

UNIVERSIDADE CATÓLICA DE PELOTAS  
ESCOLA DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

# **Proposta de Criação de uma Ontologia de Ontologias**

por  
Verlani Timm Hinz

Trabalho Individual I  
TI-2006/2-010

Orientador: Prof. Dr. Luiz Antonio Moro Palazzo

Pelotas, dezembro de 2006

# AGRADECIMENTOS

Sobre tudo e todos, agradeço a Deus. Antes de mim, é por causa dEle que este trabalho foi realizado. Foi Ele quem me deu sabedoria, discernimento, forças e amparo necessário nos momentos que precisei.

Sou eternamente grata aos meus Pais, Eno e Vanda, que se enchem de alegria e orgulho com cada conquista de suas filhas.

Agradeço a cada um da minha família, que torceu e compartilhou de cada passo deste trabalho; em especial às minhas irmãs Ervanda, Vanderlisa e Elizabete.

Ao meu marido Luciano, uma pessoa muito especial, que sempre acreditou em mim, me deu forças para continuar, mesmo nas horas que não podíamos ficar juntos.

Ao orientador Prof. Dr. Luiz Antonio Moro Palazzo, o grande incentivador deste trabalho e que mais do que orientar, soube conduzir e aconselhar.

Aos amigos do mestrado, pela amizade constituída, esta turma tão especial, onde cada um deles que de uma forma ou de outra estiveram presentes e contribuíram para o sucesso deste trabalho.

À direção, coordenação dos cursos de Tecnologia, demais funcionários e alunos da Faculdade de Tecnologia Senac, pelo incentivo e carinho.

Aos colegas, também da Faculdade de Tecnologia Senac Pelotas, Edécio, Rosaura, João Ladislau e o Coordenador do Curso de Sistemas de Informação Paulo Ricardo Porto, que muito me incentivou a ingressar neste Curso.

A Capes pelo auxílio financeiro e ao Programa de Pós Graduação em Informática, desta Universidade, pelo ambiente de trabalho oferecido.

# SUMÁRIO

LISTA DE FIGURAS .....	5
LISTA DE TABELAS .....	7
LISTA DE ABREVIATURAS E SIGLAS .....	8
RESUMO.....	9
ABSTRACT .....	10
1. INTRODUÇÃO .....	11
1.1 Objetivos .....	12
1.2 Organização do texto.....	12
2. REPRESENTAÇÃO DO CONHECIMENTO.....	13
2.1 Categorias de objetos .....	13
2.2 A composição física .....	14
2.3 Formalismos de representação .....	15
2.3.1 Lógica .....	15
2.3.2 Redes semânticas .....	15
2.3.3 Frames.....	17
2.3.4 Lógica de descrição.....	18
2.3.5 Taxonomias .....	20
2.3.6 Tesouros.....	21
3. ONTOLOGIA .....	23
3.1 Definição .....	23
3.2 Vantagens do uso de ontologias .....	23
3.3 Composição .....	25
3.4 Tipos de ontologias.....	26
3.5 Metodologias.....	27
3.6 Ferramentas para construção de ontologias .....	29
3.7 Linguagens para ontologias.....	32
3.8 Interoperabilidade de ontologias.....	34
3.9 Especificações formais com gramática de grafos .....	37
3.10 Principais Vantagens da Utilização de Gramática de Grafos .....	40
3.11 Comparação com outras áreas.....	41
3.12 Áreas de aplicação das ontologias .....	42
4. ONTOLOGIA DE ONTOLOGIAS .....	44
4.1 Representação das classes.....	45
4.2 Representação das propriedades.....	47
4.3 Representação das restrições .....	50
4.3.1 Restrição de valor.....	51

4.3.2	Restrições de cardinalidade .....	52
4.4	Representação das instâncias.....	53
4.5	Representação da Ontologia de Ontologias.....	54
5.	TRABALHOS RELACIONADOS .....	56
6.	TRABALHOS FUTUROS .....	57
7.	CONSIDERAÇÕES FINAIS .....	58
8.	BIBLIOGRAFIA .....	59
	ANEXO 1 - Glossário de termos do domínio Ontologia.....	65

# LISTA DE FIGURAS

Figura 2.1: Exemplo de rede semântica .....	17
Figura 2.2: Exemplo de representação através de <i>frames</i> .....	18
Figura 2.3: Exemplo de <i>TBox</i> .....	19
Figura 2.4: Exemplo de <i>ABox</i> .....	20
Figura 2.5.: Um exemplo típico de taxonomia. ....	21
Figura 3.1: Código OWL .....	32
Figura 3.2: Exemplo do mecanismo de combinação de ontologias .....	34
Figura 3.3: Exemplo do mecanismo de integração de ontologias .....	35
Figura 3.4: Exemplo do mecanismo de alinhamento de ontologias .....	35
Figura 3.5: Exemplo do mecanismo de mapeamento de ontologias .....	36
Figura 3.6: <i>Plugin PrompTab</i> disponível no ambiente <i>Protégé</i> .....	37
Figura 3.7: Duas representações de um grafo orientado. a) um grafo orientado $G$ que tem 4 vértices e 4 arestas. b) uma representação de $G$ como matriz de adjacências. ....	39
Figura 3.8: Duas representações de um grafo orientado. a) um grafo orientado $G$ que tem 3 vértices e 3 arestas. b) uma representação de $G$ como matriz de adjacências. ....	39
Figura 3.9: Duas representações de um grafo orientado. a) um grafo orientado $G$ que tem 3 vértices e 3 arestas. b) uma representação de $G$ como matriz de adjacências. ....	40
Figura 4.1: Representação dos níveis de ontologias .....	44
Figura 4.2: Representação hierárquica de uma ontologia .....	45
Figura 4.3: Hierarquia de classes .....	46
Figura 4.4: Hierarquia de propriedades .....	47
Figura 4.5: Propriedades da classe Termo .....	47
Figura 4.6: Propriedade simétrica .....	48
Figura 4.7: Propriedade Transitiva .....	48
Figura 4.8: Propriedade funcional .....	49
Figura 4.9: Propriedade Reflexiva .....	49
Figura 4.10: Propriedade Equivalente .....	50
Figura 4.11. Definição de restrição em OWL, no ambiente <i>Protégé</i> .....	51
Figura 4.12 Cardinalidade da Classe Concreta .....	52

Figura 4.13: Definição da restrição Termo juntamente com a classe e condição necessária, no Protégé .....	53
Figura 4.14 Exemplo de Instanciação da Classe Termo.....	54
Figura 4.15: Representação parcial da ontologia de ontologias.....	55
Figura 4.16 Trecho de código OWL da Ontologia de Ontologias.....	55
Figura 5.1. A estrutura de da Ontologia de Ontologia (O3).....	56

# LISTA DE TABELAS

Tabela 3.1: Metodologias para ontologias.....	28
Tabela 3.3: Exemplos de ferramentas para construção de ontologias.....	31
Tabela 3.4: Comparativo entre as principais linguagens para representação de ontologias .....	33

# LISTA DE ABREVIATURAS E SIGLAS

DAML	DARPA Agent Markup Language
DC	Dublin Core
DL	Description Logic
ER	Entidade e Relacionamento
HTML	Hypertext Markup Language
IA	Inteligência Artificial
OIL	Ontology Inference Layer
OWL	Web Ontology Language
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RDQL	RDF Data Query Language
RQL	RDF Query Language
SBC	Sistemas Baseados em Conhecimento
SQL	Structure Query Language
URI	Uniform Resource Identifiers
W3C	World Wide Web Consortium
XML	Extensible Markup Language

# RESUMO

Nos últimos anos, o volume de informações disponíveis cresceu consideravelmente, fazendo com que a preocupação em gerenciar esse conteúdo se tornasse maior. Dentre as propostas apresentadas, um grande número utiliza ontologias como forma de organizar e categorizar dados. No entanto, a falta de um padrão para a construção de uma ontologia, tem dificultado o reuso e compartilhamento de informações entre as ontologias. O presente trabalho propõe um estudo geral sobre ontologias, suas aplicações, linguagens e metodologias, tendo como finalidade principal a proposta de criação de uma ontologia de ontologias, através da instanciação de uma meta-ontologia. Inicialmente, foi feito um estudo básico, mas aprofundando, sobre os conceitos e principalmente assuntos relacionados às ontologias e a partir disso foi elaborado um vocabulário de termos a serem empregados os quais descrevem as entidades e as relações entre elas. Foi organizado também um glossário com uma lista de termos em ordem alfabética, acompanhados das respectivas definições e um tesouro para relação de termos sinônimos e/ou antônimos. A partir disso, foi construída uma rede conceitual para representar graficamente os termos definidos e suas relações, sendo o resultado um nível superior: Ontologia de ontologias.

**Palavras-chave:** Ontologia, Vocabulário, Glossário, Tesouro.

TITLE: “**PROPOSAL OF CREATIN OF AN ONTOLOGI OF ONTOLOGIES**”

## **ABSTRACT**

In the last years, the amount of information available has increased considerably, generating the need of managing this content. Among the proposals presented, a great number uses ontologies as a means of organizing and categorizing data. However, the lack of a standard for the construction of an ontology, has made it difficult to reuse and share information among the ontologies. The present study proposes a general investigation about ontologies, their applications, languages and methodologies, being the main objective the proposal of creation of an ontology of ontologies, through of the instantiation of a meta-ontology. Inicially, a basic but deep study, was done about the concepts and issues related to ontologies and from this perspective an vocabulary of terms to be employed, which describe the entities and the relationships among them. A glossary with a list of terms in alphabetic order, followed by their respective definitions and a thesaurus for listing synonym and/or antonym terms was organized. From this, a conceptual net was created to represent graphically the terms defined and their relationships, being the result a superior level: Ontology of Ontologies.

**Keywords:** Ontology, Vocabulary, Glossary, Thesaurus.

# 1. INTRODUÇÃO

O grande volume de informações disponíveis hoje sugere a importância e a utilidade das ontologias na tarefa de organizar informações em um domínio do conhecimento. O ambiente informacional atual é mais abrangente do que há alguns anos, principalmente em função do advento da Internet e da popularização dos computadores.

Neste contexto, ontologias oferecem um meio de lidar com a representação de recursos de informação: o modelo de domínio descrito por uma ontologia pode ser usado como uma estrutura unificadora para dar semântica e uma representação comum à informação (ALMEIDA; BAX, 2003).

Ontologias têm se tornado populares, em grande parte, pelo fato de terem como objetivo promover um entendimento comum e compartilhado sobre um domínio, que pode ser comunicado entre pessoas e sistemas de aplicação (ANTONIOU; HARMELEN, 2004). Uma ontologia define um vocabulário específico usado para descrever uma certa realidade, mais um conjunto de decisões explícitas fixando de forma rigorosa o significado pretendido para o vocabulário. Uma ontologia envolve, então, um vocabulário de representação que captura os conceitos e relações em algum domínio e um conjunto de axiomas, que restringem a sua interpretação.

No entanto, tem sido cada vez mais complexo o problema de construir o dicionário dos termos e conceitos de cada área. O crescimento é grande, e o número de grupos ao redor do mundo lidando com o mesmo assunto tende a crescer na medida em que crescem os recursos disponíveis para pesquisa e desenvolvimento. Para se manter atualizado, um cientista, por exemplo, deve ler um número crescente de artigos científicos, jornais e revistas especializados.

As ontologias, no entanto, podem, também, ser definidas como depósitos de conceitos e termos em áreas de conhecimento humano, mostrando não apenas o(s) significado(s) de cada termo ou conceito, mas principalmente as relações existentes entre os conceitos. Na grande maioria das vezes, o conhecimento avança mais no sentido das relações do que na caracterização do termo propriamente dito. O desejo de avançar por essa linha é expressado fisicamente pelas obras de referência conhecidas

como os **tesauro**, que são um tipo mais simples de ontologia, mostrando um número pequeno de relações entre os conceitos.

As ontologias se constituem uma ferramenta intelectual muito utilizada desde os primórdios da ciência sendo, sendo, no entanto, realmente contextualizadas na informática, onde segundo (FARQUHAR; FIKES; RICE, 1997): Ontologias objetivam capturar o conhecimento consensual de um modo genérico, podendo ser **reusáveis** e **compartilhadas** entre aplicações (software) e por grupos de pessoas. Ontologias são normalmente construídas por um grupo de pessoas em diferentes locais.

## 1.1 Objetivos

O presente trabalho tem por objetivo principal fazer um estudo aprofundado sobre ontologias, do qual resulta a proposta de criação de uma ontologia de ontologia;

Este trabalho tem como objetivos específicos:

- Estudar linguagens e ferramentas orientadas à ontologias;
- Investigar as áreas de aplicações das ontologias;
- Estudar métodos e técnicas para a integração de ontologias;
- Estudar especificamente a questão da similaridade semântica entre ontologias;

## 1.2 Organização do texto

O texto está organizado da seguinte forma. O Capítulo 2, é referente a representação do conhecimento, suas categorias, composição física e os formalismos de representação do conhecimento, como lógica, redes semânticas, frames, lógica de descrição, taxonomias e tesauros. O Capítulo 3 aborda aos conceitos e definições de ontologia, assim como suas vantagens, sua composição, os tipos de ontologias, as metodologias, as ferramentas e as linguagens para representação de ontologias, destacando a *OWL*, que é considerado um padrão para linguagens de representação de ontologias. Além disso descreve os mecanismos de interoperabilidade utilizado para combinar ontologias, a representação através da gramática de grafos, assim como suas respectivas vantagens de uso. Uma comparação e áreas de aplicação que se referem às ontologias também são descritas neste capítulo. O capítulo 4 é referente a Ontologia de Ontologies, o qual se constitui o foco principal deste trabalho. Além disso, também são apresentados os trabalhos relacionados e futuros nos capítulos 5 e 6, respectivamente.

## 2. REPRESENTAÇÃO DO CONHECIMENTO

Antes de expor os conceitos sobre ontologias, será apresentada uma breve explicação sobre esta área que é de grande importância, a representação do conhecimento e suas características, pois ao compreendendo será mais fácil perceber o porquê da dificuldade existente em representá-lo.

Como sabemos o conhecimento não é estático, e muito menos isolado.

O conhecimento encontra-se interligado e com o passar do tempo ele, cresce, transforma-se e acumula-se. Normalmente sempre que adquirimos conhecimento relacionamo-lo com algo já existente, e sobre ele somos capazes de raciocinar e tirar conclusões.

Através disto é fácil perceber que a representação deve ser suficientemente expressiva para representar o conhecimento de maneira completa e eficiente, de forma que o sistema seja também capaz de raciocinar e tirar as suas próprias conclusões.

Desta forma é possível construir sistemas inteligentes, onde o adjetivo “inteligente” se refere à capacidade destes sistemas em encontrar informações que estão implícitas através da análise de um conhecimento representado explicitamente. Tais sistemas são caracterizados como Sistemas Baseados em Conhecimento (SBC) e envolvem dois aspectos primários:

- A caracterização precisa da base de conhecimento. É necessário definir claramente o tipo de conhecimento a ser especificado no sistema e os serviços de raciocínio disponíveis.
- A disponibilização de um *Framework*. São necessários sistemas que facilitem tanto o processo de representação quanto o de análise do conhecimento.

### 2.1 Categorias de objetos

A organização dos objetos em categorias é uma parte vital da representação do conhecimento. Apesar da interação com o mundo ocorrer no nível de objetos individuais, muito raciocínio ocorre no nível das categorias.

Por exemplo, um cliente pode ter o objetivo de comprar uma bola de futebol, em vez de uma bola de futebol da FIFA.

As categorias servem também para fazer predições sobre objetos, uma vez que estes são classificados. Podemos inferir determinados objetos a partir da percepção, inferimos a categoria a partir das propriedades capturadas do objeto, e então utiliza a informação da categoria para fazer predições sobre os objetos.

Por exemplo, se pensarmos em algo verde, de pele pintalgada, de tamanho grande, e com forma oval, podemos inferir que o objeto trata-se de uma melancia; deste, podemos ainda inferir que este seria útil para uma salada de fruta.

Existem duas maneiras para representar categorias na lógica de primeira ordem:

1. Através de predicados e objetos. Isto é, podemos usar o predicado  $BolaDeFutebol(b)$ , ou podemos considerar a categoria como um objeto, Bolas de Futebol.
2. A lógica de primeira ordem torna mais fácil o registro dos fatos sobre categorias, relacionando os objetos com categorias ou quantificando os seus membros.

## 2.2 A composição física

A idéia que um objeto pode ser parte de outro é familiar. Por exemplo: "o nariz faz parte da cabeça" e "o Brasil faz parte da América do Sul". Usamos a relação de *ParteDe* para dizer que uma coisa é parte de outra. Os objetos podem ser agrupados em hierarquias de *ParteDe*. O exemplo abaixo ilustra o que aqui foi referido:

$ParteDe(RioGrandeDoSul, Brasil)$

$ParteDe(Brasil, AméricaDoSul)$

$ParteDe(AméricaDoSul, ContinenteAmericano)$

$ParteDe(ContinenteAmericano, Terra)$

A relação de *ParteDe* é transitiva e reflexiva; isto é,

$$ParteDe(x, y) \mid ParteDe(y, z) \rightarrow ParteDe(x, z)$$

Consequentemente, nós concluímos que  $ParteDe(RioGrandeDoSul, Terra)$ .

## 2.3 Formalismos de representação

Desde cedo um dos objetivos base da Inteligência Artificial (IA), foi a representação do conhecimento humano em sistemas computacionais, isto porque ser inteligente é manipular conhecimentos, é reconhecer símbolos, é face a um problema chegar à sua solução.

No entanto, problemas de eficiência, facilidade de uso e a necessidade de expressar conhecimento incerto e incompleto levaram ao desenvolvimento de diversos tipos de formalismos de representação de conhecimento.

A seguir, apresentam-se alguns dos formalismos de representação de conhecimento mais utilizados.

- Lógica
- Redes semânticas
- *Frames*
- Taxonomias
- Tesouros
- Ontologias

### 2.3.1 Lógica

É a base para a maioria dos formalismos de representação de conhecimento, seja de forma explícita, como nos sistemas especialistas baseados na linguagem Prolog, seja disfarçada na forma de representações específicas que podem facilmente ser interpretadas como proposições ou predicados lógicos.

Uma representação em linguagem natural, por exemplo, seria :

*Pelotas é uma Cidade.*

Em lógica, seria:

*Cidade(Pelotas)*

### 2.3.2 Redes semânticas

*Rede semântica* é um nome utilizado para definir um conjunto heterogêneo de sistemas. Em última análise, a única característica comum a todos estes sistemas é a notação utilizada: uma rede semântica consiste em um conjunto de nodos conectados por um conjunto de arcos . Os nodos em geral representam objetos e os arcos relações

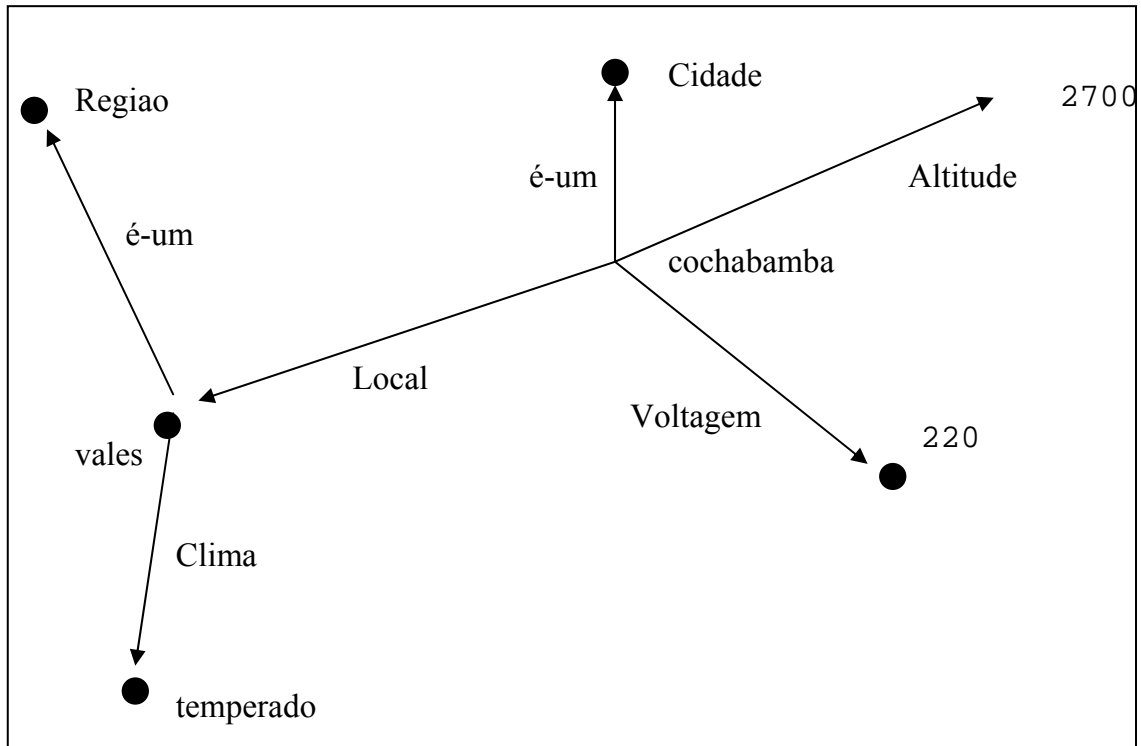
binárias entre esses objetos (é um, tipo um, tipo de, maior que) ou definem novas entidades (altura, cor). Mas os nodos também podem ser utilizados para representar predicados, classes, palavras de uma linguagem, entre outras possíveis interpretações, dependendo do sistema de redes semânticas em questão.

O objetivo de uma rede é a definição de um conceito a partir do que está relacionado (envolvente semântica).

A expansão de uma rede semântica em hierarquias representa os nós que estão mais abaixo, que são as instâncias ou indivíduos e são conectados por arcos *é-um* e os nós de níveis hierárquicos mais altos são as classes ou categorias de indivíduos.

Sendo uma simples coleção de nós e arcos, é uma estrutura semântica menos complexa que os frames, empregada como um conjunto de regras de inferência especialmente projetadas para tratar de modo correto os tipos específicos de arcos presentes na rede.

Figura 2.1, representa conceitos relacionados com regiões, climas e cidades correspondentes. Os arcos *é-um* e *é-parte* são bastante comuns em sistemas de redes semânticas. Este tipo de arco é utilizado para determinar a herança de propriedades. Os demais arcos (*local*, *clima*, *altitude*, *voltagem*, *vales*) são específicos do domínio e representam propriedades de conceitos. Esses arcos são chamados *traços*.



**Figura 2.1: Exemplo de rede semântica**

Mas há medida que aumenta o grau de formalismo conseqüentemente aumenta a potência de uma rede semântica. Isto significa que para não ter problemas de representação de caminhos e de informações conflitantes, deve-se aumentar o número de relações semânticas ou arcos, para isso utiliza-se a tripla: *objeto-atributo-valor*, que apresentam a vantagem da simplicidade de representação e da eficácia e similaridade à linguagem natural,

Durante os anos setenta, o formalismo de redes semânticas foi utilizado na implementação de diversos sistemas para a compreensão de linguagem natural.

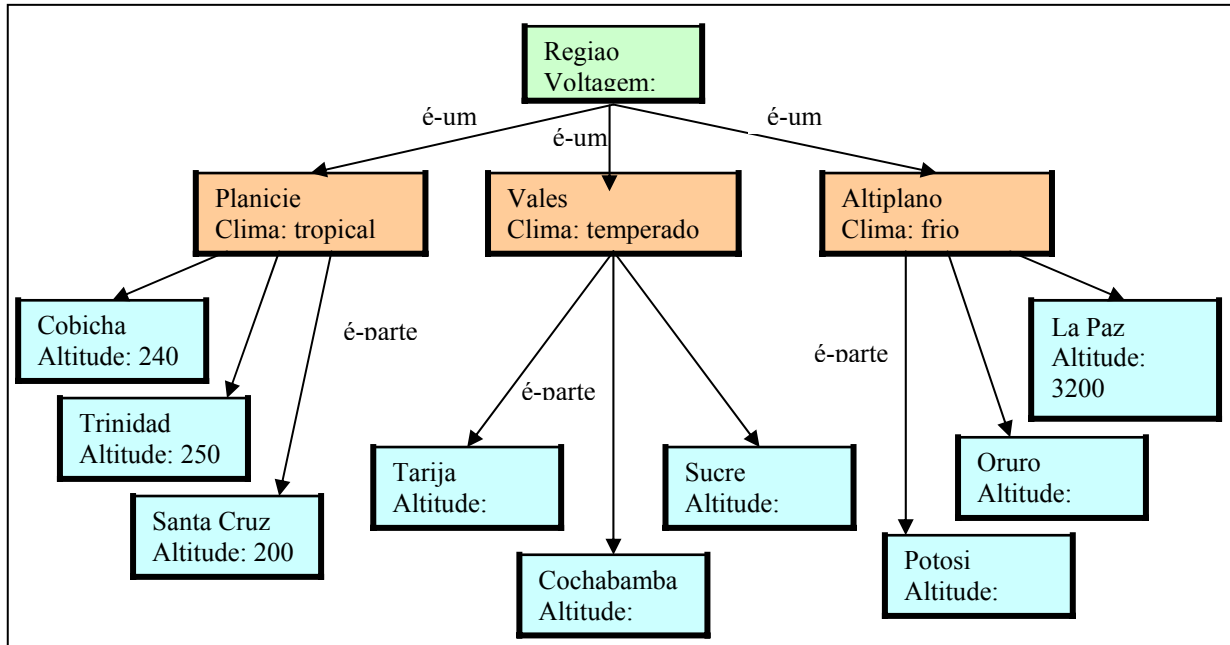
### 2.3.3 Frames

Os *frames* (ou quadros em português) foram introduzidos para permitir a expressão das estruturas internas dos objetos, mantendo a possibilidade de representar herança de propriedades como as redes semânticas.

As idéias fundamentais destes métodos foram introduzidas por Marvin Minsky (MINSKY, 1975) em seu artigo *A framework to represent knowledge*. As aplicações propostas por Minsky para o novo método foram análise de cenas, modelagem da percepção visual e compreensão de linguagem natural. No entanto, o artigo não propõe metodologia de implementação, nem definição formal do método. Desde 1975, diversos sistemas foram implementados baseados na idéia de *frames*, e diversas definições

formais foram propostas. O método de *frames* também está na origem das idéias que levaram às linguagens de programação orientadas a objetos.

A figura 2.2 apresenta um exemplo de representação de frames, em forma de hierarquia, utilizando os relacionamentos *é-um* e *é-parte*.



**Figura 2.2: Exemplo de representação através de frames**

No entanto, os *frames* são pouco adequados a novas situações. Por outro lado, a explicitação de conhecimento heurístico, comum em regras, é complexa.

### 2.3.4 Lógica de descrição

As Lógicas de Descrição ou DLs (do inglês *Description Logic*) foram projetadas como uma extensão aos frames e às redes semânticas, que não foram equipados com uma semântica formal baseada em lógica. A Lógica de Descrição foi assim chamada desde os anos 1980, anterior à isso, foi chamada (cronologicamente): sistemas terminológicos e línguas do conceito. A lógica da descrição tem-se transformado hoje uma “pedra fundamental” da web semântica para seu uso no projeto de ontologias.

Em termos gerais as lógicas de descrições, são formalismos utilizados para representar o conhecimento e raciocinar sobre ele. Isso significa que a sintaxe deste formalismo foi definida para facilitar o raciocínio, tornando-o computacionalmente menos custoso. (BAADER et al, 2003).

Embora as Lógicas de Descrições sejam menos expressivas que a lógica de primeira ordem, elas possuem grande capacidade de representação para sistemas baseados em conhecimento. Além disso, existem métodos de dedução eficientes para os serviços de raciocínio associados as DLs. Entre eles, o do Algoritmo Tableau, usado pela ferramenta Racer, uma ferramenta de raciocínio automática para as lógicas de descrição e *OWL*.

A *OWL* é baseada na Lógica de Descrições, havendo uma correspondência entre as linguagens *XML* para expressar ontologias e uma lógica de descrição. Foram definidas três sub-linguagens *OWL* cada uma correspondendo a uma lógica de descrição (*OWL Lite*, *DL* e *Full*).

A sintaxe das lógicas de descrições é formada por símbolos representando conceitos e papéis, construtores e quantificadores. Os conceitos são representações de classes, conjunto de indivíduos que apresentam as mesmas características gerais. (BAADER et al, 2003).

Uma base de conhecimento contém as informações de um determinado domínio, ou seja, ela é a representação de um conhecimento específico. Para melhor representar este conhecimento, é necessário dividir em 2 partes:

**Conhecimento Intensional:** conhecimento geral sobre o domínio do problema. São as expressões para um vocabulário genérico, onde são definidos os conceitos. Em Lógica de Descrições, este conjunto de todos os conceitos genéricos se chama se *TBox*.

**Conhecimento Extensional:** especifica um problema particular. São as expressões para atributos de objetos particulares e definem as instâncias dos conceitos, o qual em Lógica de Descrição é chamado de *ABox*.

Figura 2.3 e a Figura 2.4 representam exemplos de *TBox* e *ABox*, respectivamente, na forma estrutura e lógica.

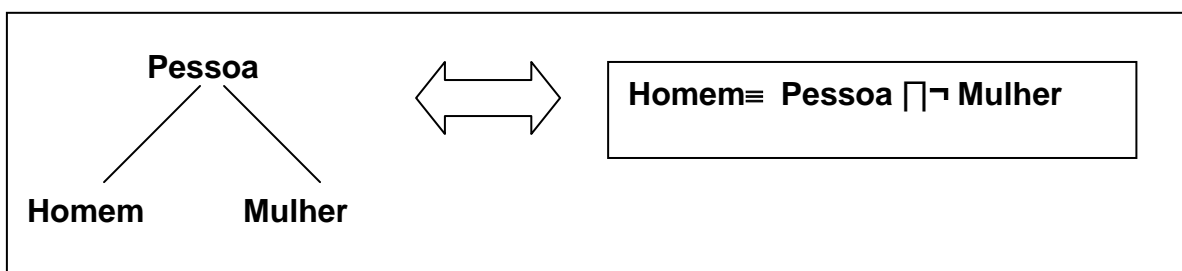
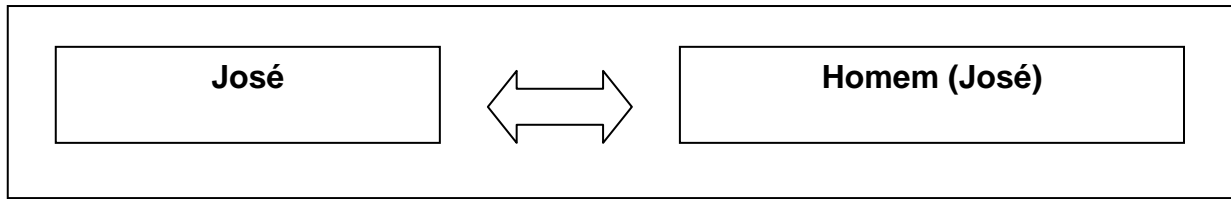


Figura 2.3: Exemplo de *TBox*



**Figura 2.4: Exemplo de ABox**

### **2.3.5 Taxonomias**

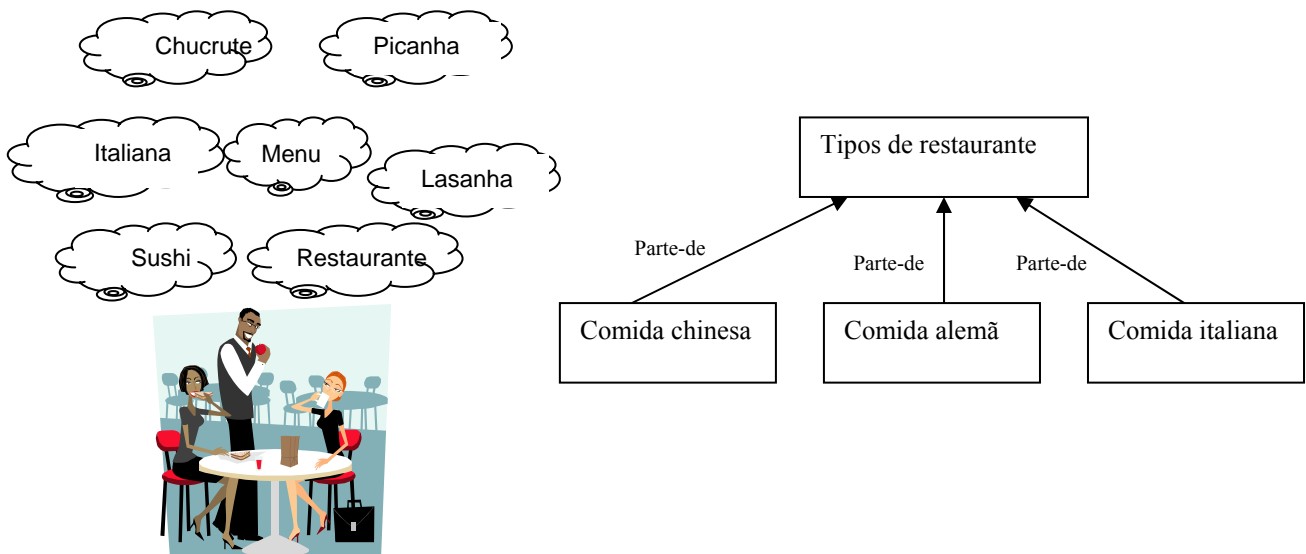
As taxonomias tiveram sua origem nas ciências da vida, sendo utilizada para nomear, descrever e classificar seres vivos.

Atualmente o termo vem sendo utilizado para designar aplicações que organizam informação em ambientes eletrônicos e virtuais.

Tem como objetivo organizar todas as entidades de um universo em uma simples hierarquia: uma grande árvore na qual cada entidade pertence apenas a uma única classe, cada espécie a um único gênero, e assim por diante. (VICKERY, 1975, p. 10).

De forma simples, uma taxonomia é uma forma de classificar ou categorizar um conjunto de coisas em forma de uma hierarquia. No mundo existem muitas taxonomias, pois as pessoas costumam classificar as coisas de forma natural, como por exemplo, tipos de restaurante, que podem ser de comida chinesa, alemã, italiana, dentre outras ou tipos de pessoas, de meios de transporte e assim por diante.

A Figura 2.5 ilustra um exemplo de termos que são categorizados e formam uma taxonomia.



**Figura 2.5.: Um exemplo típico de taxonomia.**

No entanto, as taxonomias são consideradas semanticamente muito fracas, pois não expressam de forma significativa um determinado domínio, são utilizadas como tipos de relações é sub-classe-de ou superclasse-de.

### 2.3.6 Tesouros

Um tesouro é um vocabulário controlado organizado em uma ordem conhecida e estruturado, de modo que os relacionamentos de equivalência, homográficos, hierárquicos e associativos entre termos sejam indicados claramente e identificados por indicadores padronizados dos relacionamentos (ANSI/NISO Z39.19-1993 [R1998], p. 1).

Um tesouro é um instrumento especializado e pós-coordenado de controle de vocabulário que fornece informações para indexadores e pesquisadores sobre a linguagem adotada no sistema de informação. Auxiliando com respeito a qual termo usar, o tesouro proporciona melhorias de qualidade na representação e recuperação da informação.

O controle de vocabulário é conseguido pela determinação do significado de descritores, pela ligação entre termos sinônimos através de relações de equivalência e pela solução de polissemias.

A proposta de um tesouro é para facilitar a recuperação de um documento. Um exemplo é o *WordNet* (MILLER, 1995), que organiza substantivos, advérbios e adjetivos em um conjunto de sinônimos e define relacionamento entre estes sinônimos.

Mas tanto as taxonomias como os tesouros fornecem um vocabulário de termos e relacionamentos simples entre estes termos. Conseqüentemente, as taxonomias e os tesouros estão acima de *XML*, de namespaces e de vocabulários controlados na linha da web semântica.

Entretanto, os relacionamentos que eles expressam não são tão ricos como esses fornecidos por *RDF* ou por mapas do tópico e conseqüentemente por ontologias.

O capítulo 3 aborda mais detalhes deste “novo” formalismo de representação que são as ontologias.

## **3. ONTOLOGIA**

### **3.1 Definição**

A palavra ontologia tem a sua origem nos pensamentos filosóficos de Aristóteles. A disciplina de computação resgatou este termo, inserindo-o num novo contexto, apesar de próximo do significado original. Dentre as dimensões da metafísica, a ontologia trata do ser enquanto ser. Neste sentido, apesar dos múltiplos significados do ser, ele faz referência a um único princípio, que une todos estes múltiplos significados (REALE, 2001).

De forma simples, para elaborar ontologias, definem-se categorias para as coisas que existem em um mesmo domínio. Ontologia é um "catálogo de tipos de coisas" em que se supõe existir um domínio, na perspectiva de uma pessoa que usa uma determinada linguagem (SOWA, 2000). Trata-se de "uma teoria que diz respeito a tipos de entidades e, especificamente, a tipos de entidades abstratas que são aceitas em um sistema com uma linguagem" (CORAZZON, 2002).

Muitos trabalhos sobre ontologias são encontrados na literatura e são apresentadas diversas definições sobre o termo, no entanto, (BORST, 1997) apresenta uma definição simples e completa, que será adotada neste trabalho: "Uma ontologia é uma especificação formal e explícita de uma conceitualização compartilhada". Nessa definição, "formal" significa legível para computadores; "especificação explícita" diz respeito a conceitos, propriedades, relações, funções, restrições, axiomas, explicitamente definidos; "compartilhado" quer dizer conhecimento consensual; e "conceitualização" diz respeito a um modelo abstrato de algum fenômeno do mundo real. A conceitualização é formada por um vocabulário controlado que é arranjado hierarquicamente e através de relações entre conceitos, como nas taxonomias e tesauros. Uma conceitualização é uma visão abstrata e simplificada do mundo que se deseja representar.

### **3.2 Vantagens do uso de ontologias**

Agora que já foi visto o que é uma ontologia, é importante que se destaque as vantagens de seu uso. A seguir é apresentada uma lista com as principais vantagens da utilização de ontologias na Ciência da Computação:

- Ontologias fornecem um vocabulário para representação do conhecimento. Esse vocabulário tem por trás uma conceitualização que o sustenta, evitando assim interpretações ambíguas desse vocabulário.

- Ontologias permitem o compartilhamento de conhecimento. Sendo assim, caso exista uma ontologia que modele adequadamente certo domínio de conhecimento, essa pode ser compartilhada e usada por pessoas que desenvolvam aplicações dentro desse domínio. Para exemplificar, considere que exista uma ontologia para o domínio de livrarias. Uma vez que essa ontologia está disponível, várias livrarias podem construir seus catálogos usando o vocabulário fornecido por essa ontologia sem a necessidade de refazer uma análise do domínio de livraria.

- Fornece uma descrição exata do conhecimento. Diferentemente da linguagem natural em que as palavras podem ter semântica totalmente diferente conforme o seu contexto, a ontologia por ser escrita em linguagem formal, não deixa espaço para o *gap* semântico existente na linguagem natural. Por exemplo, quando uma pessoa fala para outra a palavra “Globo” ela pode estar querendo falar a respeito de um corpo esférico, como também de um canal de televisão brasileiro. A interpretação da palavra pode ser atribuída a um conceito ou outro conforme o estado mental do indivíduo.

Porém, se há uma conceitualização comum entre essas duas pessoas a possibilidade de mal entendido diminui muito. Por exemplo, se essas pessoas concordam em uma ontologia sobre o domínio de formas geométricas, possivelmente não haverá mal entendido.

- É possível fazer o mapeamento da linguagem da ontologia sem que com isso seja alterada a sua conceitualização, ou seja, uma mesma conceitualização pode ser expressa em várias línguas.

- Pode ser possível estender o uso de uma ontologia genérica de forma a que ela se adeque a um domínio específico. Por exemplo, se alguém precisa de uma ontologia sobre bicicletas para construir uma aplicação e só encontra uma ontologia sobre o domínio genérico de veículos, pode utilizar essa ontologia estendendo-a para o domínio específico da aplicação, que no caso são de bicicletas.

Essas são as principais vantagens da utilização de ontologias. Existem muitas outras vantagens, porém a maioria é decorrente dessas.

### 3.3 Composição

Os componentes básicos de uma ontologia são classes (organizadas em uma taxonomia), relações (representam o tipo de interação entre os conceitos de um domínio), axiomas (usados para modelar sentenças sempre verdadeiras) e instâncias (utilizadas para representar elementos específicos, ou seja, os próprios dados). (GRUBER, 1996).

De forma resumida, uma ontologia é composta de: Domínio, Vocabulário, Glossário, Tesouro, Taxonomia e Rede Semântica.

O domínio define a abrangência da ontologia. Como dito em (RUSSEL; NORVIG, 2004, p. 245), o domínio de uma representação do conhecimento é uma parte do mundo sobre o qual deseja-se expressar algum conhecimento. Este domínio define os limites onde é definida toda a estrutura da ontologia.

O vocabulário é composto de classes e instâncias. As classes descrevem os conceitos dentro do domínio considerado. Estes conceitos representam um conjunto abstrato de objetos. Este conjunto considera características comuns entre estes objetos, sob um nível de abstração específico, a fim de representar os objetos através de um conceito único.

As instâncias são as ocorrências particulares do objeto em relação à classe considerada. Uma instância também descreve conceitos, mas de forma individualizada, única e concreta, fazendo referência a um objeto real (documento, página de intranet, banco de dados).

Ainda dentro dos vocabulários, as classes possuem características. Estas características são definidas pelos seus atributos ou propriedades. Os atributos possuem valor nas instâncias. As relações hierárquicas entre as classes compõem a taxonomia. Esta relação define uma forma de estruturação onde classes pai possuem classes filhos, com atributos herdados dos pais.

Finalmente, as regras desempenham um importante papel no enriquecimento semântico de uma ontologia. As regras são assertivas lógicas que impõe restrições às suas classes e aos atributos. Podem existir regras que proíbem, obrigam ou permitem valores de atributos ou instâncias de uma classe, de acordo com a regra estabelecida e a condição da regra satisfeita. Resumindo, uma ontologia é, estruturalmente, a representação de uma determinada realidade, através de uma linguagem formal, composta de vocabulário, relacionamentos e regras.

As taxonomias são compostas de termos ou conceitos sobre o universo da informação armazenada relacionados de forma hierárquica. Do mesmo modo, o tesauro define conceitos, mas permite também outros tipos de relacionamentos entre os conceitos. Utilizando taxonomias ou tesouros, o usuário poderá escolher qual a categoria de informação ele deseja acessar, podendo combiná-la com a busca por palavra chave. Este tipo de busca provê bons resultados no acesso às informações. Sua deficiência é a não disponibilidade de uma linguagem padrão que descreva a sua estrutura. Isso dificulta o compartilhamento desta estrutura por outros sistemas, a construção de ferramentas de auxílio ao desenvolvimento das estruturas e a utilização de sistemas de manipulação de regras de restrição.

### 3.4 Tipos de ontologias

As ontologias não apresentam sempre a mesma estrutura, mas existem características e componentes básicos comuns presentes em grande parte delas. Mesmo apresentando propriedades distintas, é possível identificar tipos bem definidos.

Algumas das propostas definem tipos de ontologias relacionando-as à sua função (MIZOGUCHI; VANWELKENUYSEN; YEDA, 1995), ao grau de formalismo de seu vocabulário (USCHOLD; GRUNINGER, 1996), à sua aplicação (JASPER; USCHOLD, 1999) e à estrutura e conteúdo da conceitualização (VAN-HEIJST, SCHREIBER; WIELINGA, 1997), (HAAV; LUBI, 2001). A seguir, a sintetização cada abordagem.

- **Ontologias de Representação:** definem as primitivas de representação - como *frames*, axiomas, atributos e outros – de forma declarativa.
- **Ontologias Gerais (ou de topo):** contêm definições abstratas necessárias para a compreensão de aspectos do mundo, como tempo, processos, papéis, espaço, seres, coisas, etc.
- **Ontologias centrais ou genéricas de domínio (*core ontologies*):** descrevem ramos de estudo de uma área e seus conceitos mais genéricos e abstratos.
- **Ontologias de domínio e de aplicação:** tratam de um domínio específico de uma área genérica de conhecimento, como direito tributário, microbiologia, etc.

Mesmo sem um consenso, observa-se que os tipos apresentados guardam semelhanças entre suas funções. Conhecidos os principais tipos e características, pode-se buscar ontologias existentes adequadas à utilização desejada.

### **3.5 Metodologias**

O processo de construção de ontologias ainda se encontra pouco desenvolvido. A grande maioria dos desenvolvedores usa seus próprios critérios no processo de desenvolver uma ontologia.

Devido a essa falta de uma metodologia estabelecida para o desenvolvimento, uma prática comum entre os desenvolvedores de ontologias é passar diretamente do passo de aquisição de conhecimento para o passo de implementação, o que gera os seguintes problemas como os modelos conceituais da ontologia que ficam implícitos no código da implementação, assim como gera dificuldades no desenvolvimento de ontologias complexas, pois a passagem da aquisição de conhecimento para a implementação é muito abrupta.

No entanto, algumas metodologias têm sido desenvolvidas no intuito de sistematizar a construção e a manipulação de ontologias (LÓPEZ, 1999). Existem metodologias para a construção de ontologias, para construção de ontologias em grupo, para aprendizado sobre a estrutura de ontologias e para a integração de ontologias.

Baseado nos estudos de (FARQUHAR; FIKES; RICE, 1997), apresentam-se várias metodologias para construção de ontologias.

Como exemplo são citadas: Enterprise Ontology, TOVE (Toronto Virtual Enterprise), METHONTOLOGY, On-To-Knowledge, descritos na tabela 3.1.

<b>Metodologia</b>	<b>Referência</b>	<b>Fases</b>
<b>Enterprise Ontology</b>	(USCHOLD; KING, 1995)	<ul style="list-style-type: none"> <li>• Identificar a proposta da ontologia</li> <li>• Construir a ontologia capturando, codificando e integrando conhecimento apropriado a partir de ontologias existentes</li> <li>• Avalia a ontologia</li> <li>• Documenta a ontologia</li> </ul>
<b>TOVE (Toronto Virtual Enterprise)</b>	(GRUNINGER; FOX,1995)	<ul style="list-style-type: none"> <li>• Capturar cenários de motivação</li> <li>• Formular questões de competência informal</li> <li>• Terminologia Formal</li> <li>• Formular questões em FOL</li> <li>• Especificar axiomas</li> <li>• Avaliar a ontologia</li> </ul>
<b>Methontology</b>	(GÓMEZ- PÉREZ, 1998)	<ul style="list-style-type: none"> <li>• Especificar o requisito.</li> <li>• Conceitualizar o domínio do conhecimento.</li> <li>• Formalizar o modelo conceitual em uma linguagem formal.</li> <li>• Implementar um modelo formal.</li> <li>• Fazer a manutenção de ontologias implementadas.</li> </ul>
<b>On-To- Knowledge</b>	(STAAB et al., 2001)	<ul style="list-style-type: none"> <li>• Requisitos da ontologia são capturados e especificados</li> <li>• Questões de competência são identificadas</li> <li>• Ontologias potencialmente reusadas são estudadas e uma versão “draft” da ontologia é construída</li> </ul>

**Tabela 3.1: Metodologias para ontologias**

Nenhuma das abordagens é totalmente madura se comparadas com metodologias de engenharia de software e de engenharia do conhecimento. A Methontology, recomendada pela FIPA (Foundation for Intelligent Physical Agents), atualmente encontra-se mais madura em relação aos processos de desenvolvimento de uma ontologia, pois se concentra em Especificação (uso de questões competentes),

Conceitualização (Glossário de termos, classificação de conceitos) e Implementação (codificação em uma linguagem formal).

As metodologias apresentadas possuem abordagens e características diversas. Não parece provável a unificação das propostas em uma única metodologia. Para verificar a utilidade das metodologias e compará-las, é necessário avaliar a ontologia resultante da aplicação de cada metodologia.

Segundo, [GÓMEZ-PÉREZ, 1999]:

“Apesar dos princípios de construção propostos, ainda não existem ferramentas de metodologia que guiem uma elaboração completamente padronizada, como em outros tipos de software. Em consequência disso, não há métodos sedimentados de validação, verificação, desenvolvimento e mesmo documentação de ontologias, que muito facilitaria seu reuso e aplicação. Engenharia e metodologias têm sido estudadas e propostas, visando preencher este vácuo.”

### **3.6 Ferramentas para construção de ontologias**

Além de metodologias, existem ferramentas utilizadas para a construção de uma ontologia.

Por se tratar de uma tarefa dispendiosa, qualquer apoio na construção de ontologias pode representar ganhos significativos. Exemplos de ferramentas para a construção de ontologias são apresentados a seguir, Tabela 3.2:

Ferramenta	Breve descrição
Chimaera	Sistema de software que apóia o usuário na criação e manutenção de ontologias distribuídas na Web. Suas duas principais funções são: combinação de múltiplas ontologias e diagnóstico de ontologias individuais ou múltiplas. Apóia o usuário nas tarefas de carregamento de bases de conhecimento em diferentes formatos, reorganização taxionômica, resolução de conflitos com nomes, busca de ontologias, edição de termos, etc. Sua intenção original era combinar fragmentos de bases de conhecimento (MCGUINNESS et al, 2000).
CODEA (Conceptually Oriented Description Environment)	Ferramenta de propósito geral que possui diferentes modos de herança e inferência, uma interface gráfica de fácil uso, um modo de hipertexto para navegação e utilitários para leitura de documentos e gerenciamento léxico (SKUCE; IKARUS, 1995).
JOE (Java Ontology Editor)	Ferramenta para construção e visualização de ontologias. Proporciona gerenciamento do conhecimento em ambientes abertos, heterogêneos e com diversos usuários. As ontologias são visualizadas como um diagrama entidade-relacionamento como o gerenciador de arquivos do <i>MS-Windows</i> ou como uma estrutura em árvore (MAEDCHE; VOLZ, 2001).
OilEd	Editor simples, o “NotePad” dos editores de ontologias. Ele utiliza as linguagens <i>DAML+OIL</i> . Sua intenção inicial é prover um simples editor que facilite o uso e estimule o interesse na linguagem <i>OIL</i> . Sua versão atual não provê um ambiente completo de desenvolvimento de ontologias – não suporta o desenvolvimento destas em larga escala, migração e integração, versionamento, argumentação e muitas outras tarefas envolvidas no seu processo de construção (HORROCKS; SATTLER; TOBIES, 2000).

Ferramenta	Breve descrição
OntoEdit	É um ambiente gráfico para edição de ontologias que permite inspeção, navegação, codificação e alteração de ontologias. O modelo conceitual é armazenado usando um modelo de ontologia que pode ser mapeado em diferentes linguagens de representação. As ontologias são armazenadas em bancos relacionais e podem ser implementadas em <i>XML</i> , <i>FLogic</i> , <i>RDF(S)</i> e <i>DAML+OIL</i> (MAEDCHE; VOLZ, 2001).
Ontolíngua	Conjunto de serviços que possibilitam a construção de ontologias compartilhadas entre grupos. Permite acesso a uma biblioteca de ontologias, tradutores para linguagem e um editor para criar e navegar pela ontologia. Editores remotos podem editar ontologias usando protocolos. (FARQUHAR; FIKES; RICE, 1997).
Protegé	É um ambiente interativo para projeto de ontologias de código aberto, que oferece uma interface gráfica para edição de ontologias e uma arquitetura para criação de ferramentas baseadas em conhecimento. A arquitetura é modulada e permite a inserção de novos recursos (NOY; FERGESON; MUSEN, 2000).
WebODE	Ambiente para engenharia ontológica que dá suporte à maioria das atividades de desenvolvimento de ontologias. A integração com outros sistemas é possível, importando e exportando ontologias de linguagens de marcação (ARPIREZ, 2001).
WeOnto	Ferramenta que possibilita a navegação, criação e edição de ontologias, representadas na linguagem de modelagem <i>OCML</i> . Permite o gerenciamento de ontologias por interface gráfica, inspeção de elementos, verificação da consistência da herança e trabalho cooperativo. Possui uma biblioteca com mais de cem ontologias (DOMINGUE; MOTTA; CORCHO, 1999).

**Tabela 3.2: Exemplos de ferramentas para construção de ontologias**

### 3.7 Linguagens para ontologias

Diversas linguagens foram desenvolvidas para representação de ontologias. Alguns exemplos são *XOL*, *SHOE*, *DAML*, *RDF/RDF(S)*, *OIL* e *OWL*.

Entre essas linguagens, a *OWL (Ontology Web Language)* é a mais promissora e sofisticada que suas linguagens bases porque, por exemplo, seus conceitos podem ser especificados por combinações lógicas como interseção, união, ou complementos de outros conceitos, ou por enumerações de objetos especiais. *OWL* pode indicar que uma propriedade é transitiva, simétrica, funcional ou inversa, em relação a uma outra propriedade, quais indivíduos, i.e., instâncias, pertencem à quais conceitos, que conceitos e propriedades podem usar indicações de equivalência e disjunção, que indicações de igualdade e diferença podem ser utilizadas entre indivíduos, entre outras informações fundamentais para fornecer o suporte semântico necessário para os primeiros passos da *Web Semântica*.

Foi desenvolvida baseando-se nas linguagens *DAML* e *OIL* e construída em cima da arquitetura *XML* e *RDF*. A *OWL* foi proposta como padrão pelo *W3C3*, agregando diversos pontos positivos das linguagens anteriores, e está sendo utilizada pela *WEB Semântica* (BERNERS-LEE; HENDLER; LASSILA, 2001).

A seguir, na figura 3.1, é apresentado um trecho de código *OWL* de uma ontologia de pizzas (exemplo utilizado no tutorial do *Protege*, que pode ser obtido em ([PROTEGÉ, 2005]. Disponível em: <http://protege.stanford.edu>. Acesso em: 28 novembro de 2006.

```

</owl:Ontology>
<owl:Class rdf:ID="CoberturaDeTomate">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:FunctionalProperty rdf:ID="temTempero"/>
      </owl:onProperty>
      <owl:someValuesFrom>
        <owl:Class rdf:ID="Suave"/>
      </owl:someValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

Figura 3.1: Código OWL

A Tabela 3.3, mostra uma comparação entre principais linguagens para ontologias.

	RDF	OIL	DAML-OIL	OWL	CYC
<b>Objetivo</b>	Tornar possível a especificação de regras semânticas para dados baseados em XML de maneira padronizável e interoperável Definir um mecanismo independente de domínio Definir um mecanismo para descrever recursos	Representação de uma semântica de informação acessível para a máquina na web.	Foi construída para ser uma linguagem de marcação semântica para recursos web (facilitar a representação dos conceitos de web semântica, descrevendo a estrutura de um domínio, ser compatível com padrões web existentes, fácil compreensão e uso, ter uma semântica formal associada, possuir um relativo poder de expressão)	É um documento XML, RDF, RDFS Originado de vários projetos: DARPA (DAML) + EU Project (OIL)	Proposta de uma ontologia geral com possibilidade de tratamento de contextos específicos (microteorias). Manipulação de grandes quantidades de conceitos.
<b>Primitivas de Representação</b>	Especificação composta por rdfls e rdfs Representação por conjunto de triplas: recurso propriedade valor Objetos identificados por URI Elemento base é a sentença representada por grafos dirigidos. Usa namespaces para associar propriedades, containers, statements sobre statements.		Header, class element, property elements, instances	Parecido com RDF porém OWL é uma linguagem com novas estruturas específicas para ontologias Enriquece o rdfs adicionando cardinalidade equivalência e importação de recursos, restrições	Conceitos, asserções, predicados, conectores, sentenças (Cycl baseadas em LISP)
<b>Propriedades lógicas</b>			Inverso, equivalência, diferença, disjunção, versão e importação.	Baseado em classes e tudo é derivado de OWL: thing	
<b>Vantagens</b>	RDFs define o significado da estrutura usando os tags estabelecidos em seu modelo Provê tags para definir classes, subclasses e propriedades Permite restrições de domínio e de valores (range) e, propriedades.	Consegue promover pesquisas em torno do conceito semântico contido das informações das paginas web e não apenas através das palavras-chave.	Restrições de cardinalidade, tipos de dados, classes definidas e numerações, equivalência, extensibilidade, semântica formal, herança, inferência.	Subdividido em 3 sub-linguagens: lite, DL e FULL	Representação e tratamento de grandes ontologias suporte para inferência.

**Tabela 3.3: Comparativo entre as principais linguagens para representação de ontologias**

### 3.8 Interoperabilidade de ontologias

Abaixo são destacados alguns mecanismos que podem ser usados para propiciar a compatibilidade de ontologias:

**Combinação de ontologias:** tem-se como resultado a versão das ontologias originais combinadas em uma ontologia única com todos seus termos juntos, sem a definição clara de suas origens. Normalmente as ontologias originais descrevem domínios similares ou de sobreposição (NOY; MUSEN; SMART, 1999).

A Figura 3.2 ilustra um exemplo de combinação de ontologias, onde os dois conceitos compatíveis, por exemplo **carro** da ontologia **O1** e **veículo** da ontologia **O2**, são combinados, i.e., unidos, na ontologia única **O**.

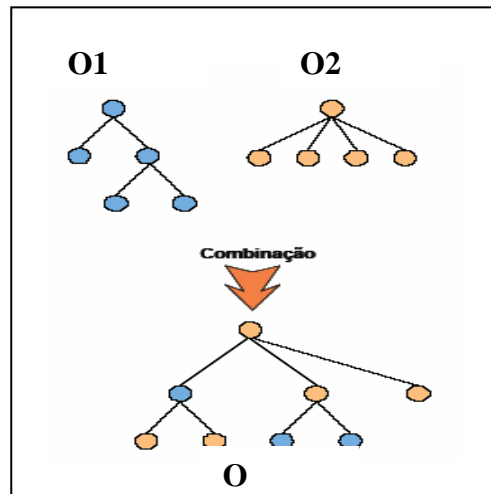


Figura 3.2: Exemplo do mecanismo de combinação de ontologias

**Integração de ontologias:** tem-se como resultado uma ontologia única criada pela montagem, extensão, especialização ou adaptação de outras ontologias que tratam não necessariamente do mesmo assunto. Na integração de ontologias é possível identificar as regiões que foram criadas a partir das ontologias originais (PINTO; GOMEZ-PEREZ; MARTINS, 1999).

A Figura 3.3 ilustra um exemplo de integração de ontologias, onde os conceitos **carro** de **O1**, **veículo** de **O2** e **casa** de **O3** são integrados, i.e., unidos, na ontologia única **O**.

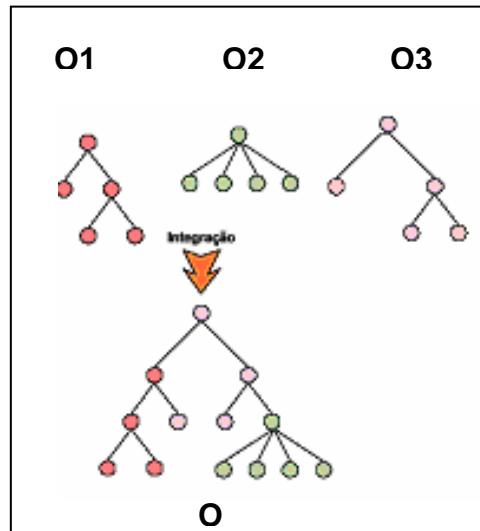


Figura 3.3: Exemplo do mecanismo de integração de ontologias

Apesar do resultado final, tanto da combinação quanto da integração de ontologias ser uma ontologia única constituída pela união dos termos das ontologias originais, a principal diferença entre estes dois mecanismos é que no primeiro as ontologias tratam do mesmo assunto, o que não acontece necessariamente no segundo.

**Alinhamento de ontologias:** tem-se como resultado as duas ontologias originais separadas, mas com as ligações estabelecidas entre elas, permitindo que as ontologias alinhadas reusem as informações uma das outras. O alinhamento normalmente é realizado quando as ontologias são de domínios complementares (NOY; MUSEN; SMART, 1999).

A Figura 3.4 ilustra um exemplo de alinhamento de ontologias, onde a ontologia **O1** se refere a **carros**, a ontologia **O2** se refere a **veículos** e a **O3** a **meios de transporte**.

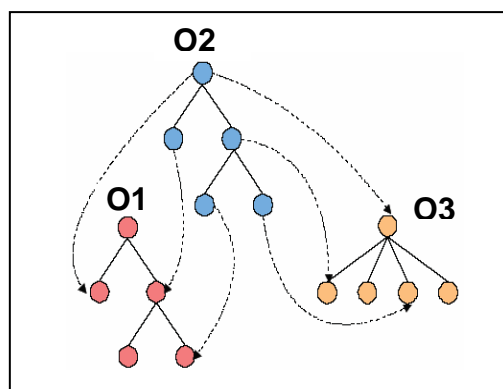
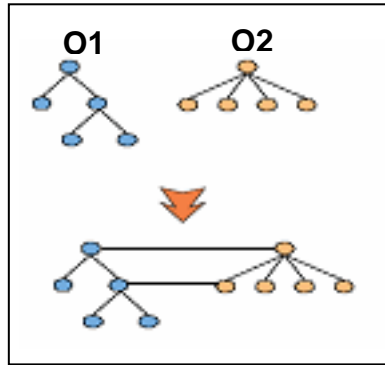


Figura 3.4: Exemplo do mecanismo de alinhamento de ontologias

**Mapeamento de ontologias:** tem-se como resultado uma estrutura formal que contém expressões que fazem a ligação de conceitos de um modelo em conceitos de um segundo modelo. Este mapeamento pode ser usado para transferir instâncias de dados, esquemas de integração, esquemas de combinação e tarefas similares (NOY; MUSEN, 2003).

A Figura 3.5 ilustra um exemplo de mapeamento de ontologias, onde os conceitos **carro** de **O1** e **veículo** de **O2** são mapeados em expressões formais.



**Figura 3.5: Exemplo do mecanismo de mapeamento de ontologias**

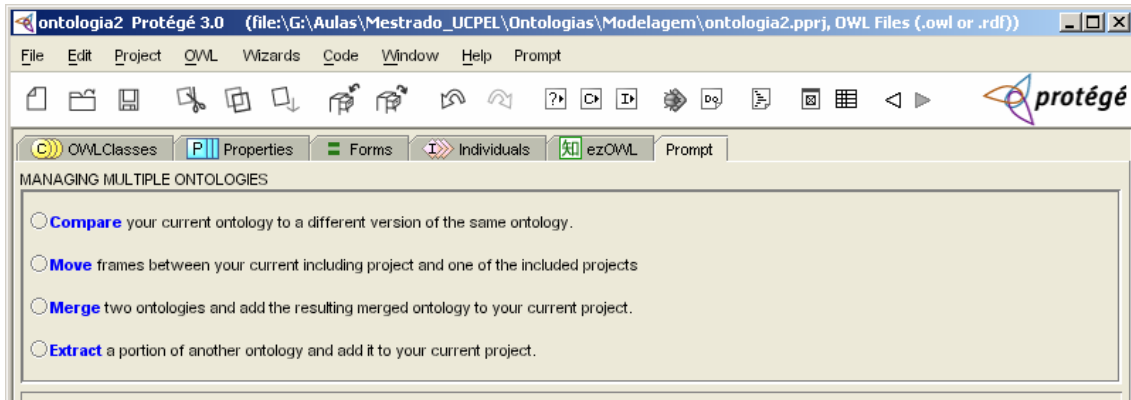
A fim de minimizarem possíveis problemas nos mecanismos de interoperabilidades citados acima, devem ser utilizados bons algoritmos de identificação, principalmente no que diz respeito a problemas de identificação dos termos a serem alinhados em ontologias comparadas, como: presença de inconsistências encontradas nas comparações dos termos de ontologias; falta de completude no alinhamento de todos seus possíveis termos e risco de inviabilidade da estratégia frente às inconsistências existentes.

O processo de alinhamento de ontologias pode ser realizado de forma automática, semi-automática, onde há a necessidade de intervenção humana em algumas etapas para a tomada de decisão, ou até mesmo manual.

Atualmente, apesar dos trabalhos realizados para garantir mecanismos que suportem a interoperabilidade de ontologias, não foi encontrada solução alguma para o alinhamento de ontologias, que esteja totalmente de acordo com os requisitos da Web Semântica. No entanto, boas idéias surgiram para elaborar uma estratégia que esteja em conformidade com tais requisitos.

Dentre elas, podemos citar o conjunto de ferramentas PROMPT apresentado em (NOY; MUSEN, 2003). Dentre as ferramentas deste conjunto, citamos: iPROMPT, uma ferramenta interativa para combinação de ontologias; AnchorPROMPT, uma

ferramenta automática baseada em grafos para alinhamento de ontologias; PROMPTFactor, uma ferramenta para extração de partes de ontologias e PROMPTDiff, uma ferramenta para identificação de diferenças entre duas versões da mesma ontologia. Além disso, o Protegé, possui um plugin chamado *PromptTab* onde se é possível aplicar mecanismos de interoperabilidade, conforme demonstra a figura



**Figura 3.6: Plugin PromptTab disponível no ambiente Protegé**

### 3.9 Especificações formais com gramática de grafos

A gramática de grafos é um formalismo de especificação na qual os estados de um sistema são representados por grafos e as transições são especificadas por regras. Neste formalismo os estados de um sistema são descritos através de estruturas algébricas (grafos que podem ou não ter atributos especificados como tipos de dados abstratos) e o comportamento do sistema é determinado operacionalmente por mudanças de estado.

Os grafos são meios muito naturais para explicar situações complexas em um nível intuitivo. A idéia básica da gramática de grafos é análoga à da gramática de Chomsky. A noção resultante da gramática de grafos generaliza a da gramática de Chomsky. A composição da gramática de grafos parte de um Grafo Tipo e de um Grafo Inicial. Um Grafo Tipo descreve as possíveis ocorrências em um grafo. É uma forma simplificada, porém bastante representativa e significativa que substitui um provável grande conjunto de regras que controlariam o grafo. Um Grafo Inicial apresenta a primeira instância do grafo, nela está representado o estado inicial do sistema aguardando a aplicação de regras que o transformarão.

Ao contrário das regras de Chomsky, uma regra de grafo  $r: L \rightarrow R$  não só consiste do grafo  $L$  (lado à esquerda) e  $R$  (lado à direita), mas tem também uma parte adicional: um grafo parcial (morphism  $r$ ) mapeando extremidades e vértices em  $L$  para extremidades e vértices em  $R$  de um modo compatível. Assim, uma gramática de grafo

especifica um sistema em termos de estados – modelado por grafos – e mudanças de estados – modelados por derivações. A seguinte interpretação operacional de uma regra  $r: L \rightarrow R$  provê a base para esta aproximação de especificação:

- Itens em  $L$  que não tem uma imagem em  $R$  são apagados;
- Itens em  $L$  que não são mapeados para  $R$  são preservados;
- Itens em  $R$  que não tem pré-imagem em  $L$  são criados.

Em lugar de usar grafos evidentes com vértices e extremidades, normalmente usa-se algum tipo de mecanismo dentro dos grafos, como tipos e atributos dos grafos. Aqui se usa grafo rotulado, isto é, cada vértice/extremidade tem um rótulo de algum alfabeto de rótulos.

O comportamento operacional de um sistema descrito por uma gramática de grafo é representado através da aplicação das regras de gramática de grafos para os grafos atuais. A aplicação de uma regra para um grafo atual, chamado “passo de derivação”, é possível existir para uma ocorrência do lado à esquerda desta regra no grafo atual. Esta ocorrência, chamada “partida”, é um morfismo de grafo total porque um espera intuitivamente que todos os elementos do lado à esquerda estejam presentes ao grafo atual para aplicar a regra.

A semântica seqüencial de uma gramática de grafos  $GG$  é determinada por todas as seqüências de passos de derivação que usam as regras de  $GG$ , começando com o grafo inicial  $GG$ , e na qual o grafo de saída de um passo é o grafo de entrada do seguinte.

Usando uma pura regra de formalismo podem-se especificar facilmente como os grafos serão transformados. Entretanto, para a especificação de quando estas transformações acontecerão, existem restrições à aplicação de condições positivas: são os vértices e as extremidades especificadas no lado à esquerda da regra que estão presentes no grafo atual, a regra pode ser aplicada se uma determinada condição for satisfeita.

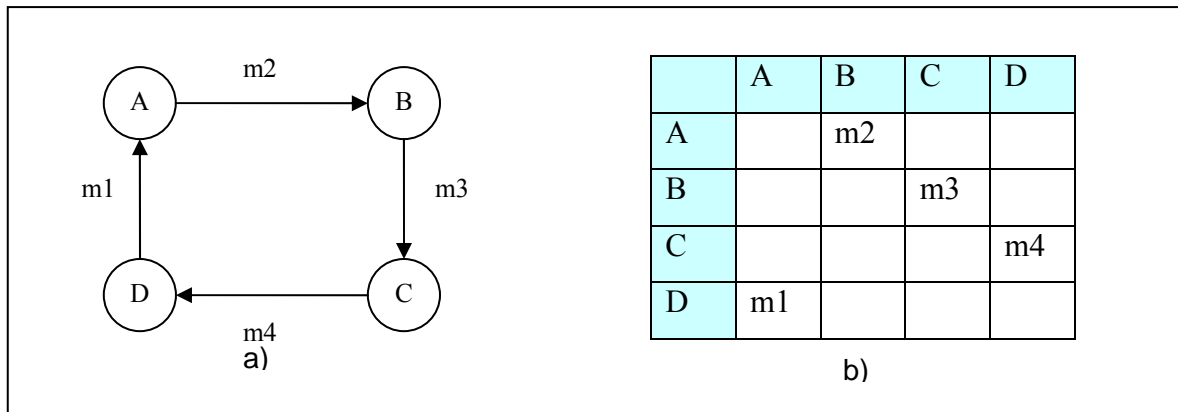
Os modelos semânticos mais usados para gramática de grafos são: linguagem semântica (conjunto de grafos gerados), semântica seqüencial (seqüência de transformações) e semântica concorrente (ordens parciais de transformações).

Uma matriz de adjacências é uma das formas de se representar um grafo.

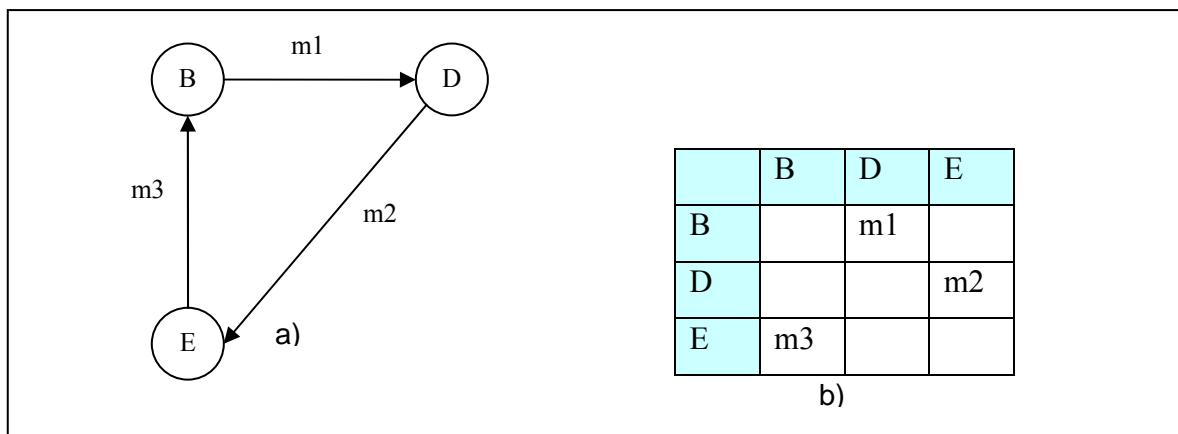
Dado um grafo  $G$  com  $n$  vértices podemos representá-lo em uma matriz  $n \times n$   $M$ . A definição precisa das entradas da matriz varia de acordo com as propriedades do grafo que se deseja representar, porém de forma geral o valor  $m_{ij}$  guarda informações

sobre como os vértices  $v_i$  e  $v_j$  estão relacionados (isto é, informações sobre a adjacência de  $v_i$  e  $v_j$ ).

A Figura 3.7 e a figura 3.7 são exemplos de grafos representados através das matriz de adjacências correspondentes.

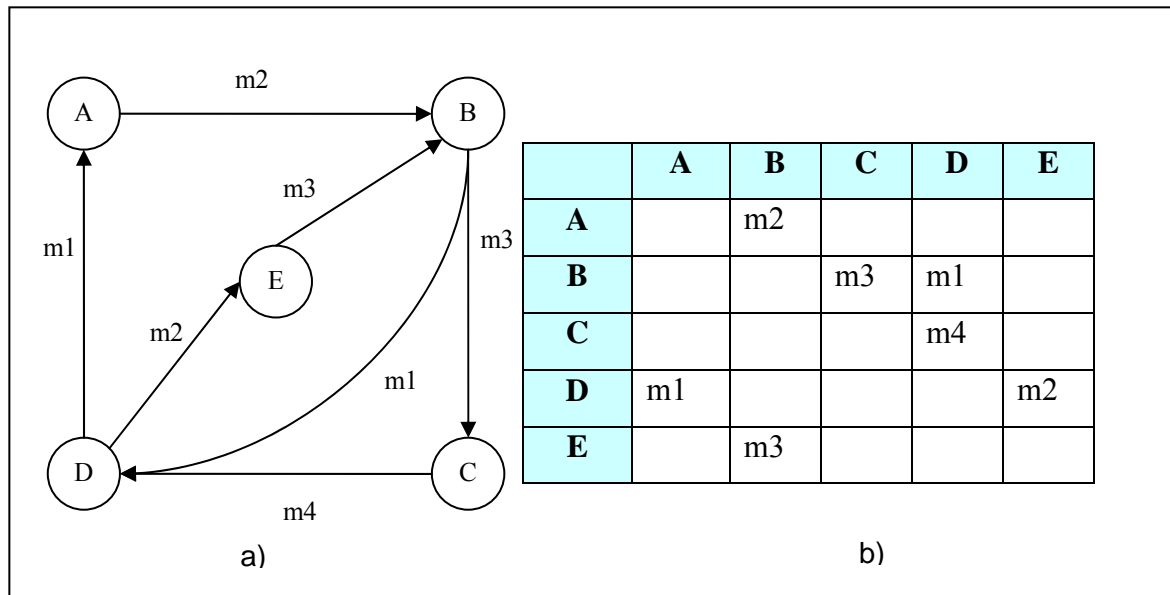


**Figura 3.7:** Duas representações de um grafo orientado. a) um grafo orientado  $G$  que tem 4 vértices e 4 arestas. b) uma representação de  $G$  como matriz de adjacências.



**Figura 3.8:** Duas representações de um grafo orientado. a) um grafo orientado  $G$  que tem 3 vértices e 3 arestas. b) uma representação de  $G$  como matriz de adjacências.

Fazendo-se a soma para um mesmo relacionamento dos dois grafos, obtemos um grafo contendo as 2 representações, conforme Figura 3.9.



**Figura 3.9:** Duas representações de um grafo orientado. a) um grafo orientado  $G$  que tem 3 vértices e 3 arestas. b) uma representação de  $G$  como matriz de adjacências.

Para representar um grafo não direcionado simples e sem pesos nas arestas, basta que as entradas  $m_{ij}$  da matriz  $M$  contenham 1 se  $v_i$  e  $v_j$  são adjacentes e 0 caso contrário. Se as arestas do grafo tiverem pesos,  $m_{ij}$  pode conter, ao invés de 1 quando houver uma aresta entre  $v_i$  e  $v_j$ , o peso dessa mesma aresta.

Uma das características da matriz de adjacência quando o grafo não é orientado é a simetria encontrada na "diagonal". É interessante que se lermos uma coluna de índice  $v$  ou uma linha de índice  $v$  vamos encontrar a mesma coisa.

### 3.10 Principais Vantagens da Utilização de Gramática de Grafos

A seguir, serão apresentadas, aquelas que atualmente, são consideradas as principais vantagens na utilização da Gramática de Grafos no desenvolvimento de sistemas:

- **Representação dos Dados:** Grafos provêm uma representação de dados expressiva e versátil de um sistema. Do ponto de vista da interação entre o usuário e o modelo do sistema pode-se observar que os grafos podem ser, até certo ponto, intuitivos e, em geral, de fácil compreensão por parte dos usuários. A rescrita de grafos também possui a característica de proporcionar uma representação de alto nível da solução de um problema computacional, qualquer que este seja.

- **Rescrita de Grafos:** Seguindo o tema anterior, outra vantagem da modelagem de um sistema através da rescrita de grafos é a possibilidade de simulação

(com o auxílio de ferramentas) o que proporciona além de sua verificação formal, a validação do sistema, ou seja, a avaliação de sua conformidade com os requisitos funcionais.

- **Incremento na Qualidade e Produtividade:** Também, as gramáticas de grafos proporcionam grande incremento na qualidade e produtividade na fase de manutenção de um sistema, em função da total previsibilidade do comportamento de um sistema em caso de alteração em seus requisitos. Esse fator é de grande importância se for considerado que a manutenção representa 60% da vida útil de um sistema.

- **Utilidade na Modelagem de Sistemas:** A utilização de gramática de grafos como método para a modelagem de sistemas, pela sua flexibilidade e representação, pode ser uma poderosa ferramenta de modelagem de diversos aspectos a serem representados durante o desenvolvimento.

A seção seguinte apresenta uma pesquisa sobre a utilização de ontologias em outras áreas, além de repositórios e ontologias conhecidas.

### 3.11 Comparação com outras áreas

A área de ontologias começou a interessar pesquisadores de Engenharia de Software nas décadas de 80 e 90, quando a especificação de software se tornou algo mais conceitual e declarativo.

O uso intensificado de tecnologias orientadas a objetos e da linguagem diagramática de modelagem unificada (UML), faz parte do cotidiano da maioria dos desenvolvedores de software, principalmente depois foi adotada como padrão em modelagem pela OMG (Object Management Group, Grupo de Gerenciamento de Objetos) [OMG 2003], organismo que recomenda linguagens e protocolos a serem adotados pelas empresas preocupadas em elaborar software orientado a objetos.

A integração entre ontologias e UML vem ocorrendo gradativamente, isto é uma decorrência de que ambas tratam de modelagem, sendo que a UML é mais voltada à modelagem estrutural e comportamental e ontologias à modelagem conceitual.

Já existe uma comunidade de pesquisadores e desenvolvedores que empregam ferramentas de ontologias para efetuar a modelagem em UML, tornando especificações sobre domínios de problemas reusáveis mesmo fora da comunidade de engenharia de software.

Outra área bastante envolvida no uso de ontologias é a área de Banco de Dados, que as utiliza para fazer a integração entre recursos, provendo padrões de

formato e mapeamento para intercâmbio de informações entre sistemas heterogêneos, que podem possuir diferentes modelos conceituais mesmo quando tratam de um mesmo domínio.

No entanto, as ontologias são semanticamente mais ricas que esquemas conceituais de bancos de dados e desta forma mais perto do modelo de conceito de banco de dados. Ontologias podem ser úteis para evitar a construção de esquemas de dados com pouco significado e consultas. Por exemplo, um determinado dado pode ser representado diferentemente em diversos bancos de dados, mas o conceito é único, no mínimo do ponto de vista de uma comunidade.

Na área de biologia, por exemplo, as ontologias são de fundamental importância, pois uma modelagem conceitual tradicional não consegue representar de forma adequada as informações biológicas. Elas estão relacionadas com a natureza dos dados biológicos e os dados biológicos, sendo que estes dados biológicos são extensos e diversos cobrindo várias áreas de conhecimento, assim como existem também diversos conceitos similares e que são difíceis de modelar.

Alguns problemas principais encontrados são:

Termos que tem o mesmo nome, mas diferentes semânticas: por exemplo a palavra colônia em Zoologia é usada para representar um grupo de animais da mesma espécie que vive junto e depende um do outro.

Termos tem diferentes nomes com a mesma semântica: por exemplo, um esquema de dados usa o termo gene para nomear o conceito gene enquanto outro esquema usa a palavra GeneHumano para definir o mesmo conceito de gene.

### **3.12 Áreas de aplicação das ontologias**

Ontologias têm sido amplamente utilizadas em várias aplicações computacionais.

A necessidade de apresentar de forma estruturada e organizada um vocabulário comum, usado em consenso por uma comunidade é de importância reconhecida aos desenvolvedores de soluções informatizadas.

São amplamente utilizadas em Inteligência Artificial, Engenharia do Conhecimento e Ciência da Computação, comércio eletrônico, processamento de linguagens naturais, recuperação da informação na *Web*, de cunho educacional, entre outros.

Para a área de Inteligência Artificial é ainda mais notória a necessidade de representação tanto do conhecimento de senso comum como o de domínios particulares. Na área mais específica de Processamento de Linguagem Natural é natural a familiaridade com ontologias, através de sua relação com a lingüística e pela necessidade de explicitar o léxico de uma língua.

No contexto da *Web Semântica*, busca-se tratar as informações da *Web* como uma rede de conceitos em contraposição a uma rede de documentos. A idéia é associar conhecimento do significado aos recursos da *Web*, tipicamente através da utilização de (meta) dados processáveis por máquinas. Cada conceito pode estar relacionado a outros conceitos e pode ter um grupo de recursos de informação associados. Esta rede de conceitos e recursos de informação é usada, então, na navegação na *Web* (BERNERS-LEE; HENDLER; LASSILA, 2001). Para definir a rede de conceitos na *Web Semântica*, ontologias têm sido sistematicamente utilizadas.

Na tarefa de processar linguagem natural, o conhecimento do domínio é muito importante para uma compreensão coerente do texto. Esse conhecimento do domínio pode ser dado por meio de uma ontologia sobre o domínio de discurso do texto. A aplicação apresentada por (EVERETT et al., 2002) é um exemplo da utilização de ontologias dentro da área de processamento de linguagem natural.

Empresas de consultoria como Arthur Andersen, Ernst & Young e Price Waterhouse têm como principal valor da empresa o conhecimento. Achar a informação certa o mais rápido possível dentro da empresa é de vital importância para a mesma.

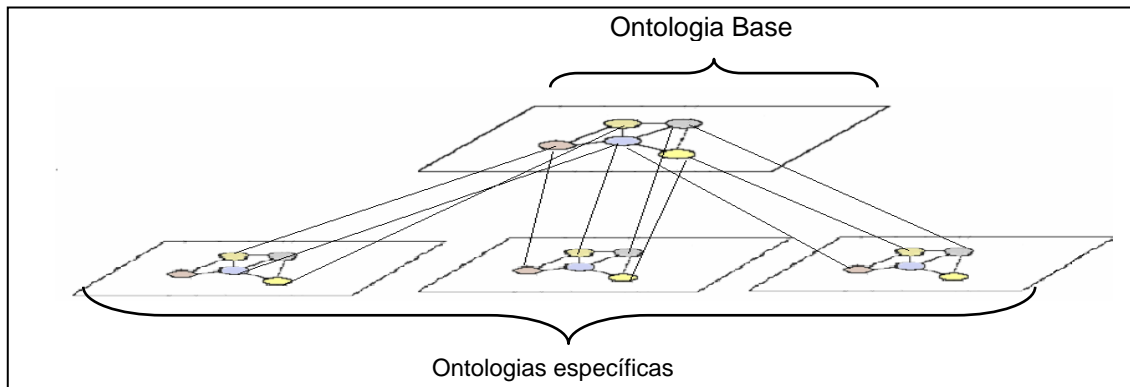
Tanto a área de Business to Business como também a área de Business to Consumer pode tirar proveito do uso de ontologias.

Na área de Business to Business, a automatização de transações requer uma descrição formal de produtos além de formatos de trocas sintáticas. Um entendimento comum dos termos e suas interpretações são capturados na forma de uma ontologia, permitindo assim interoperabilidade e meios para uma integração inteligente de informações (MAEDCHE; VOLZ, 2001).

Já na área de Business to Consumer, o uso de ontologias vem para solucionar as dificuldades existentes na construção de shopbots. Uma vez que todas as lojas on-line sigam uma mesma ontologia para a descrição de seus produtos, a tarefa de integração de catálogos das diversas lojas fica muito mais fácil. Ainda que não haja a adoção de uma ontologia única para a descrição de produtos a tarefa de mapeamento entre ontologias é muito mais simples em comparação com as dificuldades atuais.

## 4. ONTOLOGIA DE ONTOLOGIAS

Neste capítulo será ilustrado os conceitos de ontologia de ontologias que nos permitirá introduzir as idéias deste trabalho, que tem como proposta criar um nível mais alto que servirá de base para a criação de ontologias específicas para alguma área, conforme ilustra a Figura 4.1.



**Figura 4.1: Representação dos níveis de ontologias**

Esta ontologia base se caracteriza por apresentar todos os componentes que uma ontologia apresenta, ou seja, os elementos básicos que envolvem a criação de uma ontologia. As ontologias específicas se referem às ontologias de alguma área ou especialidade, como por exemplo, medicina, computação pervasiva, etc.

Existem situações que é necessário construir ontologias: para compartilhar a estrutura de informação comum entre pessoas ou softwares, para reutilizar do domínio do conhecimento, para tornar explícitos fatos consensuais, para separar um domínio do conhecimento operacional, para analisar um domínio. (NOY; GUINNESS, 2002)

A representação de uma ontologia pode ser feita em nível Meta ou em nível Objeto. O nível Meta significa a representação em um nível superior, contendo os elementos principais constituintes de uma ontologia, ou seja, um nível mais abstrato, não sendo descritos os detalhes específicos de implementação que se constituem no nível de Objeto. Neste trabalho foi utilizada a representação em nível Meta, a fim de evitar interpretações ambíguas.

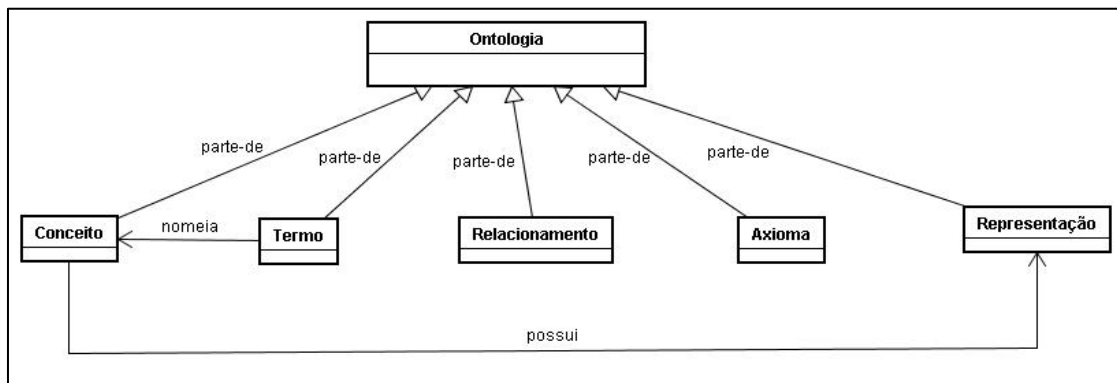
Para a criação da ontologia, os seguintes passos foram seguidos, baseados na metodologia proposta por (NOY; GUINNESS, 2002)

1. **Seleção dos termos referentes ao domínio ontologia:** Esta fase consistiu em criar uma lista com todos os termos da ontologia, através de um vocabulário

controlado, que tem a finalidade de nomear os conceitos, identificar suas propriedades e as relações desejadas. A partir disso, foi elaborada uma lista alfabética de termos, limitados a uma área de conhecimento especial, acompanhados das respectivas definições, chamada de glossário, conforme anexo I.

2. **Definir as classes e a hierarquia de classes da ontologia:** Nesta etapa os termos resultantes da etapa anterior foram colocados em hierarquias de forma que os mais genéricos foram especializados em termos mais específicos. Para isso, foi utilizada combinação das estratégias *top-down* e *bottom-up*, de acordo com o apresentado em (NOY; GUINNESS, 2002). Isto é, os principais conceitos foram definidos primeiro e foram refinados e/ou generalizados para a definição das demais classes.

A Figura 4.2, ilustra um diagrama contendo a descrição básica de uma ontologia e seus elementos básicos.



**Figura 4.2: Representação hierárquica de uma ontologia**

Nesta estrutura os termos são todas as palavras ou frases que servem para nomear um determinado conceito que, por sua vez, servem para designar uma categoria ou classe de entidades, eventos ou relações. Os relacionamentos servem para fazer a ligação entre os conceitos de um domínio e os axiomas são as sentenças que são sempre verdadeiras. Um exemplo de axioma será afirmar que “todo termo é parte de uma ontologia”.

#### 4.1 Representação das classes

Classes são os conceitos do domínio, nomeados pelo alfabeto controlado, são organizadas em hierarquias, definindo relacionamentos de generalização (subclasses → superclasse) e especialização (superclasse → subclasse).

As classes normalmente constituem uma hierarquia taxonômica, representadas por uma hierarquia é-um.

Uma instância de uma subclasse é uma instância de uma superclasse. Se a classe for pensada como um conjunto de elementos, uma subclasse é um subconjunto.

Usualmente, a existência de uma classe se justifica pelo fato de haver a possibilidade de gerar instâncias, no entanto, podem existir classes que não geram instâncias diretas, sendo utilizadas para organizar e simplificar uma hierarquia de generalização.

A figura 4.3 apresenta a hierarquia de classes, onde uma classe pode ser especializada em abstrata e concreta, sendo que o que as diferencia é o fato das classes concretas possuírem instâncias associadas.

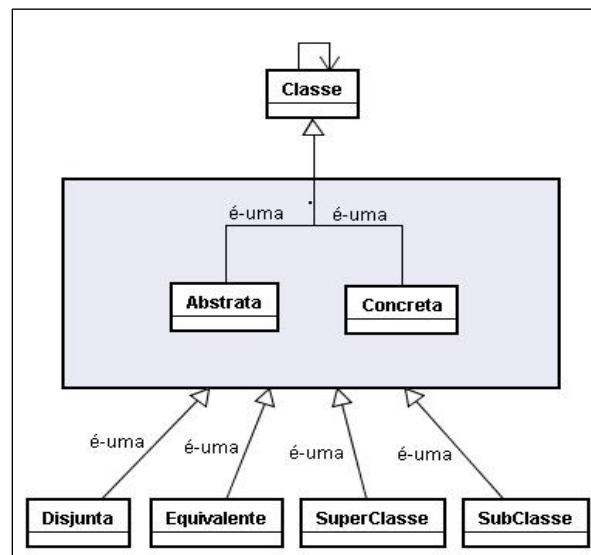


Figura 4.3: Hierarquia de classes

No entanto, em ambos os casos, as classes podem ser especializadas em disjuntas, equivalentes, superclasses e subclasses.

As classes disjuntas são todas as classes que quando relacionadas com outra não possuem nenhuma instância em comum, ou seja, a intersecção entre elas é vazia. Em OWL, a disjunção de um conjunto de classes pode ser expresso usando-se o construtor *owl:disjointWith*.

As classes são equivalentes se o relacionamento entre elas tiver as propriedades reflexiva, simétrica, transitiva. Em OWL duas propriedades podem ser ditas equivalentes usando-se a construção *owl:equivalentProperty*.

3. **Definir propriedades:** Somente as classes não fornecerão informação suficiente para responder às perguntas da competência do escopo. Uma vez que definidas as classes, deve-se descrever a estrutura interna dos conceitos.

## 4.2 Representação das propriedades

As classes apresentam propriedades (atributos e relacionamentos) em *slots*, sendo que cada classe é caracterizada por um conjunto de atributos e relacionamentos, cada um ocupando um *slot*.

As propriedades de uma classe são herdadas por suas subclasses, portanto as propriedades somente precisam ser definidas em um dos níveis da hierarquia.

A figura 4.4 mostra a hierarquia de propriedades, onde, as propriedades são especializadas em Propriedade Objeto e Propriedade de Tipo de Dados.

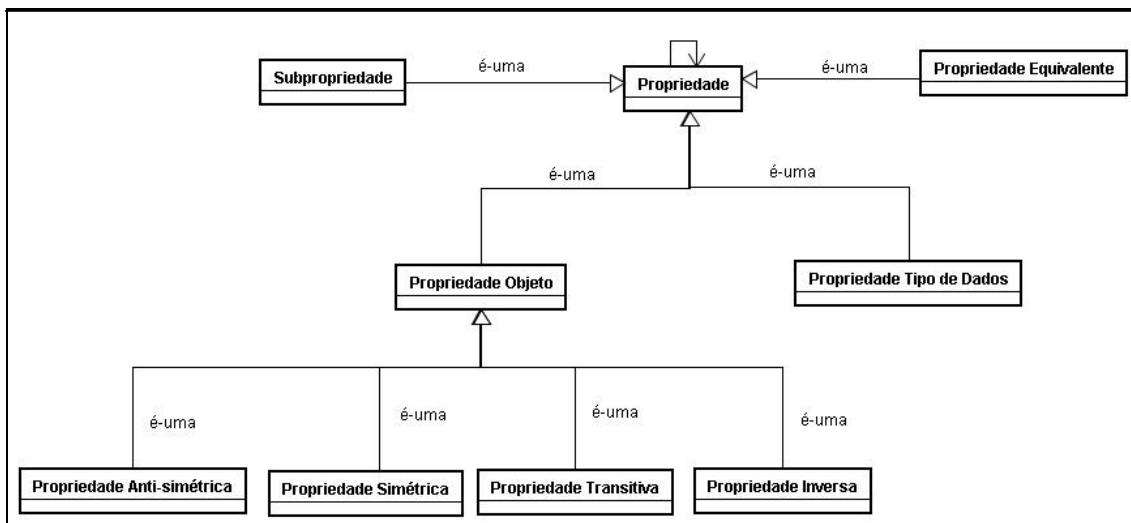


Figura 4.4: Hierarquia de propriedades

Uma propriedade é declarada como Propriedade Objeto quando tem o papel de relacionar uma classe à outra classe. Na figura 4.5, por exemplo, as classes Termo e Conceito foram relacionadas no atributo nomeia.

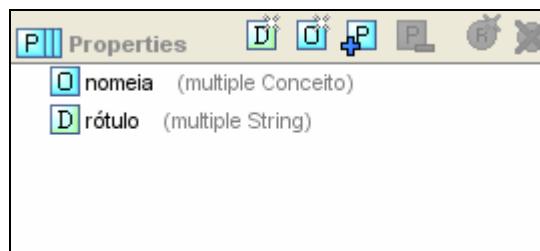


Figura 4.5: Propriedades da classe Termo

A Propriedade de Tipo de Dados se diferencia da Propriedade Objeto por utilizar uma variável para representar qualquer coisa no domínio abordado. Neste tipo de propriedade, também é necessário definir o domínio a qual ela pertence e o seu valor que, não mais será uma classe, mas um elemento do tipo string, boolean, int, entre outros.

Além disso as Propriedades Objeto são especializadas em Simétrica, Transitiva, Funcional e Inversa, que são definidas a seguir.

**Propriedade Simétrica:** Uma propriedade  $R$  é simétrica quando, se  $x$  se relaciona com  $y$ ,  $y$  se relaciona com  $x$ , assim: se para cada  $x, y \in E$ ;  $R(x,y)$  implica  $R(y, x)$ , conforme ilustra a Figura 4.6.

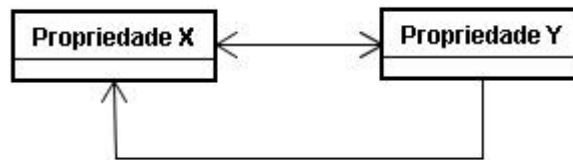


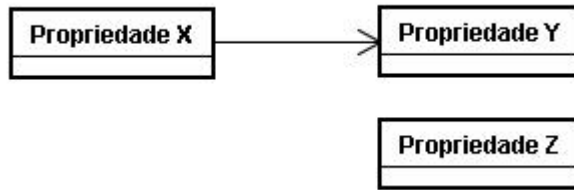
Figura 4.6: Propriedade simétrica

**Propriedade Transitiva:** Uma propriedade  $R$  é transitiva quando, se  $x$  está relacionado com  $y$  e  $y$  está relacionado com  $z$ , então  $x$  se relaciona com  $z$ , assim: se, para cada  $x,y,z \in E$ ; existe  $R(x,y)$  e  $R(y,z)$  implica  $R(x,z)$ , conforme ilustra a figura 4.7.



Figura 4.7: Propriedade Transitiva

**Propriedade Funcional:** Uma propriedade funcional é aquela em que se tem um único valor de  $y$  para cada instância de  $x$ . Portanto, não tem mais do que um único valor para cada indivíduo. Uma propriedade com esta característica é dita ter cardinalidade mínima 0 e cardinalidade máxima 1, assim: se, para cada  $x \in W$ , existe um único  $y \in W$  tal que  $R(x, y)$ , conforme ilustra a figura 4.8.



**Figura 4.8: Propriedade funcional**

Em OWL, a `owl:FunctionalProperty` é uma subclasse de `owl:DatatypeProperty` e também é uma subclasse de `owl:ObjectProperty` (BECHHOFFER et al , 2004).

As propriedades podem ser ditas funcionais inversas. Se uma propriedade é funcional inversa, então, o inverso da propriedade é funcional (HARMELEN; MCGUINNESS, 2004). Deste modo, o inverso da propriedade tem, no máximo, um valor para cada indivíduo.

**Propriedade Inversa:** As propriedades tem um sentido, no caso da inversa, este sentido pode ser dado de ambos os lados, assim para cada par  $P1(x,y)$  existe um inverso  $P2(y, x)$  e vice-versa.

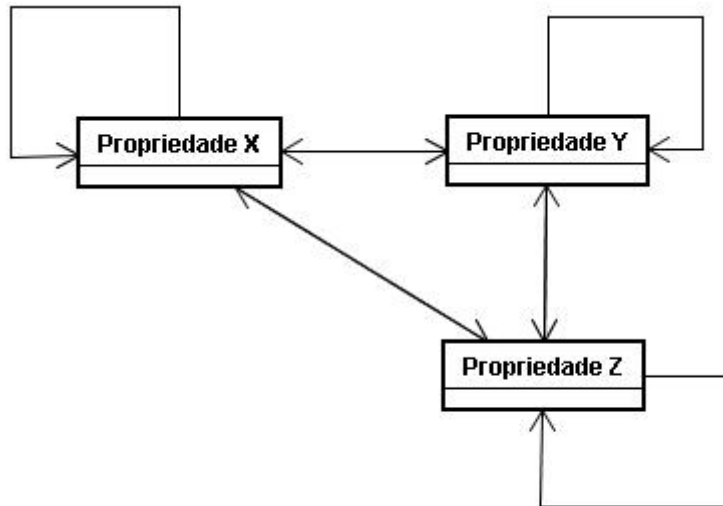
As propriedades também podem possuir subpropriedades, possuir relações de equivalência e serem reflexivas.

**Propriedade Reflexiva:** Quando todo elemento de E se relaciona com ele mesmo, assim: se, para cada  $x \in E$ , existe  $R(x,x)$  em E, conforme ilustra a Figura 4.9



**Figura 4.9: Propriedade Reflexiva**

**Propriedade Equivalente:** Uma propriedade é equivalente se possuir as propriedades reflexiva, transitiva e simétrica, conforme ilustra a figura 4.10.



**Figura 4.10: Propriedade Equivalente**

Existem outras propriedades interessantes: **Anti-simétrica**, **Irreflexiva**, **Assimétrica** e **Intransitiva**. Elas aparecem em outras áreas da ciência, mas não abrange o escopo deste trabalho.

4. **Definir restrições (facetar):** descrevem ou limitam o conjunto de valores possíveis para uma propriedade.

Cada propriedade é modelada por um conjunto de restrições que definem os valores que podem ser assumidos, os quais descrevem ou limitam o conjunto de valores possíveis para uma propriedade.

### 4.3 Representação das restrições

As restrições (facetar) mais comuns são a cardinalidade do valor da propriedade, seus limites, strings, números, elemento de um conjunto, etc.

Cardinalidade da propriedade – o número de valores que uma propriedade possui.

Tipo de Valor da propriedade – o tipo de valor que a propriedade pode apresentar.

Valores Mínimo e Máximo – um intervalo de valores para uma propriedade numérica.

Valor Default – o valor que uma propriedade apresenta, a menos que explicitamente especificado de outra forma.

Em OWL as restrições podem ser de valores (*allValuesFrom*, *someValuesFrom* e *hasValue*), de cardinalidade (*maxCardinality*, *minCardinality* e *cardinality*), de intersecção (*intersectionOf*), de união (*unionOf*) e de enumeração (*enumeration*.) A figura 4.11 ilustra exemplos de restrições definidos em OWL.

OWL Element	Symbol	Key	Example	Meaning of example
<i>allValuesFrom</i>	$\forall$	*	$\forall$ children Male	All children must be of type Male
<i>someValuesFrom</i>	$\exists$	?	$\exists$ children Lawyer	At least one child must be of type Lawyer
<i>hasValue</i>	$\ni$	\$	rich $\ni$ true	The rich property must have the value true
<i>cardinality</i>	=	=	children = 3	There must be exactly 3 children
<i>minCardinality</i>	$\geq$	>	children $\geq$ 3	There must be at least 3 children
<i>maxCardinality</i>	$\leq$	<	children $\leq$ 3	There must be at most 3 children
<i>complementOf</i>	$\neg$	!	$\neg$ Parent	Anything that is not of type Parent
<i>intersectionOf</i>	$\sqcap$	&	Human $\sqcap$ Male	All Humans that are Male
<i>unionOf</i>	$\sqcup$		Doctor $\sqcup$ Lawyer	Anything that is either Doctor or Lawyer
<i>enumeration</i>	{...}	{ }	{male female}	The individuals male or female

Figura 4.11. Definição de restrição em OWL, no ambiente Protégé.

### 4.3.1 Restrição de valor

A seguir são detalhadas as principais restrições de valores, utilizadas em OWL.

**AllValuesFrom:** A restrição *allValuesFrom* é declarada em uma propriedade com respeito a uma classe, isto é, a restrição requer que, para cada instância da classe que tenha instâncias da propriedade especificada, os valores da propriedade devem ser todos os membros da classe indicada pela cláusula *allValuesFrom*. (SMITH; WELTY; MCGUINNESS, 2004)

**SomeValuesFrom:** A restrição *someValuesFrom* descreve a classe de indivíduos *x* para os quais existe pelo menos um *y*, tal que o par (*x,y*) seja uma instância da propriedade, isto é, a restrição requer que, pelo menos para uma instância da classe que tenha instâncias da propriedade especificada, os valores da propriedade devem ser membros da classe indicada pela cláusula *someValuesFrom*. (SMITH; WELTY; MCGUINNESS, 2004)

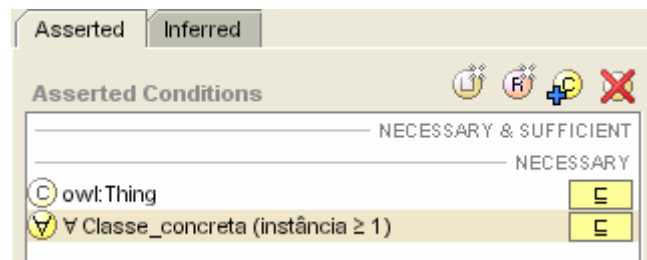
**HasValue:** A propriedade *hasvalue* permite que uma propriedade tenha um certo indivíduo como valor. Em outras palavras, *hasvalue* permite especificar classes baseadas na existência de valores de propriedades particulares, assim, um indivíduo será membro de tal classe sempre que pelo menos um dos valores das propriedades for igual ao valor do recurso *hasvalue*. (SMITH; WELTY; MCGUINNESS, 2004)

### 4.3.2 Restrições de cardinalidade

A OWL permite utilizar restrições de cardinalidade, uma vez que qualquer instância de uma classe pode ter um número arbitrário de valores para uma propriedade. As restrições de cardinalidade são restrições locais, uma vez que elas são declaradas em propriedades com respeito a uma classe. Segundo (BECHHOFFER et al , 2004), a OWL provê três construções para cardinalidade:

- **owl:maxCardinality**: descreve uma classe de todos os indivíduos que têm, no máximo, N valores semanticamente distintos.
- **owl:minCardinality**: descreve uma classe de todos os indivíduos que têm, no mínimo, N valores semanticamente distintos. Esta restrição é um meio para dizer que uma propriedade é requerida ter um valor para todas as instâncias da classe.
- **owl:cardinality**: descreve uma classe de todos os indivíduos que têm exatamente N valores semanticamente distintos.

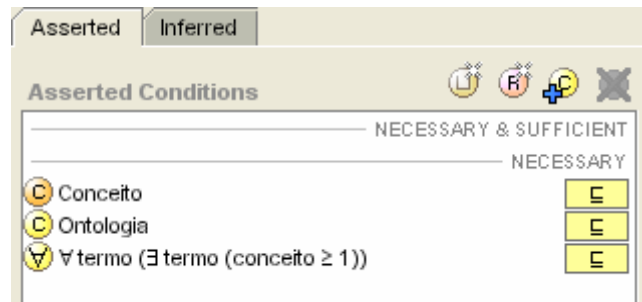
Para estas cardinalidades, a figura 4.12 mostra um exemplo da classe concreta que possui no mínimo uma instância associada e no máximo N instâncias associadas.



**Figura 4.12 Cardinalidade da Classe Concreta**

É importante ressaltar que, antes de se construir uma restrição, deve-se definir uma classe para armazenar todas as instâncias referentes ao mesmo.

Observa-se que a restrição é uma forma de resposta às questões que são de competência da ontologia. Uma classe complexa formada a partir de equivalências, ou dos quantificadores universal e existencial, é a representação em OWL das verdades do domínio, ou seja, das suas restrições. A figura 4.12 ilustra um exemplo de definição de restrição desenvolvida no ambiente Protégé, onde para todo termo existe no mínimo um conceito associado.



**Figura 4.13: Definição da restrição Termo juntamente com a classe e condição necessária, no Protégé**

Restrições também podem ser utilizadas para representar o “ou” exclusivo (xor), quando instâncias de duas ou mais classes podem se relacionar com instâncias de uma outra classe específica.

As restrições podem ainda ser utilizadas para definir melhor a semântica de classes especializadas derivadas de classes gerais. As restrições pré-definidas para classes especializadas são:

**Completa:** quando todas as sub-classes possíveis foram derivadas da classe geral;

**Incompleta:** quando ainda é possível derivar novas sub-classes.

**Separada ou Disjunta:** quando as subclasses são mutuamente exclusivas, ou seja, no momento que uma instância pertence a uma sub-classe, não poderá de forma alguma pertencer a qualquer das outras subclasses derivadas.

**Sobreposta:** quando o fato de pertencer a uma sub-classe não impede que pertença a outras.

5. **Definir instâncias:** As instâncias são entidades do domínio que atendem às especificações de uma classe. A atividade de instanciação corresponde a criar os registros de uma base de dados a partir do seu esquema (descrição das classes).

#### 4.4 Representação das instâncias

Na Ontologia de Ontologias todos os elementos que são representados pelas ontologias pode ser instanciados.

A figura 4.14 apresenta um exemplo da instanciação da Classe Termo, que é nomeado por um conceito  $x$ , que seria a própria definição da palavra termo. Os termos

são representados por Classes e podem se relacionar com outros termos através de propriedades.

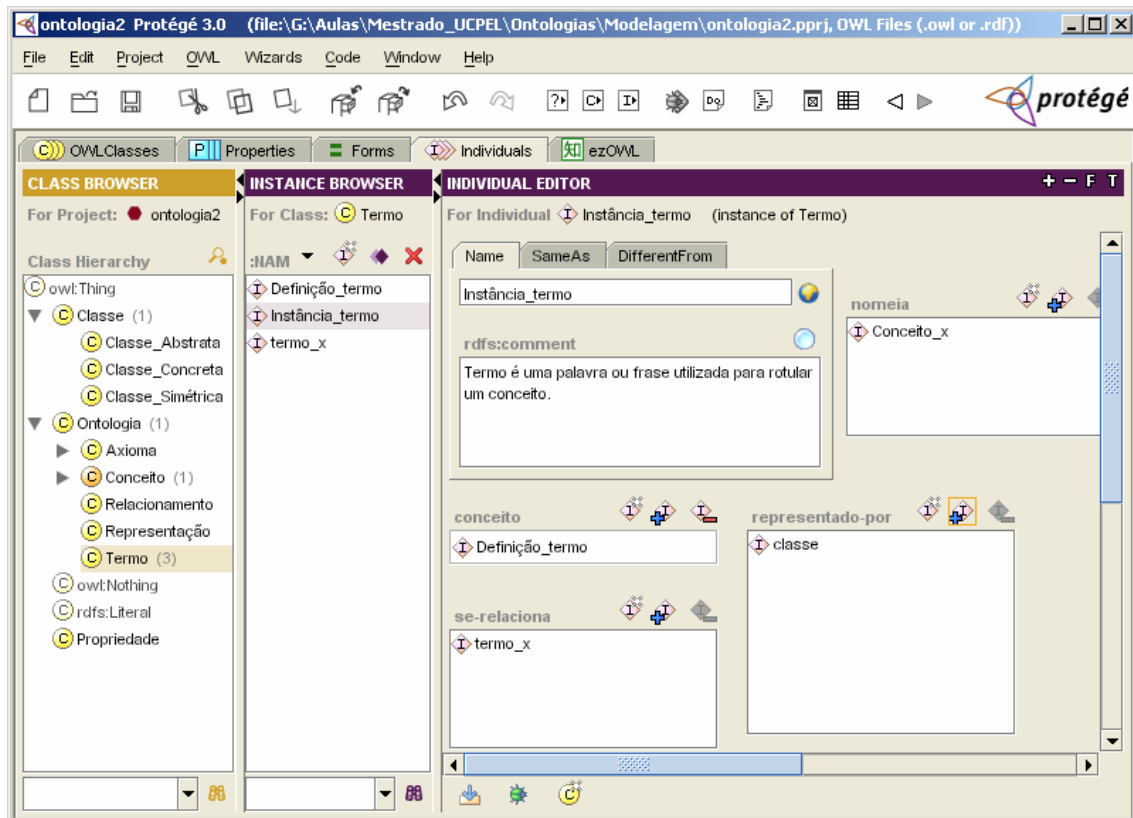


Figura 4.14 Exemplo de Instanciação da Classe Termo

## 4.5 Representação da Ontologia de Ontologias

A fase final no desenvolvimento de ontologias é realizada por meio de uma representação específica que permite o processamento e a abrangência do conhecimento pela máquina.

Isso é possível através de uma linguagem específica para a criação de ontologias e de uma ferramenta que permita sistematizar e integrar as especificações definidas à linguagem utilizada.

No desenvolvimento da ontologia de ontologias, foi utilizada a linguagem OWL e a na implementação utilizou-se o editor Protege (Protege, 2005) (NOY, et al. 2001). O Protégé possui uma interface de fácil utilização para a criação de instâncias e uma quantidade considerável de *plugins* que aumentam o número de funcionalidades do Protégé.

Neste trabalho foram utilizados *plugins*, tais como: *plugin* que permite exportar e importar ontologias no formato OWL (OwlPlugin, 2005) e um *plugin* que permite ao usuário do Protege trabalhar no desenvolvimento de ontologias na forma

de diagramas chamado ezOWL (ezOwl, 2005). A figura 4.13 ilustra a utilização do editor Protégé com o plugin ezOWL para uma representação gráfica parcial da ontologia de ontologias.

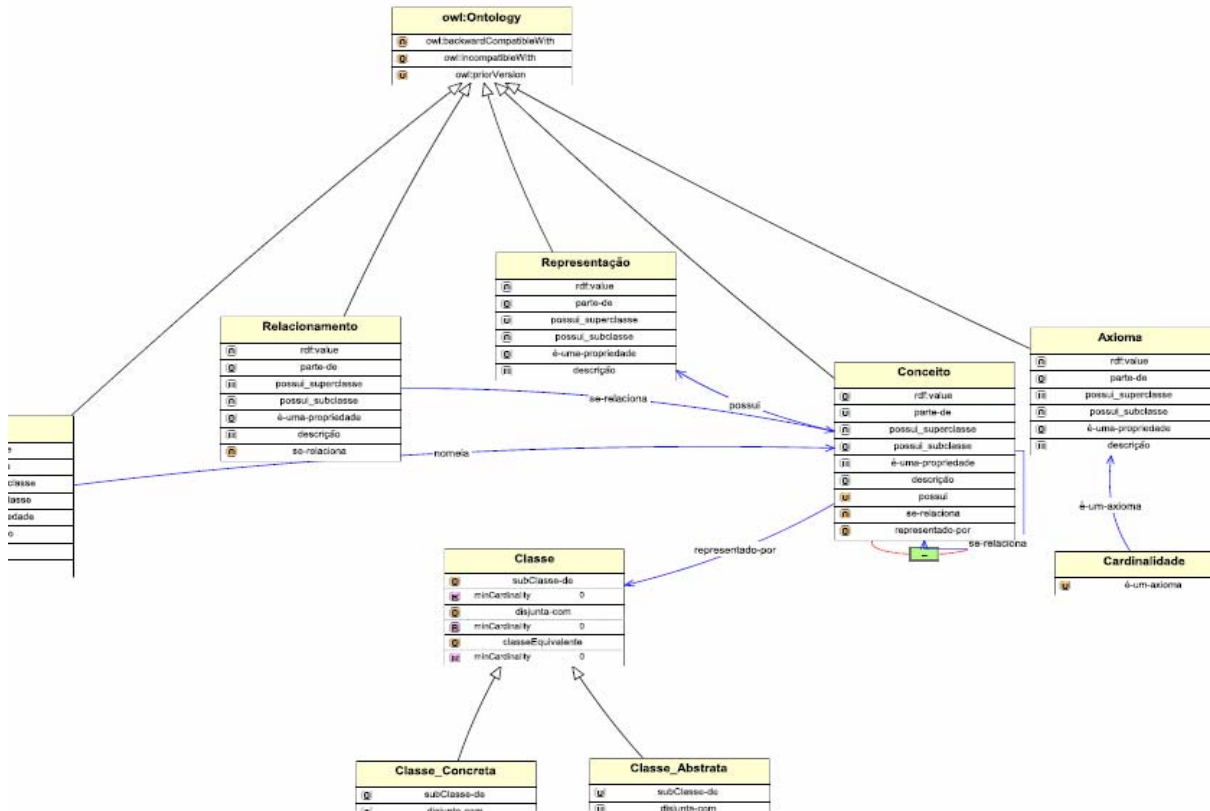


Figura 4.15: Representação parcial da ontologia de ontologias.

A figura 4.16 ilustra um trecho do código OWL gerado na Ontologia de Ontologias criado no ambiente Protégé.

```
<owl:Class rdf:ID="Cardinalidade_máxima">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Axioma"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Classe_Concreta">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Classe"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Cardinalidade_minima">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Axioma"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Axioma">
  <rdfs:subClassOf rdf:resource="#Ontologia"/>
</owl:Class>
<owl:Class rdf:ID="Representação">
  <rdfs:subClassOf rdf:resource="#Ontologia"/>
</owl:Class>
<owl:Class rdf:ID="Propriedade"/>
<owl:Class rdf:ID="Classe_Simétrica">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Classe"/>
  </rdfs:subClassOf>
</owl:Class>
```

Figura 4.16 Trecho de código OWL da Ontologia de Ontologias

## 5. TRABALHOS RELACIONADOS

Não são muitos os trabalhos que têm como proposta a criação de uma ontologia para o domínio das ontologias. Dentre os poucos trabalhos destaca-se o projeto Ontology of Ontology (O3), proposto por Xiaoshu Wang, criado para classificar ontologias de acordo com a conceitualização que uma ontologia oferece. É proposto um modelo simbólico e dicotômico, a fim de permitir compartilhamento e reuso desta ontologia.

Primeiramente, O3 divide toda a ontologia em três categorias dependendo do tipo de conceitualização a que uma ontologia se dirige, conforme Figura 5.1 abaixo:

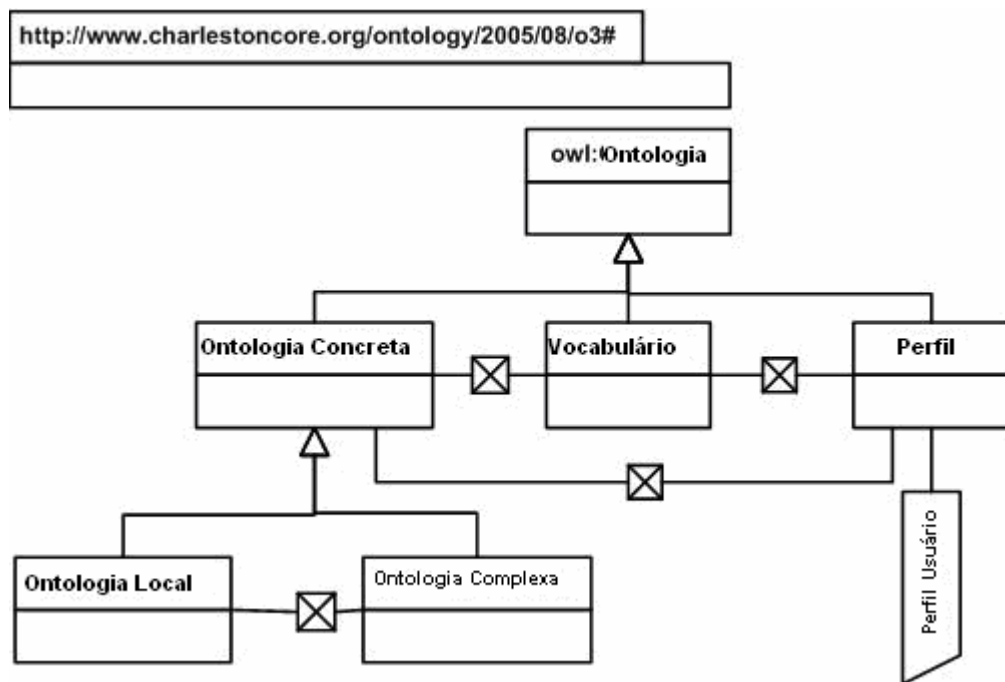


Figura 5.1. A estrutura de da Ontologia de Ontologia (O3)

O vocabulário é uma ontologia que liga conceitos de um domínio com um URI mas não oferece nenhuma descrição adicional.

Em contraste ao Vocabulário, o Perfil não liga um conceito do domínio com URI. O namespaces de um perfil é conseqüentemente um conjunto vazio. Um perfil combina um número de ontologias e reinterpreta/combina a conceitualização de ontologias individuais em um visão particular do mundo.

Ontologia Concreta é definida para ser aquelas ontologias que não são nem um Vocabulário nem um Perfil, formando uma disjunção entre elas. São divididas em Ontologia Local e Ontologia Complexa.

Outra referência é a proposta para um padrão de metadata, chamado de Ontology Metadata Vocabulary (OMV), proposto por (HARTMANN; PALMA; SURE, 2006) que é baseado em discussões e em acordo na rede thematic do EU IST do conhecimento Web1 do excellence. OMV está disponível para o download em <http://ontoware.org/projects/omv>. A versão atual 0.7 tem validado como a OWL Lite pelo Validator de Ontology da OWL de WonderWeb.

## **6. TRABALHOS FUTUROS**

Para que várias ontologias trabalhem em conjunto, trocando automaticamente as informações, são necessários mecanismos que garantam a interoperabilidade de ontologias.

Para (BISHR, 1997), interoperabilidade é a capacidade que um sistema possui de compartilhar e trocar informações em aplicações.

Conforme descrito no capítulo 3.8, atualmente existem algumas aproximações para o processo de compatibilidade de ontologias, entre elas: combinação de ontologias, alinhamento de ontologias, mapeamento de ontologias, integração de ontologias, entre outras.

No entanto, nestes mecanismos alguns fatores ainda não foram explorados, como a quantificação dos relacionamentos existentes, ou seja, todas as relações possuem a mesma intensidade não havendo diferença entre alguns atributos que podem ser considerados mais relevantes que outros. Em uma ontologia do domínio de pessoas, todas as pessoas possuem um perfil, com características e preferências próprias, sendo algumas mais destacadas que outras, como gostar de assistir televisão, por exemplo.

Além disso, a partir deste trabalho inúmeros trabalhos futuros poderão surgir, pois o assunto ontologias está se tornando cada vez mais popular atraindo, inclusive, pesquisadores de outras áreas.

## **7. CONSIDERAÇÕES FINAIS**

Atualmente existem inúmeras ontologias que estão sendo criadas, no entanto não existe um padrão oficial para a sua criação o que dificulta a reusabilidade de conteúdos.

O presente trabalho inseriu-se neste contexto, propondo um modelo generalizado para representação de uma ontologia, podendo assim ser aplicável a qualquer cenário. Para a elaboração deste modelo genérico, buscou-se abordar de forma abrangente os requisitos necessários para o modelo de ontologias. Isso foi possível através de extensas análises em trabalhos relacionados à ontologias, abordados de forma específica, relacionando-se ao que tinham em comum e os elementos que julgou necessário acrescentar para obter uma maior abrangência.

Eespera-se contribuir com este trabalho para um maior entendimento sobre o assunto ontologias e também servir como ponte para outras pesquisas, além de fornecer um conjunto de referências a serem exploradas com mais detalhes.

## 8. BIBLIOGRAFIA

**ALMEIDA, M.B.; BAX, M.P.** *Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção.* Ci. Inf., Brasília, v. 32, n. 3, p. 7-20, set./dez. 2003. Disponível em: <[www.ibict.br/cienciadainformacao](http://www.ibict.br/cienciadainformacao)>. Acessado em novembro, 2006.

**ANTONIOU, G.; VAN HARMELEN, F.** *A Semantic Web Prime* Massachusetts Institute of Technology, 2004.

**ARPIREZ, J. C. et al.** *Web ODE: a scalable workbench for ontological engineering.* In: INTERNATIONAL CONFERENCE ON KNOWLEDGE CAPTURE. PROCEEDINGS. Victoria, British Columbia, Canada, 2001.

**BAADER, F. , CALVANESE, D., D. L. MCGUINNESS, D.L., NARDI, D., PATEL-SCHNEIDER, P.F** apud : *The Description Logic Handbook: Theory, Implementation, Applications.* Cambridge University Press, Cambridge, UK, 2003.

**BECHHOFFER, S; HARMELEN, F. V; HENDLER, J; HORROCKS, I; MCGUINNESS, D. L; PATEL-SCHNEIDER, P. F; STEIN, L. A.** *OWL Web Ontology Language Reference.* [http://www.w3.org/TR/2004/ REC-owl-ref-20040210](http://www.w3.org/TR/2004/REC-owl-ref-20040210), acessado em Maio 2006.

**BENJAMINS, R.** *Knowledge Engineering and Ontologies.* Disponível em: <<http://www.swi.psy.uva.nl/usr/richard/home.html>>. Acesso em: 12 maio 2006.

**BERNERS-LEE, T.; HENDLER, J.; LASSILA, O.** *The Semantic Web., Scientific American.* May, 2001.

**BISHR, Y.** *Semantic Aspect of Interoperable GIS.* Ph.D. Thesis, Wageningen Agricultural University, The Netherlands, 1997.

**BORST, W. N.** *Construction of engineering ontologies.* 1997. Tese (Doutorado).

Disponível em: <<http://www.ub.utwente.nl/webdocs/inf/1/t0000004.pdf>>. Acesso em: 21 maio 2006.

**BRACHMAN, R.J., FIKES, R.E. and LEVESQUE, H.J..** Krypton: A functional approach to knowledge representation. *IEEE Computer (Special Issue on Knowledge Representation)*, 16(10):67-73, October 1983.

**CORAZZON, R.** *What is ontology?* [S. l. : s. n.], 2002. Disponível em: <[http://www.formalontology.it/section\\_4.htm](http://www.formalontology.it/section_4.htm)>. Acesso em: 20 jul. 2006.

**DAVIES, J; FENSEL, D; VAN HARMELEN, F. .** *Towards the Semantic Web: Ontology driven knowledge management.* John Wiley, West Sussex, 2003.

**DOMINGUE, J.; MOTTA, E. A; CORCHO, O.** Knowledge modeling in web onto and OCML: a user guide. [ S. l. : s. n.], 1999. Disponível em: <[http://kmi.open.ac.uk/projects/webonto/user\\_guide.2.4.pdf](http://kmi.open.ac.uk/projects/webonto/user_guide.2.4.pdf)>. Acesso em: junho 2006.

**FARQUHAR, A.; FIKES, R.; RICE, J.** The ontolingua server: USA: a tool for collaborative ontology construction. Duluth : Academic Press, 1997. p. 707-727.

**GÓMEZ-PÉREZ, A.; FERNANDEZ-LÓPEZ, M.; CORCHO, O.** *Ontological engineering.* London, Springer, 2004.

**GRUBER, T.** (1996). *What is an ontology?* [S. l. : s. n.], 1996. Disponível em: <<http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>>. Acesso em: 14 set. 2006.

**GRUBER, T.** *Ontolândia: a mechanism to suport portable ontologies.* Stanford: Knowledge System Laboratory, Stanford University, 2003. Technical Report.

**GUARINO, N.** (1998). Formal Ontology and Information Systems. Disponível em: <http://www.loa-cnr.it/Papers/FOIS98.pdf>. Último acesso em 21 de setembro de 2006.

- HARMELEN, F. V.; MCGUINNESS, D. L.** OWL *Web Ontology Language Overview*. <http://www.w3.org/TR/2004/REC-owl-features-20040210/>, acessado em Abril de 2006.
- HARTMANN, J.; PALMA, R.; SURE, Y.** OMV – Ontology Metadata Vocabulary. Disponível em: <http://omv.ontoware.org>. Acesso em 27 de novembro de 2006.
- HORROCKS, I.; SATTLER, U.; TOBIES, S.** Practical reasoning for expressive description logics. *Logic Journal of the IGPL*, v. 8, n. 3, p. 239-264, may 2000.
- LIBRELOTTO, G.R., RAMALHO, J.C., HENRIQUES, P.R.** *TM-Builder: Um Construtor de Ontologias Baseado em Topic Maps*, XXIX Conferencia Latinoamericana de Informática, 2003.
- MAEDCHE, A.; VOLZ, R.** The text-to-onto ontology extraction and maintenance environment to appear. In: PROCEEDINGS OF THE ICDM WORKSHOP ON INTEGRATING DATA MINING AND KNOWLEDGE MANAGEMENT, 2001, San Jose, California. [S. l. : .n.], 2001.
- MAHALINGAM, K.; HUHNS, M. N.** An ontology tool for query formulation in an agent-based context. *In: IFCIS INTERNATIONAL CONFERENCE ON COOPERATIVE INFORMATION SYSTEMS, 2., 1997, Kiawah Island, SC.* [S. l. : s. n.], 1997. p. 170.
- MATTHEW, H.; KNUBLAUCH, H.; ALAN, R.; STEVENS, R.; WROE, C..** *A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools*. Manchester University, England, 2004.
- MCGUINNESS, D. L.; FIKES, R.; RICE, J., and WILDER, S..** The Chimaera Ontology Environment." Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI 2000). Austin, Texas. July 30 - August 3, 2000.
- MILLER, G. A.** WordNet: a lexical database for English. *Communications of the ACM*, v. 38, n. 11, p. 39-41, nov. 1995.

- MINSKY**, M.A. *A Framework for Representing Knowledge*. McGraw-Hill, New York, 1975
- MIZOGUCHI, R.** *Tutorial on ontological engineering*. The Institute of Scientific and Industrial Research, Osaka University.
- NOY**, F. N.; **GUINNESS**, D. L. *Ontology development 101: a guide to create your first ontology*. Stanford University, USA, 2002. Disponível em: <<http://ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.doc>>. Acesso em: 12 setembro 2006.
- NOY**, N.; **MUSEN**, M. **SMART**: Automated Support for Ontology Merging and Alignment. Banff Workshop on Knowledge Acquisition, Modeling, and Management, Banff, Alberta, Canada, 1999.
- NOY**, N.; **MUSEN**, M. The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping. *International Journal of Human-Computer Studies*, 2003.
- PALAZZO, L.A. M.** Projeto e Construção de Ontologias. Disponível em: <http://ia.ucpel.tche.br/~lpalazzo/Aulas/Ontolog/PCO3.ppt>.
- PALAZZO**, L.; **PONTES**, A.; **ULYSSEA**, M.; **PORTO**, P. Comunidades Virtuais de Aprendizado Adaptativo. In: Conferência Nacional em Ciência, Tecnologia e Inovação. Ministério da Ciência e Tecnologia, Florianópolis, 2001.
- PINTO**, S.; **GOMEZ-PEREZ**, A.; **MARTINS**, J. Some Issues on Ontology Integration. Workshop on Ontologies and Problem Solving Methods: Lessons Learned and Future Trends, 1999.
- RICH**, E. *Artificial Intelligence*. McGraw-Hill Book Company, 1983.
- SCIME**, A. *Web Mining: Applications and Techniques*. State University of New York College at Brockport, USA, 2004.
- SMITH**, M. K.; **WELTY**, C; **MCGUINNESS**, D. L. *OWL Web Ontology Language Guide*. <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>, acessado em Novembro de 2006.
- SOWA**, J. F. *Knowledge Representation: Logical, Philosophical, and Computational Foundation*. Brooks Cole Publishing Co., Pacific Grove, CA, 2000. Actual

publication date, 16 August 1999.

**SOWA**, J. F. Building, sharing and merging ontologies. Tutorial. [S. l. : s. n.], 1999.

Disponível em: <<http://users.bestweb.net/~sowa/ontology/ontoshar.htm>>. Acesso em: 12 setembro 2006.

**SKUCE**, D. **IKARUS**. CODE4: a unified system for managing conceptual knowledge.

*International Journal of Human-Computer Studies*, n. 42, p. 413-451, 1995.

**SWARTOUT**, B. et al. Toward distributed use of large-scale ontologies. In:

PROCEEDINGS OF AAAI97 SPRING SYMPOSIUM SERIES WORKSHOP ON ONTOLOGICAL ENGINEERING, 1997. [S. l.] : AAAI Press, 1997. p. 138-148.

**TOURETZKY**, D.S., **HORTY**, J.F. and **THOMASON**, R.H.

A clash of intuitions : The current state of nonmonotonic multiple inheritance systems. In *Proceedings of IJCAI 10*, pages 476-482, 1987.

**VICKERY**, B. C. Classification and Indexing in Science. 3 ed. Londres: Butterworths, 1975.

**WOODS**, W.A. What's in a link : Foundations for semantic networks.

In D.G. Bobrow and A. Collins, editors, *Representation and Understanding: Studies in Cognitive Science*. Academic Press, New York, 1975.

[**OWL, 2004**]. Disponível em: <http://www.w3.org/TR/owl-features>. Acesso em: 28 novembro de 2006.

[**EZOWL, 2005**]. Disponível em: <http://protege.stanford.edu/plugin> . Acesso em: 28 novembro de 2006.

[**PROTEGÉ, 2005**]. Disponível em: <http://protege.stanford.edu>. Acesso em: 28 novembro de 2006.

# **ANEXOS**

## ANEXO 1 - Glossário de termos do domínio Ontologia

Termo	Sinônimo	Definição
<b>Anotação</b>	<b>Literal</b>	Anotações são comumente utilizadas para adicionar informações em <b>classes</b> , <b>objetos</b> e <b>propriedades</b> . Três tipos de anotação são suportados em OWL: rdfs:comment, rdfs:label and owl:versionInfo.
<b>Asserção</b>	<b>Afirmação</b> <b>Fato</b>	Uma asserção é qualquer declaração que é verdadeira em uma <b>ontologia</b> .
<b>Axioma</b>		São as sentenças que são sempre verdadeiras.
<b>Cardinalidade</b>		Representa o número exato de valores que devem estar associados à um <b>atributo</b> para uma classe.
<b>Cardinalidade máxima</b>		Representa o número máximo de valores que poderiam ter um <b>atributo</b> para uma classe.
<b>Cardinalidade mínima</b>		Representa o número mínimo de valores que poderiam ter um <b>atributo</b> para uma classe.
<b>Classe</b>	<b>Categoria</b>	Uma representação para um agrupamento conceitual de <b>termos</b> similares. Descrevem os <b>conceitos</b> em um domínio.
<b>Classe Abstrata</b>		Uma <b>classe</b> que não pode ter instâncias associadas.
<b>Classe Concreta</b>		São <b>classes</b> que podem ter instâncias associadas.
<b>Classe Equivalente</b>		<b>Classes</b> são equivalentes se o relacionamento entre elas tiver as propriedades reflexiva, simétrica, transitiva, etc.
<b>Classe Disjunta</b>		<b>Classes</b> são disjuntas se elas não possuem nenhuma instância em comum, ou seja, a intersecção é vazia, sendo que nenhum elemento pertence as 2 classes.
<b>Conceito</b>	<b>Definição</b>	É uma <b>entidade</b> abstrata e universal que serve para designar uma categoria ou <b>classe</b> de entidades, eventos ou relações.
<b>Domínio</b>		Um campo particular do conhecimento.

<b>Elemento</b>		Os <b>elementos</b> são todos aqueles objetos que quando coletados junto compõem uma <b>classe</b> .
<b>Faceta</b>	<b>Axioma, Restrição</b>	É uma <b>propriedade</b> de um <b>atributo</b> . A maioria de <b>facet</b> as comumente usados são: <b>Cardinalidade, Cardinalidade-mínima, Cardinalidade-máxima</b> e <b>Valor-tipo</b> .
<b>Importa</b>		Ontologias podem conter outras <b>ontologias</b> , importando suas <b>classes, subclasses, propriedades</b> e <b>individuais</b> .
<b>Individual</b>		É uma <b>instância</b> específica de uma <b>classe</b> em uma estrutura hierárquica de uma <b>ontologia</b> . Descreve mais tipicamente toda classe numericamente singular.
<b>Inferência</b>		Operação mental pela qual extraímos uma conclusão nova de uma ou mais conclusões mais conhecidas.
<b>Instância</b>	<b>Indivíduo</b>	Todos os <b>termos</b> em uma ontologia que tem uma definição associada (por exemplo classes, slots, relações, funções, facets) são uma instância de alguma <b>classe</b> . <b>Classes</b> são <b>instâncias</b> de uma <b>classe</b> , <b>funções</b> são <b>instâncias</b> de <b>funções</b> , etc. Representam <b>indivíduos</b> específicos de uma determinada <b>classe</b> .
<b>Método</b>	<b>Procedimento</b>	Um grupo de instruções em uma <b>classe</b> , que define como os objetos da <b>classe</b> se comportarão.
<b>Ontologia</b>		Um ontologia define os vários elementos que podem ser usados para descrever e representar um <b>domínio</b> do conhecimento. É uma estrutura hierárquica que define as relações semânticas de um conjunto de <b>conceitos</b> .
<b>Propriedade</b>	<b>Atributo Slot</b>	Descrevam as características e <b>atributos</b> de um <b>conceito</b> . Existem dois tipos principais de <b>propriedades</b> : de objeto e de tipo de dados.
<b>Propriedade Equivalente</b>		Uma <b>propriedade</b> é equivalente se possuir as <b>propriedades reflexiva, transitiva</b> e <b>simétrica</b> .
<b>Propriedade Funcional</b>		Uma <b>propriedade</b> funcional é aquela em que se tem um único valor de y para cada instância de x, assim: se, para cada $x \in W$ , existe um único $y \in W$ tal que $R(x, y)$ .
<b>Propriedade</b>		As <b>propriedades</b> tem um sentido, no caso

<b>Inversa</b>		da inversa, este sentido pode ser dado de ambos os lados, assim para cada par $P1(x,y)$ existe um inverso $P2(yx,)$ e vice-versa.
<b>Propriedade Objeto</b>		É toda <b>propriedade</b> que tem o papel de relacionar uma classe à outra.
<b>Propriedade Simétrica</b>		Uma <b>propriedade</b> $R$ é simétrica quando, se $x$ se relaciona com $y$ , $y$ se relaciona com $x$ , assim: se para cada $x, y \in E$ ; $R(x,y)$ implica $R(y, x)$
<b>Propriedade Reflexiva</b>		Uma <b>propriedade</b> $R$ é reflexiva quando todo elemento de $E$ se relaciona com ele mesmo, assim: se, para cada $x \in E$ , existe $R(x,x)$ em $E$ .
<b>Propriedade Transitiva</b>		Uma <b>propriedade</b> $R$ é transitiva quando, se $x$ está relacionado com $y$ e $y$ está relacionado com $z$ , então $x$ se relaciona com $z$ , assim: se, para cada $x,y,z \in E$ ; existe $R(x,y)$ e $R(y,z)$ implica $R(x,z)$ .
<b>Propriedade de tipo de dados</b>		É toda <b>propriedade</b> que utiliza uma variável para representar qualquer coisa no <b>domínio</b> abordado. Neste tipo de <b>propriedade</b> , também é necessário definir o <b>domínio</b> a qual ela pertence e o seu <b>valor</b> que, não será uma classe, mas um elemento do tipo <i>string, boolean, int</i> entre outros.
<b>Relação</b>	<b>Conexão</b>	Uma <b>relação</b> é usada para descrever um relacionamento entre dois ou mais <b>termos</b> .
<b>Relacionamento</b>	<b>Ligação Vínculo</b>	É a ligação existente entre os <b>conceitos</b> de um determinado <b>domínio</b> . Os <b>relacionamentos</b> expressam a natureza da ligação existente entre dois <b>conceitos</b> e podem possuir <b>cardinalidade</b> .
<b>Representação</b>		São os formalismos utilizados para representar tudo o que uma ontologia possui, pode ser um texto, diagramas, linguagens, etc.
<b>Subclasse</b>	<b>ClasseFilha</b>	É uma especialização de outra classe, está mais abaixo na hierarquia de <b>classes</b> do que outra <b>classe</b> , sua superclasse. Uma <b>classe</b> pode ter tantas <b>subclasses</b> quantas forem necessárias.
<b>Subpropriedade</b>	<b>PropriedadeFilha</b>	Define que a <b>propriedade</b> é uma <b>subpropriedade</b> de alguma outra <b>propriedade</b> . Formalmente isto significa que se $P1$ for uma <b>subpropriedade</b> de $P2$ , então a extensão da <b>propriedade</b> de $P1$ (um conjunto de pares) deve ser um

		subconjunto da extensão da <b>propriedade</b> de P2 (também um conjunto de pares)
<b>Superclasse</b>	<b>ClassePai</b> <b>ClasseBase</b>	São <b>classes</b> de que outras <b>classes</b> são derivadas. É o inverso da <b>subclasse</b> . Assim, que se uma <b>classe A</b> é <b>subclasse</b> da <b>classe B</b> , a <b>classe B</b> é <b>superclasse</b> da <b>classe A</b> .
<b>Superpropriedade</b>	<b>PropriedadePai</b> <b>PropriedadeBase</b>	São <b>propriedades</b> de que outras <b>propriedades</b> são derivadas. É o inverso da <b>subpropriedade</b> Assim, que se uma <b>propriedade A</b> é subpropriedade da <b>propriedade B</b> , a <b>propriedade B</b> é <b>superpropriedade</b> da <b>propriedade A</b> .
<b>Termo</b>	<b>Rótulo</b>	É uma palavra ou frase utilizada para rotular um <b>conceito</b> .
<b>Tipo</b>		Uma <b>faceta</b> da <b>propriedade</b> que identifique o tipo dos valores que uma <b>propriedade</b> pode ter - Any, boolean, float, instance, integer, string or symbol.
<b>Valor</b>		É um <b>termo</b> que é relacionado para a definição corrente através de <b>propriedades</b> ou <b>facetats</b> .