



Universidade de Brasília

Faculdade de Tecnologia
Departamento de Engenharia Mecânica

Uma proposta de modelagem ontológica para a NCM
Nomenclatura Comum do Mercosul

Charles Henrique Gonçalves Santos

Brasília

2011



Universidade de Brasília

**Faculdade de Tecnologia
Departamento de Engenharia Mecânica**

Uma proposta de modelagem ontológica para a NCM
Nomenclatura Comum do Mercosul

Charles Henrique Gonçalves Santos

Dissertação apresentada como requisito parcial
para conclusão do Mestrado em Sistemas Mecatrônicos

Orientador
Prof. Li Weigang

Brasília
2011

Universidade de Brasília — UnB
Faculdade de Tecnologia
Departamento de Engenharia Mecânica
Mestrado em Sistemas Mecatrônicos

Coordenador: Prof. Carlos Humberto Llanos

Banca examinadora composta por:

Prof. Li Weigang (Orientador) — ENM/UnB
Prof. Carlos Humberto Llanos (Membro interno) — ENM/UnB
Prof. Díbio Leandro Borges (Membro externo) — CiC/UnB

CIP — Catalogação Internacional na Publicação

Gonçalves Santos, Charles Henrique.

Uma proposta de modelagem ontológica para a NCM: Nomenclatura Comum do Mercosul / Charles Henrique Gonçalves Santos. Brasília : UnB, 2011.
127 p. : il. ; 29,5 cm.

Dissertação (Mestrado) — Universidade de Brasília, Brasília, 2011.

1. Ontologia, 2. web semântica, 3. Nomenclatura Comum do Mercosul,
4. comércio eletrônico

CDU [004.8]

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília-DF — Brasil



Universidade de Brasília

**Faculdade de Tecnologia
Departamento de Engenharia Mecânica**

Uma proposta de modelagem ontológica para a NCM
Nomenclatura Comum do Mercosul

Charles Henrique Gonçalves Santos

Dissertação apresentada como requisito parcial
para conclusão do Mestrado em Sistemas Mecatrônicos

Prof. Li Weigang (Orientador)
ENM/UnB

Prof. Carlos Humberto Llanos **Prof. Díbio Leandro Borges**
ENM/UnB CiC/UnB

Prof. Carlos Humberto Llanos
Coordenador do Mestrado em Sistemas Mecatrônicos

Brasília, 15 de agosto de 2011

Agradecimentos

Aos que tornaram possível a conclusão deste trabalho: professor e orientador Li Weigang, que confiou, apoiou e me guiou na produção deste trabalho; Pedro Ilton, colega de pesquisa carismático com uma mente criativa; Alessandro Leite, mesmo em outra área de pesquisa forneceu dicas valiosas; professores Carlos Llanos e Díbio Borges, com dedicação apresentaram correções e sugestões que tornaram este trabalho mais claro e objetivo. Existem pessoas fora do meio acadêmico que merecem agradecimentos, mas para não correr o risco de ser injusto e esquecer nomes, limitar-me-ei a desejar ‘muito obrigado’ a todos vocês.

Resumo

A Nomenclatura Comum do Mercosul (NCM) é uma taxonomia utilizada pelo Brasil para a classificação de produtos, classificação esta considerada de grande importância no processo de importação e exportação de mercadorias, assim como nas operações de mercado interno. No entanto a NCM, da forma que está, não permite que a classificação de mercadorias seja uma prática simplificada, pois a atual nomenclatura utilizada no Mercosul encontra-se com os nomes dos produtos desatualizados, os quais acabam por gerar sentidos dúbios. Uma forma de solucionar o problema seria com utilização do conceito de web semântica, que propicia a coleta de dados automáticos pela internet, e a ontologia, instrumento capaz de conceber novos sentidos aos termos existentes na NCM. Assim, a proposta de ontologia deste trabalho tem por objetivo fornecer novos domínios de interesse aos produtos da NCM, o que inclui sinônimos, línguas adicionais, restrições de comercialização dos produtos (importação e/ou exportação) bem como a possibilidade de correlacionar as leis existentes com os respectivos produtos. O estudo de caso realizado comprovou a eficiência da ontologia, uma vez que forneceu respostas esperadas a todos os questionamentos até então pendentes. Desta forma, comparando a atual taxonomia utilizada pelo Mercosul na classificação de seus produtos com a proposta deste trabalho, pode-se afirmar que a utilização da ontologia otimizou a forma de uso da Nomenclatura Comum do Mercosul.

Palavras-chave: Ontologia, web semântica, Nomenclatura Comum do Mercosul, comércio eletrônico

Abstract

The Mercosur Common Nomenclature (NCM) is a taxonomy used for product classification by Brazil, being considered of great importance in the process of importing and exporting goods, as well as in internal sales operations. However, as the way the NCM is so far, it does not allow the classification of goods to be a simplified practice, as the current nomenclature used in Mercosur is with outdated product names, which ultimately generate dubious interpretations. One way to solve the problem would be to use the concept of semantic web, which provides automatic data collection via the Internet, and ontology, an instrument capable of conceiving new meanings to existing terms from NCM. Thus, the proposed ontology of this dissertation aims to provide new areas of interest to the NCM products, including synonyms, additional languages, restrictions on product sales (import / export) and the possibility to correlate the existing laws with their products. The case study proved the efficiency of ontology, since it provided the expected answers to all pending questions. Hence, comparing the current taxonomy used by Mercosur in their products classification with the purpose of this work, it can be argued that the use of ontology improved the ways on how to use the Mercosur Common Nomenclature.

Keywords: Ontology, NCM, Mercosur Common Nomenclature, e-commerce

Nomenclature

ABox	<i>Assertional Box</i>
API	emphApplication Programming Interface
DAML	DARPA <i>Agent Markup Language</i>
DW	<i>Data Warehouse</i>
HTML	<i>Hypertext Markup Language</i>
IA	Inteligência artificial
KIF	<i>Knowledge Interchange Format</i>
MDIC	Ministério do Desenvolvimento, Indústria e Comércio Exterior
MOMIS	emphMediator envirOnment for Multiple Information System
NCM	Nomenclatura Comum do Mercosul
NESH	Notas Explicativas do Sistema Harmonizado
OIL	<i>Ontology Inference Layer</i>
OKBC	<i>Open Knowledge Based Connectivity</i>
OML	<i>Ontology Markup Language</i>
OWL	<i>Web Ontology Language</i>
OWL-DL	OWL <i>Description Logics</i>
RDF	<i>Resource Description Framework</i>
RDFS	RDF <i>Schema</i>
RFB	Receita Federal do Brasil
RGC	Regras Gerais Complementares
SGBD	Sistema Gerenciador de Banco de Dados

SH	Sistema Harmonizado
SHOE	<i>Simple HTML Ontology Extension</i>
SOAP	<i>Simple Object Access Protocol</i>
SPARQL	<i>Simple Protocol and RDF Query Language</i>
SQL	<i>Structured Query Language</i>
SWSA	<i>Semantic Web Services Architecture</i>
TBox	<i>Terminological Box</i>
TEC	Tarifa Externa Comum
TIPI	Tabela de Incidência do Imposto sobre Produtos Industrializados
UDDI	<i>Universal Description, Discovery and Integration</i>
UML	<i>Unified Modeling Language</i>
UNSPSC	<i>Universal Standard Products and Services Classification</i>
URI	<i>Uniform Resource Identifier</i>
W3C	<i>World Wide Web Consortium</i>
WSDL	<i>Web Services Description Language</i>
WSFL	<i>Web Services Flow Language</i>
XML	<i>eXtensible Markup Language</i>
XOL	<i>eXchange Ontology Language</i>

Lista de Figuras

1.1	Visão geral do sistema OntoNCM	10
2.1	Exemplos de classificação no Sistema Harmonizado [1]	15
2.2	Estrutura da NCM	16
2.3	Exemplo de classificação na NCM [1]	17
2.4	Exemplo de classificações pela UNSPSC (adaptada) [49]	17
2.5	Estrutura da codificação eClass (adaptada) [31]	18
3.1	Diagrama da proposta para a web de Tim Berners-Lee (adaptada) [9] . . .	20
3.2	Pilha de referência da web semântica (adaptada) [76]	21
3.3	Camadas da web semântica, propostas em 2000 e atualizada	22
3.4	Comparação de códigos: HTML e XML	23
3.5	Códigos HTML e XML abertos no navegador Firefox	23
3.6	Exemplo de utilização do XML <i>Schema</i>	24
3.7	Um documento RDF [76]	24
3.8	Uma rede semântica RDF (adaptada) [76]	25
3.9	Exemplo de hierarquia de classes (adaptada) [76]	26
3.10	Exemplo de restrições RDFS em um RDF (adaptada) [76]	27
3.11	As origens da OWL (adaptada) [25])	28
3.12	Algumas definições de IA, organizadas em quatro categorias [56].	34
3.13	Figura parcial do agente de limpeza “Aspirador de pó” (adaptada) [56]) . .	35
4.1	Processo de criação de uma ontologia (adaptada) [22])	39
4.2	Da ontologia informal, menos expressiva até a formal, mais expressiva (adaptada) [78].	42
4.3	Hierarquia das ontologias: do mais genérico ao mais específico (adaptada) [78].	43
4.4	Arquitetura genérica de um sistema de ontologia como gerenciador de co- nhecimento (adaptada) [42].	46
4.5	Protégé sendo executado em um MacOS X.	51
4.6	Comparação entre as versões 3.x e 4.x do Protégé [62]	52
4.7	TopBraid Composer – Edição Maestro – sendo executado em um MacOS X.	52
4.8	Comparação de recursos do TopBraid Composer (adaptada) [74].	53

5.1	Metodologia 101 (adaptada) [12]	56
5.2	Um mapa resultante do sistema MOMIS [8]	59
5.3	Domínios a serem ligados à NCM	60
5.4	Classes e hierarquias da ontologia NCM [1].	61
5.5	Visão geral da ontologia proposta.	63
5.6	Classe <i>Thing</i>	64
5.7	Classe <i>Produtos</i>	65
5.8	Classes <i>Secao_01</i> e <i>Capitulo_01</i> .	65
5.9	Classe <i>C0101</i>	66
5.10	Instâncias da classe <i>C010110</i> .	66
5.11	Outra forma de exibição das instâncias da classe <i>C010110</i> .	67
5.12	Propriedade dos objetos.	67
5.13	Propriedade dos dados.	68
6.1	Aba <i>OntoGraf</i> do Protégé.	71
6.2	Resultado da busca 01019010 na aba <i>OntoGraf</i> do Protégé.	71
6.3	Resultado da busca do termo “galo” na aba <i>OntoGraf</i> do Protégé.	71
6.4	Resultado da busca do termo “galo” na aba <i>OntoGraf</i> do Protégé com dica ativada.	72
6.5	Resultado da consulta <i>DL Query</i> “temRestricaoDeImportacao min 1”	72
6.6	Resultado da consulta <i>DL Query</i> “temRestricaoDeExportacao value Ale- manha”	73
6.7	Tela com resultado de consulta no TopBraid Composer.	74
6.8	Resultado de consulta gráfico no TopBraid Composer.	74
6.9	Resultado de consulta SPARQL 1.	75
6.10	Resultado de consulta SPARQL 2.	75
6.11	Resultado de consulta SPARQL 3.	76
6.12	Classe de acesso ao OntoNCM em Jena, no ambiente Eclipse.	82
8.1	Ciclo de vida do conhecimento (adaptada) [76])	101
8.2	Tipos de conhecimentos (adaptada) [25])	103
8.3	Ontologia de uma câmera fotográfica (adaptada) [76]	104
8.4	Exemplo de tríplexes O—A—V.	105
8.5	Entendimento linguístico associado a um fato incerto.	105
8.6	Exemplo de O—A—V com fator incerto.	105
8.8	Exemplo de uma regra lógica.	106
8.7	Exemplo de conjunto Fuzzy para a idade (adaptada) [25]).	106
8.9	Exemplo de uma rede semântica (adaptada) [25]).	107
8.10	Visão funcional da lógica descritiva (adaptada) [76])	109
8.11	Exemplo do uso de uma <i>TBox/ABox</i> .	110

Sumário

1	Introdução	8
1.1	Motivação e justificativa	9
1.2	Objetivos	10
1.3	Metodologia	11
1.4	Organização do trabalho	11
2	Definição e Aplicação da NCM	13
2.1	Histórico, estrutura e aplicação das nomenclaturas	13
2.2	Sistema Harmonizado (SH)	14
2.3	Nomenclatura Brasileira de Mercadorias baseada no Sistema Harmonizado (NBM/SH)	15
2.4	Nomenclatura Comum do Mercosul (NCM)	16
2.5	Outras nomenclaturas	17
2.5.1	UNSPSC	17
2.5.2	eClass	18
3	Web Semântica	19
3.1	Linguagens de desenvolvimento	21
3.1.1	XML e XML <i>Schema</i>	22
3.1.2	RDF e RDF <i>Schema</i>	25
3.1.3	DAML+OIL	27
3.1.4	OWL	28
3.1.4.1	OWL- <i>Lite</i>	29
3.1.4.2	OWL-DL	29
3.1.4.3	OWL- <i>Full</i>	29
3.1.5	OWL vs. RDF	30
3.1.6	Outras linguagens	30
3.2	Linguagens de consulta	31
3.2.1	SPARQL	32
3.3	Agentes	34
3.3.1	Racionalidade	35
3.3.2	Onisciência	36

3.4	Serviços da web semântica	36
4	Ontologias	38
4.1	Por que desenvolver uma ontologia?	39
4.2	O que é uma ontologia?	40
4.3	Benefícios das ontologias	43
4.4	Estado da arte	44
4.5	Construção de ontologias	47
4.6	Ferramentas para o desenvolvimento de ontologias	50
4.6.1	Protégé	51
4.6.2	TopBraid Composer	52
5	Metodologia e modelagem	54
5.1	Metodologia	55
5.1.1	Passo 1: Determine o domínio e escopo da ontologia	56
5.1.2	Passo 2: Considere a possibilidade de reutilizar ontologias existentes	58
5.1.3	Passo 3: Enumere os termos importantes na ontologia	59
5.1.4	Passo 4: Defina as classes e suas hierarquias	60
5.1.5	Passo 5: Defina as propriedades de interesse das classes	61
5.1.6	Passo 6: Defina as restrições das propriedades	62
5.1.7	Passo 7: Crie as instâncias	62
5.2	Modelagem	62
5.2.1	Visão geral	63
5.2.2	Classes	64
5.2.3	Propriedades do objeto	67
5.2.4	Propriedade dos dados	68
6	Estudo de caso	70
6.1	Testes do modelo	70
6.1.1	Pelo Protégé	70
6.1.2	Pelo TopBraid Composer	73
6.2	Validação do modelo	76
6.3	Código de programação para acesso à ontologia	81
6.4	Análise dos resultados	82
7	Conclusão	83
7.1	Avaliação do modelo proposto	83
7.2	Dificuldades observadas no projeto	84
7.3	Contribuições e trabalhos futuros	84
7.3.1	Contribuições	84
7.3.2	Trabalhos futuros	84

8	Apêndices	86
8.1	Regras e roteiro para interpretação e classificação do SH	86
8.1.1	Regras Gerais para Interpretação do Sistema Harmonizado	86
8.1.1.1	Regras Gerais Complementares (RGC)	88
8.1.1.2	Regra Geral Complementar da TIPI (RGC/TIPI)	88
8.1.2	Roteiro para Efetuar uma Classificação	88
8.1.2.1	Passo 1	89
8.1.2.2	Passo 2	89
8.1.2.3	Passo 3	90
8.1.2.4	Passo 4	91
8.1.2.5	Passo 5	91
8.1.2.6	Passo 6	91
8.1.2.7	Passo 7	91
8.2	Sumário das Seções e Capítulos	92
8.3	Representação do conhecimento	98
8.3.1	Ciência cognitiva	99
8.3.2	Informação e conhecimento	100
8.3.3	Tipos de conhecimento	102
8.3.4	Técnicas de representação do conhecimento	103
8.3.4.1	Tríplice objeto, valor e atributo	104
8.3.4.2	Fatos incertos	105
8.3.4.3	Fatos difusos	105
8.3.4.4	Regras lógicas	106
8.3.4.5	Redes semânticas	106
8.3.4.6	Baseada em quadros	107
8.3.5	Linguagens para representação do conhecimento	107
8.3.5.1	Lógica Proposicional	108
8.3.5.2	Lógica de Primeira Ordem	108
8.3.5.3	Forma intercambiável de conhecimento (KIF)	109
8.3.5.4	Lógica descritiva	109
8.3.5.5	Baseada em quadros	110
8.3.5.6	Baseada em Regras	110
	Referências Bibliográficas	113

Capítulo 1

Introdução

O crescimento da Internet tem colocado à disposição da população quantidades de informações inimagináveis, sobre todos os assuntos. O fluxo de informações ganha espaço nos celulares e, com isto, percebe-se a popularização e intensificação da rede mundial de computadores no Brasil. No entanto, estas informações ainda não são suficientes para a solução automatizada dos mais variados problemas. A informação disponível na internet é, quase em sua totalidade, voltada para o entendimento humano – mas este não é capaz de filtrar e unir a informação que realmente ajude a solucionar problemas em tempo hábil. Uma das soluções propostas para resolver este problema foi a web semântica, que tem o intuito de tornar os dados disponíveis na rede processáveis por máquinas e, assim, auxiliar na busca de informações relevantes, selecionadas e contextualizadas. A web semântica é uma área da Inteligência Artificial que opera em conjunto com outras tecnologias para seu pleno funcionamento.

Um dos benefícios da tecnologia e do acesso à informação facilitado é possibilitar que pequenos produtores possam exportar as mercadorias produzidas, ou mesmo importar matéria prima para seus trabalhos. Estes, no entanto, devem observar que o comércio brasileiro deve seguir a regras de codificação da Nomenclatura Comum do Mercosul (NCM) – com vasto conteúdo disponibilizado na rede mundial de computadores de forma não estruturada. O ponto de partida para a classificação de mercadoria é o conhecimento da mercadoria em detalhes, pois pormenores – tais como a constituição do bem, função de uso, se é um ser vivo ou morto – podem fazer diferença no código a ser adotado, até porque a codificação incorreta, de acordo com as leis brasileiras, é passível de punição.

Ao se classificar um determinado produto, também devem ser observadas as “Regras Gerais para Interpretação do Sistema Harmonizado” – um conjunto de orientações que, passo a passo, elimina etapas até se chegar à codificação apropriada – que definem os critérios a serem obrigatoriamente observados para determinação do código correto de determinado produto. Conhecidas as mercadorias e as regras de interpretação, devem ser consultadas as seções, capítulos existentes para, então, se chegar à classificação, que somente será definida pelos textos das posições e notas de seção e capítulo. Definida a posição, dever-se-á efetuar, então, o enquadramento do código pelos demais níveis de

desdobramento da nomenclatura.

Como pode-se ver, a classificação de mercadorias é uma tarefa complexa. Por meio da nomenclatura de um produto pode-se conhecer os tributos aplicáveis, tratamentos administrativos, restrições na comercialização, acompanhamento de estatísticas e manter o controle das operações por meio de sistemas informatizados. Neste sentido, o objetivo deste trabalho é propor um modelo ontológico para a NCM, de forma que esta possa ser usada futuramente pela web semântica e, assim otimizar a forma de uso da Nomenclatura Comum do Mercosul (NCM).

1.1 Motivação e justificativa

A Nomenclatura Comum do Mercosul é uma classificação obrigatória para os produtos comercializados no Brasil, sejam estes para o comércio interno, exportação ou importação. As informações para classificação podem ser obtidas por meio da internet, mas estão disponíveis em formas não estruturadas.

Neste contexto, a NCM fornece um padrão de classificação oficial para os produtos dos países que integram o Mercosul. A obrigatoriedade da classificação para a comercialização de produtos nacionais ou importados no Brasil, unida ao teor puramente técnico dos termos constantes na NCM, torna a classificação de produtos e serviços uma tarefa complexa. Além disso, a classificação incorreta pode acarretar em multas, erros na cobrança de impostos, violação das restrições aduaneiras para entrada ou saída de mercadorias.

Hoje, a forma de resolver este problema é por meio da contratação de consultorias ou empresas especializadas no comércio interno, importação e exportação – o Banco do Brasil, por exemplo, fornece este serviço para os correntistas comerciantes. Assim, a respeito da problemática apresentada, é pertinente questionar o que poderia ser implementado para:

- Incluir informações regionais na NCM de forma que os fabricantes e comerciantes – que não dominem os termos técnicos previstos naquela nomenclatura – entendam e classifiquem corretamente os produtos sem o auxílio de consultores?
- Aperfeiçoar e tornar a pesquisa de produtos existentes e, ao mesmo tempo, torná-la disponível em vários idiomas?
- Minerar periodicamente a informação existente na internet, de forma não estruturada, referente às leis, aos impostos e restrições aduaneiras aplicáveis a determinado produto? Como relacionar estes resultados com determinado código da NCM?
- Tornar possível a comunicação da NCM com as nomenclaturas de outros países, tal como o *Universal Standard Products and Services Classification* (UNSPSC¹) ou o *eClass*²?

¹Código de classificação usado nos Estados Unidos (<http://www.unspsc.org>).

²Código para classificação utilizado na Europa (<http://www.eclass-online.com>).

Estas são algumas questões que justificam o uso de web semântica com ontologia³, uma vez que a informação a ser relacionada com a NCM está disponível na internet de forma não estruturada. É nesse sentido que as ontologias se mostram de grande utilidade, eis que são utilizadas em inteligência artificial, web semântica, engenharia de *software* e arquitetura da informação, como forma de representação do conhecimento sobre o mundo, ou parte deste. Contudo, no que se refere à utilização de tal conceito para a Nomenclatura Comum do Mercosul, o passo mais importante é a criação de um modelo ontológico aplicável, proposta idealizada neste trabalho.

1.2 Objetivos

O objetivo geral deste trabalho é desenvolver um modelo ontológico – parte destacada no futuro sistema OntoNCM exibido na Figura 1.1 – para a Nomenclatura Comum do Mercosul. Isto beneficiará diretamente o *e-commerce* para o comércio interno, importações e exportações brasileiras. Para atingir o objetivo geral, será necessário alcançar os seguintes objetivos específicos:

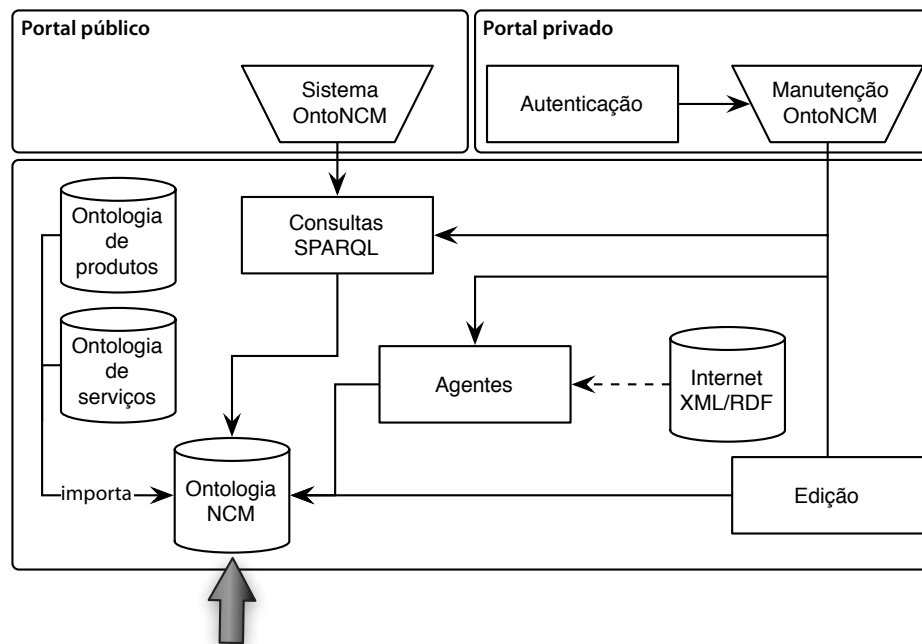


Figura 1.1: Visão geral do sistema OntoNCM

1. Propor um modelo ontológico para a NCM
2. Testar a utilidade da ontologia desenvolvida com a linguagem Java ou nos próprios ambientes de edição de ontologias.

³Embora os termos taxonomia e ontologia às vezes sejam usados como sinônimos, há uma diferença fundamental: a taxonomia indica apenas a relação classe/subclasse; já a ontologia é capaz de descrever o domínio completamente.

Os passos norteadores para se atingir aos objetivos específicos são:

1. Pesquisar a utilidade da NCM, em todos os aspectos, e traçar um conceito básico das codificações UNSPSC e eClass.
2. Pesquisar e determinar quais as tecnologias, linguagens e ferramentas existentes que dão suporte à web semântica e à ontologia que devem ser usadas para a solução do problema.
3. Avaliar os domínios que podem ser relacionados com a NCM, bem como as formas de mapear estas informações por meio de uma ontologia.

1.3 Metodologia

A metodologia utilizada neste trabalho será a pesquisa dos temas relevantes e, principalmente, das ferramentas disponíveis para utilização em conjunto com a web semântica e a ontologia. Frise-se que apesar de se ter realizado uma breve análise sobre prós e contras de cada ferramenta este não é o foco desta pesquisa e, por isto, tais impressões não foram relatadas. Deste modo, o foco foi mantido tão somente nas ferramentas que foram mais práticas e objetivas para a solução do problema proposto.

1.4 Organização do trabalho

Este trabalho está organizado da seguinte forma: o Capítulo 1 traça uma breve introdução ao tema; o Capítulo 2 – Definição e Aplicação da NCM – tem o objetivo definir a NCM, bem como sua aplicação, com intuito de compreender a origem do sistema, as regras para a interpretação, expor, a fim de melhor compreensão, as nomenclaturas utilizadas em outros países, como a UNSPSC e a eClass; o Capítulo 3 – Representação do conhecimento – traz conceitos do que é a representação do conhecimento, a diferença entre a mera informação e o conhecimento em si, os tipos de conhecimento, as técnicas de representação para uso na web semântica e os formalismos possíveis para utilização com a ontologia; o Capítulo 4 – Web Semântica – define o que é a web semântica, quais são as linguagens de desenvolvimento possíveis, quais são as linguagens de consultas e os serviços que fornecem apoio a esta tecnologia; o Capítulo 5 – Ontologias – procura entender os conceitos de ontologia, o porquê se produzir uma, quais os benefícios do uso desta tecnologia, qual o estado da arte, a boa prática na construção de uma ontologia e sobre os ambiente de criação Protégé e TopBraid Composer, líderes na criação de ontologias; o Capítulo 6 – Metodologia e modelagem – trata da metodologia utilizada para a solução do problema, bem como sobre o modelo de ontologia proposto para a solução do problema; o Capítulo 7 – Estudo de caso – demonstra resultados de consultas em *DL Query* e SPARQL sobre o modelo de ontologia proposto, expõe a validação de utilidade da proposta elaborada, propõe o código fonte de uma classe em Java, para a produção de aplicativos utilizem a

ontologia proposta e tece uma análise crítica dos resultados obtidos; por fim, o Capítulo 8 – Conclusão – conclui a pesquisa, propondo usos do modelo ontológico produzido em sistemas relacionados à mecatrônica.

Capítulo 2

Definição e Aplicação da NCM

Neste capítulo são apresentadas a definição, as aplicações, o histórico, a evolução e a estrutura da Nomenclatura Comum do Mercosul – mais conhecida pela sigla NCM. Pretende-se, também, demonstrar uma visão geral das regras e roteiros para se efetuar corretamente a classificação de determinado produto, com base nas leis correlatas.

2.1 Histórico, estrutura e aplicação das nomenclaturas

Por volta de 1970 estudiosos do setor aduaneiro sentiram a necessidade de classificar mercadorias, de forma a facilitar a fiscalização e tributação pelas autoridades competentes, nas operações que envolviam mercadorias. As primeiras classificações foram construídas de forma simplificada, com base em listas alfabéticas de um lado e respectivos tributos de outro [1].

A evolução natural do comércio ocasionou um aprimoramento na forma de tributação, bem como um aumento considerável da tabela de classificação existente. De acordo com Aduaneiras [1], a partir desse momento fez-se necessário adotar nomenclaturas aduaneiras uniformes que pudessem garantir, entre outras coisas: a criação de uma classificação internacional uniforme para satisfazer aos interesses de quem adotasse tal nomenclatura; criação de uma “linguagem aduaneira comum” aceita por todos; definição de uma forma sistêmica de cobrir todas as mercadorias comercializadas naquela época e a facilitação das análises e estatísticas do comércio internacional.

Classificar mercadorias é condição necessária para o correto procedimento das importações, exportações e operação no mercado interno. A classificação auxilia nos tratamentos administrativos, restrições de comercialização e na manutenção de um único banco de dados para ser usado nos mais variados sistemas [1].

2.2 Sistema Harmonizado (SH)

O Sistema Harmonizado de Designação e de Codificação de Mercadorias – ou simplesmente Sistema Harmonizado (SH) – é uma nomenclatura de seis dígitos aceita internacionalmente para classificar mercadorias [1].

De acordo com o Ministério do Desenvolvimento, Indústria e Comércio (MDIC), o objetivo do SH foi promover o desenvolvimento do comércio internacional, aprimorar a coleta de dados, promover a comparação e análise estatísticas – principalmente para o comércio exterior –, facilitar as negociações comerciais internacionais e produzir informações para uso nos diversos intervenientes no comércio internacional, uma vez que o SH “permite que sejam atendidas as especificidades dos produtos – tais como origem, matéria constitutiva e aplicação – em um ordenamento numérico lógico, crescente e de acordo com o nível de sofisticação das mercadorias”[44]. Segundo Aduaneiras [1], o SH é um código que foi concebido para auxiliar, dentre outras aplicações, na elaboração das tarifas aduaneiras, na previsão de fretes e para a criação de estatísticas do comércio de importação e exportação. Percebe-se que as ideias das duas fontes citadas não são divergentes, mas complementares.

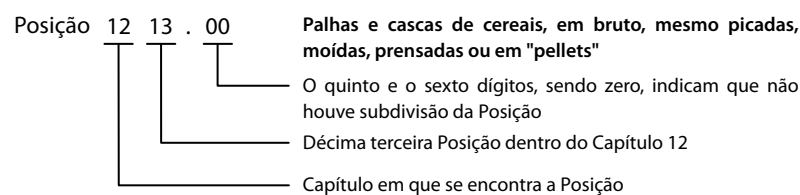
O SH é composto pelas Regras Gerais para Interpretação do SH, pelas Notas Explicativas do SH (NESH) e por uma nomenclatura, que apresenta basicamente uma estrutura lógica de 97 capítulos ordenado em 21 seções [1, 44]:

- Nomenclatura: 21 seções compostas por 97 capítulos¹, além das Notas de Seção, das Notas de Capítulo e das Notas de Subposição. A divisão dos capítulos considera posição e subposição, por meio de códigos numéricos;
- Regras Gerais para Interpretação do Sistema Harmonizado: São as regras gerais para a correta classificação das mercadorias na nomenclatura; e
- Notas Explicativas: Esclarecimentos adicionais e interpretação, de forma a estabelecer detalhes sobre o alcance e conteúdo do SH.

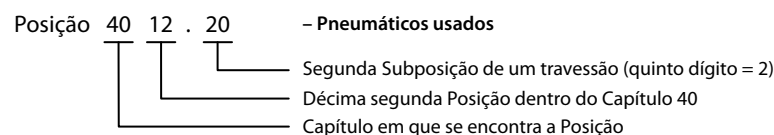
Cada posição da nomenclatura SH está identificada por quatro dígitos, onde os dois primeiros indicam o capítulo e os dois seguintes indicam a posição dentro do respectivo capítulo. As posições podem ser divididas em duas ou mais subposições, que podem ser classificadas como “subposições simples” se tiver apenas um travessão (–) ou “subposições compostas” se tiver dois travessões (––) [1]. Exemplos do uso da codificação podem ser vistos na Figura 2.1.

Nota-se pela 2.1a que se uma determinada posição não está desdobrada em subposições, esta deve ser representada por zeros (1213.00) [1].

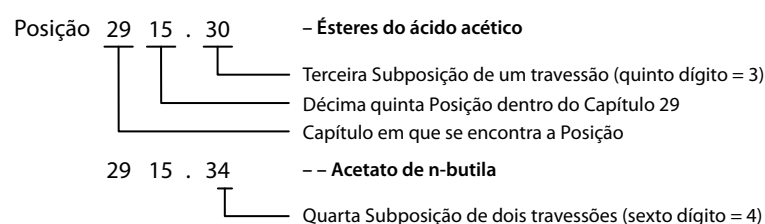
¹Existem Capítulos reservados para utilização futura no SH. O Capítulo 77 foi reservado pra eventual utilização futura, os Capítulos 98 e 99 foram reservados para usos especiais e, no Brasil, o Capítulo 99 é destinado ao registro de operações especiais na exportação [44].



(a)



(b)



(c)

Figura 2.1: Exemplos de classificação no Sistema Harmonizado [1]

2.3 Nomenclatura Brasileira de Mercadorias baseada no Sistema Harmonizado (NBM/SH)

Em 31/10/1986 o Brasil aderiu à Convenção Internacional sobre o Sistema Harmonizado de Designação e de Codificação de Mercadorias, comprometendo-se a utilizar a nomenclatura SH no lugar do sistema até então vigente – Nomenclatura do Conselho de Cooperação Aduaneira. Foi facultado aos países em desenvolvimento, partes contratantes, a aplicação em parte ou na totalidade das subposições do SH, pelo tempo necessário. No entanto o Brasil optou por utilizar a nomenclatura em sua forma completa, já no ano de 1989 [1].

Para a elaboração da nova Nomenclatura Brasileira de Mercadorias foram procedidas as adaptações necessárias, inclusive a introdução de subdivisões, em nível de itens e subitens, acrescentadas e codificadas após o código numérico de 6 dígitos do Sistema Harmonizado, de sorte a obter melhor detalhamento das mercadorias e respectivas classificações.

A aplicação dessa nomenclatura ocorrerá nas operações de importação e exportação, no comércio de cabotagem, na cobrança de impostos e nos demais casos previstos em lei.

Relativamente à cobrança de impostos cabe destacar sua aplicação para a Tabela de Incidência do Imposto sobre Produtos Industrializados.

A partir de 01/01/97 a Nomenclatura Comum do Mercosul (NCM) passou

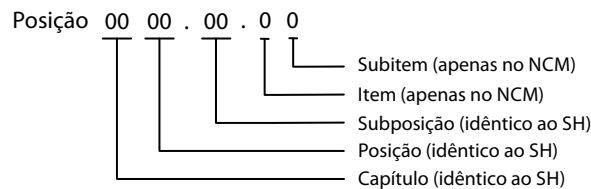


Figura 2.2: Estrutura da NCM

a substituir a Nomenclatura Brasileira de Mercadorias (NBM/SH) Aduaneiras [1, p. 5, correções nossas].

2.4 Nomenclatura Comum do Mercosul (NCM)

Segundo o Ministério do Desenvolvimento, Indústria e Comércio Exterior (MDIC) a Nomenclatura Comum do Mercosul foi adotada pelo Brasil, Argentina, Paraguai e Uruguai em 1997. O Decreto n 6.006/06 [53], regulador da Tabela de Incidência do Imposto sobre Produtos Industrializados (TIPI), foi publicado em 2006 e tem relação direta com a NCM. No Brasil, com algumas exceções², toda mercadoria para ser comercializada deve ser classificada por meio da NCM. Cabe lembrar que a NCM foi utilizada na elaboração da Tarifa Externa Comum (TEC), além de ser utilizada para fornecer outras disposições legais [1].

A NCM manteve a estrutura básica da nomenclatura SH – com 97 capítulos divididos em 21 seções –, sendo a principal modificação verificada quando do aumento das subdivisões em nível de itens e subitens. Atualmente NCM utiliza oito dígitos, sendo os seis dígitos iniciais idênticos aos do SH (Figura 2.2). O sétimo dígito da NCM representa os itens; o oitavo os subitens. Da mesma forma do SH, se não existirem subdivisões a serem representadas, estas posições devem ser marcadas com o dígito 0 (zero).

A Figura 2.3 ilustra um exemplo da classificação da NCM. O produto em questão tem o código de classificação 1704.90.10, onde as partes do código [1]:

- 1704 Representa a posição NCM, idêntica à do SH, onde os dois primeiros dígitos representam o Capítulo 17 “Açúcares e produtos de confeitaria” e os dois últimos dígitos representam a Seção 04 “Produtos de confeitaria, sem cacau (incluído o chocolate branco)”.
- 1704.90 Representa a posição NCM acrescida da 2 subposição, não se trata de subposição composta por ser o último número do grupo igual a 0 (zero).

²Bens que se incluem na balança comercial, mas de difícil identificação: comércio de navios e aeronaves que se dedicam ao tráfego internacional; comércio de plataformas de perfuração que atuam em águas internacionais; aprovisionamento de navios e aeronaves e demais veículos de transporte; encomendas postais; bens móveis de migrantes; doações; bens para reparos e movimento de lojas francas (*freeshop*) [43].

Código NCM		Mercadoria
Posição e Subposição	Itens e Subitens	
1704		PRODUTOS DE CONFEITARIA, SEM CACAU (INCLUÍDO O CHOCOLATE BRANCO)
1704.10	00	– Gomas de mascar, mesmo revestidas de açúcar – Outros
1704.90	10	Chocolate branco
	20	Caramelos, confeitos, dropes e pastilhas
	30	Outros

Figura 2.3: Exemplo de classificação na NCM [1]

UNSPSC	Descrição do produto
43171903	PIII 800/133 S1 256
43172402	S170B 17" 60kHz Color Monitor

Figura 2.4: Exemplo de classificações pela UNSPSC (adaptada) [49]

1704.90.10 Representa a posição NCM, acrescida do 1 item dentro da 2 subposição simples, não dividido em subitens compostos por ser o último número do grupo igual a 0 (zero): “Chocolate branco”.

As regras e roteiros para a interpretação e classificação no SH podem ser vistas no Anexo ???.

2.5 Outras nomenclaturas

2.5.1 UNSPSC

A *United Nations Standard Products and Services Code* (UNSPSC³) fornece um meio aberto, global e eficiente para a classificação de produtos e serviços [55].

O padrão UNSPSC contém cerca de vinte mil produtos e serviços categorizados em uma taxonomia de quatro níveis – cite-se como exemplo o código 43171903, o qual faz referência à *Central Processing units, motherboards or daughterboards* – 43171903 é uma subcategoria da 431719, que é uma subcategoria da 4317 que, por fim, é uma subcategoria da 43 [49].

As descrições dos nomes dos produtos na UNSPSC são curtos e específicos, diferentemente da nomenclatura utilizada no Brasil, NCM. Uma amostra típica de um produto classificado pela UNSPSC pode ser visto na Figura 2.4.

Dogac et al. [20] comparam o uso de uma taxonomia e de uma ontologia para o código UNSPSC. Explicam que, embora os termos “taxonomia” e “ontologia” sejam usados como sinônimos, existe uma grande diferença: uma taxonomia é capaz de indicar somente as relações entre classes e subclasses enquanto a ontologia pode descrever qualquer domínio completamente.

³<http://www.unspsc.org>

2.5.2 eClass

O eClass⁴, iniciativa da Alemanha sem fins lucrativos [31], é outro padrão para codificação de produtos e serviços – também organizado em uma taxonomia de quatro níveis, assim como a UNSPSC ou a NCM – tem cerca de doze mil produtos e serviços categorizados [49].

O eClass é um sistema hierárquico para a organização de materiais, produtos e serviços, de acordo com uma estrutura lógica e com nível de detalhe correspondente às propriedades específicas de cada produto. Esta codificação permite que se atribua a um código, todos os produtos e serviços, respeitando os quatro níveis taxonômicos propostos pelo padrão. O sistema fornece também sinônimos, de forma a permitir que o produto ou serviço desejado seja localizado mais facilmente dentro da classificação [31]. A Figura 2.5 descreve a pirâmide base do sistema eClass.

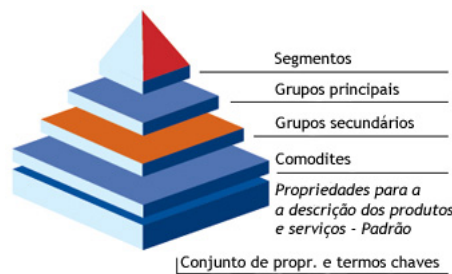


Figura 2.5: Estrutura da codificação eClass (adaptada) [31]

Segundo Omelayenko [49], enquanto a codificação UNSPSC classifica os produtos com a visão dos fornecedores, o eClass o faz por meio da visão dos consumidores.

⁴<http://www.eclass.de>, *International Standard for the Classification and Description of Products and Services*

Capítulo 3

Web Semântica

Tim Berners-Lee, em 1989, propôs a organização da internet como conhecemos hoje. Um dos diagramas publicado está ilustrado na Figura 3.1 [9]. Mais tarde, Tim Berners-Lee, Jim Hendler e Ora Lassila publicaram, em 2001, na *Scientific American* o que seria a próxima geração da internet: *The semantic web* (A web semântica)[10].

Em 2006 a publicação foi revisada por Shadbolt et al. [59], onde foi demonstrado como a web semântica pode ser utilizada para construir sistemas inteligentes, transformando a informação em conhecimento entre plataformas distintas. A web semântica apresenta a próxima geração da internet, por meio de uma camada de dados compreensíveis para agentes de software, motores de busca inteligente e serviços da internet [69, 34].

A web sintática – guiada por regras e formalidades – é uma fonte informações praticamente ilimitadas e disponível em linguagem natural e, por isso, incompreensível pelas máquinas [17]. Os homens, no entanto, têm capacidade limitada para aproveitar toda a informação da internet e seriam os maiores beneficiados se as máquinas pudessem analisar e processar a informação existente, dando resultados mais precisos [46]. A web semântica – guiada por interpretações, inferências e relações – tem o objetivo de tornar a informação presente na web sintática compreensível pelas máquinas, por meio da extração e reestruturação da informação disponível na rede [25]. A web semântica é uma malha de informações, de forma a serem facilmente processáveis por máquinas de forma modular e em escala global [50, 17].

Para fazer com que a web semântica realmente forneça conhecimento útil e de valor para a sociedade é necessário que se encontre formas eficientes de acessar e extrair o conhecimento existente na internet, por meio de ferramentas especializadas – normalmente guiadas pela ontologia, com capacidade de extrair informações de textos não estruturados – que anotem sobre o que representa dado conteúdo, relacionando-os e dando-os a devida representação que os identifique como conhecimento [2].

Nos ambientes de banco de dados existem dois métodos tradicionais para se integrar

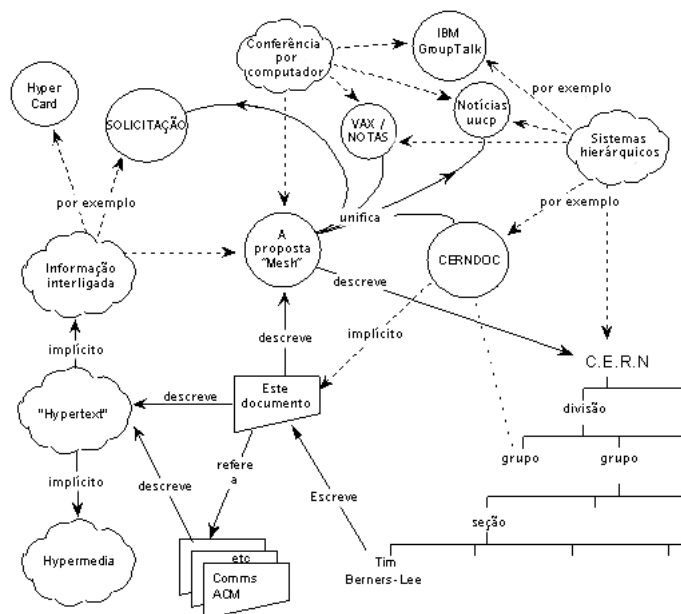


Figura 3.1: Diagrama da proposta para a web de Tim Berners-Lee (adaptada) [9]

base de dados: a criação de um *Data Warehouse*¹ (DW) ou a transição de consultas². Este métodos envolvem a utilização de um mapa entre o *modelo de dados* que represente fielmente os dados e um *modelo de dados comum*, que orientará na união das diferentes base de dados [5]. Sabe-se que na web não existe um *modelo de dados* único, mas a necessidade de relacionar esta quantidade quase infinita de informação disponível gratuitamente fez com que ferramentas e padrões fossem criadas para esta finalidade. A web semântica, com utilização do conceito ontologia, nasceu para suprir esta necessidade [17].

A *World Wide Web Consortium* (W3C³) – organização responsável pelo desenvolvimento e divulgação dos padrões para a internet – definiram os padrões a serem seguidos pela web semântica [5] e finalizou, em 2006, as especificações para a linguagem de ontologias OWL [24, 30].

Uma pirâmide das tecnologias utilizadas pela web semântica é ilustrada na Figura 3.2 [76]: .

- A camada mais baixa da pirâmide é familiar a todos, com informações voltadas para humanos e baixos níveis de automação.
- Os *Web Services* deram o primeiro passo para o processamento de informações sobre a internet, mas não fornecem qualquer tipo de inteligência para a informação

¹Envolve a transformação dos diferentes tipos de dados em padrões, uniformizados e armazenados. Se por um lado o DW é eficiente por eliminar problemas tais como lentidão nas consultas, indisponibilidade da origem, padronização dos dados, eliminação da redundância, etc., por outro sua manutenção é custosa, já que tanto os dados quanto a estrutura devem ser atualizados periodicamente [5].

²Focada na tradução das consultas, o que não modifica os dados originais, está focada apenas na adaptação das consultas, o que inspira que se utilize um mediador – *middleware* – entre a resposta da base de dados e a transição em tempo de execução para um mapa de tradução. Este método tem como porto forte a facilidade de manutenção e a sincronização de dados automática, inclusive se novas base de dados forem adicionadas, mas tem como pontos negativos o desempenho da consulta ser lento qe a necessidade de as bases de dados originais estarem sempre disponíveis [5].

³<http://www.w3c.org>



Figura 3.2: Pilha de referência da web semântica (adaptada) [76]

existente. Fornecem apenas padrões para que os mais variados *softwares* consigam acessar as informações disponíveis na internet.

- As ontologias dão capacidade de raciocínio sobre as informações existentes na internet, fornecendo um meio para que a informação sintática tenha semântica por meio de classificações e anotações.
- Os agentes são programas que agem em nome de uma entidade, e podem ser humanos ou *softwares*. Estes agentes são responsáveis pelo intercâmbio de conhecimento na web semântica.
- Os serviços da web semântica fornecem meios para a construção de aplicações para a web semântica. Tais aplicações são compostas de serviços que encapsulam os agentes, sistemas de raciocínio e bases de conhecimento.
- Na construção de uma web semântica não é necessário passar por todas as técnicas descritas na pirâmide. De fato, cada aplicação requer o uso de um subconjunto destas técnicas para alcançar o resultado desejado.

3.1 Linguagens de desenvolvimento

As principais tecnologias da web semântica se encaixam em um conjunto de especificações dispostos em camadas [5, 14, 33, 17, 25], conforme ilustra a Figura 3.3. Das tecnologias usadas destacam-se o *Resource Description Framework* (RDF), o *RDF Schema* e a *Web Ontology Language* (OWL), responsável por armazenar o conhecimento [5, 33]. Pode-se usar a linguagem de consulta padronizada *Stands Protocol and RDF Query Language* (SPARQL⁴), o que permite consultar de forma eficiente documentos RDF e OWL [15].

⁴SPARQL, pronunciada em inglês como “sparkle”, é uma linguagem de consulta RDF. Ela foi padronizada pelo *RDF Data Access Working Group* (DAWG) da *World Wide Web Consortium* (W3C), e é

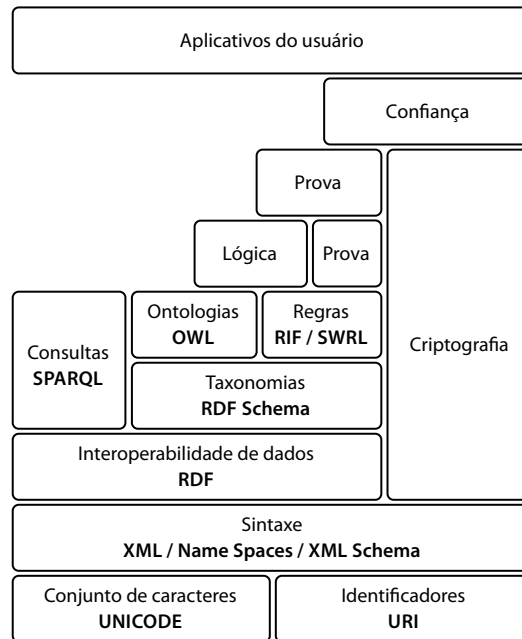


Figura 3.3: Camadas da web semântica, propostas em 2000 e atualizada

Observa-se na Figura 3.3 que as linguagens de alto nível usam a sintaxe e semântica das linguagens de baixo nível. Pode-se dizer que todas as linguagens para web semântica usam como base a *eXtensible Markup Language* (XML) [15], pois a XML é uma metalinguagem para a representação de outras linguagens na web semântica [25, 24]. Por exemplo, tem-se a XML *Schema* que define classes de documentos XML usando a sintaxe XML. Já o RDF, também baseado no XML, provê um *framework* para a representação de metadados sobre recursos da internet. Também são baseados nos XML as linguagens RDF *Schema* (RDFS), OWL e outras [15].

3.1.1 XML e XML *Schema*

Quem deseja desenvolver para a web semântica deve entender e concordar com a sintaxe e semântica XML [15]. Uma vez produzido um aplicativo fora destes padrões, as adequações de código para adaptá-lo aos padrões desejados pode ser custoso [25].

Como dito em na seção anterior, todas as linguagens atuais utilizadas na web semântica têm sintaxe baseada nos padrões XML, pois esta permite a especificação e marcação de documentos de forma inteligível às máquinas [51]. O XML permite a criação de documentos com estruturas arbitrárias, flexibilização das marcações utilizadas nos documentos e a criação de documentos autoexplicativos, ao contrário do *Hypertext Markup Language* (HTML), que tem as marcações padrões e pré-definidas. A Figura 3.4 faz a comparação de um fragmento de código escrito em HTML e outro escrito em XML, com o mesmo objetivo. Ao usar um navegador para abrir os códigos da Figura 3.4, observa-se que o

considerada a parte chave para a tecnologia de web semântica. Em 15 de janeiro de 2008 a SPARQL foi oficializada como Recomendação da W3C (<http://en.wikipedia.org/wiki/SPARQL>, acessado em 2 de novembro de 2010).

```

<HTML>
  <BODY>
    <UL>
      <LI> <EM>Departamento de Engenharia Mecatrônica</EM>,
        UnB, Brasília, DF.
    </LI>
  </UL>
</BODY>
</HTML>

```

(a) Código HTML

```

<UNIVERSIDADE>
  <SIGLA> UnB </SIGLA>
  <DEPARTAMENTO> Engenharia Mecatrônica </DEPARTAMENTO>
  <CIDADE> Brasília </CIDADE>
  <ESTADO> DF </ESTADO>
</UNIVERSIDADE>

```

(b) Código XML

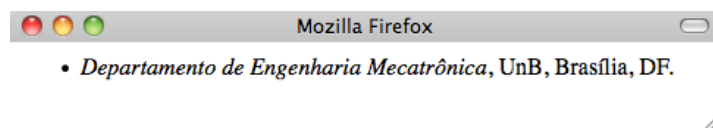
Figura 3.4: Comparação de códigos: HTML e XML

HTML é orientado à apresentação, enquanto o XML é orientado à estrutura, como mostra a Figura 3.5.

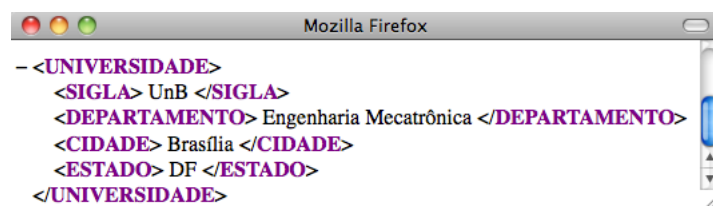
O XML sozinho não tem a capacidade de interpretação dos dados. A informação da Figura 3.4b foi codificada em sintaxe genérica e sobre a semântica dos dados nada foi especificado. Assim, nota-se que o XML puro é destinado apenas à estruturação de documentos, não possibilitando que as máquinas o interpretem dada a falta do vocabulário de dados [15].

O uso de XML para troca de dados entre aplicativos distintos requer primeiramente o uso de um vocabulário, termos e significados comuns entre as aplicações envolvidas. Parte deste problema pode ser resolvido utilizando o XML *Schema*, que provê um mecanismo para especificar a estrutura de documentos XML [25].

O XML *Schema* oferece um *framework* para a criação de documentos XML, descrevendo as marcações utilizadas, elementos e atributos de uma determinada categoria de documento. Além disso, o XML *Schema* pode validar a estrutura do documento, as restrições, os tipos de dados personalizados, o número de ocorrências de um elemento filho, valores padrões, grupo de opções para determinado atributo, etc [15].



(a) Código HTML aberto no Firefox



(b) Código XML aberto no Firefox

Figura 3.5: Códigos HTML e XML abertos no navegador Firefox

<xsd:schema xmlns:xsd="http://www.w3.org/1999/XMLSchema">	1
<xsd:element name="UNIVERSIDADE" type="UNIVERSIDADES"/>	2
<xsd:complexType name="UNIVERSIDADES">	3
<xsd:element name="SIGLA"	4
type="xsd:string"	5
minOccurs="1"	6
maxOccurs="1"/>	7
<xsd:element name="DEPARTAMENTO"	8
type="xsd:string"	9
minOccurs="1"	10
maxOccurs="unbounded"/>	11
...	12
<xsd:element name="ESTADO" type="xsd:string"/>	13
</xsd:complexType>	14
</xsd:schema>	15

Figura 3.6: Exemplo de utilização do XML *Schema*

A sintaxe do XML *Schema* é idêntica à do XML, com a utilização da marcação *namespace* que é declarada utilizando o atributo *xmlns* [25]. Um exemplo de utilização do XML *Schema* é apresentado na Figura 3.6. No exemplo o *namespace* “xsd” foi declarado e usado em todas as definições dos elementos.

Walton [76] explica que um *namespace* é uma conveniência para evitar a digitação repetida de uma URI, o que seria muito demorado. Outro uso comum é para remover a ambiguidade de elementos que são importados de locais diferentes, mas com o mesmo nome. Por exemplo, para se evitar a digitação da URI “*http://www.w3.org/1999/02/22-rdf-syntax-ns#*” repetidas vezes, basta definir um *namespace* com a sintaxe “*xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"*” e, a partir de então, usar o *namespace* “*rdf*” para fazer referência à URI completa, conforme mostra a Figura 3.7.

Observe na Figura 3.7 que existem dois *namespaces*: o principal “*xmlns:rdf*” (linha 2) e um secundário – o “*xmlns:camera*” (linha 3). Este faz referência a um documento RDF externo que define conceitos chaves sobre câmeras.

O XML fornece uma forma simples para a interoperabilidade entre sistemas computacionais, pois utiliza esquemas para determinar como estão estruturado os dados [15].

<?xml version="1.1"?>	1
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:camera="http://www.mycamera.org/photo#">	2
<rdf:Description rdf:about="NikonD70">	3
<camera:hasBody rdf:resource="SLR"/>	4
<camera:hasBack rdf:resource="DigitalBack"/>	5
<camera:lensMount>Nikon F</camera:lensMount>	6
</rdf:Description>	7
<rdf:Description rdf:about="SLR">	8
<camera:soldBy>PhotoMart</camera:soldBy>	9
</rdf:Description>	10
</rdf:RDF>	11
	12

Figura 3.7: Um documento RDF [76]

No entanto, por não fornecerem uma interpretação dos dados, não são suficientes para o conceito de web semântica [25]. Geroimenko [26] apresenta um glossário esclarecedor de termos relacionados ao XML.

3.1.2 RDF e RDF *Schema*

O RDF foi desenvolvido pela W3C para descrever recursos da internet, permitindo especificação semântica de dados baseados em XML de forma padrão e intercambiável [27].

Mas, por que usar RDF em vez de XML? A resposta é que o RDF tem um mapa de informação direta e inequívoca de um modelo de dados, de forma descentralizada e com vários leitores genéricos disponíveis [50]. Isto significa dizer que quem desenvolver um aplicativo saberá exatamente quais partes são dados semânticos e quais partes são perfumaria sintática em um documento RDF.

O RDF pode ser utilizado em uma variedade de áreas, como o exemplo dado por Lassila and Swick [39]: descoberta de recursos que ofereçam a melhor busca, catalogação, compartilhamento de conhecimento, intercâmbio de informação por meio de agentes inteligentes, classificação de conteúdos, restrições de páginas, propriedade intelectual, comércio eletrônico, etc. Não tem mecanismos para definir as relações entre propriedades e recursos, esta função é do RDF *Schema* (RDFS) [27, 15].

O RDF, em conjunto com o RDFS, oferece uma linguagem simples para representar ontologias e bases de conhecimento na internet [76]. Enquanto o RDF é usado para definir a base de conhecimento o RDFS é usado para definir a ontologia ou vocabulário sobre aquela base de conhecimento [15].

As declarações em RDF podem ser agrupadas em conjunto para a produção de redes maiores – muitas vezes referidas como uma rede semântica –, conforme ilustra a Figura 3.8 [76]. A estrutura da rede é um grafo direcionado, onde as bordas do gráfico são direcionadas e marcadas com os respectivos predicados. Para obter a documentação de referência completa sobre RDF, consulte Lassila and Swick [39].

Uma base de conhecimento RDF é composta por uma coleção de documentos RDF que são correlacionados pelos *namespaces*. As definições destes documentos não dizem o

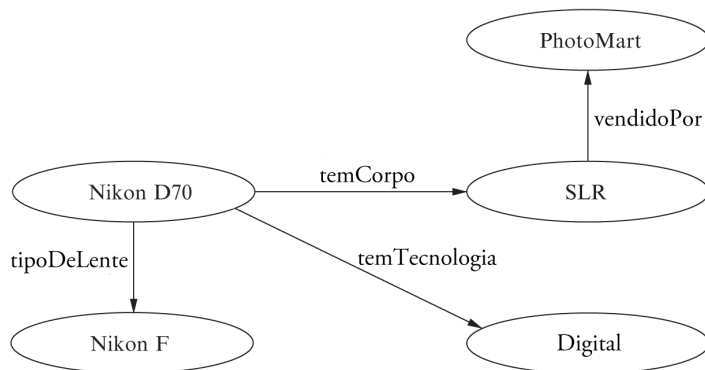


Figura 3.8: Uma rede semântica RDF (adaptada) [76]

porquê dos predicados “*hasType*” ou “*subTypeOf*” nas declarações RDF, pois esta definição é responsabilidade de uma base de conhecimento externa [76].

Deve-se dizer que o RDFS é mais expressivo que o RDF, pois traz relações entre subclasse e superclasse bem como opção para impor regras entre as declarações, especificar classes, relacionamentos e definir propriedades [5, 25]. O RDFS utiliza primitivas⁵ de modelagem baseadas em quadros – veja a Seção 8.3.4.6. Uma falha do RDFS é o não suporte à equivalência de classes e a impossibilidade de uso das restrições de cardinalidade [25].

Segundo Walton [76], o RDFS é uma linguagem de descrição ontológica simples, que define a estrutura do conhecimento de um documento RDF. A função primordial do RDFS é fornecer meios para que definição de um vocabulário de termos possa ser definida, para uso posterior na construção de declarações RDF.

A modelagem do conhecimento em RDFS é semelhante a um projeto de *software* com orientação a objetos, o que torna a linguagem de fácil compreensão para que entenda este estilo de definição [76]. Falta no RDFS muitos recursos desejados para uma linguagem completa para a ontologia, o que foi fornecido pela OWL [15]. Mas se o RDFS não é completo para a proposta de representar conhecimento, por que deve ser estudado? Porque a OWL é uma extensão do RDFS, e este ainda é útil para a tomada de certos tipos de inferências.

O objetivo maior do RDFS é identificar os vários tipos de relações que se deseja representar em uma base de conhecimento, em torno de uma noção de hierarquia de classes. Uma hierarquia de classes, conforme mostra a Figura 3.9, estabelece uma relação entre diferentes classes, sendo que o RDFS permite somente a utilização do conceito de subclasses [76].

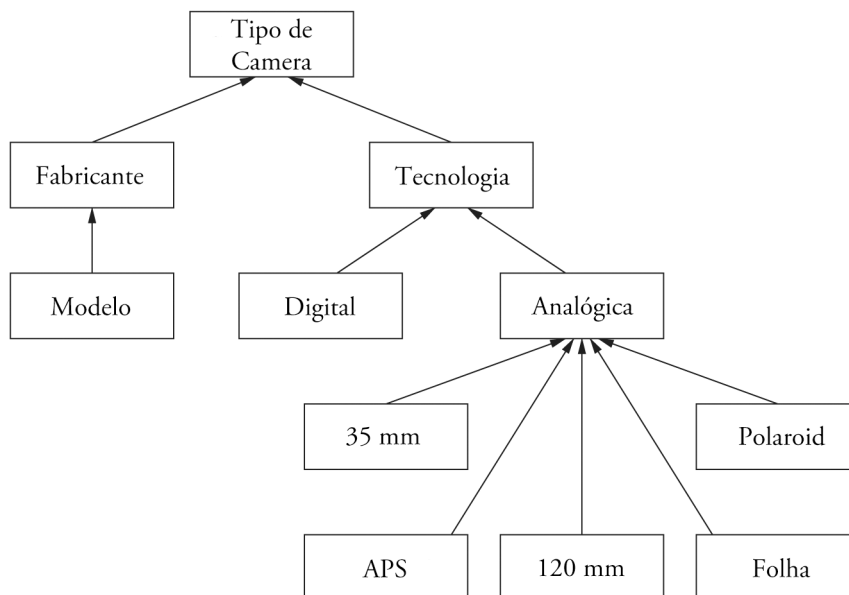


Figura 3.9: Exemplo de hierarquia de classes (adaptada) [76]

⁵Algumas primitivas são *Class*, *subClassOf*, *Property* e *subPropertyOf*.

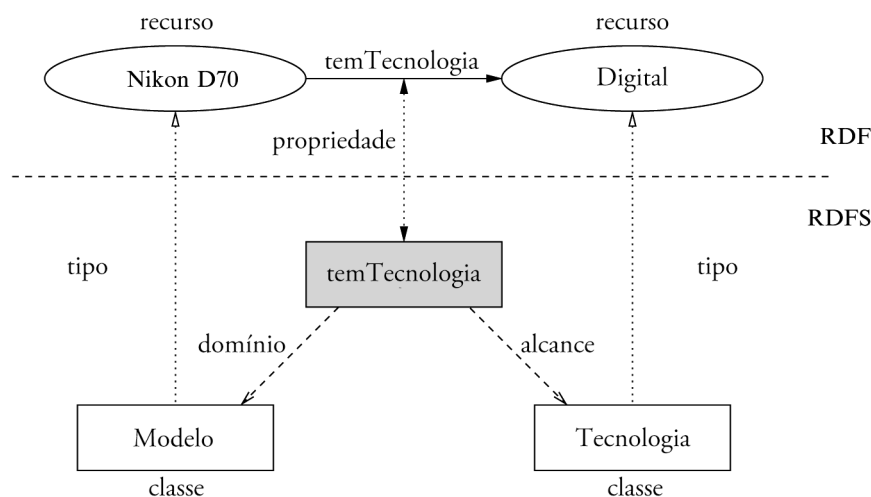


Figura 3.10: Exemplo de restrições RDFS em um RDF (adaptada) [76]

Walton [76] ensina ainda que uma outra função muito apreciada do RDFS é a possibilidade de restrições sobre o que se pode afirmar sobre determinado documento. Por exemplo, na Figura 3.9, deseja-se limitar declarações inválidas do tipo “A Nikon D70 é fabricada pela 35 mm”. Para resolver este problema, aplica-se restrições entre o domínio e o escopo da propriedade, conforme ilustra a Figura 3.10.

Para finalizar, deve-se dizer que a diferença mais importante entre o conceito orientação de objetos para o RDFS em relação ao mesmo conceito para as linguagens de programação é que nesta, quando uma propriedade é definida esta deve ser respeitada, sob pena de ter o código produzido invalidado em tempo de execução [15]. Já no RDFS a interpretação do esquema é decidido por um agente externo, que pode ou não solicitar a validação do esquema [76].

O modelo de dados do RDF consiste em três tipo de objetos [27]: os recursos que são descritos por expressões e sempre nomeados por um *Uniform Resource Identifier* (URI); as propriedades que define os aspectos, características, atributos ou relações na descrição de um recurso e; as declarações que atribuem valores para uma propriedade em um recurso específico.

3.1.3 DAML+OIL

A DAML+OIL foi desenvolvida em conjunto pelos Estados Unidos e Europa [27]. Esta evoluiu de outras duas línguas, DAML e OIL [47, 24]. Na web semântica a *DARPA Agent Markup Language* (DAML) teve como objetivo inicial dar suporte ao desenvolvimento da web semântica, provendo meios para que o significado do conteúdo das páginas fosse entendido pelas máquinas, ao invés de sua mera exibição ao usuário final [47]. Já a *Ontology Inference Layer* (OIL) foi desenvolvida para prover uma infraestrutura ontológica de apoio à web semântica, tendo forte base na lógica descritiva sistemas baseados em quadros. A *Web Ontology Language* (OWL) incorporou grande parte dos esforços feitos pela DAML+OIL [25].

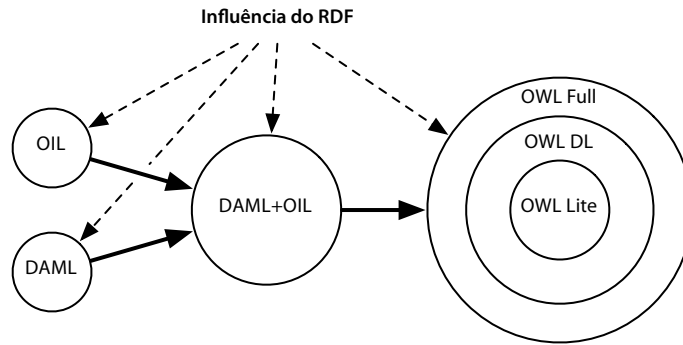


Figura 3.11: As origens da OWL (adaptada) [25])

A DAML+OIL foi um esforço para a criação de um linguagem universal da web semântica, de forma que máquinas conseguissem interpretar o conteúdo das páginas e inferir conclusões a partir disto. Para tanto utilizou regras e definições semelhantes à do RDFS, incluindo a possibilidade de aplicar restrições e relacionamentos adicionais, inclusive utilizando cardinalidade, restrições por intervalos, união, disjunção, inverso e regras transitivas [25, 66].

A DAML+OIL é dividida em duas partes distintas: o *datatype domain* – valores que pertencem aos tipos de dados do XML *Schema* – e o *object domain* – objetos individuais, considerados membros das classes descritas na DAML+OIL ou RDF [25]. Da mesma forma, existem dois tipos de propriedades básicas na DAML+OIL: aquelas que relacionam um objeto a outro objeto, especificada pelo código “*daml:ObjectProperty*” e aqueles que relaciona um objeto a valores de dados pelo código “*daml:DataTypeProperty*”. A sintaxe para as classes e propriedades são semelhantes à do DAML e OIL, já as instâncias das classes e propriedades têm sintaxe semelhantes à RDFS.

3.1.4 OWL

A *Ontology Web Language* (OWL) foi a sucessora da linguagem DAML+OIL. O vocabulário da OWL herdou da DAML+OIL o conjunto de elementos e atributos XML, com significados bem definidos, utilizados para descrever termos do domínio e seus relacionamentos em uma ontologia. Segundo Wooldrigdge [78], esta é provavelmente a mais importante e influente linguagem de ontologia dos dias atuais. A Figura 3.11 mostra as divisões e influências da linguagem OWL.

A OWL tem sintaxe semelhante ao RDFS, que por sua vez é similar ao RDF, já que todos têm base no XML. A OWL depende de uma base de conhecimento construída em RDF, e o OWL pode ser usado como um substituto ou complemento ao RDFS [76].

O destaque da OWL é a riqueza desta para descrever relações, propriedades e individualidades das classes [19]. Para se ter uma ideia da flexibilidade da linguagem, pode-se especificar em OWL que uma propriedade é simétrica (*Symmetric*) ao inverso (*InverseOf*) do equivalente (*EquivalentProperty*) de outra propriedade que já era transitiva (*Transi-*

tive), pode-se definir cardinalidades para as propriedades (*MinCardinality* e/ou *MaxCardinality*), definições de novas classes por interseções (*IntersectionOf*), uniões (*UnionOf*), complementos (*ComplementOf*) de classes já existentes, utilização do ou exclusivo (*disjoint*) para evitar que um objeto possa ser duas coisas distintas ao mesmo tempo, dentre vários outros [25, 76].

Nas linguagens ontológicas disponíveis para uso na web semântica existe uma relação de proporcionalidade inversa entre o nível de expressividade e o raciocínio eficiente [76, 78]. Por este motivo a OWL foi dividida em três sublinguagens: *OWL-Lite*, *OWL-DL*⁶ e *OWL-Full* – conforme indicado pela Figura 3.11. Estas foram concebidas para satisfazer a diferentes tipo de necessidades e sistemas, sendo a *OWL-Full* mais expressiva que a *OWL-DL* e esta mais expressiva que a *OWL-Lite*. Entende-se por expressividade o número de primitivas que podem ser usadas em cada tipo de linguagem OWL e, por isso, uma linguagem produzida pela *OWL-Lite* pode ser facilmente adaptada para a *OWL-Full*, mas o contrário não é verdadeiro [78].

3.1.4.1 *OWL-Lite*

A *OWL-Lite* é uma forma restrita da OWL destinada ao fácil entendimento e de forma simplificada para o uso em aplicações, o que é obtido à custa da expressividade [76]. Permite a construção de hierarquias e restrições de classificações simples, sendo a capacidade de expressão restrita – por exemplo a cardinalidade que pode ser definida neste modo é de apenas 0 ou 1 [25]. Esta forma da OWL implica em um número de restrições nos tipos de axiomas que se pode usar, o que torna a linguagem mais tratável e de leitura mais compreensível por humanos [78].

3.1.4.2 *OWL-DL*

A *OWL-DL*, onde o DL significa *description-logic* (lógica descritiva), reflete a base lógica de descrição de seu antecessor, o DAML+OIL, fornecendo mais expressividade e garantido que todas as conclusões serão possíveis e calculáveis em tempo finito [25]. Esta inclui todas as construções da linguagem OWL, mas impõe algumas restrições ao usá-la [76]. Pode-se dizer que esta variante estende as propriedades da *OWL-Lite* permitindo, por exemplo, especificar que algo é disjunto em duas classes [78].

3.1.4.3 *OWL-Full*

A *OWL Full* foi projetada para desenvolvedores que desejam ter a máxima expressividade unida à liberdade sintática do RDF e do RDFS, apesar do contrapeso disto ser a não garantia de cálculo das conclusões em tempo finito ou mesmo a garantia de decidir sobre determinada inferência [25, 76, 78].

⁶*OWL-Description Logics*.

Segundo Wooldrigde [78], a mesma ontologia pode ser expressada em diferentes linguagens, o que pode parecer uma ideia estranha de início. Esta estranheza é minimizada ao se pensar em uma ontologia como um algoritmo: um algoritmo existe independente da linguagem de programação. Da mesma forma é possível escrever uma ontologia independente do nível de formalismo ou linguagem adotado, desde que se use uma linguagem abstrata para se definir tal ontologia.

3.1.5 OWL vs. RDF

Embora os formatos RDF e RDFS sejam usados como padrões da web semântica, estes não são expressivos o suficiente para imprimir a representação do conhecimento em um nível suficientemente explícito para as máquinas [5]. Por nível explícito entende-se que conclusões possam ser tomadas a partir do conhecimento, em sua forma legível por máquinas. Neste sentido, uma das línguas baseadas em XML para a representação do conhecimento, de forma totalmente inteligível pelas máquinas, é a *Ontology Web Language* (OWL). A OWL tem base na lógica descritiva⁷, um dos vários formalismos disponíveis para a representação do conhecimento [5]. A OWL tem como característica básica o uso de vários construtores para gerar classes complexas a partir de simples declarações, enfatizando a capacidade de raciocínio sobre os problemas [59].

3.1.6 Outras linguagens

Uma visão geral das linguagens disponíveis para a manipulação de ontologias é apresentada a seguir, sendo a lista baseada na pesquisa Gómez-pérez and Corcho [27]:

- *Simple HTML Ontology Extension* (SHOE): foi desenvolvida na Universidade de Maryland e teve o objetivo de ser uma extensão do HTML, incorporando conhecimento semântico em documentos neste. O SHOE torna possível que os agentes colem informações de páginas e documentos da internet em três fases: definir uma ontologia, anotar dados nas páginas HTML e manter esta informação atualizada.
- *Ontology Markup Language* (OML): foi desenvolvida na Universidade de Washington e compartilha muitos recursos com o SHOE. É dividida em quatro níveis: *Core OML*, que está relacionado com aspectos lógicos da linguagem; *Simple OML*, que faz mapeamento direto com documentos RDF(S); *Abbreviated OML* que inclui recurso de gráficos conceituais e; *Standard OML* que a versão mais expressiva.
- *eXchange Ontology Language* (XOL): foi desenvolvida pela comunidade acadêmica de bioinformática dos Estados Unidos para conceber a troca de definições ontológicas entre um conjunto heterogêneo de aplicativos. Baseada na sintaxe XML, tomou as

⁷A lógica descritiva teve forte influência na concepção de OWL, particularmente sobre a formalização e semântica, na escolha dos construtores de línguas e na integração de tipos de dados e valores de dados.

línguas Ontolingua e OML como base, sendo a expressividade oriunda da *OKBC-Lite* – um subconjunto da *Open Knowledge Based Connectivity*.

3.2 Linguagens de consulta

A inspiração para as consultas baseadas em raciocínio veio do Sistema Gerenciador de Banco de Dados (SGBD), onde a base de conhecimento se comporta como um banco de dados de conhecimento e a ontologia a estrutura deste banco de dados [76].

Em um SGBD os dados são extraídos de um banco de dados, por meio de uma linguagem de consulta – geralmente a *Structured Query Language* (SQL) – definida em termos do esquema do banco de dados. Walton [76] apresenta uma adaptação deste conceito para o RDF, fazendo observações a seis diferenças importantes que devem ser consideradas nas consultas em web semântica:

1. A web semântica é uma rede estruturada, e não relacional. Deste lado tem-se que um SGBD trabalha com tabelas que têm relações lógicas, o que possibilita uma SQL ser desenvolvida logicamente para extrair dados de uma ou mais tabelas. Daquele lado, tem-se uma base de conhecimento RDF definindo uma rede semântica de recursos e, para que uma consulta seja feita neste de forma semelhante ao SGBD, deve-se dividir a rede semântica para que estas pequenas partes atuem como tabelas, sendo as colunas representadas por cada parte da tríplice O—V—A (consulte a Seção 8.3.4.1 na página 104).
2. A web semântica é, de forma geral, tem uma estrutura menos rígida que a de um banco de dados e isto pode causar falta de informação e inconsistências no conhecimento. No caso em que se tenha múltiplas fontes de conhecimento, uma ou mais destas fontes podem se tornar indisponíveis e o resultado disto será que a consulta, em determinados casos, pode trazer resultados apenas parciais ou serem intermináveis em tempo hábil.
3. O conhecimento na web semântica pode ser informado ou inferido. De forma geral não se faz distinção entre um conhecimento inferido ou informado na base de conhecimento, sendo o conhecimento inferido calculado somente quando uma consulta exigir. Convém lembrar que consultas baseadas em inferências podem resultar em respostas distintas se a base de conhecimento tiver alta dinamicidade ou for originada de diferentes fontes RDF.
4. O conhecimento da web semântica pode ser representado em formas sintáticas distintas, tais como RDF, RDFS ou uma ontologia OWL. Logo, as consultas devem levar em consideração que representação está sendo utilizada, sendo necessário alguns ajustes de uma representação para outra.

5. As consultas, de forma geral, não incluem especificações da base de conhecimento em que serão aplicadas. Para que as consultas sejam direcionadas para base de conhecimento com estruturas específicas, faz-se necessário a utilização de um serviço que selecione as fontes candidatas.
6. Os banco de dados trabalham com o conceito de mundo fechado e domínio finito. Já a web semântica trabalha com o conceito de mundo aberto, onde o que é falso deve ser explicitamente indicado e o domínio pode ser infinito.

3.2.1 SPARQL

O *Simple Protocol and RDF Query Language* (SPARQL) não tem o objetivo de servir conhecimento ou representar recursos, mas ser um padrão a execução de consultas em bases de conhecimentos na internet [60, 25].

O SPARQL adota a sintaxe baseada em SQL para as consultas, ao invés de sintaxe baseadas em XML – como o *XPath* ou *XQuery* [17]. Este nasceu com a proposta de substituir as várias linguagens de consultas existentes – RQL, RDQL, RDF *Query*, etc. – por um único padrão [76].

Walton [76] demonstra de forma direta e didática como o SPARQL pode ser usado:

1. Para a demonstração deve-se ter uma base de conhecimento em RDF, que está listada a seguir. A base de conhecimento contém informações sobre três modelos de câmeras e está usando uma notação alternativa (N3) por esta ser mais compacta que a representação baseada em XML, não cabendo explicações maiores sobre esta.

```
@prefix xsd:      <http://www.w3.org/2001/XMLSchema#>      1
@prefix camera:   <http://www.mycamera.org/photo#>          2
@prefix canon:    <http://www.mycamera.org/canon#>           3
@prefix nikon:    <http://www.mycamera.org/nikon#>           4
@prefix pentax:   <http://www.mycamera.org/pentax#>          5
canon:CanonEOS300D camera:manufacturer canon:CanonInc .      6
canon:CanonEOS300D camera:hasBack      camera:DigitalBack .  7
canon:CanonEOS300D camera:resolution   "6.3"^^xsd:decimal .  8
nikon:NikonD70    camera:manufacturer nikon:NikonCorporation . 9
nikon:NikonD70    camera:hasBack      camera:DigitalBack .    10
nikon:NikonD70    camera:resolution   "6.1"^^xsd:decimal .    11
pentax:PentaxK1000 camera:manufacturer pentax:PentaxCorporation . 12
pentax:PentaxK1000 camera:hasBack      camera:35mm .          13
```

2. Com a base de dados de referência, deseja-se definir uma consulta que selecione as câmeras existentes que tenha um “*digital back*”. Isto pode ser feito pela consulta em SPARQL:

```
SELECT ?x      1
WHERE { ?x <http://www.mycamera.org/photo#hasBack>    2
      <http://www.mycamera.org/photo#DigitalBack> }    3
```

Observe que tríplice O—A—V está contida na cláusula *where* da consulta (linha 2), sendo que a variável que está na primeira posição poderia estar em qualquer outra. Observe também que as URIs são delimitadas por “< >” e as variáveis iniciam com “?”. A saída da consulta anterior seria

```
?x = canon:CanonEOS300D
      nikon:NikonD70
```

3. A consulta anterior pode ser otimizada com a utilização de *namespace*. O mesmo resultado poderia ser obtido com o código SPARQL:

```
PREFIX camera: <http://www.mycamera.org/photo#>
SELECT ?x
WHERE { ?x camera:hasBack camera:DigitalBack }
```

4. A SPARQL pode operar sobre uma base de conhecimento real ou virtual. A consulta a seguir é executada sobre uma base de conhecimento virtual, construída a partir de duas fontes independentes de RDF:

```
SELECT *
FROM <http://www.myphotomart.co.uk/cameras.rss>
FROM <http://www.myphotomart.com/cameras.rss>
WHERE { ?name ?relation ?value }
```

5. A SPARQL também prevê a consulta de objetos com tríplices múltiplas, conforme visto na Seção 8.3.4.1:

```
PREFIX camera: <http://www.mycamera.org/photo#>
PREFIX nikon: <http://www.mycamera.org/ikon#>
SELECT ?x
WHERE { ?x camera:hasBack camera:DigitalBack .
        ?x camera:manufacturer nikon:NikonCorporation }
```

Os dois padrões na cláusula *where* são interpretados de forma conjuntiva.

6. Os comandos SQL *order by*, *limit*, *distinct*, *optional*, *union*, *quantifiers*, etc., podem ser usados normalmente em consultas SPARQL.

```
SELECT ?model1 WHERE { ?model1 ?x ?y } ORDER BY ?model1
SELECT ?model2 WHERE { ?model2 ?x ?y } LIMIT 4
SELECT DISTINCT ?model3 WHERE { ?model3 ?x ?y }
```

O SPARQL também pode ser usado para realizar consultas em estrutura de ontologias OWL, de forma semelhante às consultas sobre RDF [60]. No entanto, esta linguagem de consulta está baseada em padrões coincidentes. A produção de inferências mais avançadas depende de outras tecnologias, como a lógica descritiva [76] – veja a Seção 8.3.5.

3.3 Agentes

Ao se falar de agentes se faz necessário falar um pouco sobre inteligência artificial, que é o objetivo de alguns sistemas que fazem uso de agentes: simular a inteligência [70].

A inteligência artificial (IA) é uma ciência recente, nomeada em 1956 após a Segunda Guerra Mundial. Esta – que ainda fornece espaço para o surgimento de vários “Einsteins” – não tenta apenas compreender o mundo, mas também a construir entidades inteligentes [70]. A Figura 3.12 mostra a visão de alguns autores sobre o assunto.

Sistemas que pensam como seres humanos	Sistemas que pensam racionalmente
“O novo e interessante esforço para fazer os computadores pensarem ... <i>máquinas com mentes</i> , no sentido total e literal.” (Haugeland, 1985)	“O estudo das faculdades mentais pelo uso de modelos computacionais.” (Charniak e McDermott, 1985)
“[Automação de] atividades que associamos ao pensamento humano, atividades como a tomada de decisões, a resolução de problemas, o aprendizado ...” (Bellman, 1978)	“O estudo das computações que tornam possível perceber, raciocinar e agir.” (Winston, 1992)
Sistemas que atuam como seres humanos	Sistemas que atuam racionalmente
“A arte de criar máquinas que executam funções que exigem inteligência quando executadas por pessoas.” (Kurzweil, 1990)	“A Inteligência Computacional é o estudo do projeto de agentes inteligentes.” (Poole <i>et al.</i> , 1998)

Figura 3.12: Algumas definições de IA, organizadas em quatro categorias [56].

Pode-se dizer que um sistema é considerado racional quando faz tudo certo com os dados que tem, ainda que a racionalidade perfeita não seja algo viável para ambientes complexos [56].

Segundo Esbízaro [23], um agente é um sistema computacional que deve ser capaz de executar, autonomamente, ações de forma flexível em um determinado ambiente; um agente inteligente deve ter três características: ser reativo, ser pró-ativo e ser social.

Segundo Russel and Norvig [56], um agente é algo que age, mas em sistema computacional espera-se que este opere sob controle autônomo, perceba o ambiente, persista, adapte-se as mudanças e assuma metas de outros. Ainda segundo este autor, um agente racional é aquele que deseja alcançar o melhor resultado, mas em alguns casos a realização de inferências corretas não é possível devido a uma conclusão equivocada; existe também casos em que a inferência correta não apresenta toda a racionalidade, já que esta pode resultar na falta de ação ainda que correta. Um projeto de inteligência artificial com um agente racional tem, pelo menos, duas vantagens: é mais geral que as leis dos pensamentos e é mais acessível ao desenvolvimento científico, pois o padrão de racionalidade é claro e geral [56].

Um agente é aquele que percebe o ambiente por meio de sensores e age neste por meio de atuadores [56, 70]. Se por um lado um agente humano tem os olhos e os ouvidos como sensores e as mãos, pernas, boca agem como atuadores; por outro um agente robótico pode ter câmeras sonares como sensores e motores como atuadores. No caso de agente agente de *software* os sensores deste pode ser o teclado, arquivos como sensores e resultados na tela, impressoras como atuadores. Convém lembrar que a ação de um determinado

agente, em um instante qualquer, pode ser dependente do histórico de percepções, do início até o momento presente [56].

Em termos numéricos, diz-se que o comportamento do agente deve ser descrito por uma função matemática (função do agente) que ligue qualquer sequência de entrada em percepções, para assim gerar uma ação. Tabular a função do agente, de forma a prever todas as condições possíveis, pode resultar em uma tabela grande demais para a memória, que é finita. Pode-se construir, para modelagem, uma tabela com o objetivo desvelar todas as possíveis sequências de percepções, e para cada uma destas uma ação associada. Lembre-se, enquanto a função de agente é uma descrição matemática abstrata o programa do agente é uma implementação concreta, implementada de acordo com a arquitetura do agente. Para enfatizar isto basta lembrar do “mundo do aspirador de pó”, que com dois locais possíveis A e B [56]. O agente aspirador deve perceber onde existe sujeira, limpar e mover-se entre os lugares disponíveis. Se não houver sujeira, não faça nada. A Figura 3.13 ilustra parte desta solução, que apesar de parecer simples, não é se for tomada em todas a complexidade necessária para a solução completa do sistema.

Sequência de percepção	Ação
$[A, \textit{Limpo}]$	Direita
$[A, \textit{Sujo}]$	Limpe
$[B, \textit{Limpo}]$	Esquerda
$[B, \textit{Sujo}]$	Limpe
$[A, \textit{Limpo}] [A, \textit{Limpo}]$	Direita
$[A, \textit{Limpo}] [A, \textit{Sujo}]$	Limpe

Figura 3.13: Figura parcial do agente de limpeza “Aspirador de pó” (adaptada) [56])

3.3.1 Racionalidade

A racionalidade depende de quatro coisas: a medida de performance ligada aos casos de sucesso; o conhecimento prévio sobre o ambiente por parte do agente; as ações possíveis de serem realizadas pelo agente e a sequência de percepções até o momento atual [56]. Logo, um agente racional deve, para cada sequência de percepções, selecionar uma ação que é esperada – e que maximize a medida de performance – dada a evidência das percepções, desejando-se que o conhecimento nato, ou à *priori*, seja eliminado.

Para o agente aspirador de pó, este só poderia ser considerado racional caso este tentasse maximizar a performance. As ações esquerda e direita prende o agente nos ambientes possíveis, sendo que o agente percebe onde está e se existe sujeira. Um agente irracional poderia ficar se movendo desnecessariamente com as salas limpas, caso a medida de performance não leve em consideração penalidades para cada movimento[56].

3.3.2 Onisciência

Um agente onisciente conhece o resultado das ações e pode adaptar o comportamento de acordo com estes resultados, mas isto é algo impossível na realidade [56]. Um exemplo: decide-se atravessar uma rua sem tráfego; no entanto, ao se proceder com a travessia, um avião cai. Mas atravessar a rua sem olhar para o céu não é uma ação irracional, logo, a racionalidade maximiza a performance esperada; já a perfeição maximiza a performance atual [56]. A definição de racionalidade não requer onisciência, pois aquela depende somente da sequência de percepções até o momento. Uma parte importante da racionalidade é a obtenção de informações, por meio da exploração, para o ajuste de percepções futuras. Claro que somente obter informações soltas não é suficiente, mas deve-se aprender a partir das informações/percepções. Em alguns casos o ambiente pode ser conhecido de antemão e, por isso, o agente não precisaria perceber ou aprender: apenas agir de forma correta [56].

3.4 Serviços da web semântica

Enquanto a ontologia – que será apresentada no Capítulo 4 – é utilizada para representar o conhecimento, deve existir um algo para raciocinar sobre aquele conhecimento. Os Serviços da web semântica, que geralmente são encapsulados nos *Web Services*, são o meio pelo qual as inferências são realizadas sobre a web semântica [76].

Um serviço é um *software* que está disponível em algum meio e pode ser usado por outros clientes para a realização de alguma tarefa, em ambiente seguro ou não. Quando se fala em serviço, sabe-se que existem dois lados: o lado que serve o conhecimento e o lado que obtém e transforma o conhecimento. Aqueles estão relacionados à ontologia; estes aos agentes de conhecimento.

O *Web Service* é uma técnica de programação que padroniza vários aspectos de processamento distribuído e comunicação pela internet [76]. A vantagem disto são os padrões de interoperabilidade das informações, por ser uma arquitetura simples e flexível [30].

As características básicas desejadas em *Web Services* incluem: sistema autônomo, independente de plataforma, publicável, acessível, orientado a serviço, organizado e programado por meio do padrão XML [25]. As principais tecnologias que suportam os *Web Services* são: *Web Services Description Language* (WSDL), *Web Services Flow Language* (WSFL), *Universal Description, Discovery and Integration* (UDDI) e o *Simple Object Access Protocol* (SOAP) [76, 25].

O uso da arquitetura dos *Web Services* foi um passo importante para a construção de sistemas processados via internet, já que esta arquitetura permite que páginas sejam acessadas por meio de mecanismos padrões, obtendo-se como retorno informações adequadas ao processamento mecanizado, e não uma página HTML que provê resultados para o entendimento humano [76, 30].

De acordo com [13] o comitê da *Semantic Web Services Architecture* (SWSA), formado

em fevereiro de 2003, criou um conjunto de abstrações de arquitetura e protocolos como base para utilização com a web semântica. Algumas fases das interações de serviços do SWSA são destacadas por [13]:

- *Candidate service discovery* (busca do serviço candidato) é a busca distribuída para os serviços disponíveis que podem realizar um conjunto de metas ou objetivos.
- *Service engagement* (contratação do serviço) é o processo de aprovação – parcial o total, de acordo com a autodescrição do serviço – dos candidatos encontrados e a negociação de serviços futuros, sendo esta fase terminada quando o cliente e o serviço concordarem com os termos negociados.
- *Service enactment* (aprovação do serviço) é o processo que efetiva a aprovação da negociação entre cliente e serviço, seguindo regras definidas pelos protocolos do serviço. Este processo pode usar um outro protocolo para restaurar o estado operacional do processo, caso os objetivos primários violem as regras do protocolo.
- *Service discovery* (busca do serviço) é o processo pelo qual o solicitante do serviço identifica os serviços candidatos para atingir determinado objetivo. Este processo envolve três partes interessadas:
 - *service providers* (provedores de serviço) para indicar que o serviço será executado;
 - *service requestors* (solicitadores de serviço) buscam os serviços que podem alcançar um determinado objetivo interno;
 - *matchmakers* (verificadores) recebe as descrições dos serviços disponíveis dos provedores de serviço e faz a combinação com as descrições dos solicitadores de serviço.

É importante lembrar que a eficácia dos agentes que utilizam os serviços da web semântica aumentará exponencialmente à medida em que mais conteúdos legíveis por máquinas forem disponibilizados [10].

Capítulo 4

Ontologias

Trabalhos recentes em Inteligência Artificial têm explorado a ontologia como um meio interessante para compartilhar e reutilizar conhecimento entre *softwares* distintos [28]. Mas, por que ontologias? Porque elas são blocos essenciais para a construção da infraestrutura da web semântica [25]. Com ontologias e web semântica tem-se um amplo nível de serviços especializados para transformar as informações e dados da internet em conhecimento.

O termo “ontologia” teve origem na Grécia e utilização inicial pela Filosofia, e significa palavra (*logia*, *logos*) do ser (*ontos*) [58, 25].

A palavra “Ontologia” grafada com “O” maiúsculo representa a ciência filosófica e surgiu na cultura grega pela ação construtiva de Aristóteles, que a interpretava como *prôte philosophia*. Nas palavras de Santos [58, p. 16] se tem o conceito pela visão da ciência filosófica:

Há uma ciência que estuda o Ser enquanto ser, e seus atributos essenciais, Ela não se confunde com nenhuma das outras ciências chamadas particulares, pois nenhuma delas considera o Ser em geral, enquanto ser, mas, recortando uma parte do ser, somente desta parte estudam o atributo essencial; como, por exemplo, procedem as ciências matemáticas.

Mas já que procuramos os primeiros princípios e as causas mais elevadas, é evidente que existe necessariamente alguma realidade à qual tais princípios e causas pertencem, em virtude de sua própria natureza. Se, pois, os filósofos que buscavam os seres, procurassem esses mesmos princípios, resultaria daí necessariamente que os elementos do Ser são elementos deste, não enquanto acidente, mas enquanto ser. Eis por que devemos estudar as causas primeiras do Ser enquanto ser.

Segundo Santos [58], a compreensão da Ontologia na Filosofia foi modificada por filósofos modernos. Para estes conhecemos os fenômenos e sabemos da existência do *noumeno*¹,

¹Termo problemático que foi introduzido para se referir a um objeto fenomenal não pertencente a uma intuição sensível, mas a uma intuição intelectual.

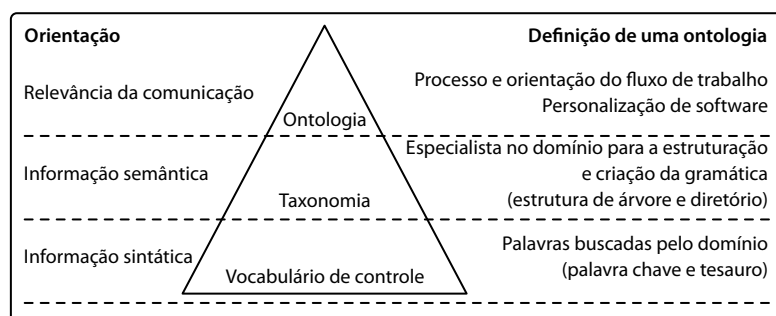


Figura 4.1: Processo de criação de uma ontologia (adaptada) [22])

mas deste não temos nenhuma experiência sensível; apenas dialética – chamada de transcendental por Kant. Desta forma, a Ontologia então seria a *ciência do noumeno*: a ela caberia o papel de estudar o que permanece atrás dos fenômenos, enquanto o estudo dos fenômenos caberiam às ciências particulares.

Na década de 80 o termo “ontologia”, agora grafado com “o” minúsculo, adquiriu significado específico em áreas que envolvem tecnologia e robótica como a Ciência da Computação na área de inteligência artificial (IA).

Um sistema que pretenda comparar ou combinar informações na internet precisa de um meio para saber que dois termos distintos querem dizer a mesma coisa, problema que é solucionado pela web semântica com o uso de ontologia [10].

Mas, por que ontologia? A ontologia, conforme pode ser observado no processo de criação ilustrado na Figura 4.1, não é apenas um vocabulário, uma representação pragmática ou uma mera taxonomia; é a estrutura básica em torno da qual o conhecimento pode ser construído, já que esta aborda conceituações de domínio livre de exigências técnicas [22]. Segundo Wooldrigdge [78], deve-se usar ontologias para adicionar informações nas páginas da internet de forma que seja possível que máquinas consigam extrair e processar o conteúdo destas. Hendler [33] afirma que a internet atual, com apoio da web semântica, é uma poderosa ferramenta para aqueles que sabem transformar o conhecimento acumulado em conhecimento compartilhado por meio de ontologias.

4.1 Por que desenvolver uma ontologia?

Segundo Noy and McGuinness [47], o desenvolvimento de ontologias – uma especificação explícita e formal de termos em um determinado domínio, com suas relações Gruber [28] – nos últimos anos deixou de ser exclusividade de laboratórios de inteligência artificial, podendo ser utilizada nos computadores de especialistas em domínios. Na Internet, as ontologias são utilizadas como grandes taxonomias para a classificação de alguns sítios, como a Amazon². Atualmente, em muitas áreas de conhecimento, é possível que um especialista em determinado domínio padronize e compartilhe o conhecimento – com a

²<http://www.amazon.com>

possibilidade de incluir as anotações explicativas de cada entrada – na forma ontológica [18].

A medicina, por exemplo, teve uma larga base de conhecimento ontológica especializada produzida, a qual é denominada SNOMED [54] e tem vocabulário estruturado, além de seguir os padrões sugeridos pela *World Wide Web Consortium* (W3C³) [47].

Já na área de ontologias genéricas, o que vai ao encontro do objetivo desta pesquisa, tem-se como exemplo a ontologia produzida pela *United Nations Development Program and Dun & Bradstreet*, que criou a ontologia UNSPSC (já tratada na seção 2.5.1), que propõe uma terminologia básica universal para produtos e serviços no comércio eletrônico [47].

Noy and McGuinness [47], Gruber [28], Guarino [29] complementam sobre o porquê deve-se utilizar ontologias:

- Para compartilhar um conhecimento comum da estrutura de uma informação para pessoas ou agentes de *software*
- Para possibilitar o reuso de um conhecimento sobre determinado domínio
- Para tornar explícita as suposições sobre determinado domínio
- Para separar o conhecimento do domínio do conhecimento operacional
- Para analisar o conhecimento de determinado domínio

4.2 O que é uma ontologia?

Embora existam muitas definições sobre o que é ontologia, o traço comum é que uma ontologia é uma descrição formal de um determinado domínio, com o objetivo de compartilhar informações entre diferentes aplicações [46, 18]. A seguir, a visão de alguns outros autores de renome no assunto são listadas:

- Gasevic et al. [25] afirmam que a ontologia está ligada a sua terminologia, ao vocabulário do domínio e a todos os conceitos essenciais: classificações, taxonomias, relações, hierarquias, restrições e os axiomas do domínio. Mais formalmente, para alguém que queira discutir um tópico do domínio **D** usando uma linguagem **L**, uma ontologia oferece uma lista dos tipos de coisas que existe em **D**; os tipos são representados em termos de conceitos, relações e predicados de **L**. A ontologia é uma parte muito importante do conhecimento sobre qualquer domínio; a ontologia é a parte fundamental do conhecimento, e todos os outros conhecimentos devem confiar e que fazer referência às ontologias.

³<http://www.w3c.org>

- Edgington et al. [22] dizem que a ontologia consiste em um modelo conceitual, um dicionário de sinônimos e um conjunto de atributos e axiomas expandidos. A preocupação da ontologia é com a representação adequada dos conteúdos, que poderão ser incrementados depois com o uso de linguagens de especificação. O modelo conceitual representa metadados que englobam o conjunto principal de conceitos e os respectivos relacionamentos.
- Gruber [28], o autor da definição mais citada, diz que “A ontologia é uma especificação formal de uma conceitualização compartilhada explícita”. Na curta definição, *conceitualização* significa uma visão abstrata e simplificada do mundo que se deseja representar para algum propósito. Toda *conceitualização* é baseada em conceitos, objetos e outras entidades que se presume existir nas áreas de interesse – juntamente com suas relações. Pragmaticamente, uma ontologia comum define o vocabulário com que as pesquisas e afirmações são trocadas entre agentes por meio de um compromisso ontológico, ou seja, uma forma compartilhada, coerente e consistente de uso do vocabulário por todos os agentes envolvidos. Os agentes que partilham determinado vocabulário não necessariamente compartilham suas bases de dados, o que implica dizer que um agente não é obrigado a responder a todas as consultas com base naquele vocabulário.
- Guarino [29] define que a ontologia pode ser vista como o estudo da organização e natureza do mundo, independente da forma que o conhecemos. O autor, em sua definição, faz distinções entre as coisas do mundo – objetos, matéria, regiões, etc. – e a forma de modelar o mundo – conceitos, propriedades, qualidades, estados, etc.
- Hendler [33] define a ontologia como um conjunto de termos e conhecimentos, que têm vocabulários, interconexões semânticas e de regras lógicas para a realização de inferências.
- Kalfoglou and Schorlemmer [36] dizem que uma ontologia é uma representação explícita de um entendimento compartilhado dos conceitos importantes entre alguns domínios de interesse. Explodindo a definição, tem-se que a palavra compartilhado indica que a ontologia consegue exprimir um conhecimento consensual sobre os objetos relacionados, um conhecimento aceito por grande parte da sociedade, já que todo conhecimento individual é sobremaneira subjetivo.
- Noy and McGuinness [47] defendem que existem várias definições sobre o que é ontologia, inclusive muita destas se contradizem. Na visão dos autores, uma ontologia é uma descrição formal e explícita dos conceitos presentes no em um determinado domínio e é composta por: conceitos, também conhecidos por classes; *slots*, também conhecidas por propriedades ou regras, que descrevem os vários atributos e características de cada conceito; *facets*, também conhecidas por restrições, aplicadas aos

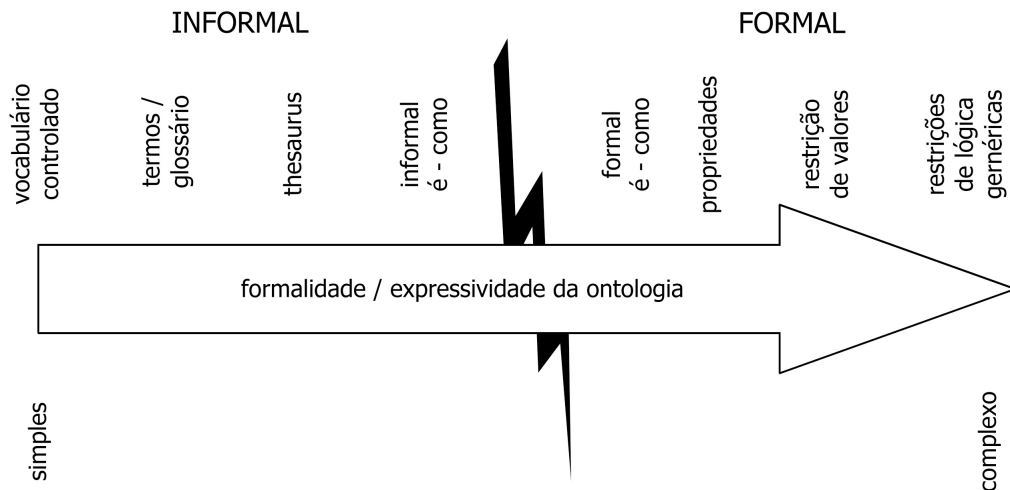


Figura 4.2: Da ontologia informal, menos expressiva até a formal, mais expressiva (adaptada) [78].

slots. Uma ontologia com um conjunto de instâncias individual de cada classe constituem a base de conhecimento. Existe uma linha tênue que define quando termina a ontologia e se inicia a base de dados.

A depender do uso que se deseja dar a um determinado conhecimento, as ontologias existem em uma variedade de tipos, variando na formalidade e especificidade [78]. A Figura 4.2 mostra uma possível separação entre as ontologias, das informais – menos expressivas – até as formais – mais expressivas. Wooldrigdge [78] define cada um dos conceitos vistos na Figura 4.2:

- Vocabulário controlado: talvez o modelo mais simples de ontologia. Faz-se uso de palavras chaves pré-definidas para classificar os domínios: sem hierarquias, propriedades ou axiomas. Em alguns domínios isto é suficiente.
- Termos / glossário: tem-se uma lista de termos, assim como no vocabulário controlado, mas existe a necessidade de se dar sentido aos termos. Este sentido é dado pelo desenvolvedor da ontologia.
- *Thesaurus*: um *thesaurus* define uma lista de sinônimos para determinada lista de termos.
- “Parte de” informal: trata-se de um vocabulário controlado, organizado em uma hierarquia informal.
- “Parte de” formal: trata-se de uma hierarquia formal, com as relações entre classes definidas e explicitadas.
- Propriedades: as classes podem ter propriedades, de forma que as relações entre classes e propriedades permitam que sejam inferidas conclusões a partir destas.

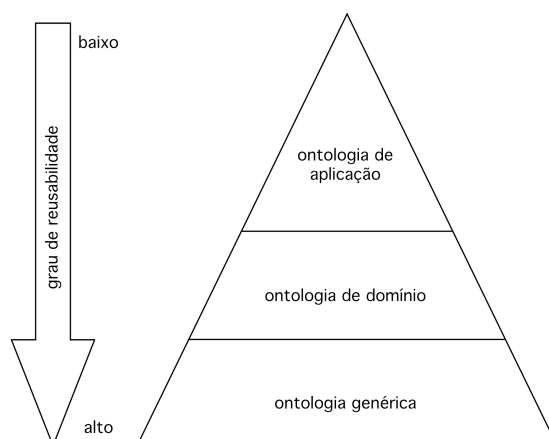


Figura 4.3: Hierarquia das ontologias: do mais genérico ao mais específico (adaptada) [78].

- Restrições de valores: fornece uma informação adicional sobre as relações, como por exemplo “toda pessoa tem apenas uma mãe”.
- Restrições lógicas arbitrárias: por último, pode-se ter ontologias com restrições lógicas arbitrárias, tais como restrições de valores, restrições na hierarquia taxonomica, etc. Quanto mais complexa forem tais restrições, maior será a probabilidade de indecibilidade do sistema computacional para inferir determinada informação.

Assim como se define ontologias baseadas no nível de formalidade e tipos de informação, também pode-se definir ontologias com base no uso e aplicação desta [78]. Na Figura 4.3 tem-se uma “pirâmide ontológica” onde na base – *upper ontology* – tem-se as ontologias mais genéricas, a serem especializadas pelas camadas superiores; no meio – *domain ontology* – tem-se as ontologias que definem conceitos apropriados para aplicações específicas em determinados domínios; no topo – *application ontology* – tem-se a ontologia utilizada nas aplicações específicas em si, sendo utilizadas e manipuladas pelos usuários finais.

4.3 Benefícios das ontologias

Muitas organizações consideram a gestão de conhecimento como a chave para manter a vantagem competitiva, no entanto estas na maioria das vezes não têm certeza da melhor forma para definir o que é o conhecimento de interesse, como compartilhá-lo de forma rápida e relevante [22, 41]. Os benefícios das ontologias são tantos que elas estão se tornando rapidamente na espinha dorsal da web semântica [79], sendo o uso destas um elemento essencial em algumas aplicações – agente de sistemas, sistemas de gerenciamento do conhecimento e plataforma de comércio eletrônico [18, 27].

O conhecimento está disponível em linguagem natural, que a grosso modo é um conjunto de declarações ontológicas expressas naturalmente pelos seres humanos – mas este

tipo de linguagem é de difícil processamento por sistemas computacionais [29, 23]. As ontologias traduzem este conhecimento em uma forma que as máquinas consigam entender e processar. Para tanto, três níveis são necessários: no primeiro nível escolhe-se o formato da linguagem de representação, no segundo nível que protocolo será usado para a comunicação do agente e, por fim, no terceiro nível especifica-se qual conteúdo do conhecimento adquirido deverá ser compartilhado [28]. A ontologia pode tanto extrair informação da linguagem natural quanto gerar linguagem natural a partir de uma base de conhecimento [27].

Para Gruber [28], que vai ao encontro das ideias de Gasevic et al. [25], as ontologias são erroneamente equiparadas à hierarquias taxonômicas. O objetivo maior da ontologia é compartilhar o conhecimento e permitir o reuso deste por diversas aplicações, como se esta pudesse materializar o nível de conhecimento [41].

Segundo Tetlow et al. [72], nos últimos anos tem havido um forte movimento para unir as metodologias de engenharia de software – como a *Unified Modeling Language* (UML) – com as tecnologias de web semântica – como o RDF e o OWL. Embora este trabalho tenha sido em grande parte motivado por interesses em explorar a popularidade e as características da UML para a definição de vocabulários e ontologias, alguns têm defendido os benefícios desta técnica para a validação, automação e novas capacidades de engenharia de *software* para a web semântica.

Várias áreas da inteligência artificial e computação usam ontologias para: representação do conhecimento, engenharia do conhecimento, modelagem qualitativa, engenharia de linguagens de programação, projeto de banco de dados, recuperação e extração da informação, automação e a gestão e organização do conhecimento [28, 29, 32, 25]. As ontologias não úteis apenas em aplicações que o conhecimento desempenhe um papel fundamental, elas podem provocar uma mudança importante no conteúdo da internet atual [27, 41].

4.4 Estado da arte

Vários sistemas foram construídos e tiveram – em maior ou menor grau – sucesso na utilização de web semântica com ontologias. A seguir alguns destes trabalhos são apresentados de forma imparcial, não tendo a ordem qualquer relação com grau de importância, qualidade da pesquisa, ordens cronológica ou alfabética.

Alani et al. [2], na pesquisa em processamento de linguagem natural, desenvolveram o sistema *Artequakt* – um extrator automático de conhecimento sobre artistas que, a partir da internet, popula a base de dados e gera biografias personalizadas. A ferramenta faz uma varredura na rede por meio dos buscadores *Yahoo*⁴ e *AltaVista*⁵, quebra os textos em parágrafos/sentenças, analisa sentença-a-sentença de forma sintática e semântica, ex-

⁴<http://www.yahoo.com>

⁵<http://www.altavista.com>

traí o conhecimento utilizando um mecanismo de expansão léxico baseado em termos, transforma este conhecimento e o armazena em um formato legível para as máquinas.

Nguyen and Phan [45] produziram uma solução híbrida de ontologia que tem o objetivo de expandir as consultas dos usuários. O motivo da pesquisa foi o fato de que os usuários frequentemente fazem consultas simples, apenas com palavras-chaves ao invés da frase inteira, o que torna difícil a recuperação de informações sem ambiguidade e dentro do contexto esperado. O sistema otimiza uma consulta como “mesa grande vermelha” para “mesa que tem a cor do tampo vermelha para seis cadeiras”.

Stecher et al. [65] propõem em a reutilização de ontologias de forma adaptativa, no qual o algoritmo procura descobrir qual a parte central de uma ontologia e se esta pode ser utilizada como subontologia. Segundo os autores isto pode reduzir o custo de criação de ontologias, melhorar a qualidade das ontologias resultantes e facilitar a interação entre diferentes sistemas. A abordagem utilizada combina métodos modulares, léxicos, linguísticos, estruturais e lógicos para encontrar a correspondência entre ontologias, tendo em vista a perspectiva de modelagem pretendida.

Kinsella et al. [37] propuseram um sistema simples, mas de grande utilidade para quem deseja analisar RDF e OWL da internet. O sistema permite visualizar os *namespaces*, classes e ligações entre base de conhecimentos. Os usuários podem ver em detalhes um domínio de interesse, analisar o uso de determinado recurso ou como uma classe está sendo usada pelos vocabulários relacionados.

Pernas and Dantas [52] apresentam a utilização de ontologias para descrever recursos em ambientes *grid*. Segundo os autores, os ambientes de computação em *grid* podem realizar o compartilhamento em larga escala de recursos e serviços, sendo soluções eficientes, com altos desempenho e disponibilidade. Neste tipo de arquitetura duas entidades têm destaque: o servidor, que estabelece as regra dos negócio, e os clientes que usam os serviços disponíveis. O estudo faz propostas para distribuir o processamento de ontologias complexas, onde a disponibilidade das fontes e balanceamento de carga sejam cruciais para o correto funcionamento do sistema.

Su et al. [68] pesquisaram sobre como melhorar a interoperabilidade semântica entre agentes de sistema, utilizando uma ontologia compartilhada entre os agentes. O resultado da pesquisa é verificada com efeito e um tópico que tem referência direta com a proposta desta dissertação: a correlação do código UNSPSC com o código *e-Class*. Ressalta-se que os autores minimizaram o problema ao assumir que os mapeamentos já existem e são mantidos por servidores externos.

Azevedo [4] desenvolveu o sistema CoreSec, uma ontologia para o domínio da segurança da informação que define conceitos genéricos e abstratos, compartilhando esta informação entre todos os envolvidos como forma de auxílio na tomada de decisões diante de um incidente. Outro produto gerado neste trabalho foi o *framework* CoreEditor, que tem o objetivo de manipular ontologias para auxiliar a Governança de Tecnologia da Informação.

Beneventano and Montanari [7] pesquisaram sobre a viabilidade de uma infraestrutura dinâmica para apoiar o comércio eletrônico. Os autores analisaram os vários produtos e sistemas de classificação que estão disponíveis, tais como a UNSPSC, *e-Class* e *RosettaNet*. Em seguida desenvolveram uma teoria, produziram um mapa de correspondência em nível teórico, mas não desenvolveram qualquer aplicação que utilizasse os conceitos propostos para testes e verificação.

Bianchini and Antonellis [11] apresentam o *E-Service*, um método baseado em ontologias para a classificação e busca de serviços eletrônicos, por meio das relações semânticas de semelhança, equivalência e generalização estabelecidos de forma semiautomática. Cada serviço é caracterizado por meio de um conjunto de atributos que são comparados com um conjunto de pré e pós condições associadas a outros serviços.

Tartir and Arpinar [71] apresentaram uma versão otimizada do OntoQA, sistema para classificar ontologias populadas usando um rico conjunto de métricas e a relação entre as palavras-chaves da pesquisa. Segundo os autores, a solução proposta é diferente de outras metodologias no sentido de que é personalizável, requer o mínimo de interação com o usuário e leva em consideração tanto os esquemas quando as instâncias de ontologias populadas.

Esbízaro [23] propõe o desenvolvimento de uma ontologia de rastreo em *log* de eventos, de forma a facilitar a recuperação de informações e apoiar no entendimento e resolução dos fatos apontados nestes *logs*.

Uma visão de uma arquitetura genérica de um sistema para o uso da ontologia como gerenciador de conhecimento pode ser visto na Figura 4.4.

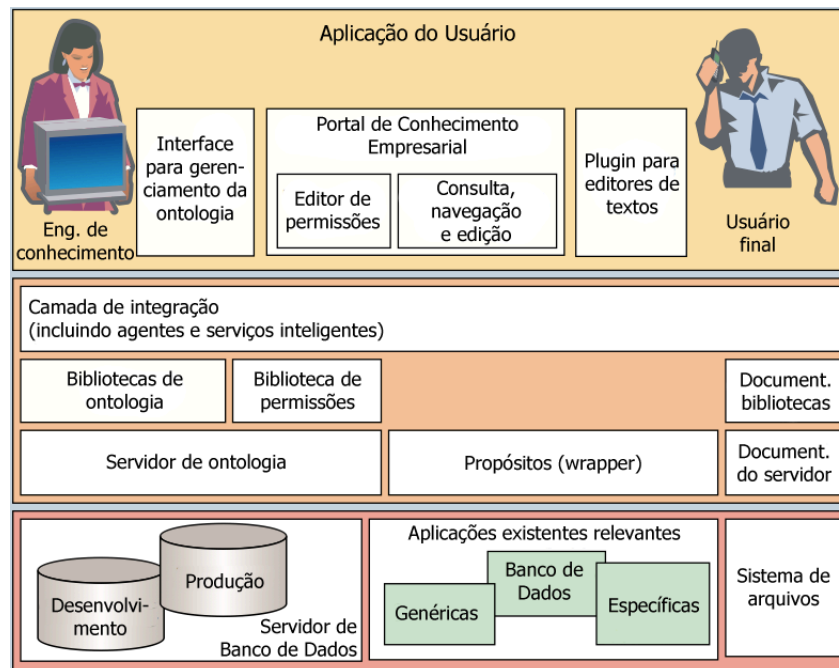


Figura 4.4: Arquitetura genérica de um sistema de ontologia como gerenciador de conhecimento (adaptada) [42].

4.5 Construção de ontologias

Construir uma ontologia formal é uma tarefa complexa, ainda que a ontologia desejada seja simples [76]. Para minimizar esta dificuldade várias metodologias e ferramentas foram produzidas para auxiliar no processo de construção de ontologias, já que estas devem ser dinâmicas, interligadas e podem mudar ao longo do tempo devido a dinamicidade da internet [76, 28]. No entanto, não existe uma maneira correta e única para se modelar um domínio e a melhor solução quase sempre depende do que se deseja modelar [6].

O mapeamento das ontologias que existem sobre o assunto a ser tratado é tão importante quanto a produção da ontologia em si, já que estes podem ser usados para minimizar o trabalho de criação. Kalfoglou and Schorlemmer [36] apresentam um estudo sobre o estado da arte nas tecnologias de mapeamento, onde analisa 35 ferramentas, apresentando alternativas viáveis para o mapeamento e união das ontologias já existentes. Noy [46] define as dimensões da integração semântica:

- Mapeamento: dadas duas ontologias, como encontrar semelhanças entre elas, determinar quais os conceitos e/ou propriedades representam similaridades.
- Representação formal declarativa dos mapeamentos: dada duas ontologias, como representar o mapeamento entre estas de forma que o raciocínio seja facilitado.
- Raciocínio com mapeamentos: depois de definido os mapeamentos, o que fazer com eles? Quais tipos de raciocínio estão envolvidos?

Após o mapeamento, os conhecimento deve ser classificado, de acordo com a ordem de relevância ou outro critério chave. Esta tarefa é complexa sem o auxílio de uma metodologia, e é neste sentido que as pesquisas de Yu et al. [79] e Alipanah et al. [3] podem ser de grande relevância para quem quer produzir uma ontologia. Uma nova abordagem para a classificação e *ranking* de ontologias na web semântica, onde cada ontologia candidata é analisada e classificada no que diz respeito à estrutura e à semântica, sem referência ao desempenho, foi proposta por Yu et al. [79]. Já Alipanah et al. [3] visa melhora no desempenho com a classificação, justificando que ontologias de um mesmo domínio podem se sobrepor e que estas devem ser classificadas com base na uniformização da sobreposição entre cada par. Pode-se dizer que estes dois trabalhos teoricamente se complementam, já que este foca no desempenho enquanto aquele foca na análise e estrutura.

Com as ontologias existentes mapeadas e classificadas, é hora de analisar o que já foi feito e o que está por fazer. Neste momento as decisões que nortearão a criação da ontologia devem ser tomadas com critérios objetivos, fundados sobre a finalidade do resultado esperado [28]. Além disso, deve-se observar a correta elaboração dos respectivos vocabulários, relações, restrições e axiomas [22], em um processo que é necessariamente iterativo [6].

Alguns pontos chaves e critérios de desenvolvimento que norteiam o que realmente se deseja, bem como a forma mais eficaz de se obter os resultados esperados, são apresentados:

- Pontos chaves da ontologia [25]
 - **Colaboração:** um sistema baseado em ontologia pode unificar todo o conhecimento para ser usado de forma comum, de forma compartilhada para futuras decisões.
 - **Interoperabilidade:** a ontologia permite a integração de informações das mais diferentes origens.
 - **Educação:** especialistas em determinadas áreas podem compartilhar os seus entendimentos sobre determinado domínio.
 - **Modelagem:** a ontologia representa blocos de construções reutilizáveis, por várias aplicações distintas e simultâneas.
- Definições sobre as partes de uma ontologia [47]:
 - **Classes:** as classes são o foco da maioria das ontologias. Estas define os conceitos de um domínio. Por exemplo, a classe vinho de representa todos os vinhos. Classes podem ter subclasses que especializam o conhecimento (ou superclasses), por exemplo a classe vinho pode ter as subclasses mutuamente disjuntas tinto, branco ou *rosé*.
 - **Slots:** os *slots* descrevem as propriedades das classes e instâncias. Por exemplo, o vinho *Château Lafite Rothschild Pauillac* está com a descrição completa; pode-se deduzir que foi produzido na videira *Château Lafite Rothschild*. Logo, alguns *slots* úteis para a descrição de um vinho são: videira, sabor, nível de açúcar, etc.
 - **Restrições:** uma explicação simples e concisa das restrições não foi apresentada por Noy and McGuinness [47], mas uma seria a restrição lógica que um vinho não pode ser tinto, branco ou *rosé* ao mesmo tempo; outra seria que os vinhos não podem ser sem açúcar e com açúcar ao mesmo tempo.
- Critério de desenvolvimento da ontologia [28]
 - **Clareza:** uma ontologia deve transmitir de forma eficaz o significado dos termos definidos, de forma objetiva e formal. Sempre que possível deve-se usar axiomas, com definição completa, dando preferência as condições necessárias *ou* suficientes – sobre as condições necessárias *e* suficientes, lembrando que todas as definições devem ser documentadas em linguagem natural.

- o **Coerência:** uma ontologia deve ser coerente, resultando em inferências que sejam consistentes com as definições, e para isto deve-se produzir axiomas logicamente consistentes. A coerência também deve ser observada nos pontos informais, tais como documentação em linguagem natural e exemplos.
- o **Extensibilidade:** o projeto de uma ontologia deve levar em consideração a utilização do vocabulário compartilhado existente, cuidando para que a representação desta possa especializar o conhecimento existente de forma monotônica.
- o **Codificação mínima:** a conceitualização deve ser especificada independente da codificação utilizada. A codificação deve ser minimizada, de forma a maximizar a capacidade de os agentes de compartilhamento se comunicarem, mesmo que usem diferentes sistemas e estilos de representação.
- o **Compromisso ontológico mínimo:** uma ontologia deve ter o mínimo de compromisso ontológico, sendo desejável apenas o suficiente para o compartilhamento de conhecimento. Deve-se afirmar o mínimo possível sobre o mundo a ser modelado, permitindo desta forma que outros sistemas possam especializar e instanciar a nova ontologia, por meio do vocabulário compartilhado.

Gruber [28] complementa os critérios levantados lembrando que deve haver um meio termo que os balanceie de forma igualitária, por exemplo, o critério “codificação mínima” ou o critério “compromisso ontológico mínimo” não pode ser mínimo ao ponto de colocar o critério “clareza” ou o critério “coerência” em falha.

Segundo Edgington et al. [22], o conhecimento ontológico considera três classes de atributos relacionados ao *conteúdo*, à *construção* e à *evolução*. Os atributos de conteúdo descrevem os objetos, como são, tamanho, forma, composição, etc. Os atributos de construção dizem como o objeto é instanciado, o que pode incluir informações de proprietário, localização, etc. Os atributos de evolução caracterizam se as considerações do sistema são dinâmicas ou estáticas, tais como a versão, identificador de revisão, hora, etc. Por fim, resta dizer que os axiomas fornecem as regras de como os objetos podem interagir entre si.

Wooldrigdge [78], tomando por base o trabalho de Noy and McGuinness [47], apresenta resumidamente os passos da metodologia 101 para a construção de uma ontologia:

- Passo 1: determinar o domínio e o escopo. Saber responder perguntas tais como “que domínio minha ontologia cobrirá?”, “Que questões minha ontologia deverá responder?” ou “Como a ontologia será usada?” podem ser de grande utilidade neste passo.
- Passo 2: considerar o reuso. O reuso é extremamente importante no desenvolvimento de ontologias, pelo simples motivo de que a ontologia será mais utilizável se todos a utilizarem como base para outras criações.

- Passo 3: enumerar todos os termos relevantes. Neste passo uma simples chuva de ideias sobre os termos associados ao domínio é suficiente.
- Passo 4: definir a hierarquia de classes e subclasses. Após a coleta das várias ideias, este passo tem o intuito de organizá-las de forma de uma taxonomia. Um erro comum neste passo é a confusão entre o que deve ser considerado classe e o que deve ser considerado propriedade.
- Passo 5: definir as propriedades. Este passo envolve a identificação, para cada classe, das propriedades associadas. As propriedades podem ser:
 - Intrínsecas: relaciona a natureza concreta do objeto – peso, comprimento, cor, etc.
 - Extrínsecas: relaciona a natureza abstrata do objeto – nome, estado civil, etc.
 - Um objeto em componentes: se o objeto estiver estruturado de alguma forma, é interessante identificar as partes que compõem o objeto.
 - Restrições de domínio: faz as restrições dos valores que podem ser atribuídos a determinada classe.
- Passo 6: definir as restrições das propriedades. Deve-se neste passo identificar as propriedades pertinentes a cada propriedade definida no passo anterior. Estas podem ser:
 - Restrições de cardinalidade
 - Restrições de tipos
 - Restrições de alcance
 - Restrições de domínio
- Passo 7: criar as instâncias. Neste estágio deve-se popular a ontologia com as instâncias das classes.

4.6 Ferramentas para o desenvolvimento de ontologias

Independente da linguagem de representação de ontologia será usada, os editores gráficos ajudam a desenvolver e a organizar toda a estrutura conceitual de uma ontologia [78]. Adicionar conceitos, propriedades, relações, restrições, lógicas e visualizar incoerências são algumas das atividades que estas ferramenta gráficas podem auxiliar [25].

Vários editores foram criados para auxiliar no desenvolvimento de ontologias, desde os mais populares – como OntoEdit [69] (comercial) e o Protégé [61] (gratuito) – até os menos conhecidos, como o Onto-ActRE [40].

4.6.1 Protégé

O Protégé é um ambiente integrado para o desenvolvimento de sistema por especialistas em domínios que desejam criar e manter ontologias. Possui código aberto, ambiente modular, uma biblioteca (API's⁶) pronta em Java para quem desejar personalizar aplicativos que façam uso da ontologia criada e a possibilidade de expandir os recursos da ferramenta por meio de *plugins* [61]. A ferramenta em execução sob um MacOS X pode ser vista, parcialmente, na Figura 4.5.

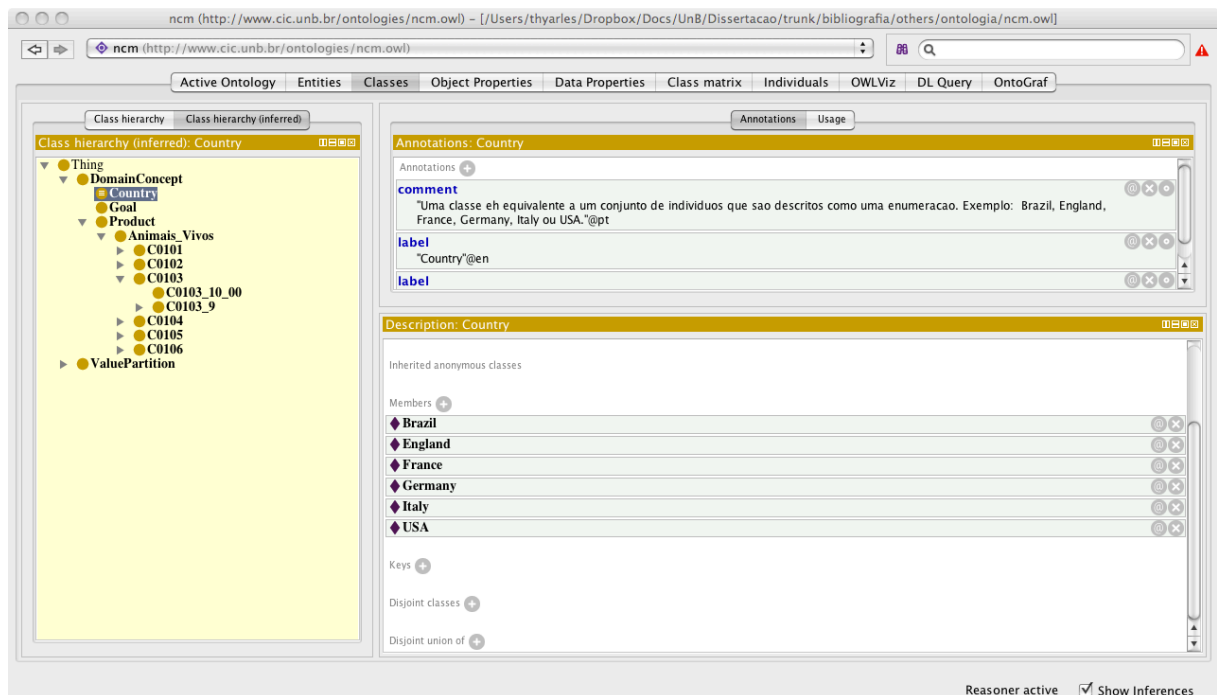


Figura 4.5: Protégé sendo executado em um MacOS X.

O Protégé permite descrever conceitos pertinentes à uma determinada ontologia de forma simplificada, por meio de uma interface gráfica. Esta ferramenta foi desenvolvida pela Universidade de Stanford, inicialmente para uso específico na medicina [61]. Esta ferramenta traz um conjunto padrão de conhecimento para a modelagem de estrutura de apoio à criação, visualização e manipulação de ontologias em vários formatos de representação. Esta ferramenta conta com duas forma para a modelagem de ontologias:

- O editor Protégé-*Frames* que permite aos usuários criar e manter ontologias baseadas em quadros, de acordo com o protocolo *Open Knowledge Base Communication* (OKBC). Neste modelo, uma ontologia consiste de um conjunto de classes organizadas em uma hierarquia de subordinação, para representar conceitos importantes de um domínio, um conjunto de *slots* associado às classes – para descrever suas propriedades e relacionamentos – e um conjunto de instâncias dessas classes que possuem valores específicos para suas propriedades.

⁶Application programming interface, ou interface de programação de aplicativos.

- O editor Protégé-OWL permite aos usuários criar ontologias para a web semântica, em particular do W3C *Web Ontology Language*. Deve-se lembrar que uma ontologia necessita de uma semântica formal específica para que os dados possam ser derivados em consequências lógicas, ou seja, fatos não presentes nas ontologias mas decorrentes da semântica.

O Protégé, atualmente, tem duas versões em uso, a 4.x (com suporte completo a OWL 2.0) e a 3.x (com suporte ao Protégé-*Frame*). Um quadro com a comparação entre os recursos das duas versões pode ser visto na Figura 4.6⁷.

Protégé 3.4	Protégé 4.0
Suporte a <i>frames</i>	Sem suporte a <i>frames</i>
OWL 1.0	OWL 2.0
OWL e RDF(S)	OWL puro
Protégé-OWL API	OWL API
Com SPARQL	Sem SPARQL
Suporte SWRL total	Suporte SWRL parcial
Suporte ao OWL <i>Full</i>	Sem suporte ao OWL <i>Full</i>
Configuração em .pprj	Configuração global
Suporte à multiusuário	Sem suporte à multiusuário
Suporte a dados em JDBC	Sem suporte a dados em JDBC

Figura 4.6: Comparação entre as versões 3.x e 4.x do Protégé [62]

4.6.2 TopBraid Composer

O “TopBraid Composer” é um ambiente de desenvolvimento de ontologias profissional para a web semântica. É totalmente compatível com os padrões W3C, oferece suporte para o desenvolvimento, gerenciamento, testes e configurações de modelo de conhecimento [74]. A aparência desta ferramenta pode ser vista na Figura 4.7.

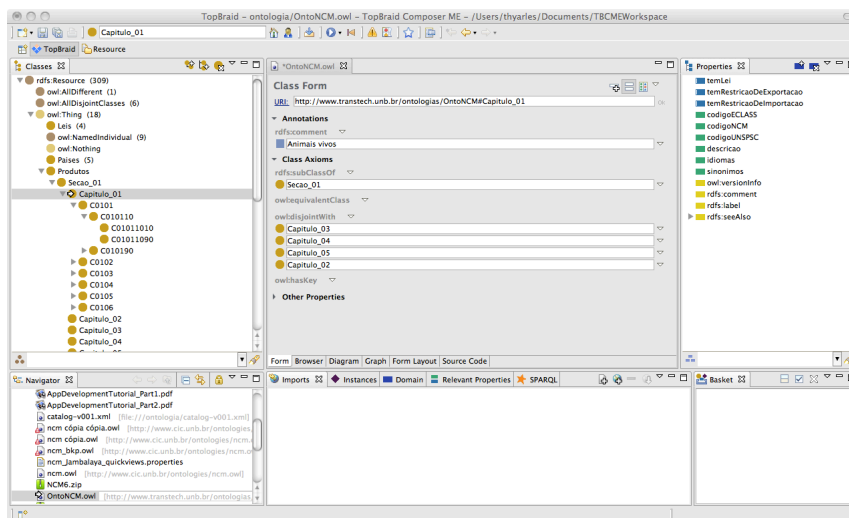


Figura 4.7: TopBraid Composer – Edição Maestro – sendo executado em um MacOS X.

⁷Os itens classificados como “sem suporte” podem ser entendidos como recursos ainda não migrados para da versão do Protégé 3.x para a 4.x.

O “TopBraid Composer” é a principal ferramenta da indústria para editar arquivos RDF e OWL, e também a melhor ferramenta SPARQL do mercado [74].

Existem três versões de uso do “TopBraid Composer”, são elas [74]: *Free Edition*; *Standard Edition* e *Maestro Edition*. A Figura 4.8 mostra uma comparação de recursos entre as três versões disponíveis.

Característica	Gratuito (free)	Padrão (standard)	Maestro
Carregar, editar e salvar arquivos RDF/XML, N3 e N-Triples	Sim	Sim	Sim
Definir ontologias utilizando formulários	Sim	Sim	Sim
Definir ontologias utilizando editor gráfico	--	Sim	Sim
Criar e executar consultas SPARQL	Sim	Sim	Sim
Criar e executar regras SPARQL (SPIN)	Sim	Sim	Sim
Criar e executar SWRL e regras Jena	--	Sim	Sim
Importar e converter para RDF de XML, UML, planilhas, RSS/Atom e BDs	--	Sim	Sim
Gerar XMLS de RDF/OWL	--	Sim	Sim
Relacionar XML e RDF/OWL (importar - exportar)	--	--	Sim
Trabalhar com bases de dados RDF	--	Sim	Sim
Trabalhar com diferentes racionadores e configurar as inferências	--	Sim	Sim
Validar a entrada do usuário usando regras SPARQL (SPIN)	Sim	Sim	Sim
Consultar banco de dados relacionais em tempo real	--	Sim	Sim
Visualizar dados RDF usando gráficos, diagramas, mapas e matrizes	--	Sim	Sim
Unir e refatorar dados RDF de diferentes origens	Básico	Sim	Sim
Executar a plataforma TopBraid Live	--	--	Sim
Gerar documentos arbitrários por meio do JSP	--	--	Sim
Definir e executar dados RDF de regras (SPARQLMotion)	--	--	Sim
Gerar inteligência de negócio (relatórios) nas páginas web	--	--	Sim
Converter e-mails em OWL, com análise semântica	--	--	Sim

Figura 4.8: Comparação de recursos do TopBraid Composer (adaptada) [74].

Capítulo 5

Metodologia e modelagem

Antes de se pensar na metodologia para desenvolver uma ontologia, deve-se entender a relação entre os dois paradigmas mais utilizados para a modelagem de ontologias: a pelo Protégé-OWL e pelo Protégé-frames. Wang et al. [77] produziram uma comparação entre as duas abordagens¹.

As construções entre os dois paradigmas são semelhantes: ambos trazem a noção de classes – que representam conceitos no domínio do discurso; as classes têm instâncias e propriedades (*slots*) – os quais descrevem os atributos e as relações entre as classes [77]. As diferenças entre os paradigmas reside na semântica de construção das ontologias, bem como na forma como são inferidos novos fatos com base nos dados existentes.

As principais diferenças entre a abordagem Protégé-OWL e Protégé-Frames são listadas a seguir [77]:

- *Nome exclusivo*: se dois objetos têm nomes diferentes, no Protégé-Frames estes serão considerados diferentes, a menos que exista explícita indicação do contrário. Já no Protégé-OWL, nenhuma suposição sobre isto é feita.
- *Suposição do mundo fechado contra suposição do mundo aberto*: No Protégé-Frames tudo é proibido se não houver permissão explícita; já no Protégé-OWL tudo é permitido se não houver negação explícita. A conclusão lógica disto é a de que nada pode ser inserido na base de conhecimento do Protégé-Frames sem que tenha um lugar correspondente; por outro lado, considerando a base de conhecimento do Protégé-OWL, qualquer coisa pode ser inserida, desde que não viole as restrições impostas.
- *Modelo simples contra modelos múltiplos*: No Protégé-Frames a ontologia tem apenas um modelo mínimo, que deve preencher cada uma das afirmações ontológicas, o que significa que o modelo só pode ter instâncias que sejam previstas. Já o Protégé-OWL fornece muitos modelos, de forma a contemplar todas as possíveis interpretações que satisfaçam cada uma das asserções em uma ontologia OWL.

¹O OWL suporta três níveis de sublinguagens, de acordo com a expressividade requerida: OWL *Lite*, OWL DL e OWL *Full*. Na pesquisa de [77] foi utilizada o subnível de linguagem OWL DL.

Em última análise, pode-se dizer que o paradigma Protégé-*Frames* é útil para aplicações que desejam: criar ontologias para domínios em que seja necessário a hipótese do mundo fechado; a aplicação está focada na aquisição de dados; a aplicação requer restrições de valores nas propriedades e, por fim, o modelo relaciona classes com outras classes. Já o paradigma Protégé-OWL é válido para as aplicações que desejam: criação de terminologias robustas em que as classes são definidas; necessidade do raciocínio para garantir que as ontologias estão consistentes; terminologias controladas são publicadas via web semânticas e acessadas por outros aplicativos e, por fim, aplicações na qual a classificação é um paradigma para o raciocínio [77].

É necessário salientar que este trabalho utilizará o paradigma Protégé-OWL por não ter foco na aquisição de dados, por exigir terminologias robustas, por requerer um raciocínio correto em tempo finito, e pela necessidade de consistência no modelo de ontologia proposto.

5.1 Metodologia

Noy and McGuinness [47] propuseram um método para a criação de ontologia intitulado *Ontology Development 101: A Guide to Creating Your First Ontology*, ou apenas “Método 101”. Na presente pesquisa será utilizado este método, por apresentar passos bem definidos, documentados e já aprovados em outras pesquisas semelhantes, das quais se destacam:

- Dogac et al. [20] utilizam o “Método 101” para fazer uma comparação entre uma taxonomia e uma ontologia, destacando os pontos fortes desta sobre aquela;
- Programme [55], ao explicarem como a ontologia UNSPSC foi criada, citam que o “Método 101” foi considerado;
- Bergamaschi et al. [8] utilizaram o “Método 101” para a integração e classificação de dados do comércio eletrônico;
- Su et al. [68] utilizaram o “Método 101” para o desenvolvimento de uma ontologia para um agente de sistema.
- Baker and Cheung [5] reuniram várias pesquisas referentes à ontologia em uma única publicação, e em várias delas o “Método 101” foi utilizado.
- Cardoso [14] reuniu pesquisas referentes à ontologia e web semântica, e em algumas delas o “Método 101” foi utilizado.

O desenvolvimento de uma ontologia é diferente da construção de classes e relações na programação orientada a objetos. Enquanto esta se preocupa com os métodos e classes – o programador projeta as classes baseadas nas propriedades operacionais da classe (foco

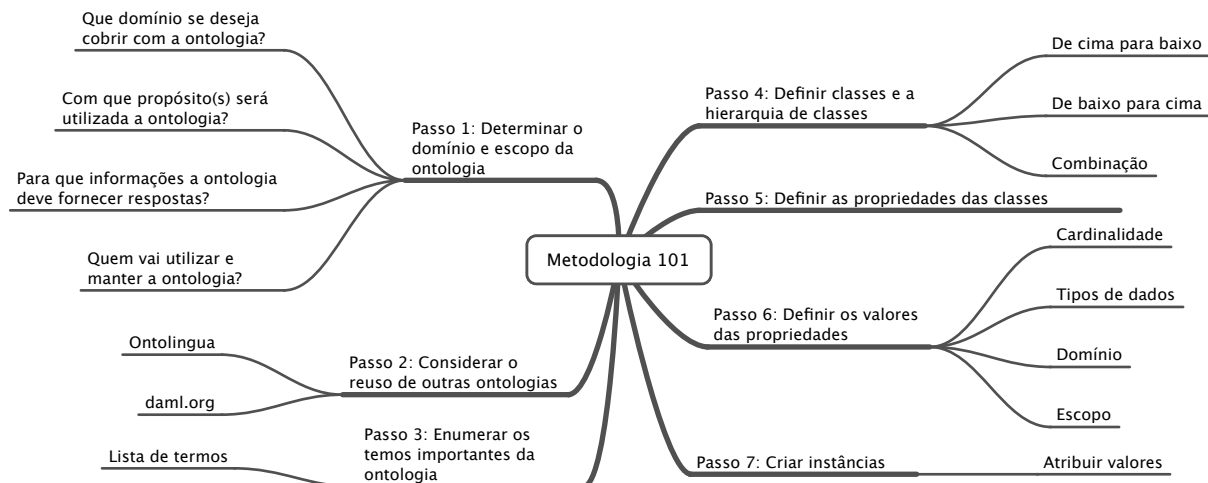


Figura 5.1: Metodologia 101 (adaptada) [12]

no comportamento da classe) –, aquela se preocupa nas propriedades estruturais da classe (foco nos dados da classe).

Em termos práticos, o desenvolvimento de uma ontologia inclui [47]: definir as classes da ontologia; arranjar estas classes na forma de uma taxonomia; definir as propriedades e os valores possíveis para elas e, por fim; preencher os valores das propriedades para cada instância. A ontologia é um modelo real do mundo e os conceitos na ontologia devem refletir tal realidade. Como dito antes, o desenvolvimento de uma ontologia é um processo iterativo que, necessariamente, sofrerá mudanças em todo o ciclo de vida de uma ontologia.

Os passos propostos por Noy and McGuinness [47] para o desenvolvimento de uma ontologia foram resumidos por Breitman [12], e estão ilustrados na Figura 5.1.

Algumas regras fundamentais no projeto de uma ontologia são [47]: não existe um modo correto para se modelar um domínio; o processo de criação de uma ontologia é necessariamente iterativo; por fim os conceitos devem ficar ligados aos objetos e relacionamentos em cada domínio de interesse.

5.1.1 Passo 1: Determine o domínio e escopo da ontologia

Neste passo deve-se determinar o domínio e escopo da ontologia. Responder aos questionamentos a seguir é o modo proposto por Noy and McGuinness [47]:

1. Qual(is) domínio(s) a ontologia deverá cobrir?

A proposta desta pesquisa é desenvolver uma ontologia para facilitar o uso da Nomenclatura Comum do Mercosul (NCM) em um meio que possa ser disponibilizado na internet, estruturar o conhecimento obtido na rede mundial de computadores e nas bibliografias especializadas no assunto.

Deseja-se produzir uma ontologia genérica – “upper ontology” – para que seja utilizada como base em outros aplicativos. A ontologia cobrirá os domínios NCM, países, restrições, leis, sinônimos e idiomas (português, inglês e espanhol) para a identificação dos produtos.

2. Para que nós vamos utilizar ontologia?

A ontologia tem inúmeras vantagens sobre a taxonomia, uma vez que aquela torna possível a interligação de distintos domínios – por meio de mapeamento manual, mapeamento automático [7, 36, 38, 67, 48] ou outro método – o que motiva o reuso das ontologias já existentes.

Os sistemas de classificação estrangeiros (UNSPSC, e-Class, RosettaNet, etc.) já possuem ontologias definidas [8, 20, 55, 31, 68, 57] e em uso, mas a Nomenclatura Comum do Mercosul (NCM) não, o que impossibilita o intercâmbio de informações desta com aquelas para a correlação dos produtos brasileiros com os estrangeiros.

Assim, a utilização de uma ontologia tem por objetivo não somente estruturar o conhecimento sobre a NCM, mas ser uma proposta também para o mapeamento entre a nomenclatura brasileira e as nomenclaturas internacionais.

3. Quais tipos de questões a ontologia deve fornecer respostas?

Espera-se que a ontologia responda às seguintes perguntas:

- (a) *Quais produtos brasileiros não podem ser exportados para os Estados Unidos da América?*
- (b) *Quais produtos da Alemanha não podem ser importados para o Brasil?*
- (c) *Qual o código NCM do produto Cavalo? E as restrições deste?*
- (d) *Quais os sinônimos do produto “Notebook”?*
- (e) *Qual o código do produto “égua” na NCM?*
- (f) *Quais leis estão ligadas ao produto “baleias”?*
- (g) *O que se deve considerar ao exportar um terminal bancário para os EUA?*
- (h) *Qual é a correta classificação do produto “ganso”?*
- (i) *Quais produtos não podem ser exportados para a China?*
- (j) *Qual o código do produto “Computador” no código UNSPSC? E no e-Class?*

4. Quem irá usar e manter a ontologia?

A ontologia será usada por pessoas interessadas em classificar suas mercadorias, por meio de um sistema “online” que terá a vistoria, inclusive, dos intermediários nas operações (bancos, importadoras, exportadoras, etc.).

A manutenção se dará, preferencialmente, por pessoas especialistas no assunto e que estejam envolvidas de alguma forma com importações e/ou exportações. Espera-se

que estes especialistas sigam estritamente as normas brasileiras para a Nomenclatura Comum do Mercosul, alterando-se a ontologia quando as normas forem alteradas.

O principal motivo destas perguntas reside na aplicação de um limite no escopo do modelo, a fim de evitar que se tente construir uma ontologia grande e que não ofereça respostas satisfatórias. Pelas questões apresentadas, observa-se que a ontologia deverá disponibilizar meios para relacionar países, produtos, mapeamento com outros códigos já existentes, línguas, sinônimos, etc.

5.1.2 Passo 2: Considere a possibilidade de reutilizar ontologias existentes

Em pesquisas extensas, mas não exaustivas², não se encontrou nenhuma ontologia existente para a Nomenclatura Comum do Mercosul. As ontologias que serviram como inspiração foram as UNSPSC [55] – citada na seção 2.5.1 – e eClass [31] – citada na seção 2.5.2 e a RosettaNet³ – não estudada neste trabalho.

As ontologias UNSPSC e eClass deram origem a várias pesquisas sobre mapeamento, alinhamento e relacionamento de ontologias, tais como:

- Omelayenko [49] propôs a integração de diferentes vocabulários por meio da descoberta, representação e execução de um mapa de vocabulário. Para validar a proposta, o autor utilizou 100 amostras de cada uma das ontologias – UNSPSC e eClass, em um contexto de mercadorias e serviços comum entre as duas nomenclaturas. Para a classificação utilizou a técnica denominada “*Bayesian*”, a qual foi aplicada com sucesso em outras propostas de classificação de produtos por meio de exemplos pré-classificados. Já para o mapeamento entre classes e propriedades o autor desenvolveu uma ferramenta denominada RDF *Transformation* (RDFT), a qual tem a capacidade de fazer ponte entre classes (*Class2Class*), entre marcações e classes ou classes e marcações (*Tag2Class* ou *Class2Tag*), entre marcações e propriedades ou propriedades e marcações (*Tag2Property* ou *Property2Tag*).
- Bergamaschi et al. [8] propõem um *framework* para a integração de dados e classificação de produtos no comércio eletrônico. Os comércios eletrônicos baseados em *business to business* (B2B) e *business to consumer* (B2C) encontram grandes barreiras ao se depararem com os inúmeros padrões para a classificação de mercadorias, e o objetivo da pesquisa em questão foi encontrar uma metodologia que fizesse a integração e conciliação das informações nesta área. O sistema *Mediator environment for Multiple Information System* (MOMIS) foi desenvolvido como produto final e, segundo os autores, este faz o mapeamento de produtos e serviços entre os padrões de

²A pesquisa foi feita no Google e em sítios especialistas no armazenamento de ontologias tais como <http://www.ksl.stanford.edu/software/ontolingua>, <http://www.daml.org/ontologies>, etc.

³<http://www.rosettanel.org>

classificação UNSPSC, eClass e ECCMA⁴. O sistema faz a extração da informação de diferentes fontes de dados, seja ela relacional, estruturada ou semi-estruturada (XML) e as integra em um esquemático virtual de todos os dados em XML, com um *thesaurus* – baseado na *WordNet*⁵ [21] – em comum para fazer a relação de termos. Na Figura 5.2 pode-se ver um exemplo mapeamento entre as nomenclaturas UNSPSC e eClass gerado pela ferramenta.

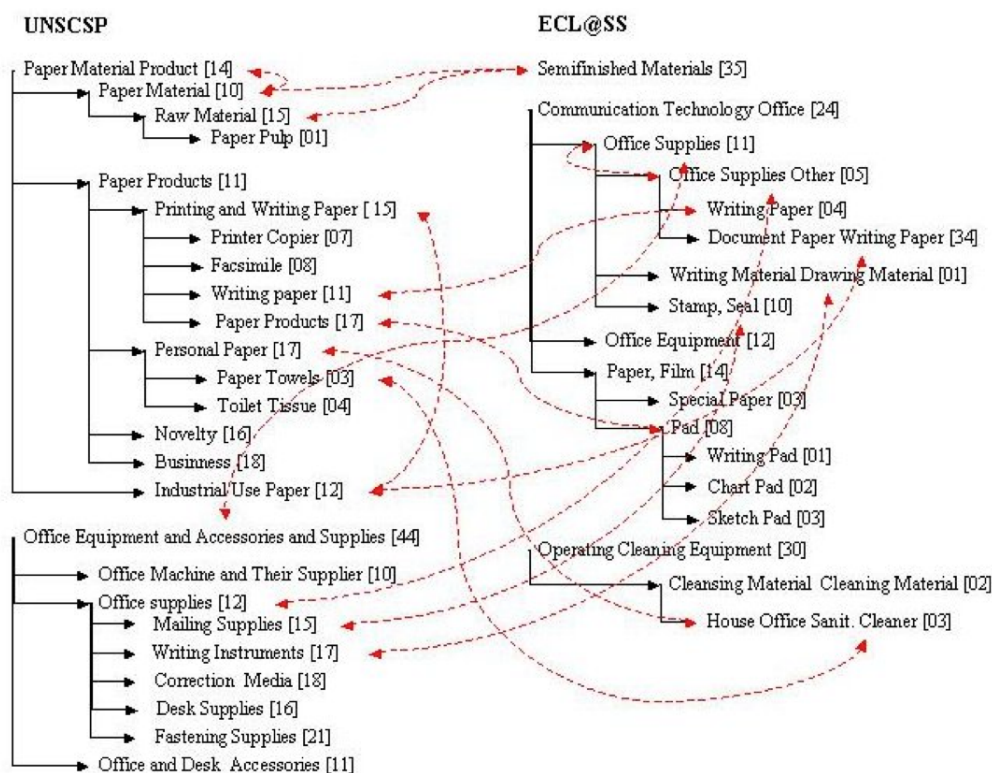


Figura 5.2: Um mapa resultante do sistema MOMIS [8]

5.1.3 Passo 3: Enumere os termos importantes na ontologia

Para o bom desenvolvimento de uma ontologia é importante escrever uma lista de termos relacionados com o domínio a ser modelado.

No caso da Nomenclatura Comum do Mercosul, o termo maior gira em torno dos produtos nelas listados e as propriedades seriam os relacionamentos da ontologia criada com outros domínios, como ilustrado na Figura 5.3.

⁴Electronic Commerce Code Management Association (<http://www.ucec.org>)

⁵<http://www.cogsci.princeton.edu/wn>

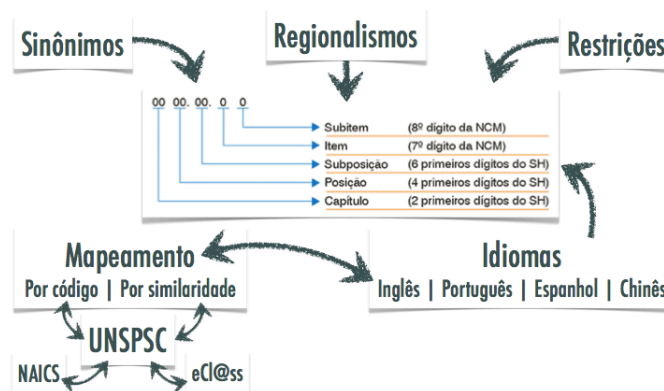


Figura 5.3: Domínios a serem ligados à NCM

Pretende-se tratar nesta pesquisa apenas os domínios “sinônimos”, “restrições”, “leis”, “código UNSPSC”, “código eClass” e “idiomas”. Em caso de sucesso, a metodologia proposta pode ser aplicada para a relação com taxas de importação, taxas de exportação, impostos, etc. Assim, a lista de termos – não exaustiva – é

- *Animais vivos, reprodutores de raça pura, cavalos, bonivos, suínos, ovinos, caprinos, etc. Ou seja, os produtos listados pela Aduandeiras [1].*
- *Quais as restrições de importação dos produtos para o Brasil*
- *Quais as restrições de exportação de produtos do Brasil*
- *Quais leis governam determinado produto*
- *Quais os termos de determinado produto nas línguas inglesa, espanhola e portuguesa*
- *Quais os termos regionais de determinado produto – sinônimos*
- *Qual o código de determinado produto*

5.1.4 Passo 4: Defina as classes e suas hierarquias

Para a definição da hierarquia, existem algumas possibilidades:

- Método *top-down*, onde o desenvolvimento se inicia dos conceitos mais gerais para os mais específicos, especializando-os. No exemplo do vinho, pode-se criar as classes genéricas vinho e alimentos; em seguida a classe vinho será especializada em branco, *rosé* e tinto, e assim por diante.
- Método *bottom-up*, onde o desenvolvimento se inicia nas classes mais específicas e, em seguida, agrupa-se tais classes em conceitos mais genéricos. Um exemplo, também dos vinhos, seria criar classes para todos os fabricantes de vinhos e, em seguida, separá-los por tipos de vinhos fabricados (tinto, branco e *rosé*).

NCM	Descrição
01.01	ANIMAIS VIVOS DAS ESPÉCIES CAVALAR, ASININA E MUAR.
0101.10	- Reprodutores de raça pura
0101.10.10	Cavalos
0101.10.90	Outros
0101.90	- Outros
0101.90.10	Cavalos
0101.90.90	Outros
01.02	ANIMAIS VIVOS DA ESPÉCIE BOVINA.
0102.10	- Reprodutores de raça pura
0102.10.10	Prenhes ou com cria ao pé
0102.10.90	Outros
0102.90	- Outros
0102.90.1	Para reprodução
0102.90.11	Prenhes ou com cria ao pé
0102.90.19	Outros
0102.90.90	Outros
01.03	ANIMAIS VIVOS DA ESPÉCIE SUÍNA.
0103.10.00	- Reprodutores de raça pura
0103.9	- Outros:
0103.91.00	- - De peso inferior a 50kg
0103.92.00	- - De peso igual ou superior a 50kg

Figura 5.4: Classes e hierarquias da ontologia NCM [1].

- A combinação dos dois métodos anteriores também é possível, onde os conceitos são definidos na medida em que aparecem. Em seguida, quando se tiver vários conceitos, estes podem ser especializados ou generalizados, de acordo com a ocasião e necessidade.

Nenhum dos três métodos é o melhor ou pior, já que dependem do caso. O método a ser seguido depende principalmente da visão do engenheiro que está projetando a ontologia. Se o engenheiro tem uma visão do genérico para o especializado, obviamente este se sairá melhor se escolher o método *top-down*. O mais comum é seguir uma combinação dos métodos, pois se adequaria a maioria da visão dos desenvolvedores.

No caso da ontologia para a NCM, será usado o método *top-down*, uma vez que a taxonomia já está completamente definida no documento produzido pela Aduandeiras [1]. Uma parte das classes e hierarquias podem ser vistas na Figura 5.4.

5.1.5 Passo 5: Defina as propriedades de interesse das classes

As propriedades são aquelas que ligam duas classes – binárias –, para que as relações e deduções de associações sejam possíveis. Algumas propriedades pensadas são:

- *temRestricoesDeImportacao*
- *temRestricoesDeExportacao*
- *temLeis*
- *codigoNCM*
- *codigoEclass*

- *codigoUNSPSC*
- sinonimos
- idiomas
- peso

5.1.6 Passo 6: Defina as restrições das propriedades

As propriedades não têm restrições específicas, pois o conteúdo de todas é um literal (*string*).

5.1.7 Passo 7: Crie as instâncias

A Nomenclatura Comum do Mercosul, diferentemente da UNSPSC ou eClass, tem descrições longas para cada hierarquia na classe.

Por exemplo, o código 85.05 traz na descrição “*Eletroímãs, imãs permanentes e artefatos destinados a tornarem-se imãs permanentes após magnetização; placas; mandris e eletromagnéticos, de fixação; acoplamento, embreagens, variadores de velocidade e freios, eletromagnéticos; cabeças de elevação eletromagnéticas*”. O mapeamento automático deste tipo de dado com qualquer nomenclatura existente é uma tarefa hercúlea, dada a complexidade dos títulos adotados no Brasil.

Assim, pretende-se nesta ontologia não somente criar as relações e restrições necessárias para que esta atenda aos requisitos propostos anteriormente, mas também que as instâncias sejam quebradas em átomos, de forma que um único código possa ser relacionado aos equivalentes. Isto facilitaria o entendimento da NCM e a busca pelo que se deseja seria mais precisa.

5.2 Modelagem

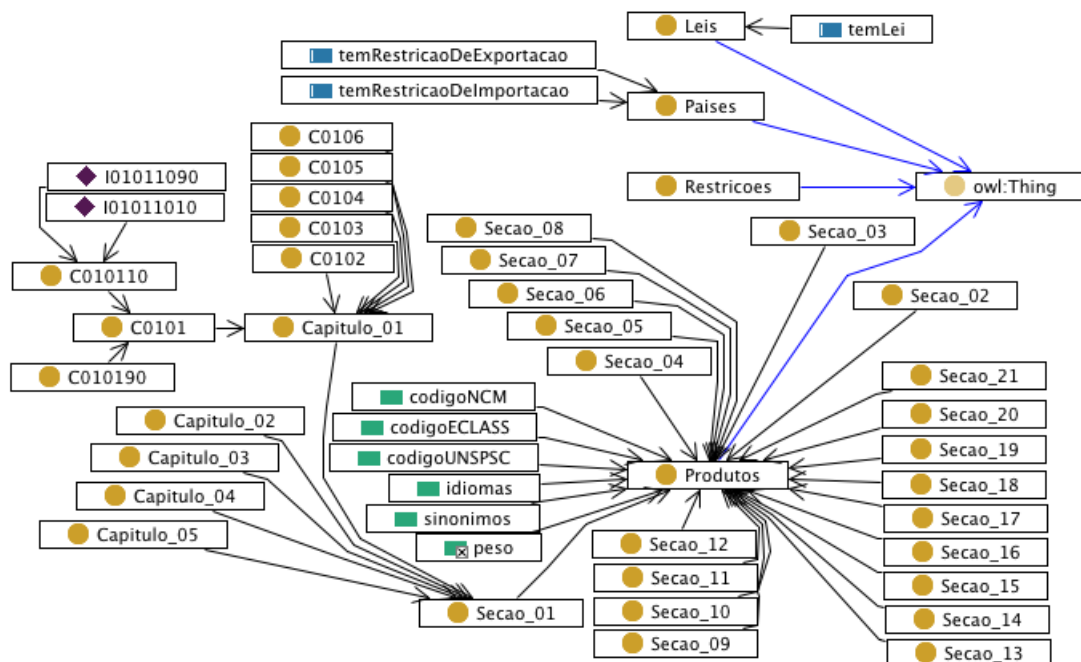
Para a solução do problema, utilizou-se o Protégé 4.1 [63] para a modelagem da ontologia, o Protégé 3.4.7 [64] para a geração de gráficos – a versão 4.1 ainda não tem os complementos necessários para esta função – e, por fim, o TopBraid Composer [74] para as consultas SPARQL sobre a ontologia pronta.

No Protégé 4.1 seguiu-se parte da taxonomia proposta pela Aduandeiras [1]. Em seguida foram cadastrados dados relacionados (leis, restrições de importação, etc.) ao problema, em especial os que têm relação com a área de interesse – Mecatrônica.

Na Seção XVI o Capítulo de interesse foi o 8471, por se tratar de equipamentos computacionais. No entanto o termo computador não consta na NCM. Segundo Dalston [16], o motivo é que o termo a ser utilizado na NCM é o “máquinas automáticas para processamento de dados”. Este termo era usado no início da informática no Brasil, início dos anos 70. Dalston [16] explica ainda que:

De acordo com as NESH, as máquinas automáticas para processamento de dados são máquinas aptas a fornecer, por meio de operações logicamente ligadas umas às outras, e que se sucedem em uma ordem predeterminada (programa), dados diretamente utilizáveis ou suscetíveis, em certos casos, de servir eles mesmos de dados para outras operações de processamento de dados. Assim, consideram-se máquinas automáticas para processamento de dados, na acepção da posição 8471, as máquinas capazes de:

5.2.1 Visão geral



5.2.2 Classes

As quatro classes iniciais produzidas no Protégé 4.1 [63] podem ser vistas na Figura 5.6:

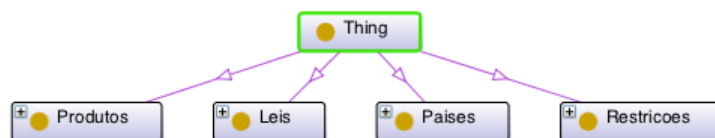


Figura 5.6: Classe *Thing*

- Classe *Thing*: classe inicial de toda ontologia, é o ponto de partida para a criação de novas classes. Esta herda quaisquer objetos que estejam nas classes filhas, e tem este nome característico – “coisa” – por poder representar qualquer coisa que exista no mundo real, seja algo abstrato ou concreto. Todas as propriedades configuradas nesta classes também são herdadas pelas classes filhas, após acionado algum motor de inferência no Protégé.
- Classe *Leis*: esta classe mantém as leis de interesse, que têm relação com os produtos listados pela taxonomia NCM. Por exemplo, um produto que tenha alguma regulamentação especial, como é o caso da “pólvora”, tem leis específicas que podem ser listadas e relacionadas por meio desta classe.
- Classe *Países*: relaciona os países de interesse que exporta e/ou importam para o Brasil. Em casos genéricos, em que uma determinada exportação a partir do Brasil seja restrita para todos os lugares do mundo, criou-se a instância “Mundo”. Pode-se ainda criar instância para continentes, ou qualquer outra representação que agrupe nações por características semelhantes. Para a melhor organização da ontologia, nestes casos a melhor opção seria criar uma classe para *Continentes*, ou o nome mais pertinente ao agrupamento em questão.
- Classe *Restricoes*: esta classe lista todas as restrições possíveis para a comercialização de produtos, exportação e/ou importação, com o Brasil. Neste trabalho, as reais restrições que envolvem a exportação e importação de produtos foram mínimas, já que o foco é tão somente provar a viabilidade de se produzir uma ontologia para tal fim. Futuramente podem ser feitas melhorias neste sentido.

As classes filhas das quatro classes anteriores são explicadas a seguir:

- Classe *Produtos*: reproduz a taxonomia da Nomenclatura Comum do Mercosul (NCM), de acordo com a documentação da Aduandeiras [1]. Devido à complexidade de toda a NCM, a taxonomia em questão foi reproduzida parcialmente. Pode-se ver na Figura 5.7 que apenas as Seções I e XVI foram elaboradas – isto pode ser observado pelo símbolo de “+” no canto superior esquerdo.

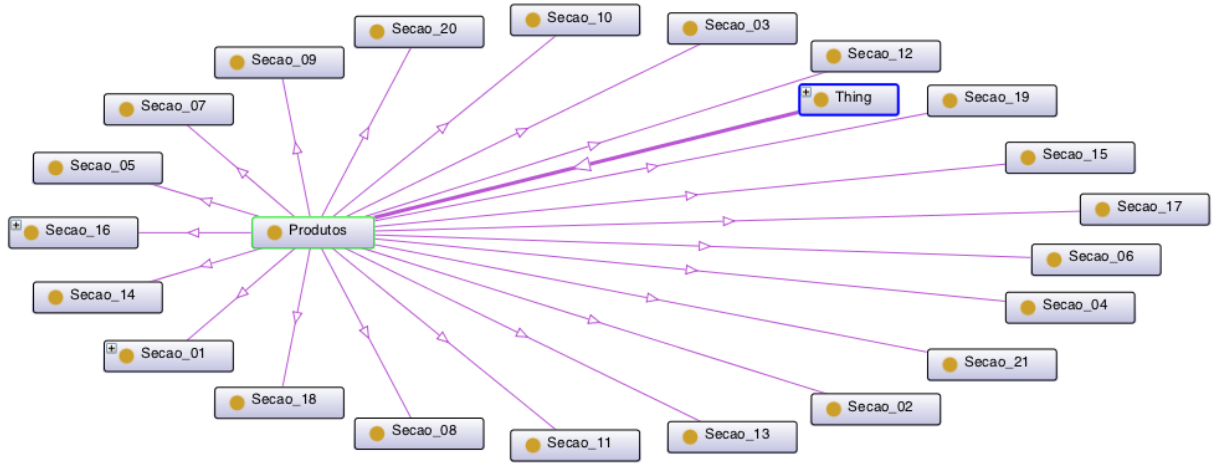


Figura 5.7: Classe *Produtos*

- Classe *Secoes_xx*: reproduz as Seções da Nomenclatura Comum do Mercosul, conforme estudado no Capítulo2, Definição e Aplicação da NCM. A variável *xx* varia de acordo com as seções, que originalmente são escrita em números romanos. Na solução do problema preferiu-se adotar os números arábicos, iniciando-se na Seção I (*Secao_01*) até a Seção XXI (*Secao_21*). Para melhor entendimento, a expansão da classe *Secao_01* está ilustrada na Figura 5.8a.

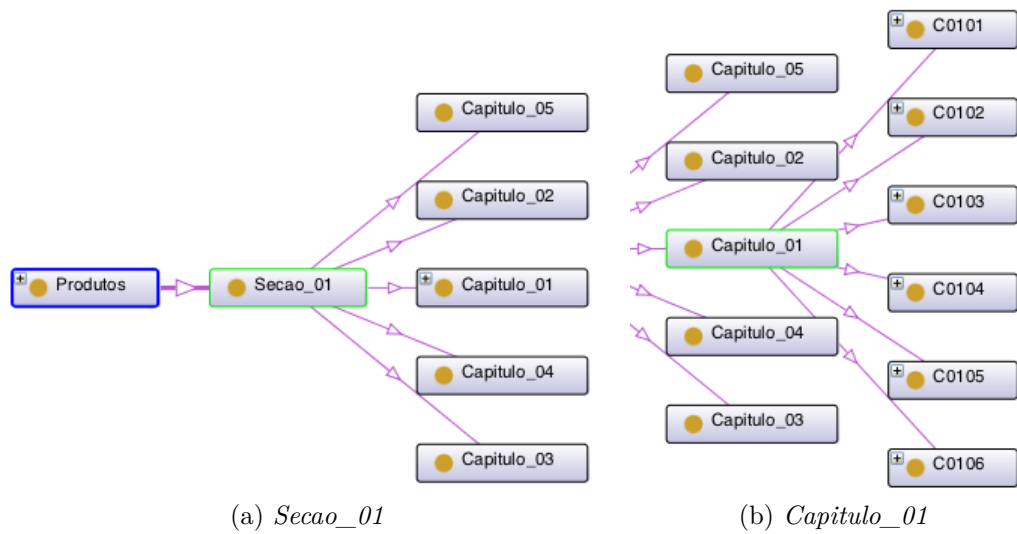


Figura 5.8: Classes *Secao_01* e *Capitulo_01*.

Observe que dos capítulos pertinentes à esta seção, apenas o Capítulo 1 foi implementado, conforme pode ser visto na Figura 5.8b. A expansão da classe *C0101*, referente à primeira seção e primeiro capítulo da NCM é ilustrada na Figura 5.9, em sua forma totalmente expandida.

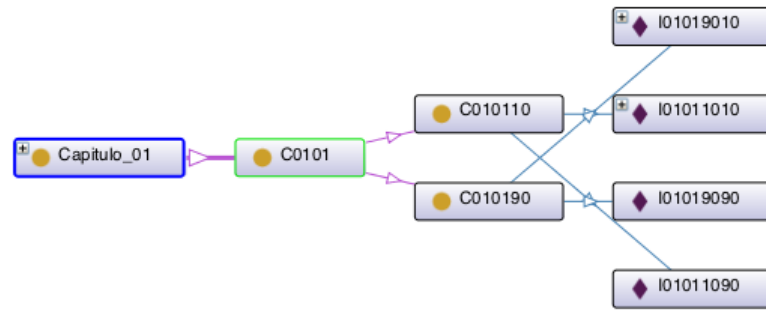


Figura 5.9: Classe *C0101*

Observa-se na Figura 5.9 que as classes iniciam com a letra “C”, e as instâncias com a letra “I”. Também que os nomes das classes e instâncias não foram pontuados, como está representado no material fornecido pela Aduandeiras [1]. O produto de código “0101.10.10 - Cavalos” está representando pela instância “I01011010” na ontologia. Esta representação foi utilizada para evitar conflitos na geração do código das classes para a linguagem Java, uma vez que esta não permite que as classes iniciem com algo que não sejam letras ou que o nome da classe contenha pontos – o ponto é um caractere reservado na linguagem Java.

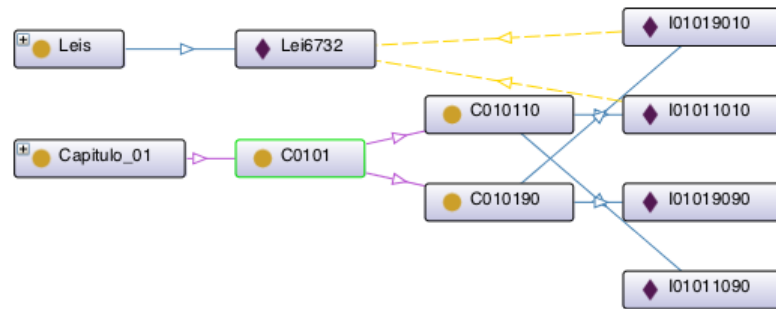


Figura 5.10: Instâncias da classe *C010110*.

Conforme observado na Figura 5.10, o código final do produto foi representado por meio de instâncias. Os códigos 0101.10.10 e 0101.90.10 estão ligados à cavalos, reprodutores de raça pura e reprodutores sem raça pura, respectivamente. Observa-se que estes tem uma ligação com a Lei 6732 (fictícia), uma instância da classe *Leis*. Assim, para a classificação destes produtos seria imprescindível uma lida nas leis relacionadas. Uma outra forma de ver a relação da Figura 5.10 pode ser vista na Figura 5.11; nesta as propriedade de exibição foram modificadas de classe para comentário.

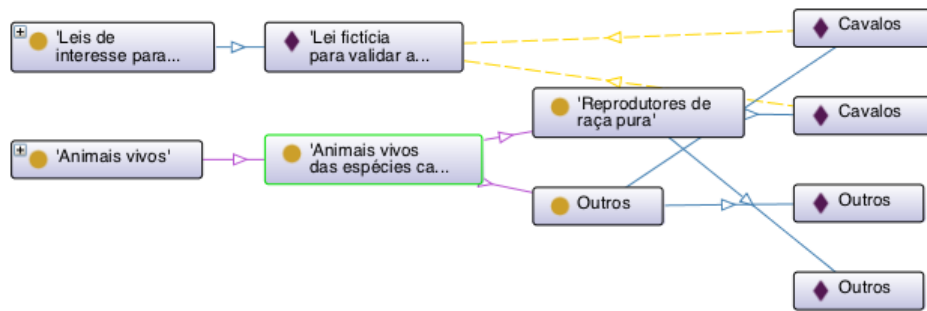


Figura 5.11: Outra forma de exibição das instâncias da classe *C010110*.

5.2.3 Propriedades do objeto

As propriedades binárias entre os objetos da ontologia podem ser vistos na Figura 5.12. Esta figura foi gerada pelo aplicativo “TopBraid Composer” [74].

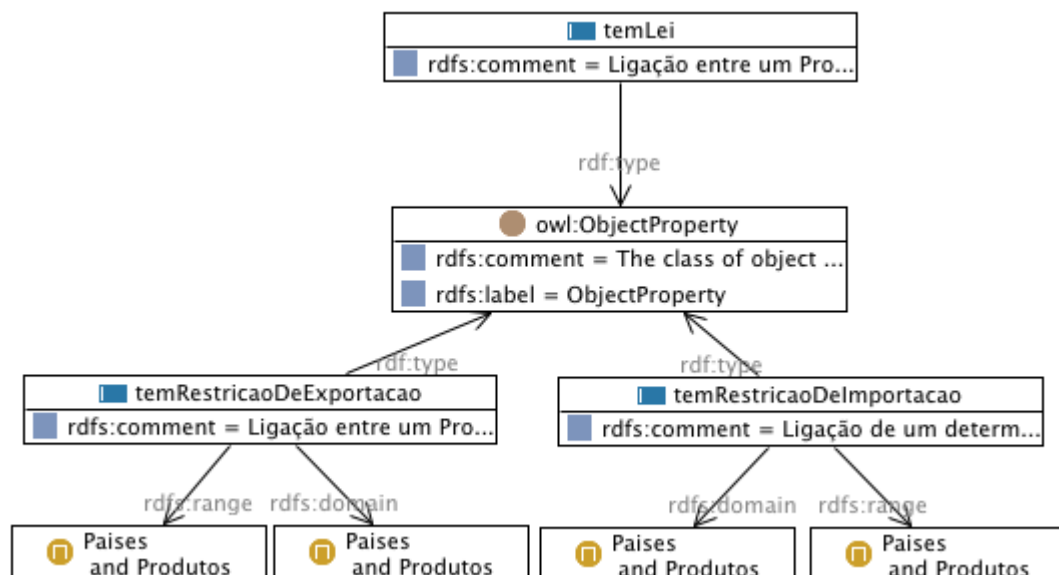


Figura 5.12: Propriedade dos objetos.

- Propriedade *temLei*: esta propriedade faz a ligação entre as instâncias de leis ou decretos registradas na ontologia com os produtos relacionados. Um produto pode ter várias leis e decretos relacionados, assim como uma mesma lei pode se aplicar a diversos produtos.
- Propriedade *temRestricaoDeExportacao*: esta propriedade faz a ligação entre determinada restrição de exportação e um produto para um destino, sendo sempre o Brasil a referência de origem.
- Propriedade *temRestricaoDeImportacao*: esta propriedade faz a ligação entre determinada restrição de importação e um produto para o Brasil.

5.2.4 Propriedade dos dados

Os dados relacionados aos produtos podem ser vistos na Figura 5.13. Esta figura foi gerada pelo aplicativo “TopBraid Composer” [74].

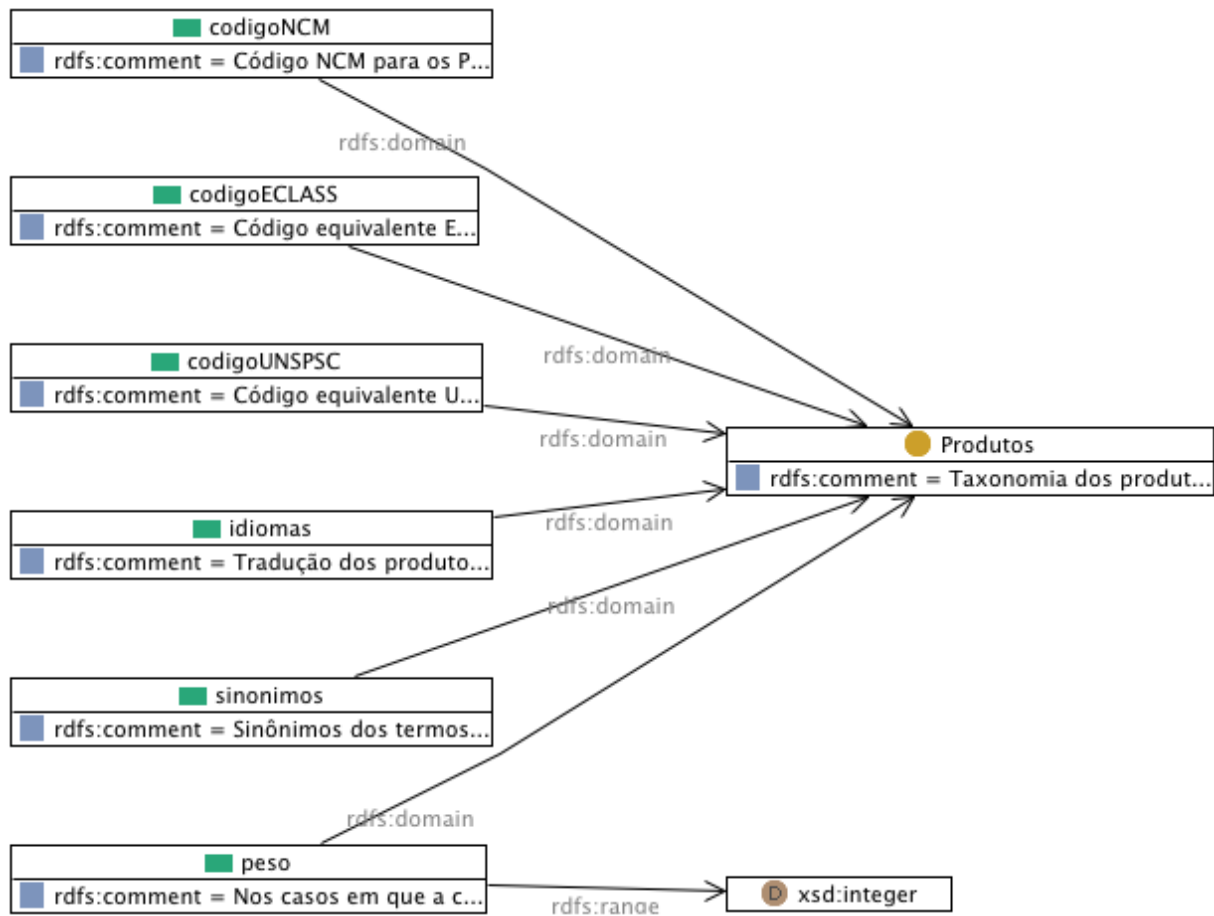


Figura 5.13: Propriedade dos dados.

- Propriedade *codigoNCM*: esta propriedade guarda o código dos produtos relacionados na Nomenclatura Comum do Mercosul tal como estão escritos, para fins de utilização em aplicações finais. Por exemplo o produto “cavalo reprodutor de raça pura” teria como resultado neste campo o código “0101.10.10”, com esta formatação padrão.
- Propriedade *codigoECLASS*: esta propriedade tem a função de guardar o código na nomenclatura eClass [31] correspondente. Os dados armazenados são todos fictícios, uma vez que objetivo é tão somente produzir ontologia que resolva alguns problemas relacionados à NCM, e neste momento não se fez necessário a consulta de um especialista no assunto para efetuar o mapeamento correto entre nomenclaturas.
- Propriedade *codigoUNSPSC*: esta propriedade tem a função de guardar o código na nomenclatura UNSPSC [55] correspondente. Os dados armazenados são todos fictícios, uma vez que objetivo é tão somente produzir ontologia que resolva alguns

problemas relacionados à NCM, e neste momento não se fez necessário a consulta de um especialista no assunto para efetuar o mapeamento correto entre nomenclaturas.

- Propriedade *idiomas*: nesta propriedade pode-se adicionar tantos quantos forem os idiomas necessários para um determinado produto. Por exemplo, o produto cavalo tem as seguintes entradas: “cavalo”@pt, “horse”@en e “caballo”@es. A indicação da língua é feita logo após o sinal de arroba, que são respectivamente: português, inglês e espanhol. Com o objetivo é produzir uma ontologia funcional, apenas os produtos relevantes foram traduzidos para os três idiomas propostos. Uma busca nesta propriedade deve sempre indicar a língua, por exemplo, quem deseja buscar *horse* deve saber de antemão que esta palavra está na língua inglesa e, então, buscar pelo termo *horse@en*.
- Propriedade *sinônimos*: esta propriedade tem o objetivo de guardar sinônimos e expressões idiomáticas para os produtos cadastrados como instâncias. Isto é muito útil para popularizar a Nomenclatura Comum do Mercosul, que é carregada de termos técnicos. Um exemplo de uso da propriedade foi no termo *cavalo*:
 - uma pessoa que pouco conhece da NCM tem que classificar seu produto, mas o que ela deseja vender é “uma égua reprodutora sem raça pura”. Ao entrar no sistema digita o termo *égua* e não obteria qualquer resultado. Com a ontologia proposta a busca retornaria as instâncias *I01011010* e *I01019010* seriam retornadas, representando respectivamente que o cavalo pode ser um reprodutor de raça pura ou um reprodutor sem raça pura.
- Propriedade *peso*: alguns produtos na NCM são classificados de acordo com o peso. Por exemplo, no código 0105.11.10 está o galo com peso não superior a 185g; se o galo tiver peso superior, deve ser classificado com o código 0105.94.00. Assim o objetivo desta propriedade é armazenar os limites de peso das mercadorias em que isto se aplique, para que este dado seja retornado nas buscas e, assim, evite erro de classificações.

Capítulo 6

Estudo de caso

O estudo de caso tem o objetivo de avaliar se o modelo proposto de ontologia fornece respostas corretas para as mais variadas questões. Para a validação inicial da ontologia produzida serão simuladas situações de uso, com questões possíveis no mundo real, por meio das ferramentas:

- Protégé [63] – utilizando a língua de consulta nativa *DL Query*;
- TopBraid Composer [74] – utilizando a língua de consulta SPARQL, vista na Seção 3.2.1;

Uma vez validadas as informações, passa-se a responder as perguntas propostas no Capítulo 5, Seção 5.1, com a ferramenta TopBraid Composer [74] – utilizando a língua de consulta SPARQL, vista na Seção 3.2.1;

Como proposta de utilização em uma aplicação nativa e específica, apresenta-se um código de programação, na linguagem Java, para acesso modelo de ontologia proposto OntoNCM. O código fez uso das bibliotecas Jena [35], e foi desenvolvido sob o ambiente de criação Eclipse [73].

Uma análise crítica dos resultados obtidos é apresentada no final do capítulo, avaliando se o modelo de ontologia proposto efetivamente otimiza a forma de uso da Nomenclatura Comum do Mercosul, bem como se este modelo poderia ser aperfeiçoado para uso em transações reais no comércio eletrônico.

6.1 Testes do modelo

6.1.1 Pelo Protégé

A primeira consulta no Protégé foi na forma mais básica, utilizando a aba *OntoGraf*, conforme ilustrado na Figura 6.1. Na Figura 6.1 foi feito uma busca pelo produto de código 0101.90.10, e então foram retornados todos os dados referentes a ele, que podem ser melhor visualizados na Figura 6.2.



Figura 6.1: Aba *OntoGraf* do Protégé.

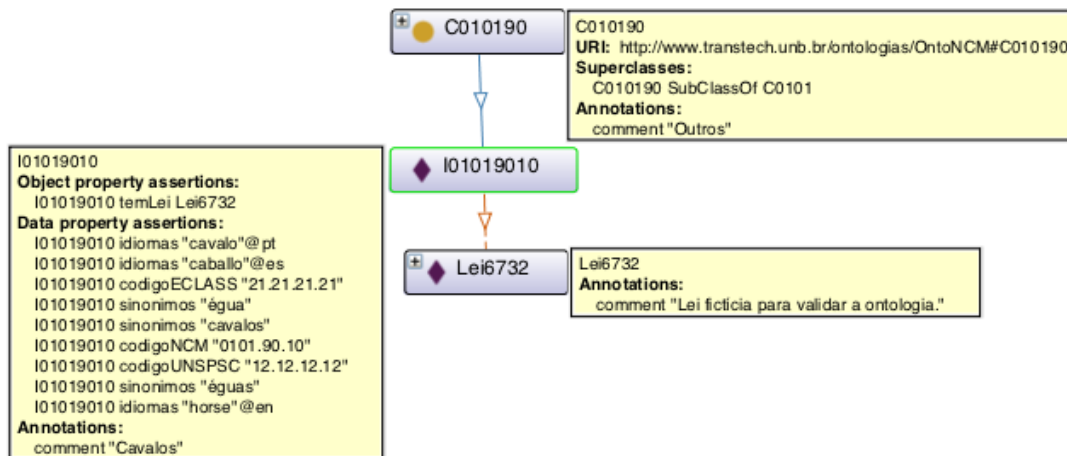


Figura 6.2: Resultado da busca 01019010 na aba *OntoGraf* do Protégé.

Observa-se que o produto 0101.90.10 é um Cavalo, que a Lei 6732 (fictícia) está relacionada a ele, que os sinônimos são “égua”, “éguas” e “cavalos”, que o produto em inglês é *horse*, em espanhol é *caballo*, tem o código eClass 21.21.21.21 (fictício) e, por fim, tem o código UNSPSC 12.12.12.12 (fictício).

Agora deseja-se comercializar galos, e o produtor entra no sistema com o intuito de descobrir o código. Por meio da aba *OntoGraf* do Protégé, poder-se-ia pesquisar por galo, e o resultado seria o exibido na Figura 6.3.

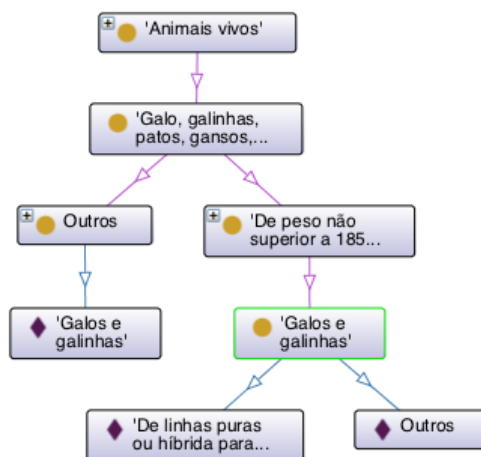


Figura 6.3: Resultado da busca do termo “galo” na aba *OntoGraf* do Protégé.

Ao se passar o ponteiro sobre todas as possibilidades, uma tela de dica se abre mostrando todas as propriedades de cada objeto, e assim o produtor identificou que o galo dele pesa mais de 185 g, então deve ser classificado com o código 0105.94.00. Outra observação

clara para o produtor é que não existe nenhuma lei relacionada a este produto, bem como não existem restrições de importação ou exportação, conforme ilustrado na Figura 6.4.

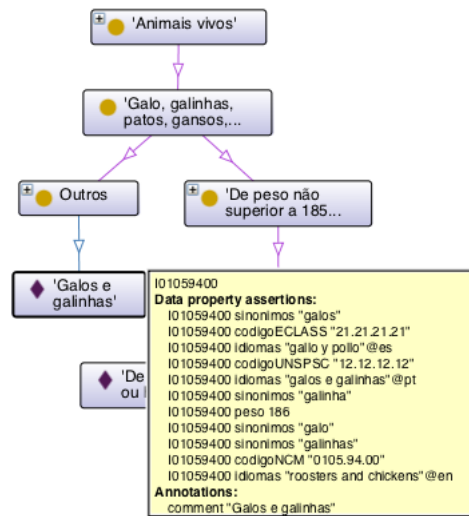


Figura 6.4: Resultado da busca do termo “galo” na aba *OntoGraf* do Protégé com dica ativada.

Da mesma forma que a pesquisa anterior, pela tela de dicas é possível saber as seguintes informações: código NCM 0105.94.00; código eClass 21.21.21.21 (fictício); código UNSPSC 12.12.12.12 (fictício); sinônimos “galos”, “galinhas”, “galinha”; idioma português *galos e galinhas*; idioma inglês *roosters and chickens*; idioma espanhol *gallo y pollo*; não tem leis relacionadas e, por fim, não tem qualquer restrição de exportação ou importação.

Outra forma de elaborar consultas no Protégé é utilizando a aba *DL Query*. Por exemplo, para saber quais produtos tem alguma restrição de importação para o Brasil deve-se digitar a *DL Query* “temRestricaoDeImportacao min 1”. Obtém-se como resultado, conforme Figura 6.5, que os produtos de códigos 0106.12.00 – Baleias, golfinhos, marsuínos, peixes-boi e dugongos – e 8471.90.90 – outros dispositivos computacionais de leituras que não estejam citados na posição 8471.90 – não podem ser importados para o Brasil, de qualquer parte do mundo.

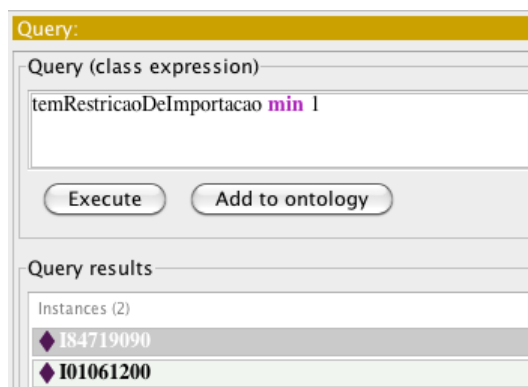


Figura 6.5: Resultado da consulta *DL Query* “temRestricaoDeImportacao min 1”

Suponha que agora deseja-se saber quais produtos não podem ser exportados para a Alemanha. A *DL Query* para tal busca é “temRestricaoDeExportacao value Alemanha” e o resultado pode ser visto na Figura 6.6. Os resultados são todos referentes a unidade de processamento de pequena, média e grande capacidade – códigos 8471.50.10, 8471.50.20 e 84.71.50.30 respectivamente.

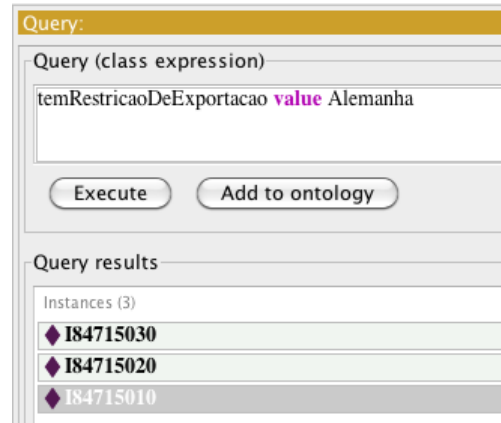


Figura 6.6: Resultado da consulta *DL Query* “temRestricaoDeExportacao value Alemanha”

6.1.2 Pelo TopBraid Composer

O TopBraid Composer [74] tem muitas funcionalidades para a busca em ontologias utilizando a linguagem SPARQL, vista na Seção 3.2.1.

Suponha que o mantenedor da ontologia desejasse buscar todos os produtos que não estavam devidamente traduzidos para outras línguas, e assim, traduzi-los. O código em SPARQL para buscar todos os produtos, em todos os idiomas é:

```
prefix OntoNCM: <http://www.transtech.unb.br/ontologias/OntoNCM#> 1
SELECT * WHERE { 2
  ?x OntoNCM:idiomas ?z 3
} 4
```

O resultado é longo, e uma parte deste é exibida na Figura 6.7 . Ao se selecionar um dos itens, e pedir a criação do gráfico, o TopBraid Composer gera a árvore gráfica do produto selecionado automaticamente, o que ajuda na visualização gráfica das consultas, conforme ilustra a Figura 6.8.

Um norte americano deseja comercializar cavalos no Brasil, e para tanto fez uma consulta no OntoNCM para ver o código do produto desejado. A seguinte SPARQL é fornecida ao sistema:

```
prefix OntoNCM: <http://www.transtech.unb.br/ontologias/OntoNCM#> 1
SELECT ?Nome ?NCM ?UNSPSC ?eCLASS WHERE { 2
  ?produto OntoNCM:idiomas "horse"@en . 3
  ?produto OntoNCM:codigoNCM ?NCM . 4
  ?produto OntoNCM:codigoUNSPSC ?UNSPSC . 5
  ?produto OntoNCM:codigoECLASS ?eCLASS . 6
  ?produto rdfs:comment ?Nome . 7
} 8
```

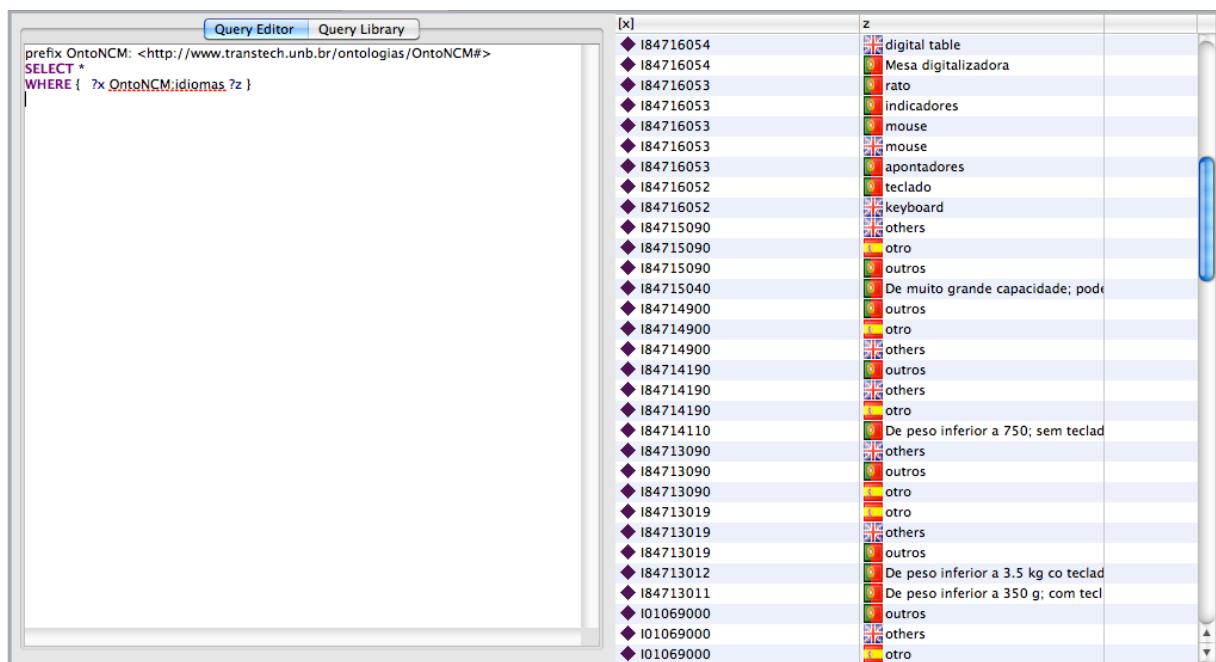



Figura 6.7: Tela com resultado de consulta no TopBraid Composer.

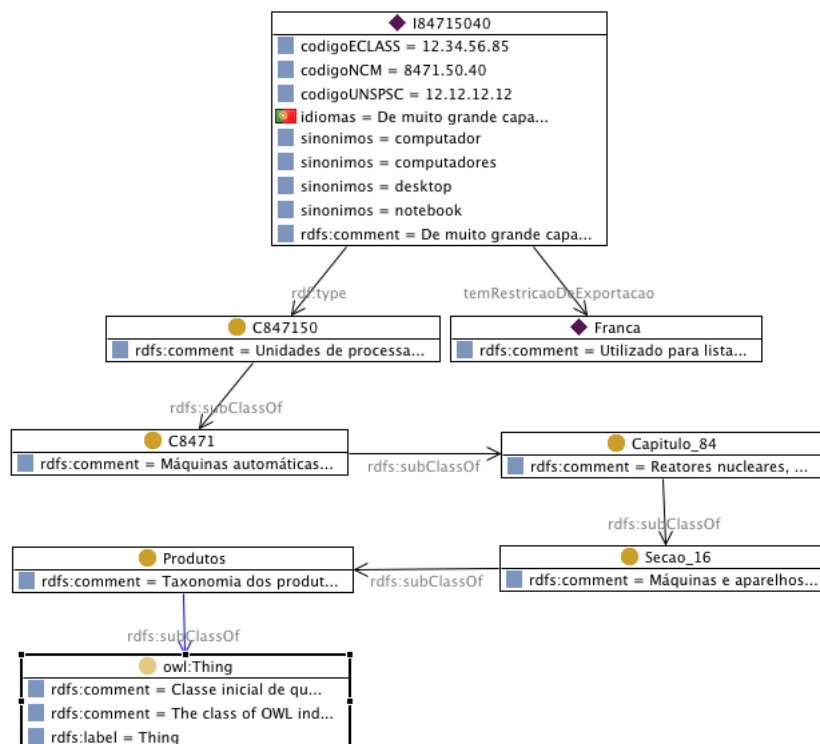
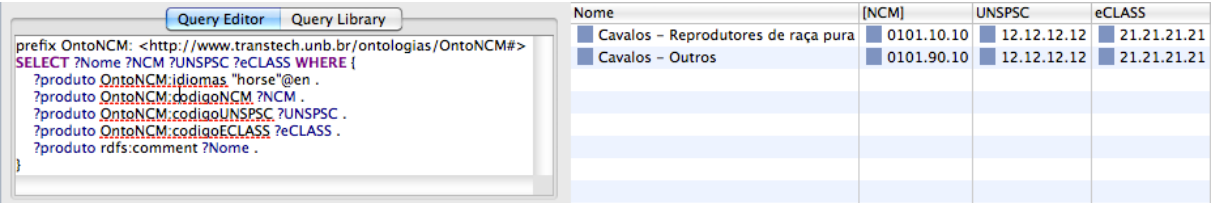


Figura 6.8: Resultado de consulta gráfico no TopBraid Composer.

Como resposta, tem-se o resultado mostrado na Figura 6.9.



Nome	[NCM]	UNSPSC	eCLASS
Cavalos - Reprodutores de raça pura	0101.10.10	12.12.12.12	21.21.21.21
Cavalos - Outros	0101.90.10	12.12.12.12	21.21.21.21

Figura 6.9: Resultado de consulta SPARQL 1.

Assim, o norte americano tem que decidir se o cavalo é um reprodutor de raça pura ou outro tipo de cavalo. Supondo que seja um reprodutor de raça pura, uma nova consulta pode ser feita para verificar as restrições de exportação deste produto para o Brasil, bem como as leis vigentes relacionadas. Para verificar restrições de exportação ele monta a consulta SPARQL:

```
prefix OntoNCM: <http://www.transtech.unb.br/ontologias/OntoNCM#>
SELECT * WHERE {
  ?produto OntoNCM:codigoNCM "0101.10.10" .
  ?produto OntoNCM:temRestricaoDeExportacao ?restricaoExportacao .
}
```

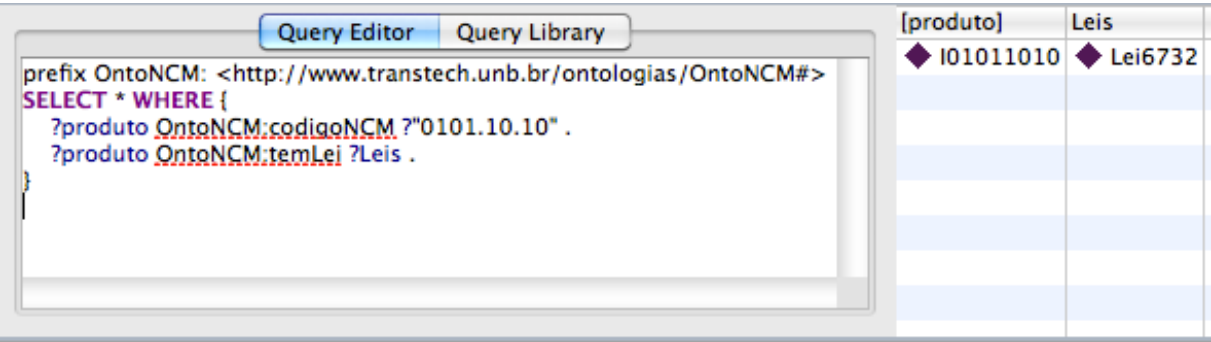
1
2
3
4
5

A consulta não retorna qualquer resultado, o que significa que não existe nenhuma restrição de exportação deste produto dos EUA para o Brasil. A consulta final é ver as leis correlatas ao produtos e, para isto, o norte americano efetua a seguinte SPARQL:

```
prefix OntoNCM: <http://www.transtech.unb.br/ontologias/OntoNCM#>
SELECT * WHERE {
  ?produto OntoNCM:codigoNCM "0101.10.10" .
  ?produto OntoNCM:temLei ?Leis .
}
```

1
2
3
4
5

E obtém a saída ilustrada na Figura 6.10, indicado que existe a Lei 6732 (fictícia) relacionada ao produto que deseja exportar para o Brasil.



[produto]	Leis
◆ I01011010	◆ Lei6732

Figura 6.10: Resultado de consulta SPARQL 2.

É notório que para montar tais consultas o norteamericano em questão é um especialista na linguagem SPARQL e, para usuários leigos, esta consulta deve ser facilitada por meio de um sistema *online*.

Suponha agora que um fiscal resolvesse consultar todas as leis no sistema que estão relacionadas com algum produto. Este poderia entrar com a SPARQL:

```

prefix OntoNCM: <http://www.transtech.unb.br/ontologias/OntoNCM#>
SELECT ?NCM ?Descricao ?Leis WHERE {
  ?produto OntoNCM:temLei ?Leis .
  ?produto OntoNCM:codigoNCM ?NCM .
  ?produto rdfs:comment ?Descricao .
}

```

O resultado é apresentado na Figura 6.11.

[NCM]	Descrição	Leis
0101.10.10	Cavalos – Reprodutores de raça pura	Lei6732
0101.90.10	Cavalos – Outros	Lei6732
0106.12.00	Baleias, golfinhos e marsuínos (mamí	Lei8745
0106.12.00	Baleias, golfinhos e marsuínos (mamí	Decreto653
0106.39.10	Avestruz (Struthio camelus) para repr	Lei9876
8471.50.90	Outros	Lei6732
8471.90.90	Outros	Decreto653

Figura 6.11: Resultado de consulta SPARQL 3.

Por ser o TopBraid Composer a ferramenta mais completa encontrada para a edição e consulta de ontologia, esta será usado para validar o modelo ontológico, e assim responder às perguntas propostas na Seção 5.1:

6.2 Validação do modelo

Para validar o modelo as perguntas do Capítulo 5, Seção 5.1 serão respondidas, por meio da ferramenta “TopBraid Composer” [74] com consultas SPARQL. Primeiro apresenta-se a pergunta, em seguida o código SPARQL equivalente a pergunta e, por fim, os resultados com análise.

A: Quais produtos brasileiros não podem ser exportados para os Estados Unidos da América?

```

prefix OntoNCM: <http://www.transtech.unb.br/ontologias/OntoNCM#>
SELECT ?NCM ?Descricao WHERE {
  ?produto OntoNCM:temRestricaoDeExportacao "EUA" ;
    OntoNCM:codigoNCM ?NCM ;
    rdfs:comment ?Descricao .
UNION
SELECT ?NCM ?Descricao WHERE {
  ?produto OntoNCM:temRestricaoDeExportacao "Mundo" ;
    OntoNCM:codigoNCM ?NCM ;
    rdfs:comment ?Descricao .
}

```

Saída

8471.50.20 Unidade de processamento de média capacidade, podendo conter no máximo uma unidade de entrada e outra de saída da subposição 8471.60, com capacidade de instalação, dentro do mesmo gabinete, de unidades de memória da subposição 8471.70, podendo conter múltiplos conectores

de expansão (*slots*) e valor FOB superior a US\$ 12.500,00 e inferior ou igual a US\$ 46.000,00 por unidade

0106.39.10 Avestruz (*Struthio camelus*) para reprodução

8471.50.10 Unidade de processamento de pequena capacidade, baseadas em micro-processadores, com capacidade de instalação, dentro do mesmo gabinete de unidades de memória da subposição 8471.60, podendo conter múltiplos conectores de expansão (*slots*) e valor FOB inferior ou igual a US\$ 12.500,00 por unidade

Análise

De fato estes são os únicos três produtos que não podem ser exportados (informação fictícia) para os Estados Unidos da América. Deve-se observar que foi feita uma união entre o que não pode ser exportado exclusivamente para os Estados Unidos da América e para o Mundo.

B: Quais produtos da Alemanha não podem ser importados para o Brasil?

```
prefix OntoNCM: <http://www.transtech.unb.br/ontologias/OntoNCM#>
SELECT ?NCM ?Descricao WHERE {
  ?produto OntoNCM:temRestricaoDeImportacao "Alemanha" ;
  OntoNCM:codigoNCM ?NCM ;
  rdfs:comment ?Descricao .
}
```

Saída

<vazio>

Análise

De fato não existe cadastrado na OntoNCM nenhum produto que não possa ser importado da Alemanha para o Brasil.

C: Qual o código NCM do produto Cavalo? E suas restrições?

```
prefix OntoNCM: <http://www.transtech.unb.br/ontologias/OntoNCM#>
SELECT ?NCM ?Descrição ?RestriçãoImportação ?RestriçãoExportação ?Lei WHERE {
  ?Produto OntoNCM:sinonimos "cavalos" .
  ?Produto OntoNCM:codigoNCM ?NCM .
  ?Produto rdfs:comment ?Descrição .
  OPTIONAL { ?Produto OntoNCM:temRestricaoDeImportacao ?RestriçãoImportação } .
  OPTIONAL { ?Produto OntoNCM:temRestricaoDeExportacao ?RestriçãoExportação } .
  OPTIONAL { ?Produto OntoNCM:temLei ?Lei } .
}
```

Saída

[NCM]	Descrição	RestriçãoImportação	RestriçãoExportação	Lei
0101.10.10	Cavalos – Reprodutores de raça pura			Lei6732
0101.90.10	Cavalos – Outros			Lei6732

Análise

De fato existem cavalos na base de dados, e estes estão ligados à Lei 6732 (fictícia). Observa-se ainda que não existe qualquer restrição, tanto de importação quando de exportação, deste produto para o Brasil.

D: Quais os sinônimos do produto “Notebook”?

```
prefix OntoNCM: <http://www.transtech.unb.br/ontologias/OntoNCM#>
SELECT DISTINCT ?Sinonimos WHERE {
    ?Produto OntoNCM:sinonimos "notebook" .
    ?Produto OntoNCM:sinonimos ?Sinonimos .
}
```

Saída

[Sinonimos]
computador
computadores
desktop
notebook

Análise

De fato os sinônimos para *notebook* cadastrado são os exibidos. Pode parecer um erro ter o termo *desktop* como sinônimo de *notebook*, mas a intenção de colocar tal sinônimo foi situar a pessoa que pesquisa em produtos próximos do que se está procurando.

E: Qual o código do produto “égua” na NCM?

```
prefix OntoNCM: <http://www.transtech.unb.br/ontologias/OntoNCM#>
SELECT DISTINCT ?NCM ?Descrição WHERE {
    ?Produto OntoNCM:sinonimos "égua" .
    ?Produto OntoNCM:codigoNCM ?NCM .
    ?Produto rdfs:comment ?Descrição .
}
```

Saída

[NCM]	Descrição
0101.10.10	Cavalos – Reprodutores de raça pura
0101.90.10	Cavalos – Outros

Análise

De fato estes são os produtos que têm o sinônimo *égua* cadastrados na ontologia OntoNCM. No caso, quem deseja descobrir o código NCM do produto terá que fazer uma análise se a *égua* é uma reprodutora de raça pura ou não, pois isto define o código NCM correto a ser adotado.

F: Quais leis estão ligadas ao produto “baleias”?

```
prefix OntoNCM: <http://www.transtech.unb.br/ontologias/OntoNCM#>
SELECT DISTINCT ?NCM ?Descrição ?Leis WHERE {
  ?Produto OntoNCM:codigoNCM ?NCM .
  ?Produto OntoNCM:sinonimos "baleias" .
  ?Produto OntoNCM:temLei ?Leis .
  ?Produto rdfs:comment ?Descrição .
}
```

1
2
3
4
5
6
7

Saída

[NCM]	Descrição	Leis
0106.12.00	Baleias, golfinhos e marsuínos (mamíferos da ordem dos cetáceos)	Decreto653
0106.12.00	Baleias, golfinhos e marsuínos (mamíferos da ordem dos cetáceos)	Lei8745

Análise

De fato estas são as leis relacionadas aos produtos baleias: Decreto653 e Lei8745 (ambos fictícios). Em um sistema próprio seria possível evitar a repetição do código, mostrando ao usuário as leis como uma única saída.

G: O que eu devo considerar ao exportar um terminal bancário para os EUA?

```
prefix OntoNCM: <http://www.transtech.unb.br/ontologias/OntoNCM#>
SELECT DISTINCT ?NCM ?Descrição ?RestriçãoExportação ?Leis WHERE {
  ?Produto OntoNCM:codigoNCM ?NCM .
  ?Produto OntoNCM:sinonimos "token" .

  OPTIONAL { ?Produto OntoNCM:temLei ?Leis } .
  OPTIONAL { ?Produto OntoNCM:temRestricaoDeExportacao ?RestriçãoExportação } .
  ?Produto rdfs:comment ?Descrição .
}
```

1
2
3
4
5
6
7
8
9

Saída

[NCM]	Descrição	RestriçãoExportação	Leis
8471.60.80	Terminais de autoatendimento	EUA	

Análise

De fato existe uma restrição à exportação de terminais de atendimentos bancários para os Estados Unidos da América (fictício). No entanto deve-se observar que não existe nenhuma lei relacionada, o que diminui a credibilidade da informação. Neste caso um especialista em manter a ontologia deveria revisá-la e incluir a lei que restringe tal exportação.

H: Qual é a correta classificação do produto “galo”?

```
prefix OntoNCM: <http://www.transtech.unb.br/ontologias/OntoNCM#>
SELECT DISTINCT ?NCM ?Descrição WHERE {
  ?Produto OntoNCM:codigoNCM ?NCM .
  ?Produto OntoNCM:sinonimos "galo" .
  ?Produto rdfs:comment ?Descrição .
}
```

1
2
3
4
5
6

Saída

[NCM]	Descrição
0105.11.10	Galos e galinhas – De linhas puras ou híbrida para reprodução
0105.11.90	Outros – De linhas puras ou híbridas para consumo
0105.94.00	Galos e galinhas – genéricas para consumo

Análise

Neste caso a correta codificação depende de uma análise do produto que se deseja comercializar. Se for um galo de linha pura ou híbrida para reprodução, o código é 0105.11.10; se for um galo de linha pura ou híbrida para consumo, o código é 0105.11.90; se for um galo genérico para consumo, o código correto é 0105.94.00.

I : Quais produtos não podem ser exportados para a China?

```
prefix OntoNCM: <http://www.transtech.unb.br/ontologias/OntoNCM#>
SELECT DISTINCT ?NCM ?Descrição WHERE {
  ?Produto OntoNCM:codigoNCM ?NCM .
  ?Produto OntoNCM:temRestricaoDeExportação "Mundo" .
  ?Produto rdfs:comment ?Descrição .
}
```

1
2
3
4
5
6

Saída

[NCM]	Descrição	RestriçãoExportação
0106.39.10	Avestruz (<i>Struthio camelus</i>) para reprodução	◆ Mundo

Análise

Como o país China não está cadastrado diretamente na ontologia, a referência de pesquisa foi o que o Brasil não pode exportar para o mundo. Como resultado temos o produto de código 0106.39.10 – Avestruz.

J: Qual o código do produto “Computador” no código UNSPSC? E no e-Class?

```
prefix OntoNCM: <http://www.transtech.unb.br/ontologias/OntoNCM#>
SELECT DISTINCT ?NCM ?UNSPSC ?eClass ?Descrição WHERE {
  ?Produto OntoNCM:sinonimos "computador" .
  ?Produto OntoNCM:codigoNCM ?NCM .
  ?Produto OntoNCM:codigoUNSPSC ?UNSPSC .
  ?Produto OntoNCM:codigoECLASS ?eClass .
  ?Produto rdfs:comment ?Descrição .
}
```

1
2
3
4
5
6
7
8

Saída

[NCM]	UNSPSC	eClass	Descrição
8471.30.11	12.12.12.12	12.34.56.81	De peso inferior a 350 g, com teclado alfanumérico de no mínimo 70 teclas e com uma
8471.30.12	12.12.12.12	12.34.56.56	De peso inferior a 3,5 kg co teclado alfanumerico de no mínimo 70 teclas e com uma tel
8471.41.10	12.12.12.12	12.34.56.83	De peso inferior a 750; sem teclado; entrada de dados e de comandos por meio de uma
8471.50.10	12.12.12.12	12.34.56.89	De pequena capacidade, baeadas em microprocessadores, com capacidade de instalaçã
8471.50.20	12.12.12.12	12.34.56.88	Unidade de processamento de média capacidade, podendo conter no máximo uma unid
8471.50.30	12.12.12.12	12.34.56.87	De grande capacidade, podendo conter no máximo uma unidade de entrada e outra de :
8471.50.40	12.12.12.12	12.34.56.85	De muito grande capacidade, podendo conter no máximo uma unidade de entrada e out
8471.80.00	12.12.12.12	12.34.56.78	Outras unidades de máquinas automáticas para processamento de dados

O resultado trouxe vários itens que fazem referência a computadores, com os códigos NCM, UNSPSC, eClass e uma descrição. Os códigos UNSPSC e eClass são fictícios, dado que buscar a equivalência correta neste trabalho foi o ponto principal.

6.3 Código de programação para acesso à ontologia

Um código simples para acesso da ontologia no Java foi produzido com auxílio da biblioteca Jena [35], um *framework* do Java criado para a construção de aplicações para uso na web semântica. O Jena oferece um ambiente para a utilização de recurso RDF, RDFS e OWL, com suporte a consultas SPARQL.

Todas as SPARQL exibidas nas seções anteriores deste capítulo podem ser embutidas em um aplicativo, de forma que pessoas leigas consigam efetuar consultas na ontologia. Uma classe básica para acessar a ontologia é listada a seguir.

```

package br.cic.unb.ncm;
import com.hp.hpl.jena.ontology.OntModel;
import com.hp.hpl.jena.ontology.OntModelSpec;
import com.hp.hpl.jena.rdf.model.ModelFactory;
import com.hp.hpl.jena.rdf.model.Property;
import com.hp.hpl.jena.rdf.model.RDFNode;
import com.hp.hpl.jena.rdf.model.Resource;
import com.hp.hpl.jena.rdf.model.Statement;
import com.hp.hpl.jena.rdf.model.StmtIterator;

public class NcmOntology {
    public static void main(String[] args) {
        OntModel model = ModelFactory.createOntologyModel(
            OntModelSpec.OWL_MEM, null);
        model.read("file:/Users/thyarles/Documents/workspace/ncmOntologyWith/OntoNCM.owl", "
            RDF/XML-ABBREV");
        StmtIterator iter = model.listStatements();
        while(iter.hasNext())
            Statement stmt      = iter.nextStatement();
            Resource  subject    = stmt.getSubject();
            Property  predicate   = stmt.getPredicate();
            RDFNode   object     = stmt.getObject();
            System.out.print(subject.toString());
            System.out.print(" " + predicate.toString() + " ");
            if (object instanceof Resource) {
                System.out.print(object.toString());
            } else {
                System.out.print(" \" " + object.toString() + "\" ");
            }
            System.out.println(" ");
        }
    }
}

```

A Figura 6.12 ilustra este código em execução, no ambiente de desenvolvimento Eclipse.

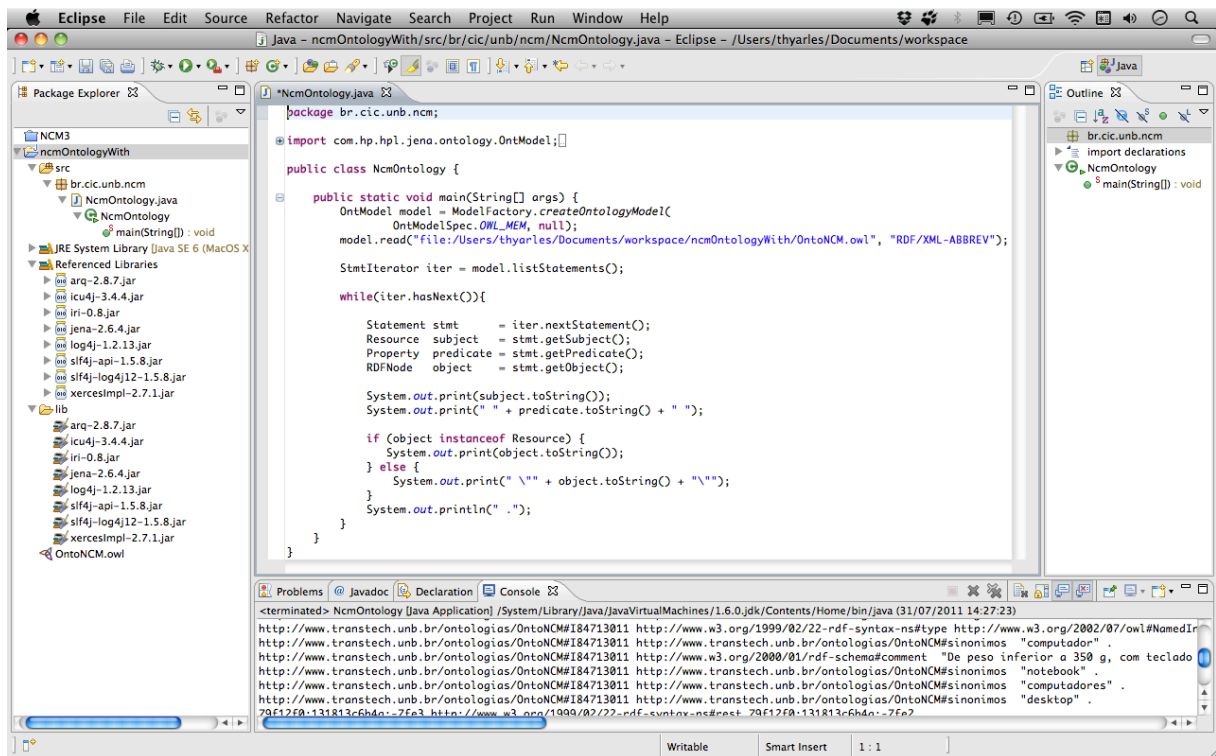


Figura 6.12: Classe de acesso ao OntoNCM em Jena, no ambiente Eclipse.

6.4 Análise dos resultados

O modelo de ontologia foi proposto levando em consideração o “Método 101” [47]. Este método sugere uma série de passos a serem seguidos de forma iterativa e, como ponto de partida, a elaboração de questões pendentes de solução. As questões propostas tiveram respostas positivas na avaliação do modelo, o que mostrou que a ontologia atendeu aos requisitos.

O modelo ontológico foi produzido no Protégé [61], uma ferramenta gratuita e com muitos recursos. A validação foi feita tanto no Protégé quanto no TopBraid Composer [74], o que demonstrou que a ontologia produzida em um ambiente é totalmente compatível em outro ambiente.

A classe em Java, utilizando a biblioteca Jena [35], demonstrou o quanto é simples produzir um aplicativo para o acesso ao modelo ontológico proposto. Apesar de o aplicativo ter apenas aberto a ontologia e listado todas as entradas desta, existem métodos prontos para a entrada de dados, consulta e extração de dados por meio da biblioteca utilizada.

Os resultados das consultas feitas, tanto em *DL Query* quanto em SPARQL produziram resultados satisfatórios, rápidos e objetivos. Desta forma, pode-se concluir que esta ontologia – com os devidos aperfeiçoamentos – pode ser utilizada para transações e classificações no comércio eletrônico.

Capítulo 7

Conclusão

A Nomenclatura Comum do Mercosul (NCM) apresenta descrições complexas para cada item, ao pé que as nomenclaturas UNSPSC e eClass apresentam descrições sucintas e objetivas. Enquanto estas têm termos modernos e objetivos, que refletem os produtos utilizados atualmente, aquela – NCM – reflete até mesmo produtos que não são mais comercializados e deixa de incluir termos modernos, tais como os *pendrivers* ou *tablets*. Outro problema na NCM é a carência de objetividade para a descrição dos códigos, já que um item pode corresponder a vários produtos diferentes, o que deixa margem a dúvidas para quem os classifica.

A falta de clareza da NCM foi minimizada por meio da ontologia proposta, já que esta fez relações dos itens cadastrados na nomenclatura com termos simples e conhecidos por todos. O modelo proposto previu um local para armazenar os códigos UNSPSC e eClass, o que permitirá a localização de determinado item pelas três nomenclaturas estudadas.

Quanto às ferramentas utilizadas, concluiu-se que a utilização do editor de ontologia Protégé facilitou o processo de criação da ontologia, por tratar-se de uma ferramenta completa, que permite desde a criação até a validação do modelo pela geração de inferências e consultas simples como a *DL Query*. Contudo, a parte de consultas avançadas, por meio da utilização da linguagem SPARQL, não funcionou adequadamente no Protégé e, por isto, a ferramenta TopBraid Composer foi utilizada com sucesso nesta função.

7.1 Avaliação do modelo proposto

O objetivo deste trabalho foi atingido, já que o modelo proposto – baseado em uma parte da taxonomia NCM – otimizou com sucesso algumas lacunas na pesquisa de mercadorias, tais como possibilitar a busca de termos por sinônimos, termos em outras línguas, restrições na comercialização de determinados produtos ou leis imprescindíveis para a comercialização legal de alguns produtos. O modelo ontológico permite que outros domínios – que sejam classificados como importantes e não pensados neste trabalho – possam ser facilmente incluídos na proposta ontológica.

7.2 Dificuldades observadas no projeto

A falta de termos modernos dificultou o entendimento da Nomenclatura Comum do Mercosul, e por isto alguns produtos de grande aceitação e conhecimento no mercado foram classificados de forma genérica. Em várias tentativas de classificação de produtos, em grande parte referentes à tecnologia, desde *pendrivers* até *tablets*, estes foram classificados como “outros”, o que pareceu demasiadamente genérico para produtos de uso comum.

7.3 Contribuições e trabalhos futuros

7.3.1 Contribuições

Este trabalho foi de especial relevância, uma vez que não se pode duvidar que propor mudanças em normas desta magnitude não se afigura um trabalho simples, tampouco rápido. Assim, um modelo ontológico é capaz de cobrir as lacunas existentes, bem como fornecer sinônimos que complementam o correto entendimento da NCM. Dessa forma, a vantagem da utilização de uma ontologia está no fato de possibilitar que cada área de mercado possa especializar uma parte da NCM, fornecendo posteriormente tal informação, de modo que outras áreas façam uso, bem como as complemente.

Na área da mecatrônica este estudo é norteador, no sentido de que fornece ideias para se classificar e correlacionar peças de sistemas mecânicos de forma semelhante, tal como Zdravkovic and Trajanovic [80] já o fizeram em pesquisa similar, com uma ontologia para integração de produtos organizacionais.

Um outro uso interessante da metodologia proposta é para a manutenção de um sistema antigo, que tenha que manter um legado. Com ontologia este pode ser reescrito, de forma que atenda, ao mesmo tempo, os conceitos novos e antigos.

Outra utilidade deste estudo para a mecatrônica seria a catalogação de compatibilidade de peças de reposição; estas poderiam ser ligadas aos dispositivos compatíveis e uma busca simples poderia retornar resultados em pesquisas por torque, impedância e resistência.

7.3.2 Trabalhos futuros

1. Complementação da ontologia proposta, adicionando informações de relevância – tais como taxas de exportação, taxas de importação, valores das mercadorias por países, altas e baixas de safras e tarifas aduaneiras – e catalogando completamente a NCM na ontologia, com a correta correlação dos códigos desta com os UNSPSC e eClass.
2. Desenvolvimento de um sistema – em Java ou outra linguagem que opere na internet – que faça uso da ontologia de forma efetiva, com uma interface amigável e em vários idiomas para uso em todo o mundo.

3. Proposta de um mapeamento automático da ontologia proposta com outras ontologias, como a UNSPSC, eClass e RosettaNet.

Capítulo 8

Apêndices

8.1 Regras e roteiro para interpretação e classificação do SH

A interpretação e classificação das mercadorias devem ser efetuadas de acordo com alguns critérios, que deverão ser rigorosamente observados para a correta determinação do respectivo código. As regras, o roteiro e o sumário de seções e capítulos para essa classificação existe sob os títulos “Regras Gerais para Interpretação do Sistema Harmonizado”, “Roteiro para Efetuar uma Classificação” e “Sumário das Seções e Capítulos”, respectivamente [1, 44]. Ressalta-se que tais regras e roteiros, em conjunto com a Instrução Normativa RFB n 740/07¹, são utilizados pelo MDIC e pela Secretaria da Receita Federal do Brasil (RFB) para a correta classificação de mercadorias; por isso foram utilizados como base para entender a classificação da NCM neste trabalho. As seções seguintes deste capítulo têm base no material produzido pela Equipe Técnica Aduandeiras [1].

8.1.1 Regras Gerais para Interpretação do Sistema Harmonizado

A classificação das mercadorias na Nomenclatura rege-se pelas seguintes regras:

1. Os títulos das Seções, Capítulos e Subcapítulos têm apenas valor indicativo. Para os efeitos legais, a classificação é determinada pelos textos das posições e das Notas de Seção e de Capítulo e, desde que não sejam contrárias aos textos das referidas posições e Notas, pelas Regras seguintes:
 - (a) Qualquer referência a um artigo em determinada posição abrange esse artigo mesmo incompleto ou inacabado, desde que apresente, no estado em que se encontra, as características essenciais do artigo completo ou acabado. Abrange

¹Disponível para consulta em <http://www.receita.fazenda.gov.br/legislacao/ins/2007/in7402007.htm>, acessado em 3 de outubro de 2010.

igualmente o artigo completo ou acabado, ou como tal considerado nos termos das disposições precedentes, mesmo que se apresente desmontado ou por montar.

- (b) Qualquer referência a uma matéria em determinada posição diz respeito a essa matéria, quer em estado puro, quer misturada ou associada a outras matérias. Da mesma forma, qualquer referência a obras de uma matéria determinada abrange as obras constituídas inteira ou parcialmente por essa matéria. A classificação destes produtos misturados ou artigos compostos efetua-se conforme os princípios enunciados na Regra 3.
2. Quando parecer que a mercadoria pode classificar-se em duas ou mais posições por aplicação da Regra 1 b) ou por qualquer outra razão, a classificação deve efetuar-se da forma seguinte:
- (a) A posição mais específica prevalece sobre as mais genéricas. Todavia, quando duas ou mais posições se refiram, cada uma delas, a apenas uma parte das matérias constitutivas de um produto misturado ou de um artigo composto, ou a apenas um dos componentes de sortidos acondicionados para venda a retalho, tais posições devem considerar-se, em relação a esses produtos ou artigos, como igualmente específicas, ainda que uma delas apresente uma descrição mais precisa ou completa da mercadoria.
 - (b) Os produtos misturados, as obras compostas de matérias diferentes ou constituídas pela reunião de artigos diferentes e as mercadorias apresentadas em sortidos acondicionados para venda a retalho, cuja classificação não se possa efetuar pela aplicação da Regra 2 a), classificam-se pela matéria ou artigo que lhes confira a característica essencial, quando for possível realizar esta determinação.
 - (c) Nos casos em que as Regras 2 a) e 2 b) não permitam efetuar a classificação, a mercadoria classifica-se na posição situada em último lugar na ordem numérica, dentre as suscetíveis de validamente se tomarem em consideração.
3. As mercadorias que não possam ser classificadas por aplicação das Regras acima enunciadas classificam-se na posição correspondente aos artigos mais semelhantes.
4. Além das disposições precedentes, as mercadorias abaixo mencionadas estão sujeitas às Regras seguintes:
- (a) Os estojos para aparelhos fotográficos, para instrumentos musicais, para armas, para instrumentos de desenho, para jóias e receptáculos semelhantes, especialmente fabricados para conterem um artigo determinado ou um sortido, e suscetíveis de um uso prolongado, quando apresentados com os artigos a que

se destinam, classificam-se com estes últimos, desde que sejam do tipo normalmente vendido com tais artigos. Esta Regra, todavia, não diz respeito aos receptáculos que confirmam ao conjunto a sua característica essencial.

(b) Sem prejuízo do disposto na Regra 4 a), as embalagens contendo mercadorias classificam-se com estas últimas quando sejam do tipo normalmente utilizado para o seu acondicionamento. Todavia, esta disposição não é obrigatória quando as embalagens sejam claramente suscetíveis de utilização repetida.

5. A classificação de mercadorias nas subposições de uma mesma posição é determinada, para efeitos legais, pelos textos dessas subposições e das Notas de Subposição respectivas, assim como, *mutatis mutandis*, pelas Regras precedentes, entendendo-se que apenas são comparáveis subposições do mesmo nível. Para os fins da presente Regra, as Notas de Seção e de Capítulo são também aplicáveis, salvo disposições em contrário.

8.1.1.1 Regras Gerais Complementares (RGC)

1. (RGC-1) As Regras Gerais para Interpretação do Sistema Harmonizado se aplicarão, *mutatis mutandis*, para determinar dentro de cada posição ou subposição, o item aplicável e, dentro deste último, o subitem correspondente, entendendo-se que apenas são comparáveis desdobramentos regionais (itens e subitens) do mesmo nível.
2. (RGC-2) As embalagens contendo mercadorias e que sejam claramente suscetíveis de utilização repetida, mencionadas na Regra 5 b), seguirão seu próprio regime de classificação sempre que estejam submetidas aos regimes aduaneiros especiais de admissão temporária ou de exportação temporária. Caso contrário, seguirão o regime de classificação das mercadorias.

8.1.1.2 Regra Geral Complementar da TIPI (RGC/TIPI)

1. (RGC/TIPI-1) As Regras Gerais para Interpretação do Sistema Harmonizado se aplicarão, *mutatis mutandis* para determinar, no âmbito de cada código, quando for o caso, o “Ex” aplicável, entendendo-se que apenas são comparáveis “Ex” de um mesmo código.

8.1.2 Roteiro para Efetuar uma Classificação

O primeiro passo para classificar uma mercadoria é conhecê-la em todas as suas características, pois um simples detalhe em sua constituição física, uma função diversa, entre outros, pode determinar outra classificação, ainda que não seja claramente apresentada. Conhecida a mercadoria, os demais passos para realizar a sua classificação estão descritos no roteiro abaixo e devem ser seguidos cuidadosamente:

8.1.2.1 Passo 1

As mercadorias são agrupadas, dentro da nomenclatura, em seções, capítulos e subcapítulos que recebem títulos genéricos indicando apenas a categoria dos produtos que se encontram ali classificados, tendo, portanto, apenas caráter indicativo.

A utilização dos textos das seções, capítulos e subcapítulos servem como ponto de partida para localizar a classificação de uma mercadoria.

Se utilizarmos como exemplo a Seção I, teremos “Animais vivos e produtos do reino animal”: Esta seção está dividida em capítulos, sendo o Capítulo 1 reservado para “Animais vivos”, o Capítulo 2 para “Carnes e miudezas comestíveis” e assim seguem outras divisões relacionadas ao texto da seção².

Imaginemos que a mercadoria objeto de classificação seja cavalos reprodutores de raça pura. Estará, então, na Seção I, Capítulo 1 (Animais vivos).

Identificando o ponto de referência (seção e capítulo) o que determina a classificação é o texto das posições, que irá definir a posição ocupada pela mercadoria e o texto das notas de seção e de capítulos.

Para compreender melhor a importância das notas de seção e capítulos pode ser tomada como exemplo a nota 1 c) do Capítulo 1 (Animais vivos): “O presente Capítulo compreende todos os animais vivos, exceto: a) [...]; b) [...]; c) animais da posição 9508”. A princípio, qualquer animal vivo pode ser classificado nesse capítulo. Entretanto, ao observar a Nota 1 c) encontram-se excetuados os animais vivos que compõem coleções de zoológicos, de circos ou ainda de outras atrações, que estão classificados no Capítulo 95 (Brinquedos, jogos, artigos para divertimento ou para esporte, suas partes e acessórios).

Também deve ser observado o texto das NESH, que estabelece o alcance e o conteúdo da Nomenclatura.

8.1.2.2 Passo 2

O alcance do texto das posições deve ser considerado para os artigos incompletos ou inacabados sempre que no estado em que se encontrem apresentem as características do bem completo ou acabado.

Desse modo, se o produto for uma bicicleta e não fizer parte dela o assento, ainda assim a classificação será a mesma da bicicleta, pois a falta do assento não fará com que a classificação seja alterada.

Se por outro lado a mesma bicicleta apresentar-se desmontada ou por montar classificar-se-á do mesmo modo, na posição do bem montado.

Considera-se como artigo desmontado ou por montar aquele cujos elementos destinam-se a ser montados utilizando-se parafusos, cavilhas etc., ou por soldagem ou qualquer outra simples operação de montagem.

Ampliando o alcance do texto das posições, os produtos misturados ou associados a outros artigos, desde que não confirmem ao produto uma característica distinta da definida

²Para uma visão global das Seções e dos Capítulos, consulte a seção 8.2 deste trabalho.

pela posição, classificam-se de acordo com os princípios enunciados pela Regra 3.

Deve-se notar que essa ampliação não pode contrariar as disposições da Regra 1.

8.1.2.3 Passo 3

Quando a classificação de uma mercadoria for suscetível de se incluir em duas ou mais posições deverão ser observados os critérios que seguem para o correto enquadramento do produto.

1. A posição mais específica deve prevalecer sobre as posições de um alcance geral. De modo geral uma posição pode ser considerada mais específica quando designar um artigo em particular ou identificar mais claramente a mercadoria. Exemplo: Pneus novos para automóveis classificam-se na posição 4011 (Pneumáticos novos de borracha) e não na posição 8708 (Partes e acessórios dos veículos automóveis ...). Porém, quando duas ou mais posições dizem respeito a um produto misturado ou artigo composto devem ser consideradas em relação a cada um como igualmente específicas. Dessa forma, a classificação deverá ser realizada com base no disposto no item que segue.
2. Quando a classificação da mercadoria não puder ser realizada com base na explicação do item 1 acima, deverá ser feita pela matéria ou artigo que confira ao produto a característica essencial. O fator que determina a característica essencial varia conforme o tipo de mercadoria, podendo ser determinado pela natureza da matéria ou de seus constituintes, pela quantidade, valor, etc. Esse conceito deve ser aplicado às mercadorias apresentadas em “sortidos”; assim considerados:
 - (a) a união de pelo menos dois artigos diferentes que à primeira vista seriam suscetíveis de se incluírem em posições diferentes;
 - (b) os artigos apresentados em conjunto para a satisfação de uma necessidade específica;
 - (c) artigos acondicionados para serem vendidos diretamente ao consumidor. Assim, em quaisquer das situações acima, a classificação será determinada de acordo com os artigos que confirmam ao conjunto o caráter essencial. A título de exemplificação, veja-se: Um conjunto de desenho, constituído por uma régua (posição 9017), um disco de cálculo (posição 9017), um compasso (posição 9017), um lápis (posição 9609) e um apontador (apara-lápis), apresentados em um estojo de folha de plástico (posição 4202) classifica-se na posição 9017.
3. Quando não for possível determinar a posição mais específica ou ainda a que confira a característica essencial, a classificação deverá seguir aquela da posição situada em último lugar na ordem numérica. Para exemplificar, vamos considerar um conjunto para escritório composto de caneta esferográfica (posição 9608), lápis comum (posição 9609), borracha de apagar (posição 4016) e apontador para lápis (posição 8214).

Nesse caso, não é possível determinar a posição mais específica, nem mesmo qual confere ao conjunto a característica essencial. Assim, a classificação será feita pela última posição em ordem numérica, no exemplo 9609.

8.1.2.4 Passo 4

Quando a classificação de uma mercadoria não puder ser realizada pela aplicação das regras anteriores, deverá ser considerada a posição correspondente aos artigos mais semelhantes.

Assim, poderão ser tomados por base elementos como denominação, características, utilização, etc. para a comparação das mercadorias.

8.1.2.5 Passo 5

A classificação de estojos e artefatos semelhantes deve seguir a classificação dos produtos a que se destinam quando apresentados com os mesmos e sendo específicos para conter determinado produto, como, por exemplo, os estojos para instrumentos musicais. Quando apresentados isoladamente seguem seu próprio regime.

Entretanto, não poderão ser classificados junto aos artigos a que se destinam os estojos e artefatos semelhantes que confirmam ao produto a sua característica essencial como, por exemplo, caixas de chá, de prata, contendo chá.

Do mesmo modo que os estojos, as embalagens que contenham mercadorias classificam-se com as mesmas sempre que forem do tipo normalmente utilizado para o seu acondicionamento. Porém, quando as embalagens forem de uso repetido, como por exemplo certos tambores metálicos destinados a conter gases, esta disposição não é obrigatória.

8.1.2.6 Passo 6

Definida a posição para a classificação nas subposições de uma determinada posição, devem ser considerados os textos dessas subposições e das notas de subposições sendo, ainda, aplicados os critérios anteriores para o correto enquadramento.

É importante destacar que apenas podem ser comparados os textos das subposições de mesmo nível.

8.1.2.7 Passo 7

Por último, para determinar a classificação de mercadorias nos itens ou subitens de uma posição ou subposição, devem ser considerados os textos desses itens ou subitens, bem como as regras explicadas anteriormente, podendo somente ser comparados os itens e/ou subitens de mesmo nível.

Vale ressaltar que a leitura das NESH é fundamental para o exato enquadramento das mercadorias em cada posição, uma vez que explica com detalhes o texto de cada uma – com exemplos em alguns casos –, o que facilita o entendimento do texto.

Como exemplo do conteúdo apresentado na NESH, pode-se apontar a posição 0410 que segue: “Produtos Comestíveis de origem animal, não especificados nem compreendidos em outras posições”.

A presente posição compreende os produtos de origem animal próprios para o consumo humano, não especificados nem compreendidos em outras posições da Nomenclatura.

Esta posição inclui, entre outros:

1. Os *ovos de tartaruga*. Esses ovos, que provêm de algumas espécies aquáticas (tartarugas marinhas ou de água doce), podem apresentar-se frescos, secos ou conservados de outro modo.

O óleo de ovos de tartaruga inclui-se na posição 1506;

2. Os *ninhos de salangana*, denominados impropriamente de “ninhos de andorinha”. Esses ninhos são constituídos por uma substância segregada pelo animal, e que se solidifica rapidamente em contato com o ar.

Podem se apresentar em bruto ou ter sofrido tratamentos destinados a desembaraçá-los da penas, penugem, poeiras e outras impurezas, de forma a torná-los próprios para consumo. Nesse estado encontram-se no comércio, em geral em tiras ou fios, de cor esbranquiçada.

Muito ricos em proteína, os ninhos de salangana são utilizados quase que exclusivamente em sopas ou em outras preparações alimentícias.

A presente posição não compreende o sangue animal, mesmo comestível, líquido ou dessecado (posições 0511 ou 3002).

8.2 Sumário das Seções e Capítulos

O sumário completo pode ser obtido *online*³.

Seção I Animais vivos e produtos do reino animal

Cap. 1 Animais vivos

Cap. 2 Carnes e miudezas, comestíveis

Cap. 3 Peixes e crustáceos, moluscos e os outros invertebrados aquáticos

Cap. 4 Leite e laticínios; ovos de aves; mel natural; produtos comestíveis de origem animal, não especificados nem compreendidos em outros Capítulos

Cap. 5 Outros produtos de origem animal, não especificados nem compreendidos em outros Capítulos

Seção II Produtos do reino vegetal

³<http://www.novuscontabilidade.com/sistema/ferramentas/TIPI%20Consolidado.pdf>, acessado em 27 de dezembro de 2010.

Cap. 6	Plantas vivas e produtos de floricultura
Cap. 7	Produtos hortícolas, plantas, raízes e tubérculos, comestíveis
Cap. 8	Frutas; cascas de cítricos e de melões
Cap. 9	Café, chá, mate e especiarias
Cap. 10	Cereais
Cap. 11	Produtos da indústria de moagem; malte; amidos e féculas; inulina; glúten de trigo
Cap. 12	Sementes e frutos oleaginosos; grãos, sementes e frutos diversos; plantas industriais ou medicinais; palha e forragem
Cap. 13	Gomas, resinas e outros sucos e extratos vegetais
Cap. 14	Matérias para entrançar e outros produtos de origem vegetal, não especificados nem compreendidos em outros Capítulos
Seção III	Gorduras e óleos animais ou vegetais; produtos da sua dissociação; gorduras alimentares elaboradas; ceras de origem animal ou vegetal
Cap. 15	Gorduras e óleos animais ou vegetais; produtos da sua dissociação; gorduras alimentares elaboradas; ceras de origem animal ou vegetal
Seção IV	Produtos das indústrias alimentares; bebidas, líquidos alcoólicos e vinagres; fumo (tabaco) e seus sucedâneos manufaturados
Cap. 16	Preparações de carne, de peixes ou de crustáceos, de moluscos ou de outros invertebrados aquáticos
Cap. 17	Açúcares e produtos de confeitaria
Cap. 18	Cacau e suas preparações
Cap. 19	Preparações à base de cereais, farinhas, amidos, féculas ou de leite; produtos de pastelaria
Cap. 20	Preparações de produtos hortícolas, de frutas ou de outras partes de plantas
Cap. 21	Preparações alimentícias diversas
Cap. 22	Bebidas, líquidos alcoólicos e vinagres
Cap. 23	Resíduos e desperdícios das indústrias alimentares; alimentos preparados para animais
Cap. 24	Fumo (tabaco) e seus sucedâneos, manufaturados
Seção V	Produtos minerais

- Cap. 25 Sal; enxofre; terras e pedras; gesso, cal e cimento
- Cap. 26 Minérios, escórias e cinzas
- Cap. 27 Combustíveis minerais, óleos minerais e produtos de sua destilação; matérias betuminosas; ceras minerais

Seção VI Produtos das indústrias químicas ou das indústrias conexas

- Cap. 28 Produtos químicos inorgânicos; compostos inorgânicos ou orgânicos de metais preciosos, de elementos radioativos, de metais das terras raras ou de isótopos
- Cap. 29 Produtos químicos orgânicos
- Cap. 30 Produtos farmacêuticos
- Cap. 31 Adubos ou fertilizantes
- Cap. 32 Extratos tanantes e tintoriais; taninos e seus derivados; pigmentos e outras matérias corantes, tintas e vernizes, mástiques; tintas de escrever
- Cap. 33 Óleos essenciais e resinóides; produtos de perfumaria ou de toucador preparados e preparações cosméticas
- Cap. 34 Sabões, agentes orgânicos de superfície, preparações para lavagem, preparações lubrificantes, ceras artificiais, ceras preparadas, produtos de conservação e limpeza, velas e artigos semelhantes, massas ou pastas de modelar, "ceras" para dentistas e composições para dentista à base de gesso
- Cap. 35 Matérias albuminóides; produtos à base de amidos ou de féculas modificados; colas; enzimas
- Cap. 36 Pólvoras e explosivos; artigos de pirotecnia; fósforos; ligas pirofóricas; matérias inflamáveis
- Cap. 37 Produtos para fotografia e cinematografia
- Cap. 38 Produtos diversos das indústrias químicas

Seção VII Plásticos e suas obras; borracha e suas obras

- Cap. 39 Plásticos e suas obras
- Cap. 40 Borracha e suas obras

Seção VIII Peles, couros, peleteria (peles com pêlo*) e obras destas matérias; artigos de correeiro ou de seleiro; artigos de viagem, bolsas e artefatos semelhantes; obras de tripa

Cap. 41	Peles, exceto a peleteria (peles com pêlo*), e couros
Cap. 42	Obras de couro; artigos de correeiro ou de seleiro; artigos de viagem, bolsas e artefatos semelhantes; obras de tripa
Cap. 43	Peleteria (peles com pêlo*) e suas obras; peleteria (peles com pêlo*) artificial
Seção IX	Madeira, carvão vegetal e obras de madeira; cortiça e suas obras; obras de espartaria ou de cestaria
Cap. 44	Madeira, carvão vegetal e obras de madeira
Cap. 45	Cortiça e suas obras
Cap. 46	Obras de espartaria ou de cestaria
Seção X	Pastas de madeira ou de matérias fibrosas celulósicas; papel ou cartão de reciclar (desperdícios e aparas); papel e suas obras
Cap. 47	Pastas de madeira ou de outras matérias fibrosas celulósicas; papel ou cartão de reciclar (desperdícios e aparas)
Cap. 48	Papel e cartão; obras de pasta de celulose, de papel ou de cartão
Cap. 49	Livros, jornais, gravuras e outros produtos das indústrias gráficas; textos manuscritos ou datilografados, planos e plantas
Seção XI	Matérias têxteis e suas obras
Cap. 50	Seda
Cap. 51	Lã e pêlos finos ou grosseiros; fios e tecidos de crina
Cap. 52	Algodão
Cap. 53	Outras fibras têxteis vegetais; fios de papel e tecido de fios de papel
Cap. 54	Filamentos sintéticos ou artificiais
Cap. 55	Fibras sintéticas ou artificiais, descontínuas
Cap. 56	Pastas ("ouates"), feltros e falsos tecidos; fios especiais; cordéis, cordas e cabos; artigos de cordoaria
Cap. 57	Tapetes e outros revestimentos para pavimentos, de matérias têxteis
Cap. 58	Tecidos especiais; tecidos tufados; rendas; tapeçarias; passamanarias; bordados
Cap. 59	Tecidos impregnados, revestidos, recobertos ou estratificados; artigos para usos técnicos de matérias têxteis

Cap. 60	Tecidos de malha
Cap. 61	Vestuário e seus acessórios, de malha
Cap. 62	Vestuário e seus acessórios, exceto de malha
Cap. 63	Outros artefatos têxteis confeccionados; sortidos; artefatos de matérias têxteis, calçados, chapéus e artefatos de uso semelhante, usados; trapos
Seção XII	Calçados, chapéus e artefatos de uso semelhante, guarda-chuvas, guarda-sóis, bengalas, chicotes, e suas partes; penas preparadas e suas obras; flores artificiais; obras de cabelo
Cap. 64	Calçados, polainas e artefatos semelhantes, e suas partes
Cap. 65	Chapéus e artefatos de uso semelhante, e suas partes
Cap. 66	Guarda-chuvas, sombrinhas, guarda-sóis, bengalas, bengalas-assentos, chicotes, e suas partes
Cap. 67	Penas e penugem preparadas, e suas obras; flores artificiais; obras de cabelo
Seção XIII	Obras de pedra, gesso, cimento, amianto, mica ou de matérias semelhantes; produtos cerâmicos; vidro e suas obras
Cap. 68	Obras de pedra, gesso, cimento, amianto, mica ou de matérias semelhantes
Cap. 69	Produtos cerâmicos
Cap. 70	Vidro e suas obras
Seção XIV	Pérolas naturais ou cultivadas, pedras preciosas ou semipreciosas e semelhantes, metais preciosos, metais folheados ou chapeados de metais preciosos, e suas obras; bijuterias; moedas
Cap. 71	Pérolas naturais ou cultivadas, pedras preciosas ou semipreciosas e semelhantes, metais preciosos, metais folheados ou chapeados de metais preciosos, e suas obras; bijuterias; moedas
Seção XV	Metais comuns e suas obras
Cap. 72	Ferro fundido, ferro e aço
Cap. 73	Obras de ferro fundido, ferro ou aço
Cap. 74	Cobre e suas obras
Cap. 75	Níquel e suas obras

- Cap. 76 Alumínio e suas obras
- Cap. 77 (Reservado para uma eventual utilização futura no SH)
- Cap. 78 Chumbo e suas obras
- Cap. 79 Zinco e suas obras
- Cap. 80 Estanho e suas obras
- Cap. 81 Outros metais comuns; ceramais (“*cermets*”); obras dessas matérias
- Cap. 82 Ferramentas, artefatos de cutelaria e talheres, e suas partes, de metais comuns
- Cap. 83 Obras diversas de metais comuns

Seção XVI Máquinas e aparelhos, material elétrico, e suas partes; aparelhos de gravação ou de reprodução de som, aparelhos de gravação ou de reprodução de imagens e de som em televisão, e suas partes e acessórios

- Cap. 84 Reatores nucleares, caldeiras, máquinas, aparelhos e instrumentos mecânicos, e suas partes
- Cap. 85 Máquinas, aparelhos e materiais elétricos, e suas partes; aparelhos de gravação ou de reprodução de som, aparelhos de gravação ou de reprodução de imagens e de som em televisão, e suas partes e acessórios

Seção XVII Material de transporte

- Cap. 86 Veículos e material para vias férreas ou semelhantes, e suas partes; aparelhos mecânicos (incluídos os eletromecânicos) de sinalização para vias de comunicação
- Cap. 87 Veículos automóveis, tratores, ciclos e outros veículos terrestres, suas partes e acessórios
- Cap. 88 Aeronaves e aparelhos espaciais, e suas partes
- Cap. 89 Embarcações e estruturas flutuantes

Seção XVIII Instrumentos e aparelhos de óptica, fotografia ou cinematografia, medida, controle ou de precisão; instrumentos e aparelhos médico-cirúrgicos; aparelhos de relojoaria; instrumentos musicais; suas partes e acessórios

- Cap. 90 Instrumentos e aparelhos de óptica, fotografia ou cinematografia, medida, controle ou de precisão; instrumentos e aparelhos médico-cirúrgicos; suas partes e acessórios

- Cap. 91 Aparelhos de relojoaria e suas partes
- Cap. 92 Instrumentos musicais, suas partes e acessórios

Seção XIX Armas e munições; suas partes e acessórios

- Cap. 93 Armas e munições; suas partes e acessórios

Seção XX Mercadorias e produtos diversos

- Cap. 94 Móveis, mobiliário médico-cirúrgico; colchões, almofadas e semelhantes; aparelhos de iluminação não especificados nem compreendidos em outros Capítulos; anúncios, cartazes ou tabuletas e placas indicadoras luminosos, e artigos semelhantes; construções pré-fabricadas
- Cap. 95 Brinquedos, jogos, artigos para divertimento ou para esporte; suas partes e acessórios
- Cap. 96 Obras diversas

Seção XXI Objetos de arte, de coleção e antiguidades

- Cap. 97 Objetos de arte, de coleção e antigüidades
- Cap. 98 (Reservado para usos especiais pelas Partes Contratantes)
- Cap. 99 Obras diversas

8.3 Representação do conhecimento

A Inteligência Artificial (IA) atua na transformação do conhecimento em formato legível por máquinas, de forma a possibilitar a utilização deste quando necessário, além de raciocinar – por meio de inferências, conclusões e explanações – para a solução de problemas diversos [25]. No entanto, o aspecto mais importante a ser observado antes de se transformar algum conhecimento é saber onde, e de que forma está organizado tal conhecimento.

A IA observa como funciona a inteligência humana, analisa a forma de trabalho de um especialista para, então, simular a inteligência deste especialista em outros ambientes, por pessoas que não são necessariamente especialistas. Caso ocorra algum erro, o sistema é notificado e pode, por si só, melhorar o processo de inferência das informações.

Gasevic et al. [25] afirma que o ser humano adquire conhecimento de forma transparente: uma pessoa que nunca teve contato com um computador de mão, no momento em que tiver com este instrumento, certamente vai examiná-lo, afim de aprender como funcionam as coisas básicas, memorizando, modelando e abstraindo o conhecimento, bem como as características do objeto. A mente desta pessoa terá, então, guardado todas as

funcionalidades do computador em questão e, caso precise, este conhecimento será recuperado no futuro para trabalhar em um outro computador de mão. Obviamente que se a pessoa em questão tiver tido algum contato com computadores de mesa ou portáteis antes, a aquisição de conhecimento do computador de mão será bem mais simples e rápida, pois a inferência do conhecimento anterior será usada.

Diferentemente dos homens os computadores não possuem um mecanismo transparente para aquisição e representação do conhecimento [25]. As máquinas dependem dos homens para a aquisição dos dados e a correta inferência destes. Um computador não irá avaliar se uma determinada preposição está de acordo com o mundo real ou não, mas tão somente cuidará para que as regras lógicas sejam atendidas. Para exemplificar, pode-se dizer que em um sistema mal projetado e/ou mal alimentado teria como uma possível saída que “pinguins voam.”

A unidade de conhecimento é, de modo resumido, um conjunto de elementos ligados entre si por uma estrutura, suposições, justificativas e processos [22]. Obviamente representar o conhecimento para uma máquina não é uma tarefa simples, e hoje a IA conta com várias representações distintas para o conhecimento humano [25]. Como na maioria das tecnologias, cada representação apresenta seus benefícios e falhas, tempos de respostas, facilidade em utilização, etc. Mas a conclusão é única: nenhuma delas pode representar todos os tipos de conhecimentos existentes. Para a solução de problemas complexos a combinação de diferentes representações será a saída mais adequada [25].

8.3.1 Ciência cognitiva

A Ciência Cognitiva é responsável pelos estudos da natureza da mente, e segundo Djuric et al. [19], a criação desta ocorreu em paralelo com a ciência de IA. Atualmente existem seis conceitos amplamente estabelecidos sobre as teorias cognitivas que explicam como a mente se comporta [25]:

1. **Lógica formal:** as pessoas fazem inferências porque têm em mente sentenças similares, bem como o processo dedutivo e indutivo para operar com estas sentenças. Tudo que se afirma com clareza em precisão em linguagem natural pode ser expresso por meio da lógica formal.
2. **Regras:** grande parte do conhecimento humano é descrito naturalmente em termos de regras da forma “se . . . então”. Pode-se dizer que boa parte do raciocínio humano pode ser modelado por um sistema baseado em regras. A capacidade de gerar novas regras a partir de regras passadas cria um comportamento primário para a simulação de inteligência.
3. **Conceitos:** são tratados pela ciência cognitiva como características nativas, tais como falar e escrever para os humanos. Pode-se dizer que os conceitos geralmente estão em várias partes de uma hierarquia de conhecimento, sendo criados em níveis mais genéricos e aproveitados em níveis mais especializados da hierarquia.

4. Analogias: os homens criam representações de determinadas situações, que podem ser usadas como casos passados ou analogias em novas situações. Os modelos computacionais que trabalham por analogia tentam simular o “recuperar, mapear e aplicar” do conhecimento em novas situações.
5. Imagens: os humanos captam as imagens visuais dos objetos, espaços e situações, sendo capazes de processar e manipular estas informações. As imagens conseguem compactar longas descrições textuais, e a computação tenta simular isto por meio da digitalização, adaptação, transformação e reconhecimento de imagens.
6. Conexões neurais: são pequenas unidades do cérebro – neurônios – que se interligam entre si, em um processo de excitação e inibição contínuo. Estas pequenas unidades interligadas são responsáveis pelo comportamento inteligente dos seres humanos.

A teoria que sustenta a afirmação de que a mente humana trabalharia apenas por meio de representações e cálculos recebe severas críticas, eis que, segundo seus opositores, negligenciaria aspectos que também incidem no trabalho da mente humana, quais sejam os aspectos emocionais e sociais [25]. A questão então é: a mente humana poderá algum dia ser descrita completamente por um sistema de computador, com a mesma dinamicidade, eficiência e velocidade?

O termo silogismo – tão conhecido nos dias atuais e diretamente relacionado com a ciência cognitiva – foi a tentativa de Aristóteles de codificar o pensamento correto por meio de processos de raciocínio irrefutáveis, e isto deu início ao estudo da lógica, ainda em 1965 [56]. Tais processos poderiam resolver, em princípio, qualquer programa que pudesse ser descrito em notação lógica.

Segundo [56], a ciência cognitiva reúne modelos computacionais da inteligência artificial e da psicologia para tentar construir teorias precisas e verificáveis, a respeito dos processos de funcionamento da mente humana por meio da introspecção – captação dos pensamentos à medida que se desenvolvem – e experimentos psicológicos.

8.3.2 Informação e conhecimento

Walton [76] propõe a criação de um cenário capaz de demonstrar a diferença entre informação e conhecimento, a fim de que tais conceitos sejam melhor compreendidos: Suponha que se esteja pesquisando na internet informações sobre uma doença específica. Em resposta à consulta tem-se uma página com todas as palavras chaves solicitadas, mas a página traz o conteúdo de um artigo médico. Pode-se ler o achado detalhadamente, buscar as palavras desconhecidas em um dicionário, examinar as referências apresentadas e tentar um contato com o autor. Ainda assim a probabilidade de se ter uma compreensão apenas um pouco maior sobre a doença em questão é grande, pois falta o conhecimento necessário para entender o artigo – o que normalmente só se alcança depois de anos de estudo.

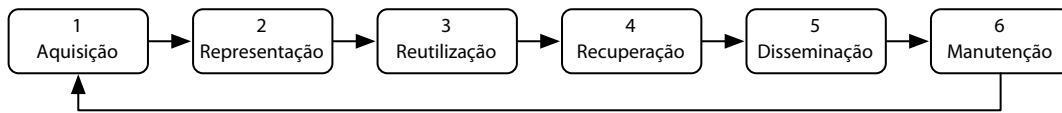


Figura 8.1: Ciclo de vida do conhecimento (adaptada) [76])

Algo semelhante ocorre ao se tentar utilizar um *software* para buscar informações na internet: o programa conseguirá facilmente contar as palavras chaves na página, baixar imagens, seguir algumas ligações de páginas e efetuar estatísticas; mas dificilmente será capaz de fazer qualquer outra coisa com a informação obtida [76].

Um problema real que enfatiza a diferença entre informação e conhecimento, também apresentado por Walton [76], é o de cálculo de rotas. Na internet encontra-se facilmente informações sobre rotas para um destino específico, mas o computador não consegue calcular a distância total da viagem sem que haja um programa específico para isto – *Google Maps*⁴, por exemplo. Isto se deve ao fato de o computador não conseguir compreender a informação da forma que está disponível, e exatamente por esta razão, assim é incapaz de realizar qualquer cálculo com os resultados obtidos.

Nesse entender, a questão fundamental está relacionada ao problema clássico de “informação *versus* conhecimento”: a internet está repleta de informações e fatos, mas estas devem ser interpretadas à luz de conceitos, verdades, crenças, perspectivas e metodologias para se tornarem em conhecimento [76]. Mas como representar o conhecimento de forma que estes possam ser compreendidos e interpretados pelo computador? Ao se conseguir a resposta, como usá-la efetivamente em aplicações reais?

As dúvidas que circundam o questionamento acerca de onde termina a informação e começa o conhecimento levou os pesquisadores a terem uma visão dinâmica do conhecimento, o que resultou em um ciclo de vida do conhecimento. O referido ciclo expressa a arte de gerir o conhecimento por meio de uma série de tarefas distintas que, como pode ser observado na Figura 8.1, organiza a tarefa de extrair e manter as informações obtidas da internet em seis fases [76]:

1. A primeira fase no modelo de ciclo é a aquisição das informações que deseja-se representar por meio de uma ontologia. As fontes desta informação podem ser distintas – especialistas no assunto, documentos das gavetas, relatórios, pareceres, etc. –, mas o foco principal deste trabalho são as informações oriundas da internet.
2. A segunda fase do ciclo de vida do conhecimento é representar as informações adquiridas na fase anterior, de forma adequada para a resolução de um determinado problema. Nesta fase a ontologia construída já deve ser capaz de classificar os conhecimentos adquiridos para o domínio alvo de forma flexível.
3. A terceira fase do ciclo visa reutilizar as fontes de conhecimentos já existentes, já que criar uma ontologia sem a utilização de conhecimentos e informações já

⁴<http://maps.google.com>

existentes não é uma tarefa comum. Existem casos que a ontologia deve ser criada do zero, mas deve-se sempre fazer uma pesquisa com o objetivo de verificar o que existe a fim de evitar trabalhos desnecessários. Como será visto adiante – Seção 4.5 – existem critérios de desenvolvimento para ontologias. Estes orientam de que forma as ontologias podem ser produzidas o mais genérica possível, no intuito de possibilitar a especialização e utilização desta por quaisquer outras aplicações.

4. A quarta fase do ciclo de vida tem o objetivo de recuperar o conhecimento da representação produzida. O conteúdo recuperado deve ser o conhecimento mais correto e relevante possível para um determinado problema, tarefa que fica mais complicada se a base de conhecimento tiver várias atualizações em curtos espaços de tempos, a exemplo das notícias.
5. A quinta fase propõe o compartilhamento do conhecimento gerado por outros usuários, para diferentes fins. Caso o conhecimento seja sensível, existem meios para que este seja restrito a determinado grupo, ou mesmo que apenas um subconjunto não sensível deste conhecimento seja compartilhado. Se a ontologia tiver sido bem construída, o limite de uso deste conhecimento será a capacidade da mente humana.
6. A sexta fase do ciclo de vida cuida da manutenção do conhecimento, para que este tenha sua utilidade preservada. Nesta fase deve-se verificar o conteúdo periodicamente, atualizá-lo para que reflita as novas informações e remover conhecimentos obsoletos; isto envolve uma profunda análise do conhecimento em questão.

O ciclo de vida do conhecimento, apresentado na Figura 8.1, não descreve uma metodologia rígida para a gestão do conhecimento ontológico; estas tarefas são destinadas tão somente a servir como um guia para as diversas questões que devemos considerar no desenvolvimento de uma base de conhecimento [76].

8.3.3 Tipos de conhecimento

A mente armazena diferentes tipos de conhecimento – o que indica uma capacidade nativa de organizar o conhecimento de forma estruturada para resolver problemas de forma mais eficiente. Os sistemas de IA tentam utilizar alguns destes para aplicar nas técnicas de representação do conhecimento humano [25]. A Figura 8.2 faz um resumo dos tipos de conhecimento humano mais comuns.

Gasevic et al. [25] descrevem em detalhes como funciona cada tipo de conhecimento: O “conhecimento procedimental” trata de como se realizar algo, como um problema é resolvido. O “conhecimento declarativo” descreve o que se conhece sobre um tópico ou problema. O “meta conhecimento” é o conhecimento que se tem sobre o conhecimento, e é usado para decidir qual a melhor estratégia para se resolver um determinado problema, dada as condições deste. O “conhecimento heurístico” contempla um conjunto de regras que guiam na solução do problema com base em experiências passadas. O “conhecimento

Categoria	Tipo
Conhecimento procedimental	regras, estratégias, agendas, procedimentos
Conhecimento declarativo	conceitos, objetos, fatos
Meta conhecimento	conhecimento sobre o conhecimento
Conhecimento heurístico	regras com base em experiências
Conhecimento estruturado	regras, relacionamentos, objetos
Conhecimento incerto	probabilidades, relações, evidências, Fuzzy
Conhecimento comum	regras, padrões, conceitos, analogias
Conhecimento ontológico	conceitos, relações, axiomas, restrições

Figura 8.2: Tipos de conhecimentos (adaptada) [25])

estrutural” descreve modelos mentais para a organização dos problemas. O “conhecimento incerto” é o responsável por caracterizar problemas em que a informação a cerca do problema é imprecisa, indisponível, incompleta, aleatória ou ambígua. O “conhecimento comum” é o termo usado para denotar o conhecimento humano sobre o mundo que não pode ser descrito facilmente em teorias. Por fim, o “conhecimento ontológico” é considerado um suplemento essencial para o conhecimento acerca de um domínio – este categoriza o problema, incluindo os diversos termos semelhantes que as pessoas usam para tratar deste.

8.3.4 Técnicas de representação do conhecimento

Os sistemas tradicionais de representação do conhecimento são centralizados, exigindo que todos compartilhem a mesma definição de conceitos, o que é sufocante para um sistema que cresça rapidamente [10]. A web semântica flexibilizou estas formas tradicionais, com a finalidade de que os sistemas produzidos pudessem obter e compartilhar conhecimento com certa independência.

Ao produzir um sistema de IA, o desenvolvedor deve escolher a técnica de representação que melhor se adeque ao caso em questão, além de ter em mente que uma mudança não prevista de variável durante a execução do código pode exigir mudanças para outras técnicas de representação [25]. Com uma técnica de representação do conhecimento adequada, pode-se produzir inferências sobre a informação e, assim, obter novas ideias [76].

Por oportuno, antes da apresentação das técnicas, será apresentado o que é conceituação, de acordo com as ideias de Walton [76]. A Figura 8.3 apresenta uma especificação que pode ser usada para classificar as câmeras em diferentes categorias. As caixas contêm os conceitos e as setas definem a relação entre eles. A raiz da hierarquia é uma classe chamada *Thing*, que representa o universo do discurso, ou seja, qualquer coisa que possa ser conceituada. A classe *Thing* tem duas subclasses diretas: uma tangível que existe fisicamente – máquina fotográfica – e uma intangível que é abstrata – propriedades co-

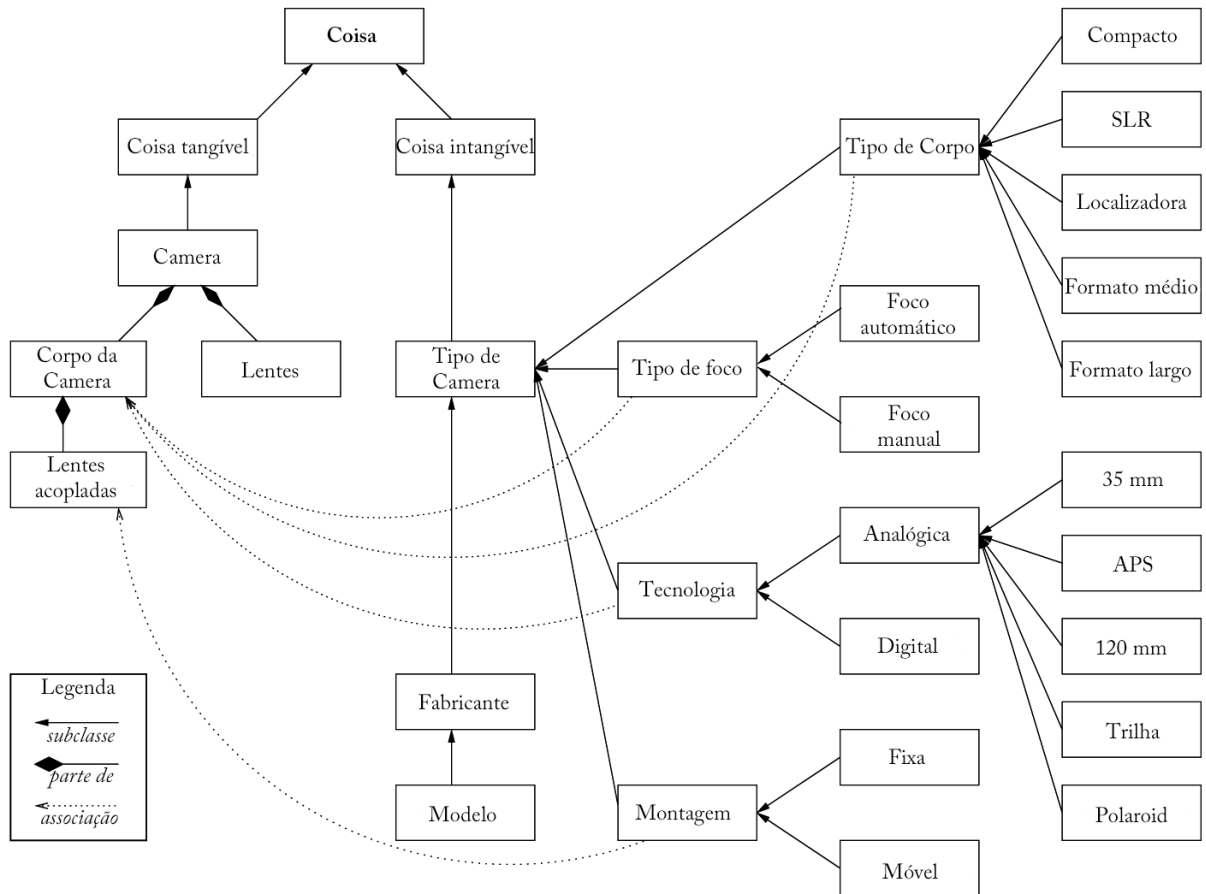


Figura 8.3: Ontologia de uma câmera fotográfica (adaptada) [76]

mo “focagem automática”. Observe que a relação *partof* é usada para relacionar objetos reais enquanto a relação *association* é usada para ligar propriedades abstratas a objetos tangíveis.

Representar o conhecimento de forma conceitual é fundamental para o processamento automático de informações da internet. Dois passos importantes neste sentido são produzir um modelo conceitual adequado e classificar as informações disponíveis na internet – esta é a tarefa mais complexa [76].

8.3.4.1 Tríplice objeto, valor e atributo

A técnica do objeto—atributo—valor (O—A—V) é usada para representar fatos sobre objetos e de seus atributos [25]. Por exemplo, a frase “a cor do carro é vermelho” na forma tríplice seria representada de acordo com a Figura 8.4a.

Os objetos, representados por elipses, normalmente têm mais de um atributo – os atributos são representados pela setas. Em uma representação gráfica de vários atributos, sairia de cada elipse múltiplas setas, cada uma ligada a uma caixa, onde as caixas representam os valores dos atributos. Um exemplo desta situação pode ser visto na Figura 8.4b.

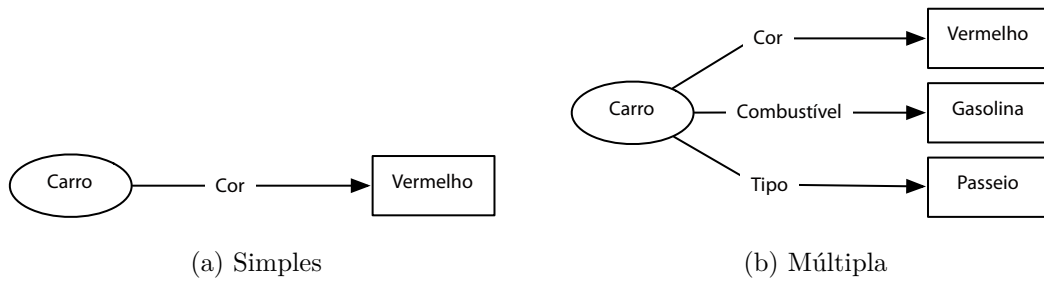


Figura 8.4: Exemplo de tríplexes O—A—V.

8.3.4.2 Fatos incertos

A tríplex O—A—V pode ser estendida para permitir que alguns fatos incertos sejam representados. Um fato certo é um valor número associado à afirmativa que representa o grau de verdade desta [25]. Para a correta compreensão dos números estes devem ser definidos em intervalos discretos, e ter um entendimento linguístico associado (Figura 8.5).

$$Fato \begin{cases} falso & = [-1, 0 \dots -0, 6] \\ provavelmente falso & = [-0, 5 \dots -0, 1] \\ desconhecido & = [0, 0] \\ provavelmente certo & = [+0, 1 \dots +0, 5] \\ certo & = [+0, 6 \dots +1, 0] \end{cases}$$

Figura 8.5: Entendimento linguístico associado a um fato incerto.

Desta forma, a representação da afirmação incerta “O carro provavelmente é vermelho ferrugem”, de acordo com a Figura 8.6, é um fato certo se utilizado o entendimento linguístico associado da Figura 8.5.



Figura 8.6: Exemplo de O—A—V com fator incerto.

Intervalos contínuos de valores possíveis devem ser divididos em subintervalos contínuos, e este relacionado ao valor discreto mais próximo.

8.3.4.3 Fatos difusos

Fatos difusos representam a incerteza utilizando termos imprecisos e ambíguos, comum na língua natural [25]. Por exemplo, a sentença “aquela pessoa é velha” pode ser interpretado de maneira distinta por diferente pessoas, pois estas têm entendimento próprio para a palavra “velha”.

Um conjunto difuso representando a sentença “aquela pessoa é velha” está representada na Figura 8.7, que é avaliada por meio de uma função. Esta função é usada para calcular o valor numérico do conjunto difuso, em uma escala que vai de 0 a 1.

8.3.4.4 Regras lógicas

As regras lógicas são as técnica de representação que estruturam e relacionam uma ou mais premissas, com o objetivo de se obter uma ou mais conclusões [25]. Este é o caso mais próximo para os programadores de *software*, pois as linguagens de programação atuais operam com este modelo de representação. Um exemplo de utilização desta técnica pode ser visto na Figura 8.8.

SE	Estiver chovendo
E	Eu tiver que sair
ENTÃO	Levarei um guarda-chuva

Figura 8.8: Exemplo de uma regra lógica.

Deve-se lembrar que quando a conclusão só pode ser inferida das premissas quando estas são verdadeiras. Também que o conhecimento de um sistema inteligente pode ser representando utilizando algumas regras lógicas, que são agrupadas em conjuntos, de acordo com a relação desta em um determinado tópico.

8.3.4.5 Redes semânticas

Redes semânticas – ou mapas de conceitos – é uma técnica de representação do conhecimento que tenta simular a mente [25]. Seguindo o conceito do modelo de memória associativa, pode-se dizer que as redes semânticas são gráficos de objetos, conceitos e situações em um determinado domínio do conhecimento – nós do gráfico – conectados por relações semânticas.

A Figura 8.9 mostra o exemplo de uma rede semântica, da qual se pode concluir: “Billy é um Labrador (raça de cão) e tem quatro patas, assim como todos os cachorros”.

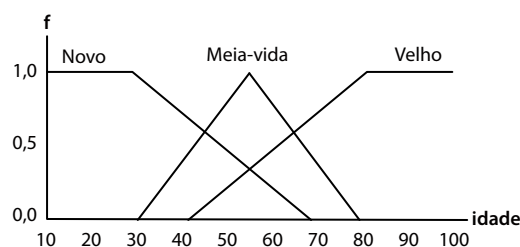


Figura 8.7: Exemplo de conjunto Fuzzy para a idade (adaptada) [25]).

Os arcos podem descrever qualquer tipo de relação entre os conceitos, sendo os mais comuns o “tipo de”, “parte de” e o “é um”. A notação gráfica mostrada para representar redes semânticas não está padronizada, mas sempre haverá na representação os conceitos, instâncias de objetos, atributos, valores dos atributos e a relação entre os conceitos de instâncias de objetos [25].

8.3.4.6 Baseada em quadros

Baseada em quadros – ou em inglês *frame-based* – é uma estrutura de dados complexa para representar conhecimento genérico, conceito, conjunto de atributos (*slots*) ou objetos. Quadros são similares as classes e objetos na linguagem de programação orientada à objetos. Existem as *class frames* (classes de quadros) para representar as características comuns de um conjunto de objetos similares e as *instance frames* (instâncias de quadros) que representa uma instância específica da *class frames* [25].

8.3.5 Linguagens para representação do conhecimento

A parte mais importante de qualquer sistema de inteligência é a sua base de conhecimento, que contém um conjunto de sentenças representadas usando uma ou mais técnicas de representações [25].

Tais sentenças devem ser representadas em uma linguagem de representação, de forma que novas sentenças possam ser adicionadas à base de conhecimento, de forma a aprimorar e tornar o sistema inteligente mais preciso e completo. Consultas e funções podem ser desenvolvidas nestas linguagens para, de acordo com a base de conhecimento, extrair o que o sistema inteligente conhece sobre o mundo.

As linguagens de para representação do conhecimento devem prover meios para que as representações sintática e semântica dos domínios, eventos, ações, processos e tempo sejam possíveis. Uma única linguagem de representação não é capaz de fornecer todas as características necessárias [25], e por isso um conjunto de linguagens por vezes deve ser usado.

As linguagens para representação do conhecimento diferem um pouco das linguagens

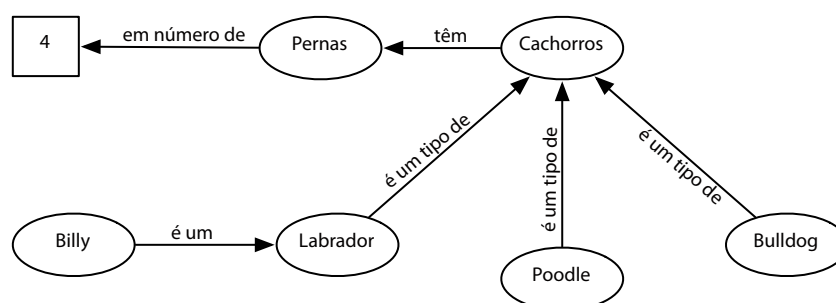


Figura 8.9: Exemplo de uma rede semântica (adaptada) [25]).

de programação. Aquelas são próprias da inteligência artificial, tais como a Lisp⁵ e a Prolog⁶, que dispõem de formas próprias para derivar fatos de outros fatos, o que é importante para a representação do conhecimento. Já as linguagens de programação se preocupam apenas com a representação dos dados, seguindo um fluxo de programa único, de acordo com as regras lógicas definidas no corpo do programa.

8.3.5.1 Lógica Proposicional

É uma sentença lógica que pode ser avaliada como verdadeira ou falsa, como no exemplo: “Se você estudar passará de série na escola”.

Ao se atribuir variáveis à sentença anterior teríamos:

$$\begin{cases} \textit{Se você estudar} & (A) \\ \textit{passará de série na escola} & (B) \end{cases}$$

Desta forma esta a sentença inteira poderia ser entendida pelo sistema inteligente apenas como:

$$A \rightarrow B$$

Cada sentença podem ser ligadas pelos operadores “e ($\wedge$)”, “ou ($\vee$)”, “não ($\neg$ ou $\sim$)”, “implicação ($\rightarrow$ ou $\Rightarrow$)” e “equivalência ($\leftrightarrow$ ou $\Leftrightarrow$)”. Em algumas situações a utilização desta lógica é inviável dada a dificuldade de atribuir uma variável a determinada parte da sentença.

8.3.5.2 Lógica de Primeira Ordem

A Lógica de Primeira Ordem estende a Lógica Proposicional, adicionando os operadores “quantificador universal ($\forall$)” e o “quantificador existencial ($\exists$)”.

Neste caso a sentença apresentada no item anterior agora poderia ser escrita como “Todo aluno que estuda passa de série na escola” e ter a seguinte representação simbólica:

$$\forall x (x \rightarrow B)$$

onde a variável A na foi substituída pela variável x , para se adequar à forma universal de representação.

Linguagens lógicas de altas ordens, lógica modal, lógica *fuzzy* e redes neurais podem ser definidas em termos da lógica de primeira ordem.

⁵O nome *Lisp* foi originado da junção dos termos ***List processing***. É uma família de linguagens de programação funcional, procedimental e orientada a objetos que usa exclusivamente funções matemáticas como estrutura de dados elementares, criada por John McCarthy em 1958 (<http://pt.wikipedia.org/wiki/Lisp>, acesso em 27 de fevereiro de 2011).

⁶Linguagem lógica e declarativa, criada por Alain Colmerauer e Robert Kowalski em 1972, com o objetivo de ter uso especial em inteligência artificial e linguística computacional (<http://pt.wikipedia.org/wiki/Prolog>, acesso em 27 de fevereiro de 2011).

8.3.5.3 Forma intercambiável de conhecimento (KIF)

Knowledge Interchange Format (KIF) é uma linguagem de representação baseada na Lógica de Primeira Ordem, projetada especificamente para ser um intermediador entre as diversas línguas de representação.

O KIF usa a notação da Lógica de Primeira Ordem com extensões de raciocínio não monotônico e definições. O exemplo a seguir mostra a sintaxe da linguagem KIF:

<code>(forall ((?x P))(B ?x))</code>	<code>;"Todos os P são B"</code>	1
<code>(exist ((?x P) (not (Q ?x))</code>	<code>;"Algum P não é Q"</code>	2
<code>(presidente 505 lula)</code>	<code>;um registro nos dados</code>	3
	<code>;de presidentes</code>	4
<code>(presidente 506 dilma)</code>	<code>;outro registro nos dados de presidentes</code>	5

Outro ponto importante do KIF é a utilização dos operadores apóstrofo e vírgula, que habilita a codificação de conhecimento sobre o conhecimento. Observe um exemplo:

<code>(nota secretario `(presidente ,?x ,?y ,?z))</code>	1
--	---

O código anterior apresenta o conhecimento de que o secretário faz nota sobre as sentenças sobre presidentes. O apóstrofo é usado na sentença interna, as interrogações denotam as variáveis, mas a vírgula antes de cada uma denota que estas devem ser usadas literalmente. Em outras palavras, o secretário faz uma anotação da sentença “(presidente ?x ?y ?z)”, e não do resultado da sentença.

8.3.5.4 Lógica descritiva

Um grupo de linguagens para representação do conhecimento, baseadas em lógica descritiva, é um interesse particular do desenvolvimento e representações de ontologias, conforme veremos no Capítulo 4.

A produção de uma base de conhecimento utilizando Lógica Descritiva implica no uso de uma terminologia – ou vocabulário composto por conceitos e papéis do domínio – sobre uma parte do conhecimento, denominada *Terminological Box* (*TBox*) [75]. A outra parte, denominada *Assertional Box* (*ABox*), cuida das afirmações sobre indivíduos nomeados pelo vocabulário da *TBox* [75]. Os conceitos podem ser atômicos ou complexos, sempre acompanhados de regras. Uma visualização funcional desta linguagem para representação do conhecimento é apresentada na Figura 8.10.

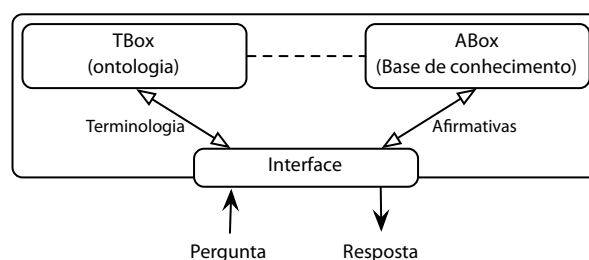


Figura 8.10: Visão funcional da lógica descritiva (adaptada) [76])

Mulher	=	$\text{Pessoa} \wedge \text{doSexoFeminino}$	Mulher(Flávia)
Homem	=	$\text{Pessoa} \wedge \neg \text{Mulher}$	Mulher(Lázara)
Mãe	=	$\text{Mulher} \wedge \exists \text{Filho.Pessoa}$	Homem(Charles)
Pai	=	$\exists \text{Filho.Pessoa}$	Mãe(Flávia, Lara)
Pais	=	$\text{Mãe} \vee \text{Pai}$	Pai(Charles, Lara)
Avó	=	$\text{Mãe} \wedge \exists \text{Filho.Pais}$	Avó(Lázara, Lara)
MuitosFilhos	=	$\text{Pessoa} \wedge \geq 3 \text{Filho.Pessoa}$	

(a) Definição *TBox*.

(b) *ABox* usando a *TBox* da figura anterior.

Figura 8.11: Exemplo do uso de uma *TBox*/*ABox*.

Por exemplo, ser uma “pessoa do sexo feminino” é um conceito atômico e “ter filho” é uma regra atômica, a *TBox* pode definir um conceito mais complexo denominado “família”, conforme ilustra a Figura 8.11a. Há de se notar que a penúltima linha da Figura 8.11a apresenta uma restrição cardinal que, neste caso, representa o número mínimo de filhos para se considerar que uma mulher tem muitos filhos.

Com as regras preparadas na *TBox* – Figura 8.11a, e as entradas da Figura 8.11b, o sistema poderia chegar facilmente à conclusões do tipo “A Flávia tem poucos filhos” ou “A Lázara é mãe da Flávia”.

8.3.5.5 Baseada em quadros

A notação desta linguagem é baseada em quadros (*frames*), conceitos, classes, propriedades e relações entre os objetos instanciados [75]. Como é bem próxima da orientação a objetos, oferece recursos de generalização e especialização, de forma a organizar conceitos de forma hierárquica. Um exemplo da sintaxe desta linguagem é apresentado a seguir:

```
(toda Compra tem                                     1
    (comprador ((um Agente)))                         2
    (objeto ((uma Coisa)))                             3
    (vendedor ((um Agente))))                         4
```

Um código com o mesmo objetivo é apresentado a seguir, na linguagem KIF:

```
(forall (?b)                                          1
    (exists (?a) (=> (isa ?b Compra) (and (comprador ?b ?a) (isa ?um  2
        Agente)))))
(fforall (?b)                                         3
    (exists (?a) (=> (isa ?b Compra) (and (objeto ?b ?a) (isa ?uma  4
        Coisa)))))
(fforall (?b)                                         5
    (exists (?a) (=> (isa ?b Compra) (and (vendedor ?b ?a) (isa ?um  6
        Agente)))))
```

8.3.5.6 Baseada em Regras

Linguagens de representação baseada em regras são populares nas aplicações de inteligência artificial comerciais, em especial nos sistemas especialistas. Este tipo de linguagem é

de fácil entendimento e existem inúmeras ferramentas disponíveis para construir sistemas com base nesta linguagem.

O código a seguir foi produzido em Jess⁷, uma ferramenta popular para a construção de sistemas especialistas:

⁷É um mecanismo para a produção e inferência de regras para a plataforma Java, criado por Ernest Friedman-Hill em 1995 (http://en.wikipedia.org/wiki/Jess_programming_language, acessado em 13 de janeiro de 2011).

```

      (deftemplate pessoa (slot idade))          1
      ...                                         2
      (defrule checa-idade                        3
        (pessoa (idade ?x))                     4
        (test (> ?x 30))                        5
=>                                               6
        (printout t ?x ``tem mais que 30 anos!'' 7
          crlf))

```

No exemplo anterior cria-se um quadro *pessoa* usando o construtor *deftemplate* e a regra *checa-idade* com o construtor *defrule*. A regra checa se a instância da *pessoa* tem mais de trinta anos e, em caso positivo, a função *printout* exibe a mensagem.

Referências Bibliográficas

- [1] Equipe Técnica Aduandeiras. *Nomenclatura Comum do Mercosul*. Aduandeiras, São Paulo, Brasil, 3 edition, 2010. ISBN 9788571295742. 3, 4, 13, 14, 15, 16, 17, 60, 61, 62, 64, 66, 86
- [2] Harith Alani, Sanghee Kim, David E Millard, Mark J Weal, Wendy Hall, Paul H Lewis, and Nigel Shadbolt. Automatic Ontology-Based Knowledge Extraction from Web Documents. *IEEE Intelligent Systems*, pages 2–9, 2003. 19, 44
- [3] Neda Alipanah, Piyush Srivastava, Pallabi Parveen, and Bhavani Thuraisingham. Ranking Ontologies Using Verified Entities to Facilitate Federated Queries. *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 332–337, August 2010. doi: 10.1109/WI-IAT.2010.147. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5616593>. 47
- [4] Ryan Ribeiro De Azevedo. *CoreSec: Uma Ontologia para o Domínio de Segurança da Informação*. Mestrado, UFPE – Universidade Federal de Pernambuco, 2008. URL <http://www.infociencia.info/obras/ARTr1051.pdf>. 45
- [5] Christopher J. O. Baker and Kei-Hoi Cheung, editors. *Semantic web : revolutionizing knowledge discovery in the life sciences*. Springer, New York, 2007. ISBN 9780387484365. 20, 21, 26, 30, 55
- [6] Tânia Barbosa, Salles Gava, and Crediné Silva De Menezes. Uma ontologia de domínio para a aprendizagem cooperativa. (27):336–345, 2003. 47
- [7] Domenico Beneventano and Daniele Montanari. Ontological Mappings of Product Catalogues. Technical report, 2008. 46, 57
- [8] S. Bergamaschi, F. Guerra, and M. Vincini. A data integration framework for e-commerce product classification. *The Semantic Web—ISWC 2002*, pages 379–393, 2002. URL <http://www.springerlink.com/index/hpexmg4mw2702c7r.pdf>. 4, 55, 57, 58, 59
- [9] Tim Berners-Lee and CERN. Information Management: A Proposal. Technical report, 1989. URL <http://www.w3.org/History/1989/proposal.rtf>. 3, 19, 20

- [10] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 21:36, 2001. URL http://kill.devc.at/system/files/scientific-american_0.pdf. 19, 37, 39, 103
- [11] D Bianchini and V De Antonellis. An ontology-based method for classifying and searching e-Services. *Forum Session at the First*, 2003. URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.101.4386&rep=rep1&type=pdf#page=69>. 46
- [12] Karin Koogan Breitman. *Web semântica: a internet do futuro*. LTC, Rio de Janeiro, 1 edition, 2005. ISBN 8521614667. 4, 56
- [13] Mark Burstein, Christoph Bussler, Michal Zaremba, Tim Finin, Michael Huhns, Massimo Paolucci, Amit Sheth, and Stuart Williams. A Semantic Web Services Architecture. *Ieee Internet Computing*, (October):52–61, 2005. 36, 37
- [14] Jorge Cardoso. *Semantic Web Services : Theory, tools and applications*. Hershey PA, 2007. ISBN 9781599040455. URL <http://books.google.com/books?hl=en&lr=&id=SkQCBBUbfNQC&oi=fnd&pg=PP1&dq=Semantic+Web+Services++Theory,+tools+and+applications&ots=A1J4FEFPzU&sig=zm9hFSgW1vxm3pa8mjpgLmtBGaWo>. 21, 55
- [15] Robert M. Colomb. *Ontology and the semantic web*. IOS Press, Amsterdam, Netherlands, 1 edition, 2007. ISBN 9781586037291. URL <http://portal.acm.org/citation.cfm?id=1565452.1565454>. 21, 22, 23, 24, 25, 26, 27
- [16] Cesar Olivier Dalston. Classificação de mercadorias: classificação de computadores, Fevereiro 2011. URL <http://classificacaodemercadoria.blogspot.com/2011/01/classificacao-de-computadores-na-ncm.html>. 62
- [17] John Davies, Rudi Studer, and Paul Warren. *Semantic Web Technologies : trends and research in ontology-based systems*. John Wiley & Sons, Ltd, Chichester, West Sussex, England, 1 edition, 2006. ISBN 9780470025963. 19, 20, 21, 32
- [18] JLG Dietz. *Enterprise Ontology - Theory and Methodology*. Springer, Berlin, New York, 2006. ISBN 9783540291695. URL http://books.google.com/books?hl=en&lr=&id=jg_CXuVjJyIC&oi=fnd&pg=PA3&dq=Enterprise+Ontology&ots=SIqZ9VaJ-f&sig=fT2fT78Yz2XGu1LSDvaGxBcXYFU. 40, 43
- [19] Dragan Djuric, Dragan Gasevic, and Vladan Devedzic. Ontology modeling and MDA. *Journal of Object technology*, 4(1):109–128, 2005. URL http://www.jot.fm/issues/issue_2005_01/article3/. 28, 99

- [20] Asuman Dogac, Gokce Laleci, Yildiray Kabak, and Ibrahim Cingil. Exploiting web service semantics: Taxonomies vs. ontologies. *Data Engineering*, 51:7, 2002. URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.81.5443&rep=rep1&type=pdf#page=12>. 17, 55, 57
- [21] Liang Dong, Pradip K. Srimani, and James Z. Wang. WEST: Weighted-Edge Based Similarity Measurement Tools for Word Semantics. *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 216–223, August 2010. doi: 10.1109/WI-IAT.2010.39. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5616536>. 59
- [22] T. Edgington, B. Choi, Katherine Henson, TS Raghu, and A. Vinze. Adopting Ontology to Facilitate Knowledge Sharing. *Communications of the ACM*, 47(11): 85–90, 2004. ISSN 0001-0782. URL <http://portal.acm.org/citation.cfm?id=1029499&coll=portal&dl=ACM&ret=1>. 3, 39, 41, 43, 47, 49, 99
- [23] André Luiz Dias Esbízaro. *Recuperação de informações sobre LOG de eventos apoiada em ontologia*. Mestrado, UnB – Universidade de Brasília, 2006. URL http://bdtd.bce.unb.br/tesdesimplificado/tde_busca/processaArquivo.php?codArquivo=1350. 34, 44, 46
- [24] Dieter Fensel, Holger Lausen, Axel Polleres, Jos De Bruijn, Michael Stollberg, Dumitru Roman, and John Domingue. *Enabling semantic web services : the web service modeling ontology*. Springer, Berlin, Heidelberg, 1 edition, 2007. ISBN 9783540345190. 20, 22, 27
- [25] Dragan Gasevic, Dragan Djuric, and Vladan Devedzic. *Model Driven Architecture and Ontology Development*. Springer, New York, 1 edition, July 2006. ISBN 9783540321804. URL <http://www.amazon.com/dp/3540321802>. 3, 4, 19, 21, 22, 23, 24, 25, 26, 27, 28, 29, 32, 36, 38, 40, 44, 48, 50, 98, 99, 100, 102, 103, 104, 105, 106, 107
- [26] Vladimir Geroimenko. *Dictionary of XML technologies and the semantic Web*. Springer, London, New York, 2004. ISBN 9781852337681. 25
- [27] Asunción Gómez-pérez and Oscar Corcho. Ontology Languages for the Semantic Web. *IEEE Intelligent Systems*, 4:54–60, 2002. 25, 27, 30, 43, 44
- [28] Thomas Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43(5-6):907–928, November 1993. ISSN 10715819. doi: 10.1006/ijhc.1995.1081. URL <http://linkinghub.elsevier.com/retrieve/doi/10.1006/ijhc.1995.1081>. 38, 39, 40, 41, 44, 47, 48, 49

- [29] Nicola Guarino. Formal ontology, conceptual analysis and knowledge representation. *International Journal of Human-Computer Studies*, 43(5-6):625–640, November 1995. ISSN 10715819. doi: 10.1006/ijhc.1995.1066. URL <http://linkinghub.elsevier.com/retrieve/doi/10.1006/ijhc.1995.1066>. 40, 41, 44
- [30] Yan Ha and Roger Lee. Integration of Semantic Web Service and Component-Based Development for e-business environment. *Management*, page 8, 2006. 20, 36
- [31] Friedhelm Hausmann. International Standard for de Classification and Description of Products and Services, 2011. URL <http://www.eclass.de>. 3, 18, 57, 58, 68
- [32] G Van Heijst, A. Th. Schreiber, and B. J. Wielinga. Using explicit ontologies in KBS development. *International Journal of Human*, pages 183 – 292, 1997. URL <http://www.cs.vu.nl/~guus/papers/Heijst97a.pdf>. 44
- [33] James Hendler. Agents and the semantic web. *IEEE Intelligent systems*, pages 30–37, 2001. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=920597. 21, 39, 41
- [34] Martin Hepp, Pieter De Leenher, Aldo De Moor, and York Sure. *Ontology Management - Semantic Web, Semantic Web Services and Business Applications*. Springer, New York, 1 edition, 2008. ISBN 9780387698991. 19
- [35] Jena. Jena Semantic Web Framework, Julho 2011. URL <http://jena.sourceforge.net>. 70, 81, 82
- [36] Yannis Kalfoglou and Marco Schorlemmer. Ontology mapping: the state of the art. *The knowledge engineering review*, 18(01):1–31, 2003. ISSN 0269-8889. URL http://journals.cambridge.org/abstract_S0269888903000651. 41, 47, 57
- [37] Sheila Kinsella, Uldis Bojars, Andreas Harth, John G. Breslin, and Stefan Decker. An Interactive Map of Semantic Web Ontology Usage. *2008 12th International Conference Information Visualisation*, pages 179–184, July 2008. doi: 10.1109/IV.2008.60. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4577944>. 45
- [38] Neeraj Koul and Vasant Honavar. Learning in Presence of Ontology Mapping Errors. *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 291–296, August 2010. doi: 10.1109/WI-IAT.2010.138. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5616569>. 57
- [39] Ora Lassila and Ralph R. Swick. Resource description framework (RDF): model and syntax specification. <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>, Janeiro 2011. URL <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>. 25

- [40] Seok Won Lee and Robin A. Gandhi. Ontology-based active requirements engineering framework. *Software Engineering Conference*, page 8, 2005. URL <http://www.computer.org/portal/web/csd1/doi/10.1109/APSEC.2005.86>. 50
- [41] E. J. Lowe. *The Four-Category Ontology : A Metaphysical Foundation for Natural Science*. Oxford University Press, New York, 2006. ISBN 9780199254392. 43, 44
- [42] Alexander Maedche, Boris Motik, Ljiljana Stojanovic, Rudi Studer, and Raphael Volz. Ontologies for Enterprise Knowledge Management. *IEEE Intelligent Systems*, pages 2—9, 2003. 3, 46
- [43] MDIC – Ministério do Desenvolvimento, Indústria e Comércio Exterior. Comércio Exterior : Principais conceitos e variáveis estatísticas, Dezembro 2010. URL <http://www.desenvolvimento.gov.br/sitio/interna/interna.php?area=5&menu=1088&refr=605>. 16
- [44] MDIC – Ministério do Desenvolvimento, Indústria e Comércio Exterior. Nomenclatura Comum do Mercosul (NCM), Dezembro 2010. URL <http://www.desenvolvimento.gov.br/sitio/interna/interna.php?area=5&menu=1090&refr=605>. 14, 86
- [45] Thanh Nguyen and Tuoi Phan. A hybrid solution of ontology-based query expansion. *Information Systems*, 4(2):215–227, 2008. doi: 10.1108/17440080810882388. 45
- [46] Natalya F. Noy. Semantic Integration : A Survey Of Ontology-Based Approaches. 33(4):65–70, 2004. 19, 40, 47
- [47] Natalya F. Noy and Deborah L. McGuinness. Ontology Development 101 : A Guide to Creating Your First Ontology. *Development*, pages 1–25, 2000. 27, 39, 40, 41, 48, 49, 55, 56, 82
- [48] Vladimir Oleshchuk and Asle Pedersen. Ontology based semantic similarity comparison of documents. *Database and Expert Systems*, page 4, 2003. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1232108. 57
- [49] Borys Omelayenko. Integrating Vocabularies : Discovering and Representing Vocabulary Maps. *Proceedings of the First International Semantic Web Conference, ISWC-2002*, pages 1–15, 2002. 3, 17, 18, 58
- [50] Sean B. Palmer. The semantic web: An introduction. <http://infomesh.net/2001/swintro/>, Novembro 2011. URL <http://infomesh.net/2001/swintro/>. 19, 25
- [51] Thomas B. Passin. *Explorer’s guide to the Semantic Web*. Manning, Greenwich, CT, 2004. ISBN 1932394206. 22

- [52] Ana M. Pernas and M. A. R. Dantas. Ontologias Aplicadas à Descrição de Recursos em Ambientes Grid. *Supercomputer*, (2003):6, 2004. 45
- [53] Casa Civil Presidência da República. Decreto nº 6006, de 28 de dezembro de 2006. <http://www.planalto.gov.br>, Dezembro 2010. URL http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2006/decreto/d6006.htm. 16
- [54] C. Price and K. Spackman. SNOMED clinical terms. *BJHC&IM - British Journal of Healthcare Computing & Information Management*, 17(3):27–31, 2000. 40
- [55] United Nations Development Programme. The United Nations Standard Products and Services Code, 2011. URL <http://www.unspsc.org>. 17, 55, 57, 58, 68
- [56] Stuart Russel and Peter Norvig. *Inteligência Artificial*. Elsevier, Rio de Janeiro, 2 edition, 2004. ISBN 9788535211771. 3, 34, 35, 36, 100
- [57] A. F. Salam and Jason R. Stevens. *Semantic web technologies and e-business : Toward the integrated virtual organization and business process automation*. Idea Group Publishing, Hershey PA, 2007. ISBN 9781599041940. 57
- [58] Mario Ferreira Dos Santos. *Ontologia e Cosmologia*. Editora Logos, São Paulo, Brasil, 2 edition, 1957. 38
- [59] Nigel Shadbolt, Wendy Hall, and Tim Berners-Lee. The Semantic Web Revisited. *IEEE Intelligent Systems*, pages 96–101, 2006. 19, 30
- [60] Steffen Staab and Heiner Stuckenschmidt, editors. *Semantic Web and Peer-to-peer : decentralized management and exchange of knowledge and information*. Springer, Koblenz, Germany, 1 edition, 2006. ISBN 9783540283461. 32, 33
- [61] Stanford Medical Informatics. The Protégé Ontology Editor and Knowledge Acquisition System, Julho 2011. URL <http://protege.stanford.edu>. 50, 51, 82
- [62] Stanford Medical Informatics. Protege4Migration - Protege Wiki, Julho 2011. URL http://protegewiki.stanford.edu/wiki/Protege4Migration/#Side_by_Side_Comparison. 3, 52
- [63] Stanford Medical Informatics. Download Protégé Software - Release 4.1, Julho 2011. URL <http://protege.stanford.edu/download/registered.html#p4.1>. 62, 64, 70
- [64] Stanford Medical Informatics. Download Protégé Software - Release 3.4.7, Julho 2011. URL <http://protege.stanford.edu/download/registered.html#p3>. 62
- [65] Rodolfo Stecher, Claudia Niederée, Wolfgang Nejdl, and Paolo Bouquet. Adaptive ontology re-use: finding and re-using sub-ontologies. *International Journal of Web Information Systems*, 4(2):198–214, 2008. ISSN 1744-0084. doi:

- 10.1108/17440080810882379. URL <http://www.emeraldinsight.com/10.1108/17440080810882379>. 45
- [66] Heiner Stuckenschmidt and Frank Van Harmelen. Information Sharing on the Semantic Web. 2003. 28
- [67] Xiaomeng Su, Terje Brasethvik, and Sari Hakkarainen. Ontology mapping through analysis of model extension. In *proceeding of the CAiSE Forum*, pages 101–104. Cite-seer, 2003. URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.10.3769&rep=rep1&type=pdf>. 57
- [68] Xiaomeng Su, Mihhail Matskin, and Jinghai Rao. Implementing explanation ontology for agent system. *IEEE/WIC International Conference on Web Intelligence (WI'03)*, pages 1–7, 2003. doi: <http://doi.ieeecomputersociety.org/10.1109/WI.2003.1241212>. URL <http://www.computer.org/portal/web/csd1/doi/10.1109/WI.2003.1241212>. 45, 55, 57
- [69] York Sure, Michael Erdmann, Juergen Angele, Steffen Staab, Rudi Studer, and Dirk Wenke. OntoEdit: Collaborative ontology development for the semantic web. *The Semantic Web—ISWC 2002*, pages 221–235, 2002. URL <http://www.springerlink.com/index/DRUX1NWEV5GHR695.pdf>. 19, 50
- [70] Xiaohui Tao, Yuefeng Li, Raymond Y.K. Lau, and Shlomo Geva. Ontology-Based Specific and Exhaustive User Profiles for Constraint Information Fusion for Multi-agents. *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 264–271, August 2010. doi: 10.1109/WI-IAT.2010.76. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5616557>. 34
- [71] Samir Tartir and I. Budak Arpinar. Ontology Evaluation and Ranking using OntoQA. *International Conference on Semantic Computing (ICSC 2007)*, pages 185–192, September 2007. doi: 10.1109/ICSC.2007.19. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4338348>. 46
- [72] Phil Tetlow, JZ Pan, Daniel Oberle, Evan Wallace, and Michael Uschold. Ontology driven architectures and potential uses of the semantic web in systems and software engineering. Technical report, 2005. URL <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Ontology+Driven+Architectures+and+Potential+Uses+of+the+Semantic+Web+in+Systems+and+Software+Engineering#0>. 44
- [73] The Eclipse Foundation. Eclipse – The Eclipse Foundation open source community webiste, Julho 2011. URL <http://www.eclipse.org>. 70

- [74] TopQuadrant. TopQuadrant | Products | ToBraid Composer, Maio 2011. URL http://www.topquadrant.com/products/TB_Composer.html. 3, 52, 53, 62, 63, 67, 68, 70, 73, 76, 82
- [75] Ubbo Visser. *Intelligent Information Integration for the Semantic Web*. Springer, Berlin, Heidelberg, 2004. ISBN 9783540286363. 109, 110
- [76] Christopher D. Walton. *Agency and the Semantic Web*. Oxford University Press, New York, 1 edition, 2007. ISBN 0199292485. 3, 4, 20, 21, 24, 25, 26, 27, 28, 29, 31, 32, 33, 36, 47, 100, 101, 102, 103, 104, 109
- [77] Hai H Wang, Samson Tu, Natasha Noy, Alan Rector, Mark Musen, Timothy Redmond, Daniel Rubin, Tania Tudorache, Matthew Horridge, and Julian Seidenberg. Frames and OWL Side by Side. *Medical Informatics*, pages 1–4, 2001. 54, 55
- [78] Michael J. Wooldrigde. *An Introduction to MultiAgent Systems*. Wiley, United Kingdom, 2 edition, 2009. ISBN 9780470519462. 3, 28, 29, 30, 39, 42, 43, 49, 50
- [79] Wei Yu, Qing Li, Junpeng Chen, and Jiaheng Cao. OS _ RANK : Structure Analysis for Ontology Ranking. *Structure*, pages 339–346, 2007. doi: 1-4244-0832-6/07. 43, 47
- [80] Milan Zdravkovic and Miroslav Trajanovic. Integrated product ontologies for inter-organizational networks. *Computer Science and Information Systems*, 6(2):29–46, 2009. ISSN 1820-0214. doi: 10.2298/CSIS0902029Z. URL <http://www.doiserbia.nb.rs/Article.aspx?ID=1820-02140902029Z>. 84