalhos futuros.....	74
6.1 Conclusões.....	74
6.2 Trabalhos futuros.....	75
BIBLIOGRAFIA	77

LISTA DE FIGURAS

Figura 1: Camadas da <i>Web Semântica</i> [Koivunen & Miller 2001].	19
Figura 2: Estrutura da linguagem XML.	22
Figura 3: Exemplo de documento XML contendo dados de uma pessoa.	22
Figura 4: Exemplo de código XML	23
Figura 5: Exemplo de documento XML.	24
Figura 6: Exemplo de estrutura em árvore.	24
Figura 7: Exemplo de definição de uma DTD para um documento XML.	25
Figura 8: Exemplo simples de um Modelo Básico RDF.	27
Figura 9: Exemplo de representação de um modelo RDF.	27
Figura 10: Evolução da Linguagem OWL [Hendler 2005].	36
Figura 11: Declaração de namespaces OWL [Smith et al 2004].	37
Figura 12: Exemplo de uma declaração de Headers.	37
Figura 13: Exemplo de declaração de classes.	38
Figura 14: Exemplo de declaração de subclasses OWL.	39
Figura 15: Exemplo de declaração de um Indivíduo.	39
Figura 16: Exemplo de declaração de um Indivíduo.	39
Figura 17: Ferramenta <i>Ontomat Annotizer</i> .	44
Figura 18: Barra de Ferramenta e Menus do <i>SemanticWord</i> .	44
Figura 19: Exemplo de anotação de um parágrafo utilizando o <i>browser</i> e editor <i>Amaya</i> com o serviço de anotações <i>Annotea</i> .	46
Figura 20 - Modelo de Anotação Semântica.	51
Figura 21 - Hierarquia e relacionamentos das Interfaces da API Jena [Verzulli 2001].	59
Figura 22 - Interação entre o modelo de uma ontologia e o grafo RDF [Carroll et al 2004].	60
Figura 23 - Hierarquia dos modelos Jena.	61
Figura 24 - Elemento de configuração para acesso remoto ao <i>OpenOffice</i> .	66
Figura 25 - <i>Service Manager</i> .	66
Figura 26 - Fluxo da anotação semântica.	69
Figura 27 - Ferramenta de Anotação implementada.	69
Figura 28 - Documento OO com a anotação semântica.	71

Figura 29 - RDF Validation Service [Validator 2005]..... 72

LISTA DE TABELAS

Tabela 1 - Profiles das linguagens de representação de ontologias.....	63
--	----

LISTA DE SIGLAS

API	<i>Application Programming Interface</i>
CASE	<i>Computer Aided Software Engineering</i>
CGI	<i>Common Gateway Interface</i>
DAML	<i>DARPA Agent Markup Language</i>
DTD	<i>Document Type Definition</i>
GML	<i>General Markup Language</i>
HTML	<i>Hypertext Markup Language</i>
IDE	<i>Integrated Development Environment</i>
JVM	<i>Java Virtual Machine</i>
LGPL	<i>Lesser General Public License</i>
NS	<i>NameSpaces</i>
OIL	<i>Ontology Inference Layer</i>
OO	<i>OpenOffice.org</i>
OWL	<i>Web Ontology Language</i>
RDF	<i>Resource Description Framework</i>
SDK	<i>Java Development Kit</i>
SHOE	<i>Simple HTML Ontology Extensions</i>
SiRPAC	<i>Simple RDF Parser & Compiler</i>
SISSL	<i>Sun Industry Standards Source License</i>
UNO	<i>Universal Network Objects</i>
UNICODE	<i>Universal Character Encoding</i>
URI	<i>Uniform Resource Indicator</i>
W3C	<i>World Wide Web Consortium</i>
WWW	<i>World Wide Web</i>
XML	<i>eXtensible Markup Language</i>
XOL	<i>XML-based Ontology Exchange Language</i>

RESUMO

Com o contínuo crescimento da *Web*, a grande quantidade de informação disponível nos documentos e a forma desorganizada em que estes estão dispostos, a tarefa de recuperação de informação na *Web* é dificultada, ocasionando problemas de relevância nos resultados. Surge, assim, a necessidade de organizar e estruturar o armazenamento das informações, visando facilitar a sua recuperação na *Web* pelos agentes de softwares especializados.

Neste sentido, a *Web Semântica* surge como uma extensão à *Web* atual, proporcionando mecanismos que possam auxiliar na resolução de tais problemas. Assim, é possível dotar os documentos disponíveis na *Web* com uma estrutura semântica, tendo seus conteúdos baseados em ontologias, agregando uma nova camada com informações a respeito da semântica dos documentos.

Nesse trabalho, é apresentado um modelo capaz de prover um suporte à criação e manutenção de documentos dotados de uma estrutura semântica, a serem disponibilizados na *Web*, visando facilitar a recuperação das informações. O modelo é desenvolvido baseado em pacotes de aplicativos para escritório, que são, normalmente, responsáveis por parte significativa da criação de documentos em uma corporação.

Palavras-chave: Web Semântica; Ontologia; Anotação Semântica.

ABSTRACT

As the Web continuously grows, large amount of information is made available in a disorganized manner. As a consequence, information retrieval in the Web is difficult and error prone. Therefore, organizing and structuring information in the Web in order to facilitate the information retrieval through specialized software agents becomes an important issue.

Towards this objective, the Semantic Web emerges as a Web extension to provide mechanisms able to solve these problems. This extension will supply documents available in the Web with semantical structure, with ontologies based content and adding a new information layer according to the document semantics.

In this work, we propose a model that provides support to the creation and maintenance of documents with a semantical structure that will be made available in the Web, aiming to facilitate the information recovery. The model is developed based in office automation packages responsible for significant amount of document creation in a corporation.

Keywords: Semantic Web; Ontology; Semantic Annotation.

1 Introdução

Com o contínuo crescimento da *Web* e o aumento no número de documentos espalhados de forma desordenada, surge a necessidade de novos serviços, cada vez mais especializados para a recuperação destes documentos pelas ferramentas de busca.

Um dos problemas da *Web* hoje, é a recuperação da informação contida nela. Para Berners-Lee [Berners-Lee et al 2001], isso acontece porque, normalmente, essa informação não está estruturada nem segue nenhum padrão, sendo que a única preocupação está em sua estética, ou seja, que essa informação esteja organizada em páginas feitas para humanos entenderem.

Nos dias atuais, as ferramentas de gerenciamento dos documentos eletrônicos tornam-se um elemento importantíssimo para que seja possível realizar a manipulação e recuperação de grandes quantidades de informações. Mesmo as melhores ferramentas atuais de busca na *Web* não são completamente capazes de identificar quais páginas satisfazem a consulta feita pelo usuário, trazendo junto com bons resultados, uma grande quantidade de páginas sem nenhuma relevância para o resultado esperado.

Segundo Berners-Lee [Berners-Lee et al 2001], a *Web Semântica* surge como uma extensão à *Web* atual, que tem por objetivo estabelecer um modelo que permita estruturar as informações nela contidas, dando a estas um significado bem definido e permitindo, desta forma, uma melhor interação entre computadores e pessoas. Assim, é possível adicionar semântica ao conteúdo dos documentos, de maneira a possibilitar que a recuperação dos mesmos garanta a sua relevância dentro do domínio de conhecimento envolvido.

Nas organizações, grande parte dos documentos é criada e manipulada através da utilização de pacotes de aplicativos para escritório, que, normalmente, representam um significativo custo financeiro devido a questões de licença de utilização. A política de software livre, neste sentido, traz alternativas livres de licença e com funcionalidades semelhante as dos pacotes pagos.

O presente trabalho explora o uso de anotações para documentos baseados em sua semântica, a fim de aumentar a qualidade dos serviços de busca em relação a sua eficácia e robustez. Neste contexto, anotações são entidades próprias, que possuem seus próprios atributos e operações, que dotam os documentos disponíveis na *Web* com uma estrutura semântica baseada na utilização de ontologias.

O modelo proposto baseia-se na utilização de anotações semânticas em documentos oriundos de pacotes de escritório, provendo mecanismos que permitam a criação e a manipulação dos mesmos com conteúdo semântico.

1.1 Motivação

O recente crescimento da quantidade, complexidade e diversidade das informações disponibilizadas na *Web*, têm motivado a comunidade de pesquisadores a buscar alternativas em relação a habilidade de gerenciar e armazenar documentos na *Web*.

Uma das alternativas sendo pesquisada é a de dotar os documentos disponíveis na *Web* de uma semântica, através da utilização da *Web Semântica* e de ontologias. Neste sentido, este trabalho propõe um modelo de anotação de documentos para codificação do conteúdo semântico durante o processo de autoria.

1.2 Objetivos

1.2.1 Objetivo geral

Este trabalho tem por objetivo propor um modelo de anotação de documentos para a codificação do conteúdo semântico durante o seu processo de autoria.

1.2.2 Objetivos específicos

Os resultados esperados neste trabalho são citados a seguir:

- Identificação das técnicas para a utilização da *Web Semântica*, apresentando seus conceitos e definições;
- Apresentação das estratégias utilizadas para anotação semântica de documentos para a *Web*;
- Apresentação de um modelo para anotação semântica de documentos em processo de autoria;
- Implementação de um protótipo envolvendo os conceitos definidos no modelo, utilizando apenas ferramentas *Open Source* a fim de validar o mesmo;
- Realização de testes de anotações utilizando a implementação do modelo.

1.3 Metodologia empregada

Para o desenvolvimento do trabalho, realizou-se, primeiramente, uma revisão bibliográfica sobre os assuntos abordados. São apresentados as técnicas e conceitos envolvidos na utilização da *Web Semântica*, verificando-se os trabalhos correlatos sobre a utilização de anotação semântica em documentos.

A etapa seguinte consistiu na realização de um estudo e a definição de um modelo de anotação semântica de documentos.

A etapa posterior consistiu em um estudo sobre as linguagens de programação e os recursos disponíveis nas mesmas para implementação do protótipo, bem como a implementação do modelo para validar sua utilização.

Finalmente, foi realizada a utilização do protótipo, aplicando-o a testes de anotação, a fim de verificar a aplicabilidade das técnicas envolvidas no modelo proposto, bem como elaborar as conclusões a respeito das técnicas e conceitos utilizados neste trabalho.

1.4 Organização do texto

O presente trabalho está organizado em 6 Capítulos. O primeiro capítulo tem um caráter introdutório, descrevendo os motivos de melhorar a semântica contida nos documentos disponibilizados na *Web* e os objetivos que o modelo proposto pretende alcançar.

No Capítulo 2 está descrita a estrutura da *Web Semântica*, a qual é dividida em camadas, onde podem ser vistos os conceitos que envolvem a linguagem XML e do modelo RDF que servem para estruturar e prover significado aos dados. Neste capítulo também são vistos os conceitos a respeito de ontologias e das camadas de lógica, prova e confiança.

O Capítulo 3 discute os conceitos envolvidos com a anotação semântica de documentos, descreve também trabalhos correlatos sobre anotação semântica de documentos.

No Capítulo 5 é descrito o modelo de anotação semântica de documentos proposto neste trabalho e as técnicas utilizadas em sua construção.

No Capítulo 6 é mostrado o protótipo de uma ferramenta implementada para realizar a realização dos testes para validação do modelo.

E por fim, no Capítulo 7 são apresentadas as conclusões deste trabalho e algumas sugestões para trabalhos futuros.

2 *Web Semântica*

Neste capítulo são apresentados os conceitos sobre a *Web Semântica*, a qual tem o objetivo de melhorar a recuperação da informação em grandes repositórios de documentos como a *Web*. É através dos conceitos e tecnologias envolvidas na estrutura da *Web Semântica* que procura-se aumentar a eficiência das ferramentas de busca e melhorar o processamento automático de documentos através da utilização de linguagens que permitam definir dados e regras para a realização de inferências sobre esses dados.

A *Web Semântica* foi idealizada por Tim Berners – Lee [Berners-Lee et al 2001], que vislumbrou o seu desenvolvimento, propondo que, através da utilização de metadados e ontologias seria possível prover um acesso automatizado sobre a informação com base em sua semântica. Segundo Berners-Lee (2001), a *Web Semântica* tem por objetivo estruturar o conteúdo significativo das páginas *Web*, criando desta forma um ambiente que permita a agentes de software percorrer a *Web*, página por página, com o objetivo de executar uma busca solicitada pelo usuário.

A *Web Semântica* é uma extensão à *Web* atual e tem por objetivo estabelecer um modelo que permita estruturar as informações nela contidas, dando a estas um significado bem definido. Desta maneira, o conteúdo da *Web Semântica* não será processado apenas por humanos, mas também por máquinas. Assim, a tendência é que a *Web* evolua de uma rede desestruturada de documentos para uma rede onde as informações possuam significado e relações bem definidos, além de mecanismos que facilitem o acesso aos mesmos [Berners-Lee et al 2001].

A *Web Semântica* fornece uma estrutura comum, que permite que os dados possam ser compartilhados e reutilizados através de aplicações por empresas e por usuários em geral. Nesta nova visão da *Web*, será possível através de metadados representar associações entre coisas que, em princípio, poderiam não estar relacionadas. Esta evolução elevará o status atual da *Web* de “*machine-readable*” (lida automaticamente) para algo que pode ser descrito como “*machine-understandable*” (entendida automaticamente) [Berners-Lee et al 2001].

A *Web Semântica* só poderá ser atingida a partir do relacionamento automático entre pequenas ontologias independentes e específicas em seus subdomínios, servindo dessa forma como resposta a uma consulta específica. Esse fato talvez justifique o

desenvolvimento e proliferação de tantas linguagens de definição de ontologias e representação de conhecimento em torno da *Web Semântica*. Todavia, à medida que tais linguagens são utilizadas, tornam-se necessários mecanismos de edição e modelagem de ontologias que, a exemplo de ferramentas CASE, possibilitem o uso de ferramentas distintas de modelagem, permitindo a utilização de várias linguagens semânticas na *Web* [Moura 2001].

A padronização da *Web Semântica* é um esforço colaborativo conduzido pelo W3C¹, com participação de um grande número de pesquisadores e empresas².

Na próxima seção, será abordada a estrutura da *Web Semântica*, as tecnologias empregadas aliadas a teoria de ontologias, o que torna possível oferecer um serviço na *Web* com um maior nível de qualidade, na qual será possível criar uma rede extensa de conhecimento humano, onde o processamento via ferramentas de busca na *Web* aumentará a qualidade e a robustez na recuperação da informação.

2.1 Arquitetura da *Web Semântica*

Em 2001, o W3C apresentou a proposta [Koivunen & Miller 2001] de padronizar a estrutura da *Web Semântica* em camadas, conforme Figura 1. As duas primeiras camadas representam os esquemas, descrevem a estrutura dos dados e define o seu significado. As camadas com a identificação de *data* definem as relações entre os dados, também podem ser chamados de camadas de nível ontológico. As camadas superiores definem os mecanismos para fazer inferências sobre os dados.

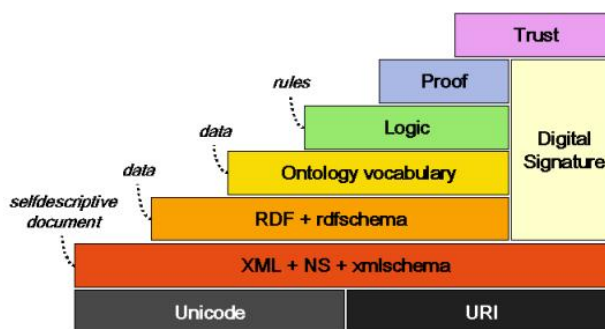


Figura 1: Camadas da *Web Semântica* [Koivunen & Miller 2001].

¹ <http://www.w3c.org/> - World Wide Web Consortium(W3C).

² <http://www.w3c.org/2001/sw/BestPractices/> - Semantic Web Best Practices and Deployment Working Group.

As camadas da estrutura da *Web Semântica* serão descritas detalhadamente nas próximas seções.

2.2 A camada Unicode e URI

Fundamentalmente, os computadores lidam com números. Gravam letras e outros caracteres na memória designando um número para cada um deles. O padrão *Unicode*³ fornece um único número para cada caracter, não importando a plataforma, o software e o idioma. O padrão *Unicode* tem sido adotado por líderes do setor de informática tais como Apple⁴, HP, IBM⁵, Oracle⁶, Microsoft e muitos outros⁷. O *Unicode* é necessário para padrões como XML e Java.

As URIs (*Uniform Resource Indicator*) são *strings* utilizadas para identificar recursos na *Web* com um endereço único, como páginas, serviços, documentos, imagens, músicas, arquivos, entre muitos outros. Há um grande número de formas de endereçamento, e outros mais podem ser ainda incorporados com o tempo. O URL (*Universal Resource Locator*) e o URN (*Universal Resource Name*) são tipos de URIs.

2.3 A camada XML

A linguagem XML (*eXtensible Markup Language*) foi criada especificamente para o uso na Internet, sendo uma recomendação oficial do W3C. Em determinados momentos, assemelha-se com o HTML (*Hipertext Markup Language*), só que enquanto o HTML se preocupa em controlar a forma como os dados serão exibidos, o XML se concentra na descrição dos dados do documento.

O XML é uma linguagem independente de plataforma e de domínio público, de forma que nenhuma empresa pode utilizá-la como linguagem proprietária.

Os *NameSpaces* (*NS*) ou espaços de nomes foram criado para que não haja colisão de nomes de forma segura em um documento XML.

³ <http://www.unicode.org> - Site oficial do padrão Unicode.

⁴ <http://www.apple.com/> - Apple.

⁵ <http://www.ibm.com/> - IBM.

⁶ <http://www.oracle.com/index.html> - Oracle.

⁷ <http://www.unicode.org/consortium/memblogo.html> - Membros do consórcio Unicode.

2.3.1 Origem da Linguagem XML

Em 1969, Lorie e Charles F. Goldfarb, da IBM *Reserch*, criaram a primeira linguagem de marcação moderna, a *General Markup Language* (GML), com o propósito de que esta fosse uma meta-linguagem que pudesse ser usada para descrever outras línguas, suas gramáticas e seus vocabulários. Mais tarde a GML tornou-se uma linguagem padrão, passando a chamar-se *Standard Generalized Markup Language* (SGML), que no ano de 1986 foi adotada como padrão de troca e armazenamento de dados internacional pela ISO (*International Organization for Standardization*), chamada ISO8879 [Martin et al 2001].

No ano de 1996, a linguagem XML começou a ser projetada pelo W3C, com o objetivo de desenvolver uma linguagem que combinaria a flexibilidade do SGML com a ampla aceitação do HTML [Bray et al 2004]. A utilização do SGML como ponto de partida, permitiu que a equipe responsável pelo desenvolvimento do projeto se concentrasse em transformar aquilo que já era aceito como padrão em algo mais simples, pois o SGML já fornecia uma linguagem aberta que poderia ser expandida por qualquer um com qualquer finalidade [Martin et al 2001].

O XML 1.0 transformou-se em recomendação do W3C em fevereiro de 1998 [Martin et al 2001] e sua especificação oficial, gramática e notação pode ser encontrada no *site* da W3C [Bray et al 2004].

2.3.2 Conceitos básicos

Com a finalidade de descrever informações, o XML permite que sejam definidas estruturas personalizadas para os documentos, onde as *tags* do XML dão nome ao conceito que está sendo descrito.

Ao contrário do HTML, que trata ao mesmo tempo dos dados e da apresentação, o XML refere-se somente aos dados. Para a apresentação do documento XML, foi desenvolvida uma forma para representar os dados, denominada folha de estilo. Na folha de estilos são definidos os estilos de formatação e normas que determinam quando certo estilo deve ser aplicado [Martin et al 2001]. A folha de estilos fica armazenada em um outro documento, assim podendo ser usada para definir a forma de apresentação de vários documentos XML.

Para definição das regras para a validação de um documento, no que diz respeito a sua sintaxe, o XML também possui um outro documento chamado DTD. O DTD é uma definição para o tipo do documento, nele esta definida a estrutura e os relacionamentos dos conceitos [Bray et al 2004].

A Figura 2 mostra a estrutura da linguagem XML, onde “XML Dados” representa as informações armazenadas ou o conteúdo do documento. A “DTD ou Esquema XML” representa a estrutura e organização do documento, que pode possuir diversos tipos de formatos. A folha de estilos é a responsável pela apresentação do documento, ou seja, a maneira como é apresentado o conteúdo de um documento XML.

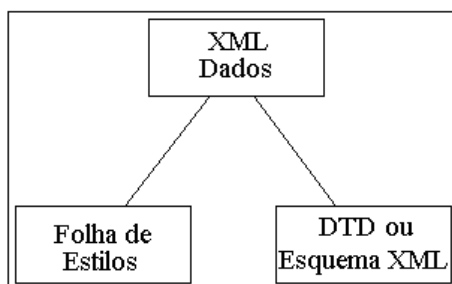


Figura 2: Estrutura da linguagem XML.

O XML é uma abordagem simples, que marca o conteúdo através de *tags*, elas delimitam o conteúdo e a sintaxe do XML, o que nos permite definir estruturas de complexidade arbitrária [Martin et al 2001].

Os dados são auto-descritivos à medida que cada item leva seu próprio nome que pode estar relacionado a um modelo externo para problemas do mundo real que o documento descreve [Martin et al 2001]. Na Figura 3, é descrito o documento XML para uma pessoa em termos de nome e idade.

```
<Pessoa>
  <Nome>
    <Primeiro>Régis</Primeiro>
    <Ultimo>Glonvezynski</Ultimo>
  </Nome>
  <Idade>27</Idade>
</Pessoa>
```

Figura 3: Exemplo de documento XML contendo dados de uma pessoa.

Um documento XML possui vários componentes que são utilizados para representar os dados e os metadados:

- *Elementos*: São blocos de construção básicos de uma marcação XML. Eles são delimitados por uma *tag* inicial, uma *tag* final, que consiste em um nome do tipo

de elemento, o qual é uma string fechada por uma par de parênteses angulares (“<>”), e ainda uma barra inclinada (“/”) na *tag* final antes do nome do tipo de elemento. As informações estão contidas entre as *tags*, que são chamadas de conteúdo do elemento. Um elemento pode também armazenar outros elementos, como pode ser visto na Figura 3, onde o elemento Pessoa possui os sub-elementos Nome e Idade;

- *Atributos*: Em alguns casos pode existir alguma informação sobre um elemento que seja necessário anexar a ele, ao contrário da informação que é contida no elemento [Martin et al 2001]. Este atributo é composto de um par nome = valor, como pode ser visto na Figura 4, onde é atribuído a um livro o seu número de ISBN;
- *Comentários*: São úteis para escrever notas ou comentários em um documento, servem para formar um registro de revisão, histórico ou qualquer outra informação que pode ser útil para o autor do documento ou outra pessoa que venha a revisar o conteúdo do documento. Na Figura 4, podem ser vistos exemplos de comentários em um documento XML, são identificados pelo uso da *tag* <!-- para início do comentário e fechada pelo uso da *tag* -->;
- *Instruções de processamento*: são definidas por *tags* iniciadas em <? e finalizadas por ?>, fornecem as instruções que serão executadas pelas aplicações. Na Figura 4, pode ser visto um exemplo de instrução de processamento na primeira linha, onde é definida a versão do XML;
- *Seções CDATA*: é um método de inclusão de texto que contenham caracteres que de outra forma seria reconhecida como marcação [Martin et al 2001], isto evita que um analisador (*parser*) interprete este conteúdo. Uma seção CDATA é representada pela seguinte sintaxe <![CDATA[...]]>, onde a parte “...” pode ser qualquer string de caractere que não inclui a string “]]>”.

```
<?xml version="1.0" encoding="UTF-8"?>
<livro isbn="85-7393-116-7">           <!-- Número do isbn -->
  <titulo>Professional XML</titulo>  <!-- Titulo do livro -->
  <valor>80.00</valor>              <!-- Valor do livro -->
  <![CDATA [Este texto não vai ser interpretado por um analisador]]>
</livro>
```

Figura 4: Exemplo de código XML .

A estrutura lógica de um documento XML é semelhante a uma árvore, onde seus elementos são estruturados através de uma hierarquia, por exemplo, o documento da Figura 5 pode ser visualizado na estrutura de uma árvore, o que pode ser visto na Figura 6.

```
<?xml version="1.0" encoding="UTF-8"?>
<bibliografia>
  <livro isbn="85-7393-116-7">
    <titulo>Professional XML</titulo>
    <autor>DIDIER MARTIN</autor>
  </livro>
  <livro isbn="0-8493-1031-8">
    <titulo>XML Databases and the Semantic Web</titulo>
    <autor>BHAVANI THURASINGHAM</autor>
  </livro>
</bibliografia>
```

Figura 5: Exemplo de documento XML.

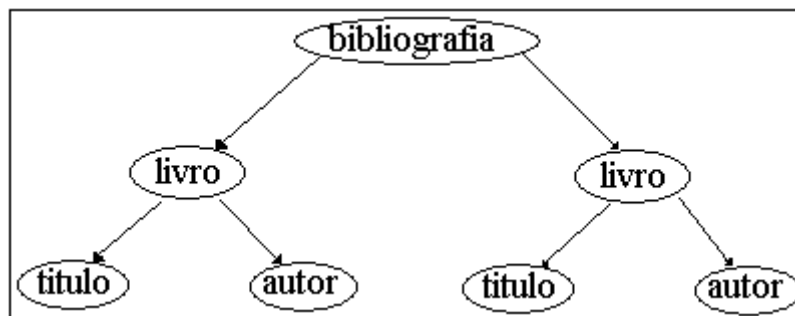


Figura 6: Exemplo de estrutura em árvore.

2.3.3 Linguagens para esquemas

As DTDs são utilizadas para definir a estrutura e a sintaxe de um documento, auxiliando na validação da estrutura do documento XML. O XML *Schema* tem a mesma função, mas possui mais recursos que as DTDs. As principais diferenças entre os dois são apresentadas a seguir.

a) DTD – Document Type Definition

A DTD (Definição de tipo de documento) define a estrutura de um documento XML, a especificação de seus elementos, relacionamentos, entidades, atributos e as *tags*

utilizadas para marcar o documento, servindo como um esquema para os dados representados pelo documento XML.

Basicamente, uma DTD é um conjunto de regras que define as instruções que serão enviadas ao analisador sintático para o documento que está sendo analisado [Tesch 2002].

Não é obrigatório que um documento tenha uma DTD associada a ele, mas caso tenha, deve seguir sua estrutura. Assim, qualquer documento XML deve ser analisado para verificar se o documento segue as regras definidas na DTD, determinando se o mesmo é ou não um documento válido.

A vantagem da utilização da DTD é que o vocabulário fica documentado oficialmente e de modo preciso. Todas as regras do vocabulário estão contidas na DTD, onde qualquer coisa que não aparece ali não é parte do vocabulário, possibilitando que um autor adicione somente os elementos ou atributos permitidos pela DTD, e de acordo com a estrutura ditada pela DTD [Martin et al 2001].

Existem duas maneiras para armazenar uma DTD, externa ou internamente. As DTDs internas combinam todos os elementos, atributos, notações e entidades dentro do próprio documento XML. Elas são colocadas no início do documento, dentro da declaração de tipo de documento (`<!DOCTYPE>`) [Tesch 2002]. No exemplo da Figura 7, é mostrado um documento XML com sua respectiva DTD, note que para este exemplo, é utilizada a declaração da DTD dentro do documento XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE LIVRO [
<IELEMENT LIVRO (TITULO, AUTOR, EDITORA)>
<IELEMENT TITULO (#PCDATA)>
<IELEMENT AUTOR (#PCDATA)>
<IELEMENT EDITORA (#PCDATA)>
]>
<LIVRO ISBN="85-7393-116-7">
  <TITULO>PROFESSIONAL XML</TITULO>
  <AUTOR>DIDIER MARTIN</AUTOR>
  <EDITORA>EDITORA CIÊNCIA MODERNA </EDITORA>
</LIVRO>
```

Figura 7: Exemplo de definição de uma DTD para um documento XML.

Segundo Castro [Castro 2001], a DTD é limitada para a construção de esquemas XML. Primeiro, porque as DTDs são escritas em uma sintaxe que possui pouca relação com XML e que um *parser* XML não consegue analisá-la. Em segundo, as declarações

são globais, ou seja, não é possível definir dois elementos diferentes com o mesmo nome, mesmo se aparecem em contextos separados. E em terceiro lugar, não é possível declarar para cada elemento os tipos de dados, como por exemplo, um inteiro, um real, uma data e assim por diante, pois o indicador *#PCDATA* apenas informa que o conteúdo do elemento é caractere.

b) XML Schema

O W3C propôs o *XML Schema* para servir como linguagem de definição de tipos de documentos XML [Henry et al 2004]. O nome “esquema” origina-se da área de banco de dados, onde é utilizado para definir a estrutura das tabelas que compõem uma aplicação.

Um *XML Schema* consiste de componentes para definição de tipos e declaração de elementos. Eles podem ser usados para validar a definição dos elementos e atributos em um documento XML [Henry et al 2004].

O *XML Schema* define como um determinado conjunto de documentos XML devem ser construídos, definindo a ordem dos elementos, bem como que atributos eles podem conter. O *XML Schema* consegue suprir algumas das deficiências citadas das DTDs, o que torna seu uso mais vantajoso do que a utilização das DTDs [Castro 2001]. Os documentos *XML Schema* são construídos através de uma sintaxe XML, dessa forma eles podem ser analisados por *parsers XML*.

2.4 A camada RDF

O RDF (*Resource Description Framework*, ou Modelo de Descrição de Recursos) é um *framework* recomendado pelo W3C para representação ou definição de metadados e informações na *Web* [Klyne & Carroll 2004], promovendo interoperabilidade entre aplicações, que trocam informações compreensíveis por máquina, na *Web*.

Com o uso do RDF é possível fazer referências a objetos com URIs e definir vocabulários para serem referenciados por URIs. Assim, o RDF descreve “recursos” da *Web* [Koivunen & Miller 2001].

O modelo RDF utiliza a notação XML como sintaxe de codificação para descrição dos metadados. A utilização do XML para a construção dos modelos permite que as descrições sejam analisadas sintaticamente e interpretadas por aplicativos capazes de processar documentos XML.

O objetivo do modelo RDF é facilitar o intercâmbio de informações compreensíveis por máquinas em aplicações para a *Web* [Pitts-Moultis & Kirk 1998].

O modelo básico RDF consiste de três tipos de objetos [Frank & Miller 2004]:

- *Recurso (Resources)*: pode ser uma página *Web*, uma parte dela ou ainda uma coleção de páginas. Os recursos são chamados também de URIs;
- *Propriedade (Properties)*: são características específicas, atributos ou relações para descrever recursos;
- *Declarações (Statements)*: um recurso, sua propriedade, e o valor daquela propriedade é uma declaração RDF. Em declarações RDF um recurso tem uma ou mais propriedades. Cada propriedade possui um tipo e um valor, onde o tipo é um nome e o valor pode ser um literal, uma string ou pode ser outro recurso. Estas três partes são chamadas respectivamente de Sujeito, Predicado e Objeto. Na Figura 8 pode-se visualizar o relacionamento entre recursos em um modelo RDF.

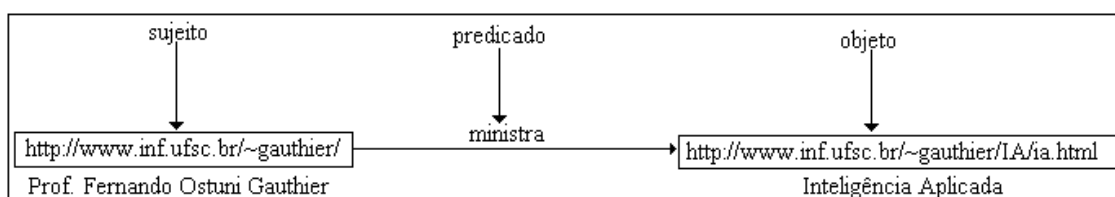


Figura 8: Exemplo simples de um Modelo Básico RDF.

Na Figura 8, é visto os conceitos de recurso (sujeito), propriedade (predicado) e valor (objeto), sendo o sujeito (URL de um professor), o predicado com a propriedade (ministra) e o objeto (URL de uma disciplina). A união destes conceitos também é conhecida como sendo uma tripla RDF (sujeito, predicado e objeto).

A Figura 9 representa o código da representação do modelo RDF para o exemplo da Figura 8.

```
<rdf:Description about= http://www.inf.ufsc.br/~gauthier/ >  
  <ministra>  
    <rdf:Description about= http://www.inf.ufsc.br/~gauthier/IA/ia.html>  
  </rdf:Description>  
</ministra>  
</rdf:Description>
```

Figura 9: Exemplo de representação de um modelo RDF.

A unicidade da relação entre um conceito e um URI garante a unicidade do significado das triplas, onde cada conceito tem uma definição única e as triplas formam uma relação na qual pode-se usar regras de inferências para fazer deduções.

2.5 A camada de ontologias

O termo Ontologia vem do grego (*ontos*, "ser", "ente"; e *logos*, "saber", "doutrina"), que significa o "estudo do ser" e, desse modo, pode equivaler à metafísica onde a Ontologia é um ramo da metafísica que trata do *ser*, incluindo teorias da natureza e tipos de seres.

Segundo o dicionário Aurélio da Língua Portuguesa o termo Ontologia significa: *"Parte da filosofia que trata do ser enquanto ser, i. e., do ser concebido como tendo uma natureza comum que é inerente a todos e a cada um dos seres"*.

Os conceitos para o uso de ontologias na Ciência da Computação têm origem nos princípios gerais da Filosofia e da Linguística, onde se parte da premissa de que a formalização do conhecimento começa a partir de uma conceitualização, que consiste em um conjunto de entidades sobre o qual o conhecimento é expresso, e a relação entre eles. Esta conceitualização é chamada Ontologia.

Conteras [Conteras et al 2004] caracterizou o conceito de ontologias para aplicação em informática, ele afirma que *"uma ontologia é uma especificação explícita e formal de uma conceitualização compartilhada"*. Esclarecendo melhor esta definição, podemos dizer que:

- Por especificação explícita, são as definições de conceitos, instâncias, relações, restrições e axiomas.
- Por formal, o fato de ela deve ser declarativamente definida, para que possa ser compreendida por máquina, o que exclui o uso da linguagem natural.
- Por conceitualização, entende-se que se trata de um modelo abstrato de uma área de conhecimento ou de um universo limitado de discurso.
- Por compartilhada, reflete a noção de que uma ontologia captura o conhecimento consensual, seja uma terminologia comum da área modelada, ou aceita por um grupo.

Horridge [Horridge 2004] define que ontologias são usadas para capturar conhecimento sobre um domínio de interesse, descrevendo os conceitos sobre um determinado domínio e suas relações entre conceitos.

Segundo Gruber [Gruber 1993], uma ontologia é uma especificação de um conceitualização. Ontologias estabelecem uma terminologia comum entre membros de uma comunidade de interesse. Estes membros podem ser agentes humanos ou automatizados.

A crescente quantidade de informações disponíveis na *Web* vem criando a necessidade de melhora nos sistemas e ferramentas de busca e tratamento de informação. A utilização de ontologias para auxiliar nestas tarefas tende a proporcionar uma grande interoperabilidade, flexibilidade e agilidade, desde que haja o perfeito entendimento entre os conceitos e vocabulários comuns entre as partes interessadas. É neste ponto que se insere a necessidade dos estudos sobre especificações consistentes de ontologias.

A integração da informação é a principal área de aplicação para ontologias. Também é sabido que se dois sistemas adotarem o mesmo vocabulário não há nenhuma garantia que podem concordar com alguma informação a menos que eles utilizem as mesmas conceitualizações. Supondo que cada sistema tem sua própria conceitualização, uma condição é necessária a fim de fazer um acordo possível para que os modelos pretendidos tenham suas conceitualizações originais sobrepostas [Guarino 1998].

2.5.1 Critérios para construção de ontologias

Para criar uma ontologia, é necessário seguir alguns critérios objetivos, com a finalidade de garantir sua consistência, bem como que represente o mais fielmente o problema modelado. Para tal, é necessário seguir alguns princípios básicos para melhor compartilhar sua conceitualização entre os sistemas envolvidos. Estes princípios para a construção de ontologias são descritos a seguir [Pérez & Benjamins 1999]:

- *Clareza e objetividade*: significa que a ontologia deve ser definida de forma clara e objetiva e também fornecer uma documentação em linguagem natural;
- *Completeza*: significa que uma definição deve expressar as condições necessárias e suficientes para expressar um termo;

- *Coerência*: para permitir inferências que sejam consistentes com as definições declaradas;
- *Extensibilidade monôtonica*: Significa que esses novos termos gerais ou especializados devem ser incluídos na ontologia de modo que não requeira uma revisão de definições existentes;
- *Mínimos Compromissos ontológicos*: significa fazer que sejam definidas poucas suposições quanto possíveis sobre o mundo que está sendo modelado, permitindo que especializações e instanciações sejam feitas com liberdade na ontologia;
- *O princípio da distinção ontológica*: significa que as classes em uma ontologia devem ser disjuntas, sem superposição de conceitos;
- *Modularidade*: para minimizar o acoplamento entre os módulos;
- *Minimização da distância semântica entre os conceitos similares*: que significa agrupar e representar os conceitos similares usando as mesmas primitivas;
- Padronização dos nomes sempre que for possível.

2.5.2 Tipos de ontologias

Ontologias podem ser descritas em diferentes níveis de abstração. Dependendo de sua aplicabilidade e com base em seu conteúdo as ontologias podem ser classificadas nas categorias que se seguem [Guarino 1998]:

- *Ontologias genéricas*: As ontologias genéricas, ou também conhecidas como de alto nível, descrevem conceitos muito gerais como o espaço, tempo, matéria, objetos, eventos, ações, etc., que são independentes de um problema particular ou domínio, parece conseqüentemente razoável, ao menos na teoria, ter ontologias genéricas unificadas para comunidades de grandes usuários de algum domínio em particular [Guarino 1998];
- *Ontologias de domínio*: Estas ontologias descrevem um vocabulário para um domínio genérico ou para serem utilizados em um micromundo, como por exemplo, na medicina ou em automóveis [Guarino 1998];
- *Ontologias de tarefas*: Descrevem conceitos sobre a resolução de problemas ou tarefas independentes do domínio em que ocorram [Guarino 1998];

- *Ontologias de aplicação:* As ontologias de aplicação descrevem conceitos dependendo de um domínio e de uma tarefa em particular, e são frequentemente especializações de ambas as ontologias relacionadas. Estes conceitos correspondem frequentemente aos papéis desempenhados por entidades do domínio, quando da execução de alguma atividade [Guarino 1998];
- *Ontologias de representação:* as ontologias de representação do conhecimento capturam as primitivas, usadas na formalização do conhecimento e expressam os conhecimentos ontológicos em formalismos de representação do conhecimento [Pérez & Benjamins 1999].

2.5.3 Profundidade ontológica

Segundo Guarino e Welty [Guarino & Welty 1998], a profundidade ontológica pode ser definida em quatro níveis:

- *Vocabulário:* em sua definição mais simples, uma ontologia é um vocabulário, onde uma DTD ou um XML *Schema* pode ser usado para definir a ontologia;
- *Taxonomia:* a conceitualização dos termos é feita pela definição dos relacionamentos entre os objetos e classes, subclasses e classes-pai. Estes tipos de ontologia são usados normalmente para estabelecer sistemas orientados a objetos;
- *Sistema relacional:* pode também ser definido pela inclusão de relacionamentos não hierárquicos, como é feito com diagramas de relacionamento de entidades e nos bancos de dados relacionais;
- *Teoria axiomática:* além dos relacionamentos, podem ser inclusas restrições, estas restrições podem ser definidas como axiomas. Um axioma é uma afirmação lógica que não pode ser provada a partir de outras afirmações, mas podendo derivar outras afirmações.

2.5.4 Objetos de uma ontologia

Para se modelar ou construir uma ontologia é necessário a utilização dos seguintes objetos [Chandrasekaran et al 1999]:

- *Entidades*: Descrevem classes ou conceitos de um determinado domínio a ser modelado;
- *Atributos*: Item de informação que descreve as propriedades das entidades;
- *Relações*: Pode ser definido como as ligações ou associações entre os objetos no modelo (entidades e atributos);
- *Restrições*: São condições que podem ser impostas ao modelo, restringir as entidades, atributos ou relações;

2.5.5 Etapas na construção de uma ontologia

As atividades principais no processo de construção de uma ontologia são descritas abaixo [Guizzardi & Duarte 2002]:

- *Identificação de Propósito e Especificação de Requisitos*: a primeira etapa na construção de uma ontologia é identificar claramente o seu propósito e quais usos são esperados dela, ou seja, qual será a sua competência, delimitando o que é relevante ou não para a mesma, incluindo seus potenciais usuários e os cenários que motivaram a sua construção;
- *Captura da Ontologia*: deve-se capturar a conceitualização do universo do discurso, com base em sua competência, as entidades relevantes do domínio (por exemplo, conceitos, relações, propriedades e papéis) devem ser identificadas e organizadas. A utilização de uma linguagem gráfica pode ajudar na comunicação com os especialistas de domínio;
- *Formalização da Ontologia*: para esta etapa, é necessária a escolha de um formalismo, para representar as diversas categorias de conhecimento da ontologia. Esta linguagem deve ser capaz de representar de uma maneira precisa e sem ambigüidades os elementos que modelam as entidades existentes no domínio em questão;
- *Integração com ontologias existentes*: durante a fase de captura e/ou formalização da ontologia, pode ser necessária sua integração com outras ontologias já existentes, isto torna-se uma boa prática ao desenvolver ontologias funcionais modulares, que sejam gerais, reusáveis, e quando necessário, integrá-las, obtendo o resultado esperado;

- *Avaliação*: a ontologia deve ser avaliada para verificar se ela satisfaz as exigências da especificação. Os principais critérios na avaliação devem ser: sua clareza, coerência, extensibilidade e seus compromissos ontológicos mínimos;
- *Documentação*: todo o processo de desenvolvimento da ontologia deve ser documentado, incluindo suas finalidades, requisitos e cenários de motivação e a ontologia formal e os critérios de projeto adotados.

2.5.6 Linguagens para representação de ontologias

Para o uso de ontologias na *Web*, é necessário que haja linguagens para representá-las, tornando possível viabilizar o intercâmbio e o processamento semântico através de máquina.

Muitas linguagens foram criadas nos últimos anos para representar ontologias em RDF, arquitetura padrão recomendada pela W3C para interoperabilidade de informações na *Web*, dentre as quais pode-se citar: SHOE [Luke et al 1997], OIL [Fensel et al 2001], DAML [DAML 2004], OWL [Bechhofer et al 2004].

O W3C recomenda a linguagem OWL (*Web Ontology Language* ou linguagem de ontologias para a *Web*) como a linguagem padrão para o uso de ontologias na *Web* Semântica, a qual deriva de uma união entre outras duas linguagens, a europeia OIL (*Ontology Inference Layer* ou camada de inferência para ontologias) e a DAML (*DARPA Agent Markup Language* ou linguagem de anotação para agentes do Departamento de Defesa dos Estados Unidos).

Assim, neste trabalho foi adotada a linguagem OWL como sendo o padrão para representação de ontologias a serem utilizadas no modelo.

2.6 As camadas de Lógica, Prova e Confiança

A camada lógica permite a construção de regras que atuam sobre instâncias e recursos, enquanto a camada de prova as executa, e, a de confiança, analisa se a prova está certa ou não [Koivunen & Miller 2001].

Nestas camadas superiores são definidos os mecanismos que executam inferências sobre os dados. Para que estas camadas funcionem corretamente, é necessário que as camadas inferiores estejam muito bem estruturadas. Quanto a esta estruturação, não é interessante o uso de ontologias com regras, pois pode restringir sua aplicabilidade.

Porém o uso de regras pode ser interessante no aspecto de restringir o uso de atributos e exprimir axiomas.

2.7 A linguagem OWL

Nesta seção apresenta-se uma descrição da linguagem *Web Ontology Language* (OWL), desenvolvida pelo W3C como uma especificação para representação de ontologias. A linguagem OWL foi desenvolvida para o uso em aplicações que precisam processar o conteúdo de informações, ao invés de simplesmente, apresentá-las para humanos. Ela permite uma melhor interpretação de conteúdo na *Web* do que a suportada por XML, RDF e RDFS, pois pode representar vocabulários e os relacionamentos existentes entre entidades desses vocabulários. A linguagem OWL ultrapassa essas linguagens no que se refere a habilidade para representar conteúdo compreensível por máquinas na *Web* [Smith et al 2004].

2.7.1 As três sub-linguagens de OWL

A linguagem OWL de acordo com sua expressividade é subdividida em três espécies ou sub-linguagens [Horridge 2004]:

- *OWL-Lite*: é a sub-linguagem sintaticamente mais simples. Deve ser usada nas situações onde apenas uma hierarquia simples de classes é permitida, onde a cardinalidade máxima ou mínima entre as classes assume apenas valores de 0 ou 1.
- *OWL-DL*: é muito mais expressiva do que o *OWL-Lite*, sendo indicada para os usuários que desejam maior expressividade sem perder a integridade computacional, ela é baseada em lógicas de descrição (*Description Logics*, daqui o sufixo DL). As classes podem ser construídas por união, interseção, e complemento, também pela enumeração de instâncias e podem ter disjunções. É também possível computar automaticamente a classificação de hierarquias e verificá-las para ver se há inconsistências em uma ontologia.
- *OWL-Full*: é a sub-linguagem mais expressiva da OWL, é indicada para usuários que desejam o máximo da expressividade e a liberdade sintática do

RDF. Fazendo este uso mais complexo, permite uma maior liberdade ao usuário, mas não garante a sua computabilidade.

Cada uma destas três sub-linguagens é uma extensão de seu antecessor mais simples, portanto, podemos expressar as seguintes relações entre elas [McGuinness & Harmelen 2002]:

- Toda ontologia OWL *Lite* válida é uma ontologia OWL DL válida;
- Toda ontologia OWL DL válida é uma ontologia OWL *Full* válida;
- Toda conclusão OWL *Lite* válida é uma conclusão OWL DL válida;
- Toda conclusão OWL DL válida é uma conclusão OWL *Full* válida.

OWL *Full* pode ser vista como uma extensão da RDF, enquanto OWL *Lite* e OWL DL são extensões de uma visão restrita de RDF. Conseqüentemente, todo documento OWL é um documento RDF e todo documento RDF é um documento OWL *Full*. Mas somente alguns documentos RDF serão válidos em OWL *Lite* ou OWL DL.

2.7.2 Contexto e evolução

O OWL surgiu da necessidade de melhoria nas linguagens para representação de ontologias, pois o RDF não tinha todo o poder de expressividade necessário para a modelagem de ontologias. Por exemplo, RDF(S) não possui um mecanismo de restrição de cardinalidade.

Para tentar resolver os problemas que o RDF apresentava, foi criada em 1997 a linguagem OIL, que permite definições baseadas em *frames* e baseada também em definições de lógicas de descrições. A linguagem OIL também possui um motor de inferência capaz de manipular tanto *frames* quanto lógicas de descrições, onde as ontologias eram geradas em cima de XML e RDF.

No ano de 1999, o Departamento de Defesa dos Estados Unidos criou o DAML, que também se baseia na linguagem RDF, com o objetivo de melhorar as relações semânticas das ontologias, para que pudessem ser lidas por máquina e que também se mantivesse em conformidade com as tecnologias da Internet sem, no entanto, prover um motor de inferência.

Em 2001, as duas linguagens foram unidas, formando a linguagem DAML + OIL, que em 2002 foi acrescida de requisitos de internacionalização (*Unicode*), apresentação e documentação, originando assim a linguagem OWL. Na Figura 10, podemos visualizar a evolução do OWL.

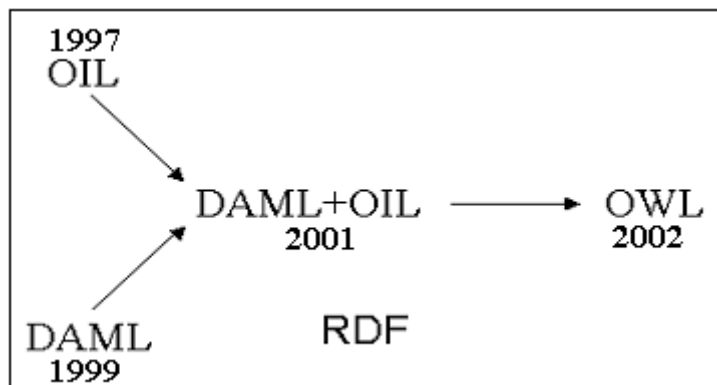


Figura 10: Evolução da Linguagem OWL [Hendler 2005].

2.7.3 Estrutura da linguagem OWL

A linguagem OWL consiste de declaração de *namespace*, um cabeçalho (*Headers*) referente a ontologia e possui também recursos para representar classes, propriedades e indivíduos.

2.7.3.1 Namespaces

A linguagem OWL foi desenvolvida para a construção de ontologias, e uma de suas funcionalidades é permitir a obtenção de informações de várias fontes diferentes, disponibilizando mecanismos para representar um domínio de forma compartilhada e acessível para a integração com outras ontologias. Por exemplo, uma classe definida na ontologia A pode ser reutilizada em uma ontologia B [Smith et al 2004]. Um dos recursos utilizados pela linguagem OWL para prover este mecanismo é a utilização de *namespace*, também utilizado para evitar possíveis colisões entre vocabulários definidos em diferentes documentos, pois é possível que sejam utilizados os mesmos nomes para elementos e atributos.

Antes de começar a definir os termos, é necessária uma indicação precisa da localização dos vocabulários específicos que serão utilizados. Um componente padrão inicial de uma ontologia inclui a declaração de um conjunto de *namespaces XML* incluídas entre *tags rdf:RDF* [Smith et al 2004]. Os *namespaces* fornecem meios para

tratar de ambigüidades e fazer com que a ontologia se torne mais legível. Uma ontologia típica OWL começa com uma declaração dos *namespaces*. Na Figura 11 tem um exemplo de declaração dos *namespaces* de uma ontologia de vinhos.

```

1. <rdf:RDF
2.   xmlns      = "http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#"
3.   xmlns:vin  = "http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#"
4.   xml:base   = "http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#"
5.   xmlns:food = "http://www.w3.org/TR/2004/REC-owl-guide-20040210/food#"
6.   xmlns:owl  = "http://www.w3.org/2002/07/owl#"
7.   xmlns:rdf  = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
8.   xmlns:rdfs = "http://www.w3.org/2000/01/rdf-schema#"
9.   xmlns:xsd  = "http://www.w3.org/2001/XMLSchema#">

```

Figura 11: Declaração de namespaces OWL [Smith et al 2004].

Os *namespaces* são declarados através do atributo *xmlns*. No exemplo da Figura 11 as duas primeiras declarações, linhas 2 e 3, identificam o *namespace* associado com a ontologia. Na linha 5, é declarado o *namespace* que representa a ontologia que faz parte do domínio de *food* (comida). As restantes localizam as definições primitivas de OWL, RDF, RDFS e XML Schema.

2.7.3.2 Headers

Uma vez que os *namespaces* são estabelecidos, é normalmente incluída uma coleção de afirmações sobre a ontologia agrupadas em uma *tag owl:Ontology*, veja Figura 12. Estas *tag* suportam as tarefas críticas de comentários, controle de versões e inclusão de novas ontologias [Smith et al 2004].

```

<owl:Ontology rdf:about="">
  <rdf:comment>An example OWL ontology</rdf:comment>
  <owl:priorVersion rdf:resource="http://www.w3.org/TR/2003/PR-owl-guide-20031215/wine"/>
  <owl:imports rdf:resource="http://www.w3.org/TR/2004/REC-owl-guide-20040210/food"/>
  <rdf:label>Wine Ontology</rdf:label>
  ...

```

Figura 12: Exemplo de uma declaração de Headers.

Na Figura 12, o elemento *owl:Ontology* é um lugar para coletar os metadados OWL para o documento. Embora não garanta que o documento descreva uma ontologia no sentido tradicional. Em algumas comunidades, as ontologias não são sobre indivíduos, mas somente para as classes e propriedades que definem um domínio. Ao usar OWL para descrever uma coleção de instâncias de dados na *tag owl:Ontology* pode ser necessário gravar a versão da informação e importar as definições que o documento tem dependência. Assim em OWL os termos da ontologia devem estar nos dados.

2.7.3.3 Classes

As classes fornecem um mecanismo de abstração agrupando recursos com características similares. De forma similar às classes de RDF, cada classe da linguagem OWL é associada com uma série de indivíduos, chamados de instâncias da classe. [Bechhofer et al 2004].

Os conceitos básicos em relação a um domínio devem corresponder as classes que são as raízes das árvores de taxonomia. Cada objeto na linguagem OWL é um membro da classe *owl:thing*, que já vem definida por padrão da linguagem OWL, onde cada classe definida pelo usuário será implicitamente uma subclasse de *owl:Thing*, tornando-se assim a raiz de todas as classes definidas pelos usuários. As classes específicas de um determinado domínio são definidas simplesmente declarando-se um nome a elas. Por exemplo, para o domínio de vinhos, são criadas três classes raiz: *Winery*, *Region*, e *ConsumableThing*, vejamos a Figura 13 [Smith et al 2004].

```
<owl:Class rdf:ID="Winery"/>
<owl:Class rdf:ID="Region"/>
<owl:Class rdf:ID="ConsumableThing"/>
```

Figura 13: Exemplo de declaração de classes.

As classes são declaradas através do elemento *owl:Class* e o nome das classes são definidos pelo atributo *rdf:ID*, o qual é semelhante ao atributo ID do XML, onde serve como identificador do recurso. Formalmente, não sabe-se nada sobre estas classes à exceção de sua existência, elas poderiam também ser chamadas de *Thing1*, *Thing2*, e *Thing3*. Por exemplo, para referenciar a classe *Region* dentro do mesmo documento, usamos a declaração *rdf:resource="#Region"*. Para que outras ontologias possam referenciar a classe *Region*, é necessário utilizar seu caminho completo através de uma URI, como visto no exemplo demonstrado por Smith [Smith et al 2004], <http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#Region>.

Outro construtor fundamental na taxonomia para classes é o *rdfs:subClassOf*, permite que se diga que uma classe mais específica esteja relacionada a uma classe mais geral [Smith et al 2004]. Se X for uma subclasse de Y, então cada instância de X é também uma instância de Y. A relação de *rdfs:subClassOf* é transitiva, se X for uma subclasse de Y e Y uma subclasse de Z, então X é uma subclasse de Z. A Figura 14, um

exemplo de declaração de uma subclasse, onde é definido que *PotableLiquid* é uma subclasse de *ConsumableThing*.

```
<owl:Class rdf:ID="PotableLiquid">
  <rdfs:subClassOf rdf:resource="#ConsumableThing" />
  ...
</owl:Class>
```

Figura 14: Exemplo de declaração de subclasses OWL.

2.7.3.4 Indivíduos

Além das classes, pode ser necessário descrever seus membros, os quais são considerados como indivíduos em universos modelados. Um indivíduo é introduzido na ontologia, declarado para ser um membro de uma classe [Smith et al 2004].

No exemplo a seguir, é mostrado que *CentralCoastRegion* (uma área específica) é membro de uma *Region*, a classe que contém todas as regiões geográficas.

```
<Region rdf:ID="CentralCoastRegion" />
```

Figura 15: Exemplo de declaração de um Indivíduo.

O próximo exemplo é idêntico no significado ao exemplo acima, onde o *rdf:type* é uma propriedade do RDF que relaciona um indivíduo a uma classe da qual é um membro.

```
<owl:Thing rdf:ID="CentralCoastRegion" />
<owl:Thing rdf:about="#CentralCoastRegion">
  <rdf:type rdf:resource="#Region"/>
</owl:Thing>
```

Figura 16: Exemplo de declaração de um Indivíduo.

Não há nenhuma exigência que os dois elementos necessitem estar juntos ou que necessitem estar no mesmo arquivo (embora os nomes devam ser estendidos com uma *URI*). As ontologias são projetadas na *Web* para serem distribuídas. Podendo ser importadas e aumentadas, criando assim ontologias derivadas.

2.7.3.5 Propriedades

As propriedades descrevem características e relacionamentos entre classes. Assim como existe a hierarquia de classes, é possível também criar uma hierarquia de propriedades. As propriedades afirmam fatos gerais sobre os membros das classes e fatos específicos sobre os indivíduos [Smith et al 2004].

A linguagem OWL define duas categorias principais de propriedades que um construtor de ontologias pode definir [Bechhofer et al 2004]:

- *Object properties (owl:ObjectProperty)*: relaciona uma classe a outra;
- *Datatype properties (owl:DatatypeProperty)*: define um tipo de dado, que pode ser uma string ou um número.

2.8 Considerações Finais

A *Web Semântica* ainda está dando seus primeiros passos, sendo difícil prever seu futuro. A complexidade envolvida entre suas diversas tecnologias ainda é um grande empecilho, mas isso poderá ser contornado com a sua consolidação e a contínua melhoria nas ferramentas disponíveis para facilitar a sua utilização.

A *Web* fornece um vasto campo de provas para diversas teorias envolvendo a busca e recuperação da informação. Desde sua criação poucas mudanças ocorreram em sua estrutura básica. Talvez a *Web Semântica* seja a mudança necessária para que a *Web* se torne realmente uma fonte de informação confiável.

3. Anotação semântica de documentos

Neste capítulo são apresentados conceitos relacionados com a anotação semântica de documentos. São descritas também as técnicas e linguagens disponíveis que foram utilizadas na construção do modelo.

A anotação semântica adiciona no documento uma camada que descreve o seu conteúdo, permitindo que agentes de software possam recuperar informações de forma mais precisa, através da associação do documento a uma ontologia.

A estrutura das anotações semânticas a serem utilizadas na *Web Semântica* deve seguir as recomendações do W3C, na elaboração de sua estrutura e na utilização do modelo RDF como a linguagem para descrever os recursos.

3.1 Considerações a respeito de anotação semântica de documentos

Do ponto de vista da *Web Semântica*, o documento dotado de uma estrutura semântica deve ser constituído de quatro componentes distintos:

- Ontologia;
- Instâncias;
- Anotação;
- Conteúdo;

A ontologia define conhecimento sobre o domínio de interesse ou sobre as informações descritas no documento. As instâncias são os *individuals* da ontologia, que também são chamados de instâncias de classe. Uma *anotação* pode ser definida como um bloco de texto que descreve a relação do recurso (documento) com uma instância da ontologia, o qual fica anexado ao conteúdo. O conteúdo representa a informação contida no documento.

O conceito de anotação semântica de documentos pode ser entendido como sendo o nome dado a algum conteúdo anexado a um documento para descrever a sua semântica. A anotação pode vir a ser parte do documento original desde que não cumpra um papel organizacional, como por exemplo, uma formatação de estilos, um resumo ou um cabeçalho.

Um *documento estruturado* é composto por *conteúdo* (texto, figuras, etc) e *informação sobre este* no documento ou como ele está *estruturado*. Por exemplo: um artigo técnico é usualmente composto por um "título", "autores", "resumo", diversas "seções" e uma "bibliografia", nesta ordem. Cada um destes componentes (ou "elementos") representa uma *parte estrutural* do documento, mas estas informações atualmente estão visíveis apenas para humanos não sendo possível serem identificadas pelas ferramentas de busca na *Web*.

Para possibilitar que estas ferramentas também tenham acesso a tal conteúdo, faz-se necessária a inserção de anotações semânticas nos documentos. Tais anotações consistem na adição de metadados a um documento estruturado, inserindo um bloco de texto, relacionado com uma ontologia sobre um determinado domínio. A anotação deve ser definida rigorosamente, de forma que o conteúdo possa ser entendido por humanos e pelas ferramentas de buscas.

Com a utilização da anotação semântica das informações, há uma nova perspectiva de possibilidades para o projeto de mecanismos de recuperação de informações. Nota-se uma preocupação atual por parte das maiores ferramentas de busca na *Web* em preparar-se para essa nova visão, que é constituída gradualmente de um número cada vez maior de documentos marcados semanticamente. Faz parte do escopo da ciência da informação o estudo de processos de indexação e recuperação de informações e, nesta perspectiva, é bem provável que venhamos a confrontar nossas linguagens artificiais de indexação com as metodologias de anotação semântica dos dados representados pelos metadados e *namespaces* da *Web Semântica* e, também, da lógica formalizada do XML e do RDF [Souza & Alvarenga 2004].

As marcações devem ser feitas através da linguagem XML, pois é uma linguagem de marcação apropriada à representação de dados, cuja essência fundamenta-se na capacidade de agregar informações [Bray 2004 et al].

A linguagem XML não possui *tags* predefinidas. Ela permite definir novos elementos para criar vocabulários específicos, sendo assim possível incluir as marcações semânticas para representação dos metadados do documento.

Uma linguagem de anotação é um mecanismo para explicitar e identificar estruturas em um documento. A especificação XML define uma forma padrão de adicionar uma anotação a documentos [Mendonça 2003].

O W3C recomenda que sejam usadas ontologias escritas na linguagem OWL para representação de conceitos sobre um determinado domínio de conhecimento a serem utilizadas na *Web Semântica*. E que as anotações semânticas, que venham a descrever a relação de recursos na *Web* e as instâncias de uma ontologia, sejam escritas utilizando o modelo RDF [Bechhofer et al 2004].

3.2 Ferramentas de Anotação Semântica de Documentos

Durante toda fase de planejamento e desenvolvimento do projeto, foram analisadas as ferramentas disponíveis atualmente destinadas a anotação semântica de documentos e páginas na *Web*.

Uma das motivações deste trabalho é propor um modelo de ferramenta que contemple aspectos, características e funcionalidades não suportados pelas ferramentas disponíveis. Partindo deste princípio, foram encontrados na *Web* algumas ferramentas com propósitos semelhantes com os deste trabalho, mas nenhuma delas com as mesmas características.

Dentre as ferramentas analisadas, ressalta-se e descreve-se a seguir três projetos. São eles: *OntoMat Annotizer* [Ontomat 2004], *SemanticWord* [Tallis 2003], *Annotea* [Koivunen et al. 2001].

3.2.1 *OntoMat Annotizer*

O *OntoMat Annotizer* é uma ferramenta interativa, amigável para o usuário para anotações em páginas *Web*, utilizando a linguagem para representação de ontologias OWL.

Na Figura 17, pode ser visualizado o ambiente onde é realizada a anotação semântica, no canto esquerdo é disponibilizado um *browser*, onde a ontologia fica disponível para o usuário realizar a sua edição, pode ser criado desde novas classes ou até mesmo novas instancias. No lado direito da tela, é disponibilizado um *browser* para navegação na *Web*, onde páginas HTML podem ser carregadas em memória para a realização da anotação semântica.

A ferramenta *OntoMat Annotizer* foi desenvolvida utilizando a linguagem Java. Para seu funcionamento, recomenda-se ter instalado a versão do Java 1.4.0 ou superior.

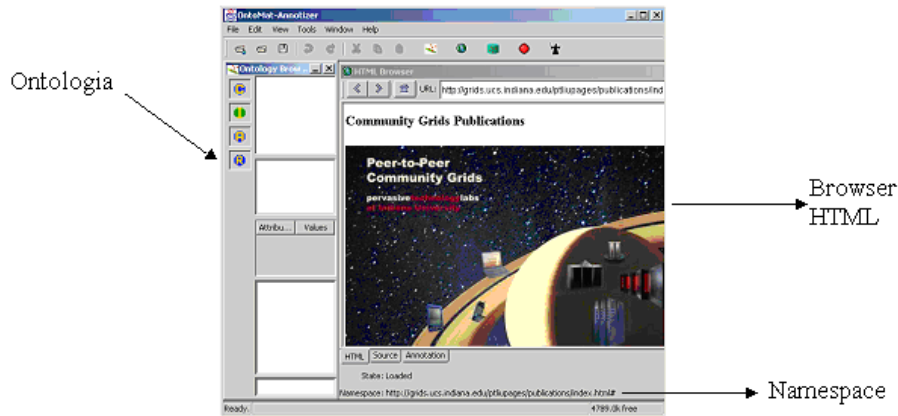


Figura 17: Ferramenta *Ontomat Annotizer*.

3.2.2 *Semantic Word*

É um ambiente baseado no Word XP, editor de textos do pacote Office da Microsoft⁸. Esta ferramenta permite ao autor de um documento anexar a uma parte do texto original um bloco de texto contendo a anotação semântica, que é restrita a apenas uma região do texto e não ao texto como um todo.

Esta ferramenta adiciona ao Word XP uma barra de ferramentas e menus, que auxiliam o autor do documento na tarefa de anotação semântica, como visto na Figura 18 [Tallis 2003].

Oferece recursos para realizar pesquisas nos documentos em busca de novos conceitos durante a digitação do conteúdo documento, agilizando o processo de anotação.

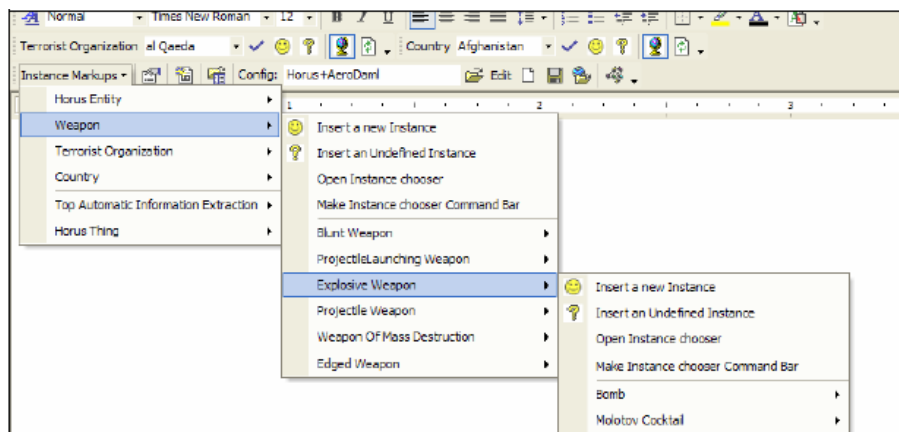


Figura 18: Barra de Ferramenta e Menus do *SemanticWord*.

⁸ <http://www.microsoft.com/> - Site oficial da Microsoft.

A ferramenta *SemanticWord* foi desenvolvida sobre a API Jena 1 e utiliza para isso a ferramenta comercial *JNBridge*⁹, que faz a conexão entre o Java e a plataforma .Net da Microsoft. As anotações desta ferramenta são baseadas na linguagem de representação de ontologias DAML+OIL [Connolly et al 2004].

A versão atual do *SemanticWord* requer que se tenha instalado o MS Word XP e o JNBridge. Ambos são produtos comerciais que necessitam a aquisição de licença.

3.2.3 Annotea

O *Annotea* [Koivunen et el. 2001] é um projeto de código aberto do W3C, que permite a criação de anotações com o uso de metadados. As anotações são recursos *Web* de primeira classe e podem ser armazenadas em um ou mais servidores de anotação. O servidor de anotação armazena as anotações em uma base de dados RDF e toda a comunicação entre o cliente e o servidor utiliza métodos HTTP.

Uma anotação do *Annotea* é representada como um conjunto de metadados e um corpo de anotação. Os metadados são modelados através de um esquema RDF que fornece informações como data de criação, nome do autor, o documento anotado e um XPointer [Maler et al 2003] que especifica a porção do documento anotado.

As anotações podem ser privadas ou compartilhadas. Anotações privadas são armazenadas localmente e restritas à máquina do usuário. Já as anotações compartilhadas são armazenadas na *Web* e podem ser lidas por todos os usuários registrados em um servidor remoto.

O *Annotea* foi desenvolvido para o editor e *browser Amaya*¹⁰. Na Figura 19 é mostrado um exemplo de anotação em um documento *Web* utilizando o *browser Amaya* com o *Annotea*. Para identificar as anotações em um documento *Web* é utilizado um ícone em forma de lápis em frente à porção anotada.

⁹ <http://www.jnbridge.com/index.htm> - Site oficial da ferramenta *Java .Net bridge*.

¹⁰ <http://www.w3.org/Amaya/> - Site oficial da editor/*browser Amaya*.

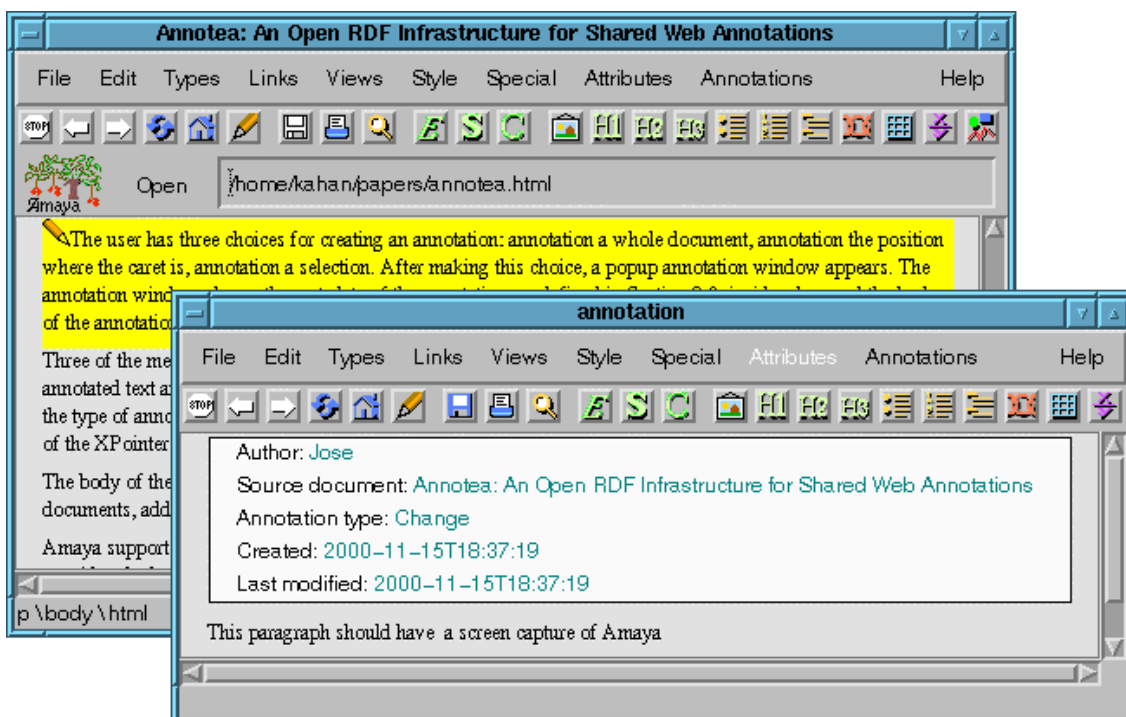


Figura 19: Exemplo de anotação de um parágrafo utilizando o *browser* e editor Amaya com o serviço de anotações Annotea.

A ferramenta *Annotea* consiste de um servidor implementado sobre CGIs Perl¹¹ interagindo com uma base de dados MySQL¹². O esquema de anotações proposto pelo *Annotea* é baseado em padrões abertos como RDF, XML e Xpointer [DeRose et al. 2004]

3.2.4 O trabalho proposto frente aos trabalhos correlatos

O presente trabalho visa a definição de um modelo para a anotação de documentos no processo de autoria, utilizando somente ferramentas e APIs disponíveis sob a política de software livre (*OpenSource*), não sendo necessária a aquisição de nenhuma licença para seu funcionamento

Dentre os trabalhos correlatos, o que mais se aproxima com a proposta deste trabalho é a ferramenta *SemanticWord*. Entretanto, exige a aquisição de licenças de produtos comerciais para seu funcionamento, diferentemente deste trabalho que se propõe a contemplar apenas pacotes de aplicativos para escritórios e ferramentas que

¹¹ <http://www.perl.org/> - Perl.

¹² <http://www.mysql.com/> - MySQL.

funcionem na política de software livre. Outra diferença importante, a ferramenta *SemanticWord* utiliza a linguagem para representação de ontologias DAML+OIL, e neste trabalho, a linguagem escolhida foi a OWL. A ferramenta *SemanticWord* utiliza a API Jena para a manipulação de ontologias. A semelhança desta solução com a deste trabalho, que ambas soluções realizam anotações semânticas em editores de texto.

Em relação à ferramenta *OntoMat Annotizer*, a diferença é que ela trabalha apenas para páginas *Web* no formato HTML. A ferramenta *OntoMat Annotizer* utiliza a linguagem OWL para representação de ontologias.

Com relação à ferramenta *Annotea*, é a que mais se distancia do objetivo deste trabalho, pois se destina a anotações colaborativas de um grupo de pessoas sobre um determinado documento. A ferramenta também só permite que as anotações sejam realizadas para páginas *Web*.

Informações adicionais sobre ferramentas de anotação podem ser obtidas no portal *Semantic Web Annotation & Authoring Portal*¹³. Este *site* reúne informações sobre ferramentas e publicações a respeito de anotações na *Web Semântica*.

3.3 Considerações Finais

Do estudo realizado a respeito de anotação semântica de documentos, podemos citar as seguintes conclusões:

- Anotações devem ser entidades com características próprias;
- Anotações semânticas anotam o documento como um todo ou partes dele;
- Anotações podem ser armazenadas em repositórios ou no próprio documento;

Com relação as ferramentas de anotação apresentadas neste capítulo, foram criadas com objetivos diferentes, com mecanismos de anotações distintos e de várias complexidades, mas em alguns pontos eles possuem certa semelhança com este trabalho, mas nenhum deles com o mesmo objetivo. Por exemplo, o *SemanticWord* utiliza a linguagem DAML+OIL e utiliza ferramentas que necessitam da aquisição de licenças, e neste trabalho é utilizada a linguagem OWL e apenas ferramentas *Opensource*. O *Annotea* realiza apenas anotações colaborativas entre um grupo de

¹³ <http://annotation.semanticweb.org/> - *Semantic Web Annotation & Authoring Portal*.

trabalho, diferentemente deste trabalho que realiza anotações semânticas em documentos. A ferramenta *Ontomat Annotizer* é bastante semelhante, utiliza a linguagem OWL para realização das anotações semânticas, mas só funciona para documentos no formato HTML.

4 Modelo para anotação semântica de documentos

Neste capítulo são apresentadas questões sobre os elementos relacionados ao modelo de anotação semântica de documentos para autores de conteúdo em pacotes de ferramentas para escritórios. O trabalho mostra uma estratégia para a inserção de anotações semânticas em documentos, para isto, são descritos módulos que possibilitam o acesso aos diversos componentes necessários a realização de uma anotação.

A anotação semântica de documentos é o que permite que agentes de software tenham uma melhor precisão na recuperação de informações solicitadas por usuários. A técnica faz com que os agentes que sejam capazes de entender uma ontologia, possam retornar informações mais precisas e confiáveis.

A estrutura das anotações semânticas utilizada no modelo segue as especificações e recomendações do W3C para elaboração de estruturas, seguindo os padrões estabelecidos para a *Web Semântica*.

O modelo de anotação proposto neste trabalho faz parte de uma solução conjunta para que empresas consigam gerenciar o seu capital relativo ao conhecimento, utilizando os conceitos da *Web Semântica*, através de técnicas de anotações semânticas e de ferramentas de busca e recuperação de informações.

A ferramenta de busca e recuperação não faz parte do modelo proposto, ficando este a cargo de uma ferramenta que está sendo desenvolvida em outro projeto de pesquisa ainda não finalizado [Zago 2005]. No item a seguir, é descrita a arquitetura do modelo de anotação semântica proposta neste trabalho.

4.2 Arquitetura do modelo de anotação

O diagrama do modelo de anotação semântica criado nesta abordagem é apresentado na Figura 20 e seus componentes são descritos nas seções subseqüentes.

Inicialmente, o autor do documento deve indicar na “Interface com o Usuário” a URI da ontologia que contém as classes instanciadas, as quais serão mapeadas e listadas para fornecer um acesso transparente e uniforme aos dados que servirão de base para a elaboração do processo de anotação semântica.

É também informado pelo autor da anotação, na “Interface com o Usuário”, a URI do documento que receberá o conteúdo da anotação semântica. Este documento deve estar armazenado no “Repositório de Documentos” que a empresa utiliza, para que,

futuramente, ele possa ser acessado pela ferramenta de busca e recuperação de informações semânticas utilizadas pela empresa.

Para compor a tripla RDF (sujeito, predicado e objeto), a qual vem a ser a anotação semântica que será incluída nas propriedades do documento, é necessário que o autor escolha na “Interface com o Usuário” as instâncias da ontologia que ele deseja usar na anotação, estas instâncias representadas por uma URI, serão o “Sujeito” da anotação. O objeto é representado pelo *namespace* da ontologia, mais o nome da classe à qual pertence a instância, formam a URI do recurso que representa o objeto.

O predicado é a propriedade que faz parte de uma tripla. Ela descreve a relação entre o sujeito e o predicado. No modelo proposto, a anotação semântica utiliza ontologias para descrever um recurso, o predicado descreve a relação da instância com sua classe, então para declarar que um recurso (instância) representado por uma URI pertence a uma classe *owl:ObjectProperty* de uma ontologia, é utilizado o *RDF:type* como valor do predicado da tripla [Carroll et al 2004].

A criação de ontologias e suas instâncias não fazem parte do modelo proposto. Esta tarefa pode ser realizada através de ferramentas, tais como o Protégé [Protégé 2005], Ontomat [Ontomat 2004] ou o OntoEdit [OntoEdit 2004]. O modelo proposto não restringe a ferramenta que será usada para a criação e instanciação da ontologia, apenas define que ela seja escrita na linguagem OWL.

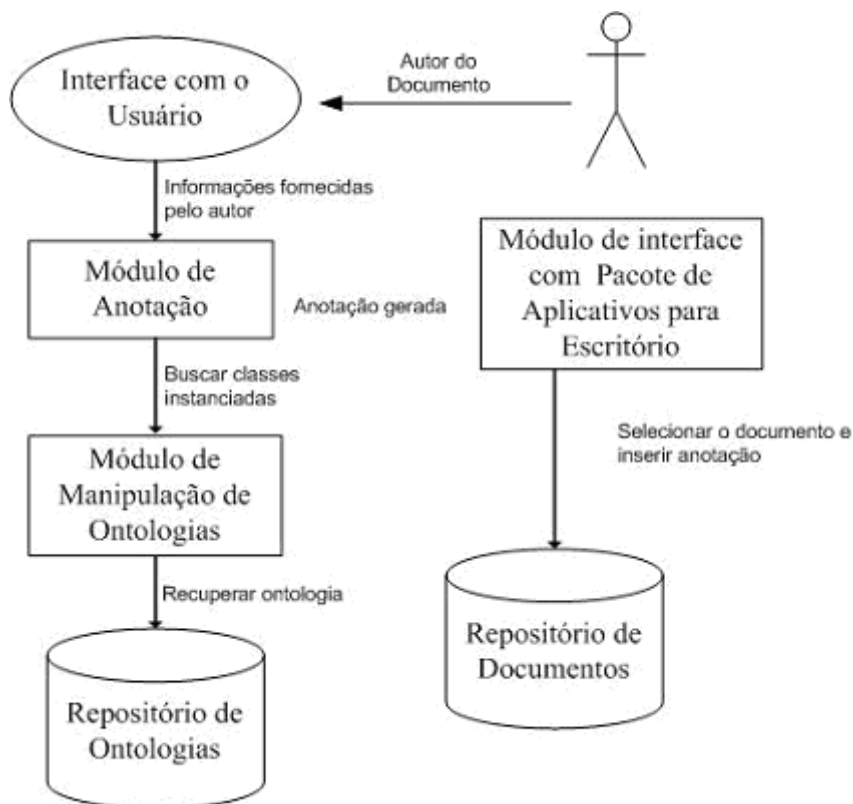


Figura 20 - Modelo de Anotação Semântica.

Na ferramenta de busca e recuperação de informações, o sistema executa uma varredura no repositório de documentos de forma sistemática e realiza o armazenamento das triplas RDF em um repositório de dados, para que sejam posteriormente utilizados na busca de informações solicitadas pelo usuário.

Para que o modelo seja implantado, uma empresa deve manter dois repositórios, um para armazenar as ontologias criadas pelos administradores e outro para concentrar os documentos que contém o conhecimento de diversos setores da empresa.

A seguir são apresentadas as funcionalidades de cada componente da arquitetura que faz parte do modelo proposto ilustrado na Figura 15.

4.2.1 Interface com o Usuário

A “Interface com o Usuário” caracteriza-se por envolver o usuário em uma interação com o ambiente para realização da anotação semântica do documento. Ela realiza a comunicação entre o “Módulo de Anotação” do modelo e o autor da anotação semântica.

Na “Interface com o Usuário”, o autor deve informar os dados necessários para que o “Módulo de Anotação” possa realizar as tarefas necessárias ao processo de anotação semântica do documento.

Primeiramente, o autor da anotação deve selecionar na “Interface com o Usuário” a ontologia que contém as instâncias. Para isto lhe é fornecida uma lista de todas as ontologias disponíveis no “Repositório de Ontologias” da empresa. Posteriormente, as instâncias são disponibilizadas em uma lista para que ele possa escolher qual delas atende as suas necessidades. O autor também deve selecionar o documento que receberá a anotação dentre uma lista de arquivos disponíveis no “Repositório de Documentos” da empresa.

4.2.2 Módulo de Anotação

O “Módulo de Anotação” deve fazer a associação da instância da ontologia ao documento. Esta associação é realizada através da criação da anotação semântica, que consiste em uma tripla escrita na linguagem RDF. Os conceitos de RDF são descritos no item 2.4 A camada RDF.

Com a URI da ontologia fornecida pelo autor na “Interface como Usuário”, o “Módulo de Anotação” pode acessar o “Módulo de Manipulação de Ontologias”, o qual deve fornecer todos os *individuals* da ontologia. Estes *individuals* são as instâncias de classes já criadas na ontologia.

Todos os *individuals* disponíveis formam uma lista de instâncias que ficam disponíveis na “Interface com o Usuário”, para que ele possa escolher a instância necessária para compor a tripla RDF. Esta instância escolhida vem a ser o sujeito da tripla RDF.

Não é objetivo do modelo gerar uma ferramenta para criar instâncias em uma ontologia, isto deve ser feito em alguma ferramenta apropriada para isto, durante a realização dos testes foi utilizada a ferramenta Protégé [Protégé 2005] para a criação de novas instâncias.

Com todas as informações necessárias para a criação da tripla RDF disponíveis, o “Módulo de Anotação” informa ao “Módulo de Manipulação de Ontologias” as informações necessárias para geração da tripla RDF.

Após a tripla gerada, o “Módulo de Manipulação de Ontologias” retorna ao “Módulo de Anotação” a tripla, a qual é disponibilizada na “Interface com o Usuário” para a visualização do autor da anotação.

O autor pode então informar ao “Módulo de Anotação” que ele pode inserir a anotação no documento. O “Módulo de Anotação” então invoca o “Módulo de Interface com Pacote de Aplicativos para Escritório” para realizar a inserção no documento da anotação gerada, o qual busca no “Repositório de Documentos”, o arquivo que recebera a anotação semântica e realiza a anotação no documento. O conteúdo da anotação semântica deve ficar armazenado nas propriedades do documento.

4.2.3 Módulo de Manipulação de Ontologias

Para que as ontologias possam ser efetivamente utilizadas por aplicações, faz-se necessário que exista um módulo que disponha de um conjunto de recursos que permitam um fácil acesso às informações descritas nas ontologias utilizando a linguagem de representação OWL, a qual possui recomendação do W3C como linguagem padrão para representação de ontologias para a área de *Web Semântica*.

O “Módulo de Manipulação de Ontologias” deve garantir o suporte à manutenção das informações contidas em uma ontologia através de operações de inserção, de remoção e de atualização de seus conceitos e instâncias.

Deve possuir componentes que permitam representar classes e as propriedades descritas em uma ontologia, bem como estrutura de dados para manipular as informações dos conceitos descritos.

O módulo também deve prover a estrutura para manipulação de grafos RDF, os quais são utilizados para representar as triplas (sujeito, predicado e objeto), as quais descrevem uma anotação semântica para representação de recursos disponíveis na *Web Semântica*.

4.2.4 Módulo de Interface com Pacote de Aplicativos para Escritório

O “Módulo de Interface com Pacote de Aplicativos para Escritórios” deve possuir as classes e funções necessárias para acessar tanto a estrutura de um documento, quanto o seu conteúdo.

Para este módulo, é proposto duas opções para o armazenamento da anotação semântica dentro do documento. A primeira alternativa é anexar nas propriedades do arquivo, onde normalmente o usuário tem a opção de colocar comentários sobre o texto. A vantagem desta abordagem é uma melhor organização do texto, facilitando ao autor do documento localizar a sua anotação.

A segunda opção é anexar a anotação a partes do texto, como se fosse uma nota do autor. A vantagem desta abordagem seria a possibilidade de inserção de diversas anotações em um mesmo documento.

4.2.5 Repositório de Ontologias

O “Repositório de Ontologias” é o local onde ficam armazenadas as ontologias a serem utilizadas no processo de anotação da *web* semântica. Estas ontologias devem estar obrigatoriamente identificadas por uma URI e podem estar armazenadas localmente ou disponibilizadas em qualquer lugar da *Web*.

No contexto deste modelo, o “Repositório de Ontologias”, seria o local da empresa, onde as diversas ontologias utilizadas pelos funcionários da empresa ficariam armazenadas, para que todos os autores de conteúdo da empresa possam localizar as ontologias necessárias para realização da codificação dos conceitos de sua anotação semântica.

O modelo não impede que ontologias escritas por autores ou grupos estudos de algum determinado domínio de conhecimento de fora da empresa sejam utilizadas, desde que obrigatoriamente estejam identificadas por uma URI.

Por ser uma recomendação do *World Wide Web Consortium*, como visto anteriormente, ficou definido no modelo que as ontologias criadas ou disponibilizadas pela empresa, devem estar escritas na linguagem OWL.

4.2.6 Repositório de Documentos

O “Repositório de Documentos” é o local onde ficam armazenadas os documentos a serem utilizadas no processo de anotação da *web* semântica. Estes documentos devem estar obrigatoriamente identificados por uma URI.

O documento é o arquivo que irá receber a anotação semântica. O modelo de anotação prevê que o documento seja armazenado no repositório de documentos da

empresa, para que ele possa ser futuramente rastreado e sua informação possa ser recuperada por um módulo de busca e recuperação de informações.

O processo de anotação semântica pode ser realizado a qualquer momento, não sendo necessária a elaboração da anotação no momento de sua autoria, o que não impede que documentos antigos recebam anotações.

O autor do documento deve sempre informar a URI que identifica a localização do arquivo que irá receber a anotação, para que o “Módulo de Anotação” possa fazer o acesso ao documento.

4.3 Considerações Finais

Do estudo realizado para geração do modelo de anotação semântica de documentos, podemos fazer algumas considerações. A utilização da linguagem OWL mantém o padrão com as demais tecnologias sendo desenvolvidas para a utilização na *Web Semântica*.

A utilização de repositórios de documentos e de ontologias auxilia na manutenção e na organização dos arquivos dentro de uma empresa.

A adoção de tecnologias *OpenSource* facilita futuras extensões dos serviços de anotação, suporta uma melhor interoperabilidade entre aplicações e a redução dos custos para a adoção da solução nas empresas.

5 Ferramenta implementada

No capítulo anterior foi apresentada uma proposta de um modelo de anotação semântica baseado nos conceitos propostos pelo W3C para construção de aplicativos para a *Web Semântica*.

O propósito deste capítulo é mostrar a implementação de uma ferramenta para utilização deste modelo, cuja especificação foi apresentada no capítulo anterior. A escolha das ferramentas para implementação do modelo foram restritas a ferramentas *OpenSource* como dito anteriormente. Neste sentido, escolhemos como meta a implementação do modelo utilizando o pacote de aplicativos para escritórios OpenOffice.org (OO).

A ferramenta lida com repositórios de documentos e de ontologias para auxiliar o autor durante o processo de anotação semântica. Entretanto, a ferramenta limita-se a criação de anotações semânticas baseadas em ontologias já com instâncias criadas. Os aspectos de criação de ontologias e suas instâncias, não fazem parte do modelo proposto, pois se encontram fora do escopo desta dissertação.

Com base no pacote de aplicativos escolhido, foi feito um estudo sobre as alternativas possíveis para a implementação do modelo. A seguir são mostradas as tecnologias empregadas na construção do modelo.

5.1 Tecnologias empregadas

Seguindo a política de software livre, decidiu-se implementar o modelo utilizando a linguagem Java. Sendo assim, foram utilizados dois *Frameworks*, desenvolvidos na linguagem Java: o primeiro foi a API Jena que possui os recursos necessários para a manipulação de ontologias, e o segundo foi a API Java UNO que possui os recursos para a manipulação de documentos do pacote de aplicativos OO. Estas tecnologias são descritas detalhadamente nos tópicos a seguir.

5.1.1 A Linguagem Java

A escolha da linguagem Java para implementação do modelo proposto deu-se pelo fato de ser a única que consegue reunir todos os requisitos citados neste específico modelo, para implementar a ferramenta, não tendo a necessidade de conexão com outras linguagens para se obter recursos para implementação do modelo.

Java foi criada com a tecnologia de orientação a objetos como uma evolução da linguagem “C++” em 1991, e lançada ao mercado em 1995. Esta linguagem atende perfeitamente ao desenvolvimento de aplicativos para a *Web*, devida a sua principal característica de portabilidade total de código entre diversas plataformas distintas, sem a necessidade da recompilação de seu código fonte.

A linguagem possui inúmeros pacotes desenvolvidos, *freeware* e *shareware*, para os mais diferentes propósitos, incluindo motores de inferência e linguagens de comunicação entre agentes. Entre estes pacotes desenvolvidos, destacamos a API Jena, melhor descrita no item 5.1.2 API , que possui os recursos necessários para a manipulação de ontologias escritas na linguagem OWL e de grafos RDF. Outro pacote desenvolvido em Java que foi utilizado na implementação do modelo foi o *Software Development kit* (SDK), descrito no item 5.1.3 , o qual contém a API Java UNO que realiza o acesso e a manipulação de arquivos do pacote de aplicativos OO.

5.1.2 API para manipulação de ontologias

Durante as pesquisas foram analisadas diversas APIs para manipulação de ontologias, dentre as API's disponíveis atualmente, que possuem suporte a linguagem OWL e a representação de modelos RDF, podemos destacar as seguintes:

- *Jena [Jena 2005]*: é uma API Java para a criação de aplicações que necessitam manipular ontologias. Desenvolvida pela Hewlett-Packard (HP), provê o suporte a manipulação das linguagens RDF, RDFS, DAML+OIL e OWL. Permite a geração de grafos RDF, que representam a base de uma anotação semântica. Possui suporte ao armazenamento de ontologias de forma persistente e em memória, bem como a consultas sobre as ontologias através da linguagem RDQL.
- *KAON (Karlsruhe Ontology and Semantic WebTool Suite) [KAON 2004]*: é um *framework* para construção de aplicações baseadas em ontologias. O principal objetivo do projeto KAON é a escalabilidade de inferência em grandes ontologias e bases de conhecimento. Sua API define um conjunto de ferramentas para criação e manipulação de ontologias, que podem estar representadas no próprio formato da ferramenta, ou também nas linguagens RDF e OWL.

- *SNOBASE (Semantic Network Ontology Base) [SNOBASE 2004]:* é um *framework* que permite acessar ontologias armazenadas em arquivos ou disponíveis na *Web*, bem como criar, modificar, consultar e armazenar informações acerca das mesmas localmente. O objetivo principal desta API é disponibilizar para aplicações a capacidade de manipular e consultar ontologias sem a necessidade da mesma conhecer detalhes acerca de onde e como a ontologia será acessada, como a consulta será processada ou como os resultados serão obtidos. Ela provê suporte as linguagens de representação de ontologias RDF, RDFS, DAML+OIL e OWL.
- *SOFA (Simple Ontology Framework) [SOFA 2004]:* é uma API desenvolvida na linguagem Java para manipulação de ontologias. Seu modelo não segue nenhum padrão definido por qualquer linguagem de ontologia, operando no nível de abstração dos conceitos relacionados com a ontologia, ao invés de adotar construtores que sejam específicos de uma determinada linguagem. Esta API permite a manipulação de ontologias escritas nas linguagens RDF, RDFS, DAML+OIL e OWL. Permite também o armazenamento das ontologias criadas em bases persistentes.

No contexto proposto no modelo de anotação, foi escolhida a API Jena para a implementação do modelo, sendo essa uma escolha pessoal. Esta API foi desenvolvida pela HP, possui referências bibliográficas razoáveis e continua em constante aperfeiçoamento, o que ajudou em muito a sua escolha.

5.1.2.1 API Jena

A API Jena é um *framework* desenvolvido na linguagem Java para construção de aplicações para a *Web Semântica*, fornecendo um conjunto de classes que possibilitam a manipulação dinâmica de modelos em RDF, RDFS e OWL, incluindo um motor de inferência baseado em regras [Jena 2005]. A Jena é uma API de código aberto desenvolvida por Brian McBride da empresa *Hewlett-Packard*, em seu laboratório de pesquisas em *Web Semântica*.

Seu desenvolvimento foi derivado do SiRPAC (*Simple RDF Parser & Compiler*), que se encontrava com suas pesquisas bem adiantadas. O SiRPAC é um conjunto de classes

Java que podem compilar documentos RDF/XML conforme as triplas do modelo de dados do RDF.

As ontologias em Jena são criadas e manipuladas como “grafos” onde os nós são os recursos ou literais e os arcos são as propriedades. Um grafo RDF em Jena é chamado de modelo e é representado pela interface *model*. As interfaces para representação de recursos, propriedades e literais são chamadas *Resource*, *Property* e *Literal*, respectivamente [Jena 2005].

A interface *model* pode ser considerada como sendo o conjunto de interfaces que formam o grafo por completo. O número de interfaces que formam o conjunto para manipulação de expressões em RDF pode ser visto na Figura 21.

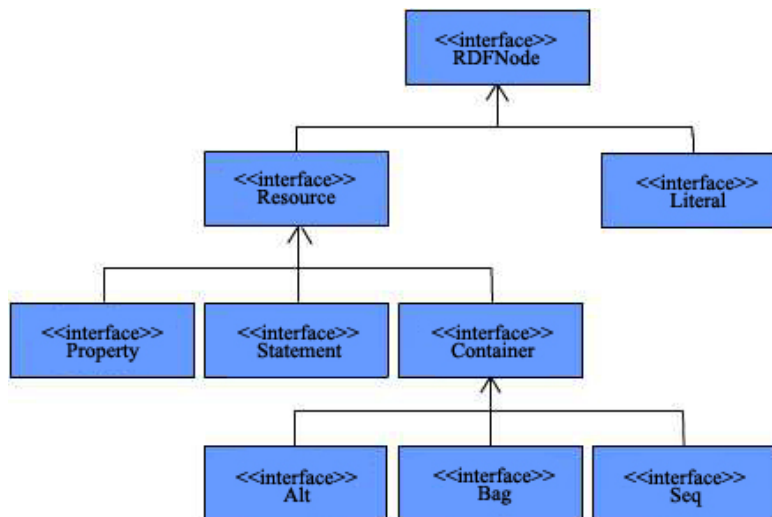


Figura 21 - Hierarquia e relacionamentos das Interfaces da API Jena [Verzulli 2001].

As interfaces de programação da API Jena podem ser visualizadas da seguinte maneira:

- *RDFNode*: fornece a ligação com outros elementos do RDF, para formar as triplas (sujeito, predicado, objeto) para a criação dos elementos;
- *Resource*: esta interface representa objetos que possuam uma URI;
- *Literals*: esta interface representa valores utilizados como objetos em triplas. Também fornecem métodos para conversão de valores para diversos tipos de dados em Java (*string*, *int* e *double*);
- *Property*: representa as propriedades inseridas nas triplas utilizadas;

- *Statement*: representa uma expressão que pode dar origem a uma nova tripla em um novo objeto, onde uma tripla representa um fato sobre um recurso;
- *Container*: representa um conjunto de objetos (*Alt*, *Bag*, *Seq*) em uma tripla.

A API Jena possui duas versões. Em sua primeira versão, não possuía métodos para manipulação de ontologias em OWL, possuindo apenas um pequeno suporte para a linguagem DAML+OIL e para representação de triplas escritas na linguagem RDF.

Na segunda versão, para resolver o problema de suporte, a manipulação de ontologias, foi adicionado o pacote API Jena 2 *Ontology*, o qual possui classes para a manipulação de ontologias em RDFS, DAML+OIL e OWL. Essas linguagens possuem suporte na API através das classes *OntClass* e *ObjectProperty* [Jena 2005].

Para a declaração de um modelo de ontologia como um objeto Java, a API Jena oferece vários pacotes que realizam a interação com um grafo RDF. Estes pacotes podem ser visualizados na Figura 22.

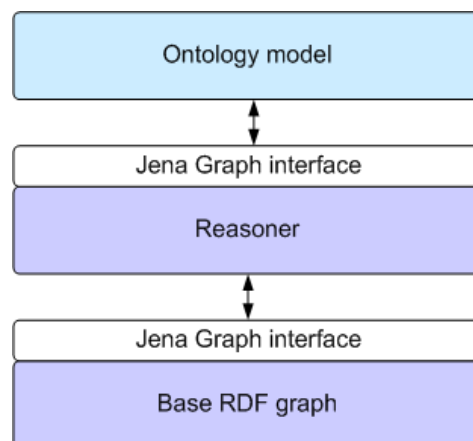


Figura 22 - Interação entre o modelo de uma ontologia e o grafo RDF [Carroll et al 2004].

A Figura 17 apresenta a base de um modelo de ontologia, mostrando a dependência do modelo com a base de um grafo RDF. Para cada novo modelo criado, uma nova interface chamada de *Reasoner* é atribuída ao modelo. Essa interface possui o conjunto de regras para cada linguagem de ontologia, como RDFS, DAML+OIL e OWL. Estes conjuntos de regras são utilizados para acessar a base do grafo RDF [Carroll et al 2004].

Os *Rasoners* disponíveis na API Jena são os seguintes:

- *Transitivo*: suportam o armazenamento e cruzamento de classes e propriedades. Implementam apenas as propriedades transitivas e simétricas do *rdfs: subPropertyOf* e *rdfs: subClassOf*;
- *Regras RDFS*: é baseado nas regras dos RDFS. Implementa um subconjunto configurável de implicações RDFS;
- *OWL FB*: Uma implementação útil, mas incompleta do OWL/*Lite* (subconjunto da linguagem OWL/*Full*);
- *DAML micro*: Utilizada internamente para auxiliar a antiga API DAML a fornecer capacidade de inferência em RDFS similar a disponibilizada pela versão 1 do Jena;
- *Regras gerais*: Baseado em regras definidas pelo usuário. Dá suporte ao *forward chaining*, *tabled backward chaining* e estratégias de execução híbridas.

Na Figura 23 pode ser visualizada a hierarquia dos modelos disponíveis da API Jena.

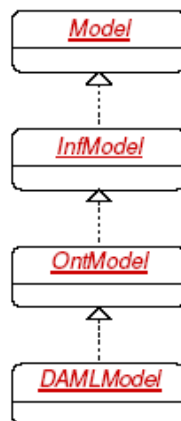


Figura 23 - Hierarquia dos modelos Jena.

A estrutura dos modelos da API Jena é a seguinte:

- *Model*: interface para um modelo RDF, disponibiliza os métodos para criar recursos, propriedades e literais. Adiciona e remove sentenças do modelo, serve também para consultar o modelo e os conjuntos de operações para combinar modelos;
- *InfModel*: extensão da interface *model*, prove o suporte a inferência e gera implicações adicionais num conjunto de dados RDF;

- *OntModel*: extensão da interface *InfModel*, contém os dados de uma ontologia sob um vocabulário específico (Ex: OWL);
- *DAMLMModel*: extensão da interface *OntModel*. Encapsula a capacidade de armazenamento e recuperação de informações de ontologias DAML de um serviço de persistência.

A interface *OntModel* possibilita o suporte às várias linguagens de representação de ontologias (RDFS, DAML+OIL ou OWL), para isso ela dispõe de um *profile* para cada linguagem. O *profile* lista os construtores permitidos e as URIs das classes e as propriedades de cada linguagem. Um exemplo disto pode ser visto nos parâmetros e na sintaxe da representação de classes e propriedades de URI's nas linguagens DAML e OWL, onde o URI é representado por *daml:ObjectProperty* e *owl:ObjectProperty*, respectivamente.

O *OntModel* é um aperfeiçoamento de modelo RDF oferecendo suporte aos tipos de objetos de uma ontologia:

- Classes
- Propriedades
- Instâncias

Para criação de um novo modelo na implementação da ferramenta proposta no modelo foram utilizados os métodos disponíveis na classe *ModelFactory*:

- *createOntologyModel()* : retorna um novo modelo para processar em memória as ontologias na linguagem *default* (OWL *Full*);

Ex.: OntModel m = ModelFactory.createOntologyModel();

- *createOntologyModel(java.lang.String languageURI)* : retorna um novo modelo para processar as ontologias em memória na linguagem de ontologias definida pela URI fornecida. A URI é utilizada para pesquisar os *profiles* das linguagens no *ProfileRegistry*, de acordo com o parâmetro fornecido na *languageURI*. Na Tabela 1 podem ser visualizados os parâmetros utilizados para

referenciar cada linguagem de representação de ontologias que são suportadas pela API Jena.

Ex.:

```
OntModel m = ModelFactory.createOntologyModel (ProfileRegistry.OWL_LANG);
```

- *createOntologyModel(OntModelSpec spec, Model base)* : retorna um novo modelo, construído de acordo com a especificação do modelo e inicializado com os dados do modelo base. Os componentes configuráveis de um *OntModel* são encapsulados em um objeto chamado *OntModelSpec*.
- *OntModelSpec*: é o esquema de armazenamento (ex.: MEM); a máquina de inferência (ex.: RDFS/ RULES/ TRANS); o *profile* da linguagem(ex.: OWL LITE / OWL DL / OWL / DAML / RDFS).

Ex.:

```
OntModel m = ModelFactory.CreateOntologyModel(OntModelSpec.OWL_MEM_RDFS_INF, null);
```

- *listIndividuals()*: lista todos os *individuals* que correspondem a uma classe definida na ontologia;
- *listNamedClasses()*: lista todas as classes definidas na ontologia;
- *listObjectProperties()*: lista todas as propriedades definidas no ontologia;
- *read()*: faz a leitura da ontologia em memória;
- *write()*: escreve a ontologia carregada em memória.

<i>Ontology Language</i>	URI	<i>Constant</i>
RDFS	http://www.w3.org/2000/01/rdf-schema#	RDFS_LANG
DAML+ OIL	http://www.daml.org/2001/03/daml+oil#	DAML_LANG
OWL Full	http://www.w3.org/2002/07/owl#	OWL_LANG
OWL DL	http://www.w3.org/TR/owl-features/#term_OWL_DL	OWL_DL_LANG
OWL Lite	http://www.w3.org/TR/owl-features/#term_OWL_Lite	OWL_LITE_LANG

Tabela 1 - Perfis das linguagens de representação de ontologias.

Os componentes configuráveis de um *OntModel*, possuem suas constantes de configuração, seguindo uma especificação de modelo com nomenclatura padrão.

Ex.: $XXX := DAML \mid OWL_LITE \mid OWL_DL \mid OWL \mid RDFS$

Onde:

- *XXX_MEM*: especificação de um modelo que não utiliza nenhum *reasoner*;
- *XXX_MEM_TRANS_INF*: utiliza um *reasoner* transitivo, um componente de inferência simples que produz cláusulas transitivas a partir das hierarquias de classes e propriedades da ontologia;
- *XXX_MEM_RULE_INF*: utiliza um *reasoner* baseado em regras. Utiliza as regras do *reasoner* transitivo, com implicações adicionais do RDFS (Ex: domínio e imagem);
- *XXX_MEM_RDFS_INF*: utiliza um *reasoner* baseado em regras com um conjunto de regras semânticas adequadas a linguagem especificada no XXX. É mais completo para a linguagem OWL (porém restrito a um subconjunto de OWL próximo ao OWL *Lite*).

O modelo RDF é formado por um conjunto de triplas (*statements*), onde uma tripla expressa um fato sobre um recurso. Um modelo é formado por um conjunto de sentenças. Desta forma, um recurso só existirá no modelo se existirem sentenças a respeito dele no modelo. Os principais métodos disponíveis para criação e manipulação de uma tripla, os quais foram utilizados na implementação do modelo, são os seguintes:

- *createResources()*: cria um novo recurso no modelo;
- *createProperty()*: cria uma nova propriedade no modelo;
- *addProperty()*: adiciona ao modelo uma nova propriedade criada anteriormente;
- *listProperties()*: lista as propriedades de um modelo;
- *getProperties()*: busca as propriedades de um modelo;
- *add(statement)*: adiciona novas sentenças ao modelo;
- *remove(statement)*: remove uma sentença do modelo;
- *removeAll()*: remove todas as sentenças de um modelo;

5.1.3 API de acesso ao OO

O conjunto de aplicativos OO possui uma característica muito útil mas pouco conhecida, que é a capacidade de integrar seu funcionamento com outros

aplicativos. Isto é possível através do UNO (*Universal Network Objects*), que é um modelo de componentes do OO. UNO oferece interoperabilidade entre diferentes linguagens de programação, diferentes modelos de objetos, diferentes arquiteturas e processos, em uma rede local ou mesmo através da internet. Seus componentes podem ser implementados e acessados por qualquer linguagem de programação que possua acesso aos *bindings* do UNO. Atualmente existem *bindings* para as seguintes linguagens:

- C
- C++
- Java
- Python

Na implementação do modelo proposto neste trabalho, foi utilizada para a criação do módulo de interface com o OO a versão do UNO para a linguagem Java, a qual é chamada de API Java UNO, a qual permitirá a comunicação entre o módulo de anotação e o documento OO que receberá a anotação.

A API UNO é um modelo de componentes desenvolvida pela comunidade de desenvolvedores do OO em conjunto com os laboratórios de desenvolvimento da *Sun Microsystems*. Esta API é descrita no item a seguir.

5.1.3.1 API Java UNO

Esta API permite a criação de programas usando os componentes da tecnologia UNO (*Universal Network Objects*) para suíte de aplicativos OO. Para ter acesso a esta API, é necessário além da instalação do OO, que se tenha também a instalação do SDK (*Software Development Kit*). A última versão do SDK e sua documentação estão disponíveis em [OpenOffice.org 2005].

Faz-se necessário a instalação do JDK 1.3.1 (*Java Development Kit*) ou superior. É preciso também a instalação de uma IDE (*Integrated Development Environment*), como o *NetBeans*. Outras IDEs podem ser utilizadas, mas o *NetBeans* é o que oferece uma melhor integração com o OO [Sun 2003].

É preciso configurar o OO para usar a JVM (*Java Virtual Machine*), para que o OO possa executar os componentes escritos em Java, isto pode ser feito através da

execução do arquivo *jvmsetup* disponível no diretório de instalação do OO. Também pode ser ativado através do OO, selecionando os menus “Ferramentas → Opções → OpenOffice.org → Segurança” e selecionar uma, entre as disponíveis.

O Java utiliza um *socket* TCP/IP para “conversar” com o OO. Para clientes Java, o OO precisa “escutar” as conexões TCP/IP usando um parâmetro URL de conexão. Para conseguir isto, é preciso fazer com que o OO “escute” sempre as conexões TCP/IP. Para fazer isso, é necessário abrir o arquivo `<OfficePath>/share/registry/data/org/openoffice/Setup.xcu` em editor de textos e localizar o elemento `<node oor:name="Office"/>`. Este elemento possui vários outros elementos `<prop/>`. É necessário Introduzir o elemento da Figura 24 no mesmo nível que os outros elementos existentes:

```
<prop oor:name="ooSetupConnectionURL" oor:type="xs:string">
  <value>socket,host=localhost,port=8100;urp;</value>
</prop>
```

Figura 24 - Elemento de configuração para acesso remoto ao OpenOffice.

A API Java UNO utiliza os conceitos de “*service managers*”, que são considerados como “*factories*” para criação de novos serviços. O *Service Manager* realiza a gerência de todos os serviços necessários ao acesso aos documentos do OO. Na Figura 25 pode ser visualizada a estrutura do *Service Manager*.

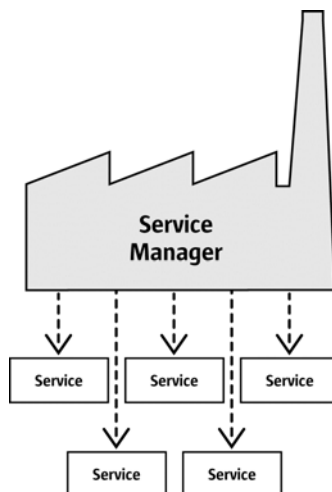


Figura 25 - Service Manager.

Para abrir a conexão com o OO e acessar as propriedades do arquivo, é necessário primeiro criar um novo serviço local, que será utilizado para manipular o arquivo que receberá a anotação semântica, este serviço é criado utilizando o método `getRemoteServiceManager()`.

Com o novo gerenciador de serviços, é então criada uma interface de acesso ao *StarOffice.ServiceManager* remoto e feita a importação das propriedades do *DefaultContext* do arquivo que receberá a anotação. O *DefaultContext* contém todas as propriedades do arquivo OO.

Após as propriedades importadas, o *ServiceManager* abre uma interface de acesso ao *XdocumentInfoSupplier*, que fornece acesso às informações descritas no *Document Info* do arquivo. O *Document Info* contém as informações disponíveis no menu “Arquivo → Propriedades”. Com acesso as propriedades do *Document Info*, então a ferramenta atualiza o valor da propriedade *Description*, com o conteúdo da anotação semântica gerada.

5.1.4 OpenOffice.org (OO)

O OO é o projeto de *software* livre, através do qual a empresa *Sun Microsystems* liberou a tecnologia de sua suíte de produtos para escritório *StarOffice*. A proposta deste projeto é desenvolver um conjunto de aplicativos multiplataforma. Seu pacote de aplicativos é composto por:

- Processador de textos;
- Planilha eletrônica;
- Gerador de apresentações;
- Programa de desenhos.

A *Sun Microsystems* participa ainda como membro da comunidade de desenvolvedores do OO. Este pacote de aplicativos é a suíte de escritórios mais difundida do tipo estação de usuário alternativa ao Office da *Microsoft*.

O OO é distribuído sob duas formas de licenciamento, LGPL (*Lesser General Public License*) e SISSL (*Sun Industry Standards Source License*), esta última permitindo a não distribuição do código-fonte das modificações realizadas e o uso de formas de licenciamento proprietário.

De acordo com as estimativas do projeto OpenOffice.org, existem atualmente em torno de 40 milhões de usuários em todo mundo [Potts 2005]. Estas estimativas foram realizadas apenas sobre o número de *downloads* da última versão do OpenOffice.org,

não considerando as cópias distribuídas através do *Linux* ou de outras formas alternativas.

Segundo sua documentação, o alcance de suas funcionalidades pode ser comparado com as conhecidas (e não livres) ferramentas compatíveis existentes no mercado. Os aplicativos estão disponíveis em diversos idiomas, inclusive o português.

O Brasil também conta com uma comunidade de desenvolvedores que colaboram no desenvolvimento desta ferramenta. Esta comunidade é liderada por Cláudio Ferreira Filho [OpenOffice.org 2005], que coordena o “OO projeto Brasil”, que tem por objetivo a união de todos os projetos e subprojetos, vinculados direta ou indiretamente com a intenção de localizar e auxiliar os usuários no Brasil.

No OO projeto Brasil, pode ser encontrado os seguintes projetos da comunidade brasileira [OpenOffice.org 2005]:

- Localização do produto OO para o português do Brasil;
- Informações em idioma Português/Brasil (*home pages* e documentação);
- Compilação da suíte na versão mais recente do OO;
- Disponibilidade de *Mirrors* para *download* das últimas versões do OO, nas mais diferentes plataformas;
- Aperfeiçoamento das ferramentas utilizadas no OO para o idioma português/Brasil;
- Implementação e aperfeiçoamento de ferramentas lingüísticas para o Brasil (dicionário/corretor ortográfico);
- Publicidade para o OO no Brasil.

Atualmente o OO projeto Brasil está disponibilizando para *download* a tradução para o português do a versão 1.1.3 do OO [OpenOffice.org 2005].

5.2 A ferramenta

O objetivo principal do trabalho proposto é um modelo pelo qual os autores de documentos de uma empresa possam efetuar a anotação semântica de seus documentos, para que posteriormente, outros usuários consigam realizar a recuperação de seus documentos através de consultas semânticas.

No modelo proposto, a anotação é realizada seguindo passos simples, não havendo a necessidade que o autor da anotação necessite de conhecimentos mais aprofundados sobre as tecnologias envolvidas no processo de anotação semântica, como por exemplo, ter que conhecer linguagens OWL, RDFS, etc. A ilustração da Figura 26 demonstra a interação entre o autor da anotação e a ferramenta de anotação semântica.

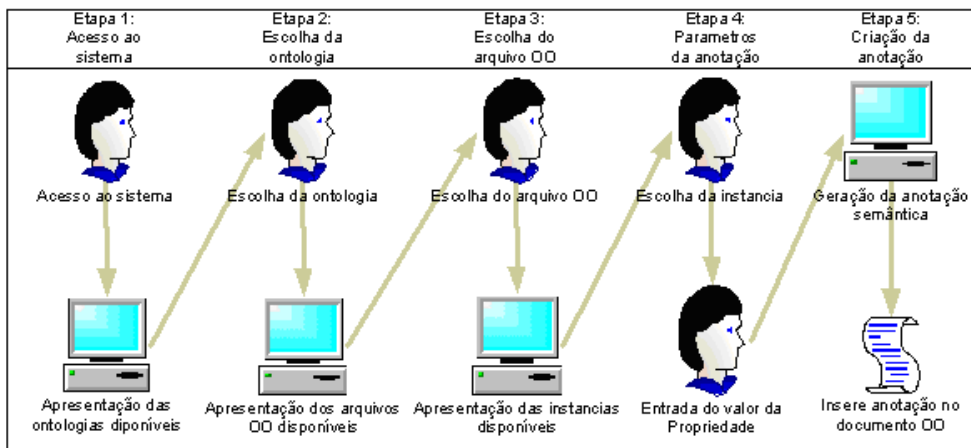


Figura 26 - Fluxo da anotação semântica.

Na Figura 27, pode ser visualizada a interface da ferramenta de anotação semântica que o autor do documento vai ter a sua disposição para realizar a sua anotação.

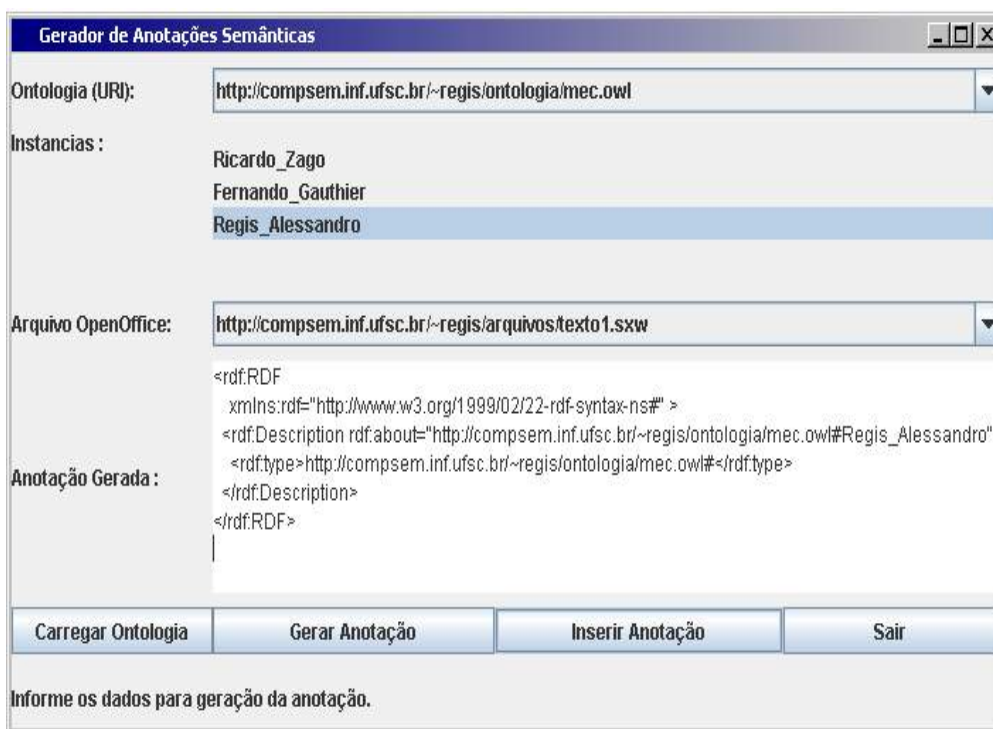


Figura 27 - Ferramenta de Anotação implementada.

Ao acessar o sistema, o autor deve fornecer os dados necessários para realizar a anotação. Na parte superior da tela é exibida uma caixa de combinação onde estão listadas as ontologias disponíveis no repositório de ontologias da empresa. O autor deve selecionar a ontologia que contém as instâncias correspondentes com seu documento e pressionar a botão “Carregar Ontologia”.

Após o autor pressionar o botão para “Carregar Ontologia”, o sistema busca a ontologia e a carrega em memória e disponibiliza as classes instanciadas a partir desta ontologia na caixa de listagem “Instâncias”.

Com as instâncias listadas na caixa de listagem “Instâncias”, o autor do documento pode percorrer a lista e selecionar as instâncias que correspondem às informações a qual o seu texto está relacionado.

Na caixa de combinação “Arquivo OO”, o sistema carrega todos os arquivos OO que estão disponíveis no repositório de documentos da empresa, aos quais o autor da anotação tem permissão de escrever. O autor deve percorrer a lista e selecionar o documento que receberá a anotação.

Após o autor informar todas as informações necessárias para a realização da anotação semântica, ele deve pressionar o botão “Gerar Anotação”. A ferramenta vai gerar a anotação semântica e mostrá-la na área de texto “Anotação Gerada”, para que o autor da anotação possa visualizá-la antes de inserir no documento, não sendo permitido que haja alteração do conteúdo desta.

Com a anotação gerada, basta pressionar o botão “Inserir Anotação”. Com base na URI do documento informado, a ferramenta vai carregá-lo em memória e inserir a anotação semântica dentro das propriedades do arquivo.

O conteúdo da anotação semântica no documento OO pode ser visualizado selecionando os menus “Arquivo → Propriedades”. As anotações foram adicionadas no campo “Comentários” da guia “Descrição”. Na Figura 28 pode ser visualizado um exemplo de arquivo OO com anotação semântica.

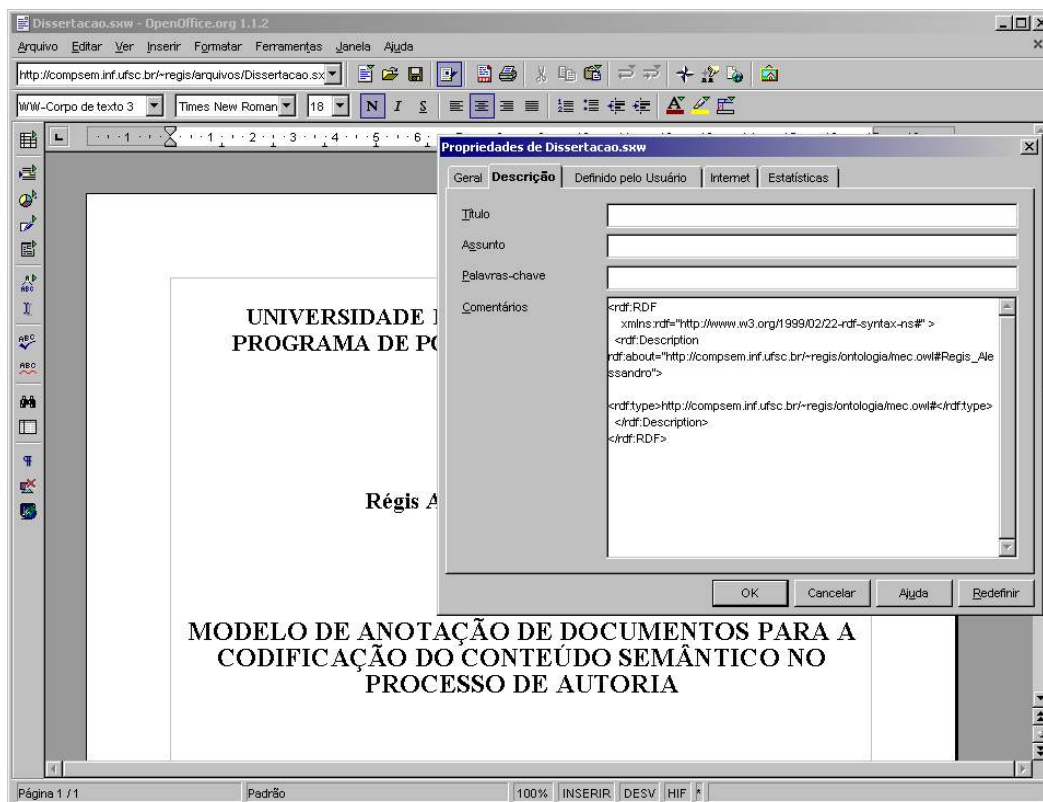


Figura 28 - Documento OO com a anotação semântica.

5.3 Testes de validação

A anotação semântica de um documento consiste de uma tripla RDF (sujeito, predicado e objeto), onde o RDF é um fundamento para o processamento de metadados. RDF provê interoperabilidade entre aplicações que trocam informação processável por máquina na *Web*. RDF enfatiza facilidades para permitir processamento automatizado de recursos na *Web*.

O período de testes consistiu em duas etapas. A primeira etapa consistiu em submeter as anotações geradas pela ferramenta, a um serviço de validação que o W3C disponibiliza. Na segunda etapa de testes, foram feitos experimentos com a ferramenta implementada para verificar suas funcionalidades, para isso foram feitos testes com algumas ontologias e documentos OpenOffice.org.

Os resultados dos testes podem ser vistos nas seções seguintes, onde na primeira seção são apresentados os resultados dos testes efetuados no serviço de validação do W3C, e na seção seguinte são mostrados os testes de bancada, realizados sobre a ferramenta.

5.3.1 Serviço de Validação do W3C

A padronização e definição do modelo RDF é uma tarefa do W3C e do seu grupo de *Semantic Web Activity* [Activity 2005], que coordena os trabalhos de padronização da *Web Semântica*.

O W3C disponibiliza a ferramenta *RDF Validation Service* [Validator 2005], onde é possível realizar testes de validação de triplas representadas em RDF. Foram realizados vários testes com as anotações geradas pela ferramenta de anotação semântica neste serviço de validação de triplas RDF. Todas as anotações geradas pela ferramenta e que foram submetidas ao validador obtiveram sucesso, comprovando a consistência das anotações geradas quando a sua sintaxe e semântica, com isto, mantendo o padrão com as regras definidas pelo W3C para recursos a serem utilizados na implementação da *Web Semântica*.

Na Figura 29 pode ser visualizado o resultado do teste de validação da anotação gerada pela ferramenta mostrada na Figura 29.

The screenshot shows the W3C RDF Validation Service results page. The browser window title is "RDF Validator Results - Microsoft Internet Explorer". The address bar shows the URL "http://www.w3.org/RDF/validator/ARPServlet". The page content includes the W3C logo, the title "RDF Validation Results", and a success message: "Your RDF document validated successfully." Below this, there is a section titled "Triples of the Data Model" with a table containing one triple. The table has columns for "Number", "Subject", "Predicate", and "Object". The triple shown is: Number 1, Subject "http://compsem.inf.ufsc.br/mec.owl#Regis_Alessandro", Predicate "http://www.w3.org/1999/02/22-rdf-syntax-ns#type", and Object "http://compsem.inf.ufsc.br/mec.owl#". Below the table, there is a section titled "The original RDF/XML document" with the following code:

```
1: <rdf:RDF
2:   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
3:   <rdf:Description rdf:about="http://compsem.inf.ufsc.br/mec.owl#Regis_Alessandro">
4:     <rdf:type>http://compsem.inf.ufsc.br/mec.owl#</rdf:type>
5:   </rdf:Description>
6: </rdf:RDF>
7:
```

 At the bottom, there is a "Feedback" section with a text input field and a "Submit problem report" button. The browser's taskbar at the bottom shows several open applications and the system clock at 20:55.

Number	Subject	Predicate	Object
1	http://compsem.inf.ufsc.br/mec.owl#Regis_Alessandro	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://compsem.inf.ufsc.br/mec.owl#

Figura 29 - RDF Validation Service [Validator 2005].

5.3.2 Aplicação de testes na ferramenta implementada

Para realização dos testes da ferramenta implementada, implementou-se alguns testes de bancada visando à determinação e análise da viabilidade da utilização do modelo de anotação.

Para testar a sua eficiência e confiabilidade na manipulação de ontologias, foram usadas algumas ontologias com instâncias já criadas. Elas foram obtidas no portal do W3C e outras construídas utilizando a ferramenta para criação de ontologias Protégé.

Durante o período de testes com as ontologias, foi verificada a eficiência da ferramenta em buscar a ontologia no repositório através de sua URI, fazer a leitura de suas classes instanciadas e disponibiliza-las ao autor da anotação para que ele possa fazer a escolha das instâncias que irão compor a anotação semântica. Em todas as ontologias testadas, a ferramenta obteve êxito em sua localização no repositório e na busca das classes que contivessem instâncias.

Nos testes com documentos OpenOffice.org, a ferramenta necessitava localizá-los no repositório, abrir o arquivo no computador, inserir a anotação nas propriedades do arquivo e salvar novamente o documento. Em todos os arquivos utilizados durante os testes, foi possível a realização da anotação e a posterior gravação das atualizações no arquivo.

5.4 Considerações Finais

A ferramenta de anotação semântica para documentos do pacote de aplicativos OpenOffice, foi implementado para a realização de testes de anotação em documentos, para por em prova os conceitos propostos no modelo criado neste trabalho.

Para alcançar este objetivo foram investigadas, na literatura e na *Web* em geral, o estado da arte das anotações, para escolha das melhores tecnologias, que poderiam dar o suporte necessário a modelagem e a construção da ferramenta proposta no modelo.

A realização dos testes com a ferramenta implementada, mostrou a sua viabilidade para a realização de anotações semânticas em documentos, em um ambiente corporativo.

6 Conclusões e trabalhos futuros

6.1 Conclusões

Este trabalho apresenta a abordagem de um modelo de anotação semântica de documentos para o conjunto de aplicativos do OO, e a sua adoção em ambientes corporativos, o que permite uma melhor gestão de seu conhecimento adquirido, pois fornece uma alternativa de solução para a atual ausência de semântica dos documentos, melhorando assim o processo de recuperação das informações por parte dos motores de busca.

A adoção de um repositório de documentos pode reduzir em muito a ambigüidade de seu armazenamento e também um melhor controle em relação à versão de seus documentos. Por isso, a adoção da *Web Semântica* em ambientes corporativos para gestão de seu conhecimento adquirido, constitui uma boa solução para seus problemas de recuperação e organização de seus documentos.

A utilização de um repositório de ontologias para o uso em seus ambientes corporativos facilita o uso da *Web Semântica* por parte dos autores de conteúdo, pois estes não precisam se preocupar com manutenção das ontologias usadas no processo de anotação semântica de documentos.

A padronização com os conceitos da *Web Semântica* propostos pelo W3C visa estabelecer um ambiente cuja flexibilidade e extensibilidade de seus componentes possam facilitar novas abordagens em trabalhos futuros. Ressalta-se que a solução proposta está fundamentada nos conceitos da *Web Semântica* recomendados pelo W3C, a fim de manter uma padronização com as demais ferramentas existentes.

Uma observação é quanto às APIs de consulta e manipulação de ontologias pesquisadas. Todas estas APIs são carentes em documentação. Embora o ambiente da API Jena seja relativamente fácil de usar, ela necessita de uma melhor documentação e de tutoriais, pois algumas características destas APIs não são fáceis de serem utilizadas e entendidas sem uma documentação mínima. Outra observação a ser feita está relacionada a falta de atualização da documentação, principalmente em relação à API Java UNO, que possibilita o acesso e a manipulação dos documentos da suíte de produtos OO.

A escolha por inserir a anotação nas propriedades do arquivo, melhorou a organização de texto, pelo fato de facilitar a visualização da anotação por parte do autor do documento, ou por qualquer outro usuário. Por isso a alternativa de inserir a anotação como um comentário anexado a partes do texto foi descartada pela dificuldade que o autor do documento teria para localizar visualmente a sua anotação, pois a ela poderia ficar livre em qualquer parte do texto. Outro problema encontrado na utilização desta alternativa é a possibilidade de o autor do documento excluir acidentalmente uma anotação gerada anteriormente, durante alguma edição do documento.

Com a implementação do modelo proposto, foi possível observar a viabilidade do seu emprego efetivo em ambientes corporativos, para a realização das anotações semânticas necessárias, para dar o suporte necessário à realização de consultas semânticas, em seus repositórios de documentos. Melhorando assim o desempenho e a precisão na recuperação das informações contidas em seus documentos.

A validação da anotação semântica gerada pela ferramenta no serviço de validação de RDFs disponível no W3C, garantiu os requisitos de padronização com as demais ferramentas disponíveis atualmente, que dão suporte a estrutura da *Web Semântica*. Isto garante que qualquer motor de busca que segue os padrões estabelecidos pelo W3C em suas consultas, e que também consiga fazer a leitura de documentos do OO, terá sucesso em suas pesquisas nos documentos anotados semanticamente pela ferramenta implementada.

A principal contribuição deste trabalho foi a construção de um modelo capaz de dar suporte a anotação semântica de documentos da suíte de produtos OO. Foram apresentadas alternativas para a construção do modelo proposto, apresentando as principais APIs disponíveis para resolver o problema de acesso a ontologias. Foram também mostradas as alternativas disponíveis para a realização do acesso a estrutura dos documentos OO. Outra contribuição deste trabalho é a ferramenta implementada para validar o modelo, a qual apresenta uma grande potencialidade para abordagens futuras.

6.2 Trabalhos futuros

A solução ora apresentada restringe-se ao problema de anotação semântica de documentos para codificação do conteúdo semântico, utilizando a linguagem para representação de ontologias OWL. Também não se preocupou com a criação de

ontologias ou de suas instâncias. A anotação semântica também ficou limitada as propriedades do arquivo.

Para dar continuidade a este trabalho, fica como sugestão para trabalhos futuros, a possibilidade de oferecer suporte a outras linguagens de representação de ontologias como SHOE, XOL, OIL, DAML ou DAML + OIL.

Outra possibilidade é adicionar recursos à ferramenta desenvolvida para possibilitar também a criação ou exclusão das instâncias de uma ontologia.

Com a estrutura do modelo definida, a possibilidade da construção da ferramenta utilizando as outras APIs disponíveis atualmente para manipulação de ontologias. Também pode ser considerada a possível implementação da ferramenta utilizando outra linguagem de programação que possuem APIs de acesso ao OO, tais como C, C++ ou Python.

Como última sugestão, a possibilidade da construção de um modelo onde a anotação semântica fique anexada a partes do texto, não ficando restrita apenas a colocação delas nas propriedades do documento.

BIBLIOGRAFIA

- [Activity 2005] **Sítio oficial - Semantic Web Activity Statement**. Disponível em <http://www.w3.org/2001/sw/Activity.html>. Acessado pela última vez em 07/06/2005.
- [Bechhofer et al 2004] Bechhofer, S., Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D., Schneider, P., Stein, L., **OWL Web Ontology Language Reference**. W3C Recommendation 10 February 2004. Disponível em <http://www.w3.org/TR/owl-ref/> . Acessado em 15/10/2004.
- [Berners-Lee et al 2001] Berners-Lee, T. Hendler, J. Lassila, O. **The Semantic Web**. Scientific American. [May, 2001]. Disponível em <<http://www.sciam.com/2001/0501issue/0501berners-lee.html>> . Acessado em 15/03/2004.
- [Bray 2004 et al] Tim, B., et al. **Extensible Markup Language (XML) 1.0 (Third Edition)**. W3C Recommendation 04 February 2004. Disponível em <http://www.w3.org/TR/2004/REC-xml-20040204/>. Acessado em 09/10/2004.
- [Carroll et al 2004] Carroll, Jeremy J; Reynolds, Dave; Dickinson, Bristol Ian. **Jena: Implementing the Semantic Web Recommendations**. *Thirteenth International World Wide Web Conference (WWW 2004), Alternate Track Papers and Posters*, pp.74-83. 2004
- [Castro 2001] Castro, E., **XML para World Wide Web**. Rio de Janeiro. Editora Campus. 2001.
- [Chandrasekaran et al 1999] Chandrasekaran, B., Josephson, J.R., Benjamins, V. R. **What Are Ontologies, and Why Do We Need Them?** *IEEE Intelligent Systems*. p. 20-26. 1999.
- [Connolly et al 2001] Connolly, Dan; Harmelen, Frank van; Horrocks, Ian; McGuinness, Deborah L; Patel-Schneider, Peter F.; Stein, Lynn Andrea. **DAML+OIL Reference Description**. Disponível em

- <http://www.w3.org/TR/daml+oil-reference>. Acessado pela ultima vez em 15/10/2004.
- [Conteras et al 2004] Contreras J., Benjamins V.R., Blázquez M., Losada S., Salla R., Sevilla, J., Navarro D., Casillas J., Mompó A., Patón D., Rodrigo L., Tena P., Martos I. **International Affairs Portal: A Semantic Web Application**. In ECAI2004 - Workshop on Application of Semantic Web Technologies to Web Communities. Valencia, Spain. 2004.
- [DAML 2004] **Sítio oficial - DAML (DARP Agent Markup Language)**. Disponível em <http://www.daml.org/>. Acessado pela última vez em 15/09/2004.
- [DeRose et al. 2001] DeRose, Steve; Maler, Eve; Daniel, Ron Jr. **XML Pointer Language (XPointer) Version 1.0**. W3C Last Call Working Draft 8 January 2001. Disponível em <http://www.w3.org/TR/WD-xptr>. Acessado pela ultima vez em 10/10/2004.
- [Fensel et al 2001] Fensel, Dieter; Harmelen, Frank van; Horrocks, Ian. **OIL: An Ontology Infrastructure for the Semantic Web**. 2001. *IEEE Intelligent Systems*, 16(2). 2001.
- [Frank & Miller 2004] Manola, F. Miller, E. **RDF Primer**. W3C Recommendation 10 February 2004.
- [Gruber 1993] Gruber, Thomas R. *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*. Padova workshop on Formal Ontology, Mar, 1993
- [Guarino 1998] Guarino, Nicola, **Formal Ontology and Information Systems**, *Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998*. Amsterdam, IOS Press, pp. 3-15.
- [Guarino & Welty 1998] Guarino, N., Welty, C., **Conceptual Modeling and Ontological Analysis**. LADSEB-CNR, 1998.
- [Guizzardi & Duarte 2002] Falbo, R.A., Guizzardi, G., Duarte, K.C. **An Ontological Approach to Domain Engineering**. Proceedings of the 14th International

Conference on Software Engineering and Knowledge Engineering. SEKE'2002, pp. 351-358, Ischia, Italy, July 2002.

[Hendler 2005] Hendler, Jim. **OWL: A Web Ontology Language**. Disponível em <http://www.w3.org/2002/Talks/www2002-ont-jh/Overview.html>. Acessado pela última vez em 10/04/2005.

[Henry et al 2004] Henry, S. David, B. Maloney, M. Mendelsohn, N. **XML Schema Part 1: Structures Second Edition**. W3C Recommendation 28 October 2004. Disponível em <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>. Acessado em 20/11/2004.

[Horridge 2004] Horridge, Matthew, **A Practical Guide To Building OWL Ontologies With The Protégé-OWL Plugin, Edition 1.0**, The University Of Manchester, 2004.

[KAON 2004] **KAON - The KARlsruhe ONtology and Semantic Web tool suite**. Disponível em <http://kaon.semanticweb.org/>. Acessado pela última vez em 10/12/2004.

[Klyne & Carroll 2004] Klyne, G. Carroll, J. **Resource Description Framework (RDF): Concepts and Abstract Syntax**. W3C Recommendation 10 February 2004. Disponível em <http://www.w3.org/TR/rdf-concepts/>. Acessado em 30/10/2004.

[Koivunen et el. 2001] Koivunen, Marja-Riitta; Swick, Ralph; Kahan, Jose; Prud'hommeaux, Eric. **Annotea: An Open RDF Infrastructure for Shared Web Annotations (2001)**. 10th *International World Wide Web Conference*. 2001.

[Koivunen & Miller 2001] Koivunen, Marja-Riitta; Miller, Eric. **W3C Semantic Web Activity**. Semantic Web Kick-off Seminar in Finland Nov 2, 2001. Disponível em <http://www.w3.org/2001/12/semweb-fin/w3csw>. Acessado em 01/10/2004.

[Jena 2005] **Sítio oficial - Jena – A Semantic Web Framework for Java**. Disponível em <http://jena.sourceforge.net/index.html>. Acessado pela última vez em 10/04/2005.

- [Luke et al 1997] Luke, Sean; Spector, Lee; Rager, David; Hendler, James. **Ontology-based Web Agents**. Proceedings of the First International Conference on Autonomous Agents (Agents'97). 1997.
- [Maler et al 2003] Maler, Eve. Grosso, Paul. Marsh, Jonathan. Walsh, Norman. **Xpointer Framework**. W3C Recommendation 25 March 2003. Disponível em <http://www.w3.org/TR/2003/REC-xptr-framework-20030325/>. Acessado pela última vez em 01/03/2005.
- [Martin et al 2001] Martin, D., et al. **Professional XML**. Editora Ciência Moderna Ltda. Rio de Janeiro.2001.
- [McGuinness & Harmelen 2002] McGuinness, D., Harmelen, F. **Web Ontology Language (OWL): Overview**. W3C Candidate Recommendation. January 22, 2002. Disponível em <http://www.ksl.stanford.edu/people/dlm/webont/OWLOverviewProblems.htm>. Acessado pela última vez em 01/11/2004.
- [Mendonça 2003] Mendonça, Eduardo. **Extração Resiliente de Dados RDF a partir de Fontes Dinâmicas em Linguagem de Marcação**. Dissertação de Mestrado. Disponível em <http://www.mcc.ufc.br/disser/EduardoEdson.pdf>. Acessado em 25/11/2004.
- [Miller et al 2004] Miller, E. Swick, R. Brickley, D. McBride, B. Hendler, J. Schreiber, G. **Semantic Web**. Disponível em <http://www.w3.org/2001/sw/>. Acessado em 13/10/2004.
- [Moura 2001] Moura, Ana Maria de Carvalho. **A Web Semântica: Fundamentos e Tecnologias**. Congreso Internacional de Ciencias de la Computación – CICC 2001. Universidad de Aquino – Bolivia. Disponível em <http://www.udabol.edu.bo/biblioteca/congresos/cicc/cicc2001/datos/Tutoriales/Tutorial4/T4.pdf>. Acessado em 05/09/2004.
- [OntoEdit 2004] **OntoEdit - Ontology Engineering Environment**. Disponível em <http://www.ontoknowledge.org/tools/ontoedit.shtml>. Acessado pela última vez em 27/11/2004.

- [Ontomat 2004] **Ontomat Homepage**. Disponível em <http://annotation.semanticweb.org/ontomat/index.html>. Acessado pela última vez em 07/12/2004.
- [Pérez & Benjamins 1999], Asunción Gómez Péres and V. Richard Benjamins, **Overview of knowledge sharing and reuse components: Ontologies and problem-solving methods**, in International Joint Conference on Artificial Intelligence(IJCAI-99), Workshop on Ontologies and Problem-Solving Methods (KRR5), V.R. Benjamins, et al., Editors. 1999: Stockolm, Sweden.
- [Pitts-Moultis & Kirk 1998] Pitts-Moultis, N. Kirk, C. **XML Black Book**. Coriolis Group Books.1998.
- [Potts 2005] Potts, Louis Souárez. **OpenOffice.org, Year Five**. OpenOffice.org Conference (OOoCon 2005). September 2005. Koper – Capodistria, Slovenia.
- [Protégé 2005] **Sítio oficial - The Protégé Ontology Editor and Knowledge Acquisition System**. Disponível em <http://protege.stanford.edu>. Acessado pela última vez em 22/04/2005.
- [Souza & Alvarenga 2004] Souza, R. Alvarenga, L. **A Web Semântica e suas contribuições para a ciência da informação**. Scielo Brasil – IBICT - Revista Ciência da Informação, vol.33, no.1, p.132-141. ISSN 0100-1965. Disponível em http://www.scielo.br/scielo.php?pid=S0100-9652004000100016&script=sci_arttext. Acessado em 23/11/2004.
- [Smith et al 2004] Smith, M., Welty, C., McGuinness, D. **OWL Web Ontology Language Guide**. W3C Recommendation. February 10, 2004. Disponível em <http://www.w3.org/TR/owl-guide/>
- [Studer et al 1998] Studer, R., Benjamins, R., Fensel, D. **Knowledge Engineering: Principles and Methods**. Data and Knowledge Engineering. Volume 25. 1998.
- [Tallis 2003] Tallis, Marcelo. **Semantic Word Processing for Content Authors**. In Knowledge Markup and Semantic Annotation Workshop (SEMANNOT 2003). At

2nd International Conference on Knowledge Capture (K-CAP 2003), October 26, 2003, Sanibel, Florida, USA.

[Tesch 2002] Tesch, J. **XML Schema**. Visual Books. Florianopolis. 2002.

[SNOBASE 2004] **Sítio oficial - Semantic Network Ontology Base - SNOBASE**. Disponível em <http://www.alphaworks.ibm.com/tech/snobase>. Acessado pela última vez em 20/11/2004.

[SOFA 2004] **SOFA - Simple Ontology Framework API**. Disponível em <https://sofa.dev.java.net/>. Acessado pela última vez em 02/12/2004.

[Sun 2003] Sun Microsystems. **OpenOffice.org 1.1 – Developer’s Guide**. 2003.

[Validator 2005] **Sítio oficial - RDF Validation Service**. Disponível em <http://www.w3.org/RDF/Validator/>. Acessado pela última vez em 01/07/2005.

[Verzulli 2001] Verzulli, J., **Using the Jena API to Process RDF**. Disponível em <http://www.xml.com/pub/a/2001/05/23/jena.html>. Acessado pela última vez em 20/10/2004.

[Zago 2005] Zago, Ricardo Flores. **Modelo de Recuperação e Indexação de Conhecimento em Documentos Corporativos Anotados Semanticamente**. Dissertação de Mestrado apresentada para conclusão do Programa de Pós-Graduação em Ciência da Computação. UFSC. 2005.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)