



**UNIVERSIDADE FEDERAL DE JUIZ DE FORA  
INSTITUTO DE CIÊNCIAS EXATAS  
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO**

**Interoperabilidade de Ontologias  
Integração no domínio de Equações  
Diferenciais Ordinárias**

**Henrique Azevedo Cardoso**

JUIZ DE FORA  
JULHO, 2011

# **Interoperabilidade de Ontologias Integração no domínio de Equações Diferenciais Ordinárias**

**Henrique Azevedo Cardoso**

Universidade Federal de Juiz de Fora  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação  
Bacharel em Ciência da Computação

Orientadora: Fernanda Claudia Alves Campos

JUIZ DE FORA  
JULHO, 2011

Interoperabilidade de Ontologias  
Integração no domínio de Equações Diferenciais Ordinárias

Henrique Azevedo Cardoso

MONOGRAFIA SUBMETIDADA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

---

Prof. Dra. Fernanda Claudia Alves Campos - Orientadora

---

Prof. Dra. Regina Maria Maciel Braga

---

Prof. Esp. Francilene Barbosa dos Santos Silva

JUIZ DE FORA, MG – BRASIL  
JULHO, 2011

## Resumo

O rápido crescimento da Web tem aumentado o número de fontes de conhecimento e quantidade de acesso ao seu conteúdo. A Web semântica surge nesse contexto apresentando uma forma de pesquisa, integração e processamento da informação para as máquinas. As ontologias vêm como solução para representar o domínio da informação e suas relações, e sua interoperabilidade possibilita a integração do conhecimento. Neste trabalho será apresentado o processo de desenvolvimento da ontologia ODEq e sua integração com outra do projeto MONET. As ontologias foram desenvolvidas usando a linguagem OWL.

Palavras-chave: Semântica. Ontologias. Web.

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>1</b>
1.1	MOTIVAÇÃO.....	1
1.2	OBJETIVOS.....	1
1.3	METODOLOGIA.....	1
1.4	ORGANIZAÇÃO DA MONOGRAFIA.....	2
<b>2</b>	<b>ONTOLOGIAS.....</b>	<b>3</b>
2.1	INTRODUÇÃO.....	3
2.2	APLICAÇÕES.....	3
2.3	TIPOS DE ONTOLOGIAS.....	4
2.4	COMPOSIÇÃO DE ONTOLOGIAS.....	6
2.5	LINGUAGENS DE ONTOLOGIAS.....	7
2.6	METODOLOGIAS DE DESENVOLVIMENTO.....	9
2.7	FERRAMENTAS.....	14
2.7.1	Protege.....	14
<b>3</b>	<b>INTEROPERABILIDADE DE ONTOLOGIAS.....</b>	<b>16</b>
3.1	INTRODUÇÃO.....	16
3.2	HETEROGENEIDADE DE DADOS.....	17
3.3	ABORDAGENS PARA USO DE ONTOLOGIAS.....	18
3.3.1	Única Ontologia .....	18
3.3.2	Múltiplas Ontologias.....	18
3.3.3	Abordagem Híbrida .....	19
3.4	MECANISMOS DE INTEROPERABILIDADE ENTRE ONTOLOGIAS.....	21
3.4.1	Combinação de Ontologias.....	21
3.4.2	Alinhamento de Ontologias.....	22
3.4.3	Integração de Ontologias.....	22
3.4.4	Mapeamento de Ontologias.....	23
3.5	COMPATIBILIDADES DE TERMOS.....	24
3.6	SIMILARIDADE SEMÂNTICA.....	25
3.7	FERRAMENTAS.....	28
3.7.1	Prompt.....	29
<b>4</b>	<b>ALINHAMENTO ODEq E GAMS.....</b>	<b>31</b>
4.1	INTRODUÇÃO.....	31
4.2	ONTOLOGIA ODEq .....	31
4.2.1	Metodologia de Desenvolvimento.....	34
4.3	PROJETO MONET.....	36
4.3.1	GAMS.....	37
4.4	ALINHAMENTO GAMS-ODEq.....	38
4.4.1	Processo de alinhamento.....	40
<b>5</b>	<b>CONCLUSÃO.....</b>	<b>42</b>
5.1	TRABALHOS FUTUROS.....	42
	<b>REFERÊNCIAS.....</b>	<b>44</b>
	<b>APÊNDICE.....</b>	<b>46</b>
	<b>Apêndice 1 – Documentação da ontologia ODEq.....</b>	<b>46</b>

## 1 INTRODUÇÃO

### 1.1 MOTIVAÇÃO

Com o rápido avanço tecnológico o acesso a informações tornou-se mais fácil nos dias de hoje, principalmente na World Wide Web. Há um grande volume de informações, que estão organizadas sem uma estruturação semântica, isso faz com que o acesso seletivo à informação necessite da intervenção humana.

Uma solução para a organização estrutural está no uso de ontologias. Segundo Gruber (1995) uma ontologia é uma especificação explícita de uma conceitualização compartilhada. Seu vocabulário é livre de ambiguidades e passível de processamento automático (FELICÍSSIMO, 2004).

Uma ontologia identifica classes que são compostas por propriedades e são hierarquizadas. 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.

Ontologias surgem como solução para representação do domínio e suas relações. A necessidade de se estender o escopo do domínio da ontologia é um problema quando vários sistemas são integrados, ou um sistema apenas integra várias ontologias. Na literatura são encontradas várias técnicas e ferramentas para prover a interoperabilidade entre ontologias e solucionar a questão do relacionamento entre ontologias.

O tema foi escolhido após o contato como bolsista de Iniciação Científica no NEnC (Núcleo de Pesquisa em Engenharia do Conhecimento) em um projeto de pesquisa, já que o grupo vem trabalhando com Ontologias desde 2008.

### 1.2 OBJETIVOS

Descrever a ontologia ODEq, – **Ordinary Differential Equations Ontology** da área de Matemática e o relacionamento dessa ontologia com outra do projeto Monet (MONET, 2006).

### 1.3 METODOLOGIA

Para a realização do trabalho foi feita uma revisão bibliográfica dos principais temas. Após o estudo de contextos, conceitos, classificações, metodologias e ferramentas, o estudo de alinhamento de duas ontologias foi abordado e aplicado a

ontologia ODEq, desenvolvida neste trabalho, e a ontologia GAMS, do projeto MONET.

Foram apresentadas metodologias de desenvolvimento, e entre várias propostas na literatura foi escolhida a QDAOntology (PALAZZI, 2010).

Para abordar o estudo de Ontologias foi utilizada a ferramenta de desenvolvimento Protégé entre outras devido a sua maior utilização em trabalhos acadêmicos. Protégé 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).

Serão também apresentadas ferramentas e técnicas que possibilitam a interoperabilidade entre ontologias, em especial o PROMPT, um plug-in do Protégé.

#### 1.4 ORGANIZAÇÃO DA MONOGRAFIA

O primeiro capítulo será introdutório, o segundo abordará Ontologias em geral como definição, conceitos, linguagens, classificação e ferramentas de desenvolvimento com ênfase no Protégé, ferramenta que será usado no estudo de caso específico.

O terceiro capítulo especificará mecanismos e metodologias de interoperabilidade de Ontologias e ferramentas, com enfoque no PROMPT.

O quarto capítulo apresentará o nosso estudo de caso. As duas ontologias serão apresentadas assim como as pontes semânticas geradas pelo alinhamento. A ferramenta PROMPT será introduzida e a metodologia de integração proposta será abordada.

O quinto capítulo apresentará a conclusão e possíveis trabalhos futuros.

## 2 ONTOLOGIAS

### 2.1 INTRODUÇÃO

Com a necessidade de estruturar e organizar de uma forma melhor o conhecimento em um contexto de aumento exponencial de dados na Web, passou-se a usar Ontologias. Segundo Gruber (1995) uma ontologia é uma especificação explícita de uma conceitualização compartilhada. O termo especificação explícita diz respeito à definição clara e objetiva dos conceitos e suas restrições utilizadas, e ser compartilhada é ser consensual, ou seja, aceita por um grupo. Seu vocabulário é livre de ambigüidades e passível de processamento automático (FELICÍSSIMO, 2004). Esta outra definição complementa a anterior afirmando que a ontologia deve ser capaz de ser lida por uma máquina.

Uma Ontologia é constituída de um vocabulário que descreve certa realidade, mais um conjunto de assertivas que dão um significado semântico ao vocabulário. Para fazer essas assertivas normalmente é utilizada a Lógica de Primeira Ordem, onde o vocabulário aparece em predicados lógicos trazendo conceitos e relações.

Segundo Gruber (1995) Ontologia deve ter clareza e ser objetiva, para isso ela é definida sobre um formalismo composto de axiomas lógicos. Deve ter coerência, isto é, nenhuma sentença que possa ser inferida a partir dos axiomas pode contradizer alguma definição ou exemplo informal do contexto que ela está inserida. Ela deve ser projetada de forma a suportar a inserção de novos termos e conceitos sobre os dados já existentes, podendo ser extensível. Outra característica importante que deve ser observada em nível de projeto é a possibilidade de se integrar a Ontologia futuramente, para isso devem-se definir somente os termos que são essenciais para a comunicação do conhecimento consistente com a teoria e os dados representados.

Gómez-Perez, Fernández-López e Corcho (1999) acrescentam outras características às ontologias:

- identidade: as classes nas ontologias devem ser disjuntas;
- modularidade: devem permitir a integração de ontologias;
- os nomes devem ter um padrão;
- conceitos similares devem ser representados como subclasses.

### 2.2 APLICAÇÕES

Podemos descrever quatro aplicações mais gerais para ontologias segundo Abu-Hanna e Jansweijer (1994):

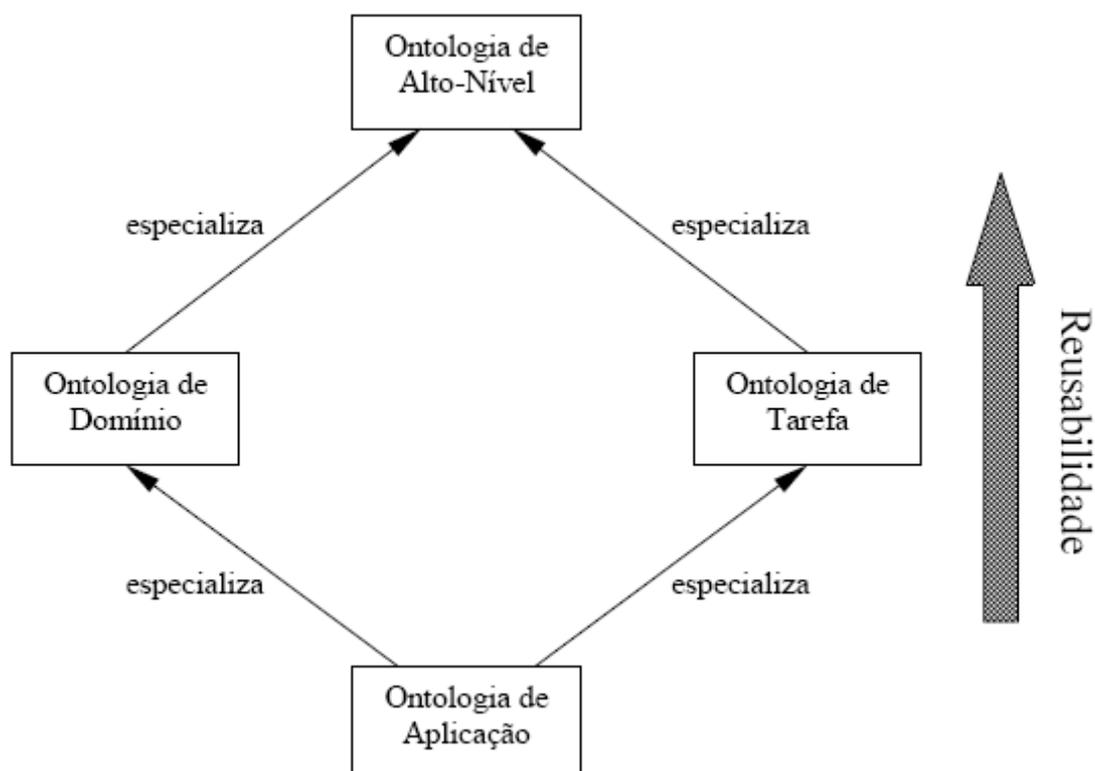
- **Compartilhamento:** O propósito seria tornar o acesso a informação mais fácil e difundir o conhecimento. Para isso os engenheiros devem se comprometer a usar os mesmos termos e conceitos determinados pela ontologia.
- **Aquisição de conhecimento:** A ontologia conterá a definição de termos para uma aplicação, e ferramentas de aquisição do conhecimento usarão essas definições para interagir com um domínio específico. A função deste tipo de ontologia então é evitar erros na aquisição do conhecimento aplicando a terminologia definida.
- **Organização do conhecimento:** A ontologia pode organizar melhor a base de dados para então serem mais bem acessadas, modificadas e reusadas.
- **Auxílio de raciocínio:** A ontologia pode ser usada junto com uma base de dados para ajudar na realização de uma tarefa.

### 2.3 TIPOS DE ONTOLOGIAS

Temos vários tipos de Ontologias, algumas podem ser mais compartilhadas e gerais, outras mais destinadas a uma aplicação ou problema específico. A utilidade da Ontologia é definida pela utilidade da teoria que ela representa (GRUBER, 1995). Não existe, portanto, um jeito correto de modelar um domínio, devemos ter em mente a aplicação que a ontologia servirá e até onde ela poderá se estender. A modelagem do domínio deve se manter fiel ao objeto do mundo real.

Segundo Guarino (1998) podemos classificar ontologias de acordo com seu nível de generalidade (Figura 1):

- **Ontologias de Alto Nível:** Descrevem domínios bem gerais como o tempo por exemplo. São ontologias com alto índice de reuso.
- **Ontologias de Domínio:** São mais específicas que as de alto nível e descrevem domínios genéricos como Economia por exemplo.
- **Ontologias de Tarefas:** São mais específicas do que as ontologias de domínio e descrevem uma tarefa ou atividade genérica como Planejamento por exemplo.
- **Ontologias de Aplicação:** São ontologias que descrevem certa tarefa em um determinado domínio.



**Figura 1. Representação das diferentes classes de ontologias de acordo com sua generalidade (GUARINO, 1998).**

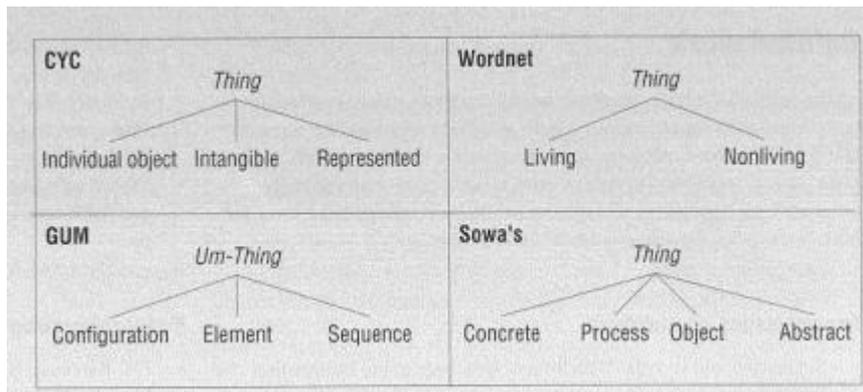
A figura 1 relaciona as classes de ontologias com o nível de reusabilidade de cada classe. Uma ontologia de Alto-Nível é menos especializada do que ontologias de domínio ou de tarefas, isso faz com que seja mais fácil de ser reutilizada. As ontologias de aplicação são mais especializadas, definindo termos menos gerais, e por isso estão no nível menor de reusabilidade.

As ontologias de alto-nível são mais reutilizadas, por isso são encontradas maiores informações sobre elas na literatura. Entre as mais conhecidas estão as ontologias SOWA, Sowa<sup>1</sup> (apud FREITAS, 2006), e Cyc. Lenat e Guha<sup>2</sup> (apud FREITAS, 2006), buscam representar raciocínio de senso comum, enquanto WordNet, Miller<sup>3</sup> (apud FREITAS, 2006), contém sinônimos e é utilizada em sistemas de processamento de linguagem natural (Figura 2).

<sup>1</sup> [Miller 95] Miller, G. 1995. WordNet: A Lexical Database for English. Communications of the ACM. 38(11):39-41. EUA.

<sup>2</sup> [Lenat & Guha 90] Lenat, D. B. and R. V. Guha. 1990. Building Large Knowledge Based Systems. Reading, Massachusetts: Addison Wesley.

<sup>3</sup> [Miller 95] Miller, G. 1995. WordNet: A Lexical Database for English. Communications of the ACM. 38(11):39-41. EUA.



**Figura 2. Ontologias de alto-nível e suas classes primitivas (FREITAS, 2006)**

## 2.4 COMPOSIÇÃO DE ONTOLOGIAS

A Ontologia se baseia em Classes ou conceitos. As relações entre as classes são definidas com o termo de Slots e descrevem aspectos dos conceitos. Uma ontologia e seu conjunto de instâncias individuais formam uma base do conhecimento.

As classes são o foco da ontologia, elas podem ter subclasses que são mais específicas do que a superclasse, por exemplo, os conceitos “Livro” e “Romance” são respectivamente classe e subclasse.

Os Slots descrevem propriedades das classes, por exemplo, o número de páginas, ano de lançamento entre outros.

A figura 3 mostra parte do código OWL da ontologia Pizza, um exemplo de ontologias da ferramenta Protegé. Podemos observar a classe “CoberturadeTomate”, ela contém a propriedade “temTempero”. A ontologia descreve ainda um possível valor para a propriedade que é “Suave”. Abordaremos ferramentas e linguagens de ontologias mais adiante.

```

</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. Exemplo de código OWL da ontologia Pizza, um exemplo do Protégé.**

Segundo Noy et al (2001) o desenvolvimento de ontologias inclui:

- Definição de classes na ontologia
- Criação de uma taxonomia (superclasse-subclasse) hierárquica.
- Definição de Slots e atribuição dos possíveis valores para eles.
- Preenchimento dos valores dos Slots para as instâncias.

## 2.5 LINGUAGENS DE ONTOLOGIAS

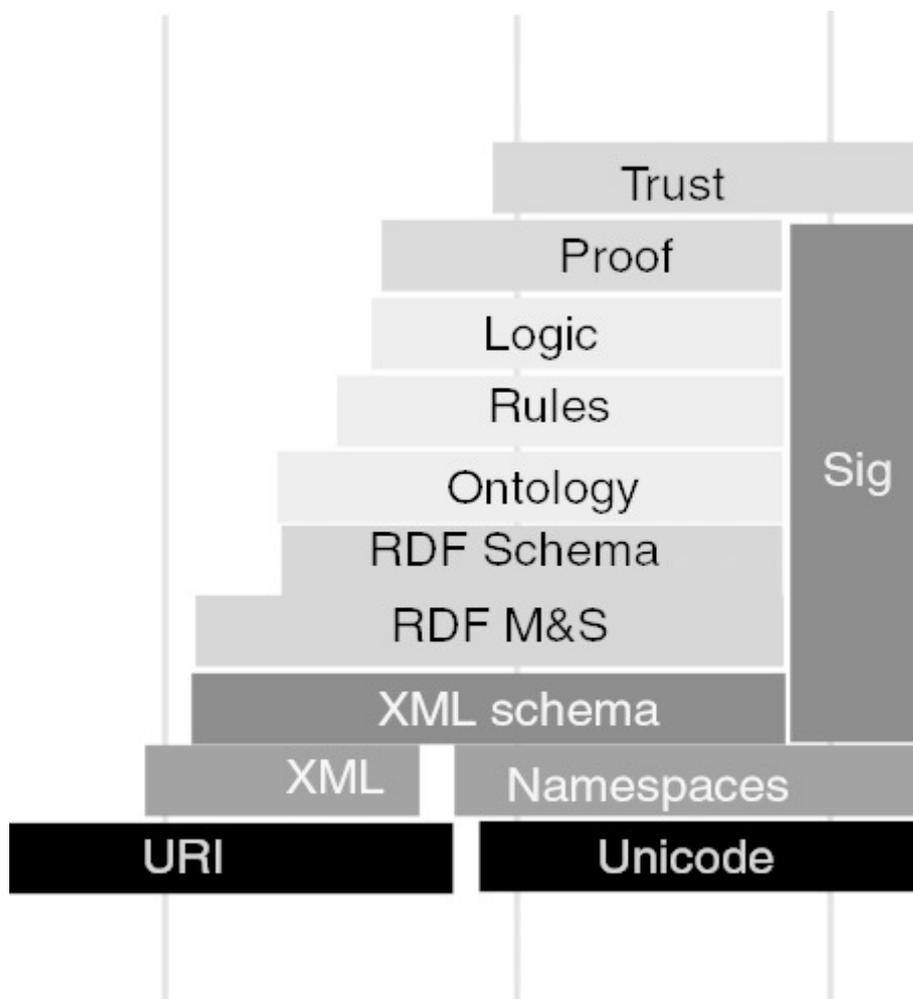
Algumas tecnologias foram criadas antes da Ontologia para resolver problemas semânticos e são a base das Ontologias. A princípio o URI (Uniform Resource Identifier) e o Unicode foram criados. O URI consiste em um conjunto de caracteres para identificação de um recurso físico ou abstrato. O Unicode é um sistema de codificação de caracteres projetado para suportar o intercâmbio na web, processar e mostrar textos escritos de diversos idiomas do mundo moderno. Ele define um conjunto universal de caracteres. O XML (eXtensible Markup Language) fica, logicamente, acima dessas duas tecnologias.

Um namespace de XML é uma coleção dos nomes, identificada por uma referência de URI que são usados em documentos XML como tipo de elementos e nomes de atributo. O XML permite a criação de tags, e assim o usuário pode definir uma estrutura arbitrária, essa linguagem é utilizada para transporte de dados mas tem significado apenas sintático, não traz nenhuma informação semântica aos dados.

Já XML Schema expressa vocabulários compartilhados, permitindo que máquinas possam realizar referências feitas pelos humanos. O XML Schema fornece

meios para definir a estrutura, o índice e a semântica de documentos XML. É um padrão complexo que especifica estruturas e mecanismos para validar o conteúdo dos elementos do XML, especificando uma sofisticada tipagem para cada elemento (SUN, 2002). A figura 4 representa a estrutura dessas tecnologias de acordo com seu nível semântico.

A RDF (Resource Description Framework) é uma linguagem de propósito geral para representar informações na Web. A RDF pode ser usada em conjunto com as tags XML descrevendo relações e conceitos entre elas. É composta por uma tripla: sujeito, verbo e objeto de uma sentença elementar. A linguagem RDF Schema vai mais além e descreve um conjunto de propriedades e não somente as propriedades, assim ela permite a criação de classes e objetos destas classes. A estrutura lógica dessas tecnologias é apresentado na figura 4.



**Figura 4. Representação das tecnologias da Web Semântica.**  
(Fonte: [www.w3.org/2001/Talks/0228-tbl/slide5-0.html](http://www.w3.org/2001/Talks/0228-tbl/slide5-0.html))

Como podemos observar na figura 4 a ontologia é formada por sentenças lógicas. Essas seguem certas regras pré-estabelecidas, que são os axiomas.

OWL é um documento XML, RDF, RDFS (RDF Schema). Originado de vários projetos: DARPA (DAML) + EU Project (OIL). Parecido com RDF porém OWL é uma linguagem com novas estruturas específicas para ontologias. Enriquece o RDFS adicionando cardinalidade, equivalência, e importação de recursos e restrições (PALAZZI, 2010).

OWL é subdividida em três linguagens de acordo com sua expressividade:

- OWL Lite – É mais indicada para usuários iniciantes, pois é mais simples e não contém todas as funcionalidades que a OWL pode assumir. Atributos podem ter transitividade, simetria, atributos inversos, propriedades funcionais (se  $P(x,y) \wedge P(y,x) \Rightarrow x=y$ ), funcionais inversas (se  $P(x,y) \wedge P(z,x) \Rightarrow x=z$ ) e papéis (FREITAS, 2006).
- OWL DL - Tem maior expressividade que OWL Lite, as classes podem ser construídas por união, interseção e complemento, pela enumeração de instâncias e podem ter disjunções. O nome DL se deve a Lógica de Descrições (Description Logic) que é uma das bases da OWL, e neste nível a expressividade da linguagem atinge o mesmo nível da Lógica de Descrições.
- OWL Full – Tem a toda a expressividade da OWL mais a flexibilidade da RDF. Nesta versão metaclasses podem ser criadas, e este nível de complexidade não permite que se faça uma computação.

## 2.6 METODOLOGIAS DE DESENVOLVIMENTO

Durante a fase de desenvolvimento da ontologia surgem questões que devem ser resolvidas como a ferramenta que será utilizada, a linguagem, a generalização da ontologia, o ciclo de vida da ontologia entre outras. Neste contexto surge a Engenharia Ontológica ou Engenharia de Ontologias que propõe métodos e metodologias para auxiliar a construção de ontologias.

Não existem métodos que sejam amplamente aceitos e utilizados, apenas propostas. Essas propostas buscam identificar e especificar a ontologia. Em Palazzi (2010) são listados várias metodologias de processo de desenvolvimento de ontologias.

As metodologias podem se dividir em três subtipos segundo Jarrar <sup>4</sup>(apud PALAZZI, 2010):

- Metodologia de Etapas: Nesta metodologia as fases do desenvolvimento contêm etapas e ações que devem ser seguidas. Por exemplo, as metodologias Methontology e On-To-Knowledge
- Metodologia de Modelagem: O domínio é o foco desta metodologia. A maior preocupação está nas classes, propriedades e suas relações. Por exemplo, a metodologia OntoClean (ONTOCLEAN, 2010)
- Metodologia de Engenharia: Segue o paradigma de Orientação a Objetos e os princípios da Engenharia de Software. A reusabilidade, escalabilidade, manutenibilidade e outros aspectos de Engenharia são o foco deste tipo de metodologia.

É possível observar características em comum entre as várias metodologias existentes. Há sempre uma fase de viabilização do processo e especificação do mesmo, uma fase de desenvolvimento, e uma fase de avaliação ou revisão da ontologia. Essas fases ficam mais claras no quadro apresentado por (PALAZZI, 2010). Em algumas metodologias se indica a construção de uma primeira versão da ontologia, outras apenas a definição formal e lógica da ontologia. Algumas propõem uma documentação, e quase todas tem a fase de integração da ontologia com alguma já existente, o que é a base para se construir conhecimento confiável.

<b>Metodologia</b>	<b>Especificação dos requisitos</b>	<b>Modelagem Conceitual</b>	<b>Formalização</b>	<b>Implementação</b>	<b>Manutenção</b>	<b>Integração</b>	<b>Avaliação</b>	<b>Documentação</b>
<b>USCHOLD E KING</b>	Identifica o propósito e escopo da ontologia	Define os conceitos e relacionamentos textualmente	Ausente	Codificação em linguagem formal	Ausente	Integração com ontologias existentes	Questões de competência	Descreve o processo de construção
<b>TOVE</b>	Cenários motivacionais	Ausente	Especificação dos axiomas	Linguagem de primeira ordem KIF	Ausente	Ausente	Verificação da completude da ontologia	Ausente

<sup>4</sup> JARRAR, M. Towards Methodological Principles for Ontology Engineering. 2005. Tese (Doutorado) – Vrije Universiteit Brussel.

<b>METHONTOLOGY</b>	Definição do escopo da ontologia	Atividades envolvendo a conceitualização da ontologia	Modelo conceitual	Ambiente de desenv. adequado à integração	Em todo o ciclo	Integração com ontologias de alto nível	Verificação e validação da ontologia	Em todas as fases.
<b>CYC</b>	Extração manual do conhecimento	Ausente	Ausente	Codificação e extração do conhecimento	Ausente	Ausente	Ausente	Ausente
<b>ON-TO-KNOWLEDGE</b>	Definição dos objetivos, domínio e escopo da ontologia	Construção de uma ontologia modelo	Ausente	Ausente	Aplicação e evolução da ontologia	Utilização de reutilização do conhecimento através de ontologias	Verificação do documento de especificação e questões de competência	Ausente
<b>Framework para Ontologias Multilíngües</b>	Ausente	Construção de uma pequena ontologia núcleo	Definição dos conceitos e relacionamentos	Ausente	Ausente	Ausente	Avaliação dos termos por especialistas	Ausente
<b>KUP</b>	Definição do escopo e levantamento dos requisitos	Ausente	Ausente	Construção do projeto ou	Adequação às mudanças no projeto	Integração de novos requisitos	Ausente	Ausente
<b>DILIGENT</b>	Especificação do conhecimento	Construção de uma ontologia consensual (1ª versão)	Ausente	Construção de uma ontologia local	Revisão das ontologias locais e a compartilhada	Identificação de similaridades na ontologia local	Verificação de mudanças na ontologia compartilhada	Documentação da ontologia compartilhada
<b>Kactus</b>	Lista de requisitos	Identificação dos termos relevantes	Ausente	Ausente	Ausente	Ausente	Ausente	Ausente
<b>Método 101</b>	Definição de questões de competência	Definição de classes, propriedades, restrições e instâncias	Através de frames	Ferramenta Protégé	Ausente	Considera a reutilização de ontologias	Ausente	Através do protégé

**Quadro 1. Comparação das características de metodologias de desenvolvimento de Ontologias (PALAZZI, 2010).**

Neste trabalho foi utilizada a **QDAontology**: Quality Driven Approach for e-Science Ontologies. Esta abordagem é composta por etapas, atividades, participantes, artefatos e características de qualidade. Ressalta-se que o processo de desenvolvimento adotado é centrado em um modelo evolutivo e, portanto, os ciclos podem se repetir a cada evolução da ontologia (PALAZZI, 2010).

A figura 5 mostra as etapas de desenvolvimento, segundo esta abordagem são seis: Especificação, Conceitualização, Formalização, Implementação, Integração e Evolução. A cada etapa são realizadas atividades, estas geram artefatos que servem

como auxílio e aumento da qualidade. Ainda são descritos os participantes que estarão envolvidos nessas atividades. A cada ciclo é gerada uma versão da ontologia, lembrando que pode e deve haver mais de um ciclo uma vez que esta é uma abordagem evolutiva. Os ciclos se encerram quando a ontologia obtida atende aos atributos especificados à ela no início do processo.

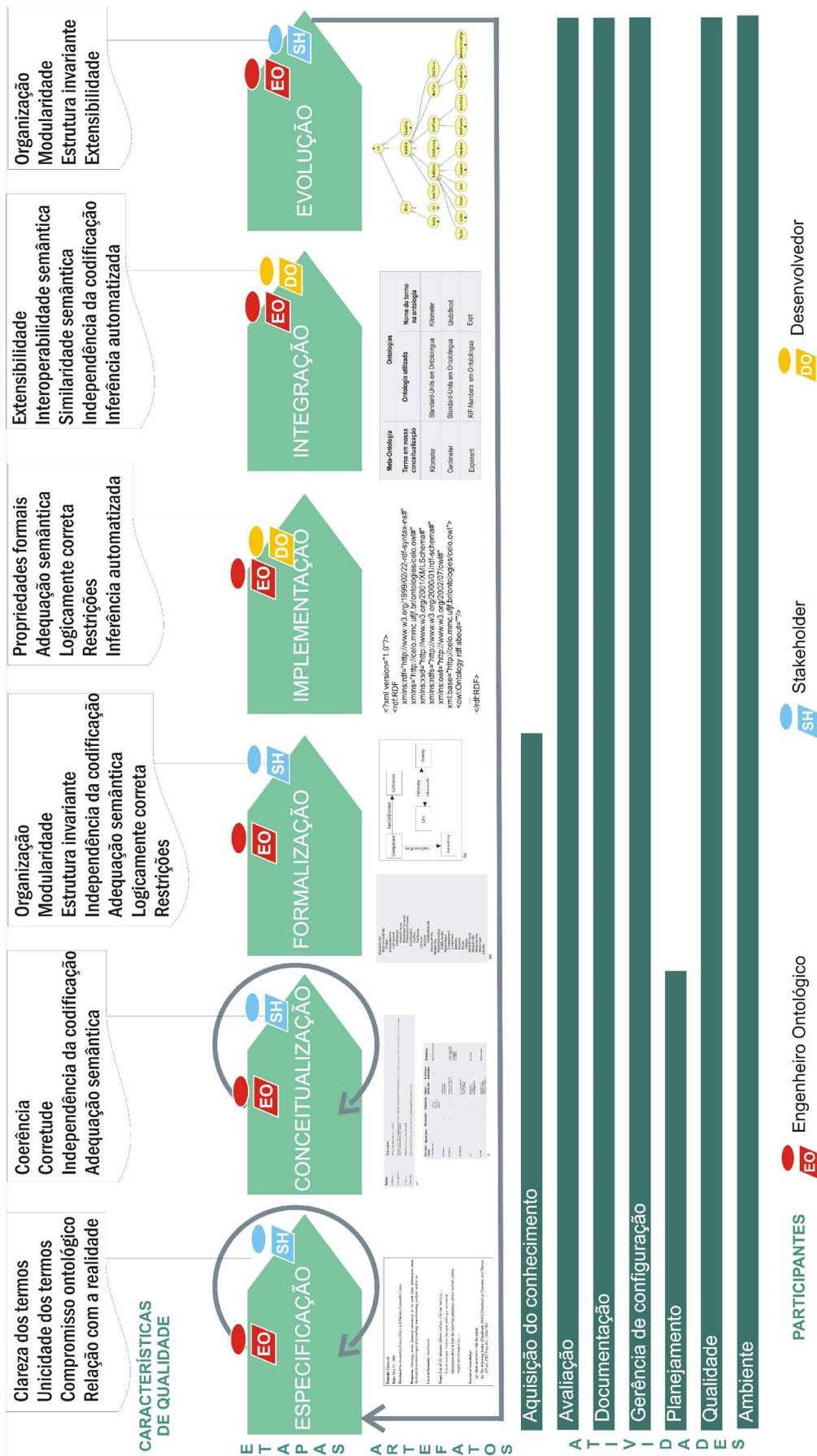


Figura 5. QDAOntology

## 2.7 FERRAMENTAS

Ontologia se tornou alvo de estudo por vários grupos com interesse distintos pelas vantagens que ela traz, e isto propiciou a criação de várias ferramentas que dão suporte ao manuseio de ontologias. Editores, servidores, repositórios entre outros estão sendo desenvolvidos e publicados a todo o momento com o objetivo de facilitar a criação, a edição, o reuso e o compartilhamento de ontologias.

Os editores OntoEdit, o WebODE e o Protégé são alguns exemplos de editores que obtiveram êxito e são mais conhecidos. Eles trabalham com a mesma linguagem, lógica descritiva.

### 2.7.1 PROTÉGÉ

Foi desenvolvido pelo Departamento de Informática Médica da Universidade de Stanford desde os anos 80 com a necessidade de gerar ontologias na área de medicina. Inicialmente era um sistema voltado para a área de Oncologia.

Ela foi evoluindo e se tornando mais genérica, possibilitando aquisição de conhecimento diretamente de especialistas de domínios - com menos dependência de engenheiros de conhecimento. Entre suas funções encontramos: permitir diversos formalismos e estratégias de inferência, integrar tarefas (aquisição de ontologias e instâncias, ambiente de teste com inferência) num mesmo ambiente, criar automaticamente formulários para entrada de conhecimento e combinar de ontologias (FREITAS, 2006).

Com o aumento do número de usuários o grupo de desenvolvedores resolveu abrir o código do Protégé, e assim o software passou a ser integrável com outras aplicações através de componentes que são conectados ao sistema. O OntoViz é um exemplo de componente adicionado que possibilita a visualização gráfica da ontologia, mais adiante apresentaremos um exemplo usando esta ferramenta.

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 et al 2000).

Noy et al (2000) ainda apresenta outras características:

- A linguagem axiomática PAL (Protégé Axiomatic Language), que permite a inserção de restrições e axiomas que incidem sobre as classes e instâncias de uma ou mais ontologias.

- A geração de arquivos de saída alteráveis, permitindo que sejam implementados componentes para traduzir o conhecimento para outros formalismos através das metaclasses. Atualmente podem ser criadas classes e instâncias em CLIPS – a base de conhecimento é gerada nativamente para esse motor de inferência, o mais popular quando se iniciou a construção do Protégé, nos anos 80 -, Jess, F-Logic, Prolog, RDF, OIL, XML, Topic Maps (linguagem padrão ISO para descrever estruturas de conhecimento em páginas da Web, nos mesmos moldes de RDF e OIL) – todos estes últimos através de componentes e metaclasses específicas. A saída pode ser armazenada ainda em outros formatos, como bancos de dados relacionais, ainda que as tabelas geradas não sejam normalizadas.
- Uma excelente interface para entrada de conhecimento, incluindo um gerador automático de formulários para as classes definidas.

Na figura 6 temos uma imagem gerada pelo plug-in OWLViz que pode ser instalado no Protégé, ele representa a hierarquia de classes da ontologia. Podemos ver a classe-pai relacionada através da expressão “is - a” com as classes-filha. Esta ontologia será apresentada mais adiante.



Figura 6. Parte da Ontologia desenvolvida

### 3 INTEROPERABILIDADE DE ONTOLOGIAS

#### 3.1 INTRODUÇÃO

Na seção 2.2 foram apresentadas algumas áreas onde ontologias são aplicadas. Para melhorar a qualidade ou aumentar o escopo dessas aplicações se torna interessante fazer um tipo de relacionamento entre as ontologias. Na Web se encontra uma grande quantidade de dados e informações tornando-se difícil a busca e a utilização pelos diversos sistemas existentes. As ontologias nos ajudam a organizar, estruturar e representar o conhecimento, para facilitar a troca de informações entre sistemas diferentes seria necessário criar um relacionamento entre os dados.

A interoperabilidade de ontologias esta sendo largamente usada, em Souza (2007) são descritos diversos sistemas que utilizam processos de interoperabilidade para diferentes objetivos.

O OBSERVER (Ontology Based System Enhanced with Relationships for Vocabulary heterogeneity Resolution) MENA<sup>5</sup> (apud SOUZA, 2007) é um exemplo de um sistema que utiliza a integração de ontologias. Seu objetivo é o processamento de consultas utilizando repositórios e retirando suas informações das pré-ontologias.

O SIMS (Services and Information Management for Decision Systems) Arens<sup>6</sup> (apud SOUZA, 2007) é outro exemplo de sistema que utiliza a integração de ontologias para gestão da informação. A partir de uma ontologia de domínio são integradas ao sistema novas fontes de informação como: banco de dados, bases de conhecimento entre outros.

As aplicações citadas já são realidade, e grandes empresas utilizam e investem em gestão do conhecimento. De acordo com Souza (2007) podemos citar Schlumberger, Daimler Chrysler e BTelecom que investem neste tipo de tecnologia e incentivam a pesquisa e pequenas empresas de desenvolvimento dessa área.

---

<sup>5</sup> MENA, E., KASHYAP, V., SHETH, A., et al., 1996, "OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies". In: 1st IFCIS International Conference on Cooperative Information Systems (CoopIS '96), pp. 14-25, Brussels, Bélgica, Junho de 1996.

<sup>6</sup> ARENS, Y., KNOBLOCK, C.A., SHEN, W.M., 1996, "Query reformulation for dynamic information integration", Journal of Intelligent Information Systems, v. 6, n. 2-3 (1996), pp. 99-130.

### 3.2 HETEROGENEIDADE DE DADOS

Existem várias metodologias de desenvolvimento de ontologias, mostramos no capítulo passado um quadro comparativo entre algumas metodologias e foi apresentada a metodologia QDAontology. Porém mesmo com uma metodologia bem definida ainda enfrentamos um problema, a incompatibilidade de termos utilizados.

A fase de integração necessita de um consenso entre especialistas do domínio para definição dos termos, e as ferramentas de integração devem dar suporte e ajudar a resolver este problema.

A heterogeneidade de dados é a maior barreira encontrada, ela pode ser dividida em dois tipos:

- Heterogeneidade estrutural - Refere-se à capacidade dos sistemas representarem suas entradas segundo diversas representações e modelos. Existem mecanismos que avaliam a heterogeneidade estrutural segundo as hierarquias de conceitos e relacionamentos encontrados nas ontologias, visando encontrar estruturas equivalentes. No caso mais simples, uma ontologia descreve uma hierarquia de conceitos relacionados por meio de regras, enquanto que, em casos mais sofisticados, são adicionados axiomas adequados para expressar relacionamentos mais complexos e para restringir a interpretação desejada de seus conceitos (GUARINO, 1998).
- Heterogeneidade sintática - Tem como elementos principais: metadados, termos e linguagens, que são utilizados na representação do conhecimento. A heterogeneidade pode ser resolvida com a utilização, por exemplo, de dicionários de termos léxicos.
- Heterogeneidade semântica – Ocorre quando há mais de uma interpretação para um mesmo termo. Homônimos e sinônimos são a maior causa, por exemplo, o termo manga pode ser sentido da fruta ou da parte da camisa. Há também o caso em que a escala utilizada para representar um termo difere entre duas ontologias, o tempo é

um exemplo deste tipo de heterogeneidade semântica pois possui várias escalas a ser representado (mês, dia, ano, horas).

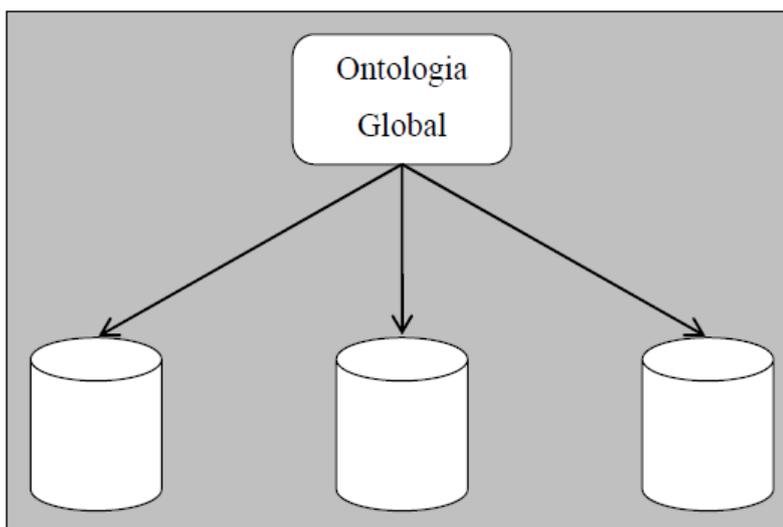
Há várias linhas de pesquisa que tratam da compatibilização entre dados. A similaridade semântica é a medida que auxilia o tratamento dos dois tipos de heterogeneidade. Mecanismos como cálculo de distância em strings, na análise sintática, e cálculo de diferença estrutural usando grafos são os mais utilizados, abordados mais sobre similaridade e compatibilidade de termos adiante.

### 3.3 ABORDAGENS PARA USO DE ONTOLOGIAS

Vamos ver três tipos de abordagens de utilização de ontologias para entender melhor o contexto da Interoperabilidade de Ontologias.

#### 3.3.1 Única Ontologia

Segundo Souza (2007) nesta abordagem, todas as fontes de informação estão relacionadas com a ontologia global (Figura 7).



**Figura 7. Abordagem de uma única ontologia (SOUZA, 2007)**

Com esta abordagem, a partir de uma ontologia geral é inserida toda a informação semântica. O vocabulário é o mesmo em toda ontologia que é obtida independentemente da fonte de informação, fazendo a tradução da fonte para o vocabulário da ontologia.

A desvantagem está na hora de se evoluir ou estender a ontologia. Por exemplo, se ontologias fazem referência a um mesmo catálogo de itens

heterogêneos, a geração da a ontologia global combinando os diferentes itens será um tarefa custosa.

### 3.3.2 Múltiplas Ontologias

Nesta abordagem, a semântica de cada fonte de informação é descrita por sua própria ontologia (SOUZA, 2007).

Com esta arquitetura cada fonte de ontologia é integrada, sem ter a necessidade de nenhuma relação entre as ontologias. Essa arquitetura pode simplificar mudanças, por exemplo, modificações em uma fonte de informação ou adição e remoção de fontes (SOUZA, 2007) (Figura 8).

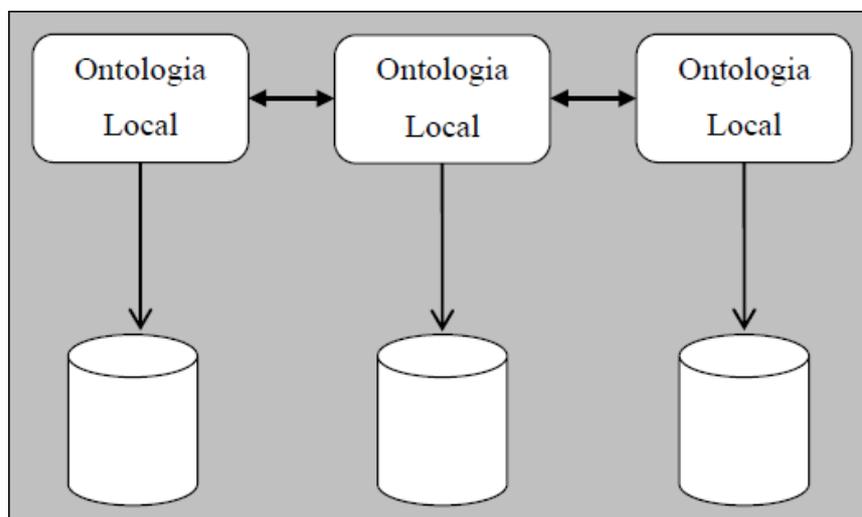


Figura 8. Abordagem de múltiplas ontologias (SOUZA, 2007)

### 3.3.3 Abordagem Híbrida

Na abordagem com uma ontologia temos uma boa garantia na consistência dos dados, pois há apenas um único vocabulário. Porém a operação de extensão da ontologia se torna um processo custoso.

Na abordagem de múltiplas ontologias é mais fácil estender o escopo da ontologia, porém há o problema de heterogeneidade de dados ao se fazer o alinhamento das ontologias envolvidas, não é um problema exclusivo desta área e sim um problema inerente à Web Semântica.

Como solução dos problemas dessas duas abordagens surge a abordagem híbrida. Para fazer com que as ontologias fontes sejam comparáveis com as

outras, elas são construídas utilizando um vocabulário global compartilhado. O vocabulário compartilhado contém os termos básicos (primitivas) do domínio. Para construir os termos complexos de uma ontologia fonte, as primitivas são combinadas por alguns operadores. Uma vez que os termos de cada ontologia fonte são baseados nas primitivas, os termos se tornam mais fáceis de comparar do que os descritos na abordagem múltipla. Algumas vezes, o vocabulário compartilhado também é uma ontologia (SOUZA, 2007) (Figura 9).

A evolução e extensão se tornam fáceis quando se utiliza esta abordagem, com a utilização do processo de integração de ontologias, e o vocabulário compartilhado traz maior consistência.

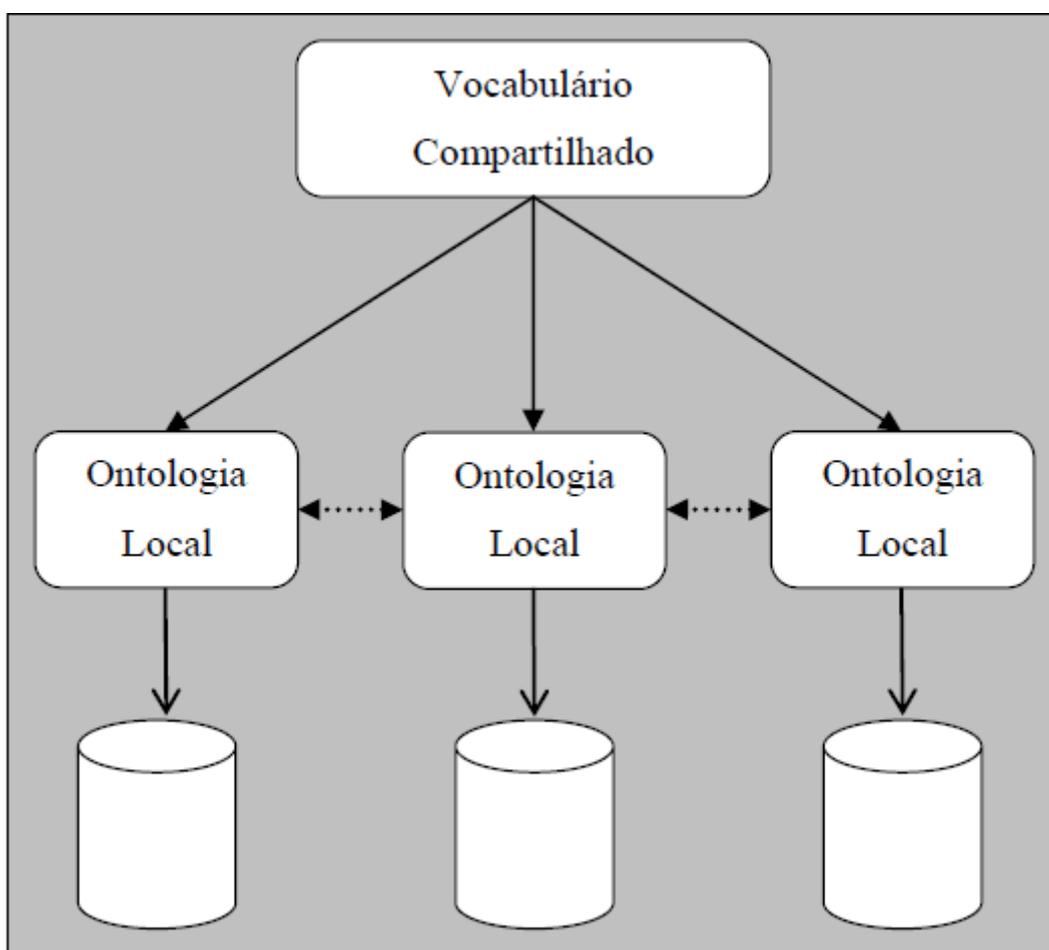


Figura 9. Abordagem híbrida (SOUZA, 2007)

O problema da abordagem híbrida é que as ontologias existentes não podem ser facilmente reusadas, mas devem ser recriadas, uma vez que todas as ontologias fontes devem referir ao vocabulário compartilhado. Dessa forma, qualquer forma de reutilização de ontologias ou comunicação com outros sistemas

fica prejudicada caso as ontologias não sejam recriadas usando o vocabulário compartilhado (SOUZA, 2007).

### 3.4 MECANISMOS DE INTEROPERABILIDADE ENTRE ONTOLOGIAS

Uma vez que uma ontologia é passível de ser relacionada com outra podemos escolher diferentes mecanismos para realizar esta ação. No trabalho de Felicíssimo (2004) são citados quatro mecanismos que também serão abordados neste trabalho, são eles: combinação de ontologias, integração de ontologias, alinhamento de ontologias e mapeamento de ontologias.

O mecanismo a ser escolhido depende da necessidade do sistema, cada um com sua particularidade e com resultado específico.

#### 3.4.1 Combinação de Ontologias

Na combinação de ontologias tem-se como resultado a versão das ontologias originais combinadas em uma ontologia única, com todos seus termos juntos e sem a definição clara de suas origens (FELICÍSSIMO, 2004). É difícil identificar na ontologia resultante quais partes foram geradas de qual ontologia fonte. A Figura 10 a seguir ilustra este processo, a ontologia 1 representaria o domínio Carro e a ontologia 2, compatível com a primeira, seria o domínio Veículo, seus termos seriam processados e combinados gerando uma terceira ontologia com a união dos termos das duas ontologias fontes.

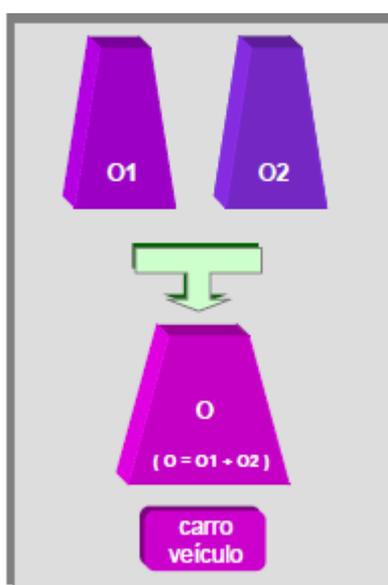


Figura 10. Compatibilidade das ontologias carro e veículo (FELICÍSSIMO, 2004)

Neste mecanismo as ontologias fonte são ontologias diferentes, e não versões de uma mesma ontologia. O objetivo é gerar uma ontologia mais geral completando os termos das duas ontologias, pode-se também usar esta abordagem com mais de duas ontologias aumentando ainda mais a generalidade da ontologia resultante.

### 3.4.2 Alinhamento de Ontologias

No alinhamento de ontologias tem-se como resultado as duas ontologias originais separadas, mas nestas são adicionadas às ligações entre seus termos equivalentes (FELICÍSSIMO, 2004).

Este processo consiste em gerar ligações entre duas ontologias com o objetivo de prover a reutilização da informação de cada ontologia entre elas.

A figura 11 mostra o processo de alinhamento, podemos observar que os conceitos de Carro são alinhados com os de Veículo criando uma ligação entre eles.

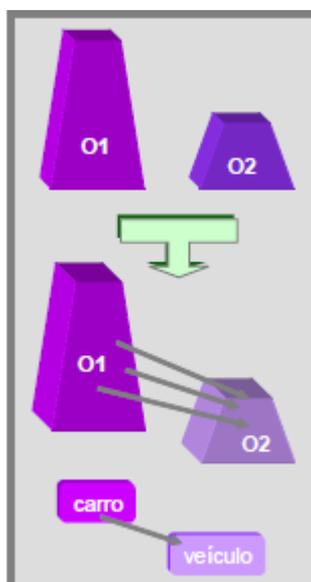


Figura 11. Alinhamento das ontologias carro e veículo (FELICÍSSIMO, 2004)

### 3.4.3 Integração de Ontologias

Na integração de ontologias tem-se como resultado uma ontologia única criada pela montagem, extensão, especialização ou adaptação de outras ontologias de assuntos diferentes (FELICÍSSIMO, 2004).

A Integração de ontologias é um processo parecido com a Combinação, duas ontologias fonte são combinadas gerando uma resultante. Porém na Integração é possível diferenciar na ontologia resultante de qual ontologia fonte veio cada parte.

Segundo Souza (2007) quando a ontologia integrada é reutilizada pela ontologia resultante, os conceitos integrados podem ser, entre outras coisas Resnik<sup>7</sup> (apud SOUZA, 2007): (1) utilizados como estão, (2) adaptados (ou modificados), (3) especializados (levando a uma ontologia mais específica no mesmo domínio) ou (4) acrescidos de novos conceitos (ou por conceitos mais gerais ou por conceitos do mesmo nível).

Na figura 12 temos a representação da integração, uma ou mais ontologias são integradas gerando uma nova ontologia, nesse caso temos os conceitos de Carro, Veículo, Automóvel e Meio de transporte terrestre.

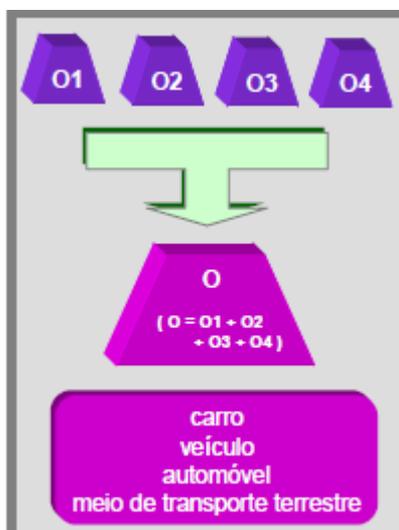


Figura 12. Integração das ontologias carro e veículo (FELICÍSSIMO, 2004)

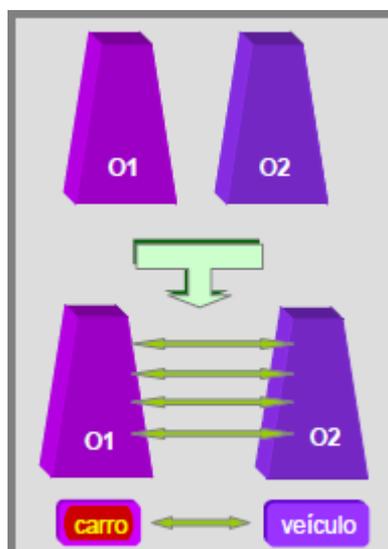
Neste processo os domínios das ontologias fonte são diferentes, mas se relacionam em uma parte permitindo a integração e estendendo o escopo das duas ontologias gerando uma mais geral.

#### 3.4.4 Mapeamento de Ontologias

No mapeamento de ontologias tem-se como resultado uma estrutura formal com expressões que ligam os termos de uma ontologia nos termos de outra ontologia (SOUZA, 2007).

<sup>7</sup> RESNIK, P., ELKISS, A., LAU, E., et al., 2005, "The Web in Theoretical Linguistics Research: Two Case Studies Using the Linguist's Search Engine". In: 31st Meeting of the Berkeley Linguistics Society, Berkeley, Califórnia, USA, Fevereiro de 2005.

Na figura 13 podemos observar as expressões formais criadas a partir do mapeamento das ontologias 1 e 2 relacionando os conceitos de carro e veículo.



**Figura 13. Mapeamento das ontologias carro e veículo (FELICÍSSIMO, 2004)**

Este processo pode ser automático ou semiautomático, tendo a intervenção humana no segundo caso. Alguns estudiosos acreditam que o mapeamento automático sem a intervenção humana não é possível. Esta intervenção pode ser alimentando o algoritmo com pares iniciais, verificando as ligações ou configurando os termos a ser usados.

É possível ainda usar mapeamentos existentes para criar novas relações. Se uma ontologia A tem ligações com outra ontologia B, e existe uma terceira C com mapeamentos com B, poderíamos inferir mapeamentos entre A e C. Seguindo esta abordagem geraríamos uma rede de ontologias com o mapeamento existente entre algumas delas.

### 3.5 COMPATIBILIDADES DE TERMOS

Para se relacionar ontologias é preciso buscar suas diferenças, completude e analisar se há possíveis inconsistências, no trabalho de Felicíssimo (2004) são listados os seguintes pontos a ser analisados:

1. Conceitos com um mesmo significado, mas rotulados com nomes diferentes;
2. Conceitos rotulados com o mesmo nome, mas com significados diferentes;
3. Diferenças na escrita dos termos das ontologias como, por exemplo: um termo no plural e outro no singular, um no feminino e outro no masculino, e em diferentes tempos verbais;

4. Propriedades com um mesmo significado, mas rotuladas com diferentes nomes;
5. Propriedades rotuladas com o mesmo nome, mas com significados diferentes;
6. Diferenças nas restrições e nas propriedades utilizadas;
7. Diferenças nas propriedades utilizadas em restrições – conceitos relacionados que são similares;
8. Diferenças nos conceitos relacionados utilizados em restrições – propriedades que são similares;
9. Diferenças no número de restrições (diferenças nos casos onde exista uma interseção de restrições);
10. Verificação se todos os conceitos a que as propriedades se relacionam são equivalentes em ambas as ontologias comparadas;
11. Verificação se todos os conceitos que utilizam uma propriedade são consistentes em ambas as ontologias comparadas.

Os itens citados ajudam a medir a inconsistência entre as ontologias, e a partir do resultado desta medição decidir qual ação tomar. Não se deve ignorar inconsistências, mas também não é necessário tratar todas, deve-se manter um limite de tolerância para tratá-las a partir de uma medida de similaridade.

### 3.6 SIMILARIDADE SEMÂNTICA

Será cada vez mais constante a necessidade de se calcular a similaridade entre ontologias. Com o aumento do número de ontologias existirão várias ontologias para descreverem um mesmo domínio. Seria difícil pensar em ontologias aceitas por todos sobre um determinado assunto, pois no processo de desenvolvimento é criada uma ontologia que atende a necessidade de um pequeno grupo. Esse problema não pode ser impeditivo para a interoperabilidade de ontologias, isso aumenta a importância do cálculo de similaridade semântica.

Em Souza (2007) são abordados dois tipos de divergências entre ontologias: **divergência na linguagem de descrição** e **divergência na conceitualização**.

Quando há divergência na linguagem a solução é a tradução da ontologia para a linguagem escolhida. Os maiores problemas encontrados na tradução são: a sintaxe, a representação lógica, a semântica das primitivas e a expressividade da linguagem.

A incompatibilidade sintática é fácil de resolver, porém proporciona outros problemas. Se a representação lógica estiver correta, mas sintaticamente estiver representada de forma diferente haverá problemas para tradução. Em algumas linguagens é possível representar a disjunção de forma explícita (por exemplo, disjoint

A B), em outras é necessário usar negações nas subclasses (por exemplo, A subclass-of (NOT B), B subclass-of (NOT A)) (SOUZA, 2007). Neste caso não temos problemas com conceitos e expressividade, mas sim com a representação lógica. Há regras de tradução lógica para este tipo de caso que resolvem este problema.

O problema na semântica das primitivas acontece quando um construtor sintaticamente igual tem significado semântico diferente. Se fizermos uma tradução das ontologias para uma representação comum intermediária resolvemos os problemas de sintaxe, semântica das primitivas e expressividade.

As diferenças no nível de conceitualização, ou nível de modelo, podem ocorrer entre ontologias de mesma linguagem ou linguagens diferentes. Está incluída neste tipo de divergência a diferença na codificação ou no significado de conceitos.

Essas divergências podem ser divididas entre divergências de conceitualização, divergências na explicação (ou estruturação) e divergências terminológicas (SOUZA, 2007). A diferença no modo como um domínio é interpretado causa a **divergência na conceitualização**, pode haver diferentes visões sobre uma conceitualização e grupos de especialistas podem expressá-la diferentemente. Se a granularidade do domínio representada for diferente teremos uma divergência de conceitualização. A figura 14 mostra o domínio de computadores representado em versões diferentes, um exemplo de divergência ao nível de conceitualização.

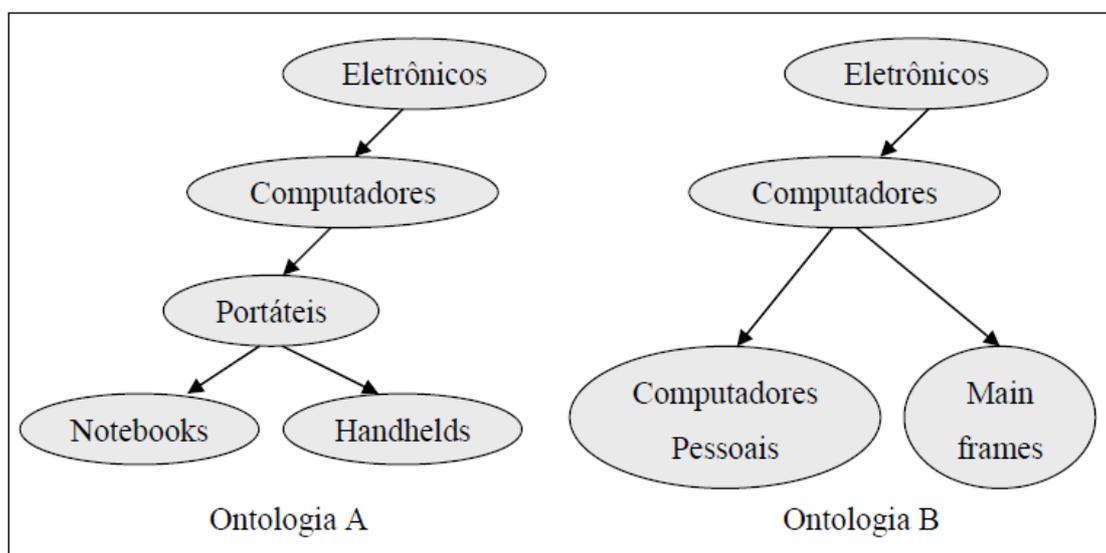
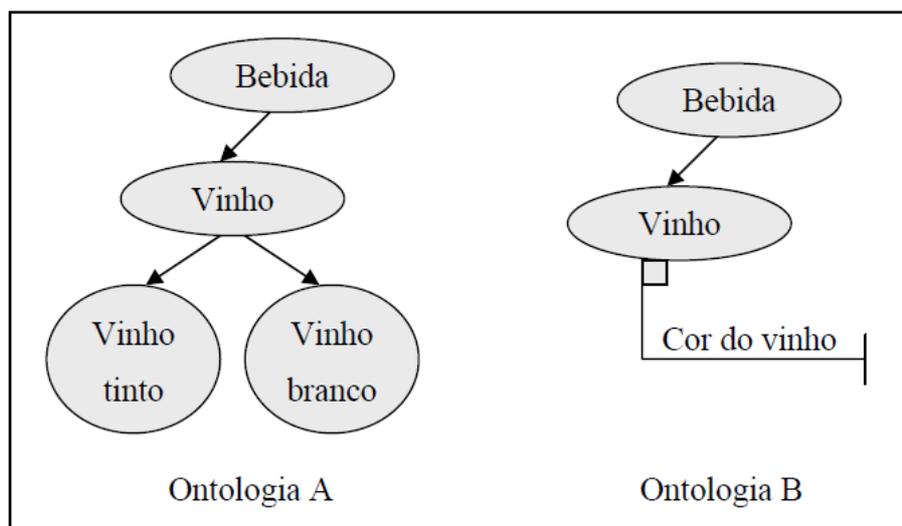


Figura 14. Exemplo de divergência na conceitualização (SOUZA, 2007)

Uma **divergência de explicação** é a diferença no modo em que a conceitualização é especificada. Ela pode ocorrer em divergências nas definições, termos ou combinação dos dois, ocorrendo por diferenças no estilo de modelagem

usado (SOUZA, 2007). Isso é comum em ontologias que representam assuntos genéricos, o tempo, por exemplo, pode ser representado através de intervalos ou pontos gerando diferenças entre ontologias. Na modelagem ainda ocorre divergência no modo como é descrito o domínio, em uma ontologia pode haver uma classe que em outra é uma propriedade, as duas com o mesmo objetivo de conceitualização. Em Souza (2007) é proposto um exemplo, como mostra a figura 15, em que a cor do vinho pode ser uma propriedade da classe Vinho ou uma classe Cor à parte.



**Figura 15. Exemplo de divergência de explicação (SOUZA, 2007)**

Ainda temos, por último, as divergências terminológicas causadas por homônimos ou sinônimos. A diferença na codificação também se enquadra nesse tipo de divergência. Um exemplo de codificação diferente seria a representação de data: "dd-mm-aaaa" e "aaaa-mm-dd".

A solução para resolver inconsistências de conceitualização não pode ser automatizada, dependendo de especialistas do domínio para isso.

Os algoritmos que resolvem problemas de similaridade basicamente levam em conta a comparação de strings, que calcula a similaridade léxica, e a comparação de estrutura, em que a ontologia é trabalhada como um grafo orientado, e sua estrutura é percorrida e analisada fazendo uma verificação semântica.

No trabalho de Assis e Souza (2011) é proposto um cálculo de similaridade utilizando algoritmo genético, em que a medida de similaridade é calculada levando em conta valores já atribuídos à dois termos de duas ontologias.

### 3.7 FERRAMENTAS

A interoperabilidade de ontologias é bastante usada no contexto da Web Semântica, tanto para ganho em qualidade como para reuso e aproveitamento de código. Para isto foram desenvolvidas diversas ferramentas que auxiliam este processo, algumas vezes o tornando completamente automático.

Em Ehrig<sup>8</sup> (apud PALAZZI, 2010) são descritas algumas ferramentas para alinhamento, são elas:

1. NOM – Naive Ontology – As ontologias devem estar na linguagem RDFS para permitir o processamento, então todas as entidades da primeira ontologia são comparadas com a da segunda. A similaridade entre os termos é computada de acordo com o quadro 2. A ferramenta ainda destaca os termos com maior grau de similaridade, e o processo pode se repetir de forma iterativa, refinando a comparação e o alinhamento.

Comparing	No.	Feature	Similarity Measure
Concepts	1	(label, $X_1$ )	string similarity( $X_1, X_2$ )
	2	( $URI_1$ )	string equality( $URI_1, URI_2$ )
	3	( $X_1, sameAs, X_2$ ) relation	explicit equality( $X_1, X_2$ )
	4	(direct properties, $Y_1$ )	SimSet( $Y_1, Y_2$ )
	5	all (inherited properties, $Y_1$ )	SimSet( $Y_1, Y_2$ )
	6	all (super-concepts, $Y_1$ )	SimSet( $Y_1, Y_2$ )
	7	all (sub-concepts, $Y_1$ )	SimSet( $Y_1, Y_2$ )
	8	(concept siblings, $Y_1$ )	SimSet( $Y_1, Y_2$ )
	9	(direct instances, $Y_1$ )	SimSet( $Y_1, Y_2$ )
	10	(instances, $Y_1$ )	SimSet( $Y_1, Y_2$ )
Relations	1	(label, $X_1$ )	string similarity( $X_1, X_2$ )
	2	( $URI_1$ )	string equality( $URI_1, URI_2$ )
	3	( $X_1, sameAs, X_2$ ) relation	explicit equality( $X_1, X_2$ )
	4	(domain, $X_{d1}$ ) and (range, $X_{r1}$ )	object equality( $X_{d1}, X_{d2}$ ) and ( $X_{r1}, X_{r2}$ )
	5	all (super-properties, $Y_1$ )	SimSet( $Y_1, Y_2$ )
	6	all (sub-properties, $Y_1$ )	SimSet( $Y_1, Y_2$ )
	7	(property siblings, $Y_1$ )	SimSet( $Y_1, Y_2$ )
	8	(property instances, $Y_1$ )	SimSet( $Y_1, Y_2$ )
Instances	1	(label, $X_1$ )	string similarity( $X_1, X_2$ )
	2	( $URI_1$ )	string equality( $URI_1, URI_2$ )
	3	( $X_1, sameAs, X_2$ ) relation	explicit equality( $X_1, X_2$ )
	4	all (parent-concepts, $Y_1$ )	SimSet( $Y_1, Y_2$ )
	5	(property instances, $Y_1$ )	SimSet( $Y_1, Y_2$ )
Property-instances	1	(domain, $X_{d1}$ ) and (range, $X_{r1}$ )	object equality( $X_{d1}, X_{d2}$ ) and ( $X_{r1}, X_{r2}$ )
	2	(parent property, $Y_1$ )	SimSet( $Y_1, Y_2$ )

**Quadro 2- Características e medidas de similaridade entre ontologias**

2. QOM – Quick Ontology Mapping – É uma otimização do NOM, a principal diferença está na estratégia de comparação, nesta ferramenta não se compara toda a ontologia de uma só vez, mas sim partes dela.

<sup>8</sup> EHRIG, M., STAAB, S. (2004). Efficiency of Ontology Mapping Approaches. In: International Workshop on Semantic Intelligent Middleware for the Web and the Grid at ECAI 04, Valencia, Espanha. Disponível em <http://www.aifb.uni-karlsruhe.de/WBS/meh/publications/ehrig04efficiencySIMIS.pdf>

3. FOAM – Framework for Ontology Alignment and Mapping – Nesta ferramenta o usuário tem uma participação maior, escolhendo duas entidades a ser comparadas e definindo a medida aceitável de similaridade para se fazer o alinhamento.

Por último vamos falar mais um pouco sobre o PROMPT, a ferramenta que será utilizada no próximo capítulo.

### 3.7.1 Prompt

Foi apresentado em Noy e Musen (2001) este conjunto de ferramentas, o PROMPT surge como um plugin do Protégé, e é composto pelas seguintes ferramentas: 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 Interoperabilidade de Ontologias e PROMPTDiff, uma ferramenta para identificação de diferenças entre duas versões da mesma ontologia. A figura 16 mostra como a ferramenta é estruturada:

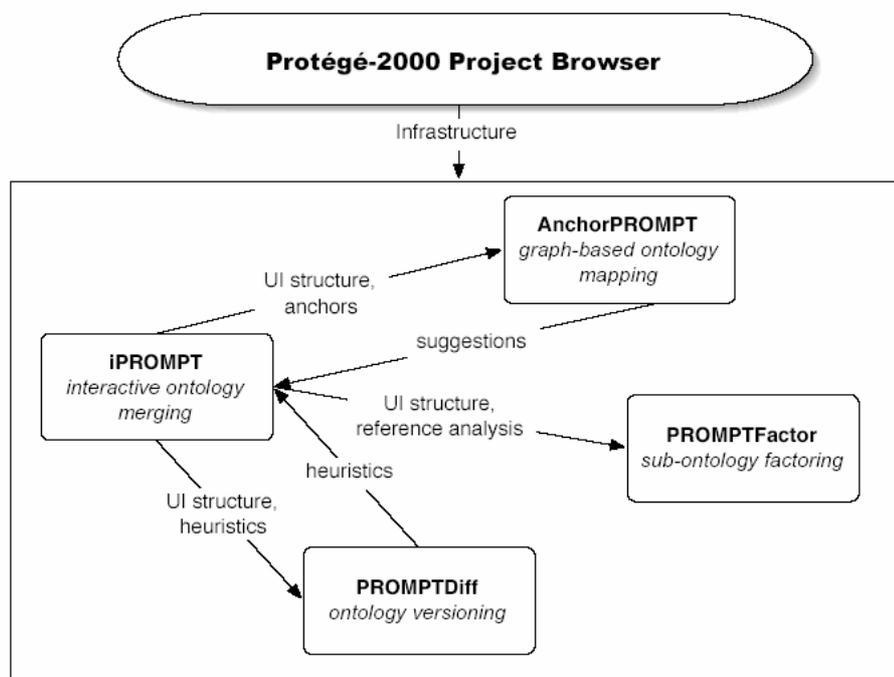
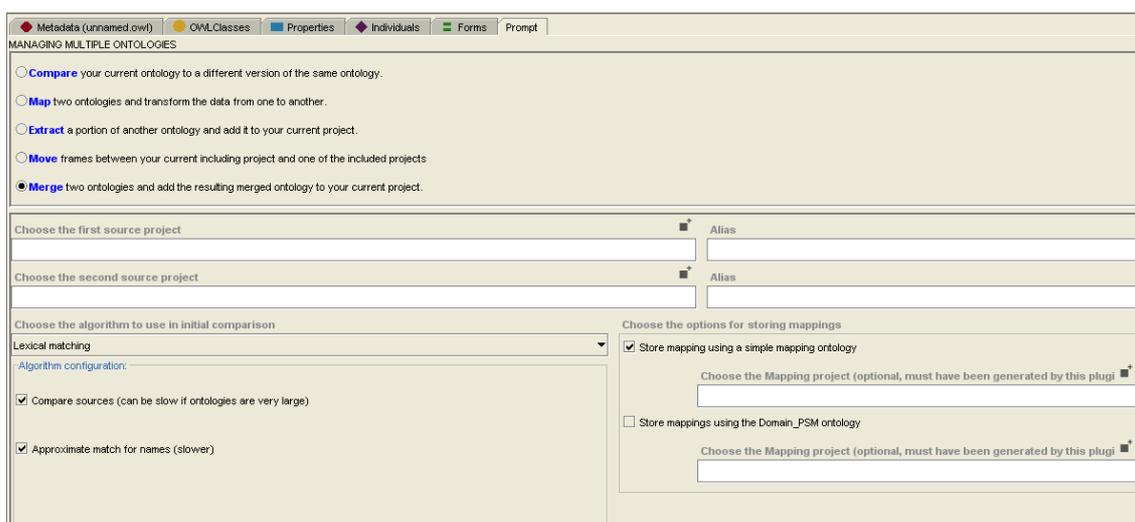


Figura 16. Ferramentas do PROMPT, Noy e Musen (2003).

O iPROMPT faz um processamento semiautomático guiando o usuário entre as possíveis opções de combinações, apresentando inconsistências e sugerindo estratégias para resolvê-los.

A ferramenta utilizada para o mapeamento entre o conjunto de ferramentas do PROMPT é o Anchor\_PROMPT. A principal característica desta ferramenta é o uso de caminho em grafos para comparação de termos, a ontologia é transformada e processada como um grafo orientado.

A figura 17 mostra a versão mais recente do PROMPT sobre a última versão do Protégé que suporta o software:



**Figura 17. Tela do plugin PROMPT**

A ferramenta PROMPTFactor é uma ferramenta para separação de subontologias, semanticamente independentes, de ontologias extensas (FELICÍSSIMO, 2004). Esta ferramenta é importante para auxiliar a criação de uma ontologia a partir de outra sem que todos os termos sejam aproveitados. Isto é útil especialmente para o reuso de ontologias extensas.

O PROMPTDiff é usado para mostrar diferenças entre duas versões de uma mesma ontologia, é muito útil para o controle de versões. Nesta ferramenta, a comparação não é feita apenas por comparação de textos, como tradicionalmente é realizado em comparação de versões de programas, mas também, por comparação estrutural (FELICÍSSIMO, 2004).

## 4 ALINHAMENTO ODEq E GAMS

### 4.1 INTRODUÇÃO

Este trabalho faz parte de uma pesquisa de Iniciação Científica cujo projeto incluiu a construção da ontologia ODEq – *Ordinary Differential Equations Ontology*, a qual foi modelada em Silva et al. (2009). Após sua criação, a ontologia foi alinhada com a ontologia GAMS – *Guide to Available Mathematical Software* (Caprotti O.; Dewar M.; Turi D.,2004), uma subontologia do Projeto MONET.

Neste capítulo apresentaremos a ontologia ODEq e seu processo de desenvolvimento, assim como o Projeto MONET - *Mathematics on the Net* e a ontologia GAMS. Por fim será apresentado o processo de alinhamento e os resultados obtidos.

### 4.2 ONTOLOGIA ODEq

O domínio descrito nesta ontologia foi o de Métodos de Resolução de Equações Diferenciais. Seu objetivo é encontrar um método de resolução a partir de uma classificação hierárquica de equações diferenciais ordinárias.

A ontologia foi modelada no trabalho de Silva et al (2009), e para entendermos melhor sua estrutura vamos descrever seu domínio.

Uma equação diferencial é uma equação cuja incógnita é uma função que aparece na equação sob a forma das respectivas derivadas (BOYCE & DIPRIMA, 2002). As equações diferenciais são usadas para construir modelos matemáticos de fenômenos físicos e têm inúmeras aplicações práticas em medicina, engenharia, química, biologia e outras áreas do conhecimento (SILVA, 2009).

A ontologia se divide da seguinte forma: em um primeiro nível, descreve os métodos de resolução de uma equação diferencial de acordo com o tipo de equação. O nível 2 descreve alguns métodos de resolução de equações diferenciais existentes. O nível 3 os tipos de equações diferenciais que são resolvidas a partir do método especificado no nível 2. O nível 4 determina a ordem da equação e no último nível se a equação especificada na classe anterior é uma equação linear ou não linear (SILVA, et al., 2010).

A ontologia se inspirou no projeto MONET e foi validada com professores da área de Matemática e de Engenharia de Software da Universidade Federal de Juiz de Fora.

O quadro 3 abaixo representa a modelagem do domínio, e foi a partir dele que a ontologia foi desenvolvida.

### Methods of solving ordinary differential equation

Level 2	Level 3	Level 4	Level 5
- Integrating factors - Separation of variables	- Linear Equations with constants coefficients - Separable Equations - Exact Equation	- First Order Linear Differential Equation	- Linear Ordinary Equation
- Euler's Method	- Homogeneous Equations	- Second Order Linear Differential Equation	
- Method of undetermined Coefficients - Variation of Parametres	- Nonhomogeneous Equations	- First Order Linear Differential Equation	
- Laplace Transform	- Linear Equations with constants coefficients - Separable Equations - Exact Equation	- Second Order Linear Differential Equation	
	- Homogeneous Equations with constants coefficients - Nonhomogeneous Equations with constants coefficients	- Differential equations in other orders	
- Princípio da Linearização - Nullclines	- Autonomous Equations	- First order	- Nonlinear Ordinary Equation
	- Nonautonomous Equations		

**Quadro 3- Representação da hierarquia de classes**

A primeira versão da ontologia foi fiel a esta representação, mais adiante quando foi estudada o alinhamento entre ODEq e GAMS foi verificada a necessidade de fazer alterações na hierarquia da ontologia. Como o objetivo é chegar a um método de resolução a partir de uma equação diferencial a ordem da hierarquia foi invertida, partindo das equações lineares e não lineares como classes mais genéricas para então chegar a um método, no nó-folha.

Na figura 18 está a versão final da ontologia ODEq, a imagem foi gerada a partir do plugin OWLVizTab do Protégé.

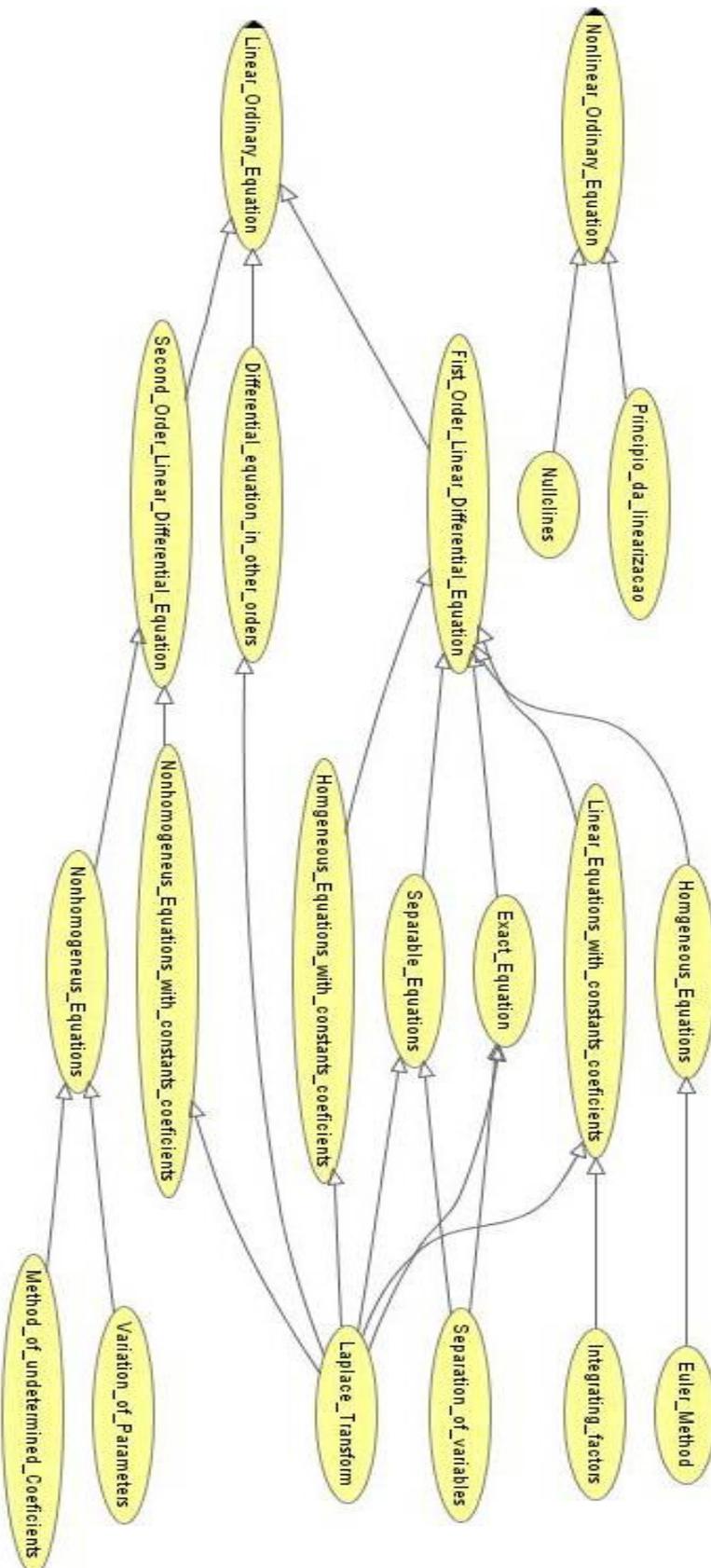


Figura 18 – Hierarquia de classes gerada pelo OWLVizTab

A estrutura de classes apresentada na ontologia possibilitou fazer o alinhamento com a ontologia GAMS, há classes com o mesmo significado semântico entre as duas ontologias. Antes de mostrar o alinhamento vamos falar sobre o projeto MONET e a sua subontologia GAMS.

#### 4.2.1 Metodologia de Desenvolvimento

A metodologia adotada para o desenvolvimento da ontologia foi a QDAontology citada na seção 2.6. A cada etapa da metodologia são gerados artefatos, em seguida vamos mostrar os documentos gerados.

##### ETAPA 1 – ESPECIFICAÇÃO

Segundo Palazzi (2010) a etapa de Especificação identifica a motivação pela qual a ontologia está sendo construída e quais são os objetivos de seu uso, e especifica os usuários finais. Nesta etapa foi gerado o documento de requisitos apresentado no Quadro 4, ele define informações importantes para o desenvolvimento da ontologia. Os atributos de qualidade alvo desta etapa são: clareza dos termos, unicidade dos termos, compromisso ontológico, relação com a realidade.

<b>Domínio</b>	<i>Ordinary differential equations</i>
<b>Data</b>	2010
<b>Desenvolvida por</b>	Francilene Barbosa dos Santos Silva
<b>Objetivo</b>	<i>Provides representation of the methods of solving ordinary differential equation</i>
<b>Escopo</b>	<i>Itens: Equações diferenciais e métodos de resolução especificados na árvore de termos Conceitos: Descritos na tabela de conceitualização</i>
<b>Fontes de conhecimento</b>	<i>Artigo [Silva, 2010]</i>

Quadro 4- Documento de Requisitos

##### ETAPA 2 – CONCEITUALIZAÇÃO

A atividade de Conceitualização organiza e estrutura o conhecimento do domínio como modelos significativos no nível de conhecimento, usando representações externas, que são independentes da linguagem de implementação e do ambiente de desenvolvimento (PALAZZI, 2010). O documento gerado nesta etapa descreve os termos utilizados na ontologia, seu objetivo é propor uma representação intermediária para a ontologia, facilitando o desenvolvimento da mesma. O quadro 5 mostra o artefato gerado nesta etapa, chamado de Documento de Conceitualização.

Name	Description
Integrating factors	Is a function used to facilitate integration and solve the equation
Separation of variables	To write an equation so that each of the two variables appear on different sides of the equation.
Euler's Method	Is a first-order numerical procedure for solving ordinary differential equations (ODEs) with a given initial value

Quadro 5- Documento de Conceitualização

### ETAPA 3 – FORMALIZAÇÃO

Nesta etapa é desenvolvido um documento que transforma o modelo conceitual em um modelo formal. No nosso caso a hierarquia dos termos foi montada, o quadro 6 apresenta a Árvore de Classificação. As características de qualidade abordadas nesta fase são: organização, modularidade, estrutura invariante, independência da codificação, adequação semântica, lógica correta e restrições.

Nonlinear differential equation
Pincípio da linearização
Nullclines
Linear differential equation
First order linear differential equations
Homogeneous equations
Euler's method
Homogeneous equations with constants coefficients
Laplace transform
Exact equations
Separation of variable
Laplace transform
Separable equations
Laplace transform
Separation of variable
Linear equations with constants coefficients
Laplace transform
Integrating factors
Second order linear differential equations
Nonhomogeneous equations
Variation of parameters
Method of undetermined coefficients
Nonhomogeneous equations with constants coefficients
Laplace transform

Differential equations in other orders  
Laplace transform

Quadro 6- Árvore de classificação

#### ETAPA 4 - IMPLEMENTAÇÃO

Constrói modelos computáveis utilizando ferramentas para o desenvolvimento de ontologias. Define a organização formal dos termos, conceitos e relacionamentos de forma a permitir inferência automatizada, através da definição de regras e restrições. O resultado é a ontologia codificada em uma linguagem formal (SILVA, et al., 2010). A linguagem escolhida foi OWL, e foi utilizado o editor Protégé (PROTEGE, 2010), o artefato gerado foi a ontologia codificada, ilustrada no quadro 7.

```
<owl:Class rdf:about="#Nonhomogeneous_Equations">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Are ordinary differential
equations in the form dw
= g(t).</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#Second_Order_Linear_Differential_Equation"/>
</owl:Class>
<owl:Class rdf:ID="Principio_da_linearizacao">
  <rdfs:subClassOf rdf:resource="#Nonlinear_Ordinary_Equation"/>
</owl:Class>
<owl:Class rdf:ID="Differential_equation_in_other_orders">
  <rdfs:subClassOf rdf:resource="#Linear_Ordinary_Equation"/>
</owl:Class>
<owl:Class rdf:ID="Separable_Equations">
  <rdfs:subClassOf rdf:resource="#First_Order_Linear_Differential_Equation"/>
```

Quadro 7 – Código OWL

#### ETAPA 5 – INTEGRAÇÃO

Esta etapa será abordada na seção 4.4, onde será discutido o alinhamento entre GAMS e ODEq.

#### 4.3 PROJETO MONET

As ontologias do projeto MONET foram projetadas para modelar tanto buscas como descrição de serviços, elas podem ser divididas entre as que foram

desenvolvidas pelo próprio projeto e outras de terceiros (GAMS, por exemplo). O projeto foi fundado em 2002 pela Sociedade de Informações Tecnológicas dos Estados Unidos.

O objetivo do projeto Monet é utilizar a web semântica de um domínio matemático, usando algoritmos sofisticados para selecionar as características de um problema para as capacidades dos serviços disponíveis e então escolher um serviço através de um mecanismo padrão (SILVA, 2009).

A ontologia MONET pode ser dividida entre várias subontologias, o projeto foi desenvolvido quando a linguagem OWL ainda não estava completa, e o software utilizado não tinha recursos para dividir e integrar ontologias. Por isso foi desenvolvida uma ferramenta a parte para fazer esta ação.

As subontologias são divididas nos seguintes domínios:

- A ontologia genérica MONET
- Algoritmos
- Bibliografia
- Diretivas
- Codificação
- The Guide to Available Mathematical Software (GAMS)
- Hardware
- OpenMath Símbolos
- Problemas Matemáticos
- Software
- Taxonomia de problemas de simbologia
- Teorias

Em MATOS (2008) são apresentados alguns trabalhos gerados pelo projeto MONET, são eles: a implementação de um broker, a definição de uma linguagem para descrição de serviços (MSDL – Mathematical Service Description Language), várias ontologias associadas ao domínio e o componente InstanceStore.

A ontologia escolhida a ser mapeada foi GAMS, equações diferenciais é uma classe de GAMS, e podemos afirmar que o domínio ODEq está contido em GAMS. Vamos falar mais sobre ela na próxima seção.

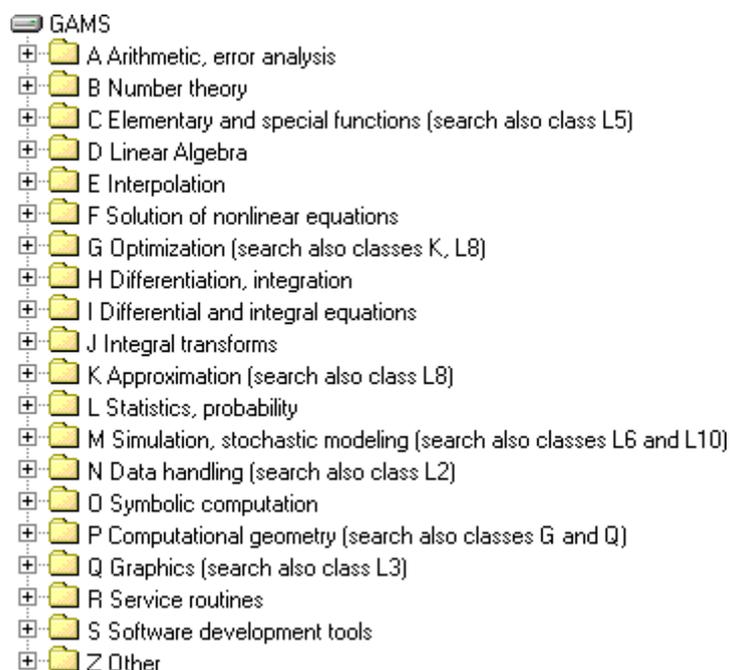
#### 4.3.1 GAMS

O Guia para os Programas Matemáticos Disponíveis (GAMS) é um serviço oferecido pelo Instituto Nacional de Ciência e Tecnologia que provê uma lista de

programas matemáticos disponíveis classificados de acordo com o tipo de problema que são capazes de resolver (SILVA et al., 2009).

Cada classe GAMS corresponde a uma classe OWL e a especialização de uma classe de problemas é representada com um relacionamento de subclasse na ontologia (MATOS, 2008).

Essa ontologia é estruturada como uma taxonomia, a figura 19 mostra o nível mais genérico:



**Figura 19. Hierarquia de classes da ontologia GAMS**

A classe de nosso interesse é a "I Differential and Integral equations", são subclasses dessa classe os seguintes termos: GAMS11b1 (Descrição: Linear) e GAMS11b2 (Descrição: Nonlinear).

#### 4.4 ALINHAMENTO GAMS-ODEq

Após análise e estudo das ontologias GAMS e ODEq foi necessário fazer uma alteração da hierarquia da ontologia ODEq. O objetivo desta alteração foi deixar as classes que fazem referência aos métodos de desenvolvimento no nível mais baixo, se imaginar a ontologia como um grafo orientado essas classes seriam o nó-folha. Essa alteração permite alcançar o objetivo geral que é encontrar o serviço ou algoritmo de resolução da equação a partir da hierarquia de classes contida na ontologia.

A ferramenta utilizada foi o PROMPT, ela é um plugin do editor Protégé. A partir do alinhamento as instâncias das duas ontologias ficaram ligadas pelas classes escolhidas para o relacionamento.

Como o nome das classes da ontologia GAMS não faz referência ao seu significado semântico o alinhamento teve de ser inteiramente manual, a escolha das classes não pôde ser feita de forma automática, apesar de a ferramenta prover esse tipo de mecanismo.

As equações diferenciais são classes de GAMS e suas subclasses são as classes mais genéricas da ontologia ODEq. As classes com correspondência semântica entre as duas ontologias estão representadas na tabela abaixo:

<i>ODEq - Classes</i>	<i>GAMS - Classes</i>
<i>Linear</i>	<i>GAMS11b1</i>
<i>Nonlinear</i>	<i>GAMS11b2</i>

**Quadro 8 – Alinhamento das classes**

Como resultado do alinhamento temos as duas ontologias originais com a ligação entre elas que permite transferir instâncias de dados, esquemas de integração combinação e outras tarefas similares. Assim um serviço web que resolva um tipo de equação diferencial poderia ser uma instância da ontologia ODEq e seu acesso se tornaria possível a partir da ontologia mais geral GAMS.

De acordo com Kalfoglou e Schorlemmer (2003) um dos objetivos na especificação de pontes semânticas entre ontologias é manter e explorar as construções existentes e minimizar construções extras, o que poderia maximizar a aceitação e compreensão pelo conjunto de ferramentas de serviços web.

O Alinhamento de um conceito em uma ontologia permite uma consulta sobre outras ontologias, que adquire as informações relevantes, navegando e agregando conceitos, mais adiante vamos falar sobre os resultados do alinhamento. A imagem abaixo mostra a tela do PROMPT durante o alinhamento com as classes escolhidas:

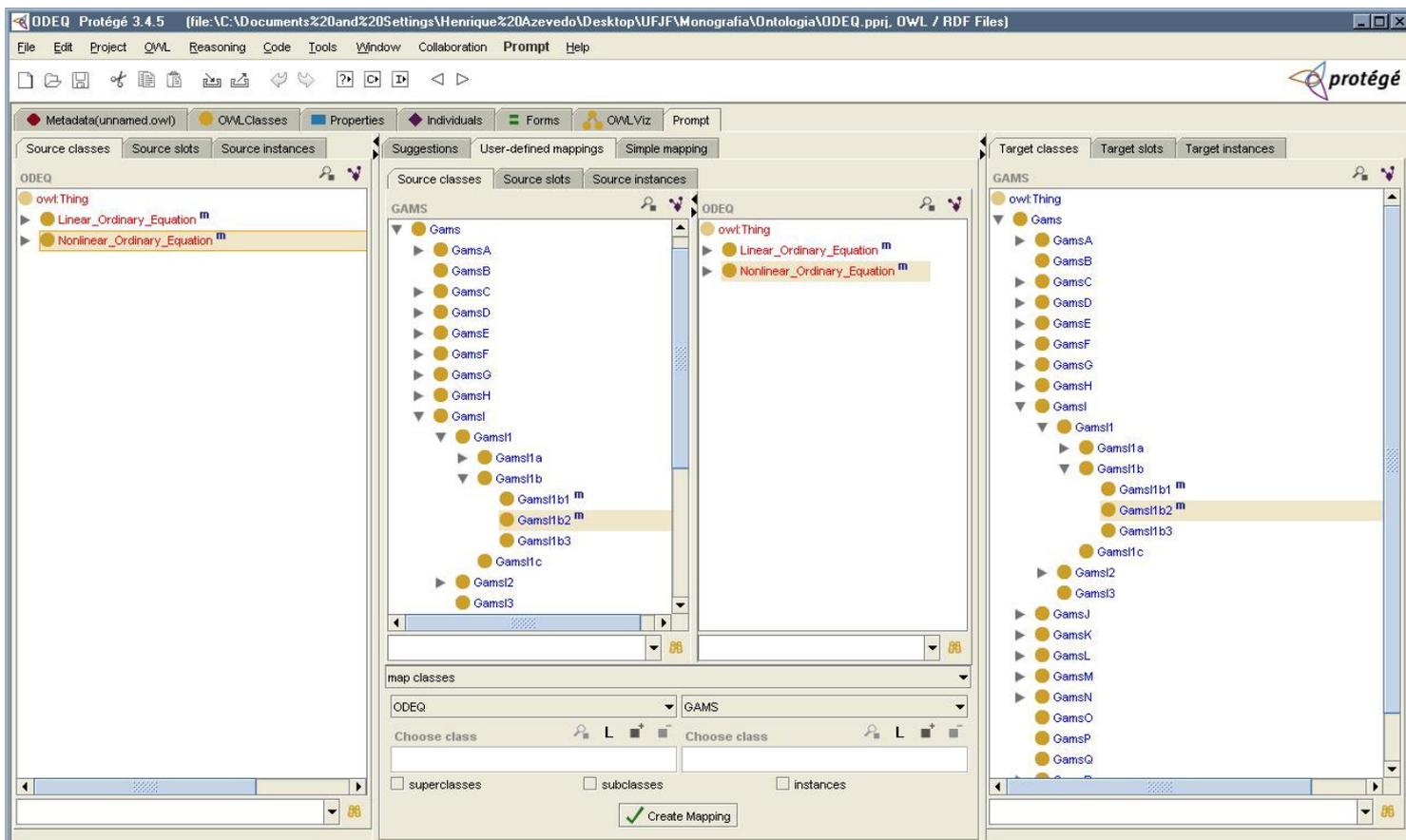


Figura 20. Tela da ferramenta PROMPT durante o alinhamento

Após o alinhamento foram gerados 3 arquivos, em um novo projeto do editor Protégé, pela ferramenta PROMPT com informações sobre o processo.

#### 4.4.1 Processo de alinhamento

Desde o início do trabalho a interoperabilidade entre a ontologia ODEq era um dos objetivos. Isto motivou o uso da linguagem OWL, que foi a utilizada pelo projeto MONET na subontologia GAMS.

O fato das duas ontologias estarem representadas na mesma linguagem facilitou a interoperabilidade evitando divergência no tipo de linguagem. Outro fator positivo foi a escolha da ferramenta Protégé que foi utilizada na criação da ODEq e foi usada também para a edição da ontologia GAMS. O PROMPT forneceu o serviço de alinhamento semiautomático, bastando indicar as classes a serem relacionadas.

O alinhamento estendeu o escopo das duas ontologias, inserindo em GAMS mais informação sobre o domínio de resolução de equações diferenciais, e possibilitando a associação a um serviço web que processe essas equações. A partir de instâncias de uma das duas ontologias é possível acessar informações da outra em consultas e então, no nosso caso, encontrar o método de resolução da equação, ou até futuramente encontrar o serviço que a resolva.

No próximo capítulo vamos discutir os resultados e os trabalhos futuros a partir deste alinhamento.

## 5 CONCLUSÃO

Este trabalho pode ser considerado uma extensão de Silva (2009), onde houve uma proposta de ontologia e foi sugerida sua possível extensão, relacionando a ontologia modelada com ontologias do projeto MONET.

Após introduzir conceitos importantes sobre ontologias, como metodologias de desenvolvimento, linguagens, e ferramentas, foram apresentados mecanismos de interoperabilidade assim como ferramentas e problemas que devem ser resolvidos para possibilitar a extensão de ontologias através de integração, mapeamento, alinhamento ou combinação entre ontologias.

Por fim os resultados foram apresentados no capítulo 4, onde a documentação da criação da ontologia ODEq foi apresentada, com base na metodologia QDAOntology. A escolha desta metodologia teve o objetivo de inferir mais qualidade no processo de desenvolvimento, segundo Palazzi (2010) o uso de um modelo de processo no desenvolvimento de uma ontologia torna esta tarefa mais impessoal, menos complexa e mais sistemática, gerando assim menos erros no desenvolvimento. A metodologia ainda visa o desenvolvimento de ontologias com a participação de equipes multidisciplinares, no nosso caso tivemos a participação de professores da área de Matemática, que participaram na validação da ontologia, e Engenharia de Software da Universidade Federal de Juiz de Fora.

A ontologia teve alterações em sua versão original modelada sendo validada com profissionais da área do domínio. Após o estudo da possível interoperabilidade foi identificada a necessidade de alterações na ontologia ODEq, para atender os objetivos da ontologia GAMS, que provê serviços e algoritmos para resolução de uma série de problemas matemáticos, deixando as classes que representam os métodos de resolução de equações diferenciais ordinárias no último nível da ontologia. O mecanismo escolhido foi o alinhamento, que permite relacionar as classes equivalentes criando uma ponte semântica entre as ontologias.

### 5.1 TRABALHOS FUTUROS

Futuramente poderia haver alterações na ontologia ODEq, segundo (SILVA, 2009) a representação em níveis explicitada da ontologia não é única e pode ser melhorada. A ontologia poderia ser expandida aumentando o domínio para prover mais métodos de resolução em um maior grupo de tipos de equações.

Seria possível uma aplicação utilizar a ODEq para relacionar um tipo de equação a um método. A partir de consultas sobre a ontologia, o retorno seria uma

URI que a partir de parâmetros passados na aplicação teria o resultado da aplicação do método de resolução sobre a equação.

## REFERÊNCIAS

ABU-HANNA, A., JANSWEIJER, W.; "Modelling Domain Knowledge Using Explicit Conceptualization"; IEEE Expert; pp 53-63; 1994.

ASSIS, B. A. C; SOUZA J. F. Utilização de algoritmos genéticos para solução do problema de alinhamento de ontologia. II Workshop de Trabalhos de Graduação e Pós-Graduação, Universidade Federal de Juiz de Fora. Juiz de Fora, MG. 2011.

BOYCE, WILLIAM E; DIPRIMA, RICHARD C. Elementary Differential Equations and Boudary value Problems. 7th Ed. 2002. Disponível em:  
<[http://monet.nag.co.uk/monet/publicdocs/monet\\_ontologies.html](http://monet.nag.co.uk/monet/publicdocs/monet_ontologies.html)>  
Acessado em Junho de 2011

FELICÍSSIMO, C. H. Interoperabilidade semântica na Web: uma estratégia para o alinhamento taxonômico de ontologias. Dissertação (mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, RJ, Departamento de Informática, 2004.

FREITAS, F., STUCKENSCHIMDT, H., NOY, N., 2005, "Ontology Issues and Applications", Journal of the Brazilian Computer Society, v. 11, n. 2 (Novembro de 2005), pp. 4-13.

FREITAS, J. B. SiSe: Medida de similaridade semântica entre ontologias em português. Dissertação (mestrado) – Fac. De Informática PUCRS, Rio Grande do Sul. 2006

GOMEZ-PEREZ, A.; FERNÁNDEZ-LÓPEZ, M.; CORCHO, O. Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. Series: Advanced Information and Knowledge Processing, 2004, XII, 403 p. 159 illus., Hardcover ISBN: 1-85233-551-3.

GRUBER, T. R. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing". In: International Journal Human-Computer Studies Vol. 43, Issues 5-6, Novembro 1995, pp.907–928.

GUARINO, N. Formal ontology and information systems. In Proceedings of the 1st International Conference on Formal Ontologies in Information Systems. IOS Press. Trento, Itália, 1998.

KALFOGLOU, Y., SCHORLEMMER, M., 2003, "Ontology Mapping: the state of art", The Knowledge Engineering Review, v. 18, n. 1 (Janeiro de 2003), pp. 1-28.

MATOS, E. E. S. CelOWS : um framework baseado em ontologias com serviços web para modelagem conceitual em Biologia Sistêmica. 2008. Dissertação (Mestrado em Modelagem Computacional) - Universidade Federal de Juiz de Fora, Juiz de Fora, MG, 2008.

MONET. MONET Consortium. MONET Home Page. Disponível em:  
<<http://monet.nag.co.uk>>  
Acessado em: Agosto de 2010

NOY, N., MCGUINNESS, D.. Ontology Development 101 – A guide to creating your first ontology. KSL Technical Report, Stanford University, Stanford, CA, USA, 2000.

NOY, N. F., MUSEN, M. A. Anchor-PROMPT: Using Non-Local Context for Semantic Matching. In: WORKSHOP ON ONTOLOGIES AND INFORMATION SHARING AT THE INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 17. 2001a, Seattle. Anais do Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001), Seattle, 2001<sup>a</sup>. Disponível em:

<[http://smiweb.stanford.edu/pubs/SMI\\_Abstracts/SMI-2001-0889.html](http://smiweb.stanford.edu/pubs/SMI_Abstracts/SMI-2001-0889.html)>.

Acessado em Novembro de 2010.

ONTOCLEAN. 2010. OntoClean Central. Disponível em:

<<http://www.ontoclean.org>>

Acessado em Maio de 2011.

PRADO, S. G. D. “Um experimento no uso de ontologias para reforço da aprendizagem em Educação a Distância”. Tese de Doutorado. São Paulo – SP, 2004

PROTEGE (2007) Protege OWL – Ontology Editor for Semantic Web.

<http://protege.stanford.edu/plugins/owl/>

PALAZZI, D. C. QDAontology – Abordagem para o desenvolvimento de ontologias em e-Science: um estudo de caso em biologia.. Dissertação (Mestrado em Modelagem Computacional) - Universidade Federal de Juiz de Fora. Juiz de Fora, MG, Brasil, 2010.

SILVA, F. B. S.; CARDOSO, H. A.; CAMPOS, F.; BORGES, C. C. H. Resolução de Equações Diferenciais Ordinárias: Proposta de Ontologia. I Workshop de Trabalhos de Graduação e Pós-Graduação do DCC, Universidade Federal de Juiz de Fora. Juiz de Fora, MG. 2010.

SOUZA JUNIOR, H. C. Ontologias Emergentes: Uma abordagem para construção de ontologias a partir de mapeamentos ponto-a-ponto. 2008. Dissertação: (mestrado) - Instituto Militar de Engenharia, Rio de Janeiro, 2008

SOUZA, J. F. Negociação de significado para viabilizar interoperabilidade semântica. Dissertação (Mestrado) – COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, 2007.

## APÊNDICE

### Apêndice 1 - Documentação da ontologia ODEq

#### A1.1 Glossário de termos

Name	Description
Integrating factors	Is a function used to facilitate integration and solve the equation
Separation of variables	To write an equation so that each of the two variables appear on different sides of the equation.
Euler's Method	Is a first-order numerical procedure for solving ordinary differential equations (ODEs) with a given initial value
Method of undetermined Coefficients	Is an approach to finding a particular solution to certain inhomogeneous ordinary differential equations and recurrence relations. It is closely related to the annihilator method, but instead of using a particular kind of differential operator (the annihilator) in order to find the best possible form of the particular solution, a "guess" is made as to the appropriate form, which is then tested by differentiating the resulting equation.
Variation of Parametres	Is a due Lagrange and complements the method of undetermined coefficients rather well.
Laplace Transform	Is a simple method to transform a Problem consists in solving differential equations as if they were

	algebraic equations.
Linear Equations with constants coefficients	Is a differential equation that consists of a sum of several terms, each one being a derivative of the dependent variable in an independent variable, and multiplied by some constant
Separable Equations	Are equations in which each of the two variables appear in different sides of the equation.
Exact Equation	A differential equation in which there are terms that do not depend solely on $x$ . Where, $M$ is a function only of the variable $x$ , and $N$ is a function only of the variable $y$ .
Homogeneous Equations	Is are ordinary differential equations in the form $dw = 0$ .
Nonhomogeneous Equations	Are ordinary differential equations in the form $dw = g(t)$ .

### A1.2 Árvore de Conceitualização

<ul style="list-style-type: none"> <li>Nonlinear differential equation <ul style="list-style-type: none"> <li>Pincípio da linearização</li> <li>Nullclines</li> </ul> </li> <li>Linear differential equation <ul style="list-style-type: none"> <li>First order linear differential equations <ul style="list-style-type: none"> <li>Homogeneous equations <ul style="list-style-type: none"> <li>Euler's method</li> </ul> </li> <li>Homogeneous equations with constants coefficients <ul style="list-style-type: none"> <li>Laplace transform</li> </ul> </li> </ul> </li> <li>Exact equations <ul style="list-style-type: none"> <li>Separation of variable</li> <li>Laplace transform</li> </ul> </li> <li>Separable equations <ul style="list-style-type: none"> <li>Laplace transform</li> <li>Separation of variable</li> </ul> </li> <li>Linear equations with constants coefficients <ul style="list-style-type: none"> <li>Laplace transform</li> <li>Integrating factors</li> </ul> </li> <li>Second order linear differential equations</li> </ul> </li> </ul>
---

Nonhomogeneous equations

Variation of parameters

Method of undetermined coefficients

Nonhomogeneous equations with constants coefficients

Laplace transform

Differential equations in other orders

Laplace transform