

Universidade Federal de Juiz de Fora
Pós-Graduação em Ciência da Computação
Mestrado em Modelagem Computacional

Daniele Cristina Palazzi

**QDAontology – ABORDAGEM PARA O DESENVOLVIMENTO DE ONTOLOGIAS
EM E-SCIENCE: UM ESTUDO DE CASO EM BIOLOGIA**

Juiz de Fora
2010

**QDAontology – ABORDAGEM PARA O DESENVOLVIMENTO DE ONTOLOGIAS
EM E-SCIENCE: UM ESTUDO DE CASO EM BIOLOGIA**

Daniele Cristina Palazzi

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Modelagem Computacional da Universidade Federal de Juiz de Fora, como parte dos requisitos necessários à obtenção do grau de Mestre em Ciências em Modelagem Computacional.

Orientador: Fernanda Cláudia Alves
Campos

Juiz de Fora
Fevereiro de 2010

Daniele Cristina Palazzi

**QDAontology – ABORDAGEM PARA O DESENVOLVIMENTO DE ONTOLOGIAS
EM E-SCIENCE: UM ESTUDO DE CASO EM BIOLOGIA**

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do grau de Mestre.

Aprovada em 25 de fevereiro de 2010.

BANCA EXAMINADORA

Prof. Dra. Fernanda Cláudia Alves Campos – Orientadora
Universidade Federal de Juiz de Fora

Prof. Dra. Regina Maria Maciel Braga
Universidade Federal de Juiz de Fora

Prof. Dr. Alcione de Paiva
Universidade Federal de Viçosa

Aos meus pais, irmãos, noivo e sobrinhas.

AGRADECIMENTOS

Agradeço à minha orientadora Fernanda Campos por toda a dedicação e incentivo ao longo deste trabalho.

À prof. Regina Braga pelo auxílio nos trabalhos realizados.

À prof. Elaine Coimbra pela sua dedicação e disposição durante a elaboração deste trabalho, contribuindo com seu conhecimento na área da Biologia.

Aos membros da Banca Examinadora pelo trabalho de avaliação.

Ao meu noivo, Eduardo, pelo amor, carinho, paciência, companheirismo, incentivo e a dedicação de sempre.

Aos meus pais, Roselaine e Odazio, por tudo o que me proporcionaram na vida e pelo incentivo de sempre.

Aos meus irmãos José Américo, Rosimeire e Júnior, pelo apoio ao longo destes anos.

Às minhas queridas sobrinhas, Cinthia e Vivian, pelas alegrias proporcionadas.

À FAPEMIG pelo apoio financeiro através da bolsa de mestrado.

À todos que, direta ou indiretamente, contribuíram para a realização deste trabalho.

Ciência da Computação tem tanto a ver com o computador como a Astronomia com o telescópio, a Biologia com o microscópio, ou a Química com os tubos de ensaio. A Ciência não estuda ferramentas, mas o que fazemos e o que descobrimos com elas.

Edsger Dijkstra

Resumo da Dissertação apresentada à Universidade Federal de Juiz de Fora como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências

A utilização de ontologias em sistemas computacionais tem se tornando cada vez mais importante e difundida. Entretanto, por ser uma área em constante evolução, não existem modelos de processo de Engenharia Ontológica consolidados para a construção de uma ontologia. O uso de um processo de desenvolvimento torna esta tarefa mais impessoal, menos complexa e mais sistemática. A abordagem QDAontology - Quality Driven Approach for e-Science Ontologies, proposta nesta dissertação, foi elaborada para projetos de desenvolvimento de ontologias para e-Science. Nestas aplicações, as ontologias, em geral, se caracterizam por serem desenvolvidas por equipes multidisciplinares, onde os contuidistas pertencem à área do domínio da aplicação e os construtores são engenheiros ontológicos oriundos da Ciência da Computação. A abordagem proposta é composta por etapas, atividades, participantes, artefatos e características de qualidade. São seis etapas: Especificação, Conceitualização, Formalização, Implementação, Integração e Evolução. Cada etapa é constituída de atividades, em cada atividade são gerados artefatos e os participantes estão relacionados com as etapas e atividades do processo. A partir da evolução dos artefatos ocorre o desenvolvimento da ontologia. Com a implementação da ontologia a mesma deve ser integrada a outras ontologias, através de mecanismos de correspondência ontológica. O processo de desenvolvimento adotado é centrado em um modelo evolutivo e, portanto, os ciclos podem se repetir a cada evolução da ontologia. Para validar a proposta foi elaborado um estudo de caso no domínio biológico. O estudo constou de dois ciclos: no primeiro foi feita a reengenharia da ontologia CELO e no segundo sua expansão, no subdomínio Doenças Humanas. Todo o processo de modelagem do conhecimento e atuação da equipe multidisciplinar está detalhado e contou com o apoio de especialistas do domínio da Biologia. Para a correspondência ontológica foi adotado o mecanismo de alinhamento e construída a ferramenta A3O, que gera ligações e termos equivalentes entre ontologias descritas em OBO e OWL.

Palavras-chave: Ontologia, Processo de Desenvolvimento, Correspondência ontológica.

Abstract of Dissertation presented to Federal University of Juiz de Fora as a partial fulfillment of the requirements for the degree of Master of Science

The use of ontologies in computational systems has become more important and spread out. However, as an area in constant evolution, it does not exist yet a consolidated process model of Ontological Engineering for ontology construction. The use of a development process makes this task more impersonal, less complex and more systematic. The QDAontology - Quality Driven Approach for e-Science Ontologies, proposed in this research, was developed for e-Science ontologies. In this kind of applications, ontologies are, in general, developed by multidisciplinary teams, where knowledge experts belong to the domain area and developers are ontological engineers from the Computer Science. This approach is composed of stages, activities, participants, artifacts and quality characteristics. There are six stages: Specification, Conceptualization, Formalization, Implementation, Integration and Evolution. Each stage is composed of activities, in each activity artifacts are generated and participants are related to process stages and activities. Based on the artifact evolution, the development of the ontology occurs. At implementation stage, the ontology must be then integrated to other ontologies through mechanisms of ontological correspondence. The development process is centered in an evolutionary model, therefore, the cycles can be repeated at each evolution of the ontology. To validate the proposal, a case study in the biological domain was done. The case study had two cycles: during the first one the reengineering process of the CELO ontology was made and in the latter the ontology was expanded with the Human Disease sub domain. All the knowledge modeling process and the multidisciplinary team tasks are detailed and they were supported by experts of the Biology domain. For the ontological correspondence, the alignment mechanism was adopted and the A3O tool was built to generate the equivalent links and terms between ontologies described in OBO and OWL.

Keywords: Ontology, Development Process, Ontological Correspondence.

LISTA DE FIGURAS

Figura 1	Classificação de ontologias de acordo com a sua generalidade [GUARINO, N. 1998].....	12
Figura 2	Estruturação do documento OBO.....	16
Figura 3	Visão geral do OBO-Edit.....	21
Figura 4	Visualização de classes da ontologia no OBO-Edit.....	21
Figura 5	Exemplo de relações da ontologia no OBO-Edit.....	22
Figura 6	Visão geral da <i>Cell Ontology</i> enfatizando o termo <i>hepatocyte</i>	30
Figura 7	Subontologias da ontologia ChEBI [DEGTYARENKO, K. <i>et al</i> 2007]	31
Figura 8	Relações da ontologia ChEBI [DEGTYARENKO, K. <i>et al.</i> 2007]	32
Figura 9	Representação das principais classes da ontologia <i>Anatomy taxonomy</i> (At) [ROSSE, C. e MEJINO, J. L. V. 2003].....	34
Figura 10	Taxonomia da classe <i>cell</i> da ontologia FMA [ROSSE, C. e MEJINO, J. L. V. 2007].....	34
Figura 11	Exemplos de representação GO. (a) ontologia para processo biológico; (b) ontologia para função molecular; (c) ontologia para componente celular [ASHBURNER, M. <i>et al.</i> 2000].....	37
Figura 12	Parte da ontologia SO mostrando como termos e relacionamentos são utilizados em conjunto para descrever os conhecimentos sobre seqüência. As relações <i>kind_of</i> são representadas por setas marcadas com 'i', <i>part_of</i> usam setas com "P" e <i>derives_from</i> 'd'. Por exemplo, um <i>Exon</i> é <i>part_of</i> um <i>Transcript</i> , um tRNA é um <i>kind_of</i> ncRNA que é um <i>kind_of</i> <i>Processed_Transcript</i> . [EILBECK, K. <i>et al.</i> 2005].....	38
Figura 13	Operadores EM (x e y referem-se a duas partes) [EILBECK, K. <i>et al</i> 2005]	39
Figura 14	Taxonomia da ontologia CARO [HAENDEL, M. A. <i>et al.</i> 2006].....	40
Figura 15	Hierarquia da ontologia PRO [NATALE, D. A. <i>et al.</i> 2007]	42
Figura 16	Integração de estruturas e dados seqüenciais na RnaO [LEONTIS, N. B. <i>et al.</i> 2006].....	43
Figura 17	Níveis de abstração do modelo [MATOS, E. E. S. 2008].....	45
Figura 18	Estrutura da classe <i>S IEntity</i> da ontologia CelO.....	46
Figura 19	Estrutura da classe <i>DomainEntity</i> da ontologia CelO.....	46
Figura 20	Estrutura da classe <i>ModelEntity</i> da ontologia CelO.....	47
Figura 21	Estágios da metodologia de Uschold e King [BREITMAN, K. K. <i>et</i>	

	al 2007].....	51
Figura 22	Etapas do método TOVE [CHENINE, M. <i>et al</i> 2006]	52
Figura 23	Ciclo de vida de uma ontologia na metodologia METHONTOLOGY [CORCHO, O. <i>et al.</i> 2004]	55
Figura 24	Iterações entre atividades nas fases do processo KUP	60
Figura 25	Etapas da metologia DILIGENT [TEMPICH, C. 2006]	62
Figura 26	Classificação das técnicas de correspondência ontológica [EUZENAT, J. SHVAIKO, P. 2007]	71
Figura 27	Combinação de ontologias [SOUZA JUNIOR, H. C. 2008]	73
Figura 28	Alinhamento de ontologias [SOUZA JUNIOR, H. C. 2008]	74
Figura 29	Integração de ontologias [SOUZA JUNIOR, H. C. 2008]	75
Figura 30	Mapeamento de ontologias [SOUZA JUNIOR, H. C. 2008]	76
Figura 31	Comparação de dois conjuntos de propriedades [EUZENAT, J. SHVAIKO, P. 2007]	79
Figura 32	Processo de alinhamento ontológico [EHRIG, M., STAAB, S. 2004]	83
Figura 33	A Infraestrutura do PROMPT e interações entre suas ferramentas.	86
Figura 34	Arquitetura GLUE [DOAN <i>et al.</i> , 2004] in [EUZENAT, J. SHVAIKO, P. 2007]	87
Figura 35	Estratégia para alinhamento taxonômico de ontologias [FELICÍSSIMO, C H. 2004]	88
Figura 36	Componentes da abordagem QDAontology	93
Figura 37	Representação da etapa de Especificação na abordagem QDAontology	95
Figura 38	Representação da etapa de Conceitualização na abordagem QDAontology	97
Figura 39	Representação da etapa de Formalização na abordagem QDAontology	98
Figura 40	Representação da etapa de Implementação na abordagem QDAontology	110
Figura 41	Representação da etapa de Integração na abordagem QDAontology	102
Figura 42	Representação da etapa de Evolução na abordagem QDAontology	103
Figura 43	Modelo de documento de requisitos para a ontologia <i>Chemicals</i> [adaptado de FERNÁNDEZ-LÓPEZ, M. <i>et al.</i> 1999]	108
Figura 44	Modelo de documento do glossário de termos [adaptado de FERNÁNDEZ-LÓPEZ, M. <i>et al.</i> 1999]	108
Figura 45	Modelo de documento do dicionário de conceitos [adaptado de FERNÁNDEZ-LÓPEZ, M. <i>et al.</i> 1999]	109

Figura 46	Modelo do diagrama de relacionamentos [adaptado de FERNÁNDEZ-LÓPEZ, M. <i>et al.</i> 1999]	109
Figura 47	Árvore de classificação dos conceitos [adaptado de FERNÁNDEZ-LÓPEZ, M. <i>et al.</i> 1999]	110
Figura 48	Documento da implementação. Código gerado através do ambiente gráfico para edição de ontologias	110
Figura 49	Exemplo de documento de integração [adaptado de FERNÁNDEZ-LÓPEZ, M. <i>et al.</i> 1999]	111
Figura 50	Exemplo de representação gráfica de ontologias [CELL TYPE 2010]	111
Figura 51	Representação intermediária para a ontologia CeIO, parte do glossário de termos	119
Figura 52	Representação intermediária para a ontologia CeIO, dicionário parcial de conceitos	119
Figura 53	Representação intermediária para a ontologia CeIO: (a) parte de uma árvore de classificação dos conceitos; (b) um diagrama de relacionamento binário de conceitos	120
Figura 54	Tela da implementação, criação de subclasse no ambiente Protégé [PROTÉGÉ 2010]	121
Figura 55	Tela da implementação, edição de classes no ambiente Protégé [PROTÉGÉ 2010]	122
Figura 56	Ontologia CeIO após a etapa de Integração	124
Figura 57	Glossário de termos produzido na Conceitualização do ciclo de expansão da CeIO	127
Figura 58	Dicionário de conceitos produzido na Conceitualização do ciclo de expansão da CeIO	128
Figura 59	Representação intermediária para a ontologia CeIO Human Disease: árvore de classificação parcial dos conceitos	129
Figura 60	Representação intermediária para a ontologia CeIO-Human Disease: diagrama de relacionamento parcial dos conceitos	129
Figura 61	Regras obtidas durante a etapa de Formalização	129
Figura 62	Construção da ontologia CeIO – Human Disease através do ambiente Protégé apresentando todas as informações relacionadas à classe <i>Helminths</i>	130
Figura 63	Codificação parcial da ontologia CeIO - Human Disease	131
Figura 64	Diagrama da ferramenta A3O	134
Figura 65	Interface da ferramenta A3O para a escolha das ontologias para o alinhamento	134
Figura 66	Similaridades encontradas utilizando a ferramenta A3O para o termo DISEASE da ontologia CeIO – Human Disease em relação à ontologia Human Diseases da linguagem OBO	136

Figura 67	Similaridades encontradas utilizando a ferramenta A3O para o termo FASCIOLIASIS da ontologia CeLO – Human Disease em relação à ontologia Human Diseases da linguagem OBO	136
Figura 68	Adicionando termos equivalentes entre as ontologia CeLO – Human Disease e Human Disease da linguagem OBO	137

LISTA DE QUADROS

Quadro 1	Conjunto de <i>tags</i> para a <i>tag header</i>	16
Quadro 2	Conjunto de <i>tags</i> para a <i>tag stanza term</i>	17
Quadro 3	Conjunto de <i>tags</i> exclusivas para a <i>tag stanza Typedef</i>	18
Quadro 4	Conjunto de <i>tags</i> para a <i>tag stanza instance</i>	18
Quadro 5	Classificação das atividades TOVE [PINTO, H. S., MARTINS, J. P. 2004]	53
Quadro 6	Classificação das atividades METHONTOLOGY [PINTO, H. S., MARTINS, J. P. 2004]	55
Quadro 7	Análise comparativa entre metodologias e métodos estudados para a identificação das etapas comuns nos processos de desenvolvimento de ontologias	65
Quadro 8	Características e Medidas de Similaridade para Diferentes Tipos de Entidade em NOM [EHRIG, M., STAAB, S. 2004]	82
Quadro 9	Etapas de Especificação do QDAontology	95
Quadro 10	Etapas de Conceitualização do QDAontology	95
Quadro 11	Etapas de Formalização do QDAontology	98
Quadro 12	Etapas de Implementação do QDAontology	99
Quadro 13	Etapas de Integração do QDAontology	101
Quadro 14	Etapas de Evolução do QDAontology	103
Quadro 15	Características e Subcaracterísticas de Qualidade para Ontologias	112
Quadro 16	Especificação da ontologia CelO	117
Quadro 17	Documento de Integração do primeiro ciclo	123
Quadro 18	Especificação da expansão da ontologia CelO (versão 1)	125
Quadro 19	Especificação da expansão da ontologia CelO (versão 2)	126
Quadro 20	Documento de Integração do segundo ciclo	132

SUMÁRIO

1	INTRODUÇÃO	1
1.1	JUSTIFICATIVA	1
1.2	OBJETIVOS	3
1.3	METODOLOGIA	4
1.4	ESTRUTURA DO TRABALHO	4
2	ONTOLOGIAS NA ÁREA BIOLÓGICA	6
2.1	DEFINIÇÕES	7
2.2	PRINCIPAIS CONCEITOS	7
2.2.1	Classes	8
2.2.2	Relações	8
2.2.3	Instâncias	10
2.3	CLASSIFICAÇÃO DE ONTOLOGIAS	10
2.3.1	Classificação de Acordo com o Espectro Semântico	11
2.3.2	Classificação de Acordo com sua Generalidade	12
2.3.3	Classificação de Acordo com a Informação Representada	13
2.4	LINGUAGENS PARA O DESENVOLVIMENTO DE ONTOLOGIAS	13
2.4.1	Linguagem OWL	14
2.4.2	Linguagem OBO	15
2.5	FERRAMENTAS PARA A EDIÇÃO DE ONTOLOGIAS	19
2.5.1	Editor Protégé	19
2.5.2	Editor OBO- <i>Edit</i>	20
2.6	QUALIDADE DE ONTOLOGIAS	22
2.7	ONTOLOGIAS DE DOMÍNIO BIOLÓGICO	26
2.7.1	Ontologias OBO <i>Foundry</i>	27
2.7.2	CeO (<i>Cell Component Ontology</i>)	44
2.8	COMENTÁRIOS FINAIS	47
3	METODOLOGIAS E MÉTODOS PARA A CONTRUÇÃO DE ONTOLOGIAS	48
3.1	METODOLOGIA DE USCHOLD E KING	50
3.2	METODOLOGIA TORONTO VIRTUAL ENTERPRISE METHODODO (TOVE)	51

3.3	METHONTOLOGY	53
3.4	MÉTODO CYC	56
3.5	METODOLOGIA ON-TO-KNOWLEDGE	56
3.6	UM FRAMEWORK PARA ONTOLOGIAS MULTILÍNGÜES	58
3.7	PROCESSO KUP	59
3.8	METODOLOGIA DILIGENT	60
3.9	MÉTODO KACTUS	62
3.10	MÉTODO 101	63
3.11	CONSIDERAÇÕES SOBRE AS METODOLOGIAS E MÉTODOS PARA O DESENVOLVIMENTO DE ONTOLOGIAS	65
4	INTEROPERABILIDADE ENTRE ONTOLOGIAS	67
4.1	CORRESPONDÊNCIA ONTOLÓGICA	69
4.2	MECANISMOS DE INTEROPERABILIDADE	72
4.2.1	Combinação	72
4.2.2	Alinhamento	73
4.2.3	Integração	74
4.2.4	Mapeamento	75
4.3	MEDIDAS DE SIMILARIDADE	76
4.3.1	Técnica baseada no nome da entidade	77
4.3.2	Técnica baseada na estrutura da entidade	78
4.3.3	Técnica baseada na extensão da entidade	79
4.3.4	Técnica baseada na semântica da entidade	80
4.4	TECNOLOGIAS DE ALINHAMENTO	80
4.4.1	NOM – Naive Ontology Mapping	81
4.4.2	QOM – Quick Ontology Mapping	82
4.4.3	FOAM – Framework for Ontology Alignment and Mapping	83
4.4.4	PROMPT	84
4.4.5	GLUE	86
4.4.6	CATO	87
4.4.7	Análise das Tecnologias de Alinhamento	89
5	QDAontology – UMA ABORDAGEM PARA O DESENVOLVIMENTO DE ONTOLOGIAS EM E-SCIENCE	91
5.1	QDAontology - QUALITY DRIVEN APPROACH FOR E-SCIENCE ONTOLOGIES	92
5.1.1	Descrição das Etapas	94

5.1.2	Descrição das Atividades	103
5.1.3	Participantes do Processo	106
5.1.4	Artefatos de documentação	107
5.1.5	Características de Qualidade	112
5.2	TRABALHOS RELACIONADOS	114
6	APLICAÇÃO DA ABORDAGEM QDAontology: ESTUDO DE CASO EM BIOLOGIA	116
6.1	CICLO 1 – REENGENHARIA DA ONTOLOGIA CeLO	116
6.2	CICLO 2 – EXPANSÃO DA CeLO EM CeLO – Human Disease.	124
6.3	DESCRIÇÃO DA FERRAMENTA A3O PARA ALINHAMENTO ENTRE OBO E OWL	133
6.3.1	Implementação	137
6.3.2	Documentação da ferramenta A3O	137
6.4	AVALIAÇÃO DA APLICAÇÃO DA ABORDAGEM QDAontology	139
7	CONSIDERAÇÕES FINAIS	141
	REFERÊNCIAS	145
	APÊNDICE	153
	Apêndice 1 – Artefatos gerados durante a aplicação da abordagem QDAontology	153
	Apêndice 2 – Versões das ontologias obtidas durante o ciclo de expansão da ontologia CeLO em CELO – Human Disease	174
	Apêndice 3 – Questionário de validação para a ontologia CELO – Human Disease.....	178

1 INTRODUÇÃO

1.1 JUSTIFICATIVA

A tecnologia da informação vem revolucionando o modo como diversas áreas da ciência estão sendo conduzidas. A utilização de recursos computacionais no desenvolvimento da pesquisa beneficia o trabalho das comunidades científicas facilitando o compartilhamento de dados e serviços computacionais, além de contribuir para a construção de uma infraestrutura de dados e de uma comunidade científica distribuída [LUDASCHER, B *et al.*, 2006]. Este contexto, em que a computação se torna parte integrante e imprescindível para o sucesso na realização de pesquisas científicas das mais variadas áreas, é o contexto de *e-Science*, em que a ciência é realizada com o apoio computacional, se tornando assim, mais eficiente.

O termo *e-Science* geralmente é empregado para descrever o desenvolvimento de infraestruturas de serviços de software capazes de prover acesso a facilidades remotas, recursos computacionais distribuídos, armazenamento de informações em bancos de dados dedicados, disseminação e compartilhamento de dados, resultados e conhecimento. Em essência, segundo [LAUSCHNER, T 2005]

Um ambiente de *e-Science* deve permitir o compartilhamento de recursos coordenados em larga-escala entre comunidades dinâmicas de indivíduos, grupos, laboratórios e instituições, permitindo o gerenciamento cooperativo de facilidades (equipamentos, instrumentação, experimentos entre outros) e análise colaborativa de produtos (dados) oriundos dessas facilidades.

São características de *e-Science* o acesso a uma vasta coleção de dados, utilização de recursos computacionais em larga escala, utilização de recursos heterogêneos e dinâmicos de múltiplas organizações e *workflows* [CARDOSO, J. 2007] [MATOS, E. E. S. *et al.*, 2007].

Considerando este cenário, tecnologias como agentes, ontologias e serviços Web semânticos podem ser utilizadas como base para a composição de uma infraestrutura de apoio a *e-Science*. Na área da Ciência da Computação, uma ontologia define uma especificação formal e explícita dos termos de um domínio e das relações entre eles [GRUBER, T. R 1995]. Uma ontologia provê um mecanismo

para capturar a compreensão comum sobre objetos e seus relacionamentos, em certo domínio de interesse, e para prover um modelo formal e manipulável desse domínio. O uso de ontologias em aplicações de *e-Science* pode ser aplicado em diversos contextos tais como: auxiliar na troca de informações entre aplicações científicas que trabalham em domínios correlatos, permitir a descoberta semântica de serviços Web e auxiliar na composição de serviços Web em *workflows* científicos [PALAZZI, D. *et al.*, 2009].

O Núcleo de Pesquisa em Qualidade de Software - NPQS - nasceu de parcerias interinstitucionais e vem trabalhando com ontologias desde 2000. Os trabalhos se iniciaram com uma ontologia para classificação de software agropecuário em parceria com o Núcleo Softex Agrosoft, Embrapa-CNPTIA, Embrapa-CNPGL, UFV e UFLA [BRAGA, R. M. M., CAMPOS, F. C. A. 2004]. Na sequência o grupo de se dedicou, em parceria com a UERJ, ao desenvolvimento de uma ontologia para os ambientes virtuais de aprendizagem [SANTOS, N. *et al.*, 2008].

As pesquisas deste grupo relacionadas ao desenvolvimento de infraestrutura para *e-Science* tiveram início em 2007, com o desenvolvimento do *broker MathWS* [MATOS, E. E. S. *et al.* 2007] e da elaboração da dissertação de Ely da Silva Matos [MATOS, E. E. S. 2008], onde também foi desenvolvida a ontologia CelO, a qual é aplicada à representação de modelos biológicos descritos em CellML [CELLML 2010], particularmente modelos em eletrofisiologia cardíaca.

Essa infraestrutura denominada ASOW-Science [MATOS, E. E. S *et al.*, 2009] usa conceitos de agentes, serviços Web semânticos e repositórios de ontologias com o objetivo de prover um *framework* para registro, busca, recuperação, composição e execução de aplicações científicas utilizando semântica. A partir da experiência do grupo em trabalhar no desenvolvimento de ontologias para domínios diferentes como Matemática, Educação, Geoprocessamento, Agropecuária e Biologia, é que se percebeu a necessidade de desenvolvimento de uma abordagem para o desenvolvimento de ontologias no contexto de *e-Science* [SILVA, L. *et al.*, 2009].

Uma ontologia para projetos de *e-Science* possui características comuns a qualquer ontologia, entretanto, os domínios científicos são bem-demarcados, a evolução é constante, em geral ela é usada em conjunto com outras ontologias já existentes, e deve ser interoperável para que seja utilizada em diferentes aplicações

e por diferentes grupos de cientistas. Nestas aplicações, as ontologias, em geral, se caracterizam por serem desenvolvidas por equipes multidisciplinares, onde os conteudistas pertencem à área do domínio da aplicação e os construtores são Engenheiros Ontológicos oriundos da Ciência da Computação.

1.2 OBJETIVOS

Essa dissertação tem por objetivo o desenvolvimento e a avaliação de uma abordagem metodológica que permita a corretude e a completude da ontologia em desenvolvimento. No contexto diversificado de aplicações em *e-Science* o desafio não é esgotar as possibilidades, mas chegar a um modelo instanciável e que atenda às necessidades dos participantes dessa área.

A abordagem QDAontology - Quality Driven Approach for *e-Science* Ontologies, proposta nesta dissertação para a construção de ontologias no domínio da *e-Science*, é composta por seis etapas. Para cada etapa o modelo descreve (i) participantes, (ii) atividades, (iii) artefatos (iv) características de qualidade. Cada etapa é constituída de atividades, em cada atividade são gerados artefatos e os participantes estão relacionados com as etapas e atividades do processo. A partir da evolução dos artefatos ocorre o desenvolvimento da ontologia. Com a implementação da ontologia a mesma pode ser integrada a outras ontologias, através de mecanismos como alinhamento ou mapeamento, e, com base nos resultados obtidos, a ontologia pode sofrer uma evolução e produzir o resultado desejado. O processo de desenvolvimento adotado é centrado em um modelo evolutivo e, portanto, os ciclos podem se repetir a cada evolução da ontologia.

Como objetivos específicos essa dissertação pretende-se desenvolver uma ferramenta para a realização do alinhamento entre ontologias descritas nas linguagens OWL e OBO e ainda disponibilizar as ontologias CeIO e CeIO-Human Disease de acordo com padrões interoperáveis.

1.3 METODOLOGIA

A metodologia de pesquisa incluiu a revisão bibliográfica, a proposta de uma abordagem para desenvolvimento de ontologias no contexto de *e-Science* e a sua avaliação através de um estudo de caso na área da Biologia.

Na revisão bibliográfica foram revistos os conceitos, metodologias, métodos e ferramentas que permitiram caracterizar o desenvolvimento de ontologias científicas.

Após essa caracterização foi feita a proposta da abordagem QDAontology e sua avaliação através de um estudo de caso. Um estudo de caso, segundo [GOMEZ, G. R. *et al*, 1999, pp95-96], pode ser entendido como uma “estratégia de desenho de investigação”. Por centrar em uma única situação, programa ou fenômeno concreto, os autores consideram ser um método muito útil para a análise de problemas práticos, situações ou acontecimentos que surgem no cotidiano. A utilização de desenho de um único caso é justificada pelas seguintes razões: Pelo seu caráter crítico, permite confirmar, alterar, modificar ou ampliar o conhecimento sobre o objeto de estudo; O caráter único e peculiar de cada sujeito que intervém em um contexto educativo justifica, por si mesmo, este tipo de desenho; Pelo caráter revelador de uma situação concreta.

1.4 ESTRUTURA DO TRABALHO

O presente trabalho está organizado em sete capítulos. No Capítulo 2, é feita uma revisão sobre ontologias. Para tanto, são apresentadas definições, conceitos, classificações, linguagens e ferramentas para o desenvolvimento de ontologias. O Capítulo aborda também a qualidade de ontologias, sugerindo atributos de qualidade fundamentais que uma ontologia deve apresentar. E ainda, uma descrição de todas as ontologias envolvidas na abordagem proposta é apresentada.

O Capítulo 3 refere-se à revisão bibliográfica acerca das metodologias e métodos existentes para o desenvolvimento de ontologias. São apresentadas dez

metodologias cuja seleção foram baseadas nas referências da literatura e nos critérios de relevância das mesmas.

No Capítulo 4 são analisados os principais mecanismos de interoperabilidade entre ontologias, Combinação, Integração, Alinhamento e Mapeamento, os quais possibilitam a troca de informações e o reuso do conhecimento fornecido pelas ontologias. Medidas de similaridade são introduzidas devido a sua importância durante a realização dos mecanismos de interoperabilidade. Aplicar uma medida de similaridade garante aos resultados produzidos um alto grau de confiabilidade e uma maior eficiência. E finalmente, tecnologias para alinhamento entre ontologias são apresentadas.

O Capítulo 5 é destinado a apresentação da abordagem QDAontology-Quality Driven Approach for e-Science Ontologies, proposta nesta dissertação para a construção de ontologias no domínio da e-*Science*. Uma visão geral da abordagem é apresentada, detalhando as etapas do processo e suas atividades, os participantes envolvidos e os artefatos gerados nas etapas e atividades. As características de qualidade essenciais para uma ontologia são também destacadas.

No Capítulo 6 é feita a avaliação da QDAontology. O estudo de caso foi dividido em dois ciclos – Reengenharia da CeLO e Expansão da CeLO para o domínio de doenças humanas, resultando na CeLO-Human Disease. O primeiro com a finalidade de se fazer a reengenharia da CeLO e gerar os artefatos para sua documentação e o segundo com o objetivo de expandir o domínio da CeLO. O estudo de caso foi apoiado pelo mecanismo de interoperabilidade entre ontologias denominado alinhamento. Entretanto, pela inexistência de um mecanismo que possibilitasse o alinhamento entre ontologias desenvolvidas nas linguagens OWL e OBO, a ferramenta A3O foi construída a fim de apoiar a concretização do estudo de caso. Detalhes de sua implementação são apresentadas no capítulo.

Finalmente, no Capítulo 7 são feitas as considerações finais e a sugestão para trabalhos futuros.

2 ONTOLOGIAS NA ÁREA BIOLÓGICA

A origem de ontologias em Ciência da Computação pode ser referenciada por volta de 1991, no contexto da *DARPA Knowledge Sharing Effort*. Este projeto teve como objetivo conceber novas formas de construção de sistemas baseados em conhecimento, de modo que as bases de conhecimento não precisassem ser construídas a partir do zero, mas por componentes reutilizáveis. Desde que passaram a ser utilizadas, ontologias têm passado por progressos significativos. Elas podem ser usadas para o desenvolvimento de um grande número de aplicações como na gerência do conhecimento, processamento da linguagem natural, *e-commerce*, integração de informação, recuperação de informação, integração de dados, e áreas de aplicação como bioinformática, educação e sistemas biológicos [CORCHO, O. *et al.* 2007] [LAMBRIX, P. *et al.* 2007].

Ainda de acordo com [CORCHO, O. *et al.* 2007], o surgimento da *Web Semântica* tem causado uma crescente necessidade de se reutilizar conhecimento, e tem reforçado o seu potencial ao mesmo tempo. Por isso, ontologias têm um importante papel neste contexto e são utilizadas para fornecer uma forma de compreender o significado semântico das informações. A ontologia representa computacionalmente o conhecimento humano e permite o compartilhamento e o reuso desse conhecimento.

Ontologias fornecem um vocabulário comum sobre um domínio, promovem a interoperabilidade entre sistemas, servem como base para a integração e consulta de modelos de fontes de informações. Apresenta como benefícios de sua utilização o reuso, o compartilhamento e portabilidade de conhecimento, e melhorar a manutenibilidade, a documentação e a confiabilidade [LAMBRIX, P. *et al.* 2007].

De acordo com [SMITH, B. 2008], pode-se apontar algumas características do desenvolvimento de ontologias para fins científicos: (1) desenvolvida por recursos comuns a qualquer ontologia, (2) representa domínios científicos bem-demarcados, (3) sujeita a manutenção constante do domínio por especialistas, (4) concebida para ser usada em conjunto com outras ontologias complementares, e (5) independente do formato e da execução.

Ontologias são expressas em uma linguagem própria e seus termos ou conceitos podem estar relacionados entre si através de propriedades criadas ao

longo de sua criação. O seu desenvolvimento pode ser realizado em uma ferramenta apropriada e seguir um modelo de processo de desenvolvimento. Este capítulo tem como objetivo fazer uma revisão sobre ontologias. Para isso serão apresentadas algumas definições, conceitos, classificações, linguagens e ferramentas para o desenvolvimento de ontologias.

2.1 DEFINIÇÕES

Atualmente, na literatura podem ser encontradas várias definições sobre ontologias. Segundo [GUARINO, N. 1997], uma ontologia representa um vocabulário comum de um domínio. Em um contexto mais simples representa uma hierarquia especificando classes e o relacionamento entre elas.

Uma outra definição pode ser encontrada em [OBO-Edit 2010], onde ontologias representam domínios de conhecimento através da definição de entidades do domínio (e os termos usados para se referir a elas), bem como a forma como as entidades estão ligadas umas às outras, proporcionando assim, um vocabulário para comunicar os conhecimentos sobre um tópico, bem como uma representação computável de uma realidade subjacente.

Ontologias biológicas definem os termos básicos e relações em domínios biológicos e são usadas como base para a interoperabilidade entre sistemas, e para busca, integração e troca de dados biológicos [LAMBRUX, P. *et al.* 2007]. Muitas ontologias, incluindo as pertencentes ao [OBO FOUNDRY 2010], modelam conceitos em termos de classes (tipos de entidades) e relações (as características de uma entidade).

2.2 PRINCIPAIS CONCEITOS

Existem diferentes formalismos de representação do conhecimento (e linguagens correspondentes) para a formalização (e aplicação) das ontologias. Cada um deles oferece diferentes componentes que podem ser utilizados para essas tarefas. No

entanto, eles partilham um conjunto mínimo de componentes [CORCHO, O. *et al.* 2007].

A seguir serão definidos os componentes básicos para a determinação de uma ontologia. Como essa dissertação aborda ontologias de domínio biológico, para cada componente, será apresentada, além das definições comuns em qualquer ontologia, uma definição para o domínio em estudo.

2.2.1 Classes

Classes representam conceitos. Elas, normalmente, são organizadas na ontologia em taxonomias através das quais mecanismos de herança podem ser aplicados. Metaclasses também podem ser definidas, são classes cujas instâncias são classes [CORCHO, O. *et al.* 2007].

Uma outra definição para classes pode ser encontrada em [OBO-Edit. 2010]. As classes podem também ser referenciadas por “termos”, elas modelam tipos de objetos do mundo real. É importante ressaltar que classes modelam “tipos” e não instâncias. Elas também podem ter relações entre si. Estas relações podem ser de tipos pré-definidos, em relação à sintaxe OBO (como *is_a*) ou podem ser personalizadas. As classes definidas na sintaxe OBO são análogas às classes OWL ou DAML + OIL.

2.2.2 Relações

As relações representam um tipo de associação entre conceitos do domínio. Elas são formalmente definidas como qualquer subconjunto de um produto de n conjuntos, isto é:

$R \subset C_1 \times C_2 \times \dots \times C_n$. Ontologias geralmente contêm relações binárias. O primeiro argumento é conhecido como o domínio (*domain*) da relação, e o segundo argumento é o *range*.

Relações binárias são utilizadas para exprimir conceito de atributos (também chamadas de *slots*). Atributos são normalmente distinguidos de relações por que o *range* é um *datatype* (*string*, *int*), enquanto que o *range* de relações é um conceito [CORCHO, O. *et al.* 2007].

Em [OBO-Edit 2010], as relações (também conhecidas por relações de tipos ou propriedades) modelam tipos de relações entre as entidades. Elas podem ainda ser aplicadas de duas maneiras: (1) relações “nível classe”, que se referem a duas classes, e (2) relações “nível instância”, que se referem a duas instâncias.

As relações na sintaxe OBO, assim como em outras sintaxes, apresentam atributos próprios:

- **Direcionalidade:** são relações aplicadas num único sentido. As relações *has_part* e *part_of* são exemplos desse tipo de relação.
- **Simétricas:** são relações aplicadas em ambos os sentidos. Se P é uma relação simétrica, e A tem um relacionamento P para B, então B tem também a relação P para A.
- **Transitivas:** uma relação é transitiva se mantêm relações deste tipo em toda uma cadeia de ligações. Se P é uma relação simétrica, e A tem um relacionamento de P para B, e B tem um relacionamento de P para C, então por definição A tem um relacionamento de P para C. Por exemplo, a relação "*is part of*" é uma relação simétrica. Se *A is part of B*, e *B is part of C*, *A is part of C*.
- **Cíclicos:** um ciclo de uma determinada relação P pode conter outros tipos de relacionamento em P. O ciclo pode incluir ligações *is_a* ou subrelações de P. Relações cíclicas muitas vezes são as que envolvem algum sentido da mudança ao longo do tempo.
- **Domain e Range:** O *domain* e o *range* de uma relação implicam em certas relações *is_a* para termos que tem uma relação de um determinado tipo ou são alvo de uma relação de um determinado tipo. Se a relação P tem *domain* D, algum termo com uma relação do tipo P para algum outro termo é por definição uma subclasse de D. Se uma relação P tem *range* R, algum termo que é o *range* de uma relação do tipo P é por definição uma subclasse de R. Esta definição de *domain* e *range* é idêntico ao utilizado

pela linguagem OWL. Para um melhor entendimento, digamos que possamos definir uma relação *has_pet*. A relação *has_pet* tem o *domain* "pessoa", porque alguém que tenha um animal de estimação deve ser uma pessoa, e o *range* "animal", porque tudo o que é o alvo de uma relação *has_pet* tem de ser um animal. Na linguagem OBO e OWL, *domain* e *range* são utilizados para inferir informações adicionais sobre classes e instâncias.

Relações podem ter relações para outras relações. Normalmente, relações só deverão ter relações *is_a* ou *inverse_of* para outras relações. Assim como classes, as relações podem ser subrelações. Se uma relação A tem um relacionamento *is_a* com uma outra relação B, A é uma subrelação de B. Qualquer relação "nível classe" ou "nível instância" do tipo A é também uma relação de tipo B.

2.2.3 Instâncias

As instâncias, ou indivíduos como também são conhecidas, representam os objetos do domínio de interesse. Instâncias são usadas para representar elementos ou indivíduos em uma ontologia [CORCHO, O. *et al.* 2007]. Em [OBO-Edit 2010] é dada uma outra definição, as instâncias representam entidades concretas que instanciam uma classe abstrata. OBO tem a capacidade de representar instâncias, mas até o presente a ferramenta OBO-Edit não permite aplicações utilizando informações sobre as instâncias.

2.3 CLASSIFICAÇÃO DE ONTOLOGIAS

Segundo [BREITMAN, K. K. *et al* 2007], ontologias podem ser classificadas em três categorias: de acordo com o espectro semântico, de acordo com sua generalidade, e de acordo com a informação representada.

2.3.1 Classificação de Acordo com o Espectro Semântico

O espectro semântico varia de catálogos de termos informais a sofisticadas ontologias, que variam dependendo do nível de formalização e expressividade, como se segue [BREITMAN, K. K. *et al* 2007]:

- Vocabulários controlados: são listas finitas de termos.
- Glossários: são listas de termos cujo significado é descrito em linguagem natural. O formato de um glossário é semelhante ao de um dicionário, em que termos estão organizados em ordem alfabética, seguidas pelas respectivas definições.
- Tesouros: são listas de termos e definições que padronizam palavras para indexação. Além de definições, um tesouro fornece também as relações entre os termos: hierárquico, associativo, ou equivalência (sinônimos).
- Hierarquias *is-a* informais: são hierarquias que usam relações de generalização (tipo de) em uma direção informal. Neste tipo de hierarquia, conceitos relacionados podem ser agregados em categorias mesmo se eles não respeitam o relacionamento de generalização.
- Hierarquias *is-a* formais: são hierarquias que respeitam plenamente o relacionamento de generalização.
- *Frames*: são modelos que incluem classes e propriedades. As primitivas do modelo são classes (ou frames), que apresentam propriedades (ou atributos). São largamente usadas na modelagem do conhecimento.
- Ontologias que expressam restrição de valor: são ontologias que fornecem construções para restringirem os valores que suas classes podem assumir.
- Ontologias que expressam restrição lógica: são ontologias que permitem restrições lógicas de primeira ordem para serem expressas.

2.3.2 Classificação de Acordo com sua Generalidade

Segundo [GUARINO, N. 1998], a classificação de ontologias baseada em sua generalidade é como se segue:

- Ontologias de alto nível: descrevem conceitos genéricos, tais como espaço, tempo e eventos. Em geral, são independentes do domínio e podem ser reutilizadas na construção de novas ontologias.
- Ontologias de domínio: descrevem um vocabulário relacionado a um domínio genérico, por meio das especificações de conceitos introduzidos nas ontologias de alto nível.
- Ontologias de tarefas: descrevem um vocabulário relacionado a uma tarefa ou a uma atividade genérica, por meio da especialização de conceitos introduzidos as ontologias de alto nível.
- Ontologias de aplicação: descrevem um vocabulário de uma aplicação específica. Esse tipo de ontologia especializa conceitos tanto das ontologias de domínio, como das de tarefas.

A Figura 1 exibe a classificação de ontologias baseada em sua generalidade.

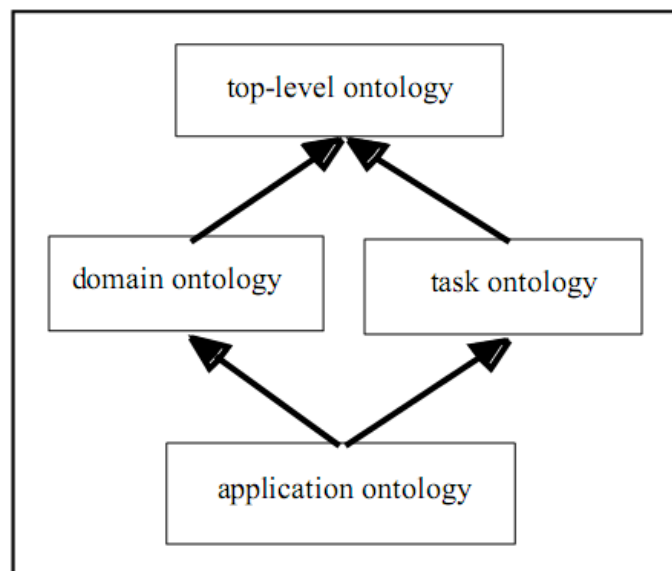


Figura 1. Classificação de ontologias de acordo com a sua generalidade [GUARINO, N. 1998].

2.3.3 Classificação de Acordo com a Informação Representada

Através do tipo de informação representada por uma ontologia são identificadas as seguintes categorias [BREITMAN, K. K. *et al* 2007]:

- Ontologias de representação do conhecimento: fornecem modelagem de elementos primitivos dos modelos de representação do conhecimento. Elas oferecem representações baseadas em classes, subclasses, atributos, axiomas.
- Ontologias genéricas e de uso comum: representam um conhecimento comum que pode ser utilizado em diferentes domínios. Geralmente, incluem um vocabulário que relaciona classes, eventos, espaço, entre outros conceitos.
- Super ontologias: descrevem conceitos gerais.
- Ontologias de domínio: oferecem conceitos que podem ser reutilizadas dentro de um domínio específico (biológico, jurídico, entre outros).
- Ontologias de tarefas: descrevem o vocabulário relacionado a uma tarefa ou atividade.
- Ontologias de domínio-tarefa: são ontologias de tarefas que podem ser reutilizadas em um domínio específico, porém, não podem ser reutilizadas genericamente em domínio similares.
- Ontologias métodos: fornecem definições para conceitos e relacionamentos relevantes para um processo.
- Ontologias de aplicação: contém todos os conceitos necessários para modelar uma aplicação. É usada para especializar e estender ontologias de domínio ou de tarefa para uma aplicação específica.

2.4 LINGUAGENS PARA O DESENVOLVIMENTO DE ONTOLOGIAS

Para que uma ontologia possa ser compreendida por máquinas e que permita extrair conhecimento, é importante representá-la de forma que isso seja possível. Para isso

é necessária uma linguagem com capacidade de expressar o conhecimento sem ambigüidades e da forma mais clara possível.

Existem atualmente várias linguagens para representar ontologias, umas mais utilizadas do que as outras. Neste trabalho serão abordadas duas linguagens em especial, as linguagens OBO e OWL, pelo fato destas serem utilizadas nas ontologias do grupo OBO Foundry [OBO Foundry 2010] e CelO [Matos, E. E. S 2008] respectivamente, que serão utilizadas no estudo de caso dessa dissertação.

2.4.1 Linguagem OWL

De acordo com [OWL 2010], a linguagem OWL - *Ontology Web Language* foi projetada para ser utilizada por aplicações que necessitam processar o conteúdo da informação em vez de apenas apresentar informações para os seres humanos. Ela é uma revisão da DAML+OIL *Web Ontology Language* incorporando melhorias necessárias da aplicação de DAML+OIL, e hoje é uma recomendação do W3C (*World Wide Web Consortium*).

OWL pode ser utilizada para representar explicitamente o significado dos termos em vocabulários e os relacionamentos entre os termos, definindo assim uma ontologia. Além disso, a linguagem tem mais facilidades para expressar o significado e a semântica do que XML [XML 2010], RDF [RDF 2010] e RDF-S [RDF-S 2010], permitindo dessa forma uma maior compreensão do conteúdo WEB pelas máquinas, já que ela adiciona mais vocabulário para descrever propriedades e classes, tais como: relações entre classes (por exemplo, *disjointness*), cardinalidade (por exemplo, *exactly one*), características de propriedades (por exemplo, simétrica).

A linguagem OWL oferece três sublinguagens mais expressivas projetadas para uso de comunidades específicas de implementadores e usuários:

- **OWL Lite:** dá suporte àqueles usuários que necessitam, principalmente, de uma hierarquia de classificação e restrições simples. Por exemplo, enquanto o OWL Lite suporta restrições de cardinalidade, ele somente permite 0 ou 1 como valores de cardinalidade. Dessa forma, é mais simples criar ferramentas que suportam o OWL Lite do que as outras

sublinguagens. Além disso, fornece um rápido caminho de migração para o tesouro e outras taxonomias.

- **OWL DL:** dá suporte para aqueles usuários que desejam o máximo de expressividade sem perder a completude computacional (todas as conclusões são garantidas de serem computadas) e capacidade de decisão (todas as computações serão finalizadas em um tempo finito) dos sistemas de raciocínio. O OWL DL inclui todos os construtores da linguagem OWL com restrições como separação entre tipos (uma classe não pode ser ao mesmo tempo um indivíduo ou tipo e uma propriedade não pode ser ao mesmo tempo um indivíduo ou uma classe). OWL DL tem esse nome devido a sua correspondência a Lógica de Descrição (*Description Logic*), um campo de pesquisa que tem estudado as lógicas que formam a base formal da OWL.
- **OWL Full:** é significativa para aqueles usuários que desejam o máximo de expressividade e a liberdade sintática do RDF, sem nenhuma garantia computacional. Por exemplo, em OWL Full uma classe pode ser tratada simultaneamente como uma coleção de indivíduos ou, simplesmente, como um indivíduo. O OWL Full permite uma ontologia aumentar o significado de um vocabulário predefinido (RDF ou OWL). É improvável que todo software de raciocínio seja capaz de suportar todas as características do OWL Full.

2.4.2 Sintaxe OBO

Na sintaxe OBO, as ontologias descritas por ela são similares às ontologias especificadas em linguagens como OWL e DAML + OIL, sendo, no entanto, ainda mais simples. As ontologias OBO foram projetadas de acordo com as necessidades da comunidade biológica. A sintaxe fornece ainda uma capacidade de monitorar uma grande quantidade de metadados. E por ser mais simples, não inclui todas as características de OWL ou DAML + OIL, e, por vezes, utiliza semânticas diferentes, mas idênticas às características de uma descrição lógica [OBO-Edit 2010].

Segundo [OBO FORMAT 2008], OBO é uma sintaxe para a representação de ontologias. A linguagem tem como objetivo alcançar as seguintes metas: legibilidade humana, fácil interpretação, extensibilidade e redundância mínima.

A sintaxe OBO é estruturada com a utilização de *tags*. Uma diferença importante é que *tags* não reconhecidas em algum contexto não geram necessariamente erros fatais, permitindo dessa forma, a leitura de arquivos que contenham informações não usadas por uma ferramenta particular.

Um documento OBO é estruturado conforme a figura 2. O cabeçalho é uma seção marcada no início do documento contendo pares de tag-valor. O cabeçalho termina quando a primeira *tag* denominada *stanza* é encontrada. A *tag stanza* é uma seção do documento que indica que um objeto de um tipo específico está sendo descrito, especificamente, isso significa que com esta *tag* são definidas as classes e as instâncias da ontologia. Esta tag suporta três tipos: *Term*, *Typedef* e *Instance*.

```
<header>

<stanza>
<stanza>
...

```

Figura 2. Estruturação do documento OBO.

O conjunto de *tags* que definem o cabeçalho, as classes e as instâncias podem ser obrigatório ou opcional. Os quadros a seguir apresentam esse conjunto de *tags* para cada uma das duas, bem como uma explicação de cada uma e alguns exemplos.

Quadro 1. Conjunto de *tags* para a *tag* header

<i>Tag Header</i>	
Tags obrigatórias	
<code>format-version</code>	Fornece a versão da especificação OBO que o arquivo utiliza. Isso é útil se a semântica da <i>tag</i> mudar de uma especificação OBO para a próxima versão.
Tags opcionais	
<code>data-version</code>	Fornece a versão atual da ontologia.
<code>version</code>	Depreciado. Usa <code>data-version</code> instanciado.
<code>date</code>	A data atual no formato dd: mm: aaaa HH: mm.
<code>saved-by</code>	O nome de usuário (<i>username</i>) de uma pessoa que salvou o arquivo pela última vez. O significado de " <i>username</i> " é inteiramente da responsabilidade do aplicativo que gerou o arquivo.

auto-generated-by	O programa que gerou o arquivo.
subsetdef	Uma descrição de um termo do subconjunto. O valor para essa <i>tag</i> deve conter um nome do subconjunto, um espaço e uma descrição de um subconjunto fechado. Exemplo: subsetdef: GO_SLIM "GO Slim".
import	Uma url OBO apontando para um outro documento. O conteúdo do documento alvo será anexado para analisar este documento em tempo de interpretação. Se o documento alvo também contém declarações de importação, eles serão solucionados. Esta <i>tag</i> substitui a <i>tag typeref</i> de versões anteriores da especificação OBO.
synonymtypedef	Uma descrição de um tipo de usuário definido sinônimo. O valor para essa <i>tag</i> deve conter um nome tipo sinônimo, um espaço, uma descrição fechada, e um escopo opcional especificador. Por exemplo: synonymtypedef: UK_SPELLING "British spelling" EXACT.
idspace	Um mapeamento entre um ID "local" e um ID "global". O valor para esta <i>tag</i> deve ser um <i>idspace</i> local, um espaço, uma URI, e opcionalmente seguido por uma. Por exemplo: idspace: GO urn:lsid:bioontology.org:GO: "gene ontology terms".
default-relationship-id-prefix	Qualquer relação que falte um ID será prefixado com o valor dessa <i>tag</i> . Por exemplo: default-relationship-id-prefix: OBO_REL
id-mapping	Mapeia um Termo ou <i>Typedef ID</i> para outro Termo ou <i>Typedef ID</i> . A principal razão para essa <i>tag</i> é aumentar a interoperabilidade entre diferentes ontologias OBO. Por exemplo: id-mapping: part_of OBO_REL:part_of.
remark	Comentários gerais para o arquivo. Esta <i>tag</i> é diferenciada de um comentário (definido com o sinal !) em que o conteúdo de uma <i>tag remark</i> está garantida para ser preservado por um interpretador.

Quadro 2. Conjunto de tags para a tag stanza term

Tag Stanza Term	
Tags obrigatórias	
id	O único id do termo corrente.
name	Qualquer termo pode ter apenas um nome definido. Se múltiplos nomes de termos são definidos ocorre um erro no interpretador.
Tags opcionais	
is_anonymous	Se um objeto tem ou não uma id anônima.
alt_id	Define um ID alternativo para o termo. Um termo pode ter números de ids alternativos.
def	A definição do termo corrente.
comment	Um comentário para o termo.
subset	Indica um subconjunto de termos para os quais o termo pertence.
synonym	Fornece um sinônimo para o termo e pode indicar uma categoria de sinônimo ou informação do escopo. Por exemplo: synonym: "The other white meat" EXACT MARKETING_SLOGAN [MEAT:00324, BACONBASE:03021]
exact_synonym	Depreciado. Um pseudônimo para a <i>tag synonym</i> com o conjunto do escopo modificado para EXACT.
narrow_synonym	Depreciado. Um pseudônimo para a <i>tag synonym</i> com o conjunto do escopo modificado para NARROW.
broad_synonym	Depreciado. Um pseudônimo para a <i>tag synonym</i> com o conjunto do escopo modificado para BROAD.
xref	Descreve um termo análogo em outro vocabulário.
xref_analog	Depreciado. Um pseudônimo para a <i>tag xref</i> .
xref_unk	Depreciado. Um pseudônimo para a <i>tag xref</i> .
is_a	Descreve uma relação entre um termo e outro. O valor é o id do termo do qual este termo é uma subclasse. Um termo pode ter vários relacionamentos <i>is_a</i> .
intersection_of	Indica que o termo representa a interseção de vários outros termos. O valor ou é o id de um termo ou um id. Por exemplo: intersection_of: GO:00001 intersection_of: part_of GO:00002
union_of	Indica que o termo representa a união de vários outros termos. O valor é o id de um dos outros termos dos quais este termo é unido.
disjoint_from	Indica que um termo é distinto de outro, significando que um termo não tem instâncias ou subclasses em comum.

relationship	Descreve um tipo de relação de um termo com outro. Por exemplo: not_necessary true OR false inverse_necessary true OR false namespace <any namespace id> derived true OR false cardinality any non-negative integer maxCardinality any non-negative integer minCardinality any non-negative integer
is_obsolete	Indica se um termo é ou não ultrapassado. Termos ultrapassados não devem ter nenhum relacionamento e nem <i>tags</i> definidas como, <i>is_a</i> , <i>inverse_of</i> , <i>disjoint_from</i> , <i>union_of</i> , ou <i>intersection_of</i> .
replaced_by	Fornece um termo que substitui um termo ultrapassado. Um único termo ultrapassado pode ter mais de uma dessa <i>tag</i> .
consider	Fornece um termo que pode ser uma substituto adequado para um termo ultrapassado, mas precisa ser cuidadosamente analisado por um especialista.
use_term	Depreciado. Equivalente a <i>tag consider</i> .
builtin	Indica se um termo ou relação foi ou não construído no formato OBO.

O conjunto de tags *Stanza Typedef*, suporta quase todas as *tags* definidas para *Stanza Term*, exceto *union_of*, *intersection_of* e *disjoint_from*. O quadro 3 define as *tags* definidas apenas para este tipo.

Quadro 3. Conjunto de *tags* exclusivas para a *tag stanza Typedef*

Tag Stanza Typedef	
domain	Indica o id de um termo, ou um identificador especial reservado, que especifica o domínio para este tipo de relação.
range	Indica o id de um termo, ou um identificador especial reservado, que especifica uma classe alvo aceitável para este tipo de relação.
inverse_of	Fornece o id de outro tipo de relação que é a inversa deste tipo de relação.
transitive_over	Fornece o id de outro tipo de relação que este tipo de relação é transitivo.
is_cyclic	Indica se este tipo de relação pode ou não ter ciclos.
is_reflexive	Se a relação é reflexiva. Todos os relacionamentos também são cíclicos. Permite valores: <i>true</i> ou <i>false</i> .
is_symmetric	Se a relação é simétrica. Todos os relacionamentos também são cíclicos. Permite valores: <i>true</i> ou <i>false</i> .
is_anti_symmetric	Se a relação é anti-simétrica. Permite valores: <i>true</i> ou <i>false</i> .
is_transitive	Se a relação é transitiva. Permite valores: <i>true</i> ou <i>false</i> .
is_metadata_tag	Se a relação é uma <i>tag</i> de metadados. Propriedades marcadas com esta <i>tag</i> são usadas para gravar objetos metadados.

O conjunto de *tags Stanza Instance*, permite algumas *tags* opcionais já definidas no conjunto *Stanza Term*, apresentando a mesma sintaxe e semântica. Elas são: *is_anonymous*, *namespace*, *alt_id*, *comment*, *xref*, *synonym*, *is_obsolete*, *replaced_by*, *consider*. O quadro a seguir apresenta as outras *tags*.

Quadro 4. Conjunto de *tags* para a *tag stanza instance*.

Tag Stanza Instance	
Tags obrigatórias	
id	O único id do termo corrente.
name	O nome da instância. Qualquer instância pode ter apenas um nome definido.
instance_of	O id do termo da classe do qual este é uma instância.
Tags opcionais	

property_value	Liga uma propriedade para um valor nesta instância. Por exemplo: [Instance] id: john name: John Day-Richter instance_of: boy property_value: married_to heather property_value: shoe_size "8" xsd:positiveInteger
----------------	---

2.5 FERRAMENTAS PARA A EDIÇÃO DE ONTOLOGIAS

Neste trabalho serão apresentadas duas ferramentas. O editor Protégé [PROTÉGÉ 2008], utilizado para o desenvolvimento da ontologia CelO, e o editor OBO-Edit empregado no desenvolvimento das ontologias do OBO *Fondry* [OBO FOUNDRY 2010].

2.5.1 Editor Protégé

De acordo com [PROTÉGÉ 2008], esta ferramenta é um editor de ontologias de código-fonte aberto e escrito em Java desenvolvido pela *Stanford Medical Informatics na Stanford University School of Medicine*. Permite a criação, visualização e manipulação de ontologias em diferentes formas de representação. As ontologias editadas nesta ferramenta podem ainda ser exportadas em uma variedade de formatos, incluindo RDF (S), OWL, e XML Schema.

Outro fator importante da ferramenta é a sua característica de ser extensível, as suas funcionalidades podem ser estendidas utilizando *plug-ins* disponíveis na internet ou podendo ser criados pelos próprios usuários. Ela também é apoiada por uma forte comunidade de desenvolvedores, além de acadêmicos, governamentais e usuários corporativos.

O Protégé é um editor de ontologias e um *framework* de bases de conhecimento, e suporta duas principais formas de modelagem de ontologias através dos editores Protégé-Frames e Protégé-OWL.

O editor Protégé-Frames permite ao usuário construir e preencher ontologias que são baseadas em frames de acordo com o protocolo OKBC (*Open Knowledge Base Connectivity*) [OKBC 2010]. Neste modelo, uma ontologia consiste em um conjunto de classes organizadas em uma classificação hierárquica para representar os conceitos importantes de um domínio, um conjunto de *slots* associados com as classes para descrever suas propriedades e relacionamentos, e um conjunto de instâncias destas classes - exemplares individuais dos conceitos que contém valores específicos para as suas propriedades.

Já o editor Protégé-OWL permite aos usuários construir ontologias para a Web Semântica, em especial utilizando a linguagem OWL especificada pela W3C. Uma ontologia desenvolvida em OWL pode incluir descrições de classes, propriedades e suas instâncias. Dada uma determinada ontologia, a semântica da OWL especifica como derivar conseqüências lógicas, ou seja, fatos que não estão explicitamente descritos na ontologia, mas apresentados pela semântica. Essas implicações podem ser baseadas em um único documento ou em múltiplos documentos que foram combinados usando os mecanismos definidos pela OWL.

2.5.2 Editor OBO-Edit

Assim como o Protégé, a ferramenta OBO-Edit é um editor de ontologias de código-fonte aberto e também foi desenvolvido em Java. Segundo [OBO-Edit 2010], foi desenvolvida inicialmente para uso exclusivo na edição Gene Ontology, sendo hoje o editor utilizado para a criação de ontologias do OBO *Foundry*. A interface é de fácil uso e é um projeto financiado pelo *Gene Ontologies Consortium*.

A Figura 3 apresenta uma visão parcial do OBO-Edit. Do lado esquerdo localiza-se o painel do editor, onde é exibida a hierarquia das classes. E do lado direito, é possível adicionar informações pertinentes a cada termo.

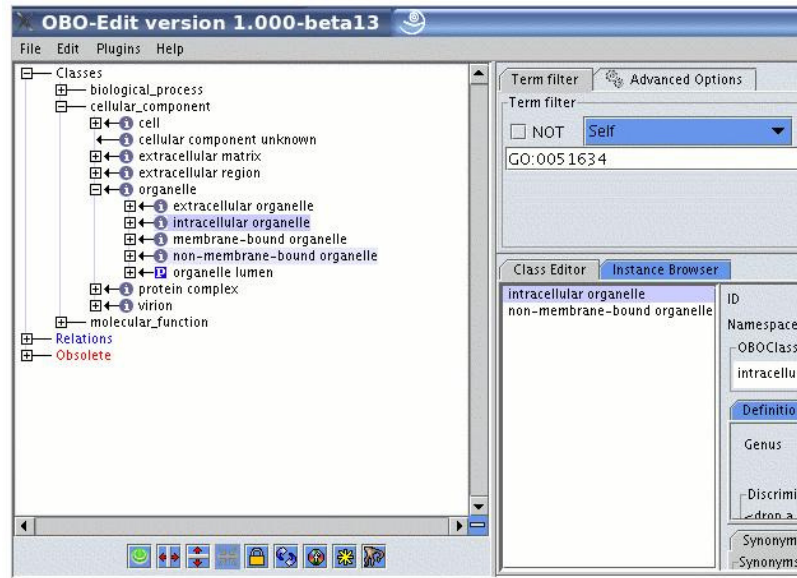


Figura 3. Visão geral do OBO-Edit.

A Figura 4 a seguir exibe o painel do editor, após a ontologia ser carregada, existem várias maneiras de visualizá-la. Este painel é a principal forma de visualização das classes na ontologia, além disso, ele sempre apresenta três nós raiz denominados "Classes", "Relations", e "Obsolete", este último para indicar termos que não são mais usados na ontologia.

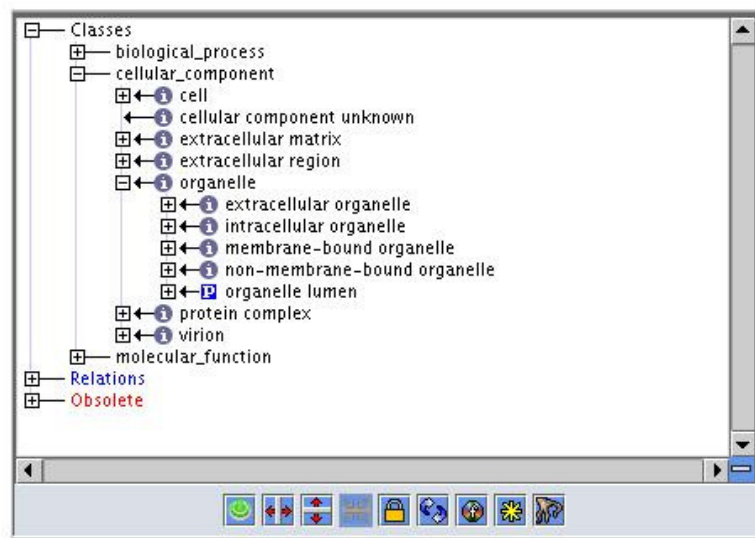


Figura 4. Visualização de classes da ontologia no OBO-Edit.

O editor exibe a ontologia como uma árvore. Cada nó na exibição representa uma relação entre os dois termos. Como pode ser observada na parte destacada da Figura 5, cada linha descreve uma relação entre um nó filho e um nó pai. A seta indica que as relações devem ser lidas da direita para a esquerda. O símbolo **i** significa que o tipo de relacionamento é *is_a* e o símbolo **P** significa que o

relacionamento é *part_of*. Dessa forma, temos que, “*extracellular organelle is_a organelle*” e “*organelle lumen part_of organelle*”.

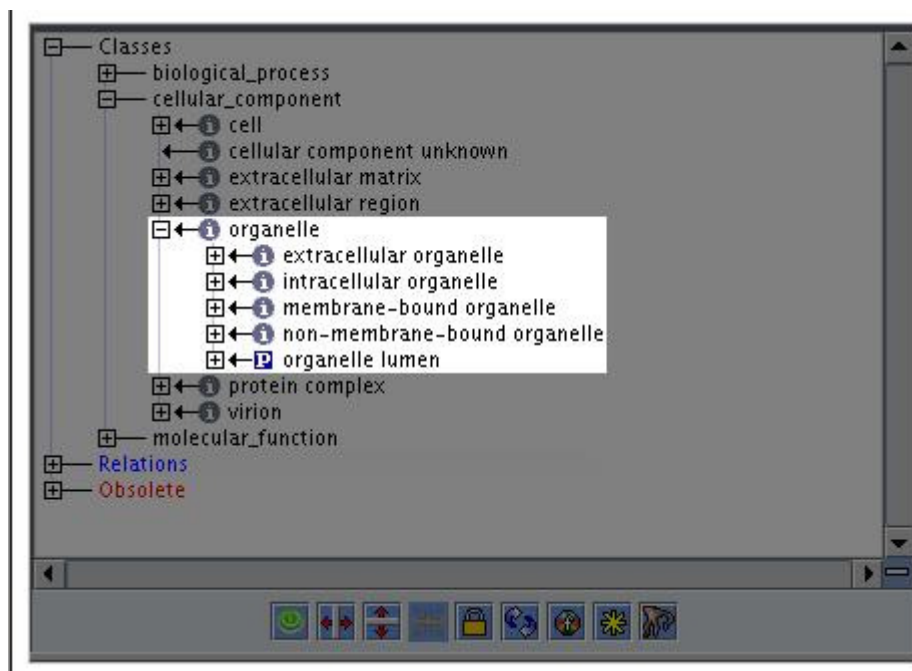


Figura 5. Exemplo de relações da ontologia no OBO-Edit.

2.6 QUALIDADE DE ONTOLOGIAS

No contexto de desenvolvimento de software, qualidade pode ser entendida como sendo a satisfação de requisitos funcionais e de desempenho explicitamente declarados e das características implícitas esperadas em todo o software desenvolvido [PRESSMAN, R. S. 2006]. A qualidade de ontologias visa obter uma ontologia que seja a mais coerente possível, e assim, descrever um domínio que seja o mais próximo da realidade. Comparando a qualidade nos contextos citados anteriormente concluímos que a qualidade depende fortemente de seu processo de desenvolvimento.

Na medida em que a utilização de ontologias, para os mais variados fins, cresce, a qualidade se torna um fator essencial no seu processo de desenvolvimento. Dessa forma o investimento em qualidade se faz necessário.

Em [CECH, P. 2009 (apud RECTOR 2005)], a ontologia é considerada como um bloco de informação/conhecimento, o que favorece a integração de elementos

e o compartilhamento e reuso do conhecimento. Segundo o autor, uma ontologia deve então apresentar as seguintes características:

- **Logicamente correta** e, portanto, adequada para inferência e uso em recuperação de informações, sistemas baseados em regras, etc. Por exemplo, todas e apenas as "doenças do coração" devem ser classificadas como "doença do coração".
- **Reutilizável** e, portanto, adequada a integração de sistemas, de comunicação, etc. As classificações resultantes devem conter detalhes refinados e suportar as várias alternativas de visualização requeridas pela união das aplicações que poderão utilizá-las futuramente.

De acordo com [CECH, P. 2009 (apud GRUBER, T. R. 1983)], a fim de permitir a interoperabilidade e ser útil na construção de sua infra-estrutura, uma ontologia deve apresentar critérios, tais como:

- **Clareza:** uma ontologia deve comunicar eficazmente o significado pretendido dos termos definidos. Definições devem ser objetivas e também documentadas em linguagem natural.
- **Coerência:** a ontologia deve ser coerente, ou seja, ela deve aprovar apenas inferências que sejam consistentes com as definições.
- **Extensibilidade:** uma ontologia deve ser projetada para fornecer o uso de vocabulário compartilhado. Ela deve ser capaz de definir novos termos para usos especiais com base no vocabulário existente, de uma forma que não seja necessário a revisão das definições existentes.
- **Dependência mínima da codificação:** a conceitualização deve ser especificada a nível de conhecimento, sem depender de um determinado nível de codificação. Essa dependência deve ser minimizada porque o conhecimento pode ser implementado em diferentes sistemas e estilos de representação.
- **Compromisso ontológico mínimo:** uma ontologia deve exigir um compromisso ontológico mínimo suficiente para apoiar o compartilhamento de conhecimento. Uma ontologia também deve atender o mínimo possível das hipóteses sobre o mundo que está sendo modelado.

O autor ainda afirma que esses critérios são importantes porque, além da ontologia ser uma parte da análise de requisitos e mapeamento do conhecimento, é

GRUBER, T. R.. A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition. 1983.

RECTOR, A. L. Untangling Taxonomies and Relationships: Personal and Practical Problems in Loosely Coupled Development of Larger Ontologies.

também uma parte necessária da futura infra-estrutura de conhecimento/informação. As ontologias devem servir como ponto de partida para a construção da base de conhecimento. E também constituem uma base para aplicações de *workflow* e sistemas de conhecimento que pertencem à infra-estrutura. As ontologias desempenham um papel integrador, que é necessário para a infra-estrutura do trabalho como uma unificadora e não como vários sistemas separados [CECH, P. 2009 (apud GRUBER, T. R. 1983)].

O trabalho de [TARTIR, S. *et al.* 2005] destaca que avaliar a qualidade de uma ontologia é importante por diversas razões, permitindo ao desenvolvedor da ontologia reconhecer automaticamente as áreas que podem precisar de mais trabalho e ao usuário saber que partes da ontologia podem causar problemas, e ainda a comparação entre diferentes ontologias.

Segundo [TARTIR, S. *et al.* 2005], a qualidade de ontologias pode ser avaliada em diferentes dimensões. Por exemplo, métricas de qualidade podem ser usadas para avaliar o sucesso de um esquema de modelagem de um domínio do mundo real, como os pesquisadores em Ciência da Computação e suas publicações. A profundidade, largura e altura do esquema de herança podem ter um papel importante na avaliação da qualidade. Além disso, a qualidade de uma ontologia popular (ou a base de conhecimento) pode ser medida para verificar se ela é uma representação rica e precisa de entidades e relações do mundo real. E, finalmente, a qualidade da base de conhecimento pode ser avaliada para ver se as instâncias e relações estão de acordo com o esquema.

Ainda de acordo com [TARTIR, S. *et al.* 2005], a categorização da qualidade de ontologias em três grupos (esquema, base de conhecimento e métricas), serve como meio para avaliar a qualidade de uma única ontologia ou comparar ontologias quando mais de uma candidata se assemelha em alguns requisitos. E afirma que, na maioria dos casos, a forma como a ontologia é construída é fortemente dependente do domínio no qual ela foi concebida. Ontologias que modelam atividades humanas (por exemplo, viagem ou terrorismo), terão características muito diferentes das que modelam o mundo natural (por exemplo, genomas ou carboidratos complexos).

Outro trabalho encontrado na literatura é o de [JANOWICZ, K., *et al.* 2008]. O autor afirma que a similaridade semântica pode ser vista como um fator indicador de potencial qualidade. As medidas de similaridade não abrangem todos os aspectos de garantia de qualidade, mas sugere que uma ontologia reflete a conceitualização

inicial e os usuários pretendidos na aplicação. Conseqüentemente, a similaridade semântica é um candidato para se avaliar a aptidão proposta na engenharia da ontologia.

[WELTY, C. 2006] afirma que ontologias de boa qualidade custam mais, porém algumas aplicações necessitam dessa característica e que melhorias na qualidade podem melhorar o desempenho em tais aplicações. No trabalho, o autor apresenta um conjunto de características de qualidade, oriundas de outros autores e de seus trabalhos anteriores, que considera serem cruciais para a obtenção de ontologias de boa qualidade. Entre elas destacamos:

- Definição abrangente, correta, rica e fiel do domínio
- Organização
- Modularidade
- Relação com a realidade
- Construção de um significado claro
- Meta-nível coerente
- Captura da estrutura invariante do domínio
- Clareza do significado, adicionando para isso, restrições para eliminar modelos não pretendidos
- Critérios de identificação claros
- Propriedades consistentes no meta-nível
- Especificação da estrutura invariante do domínio.

Finalmente, o trabalho de [HUNTER, L. *et al.* 2007] avalia a abrangência de uma ontologia com base no seu grau de sobreposição e consenso com outras ontologias, o que pode ser particularmente útil quando há subdomínios relacionados. Por exemplo, o esforço em alinhar termos ligados a partir de múltiplas ontologias identifica uma série de impropriedades nas ontologias, incluindo ligações de relações perdidas.

De acordo com [HUNTER, L. *et al.* 2007] um método para se avaliar a integridade de uma ontologia é a medição da convergência entre a taxa de determinação das necessidades de novos termos e a resolução de tais necessidades. Outro método discutido é a medida de corretude da ontologia. Segundo o autor, na literatura a maioria das métricas de qualidade de ontologias é centrada neste aspecto. Muitas medidas internas de corretude da ontologia têm sido

propostas, tanto em termos de conteúdo/estrutura da ontologia quanto em termos do processo de produção. Usualmente as métricas de corretude devem ser:

- Bem definidas: a proporção de termos em uma ontologia que têm uma definição feita exclusivamente em termos de outras ontologias que são necessárias, individualmente e coletivamente, é suficiente para definir o termo. Quando entidades em uma ontologia são definidas através de combinações automatizadas de componentes atômicos, então é possível avaliar a qualidade dos métodos de combinação através da proporção de termos que podem ser definidos por uma combinação que atualmente está incluída na ontologia, levando em consideração as razões para a exclusão de algumas combinações.
- Bem estruturadas: qual proporção da ontologia é possível mapear corretamente e reversivelmente para representações em padrões, como OWL-DL? Qual proporção de termos participa na hierarquia IS_A? Qual proporção dos termos que tem todos os metadados desejados (por exemplo, definição, sinônimos etc)?
- Avaliador independente: qual proporção de avaliadores independentes concorda que a estrutura e o conteúdo da ontologia estão corretos? Qual a proporção de especialistas de domínio concorda que a ontologia está correta?

2.7 ONTOLOGIAS DE DOMÍNIO BIOLÓGICO

A área de ciências biológicas é uma importante área de pesquisa nos dias atuais e tem apresentado um grande crescimento, contribuindo assim para a sua evolução. O avanço desses estudos aliado ao desenvolvimento de novas tecnologias tem possibilitado um melhor aproveitamento das informações geradas. Atualmente, os estudos na área de ciências biológicas estão fazendo uso de ontologias a fim de padronizar os termos e conceitos utilizados.

O domínio biológico é marcado pelo uso de um grande número de termos. Esta área apresenta inúmeras terminologias e muitas das vezes ambíguas, a mesma palavra é muitas vezes usada para descrever mais de um conceito, além da existência de muitos dialetos [EILBECK, K. *et al.* 2005]. O uso de ontologias é útil

para o apoio a estas pesquisas, possibilitando um maior entendimento do domínio tanto por especialistas como por pessoas comuns, e é também uma forma de padronização dos termos empregados. Além disso, a utilização de ontologias permite ao cientista uma maneira mais fácil de comunicação de certo domínio com outros.

Ontologias, principalmente as ontologias pertencentes ao OBO *Foundry* [OBO Foundry 2010], tem como objetivo definir formalmente um vocabulário estruturado de um domínio específico da biologia. Dessa forma, permite ao cientista o emprego de termos sem ambigüidade e melhor conhecimento do domínio estudado. No caso da ontologia CelO [Matos, E. E. S 2008], o uso de ontologias é empregado a fim de apoiar a tarefa de modelagem da eletrofisiologia cardíaca, descrevendo os modelos CellML existentes. E apesar de apresentarem objetivos diferentes, ambas as ontologias descrevem o conhecimento sobre um domínio biológico. CellML é uma linguagem de marcação baseada em XML para descrever modelos biológicos, especifica elementos que podem ser usados para representar um modelo de maneira formal, sem ambigüidades, legível por humanos e processável por máquinas [CELLML 2010].

Atualmente, existem diversas ontologias no contexto biológico com diferentes objetivos e diferentes características. Diante desse fato, esta seção tem como um dos seus objetivos, apresentar algumas dessas ontologias dando ênfase às ontologias do grupo OBO Foundry que tem se destacado nesta área no desenvolvimento de inúmeras ontologias e a ontologia CelO, que surgiu a partir de um trabalho de mestrado [MATOS, E. E. S. 2008].

2.7.1 Ontologias OBO *Foundry*

O repositório *Open Biomedical Ontologies* (OBO) foi criado em 2001 por Michael Ashburner e Suzanna Lewis como um meio de proporcionar acesso ao *Gene Ontology* (GO) e a outras ontologias da área biológica. Em 2005, O OBO *Foundry* foi iniciado por Ashburner, Lewis e Smith como um experimento colaborativo destinado a melhorar a qualidade e a interoperabilidade de ontologias tanto de conteúdo

biológico quanto da estrutura lógica. É uma iniciativa voluntária de desenvolvedores de ontologias no contexto biológico e envolve um conjunto de princípios com um objetivo de maximizar o grau em que ontologias podem apoiar as necessidades de trabalho dos cientistas [SMITH, B. 2008].

As ontologias OBO *Foundry* aderem a um conjunto de princípios para as melhores práticas no desenvolvimento de ontologias. Por ser uma iniciativa aberta, um autor ao aderir a esta iniciativa compromete-se em manter o avanço científico, além de garantir a melhoria desses princípios. Os dez princípios seguidos para o desenvolvimento de ontologias são [OBO Foundry 2010]:

- 1) A ontologia deve ser aberta e disponível para ser usada por todos sem nenhuma restrição, a não ser que (a) a sua origem deve ser reconhecida e (b) não está sendo alterada e posteriormente redistribuída sob o nome original ou com os mesmos identificadores.
- 2) A ontologia é, ou pode ser expressa em uma sintaxe partilhada comum. Esta pode ser na sintaxe OBO, extensões desta sintaxe, ou em OWL.
- 3) As ontologias possuem um único identificador no espaço dentro do OBO *Foundry*.
- 4) A ontologia fornecida tem procedimentos para a identificação de sucessivas versões distintas.
- 5) A ontologia tem especificação e conteúdo delimitado claramente.
- 6) As ontologias incluem definições textuais para todos os termos.
- 7) A ontologia utiliza as relações que são definidas claramente seguindo o padrão das definições previstas na ontologia OBO *Relation Ontology* (RO).
- 8) A ontologia é bem documentada.
- 9) A ontologia tem uma pluralidade de usuários independentes.
- 10) A ontologia poderá ser desenvolvida colaborativamente com outros membros da OBO Foundry.

Atualmente, OBO *Foundry* compreende cerca de 60 ontologias, dentre elas: Cell Ontology (CL), Chemical Entities of Biological Interest (ChEBI), Foundational Model of Anatomy (FMA), Gene Ontology (GO), Phenotypic Quality Ontology (PaTO), Sequence Ontology (SO), Common Anatomy Reference Ontology (CARO), Protein Ontology (PRO), RNA Ontology (RnaO), Human Disease (DOID).

A seguir serão apresentadas cada uma das ontologias OBO *Foundry* citadas acima, dando enfoque às suas características e objetivos de seu desenvolvimento.

A – *Cell Ontology* (CL)

De acordo com [BARD, J. *et al.* 2005], esta ontologia descreve tipos de células que abrangem os procariontes, fungos, animais e plantas no mundo. Ela inclui cerca de 680 tipos de células. Esses tipos de células são classificados sobre várias categorias e são organizadas como um grafo acíclico direcionado. A ontologia está disponível no formato OBO, adotado pelo OBO *Foundry* e é designada para ser usada no contexto do genoma de organismo modelo e outros bancos de dados biológicos.

A ontologia consiste de termos ou conceitos (nós) que estão ligados a dois tipos de relações (arestas). Um termo ou conceito pode ter vários filhos e vários pais. Os termos pai e filho podem ser conectados entre si pelas relações *is_a* e *develops_from*. A primeira relação indica que o conceito filho é um termo mais restritivo do que o seu conceito pai (assim *Condrócito is_a mesenchyme_cell*). E a outra é usada para codificar as relações de desenvolvimento da linhagem entre os conceitos, por exemplo, um *hepatócitos develops_from um mesenchymal_cell*. A relação *is_a* implica em uma relação de herança, o mesmo não acontece para a relação *develops_from*.

O nível topo da ontologia é formado por dois termos, *cell_in_vivo* e *experimentally_modified_cell*. As células são classificadas por categorias de organismo-independente e por organismo (células animais, células vegetais, células procariontes). Atualmente a versão da *Cell Ontology* possui uma média de cerca de 10 nós em profundidade. Como pode ser observada, a Figura 6 exibe todas as informações associadas ao termo *hepatocyte*. O painel esquerdo mostra todos os termos do nível topo e também a localização do termo *hepatocyte* na classificação de *hepatocyte cell_by_histology*. O painel direito mostra todas as hierarquias as quais podem ser encontrados o termo *hepatocyte*. E o painel superior central ilustra a forma como o termo é encontrado, ao mesmo tempo em que dá a definição do termo na parte inferior.

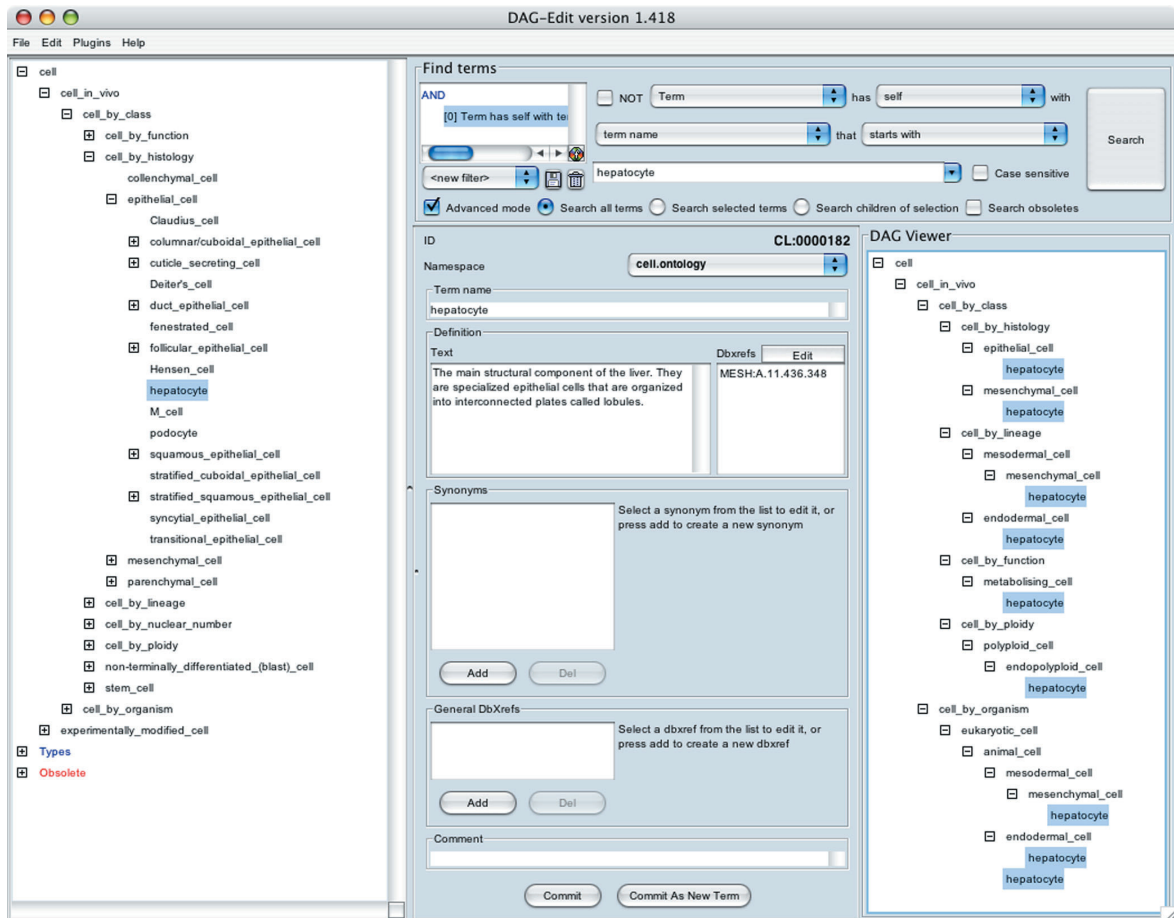


Figura 6. Visão geral da *Cell Ontology* enfatizando o termo *hepatocyte*.

B – Chemical Entities of Biological Interest (ChEBI)

Segundo [DEGTYARENKO, K. *et al.* 2007] a ontologia *Chemical Entities of Biological Interest* é um dicionário de entidades moleculares focada em compostos químicos. Entidades moleculares são produtos naturais ou sintéticos utilizados para intervir nos processos de organismos vivos. ChEBI inclui uma classificação ontológica, em que as relações entre entidades moleculares ou classes de entidades e de seus pais e/ou crianças são especificadas.

Ao contrário da maioria das ontologias existentes, a ontologia ChEBI não é organizada como um grafo acíclico, existindo portanto a inclusão de relações cíclicas. Ela é formada por outras quatro ontologias:

- *Molecular Structure*, no qual entidades moleculares ou partes delas são classificadas de acordo com a estrutura;

- *Biological Role*, que classifica entidades com base no seu papel dentro de um contexto biológico;
- *Application*, que classifica entidades, caso necessário, com base na sua utilização pelos humanos (por exemplo, pesticidas, medicamentos);
- *Subatomic Particle*, que classifica partículas menores do que átomos.

A Figura 7 exhibe as subontologias da ontologia ChEBI, apresenta a definição e exemplos de cada uma.

Sub-ontology	Definition	Example
Molecular structure	A description of the molecular entity or part thereof based on its composition and/or the connectivity between its constituent atoms.	CHEBI:23091 ChEBI ontology \triangleleft CHEBI:24431 molecular structure \triangleleft △CHEBI:23367 molecular entities \triangleleft △CHEBI:33259 homoatomic molecular entities \triangleleft △CHEBI:33262 elemental oxygen \triangleleft △CHEBI:33263 diatomic oxygen \triangleleft △CHEBI:15379 dioxygen \triangleleft △CHEBI:26689 singlet dioxygen \triangleleft △CHEBI:27140 triplet dioxygen
Subatomic particle	A particle smaller than an atom.	CHEBI:23091 ChEBI ontology \triangleleft △CHEBI:36342 subatomic particle \triangleleft △CHEBI:33233 fundamental particle \triangleleft △CHEBI:36338 lepton \triangleleft △CHEBI:10545 electron
Biological role	A role played by the molecular entity or part thereof within a biological context.	CHEBI:23091 ChEBI ontology \triangleleft △CHEBI:24432 biological role \triangleleft △CHEBI:33280 molecular messenger \triangleleft △CHEBI:24621 hormone \triangleleft △CHEBI:28918 (<i>R</i>)-adrenaline
Application	Intended use of the molecular entity or part thereof by humans.	CHEBI:23091 ChEBI ontology \triangleleft △CHEBI:33232 application \triangleleft △CHEBI:25944 pesticide \triangleleft △CHEBI:22153 acaricide \triangleleft △CHEBI:38593 fenazaquin

Figura 7. Subontologias da ontologia ChEBI [DEGTYARENKO, K. *et al* 2007].

Das relações existentes em ChEBI duas (*is a* e *is part of*) são definidas na *Relations Ontology* (RO), que é um padrão para as relações de ontologias biomédicas, as outras são específicas da ontologia. Como dito anteriormente, alguns desses relacionamentos são necessariamente cíclicos. O relacionamento “*A is conjugate acid of B*” significa que a relação “*B is conjugate base of A*” é sempre verdadeiro, enquanto que o relacionamento “*E is tautomer of K*” e “*R is enantiomer of S*” também significa que “*K is tautomer of E*” e “*S is enantiomer of R*” é sempre verdade. Os membros dessas relações cíclicas são colocadas no mesmo nível hierárquico da ontologia. As relações foram introduzidas a partir de uma necessidade de formalizar as diferenças entre os termos que são muitas vezes (incorretamente) usados permutavelmente, especialmente na literatura bioquímica. Por exemplo, “lactate” é frequentemente utilizada como um sinônimo de “lactic acid”.

Em ChEBI, *lactate* (ChEBI: 24996) *is conjugate base of lactic acid* (ChEBI: 28358). A figura 8 apresenta as relações desta ontologia.

Relationships in ChEBI ontology			
Relationship	Symbol	Description	Example
<i>is a</i>	△	Relationship between more specific and more general concepts.	(<i>R</i>)-lactate (CHEBI:16004) <i>is a</i> lactate (CHEBI:24996)
<i>is part of</i>	◇	Relationship between part and whole.	tetracyanonickelate(2-) (CHEBI:30025) <i>is part of</i> potassium tetracyanonickelate(2-) (CHEBI:30071)
<i>is conjugate acid of</i>	⊠	A pair of relationships used to connect acids with their conjugate bases.	lactic acid (CHEBI:28358) <i>is conjugate acid of</i> lactate (CHEBI:24996)
<i>is conjugate base of</i>	⊡		lactate (CHEBI:24996) <i>is conjugate base of</i> lactic acid (CHEBI:28358)
<i>is tautomer of</i>	⊠	A cyclic relationship used to show the interrelationship between two tautomers.	9 <i>H</i> -purine (CHEBI:35589) <i>is tautomer of</i> 1 <i>H</i> -purine (CHEBI:35586)
<i>is enantiomer of</i>	⊠	A cyclic relationship used in instances when two entities are mirror images of and non-superposable upon each other.	D-cysteine (CHEBI:16375) <i>is enantiomer of</i> L-cysteine (CHEBI:17561)
<i>has functional parent</i>	⊠	The relationship between two molecular entities (or classes of entities), one of which possesses one or more characteristic groups from which the other can be derived by functional modification.	codeine (CHEBI:16714) <i>has functional parent</i> morphine (CHEBI:17303)
<i>has parent hydride</i>	⊠	The relationship between an entity and its parent hydride.	perfluorodecane (CHEBI:38851) <i>has parent hydride</i> decane (CHEBI:32894)
<i>is substituent group from</i>	⊠	The relationship between a substituent group (or atom) and its parent molecular entity, from which it is formed by loss of one or more protons or simple groups such as hydroxy groups.	ethyl group (CHEBI:37807) <i>is substituent group from</i> ethane (CHEBI:23975) L-cysteino group (CHEBI:32448) <i>is substituent group from</i> L-cysteine (CHEBI:17561)

Figura 8. Relações da ontologia ChEBI [DEGTYARENKO, K. *et al.* 2007].

Esta ontologia está disponível na *Web*, sendo que o usuário pode acessar seus dados através de uma tabela periódica ou pela ontologia, e também através de uma busca textual simples ou avançada.

Um diferencial para esta ontologia, é que além dela estar disponível no formato OBO, adotado pelo OBO Foundry, ela também se encontra em outros formatos, como *Oracle*, e SQL (*Structured Query Language*).

C – Foundational Model of Anatomy (FMA)

De acordo com [FMA 2010], a ontologia Foundational Model of Anatomy é uma fonte de conhecimento preocupada com a representação de classes ou tipos e relações necessárias para a representação simbólica da estrutura de fenótipos do corpo humano em um formato que seja compreensivo por humanos e também interpretável por sistemas. Especificamente, FMA é uma ontologia de domínio que representa um conjunto de conhecimento explícito acerca da anatomia humana. Esta ontologia pode ser ainda aplicada e estendida para outras espécies.

FMA possui quatro ontologias relacionadas entre si, podendo esta ser representada abstratamente por $FMA = (At, ASA, ATA, Mk)$. Na Figura 9 é apresentada uma estruturação para ontologia At de modo a identificar uma parte abrangida pela ontologia FMA [ROSSE, C. e MEJINO, J. L. V. 2003]. A seguir são apresentadas as definições de cada uma das ontologias que compõem a FMA:

- 1) *Anatomy taxonomy (At)*, classifica entidades anatômicas de acordo com as características que lhes são comuns (gênero) e através dos quais elas podem ser distinguidas a partir de outra. Era designada anteriormente em publicações por *Anatomy ontology (Ao)*;
- 2) *Anatomical Structure Abstraction (ASA)*, especifica as relações *parte-todo* e *espacial* que existem entre as entidades representadas em At;
- 3) *Anatomical Transformation Abstraction (ATA)*, especifica a transformação morfológica de entidades representadas em At durante o desenvolvimento pré-natal e o ciclo de vida pós-natal;
- 4) *Metaknowledge (Mk)*, especifica os princípios, regras e definições de acordo com classes e relações que estão representados nos outros três componentes da ontologia FMA.

Segundo [ROSSE, C. e MEJINO, J. L. V. 2007], FMA é a maior ontologia no domínio biomédico, consiste em mais de 135000 termos e 198 relações. Pode-se dar destaque para a classe *cell*, pois é a maior de toda a ontologia, e representa as várias células do corpo humano. A Figura 10 apresenta uma taxonomia para esta classe.

A ontologia foi originalmente desenvolvida em UMLS (*Unified Medical Language System*), e atualmente passa por um processo de migração para o formato OBO.

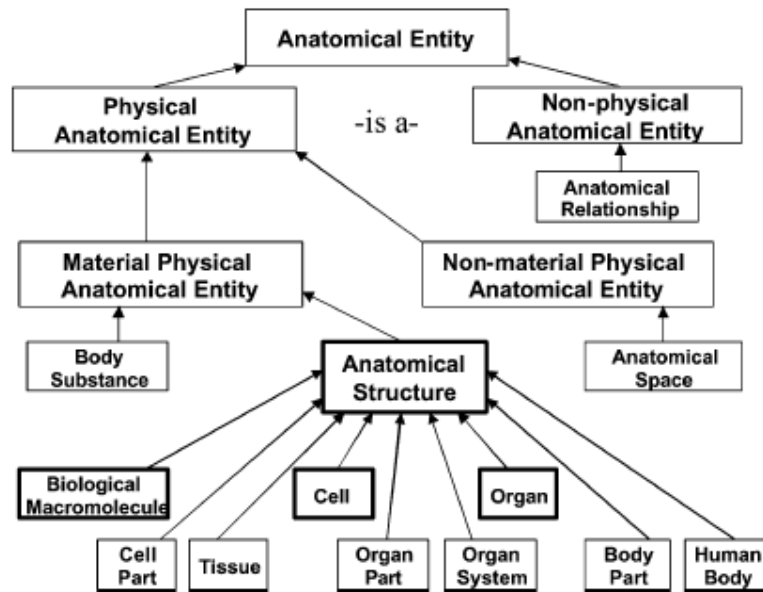


Figura 9. Representação das principais classes da ontologia *Anatomy taxonomy* (At) [ROSSE, C. e MEJINO, J. L. V. 2003].

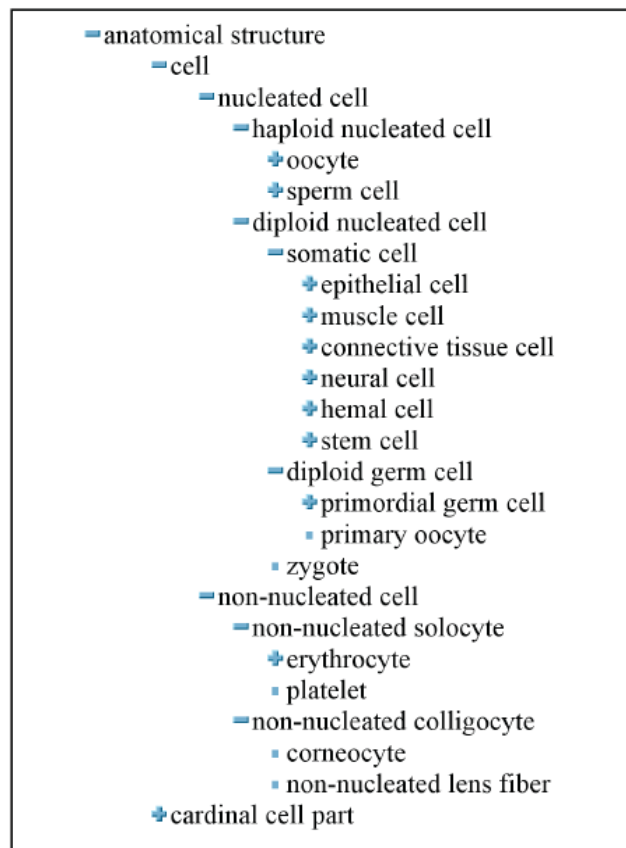


Figura 10. Taxonomia da classe *cell* da ontologia FMA [ROSSE, C. e MEJINO, J. L. V. 2007].

D – Gene Ontology (GO)

A ontologia *Gene Ontology*, na verdade faz parte de um grande projeto, o *Gene Ontology Project*, no qual são desenvolvidos e utilizados um conjunto de vocabulários estruturados e controlados para uso comum em anotações genéticas, produtos genéticos e seqüências. O projeto fornece um sistema de anotação ontológica que permite biólogos inferirem conhecimento a partir de grandes quantidades de dados. As anotações GO são úteis para a extração de significado funcional e biológico de grandes conjuntos de dados, como *microarrays*. E também facilita a organização de dados genômicos e a comparação de dados biológicos. [HARRIS, M. A. *et al.* 2006].

Gene ontology é composta por três ontologias [ASHBURNER, M. *et al.* 2000]:

- *Cellular component*: refere-se a um lugar na célula onde um produto genético é ativo. Estes termos refletem o entendimento da célula eucarionte. Assim como em outras ontologias, nem todos os termos são aplicáveis para todos os organismos, o conjunto de termos se estende para ser inclusivo. Esta ontologia inclui termos como “*ribosome*” ou “*proteasome*”, especificando onde múltiplos produtos genéticos poderiam ser encontrados. Também inclui os termos “*nuclear membrane*” ou “*Golgi apparatus*”.
- *Biological process*: refere-se a um objetivo biológico para o qual o gene ou produto genético contribui. Um processo é realizado através de um ou mais conjuntos ordenados de funções moleculares. Processos muitas vezes envolvem uma transformação química ou física, no sentido que alguma coisa entra em um processo e alguma coisa diferente sai do mesmo.
- *Molecular function*: é definido como uma atividade bioquímica (incluindo as específicas obrigatórias para ligantes ou estruturas) de produtos genéticos. Esta definição também se aplica à capacidade que um produto genético (ou produto genético complexo) transporta como potencial. Ele descreve apenas o que é feito sem especificar onde ou quando o evento realmente ocorre. Exemplos de termos funcionais mais amplos são “*enzyme*”, “*transporter*” ou “*ligand*”. Exemplos de termos funcionais mais restritos são “*adenylate cyclase*” ou “*Toll receptor ligand*”.

De acordo com [GENE ONTOLOGY 2010], as ontologias são organizadas como um grafo direcionado acíclico, que são similares a hierarquias, mas diferem no ponto em que o termo mais especializado pode ser relacionado com mais de um termo menos especializado. A Figura 11 apresenta a estrutura e estilo usado pela GO para representar as ontologias GO e para associar genes com nós dentro da ontologia. Ela exibe a representação para as três ontologias associadas à GO [ASHBURNER, M. *et al.* 2000].

A ontologia pode ser ligada por cinco tipos de relação: *is_a*, *part_of*, *regulates*, *positively_regulates* and *negatively_regulates*. A relação *is_a* é uma relação simples de classe e subclasse. A relação *part_of* é um pouco mais complexa; “C *part_of* D” significa que sempre C está presente, é sempre uma parte de D, mas C nem sempre têm de estar presente. E as relações *regulates*, *positively_regulates* and *negatively_regulates* descrevem interações entre processos biológicos e outros processos biológicos, funções moleculares ou qualidades biológicas. Quando um processo biológico E regulamenta uma função ou um processo F, ele modula a ocorrência de F. Se F é uma qualidade biológica, então E modula o valor de F.

E – Phenotypic Quality Ontology (PATO)

De acordo com [PATO 2010], a ontologia *Phenotypic Quality Ontology* é uma ontologia das “qualidades fenóticas” destinada para uso em um número de aplicações, principalmente anotações de fenótipos e projetada para ser utilizada em conjunto com ontologias de “entidades de qualidade de suporte”, como por exemplo, *Cell Ontology* (CL), e *Gene Ontology* (GO) para processos biológicos ou componentes celulares e ontologias de entidades anatômicas.

PATO é independente de qualquer formato de troca ou esquema de banco de dados. Uma forma de expressar anotação de fenótipos usando PATO é através da sintaxe pheno, por exemplo, o fenótipo para olhos vermelhos seria escrito como “*E=FBbt:eye Q=PATO:red*” ou *xml-pheno*. Também é possível utilizar a linguagem OWL para a representação fenótipos.

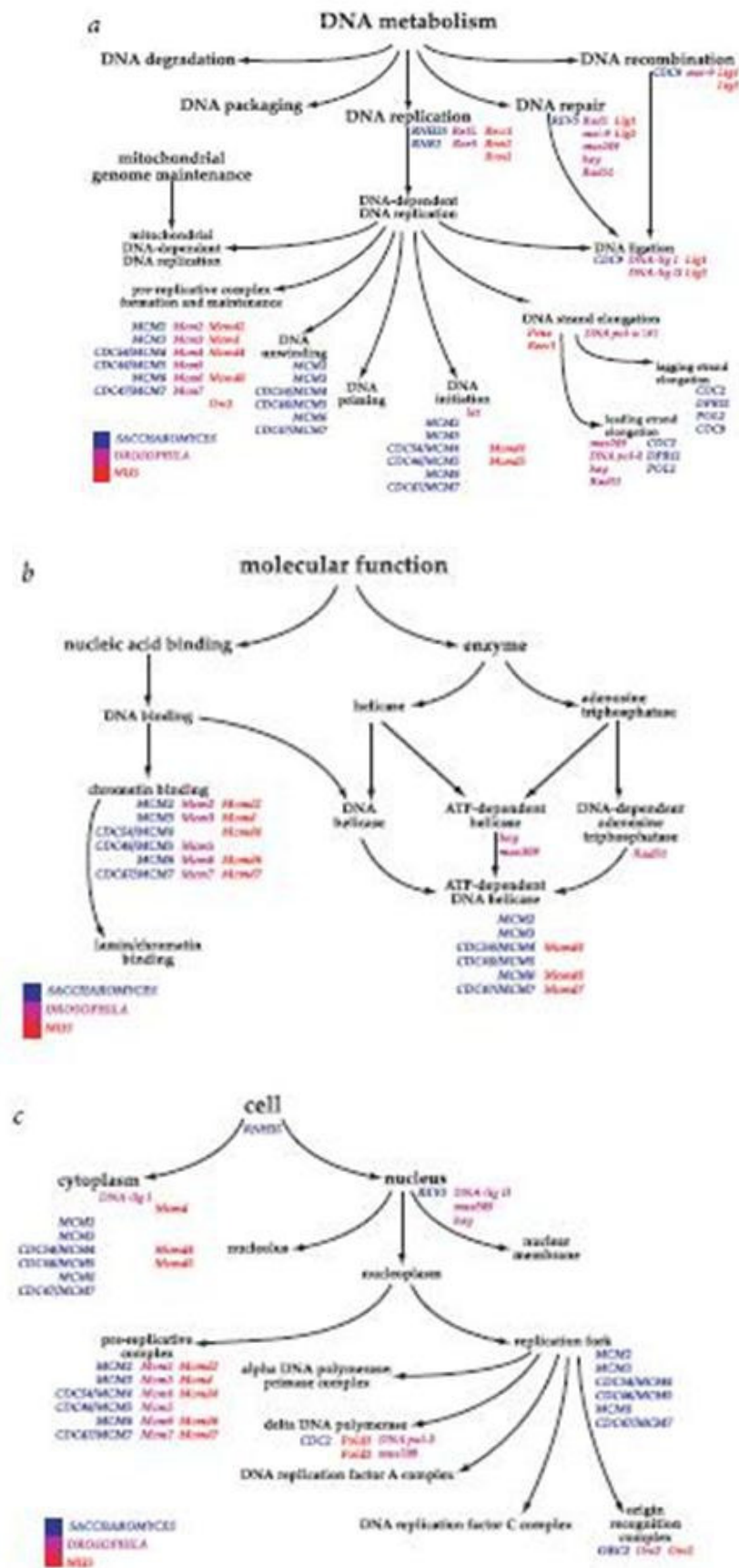


Figura 11 Exemplos de representação GO. (a) ontologia para processo biológico; (b) ontologia para função molecular; (c) ontologia para componente celular [ASHBURNER, M. *et al.* 2000].

F – Sequence Ontology (SO)

De acordo com [EILBECK, K. *et al.* 2005], a ontologia *Sequence Ontology* é um vocabulário controlado estruturado para as partes de uma anotação genômica, a ontologia também fornece um conjunto comum de termos e definições que irá facilitar o intercâmbio, análise e gerenciamento de dados genômicos. Isso se deve ao fato de que SO trata de relações parte-todo rigorosamente, dados descritos com ela pode se tornar substrato para o raciocínio automatizado, e de instâncias de seqüência características descritas pelo SO pode ser submetido a um grupo de operações lógicas denominado de operadores *extensional mereology* (EM).

O objetivo da SO é permitir a padronização de um conjunto de termos e relações com os quais é possível descrever anotações genômicas e permitir a estrutura necessária para raciocínio automatizado sobre seus conteúdos, facilitando as mudanças de dados e análise comparativa de anotações. Esta ontologia, além de fazer parte das ontologias OBO Foundry, é uma ontologia desenvolvida pela a equipe da GO. O escopo dessa ontologia é a descrição de recursos e propriedades da seqüência biológica. E está disponível no formato OBO, além de ser organizada como um grafo acíclico direcionado.

Em sua estrutura ela utiliza três tipos básicos de relações entre seus termos, as quais são definidas na ontologia de relações do OBO: **kind_of**, **derives_from**, and **part_of**. A figura 12 apresenta uma parte da ontologia SO e as relações citadas entre os termos.

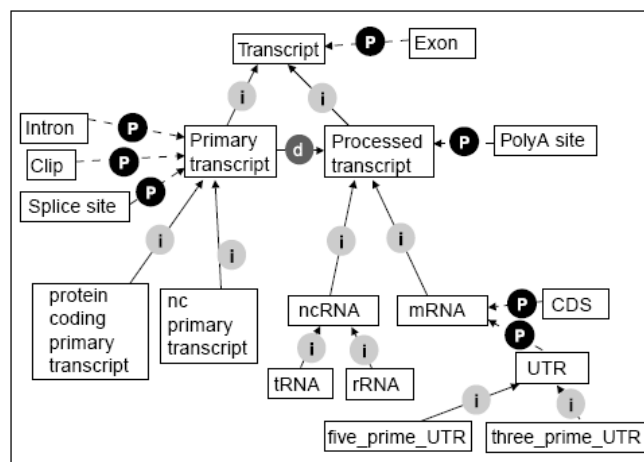


Figura 12. Parte da ontologia SO mostrando como termos e relacionamentos são utilizados em conjunto para descrever os conhecimentos sobre seqüência. As relações *kind_of* são representadas por setas marcadas com 'i', *part_of* usam setas com "P" e *derives_from* com 'd'. Por exemplo, um *Exon* é *part_of* um *Transcript*, um *tRNA* é um *kind_of* *ncRNA* que é um *kind_of* *Processed_Transcript*. [EILBECK, K. *et al.* 2005]

SO também permite alguns modos de análises de dados genômicos que são completamente novos para o campo. Uma dessas classes de análises envolve o uso de operadores EM para fazerem perguntas sobre partes do gene, proporcionando uma base para perguntas e respostas referentes ao modo como as partes são distribuídas dentro e entre as partes. EM é uma teoria formal das partes: ela define as propriedades das relações **part_of** e, em seguida, fornecem um conjunto de operações (Figura 13) que podem ser aplicadas a essas partes.

The EM operators	
EM operation	Definition
Overlap ($x \circ y$)	x and y overlap if they have a part in common.
Disjoint ($x \uparrow y$)	x and y are disjoint if they share no parts in common.
Binary product ($x \cdot y$)	The parts that x and y share in common.
Difference ($x - y$)	The largest portion of x which has no part in common with y.
Binary sum ($x + y$)	The set consisting of individuals x and y.

Figura 13. Operadores EM (x e y referem-se a duas partes) [EILBECK, K. et al 2005].

G – Common Anatomy Reference Ontology (CARO)

De acordo com [HAENDEL, M. A. et al. 2006], a ontologia *Common Anatomy Reference Ontology* foi desenvolvida para facilitar a interoperabilidade entre as ontologias de anatomia existentes para diferentes espécies, e fornecer um modelo para a construção de novas ontologias sobre anatomia.

A ontologia CARO é uma ontologia de anatomia comum que tem seu desenvolvimento baseado na ontologia FMA e adere aos princípios estabelecidos pela OBO *Foundry*. Além disso, ela fornece as relações e as definições para tipos anatômicos de alto nível para anatomias canônicas. A anatomia canônica fornece uma relação da composição dos membros de um “protótipo” de uma dada espécie, facilitando a tarefa de construção de ontologias de anatomia, uma vez que as informações capturadas, por exemplo, referentes às relações para parte e localização, podem diferir radicalmente em tipo não-canônicas. A Figura 14 exibe uma taxonomia da ontologia CARO.

CARO complementa o processo de desenvolvimento da subontologia *Biological Process* da ontologia GO. Suporta a coordenação de ontologias de espécies cruzadas em todos os níveis de granularidade anatômica através de referenciamento cruzado dentro das ontologias CL e GO-*Cellular Component*. É uma

completa ontologia CARO de espécies cruzadas pode ser utilizada por outras ontologias para a geração de produtos cruzados.

A ontologia *Relations Ontology* abrange os relacionamentos mais importantes para ontologias de anatomia, mas carece de muitas definições explícitas de relações espaciais que seria desejável que fossem incluídas. Portanto, relações como, por exemplo, *develops_from*, *has part* e *part of*, foram incluídas devido às necessidades da ontologia CARO.

O desenvolvimento da ontologia CARO tem o objetivo de servir como uma padronização, fornecendo um conjunto de relações para uso na construção de ontologias de anatomia, entendendo dessa forma o conjunto de relações já existentes em RO.

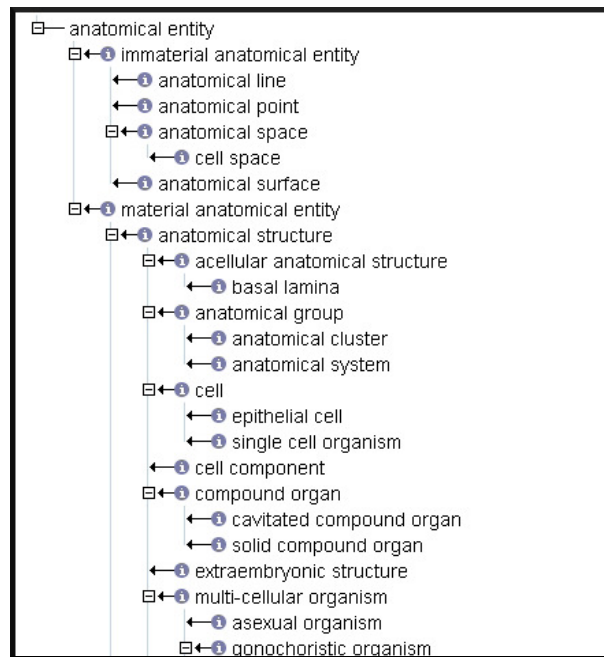


Figura 14. Taxonomia da ontologia CARO [HAENDEL, M. A. *et al.* 2006]

H – Protein Ontology (PRO)

Segundo [NATALE, D. A. *et al.* 2007], a ontologia *Protein Ontology* foi projetada para ser uma ontologia de proteínas formal, bem fundamentada e extensível do OBO *Foundry*, com um conjunto básico bem definido de relações para apoiar a integração semântica e máquina de inferência. O desenvolvimento da PRO começou incluindo classes relevantes para proteínas de humanos e de ratos com um foco em doenças relacionadas com proteínas, mas este está em processo de expansão.

Segundo [PRO 2010], PRO é um dos meios de formalizar dados e conhecimentos de proteínas. A ontologia inclui conceitos relevantes para o domínio, definições de conceitos, e relações definidas entre os conceitos. PRO integra dados de proteína e fornece um vocabulário estruturado e unificado para representar conceitos de síntese protéica. Também, prevê integração de proteínas heterogêneas e fontes de dados biológicos. Além disso, converte grandes quantidades de dados coletados pelos geneticistas e biólogos moleculares em informações que os cientistas, médicos e outros profissionais de saúde e pesquisadores podem utilizar para compreender facilmente o mapeamento das relações entre moléculas de proteínas, a interação entre duas moléculas de proteínas e outras macromoléculas em nível celular. E esta ontologia também contribui para a codificação de dados protéicos para análise feita por pesquisadores. A Figura 15 apresenta a hierarquia desta ontologia.

Esta ontologia descreve tipos de proteínas e os relacionamentos entre elas. E concentra-se em [NATALE, D. A. *et al.* 2007]:

- uma ontologia de proteínas baseadas em parentesco evolutivo (ontologia para evolução da proteína)
- uma ontologia de múltiplas formas de proteína produzidas a partir de um dado gene *locus* (ontologia para formas de proteína)

I – RNA Ontology (RnaO)

A ontologia *RNA Ontology*, será identificada neste trabalho através da sigla RnaO. Segundo [LEONTIS, N. B. *et al.* 2006], as propostas para a criação da RnaO são: (1) integrar seqüência e bases de dados estruturais; (2) permitir diferentes ferramentas computacionais para interoperar ontologias; (3) criar poderosas ferramentas de software que trazem métodos computacionais avançados para os cientistas; e (4) facilitar pesquisas precisas para todas as informações pertinentes para o RNA.

O domínio da ontologia RnaO atende as seguintes áreas :

- Seqüências de RNA (1D): codifica e decodifica, e sua identificação em genomas (pode ser incorporado com SO).
- Estruturas secundárias de RNA.
- Estruturas de RNA (3D).

- Alinhamento de seqüências homólogas de RNA.
- Relações entre alinhamento e estruturas 3D.
- Interações RNA–RNA, RNA–protein, and RNA–ligand.
- Biologia molecular de RNA.
- Dados experimentais bioquímicos e biofísicos relativos à estrutura do RNA.
- RNA como regulador das redes e percursos biológicos.

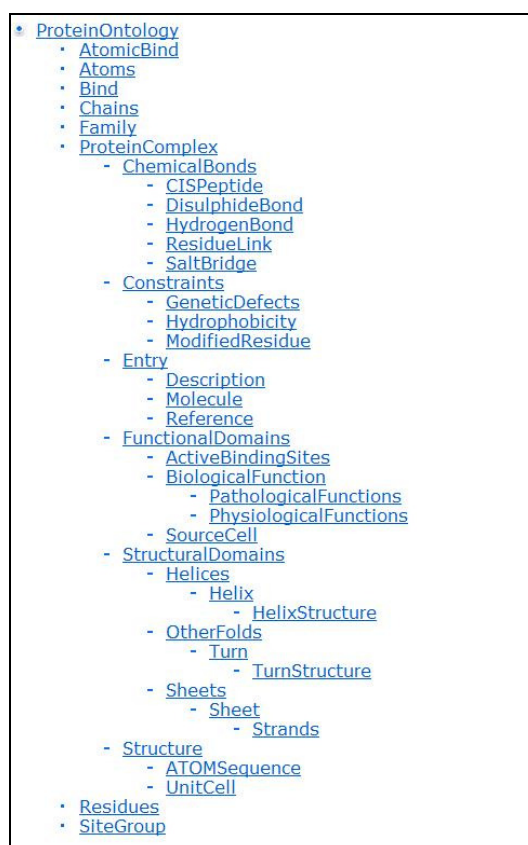


Figura 15. Hierarquia da ontologia PRO [NATALE, D. A. *et al.* 2007].

A figura 16 sumariza a integração de estruturas e dados seqüenciais para atingir o alinhamento automático da seqüência. A integração de dados heterogêneos incluindo estruturas de RNA experimental 3D (superior esquerdo) e seqüências de nucleotídeos (inferior esquerdo). O RnaO irá representar as informações sobre RNA que são compreensíveis por humanos (superior direita) e máquinas (inferior direita).

De acordo com [ROC 2010], deverá ser realizado, a longo prazo, a derivação de regras para a formação de estruturas e interações com RNA, e disponibilizá-las através de ferramentas computacionais para que, não só especialistas, mas todos possam usá-la. Para que isso seja consolidado, será

necessário um grande esforço de comunicação entre os vários elementos da comunidade que lidam com determinação estrutural, previsão e conseqüências biológicas.

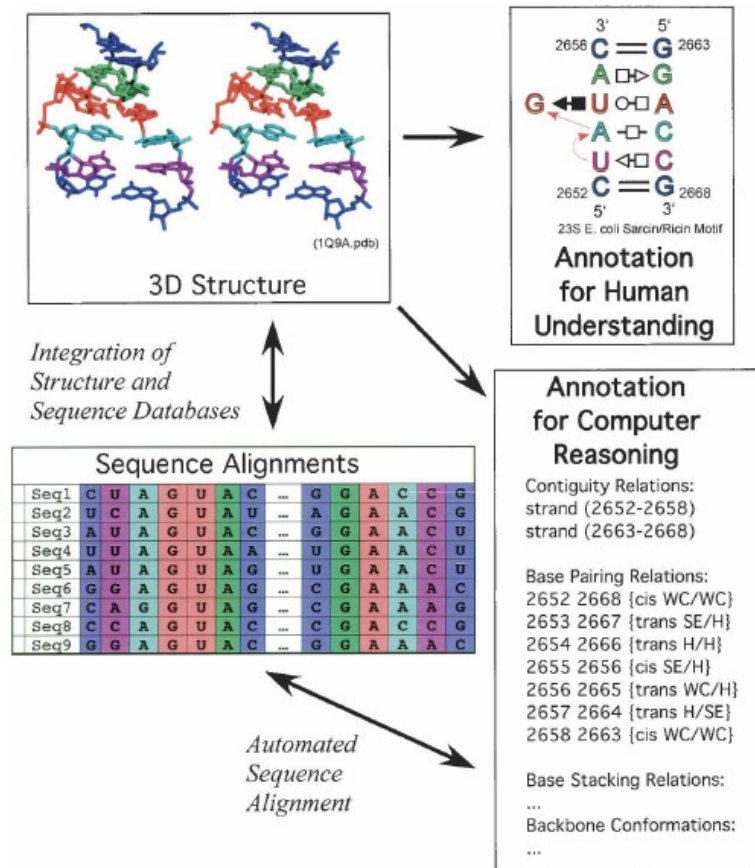


Figura 16. Integração de estruturas e dados seqüenciais na RnaO [LEONTIS, N. B. *et al.* 2006].

J – Disease Ontology (DOID)

A ontologia *Disease Ontology*, será identificada neste trabalho através da sigla DOID. Segundo [DISEASE ONTOLOGY 2010], a ontologia tem como objetivo fornecer uma fonte aberta para a integração de dados biomédicos. Os termos presentes são definidos seguindo referências padrão da área como a SNOMED.

Disease Ontology foi projetada para ligar os diferentes conjuntos de dados através de conceitos da doença. Ela fornece uma estrutura computável e facilita a ligação dos termos entre doenças infecciosas humanas e dados genéticos, dados clínicos e sintomas.

2.7.2 CelO (*Cell Component Ontology*)

A ontologia CelO [MATOS, E. E. S. 2008] [MATOS, E. E. S. *et al.*, 2009] se aplica à representação de modelos biológicos, particularmente modelos em eletrofisiologia celular. A ontologia foi desenvolvida na linguagem OWL-DL e utilizando o editor de ontologias Protégé. A escolha da linguagem OWL-DL se deve ao fato de que ela permite expressividade suficiente para representar os componentes e propriedades dos modelos e também permite o uso de mecanismos de inferência e a definição de regras semânticas. Além disso, OWL-DL é um formato padronizado e recomendado pelo W3C, o que possibilita que modelos biológicos criados sejam portáteis e integráveis, independentes da aplicação, formato ou linguagem de programação utilizada.

Pode-se ainda enumerar vários objetivos e aplicações para o desenvolvimento da ontologia CelO:

- 1) Representar um modelo biológico através de uma linguagem lógica.
- 2) Representar modelos existentes descritos em CellML utilizando anotações semânticas.
- 3) Permitir a criação de modelos onde variáveis possam ser definidas a partir de seu significado no modelo.
- 4) Permitir a criação de um repositório de modelos podendo este ser pesquisado semanticamente.
- 5) Prover o reuso de componentes existentes.

A ontologia apresenta duas características principais: integração com a linguagem CellML e extensibilidade.

A integração com a linguagem CellML permite que sejam construídas ferramentas que integrem simuladores já existentes descritos na linguagem, de forma que o modelador trabalhe com um nível alto de abstração e possa executar o modelo de forma transparente. A Figura 17 mostra a integração CellML em níveis de abstração.

	Modelo
	Biológico
Nível Conceitual	Ontologia CeLO
Nível Lógico	CellML
Nível Físico	Simuladores

Figura 17. Níveis de abstração do modelo [MATOS, E. E. S. 2008].

Já a extensibilidade, permite a representação de novos elementos ao modelo. De acordo com o autor, caso se deseje representar algo que não está definido na ontologia atual, ou testar novos componentes, pode-se estendê-la, criando novas classes, indivíduos, propriedades ou regras. Isso permite a criação de uma nova ontologia, sem que haja alteração da estrutura da ontologia original. Mas para que a adição de novos elementos ao modelo seja feita de forma eficiente, deve-se garantir que essa nova ontologia possa ser integrada umas com as outras, além de não perder a consistência com o modelo CellML representado.

A ontologia é composta de três classes gerais, definidas no nível mais alto da ontologia: *SIEntity*, *DomainEntity* e *ModelEntity*. Essas classes definem três tipos essenciais de conhecimento sobre um modelo.

A classe *SIEntity* e suas subclasses, definem um dicionário, baseado no Sistema Internacional de Unidades (SI), de unidades e grandezas que são utilizadas em conjunto com os conceitos do domínio e as variáveis dos modelos. Novas unidades podem ainda ser definidas pelos usuários, para permitir a integração com CellML. A Figura 18 apresenta a estrutura da classe.

A classe *DomainEntity* e suas subclasses, definem os termos que serão usados como um vocabulário compartilhado pelos pesquisadores na criação de modelos. Esses termos possuem uma semântica associada, permitindo a descrição do modelo em nível mais alto. Esta classe define ainda termos de três áreas de pesquisa, Biologia, Química e Bioquímica, ressaltando que como o escopo deste trabalho está restrito a eletrofisiologia celular, apenas conceitos relativos à célula foram anotados. A figura 19 apresenta a estrutura da classe.

E a classe *ModelEntity* e suas subclasses definem os conceitos que serão usados no modelo biológico representado. Essa classe tem como objetivo ter uma

descrição de alto nível, fazendo referência ao modelo CellML para as questões de simulação. A figura 20 apresenta a estrutura da classe.

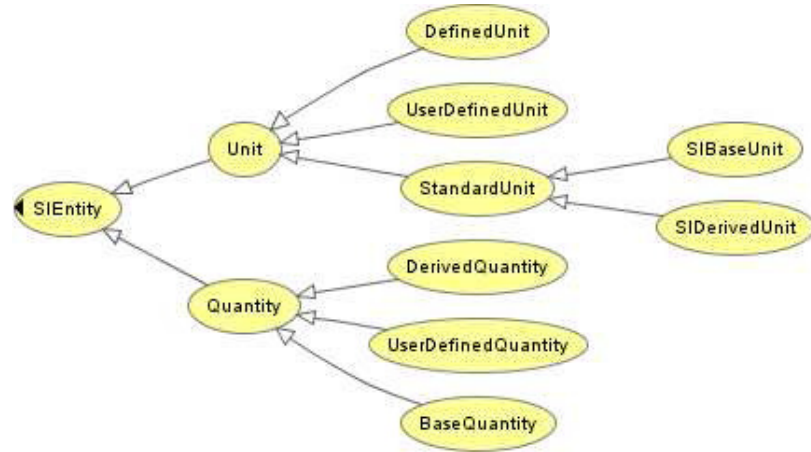


Figura 18. Estrutura da classe *SIEntity* da ontologia CelO.

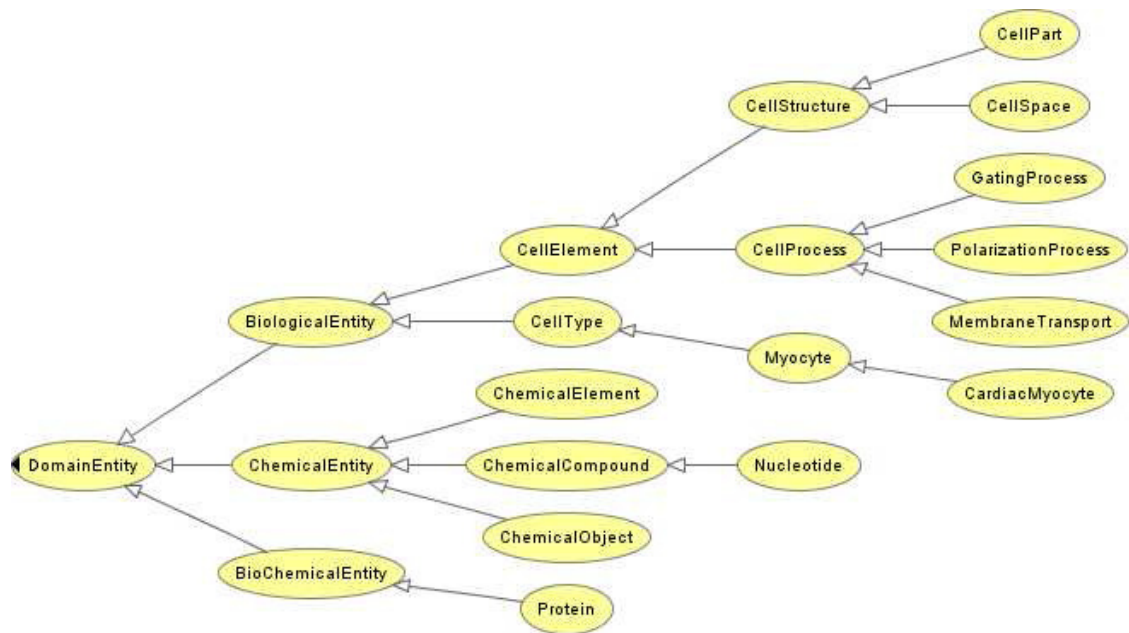


Figura 19. Estrutura da classe *DomainEntity* da ontologia CelO.

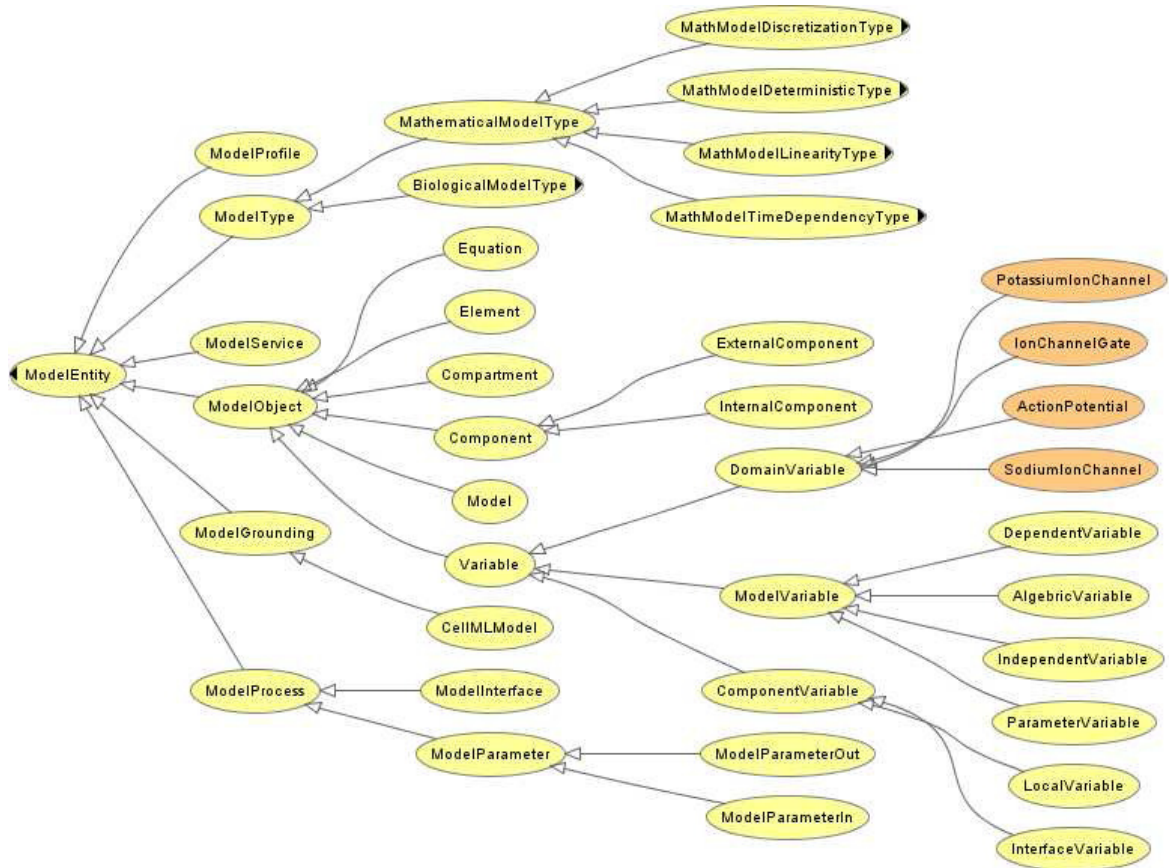


Figura 20. Estrutura da classe *ModelEntity* da ontologia CelO.

2.8 COMENTÁRIOS FINAIS

Este capítulo abordou uma discussão preliminar sobre ontologias na área biológica. Conceitos básicos de uma ontologia foram apresentados. Assim como as classificações possíveis, linguagem e ferramentas para o desenvolvimento de ontologias. Destacamos as características de qualidade desejáveis na elaboração de uma ontologia. E ao final apresentamos algumas das ontologias existentes na área biológica.

De acordo com o que foi apresentado, é notório a necessidade de se aplicar uma metodologia para o desenvolvimento de ontologias a fim de permitir uma construção correta e completa da ontologia. O Capítulo 3 apresenta então metodologias e métodos para o desenvolvimento de ontologias.

3 METODOLOGIAS E MÉTODOS PARA A CONTRUÇÃO DE ONTOLOGIAS

Assim como na Engenharia de Software são definidos modelos de processos para a construção de softwares, nas ontologias também se faz necessário a aplicação de metodologias para o seu desenvolvimento. Nesse contexto, uma nova área de pesquisa denominada Engenharia de Ontologias ou Engenharia Ontológica (*Ontology Engineering*) surge para atender essa necessidade.

A Engenharia de Ontologias é definida como:

“o conjunto de atividades que concentra o processo de desenvolvimento de ontologias, o ciclo de vida da ontologia, e as metodologias, ferramentas e linguagens para a construção de ontologias.”

[TEMPICH, C. 2006 (apud GÓMEZ-PÉREZ *et al.*, 2003)]

Por ser uma área de pesquisa recente, ainda não existem metodologias para o desenvolvimento de ontologias que sejam largamente utilizadas e aceitas pela comunidade científica. O que existe são propostas de metodologias, algumas delas mais testadas que outras [BRANDÃO, A. A. F., LUCENA, C. J. P. 2002]. Ainda segundo [BREITMAN, K. K. *et al* 2007], vários grupos de pesquisa estão procurando adequar métodos de desenvolvimento de ontologias. No entanto, as possibilidades são tantas que pode ser impossível chegar a um único método que seja adequado para todas as situações. Provavelmente, a melhor solução será uma escolha entre várias possibilidades ou uma composição de diferentes métodos de desenvolvimento de ontologia.

O processo de construção de ontologias é um procedimento, muitas vezes mais artesanal do que de engenharia. Cada equipe de desenvolvimento segue usualmente os seus próprios conjuntos de princípios, critérios de projeto e fases no processo de desenvolvimento de ontologias. A inexistência de um acordo comum sobre orientações e métodos gera alguns problemas como: a dificuldade no desenvolvimento de ontologias dentro e entre as equipes, a extensão de uma ontologia por outros e ainda a sua reutilização em outras ontologias e aplicações. Entretanto, se as ontologias forem construídas em uma escala pequena, algumas atividades ou etapas podem ser eliminadas, caso contrário, é aconselhável seguir

uma abordagem metodológica a fim de se garantir a corretude e completude da ontologia em desenvolvimento [GOMEZ-PEREZ, A., BENJAMINS, V. R. 1999].

De acordo com [CHENINE, M. *et al* 2006], metodologias de desenvolvimento de ontologias sugerem um processo no qual a identificação e especificação da ontologia pode ser completado.

Segundo [JARRAR, M. 2005], com base no cenário que a metodologia proporciona, é possível fazer uma distinção entre as metodologias para o desenvolvimento de ontologias. Elas podem ser classificadas como metodologia de etapas, de modelagem e de engenharia.

- **Metodologia de etapas.** Divide o processo de desenvolvimento em um conjunto de etapas e guias a ser seguido em cada fase. As metodologias Methontology [FERNÁNDEZ, M. *et al* 1997] e On-To-Knowledge [SURE, Y., STAAB, S., STUDER, R. 2002] são exemplos desta classificação.
- **Metodologia de modelagem.** Preocupa-se com a análise formal de um dado domínio: que tipos de decisões de modelagem precisam ser feitas e como essas decisões podem ser avaliados. Essa análise pode ser realizada normalmente por meio de um conjunto bem definido de primitivas, como por exemplo, as noções de conceito/classe, relações n-árias, funções, propriedades/atributos, tipos de regra etc. A metodologia OntoClean [OntoClean 2010] é um exemplo desta classificação.
- **Metodologia de engenharia:** preocupa-se com o projeto, representação, arquitetura e gerenciamento da ontologia. Procura responder questões como permitir o reuso, usabilidade, manutenibilidade, desenvolvimento distribuído, escalabilidade de ontologias entre outras, mas não envolve diretamente com decisões de modelagem ou fases. Análogo ao ciclo de vida do desenvolvimento de um software, o paradigma de orientação a objetos é a base para essa metodologia, tornando dessa forma a ontologia mais reusável, e de fácil manutenção e construção.

Nas próximas seções são apresentadas dez metodologias, cuja seleção foram baseadas nas referências da literatura e nos critérios de relevância das mesmas. *Uschold and King* [USCHOLD, M., KING, M. 1995] que é baseada na experiência de desenvolvimento da ontologia Enterprise. O método *Toronto Virtual Enterprise* (TOVE) que é baseado no desenvolvimento de ontologias no domínio de processos de negócios e modelagem de atividades. O framework *METHONTOLOGY*

[FERNÁNDEZ, M. *et al* 1997] que dá suporte à construção de ontologias no nível do conhecimento baseado no padrão IEEE para desenvolvimento de software. Método *Cyc* [CYC 2010] desenvolvido a partir da base de conhecimento *Cyc* que considera o conhecimento consensual sobre o mundo, incluindo regras e heurísticas sobre objetos e eventos do cotidiano. A metodologia ON-TO-KNOWLEDGE [SURE, Y. *et al.* 2002] direcionada à construção a sistemas de gestão do conhecimento. Um framework para ontologias multilíngües [SOUZA, J. F. 2007 (apud (LAUSER *et al.*, 2002)], que com base na METHONTOLOGY acrescenta modificações específicas para que o processo de criação e evolução de ontologias de domínio seja semi-automático. O processo unificado KUP para o desenvolvimento de ontologias e bases de conhecimento voltadas para a Web Semântica, adapta e integra as melhores práticas de desenvolvimento apresentadas pelo RUP (*Rational Unified Process*). DILIGENT, uma metodologia para construir, desenvolver e conciliar ontologias em configurações DKM (*Distributed Knowledge Management*). O método KACTUS baseado no projeto *Espirit Kactus*, o qual tem como objetivo investigar o reuso do conhecimento em sistemas e o papel da ontologia no processo. E finalmente o método 101 [NOY, N., McGUINNESS, D. 2001] desenvolvido para guiar os usuários na construção da primeira ontologia.

3.1 METODOLOGIA DE USCHOLD E KING

Esta metodologia teve como base a experiência de desenvolvimento da ontologia *Enterprise Ontology*, sendo concebida para dar suporte à modelagem de processos empresariais.

O processo de construção desta metodologia é composto por quatro estágios diferentes, (1) identificação de propósito que é importante para definir a meta e o objetivo da ontologia, (2) construção da ontologia através da captura de conhecimento, código deste conhecimento e integração de ontologias existentes, (3) avaliação da ontologia e (4) documentação da ontologia [CHENINE, M. *et al* 2006]. A descrição de cada estágio é como segue [BREITMAN, K. K. *et al* 2007]:

1. Identificação de propósito e escopo da ontologia: Define o porquê de construir a ontologia e para que ela será utilizada;

2. Construção da ontologia

2.1 Captura: Define os conceitos e os relacionamentos textualmente;

2.2 Codificação: Formaliza os conceitos e relacionamentos acima em uma linguagem formal;

2.3 Integração: Questiona sobre reutilização de ontologias existentes. Esta atividade deve ser feita em paralelo com as duas atividades anteriores;

3. Avaliação da ontologia: Utiliza critérios técnicos para verificação da especificação usando questões de competência e comparação com o mundo real;

4. Documentação da ontologia: Descreve o processo de construção da ontologia, podendo ter formato diferente para tipos distintos de ontologias.

A Figura 21 descreve os estágios da metodologia de Uschold e King [BREITMAN, K. K. *et al* 2007].

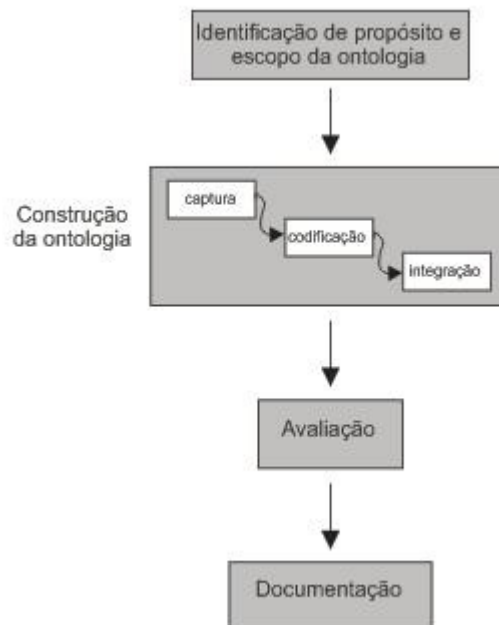


Figura 21: Estágios da metodologia de Uschold e King [BREITMAN, K. K. *et al* 2007].

3.2 METODOLOGIA TORONTO VIRTUAL ENTERPRISE (TOVE)

Esta metodologia foi baseado na experiência dos autores no desenvolvimento de ontologias no domínio de processos de negócios e modelagem de atividades [BREITMAN, K. K. *et al* 2007].

TOVE é baseado na construção de um modelo lógico que será especificado em uma ontologia. A Figura 22 ilustra cada etapa deste método. Isto é feito por (1) que cria cenários motivacionais sobre ontologia, então, (2) questões de competência informais são formuladas com base nos cenários estabelecidos. (3) A especificação dos termos é feita com base nos termos extraídos de (2). O processo então continua, mas desta vez com (4) questão de competência formal a partir da qual (5) a especificação dos axiomas e a definição dos termos são formalizadas [CHENINE, M. *et al* 2006]. A descrição de cada etapa do método e apresentada abaixo [BREITMAN, K. K. *et al* 2007]:

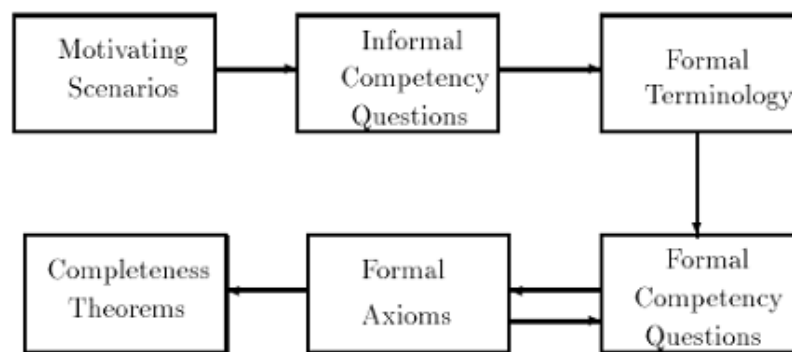


Figura 22: Etapas do método TOVE [CHENINE, M. *et al* 2006].

1. Descrição de cenários motivacionais: Os cenários motivacionais são descrições de problemas ou exemplos que não são cobertos adequadamente por ontologias existentes. A partir destes cenários é permitido o desenvolvimento de um conjunto de soluções possíveis que carregam a semântica informal dos conceitos e relações que serão incluídos na ontologia posteriormente;
2. Formulação informal das questões de competência: Baseados nos cenários, são elaboradas questões de competência, para a qual a ontologia a ser desenvolvida deverá fornecer respostas válidas;
3. Especificação dos termos da ontologia numa linguagem formal: Define um conjunto de termos a partir das questões de competência. Estes conceitos servirão de base para a especificação numa linguagem formal usando uma linguagem de representação de conhecimento, como por exemplo, lógica de primeira-ordem ou Knowledge Interchange Format (KIF);
4. Descrição formal das questões de competência: Descreve as questões de competência usando uma linguagem formal;

5. Especificação dos axiomas: Descreve as regras em linguagem formal, a fim de definir a semântica dos termos e relacionamentos da ontologia;
6. Verificação da completude da ontologia: Estabelece condições que caracterizem a ontologia como completa através das questões de competência formal.

De acordo com [PINTO, H. S., MARTINS, J. P. 2004], nesta metodologia as atividades de conceituação, formalização e implementação não estão claramente divididas em etapas. Já as atividades de manutenção, aquisição de conhecimento e documentação não são referidas explicitamente. O Quadro 5 apresenta a classificação das atividades do método TOVE de acordo com a terminologia usual em Engenharia de Ontologias [PINTO, H. S., MARTINS, J. P. 2004].

Quadro 5. Classificação das atividades TOVE [PINTO, H. S., MARTINS, J. P. 2004].

Atividades TOVE	Correspondente
Descrição de cenários motivacionais e formulação informal das questões de competência	Especificação
Especificação dos termos, descrição formal das questões de competência e especificação dos axiomas	Conceituação, Formalização e Implementação
Verificação da completude da ontologia	Avaliação

3.3 METHONTOLOGY

Esta metodologia foi desenvolvida no laboratório de Inteligência Artificial da Universidade de Madri, e é na verdade um framework que, dentre outras funcionalidades, dá suporte à construção de ontologias no nível do conhecimento baseado no padrão IEEE para desenvolvimento de software. É um método essencialmente descritivo. Associado a este framework existe um ambiente de desenvolvimento de ontologias: ODE (Ontology Development Environment). Diferentemente das demais, esta metodologia descreve a identificação do processo de desenvolvimento da ontologia dividindo-o em tipos de atividades a serem desenvolvidas, descreve o ciclo de vida de uma ontologia, a partir da evolução de

protótipos assim como técnicas específicas para cada atividade executada [BRANDÃO, A. A. F., LUCENA, C. J. P. 2002]. A seguir são apresentadas as atividades e subatividades constituintes da metodologia. A Figura 23 exibe o ciclo de vida de uma ontologia utilizando a metodologia.

1. Atividades de gerenciamento

1.1 Planejamento: desenvolvimento de um plano contendo todas as tarefas a serem desempenhadas, incluindo uma estimativa de quanto tempo, recursos e ferramentas que serão necessárias. Esta atividade é essencial quando se pretende fazer reuso de ontologias existentes;

1.2 Controle: atividade que garante que as tarefas planejadas na fase anterior sejam executadas completamente;

1.3 Garantia de qualidade: atividade que assegura que os produtos resultantes das atividades (ontologia, software, documentação) sejam satisfatórios;

2. Atividades orientadas ao desenvolvimento

2.1 Especificação: define o escopo e objetivo da ontologia. Questionamentos como porque a ontologia será construída, que uso será feito dela e quem serão seus usuários finais, são feitos nesta etapa do desenvolvimento. As respostas destes questionamentos podem ser incluídas em um documento escrito que servirá como especificação para a ontologia;

2.2 Conceituação: atividades de estruturação do domínio de conhecimento da ontologia usando modelos de significado no nível do conhecimento;

2.3 Formalização: formaliza o desenvolvimento do modelo conceitual da atividade anterior num modelo formal;

2.4 Implementação: desenvolvimento da ontologia em uma linguagem ontológica como, por exemplo, OWL;

2.5 Manutenção: atividades de atualização e correção da ontologia.

3. Atividades de suporte - desempenhadas em paralelo ao desenvolvimento

3.1 Aquisição de conhecimento: atividades de aquisição de conhecimento sobre um determinado domínio;

3.2 Avaliação: atividades de julgamento técnico das ontologias, dos ambientes de software associados e da documentação produzida, usando frames de referência;

3.3 Integração: atividades essenciais quando há reuso de ontologias existentes;

3.4 Documentação: atividades de detalhamento claro e exaustivo das fases de desenvolvimento a fim de facilitar a manutenção e reuso de ontologias.

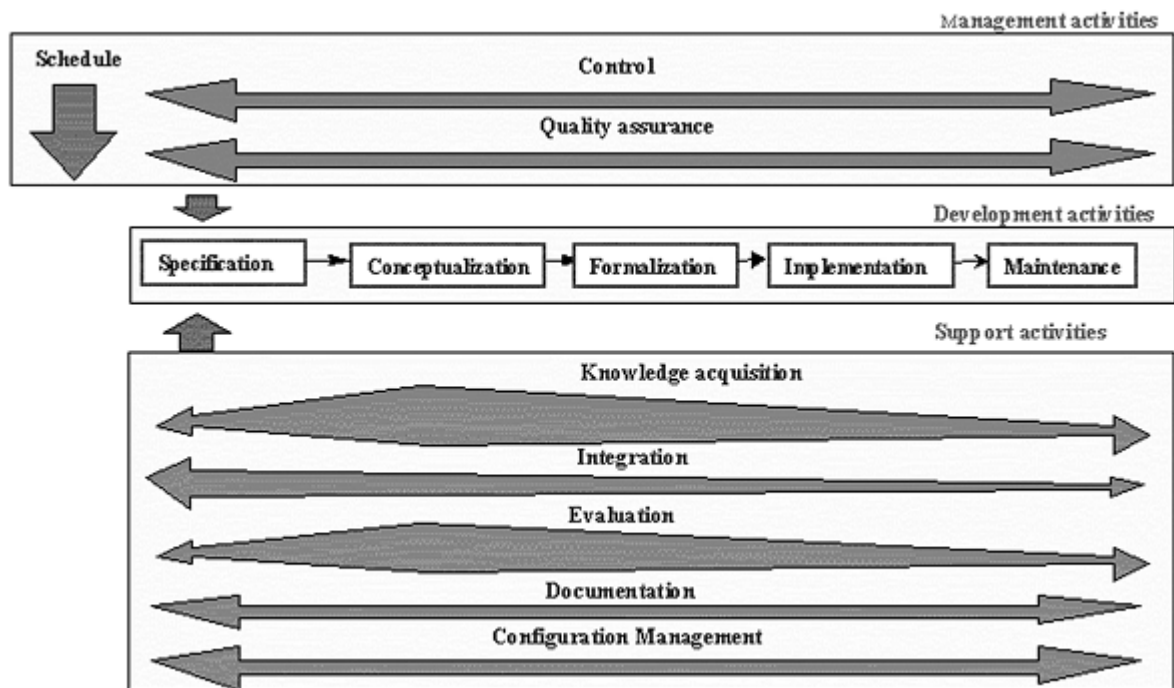


Figura 23 . Ciclo de vida de uma ontologia na metodologia METHONTOLOGY [CORCHO, O. *et al.* 2004]

METHONTOLOGY é muito influenciada pelas metodologias de Engenharia de Software. O método propõe explicitamente uma evolução do ciclo de vida de prototipagem e tem sido continuamente atualizada. O reuso de ontologias é proposto como uma atividade no processo de desenvolvimento [PINTO, H. S., MARTINS, J. P. 2004]. No quadro 6 é apresentada uma classificação das atividades no METHONTOLOGY de acordo com a terminologia usual em Engenharia de Ontologias.

Quadro 6. Classificação das atividades METHONTOLOGY [PINTO, H. S., MARTINS, J. P. 2004]

Atividades METHONTOLOGY	Correspondente
Especificação requerida	Especificação
Conceituação do domínio de conhecimento	Conceituação
Formalização do modelo conceitual	Formalização
Implementação do modelo conceitual	Implementação
Manutenção	Manutenção
Aquisição de conhecimento	Aquisição de conhecimento
Documentação	Documentação
Avaliação	Avaliação

3.4 MÉTODO CYC

De acordo com [SILVA, D. L. *et al* 2008], o método Cyc foi desenvolvido a partir da base de conhecimento Cyc [CYC 2010], uma ampla base de conhecimento que considera o conhecimento consensual sobre o mundo, incluindo regras e heurísticas para dedução sobre objetos e eventos do cotidiano. CycL é a linguagem de representação da Cyc, sendo considerada híbrida por combinar frames com cálculos de predicados. A linguagem também possui uma máquina de inferência que permite herança múltipla, classificação automática, manutenção de links inversos, verificação de restrições, busca ordenada, detecção de contradição e módulo de resolução.

A base de conhecimento Cyc apresenta três processos no seu desenvolvimento:

- Extração do conhecimento de senso comum: obtida de forma manual em fontes diferentes, tais como artigos, livros e jornais.
- Extração auxiliada por computador: através do uso de ferramentas computacionais de processamento de linguagem natural e aprendizado de máquina capazes de usar conhecimento de senso comum suficiente para investigar e descobrir novos conhecimentos.
- Extração gerenciada por computador: conduzido por um número maior de ferramentas no sentido de gerenciar a extração de conhecimento de senso comum (partes consideradas difíceis de serem interpretadas nas fontes de conhecimento envolvidas) na base Cyc.

3.5 METODOLOGIA ON-TO-KNOWLEDGE

A metodologia On-To-Knowledge (OTK) [SURE, Y. *et al.* 2002] tem como principal interesse a construção de grandes sistemas de gestão do conhecimento.

O processo de desenvolvimento desta metodologia é constituído por duas fases – Meta-Processo de Conhecimento e Processo de Conhecimento – apresentadas a seguir.

Meta-Processo de Conhecimento - Esta é a fase de introdução de uma aplicação baseada em ontologia, onde os documentos e o conhecimento necessários são identificados. Este processo consiste das etapas mostradas a seguir:

1ª Etapa – Estudo de viabilidade - ocorre a identificação de problemas e suas possíveis soluções. Em geral, o estudo de viabilidade serve como suporte para determinar as viabilidades econômicas, técnicas e de projeto, determinando o foco do domínio e a solução que deseja atingir. Deve ser realizada antes do desenvolvimento de ontologias e serve como uma base para a fase de inicialização.

2ª Etapa – Inicialização - são realizadas atividades para a identificação dos requisitos, onde são definidos o objetivo, o domínio e o escopo da ontologia. Nesta etapa também pode ser examinada a possibilidade de haver uma integração com ontologias existentes. Em adição, um número de questões de competência devem ser formuladas para capturar os requisitos necessários para a ontologia. Esta gera, como produto final, um documento de requisitos que descreve como a ontologia deve apoiar o domínio de aplicação especificado.

3ª Etapa – Refinamento - ocorre a extração do conhecimento, utilizando o método de extração mais aplicável (*top-down*, *middle-out* ou *bottom-up*) à fonte de conhecimento disponível. É realizada a construção de uma ontologia modelo contendo os conceitos relevantes e descrevendo os relacionamentos entre eles, onde esta ontologia será usada para se chegar à ontologia desejada, expressada numa linguagem formal.

4ª Etapa – Avaliação - nesta fase a ontologia é checada de acordo com o documento de especificação e as questões de competência formuladas anteriormente, sendo que esta fase funciona de maneira iterativa com a anterior, podendo, inclusive, detectar falhas na representação da ontologia. Assim, vários ciclos podem ser executados até a ontologia ser considerada boa o bastante.

5ª Etapa – Aplicação e Evolução - ocorre a aplicação da ontologia em sistemas baseados em ontologia. O engenheiro recebe a recomendação de práticas para lidar a evolução/atualização da ontologia, a manutenção, que faz com que mudanças no mundo real reflitam em mudanças na ontologia.

Processo de Conhecimento - É um processo iterativo da gestão de conhecimento que é aplicado na organização. O processo é composto pelas seguintes etapas:

1ª Etapa – Criação e/ou importação de documentos e metadados: o conhecimento deve ser criado ou adaptado para se moldar aos padrões da organização.

2ª Etapa – Captura do conhecimento: inclusão dos dados que se referem aos conceitos da ontologia.

3ª Etapa – Recuperação e acesso ao conhecimento: visões adicionais do conhecimento podem ser derivadas, permitindo a inferência de relacionamentos e descrições através de um mecanismo de inferência apropriado.

4ª Etapa – Utilização do conhecimento: a utilização e a reutilização do conhecimento são facilitadas pela ontologia, sendo que este conhecimento pode ser derivado ou adicionado de alguma outra fonte para que as metas do usuário sejam alcançadas.

3.6 UM FRAMEWORK PARA ONTOLOGIAS MULTILÍNGÜES

De acordo com [SOUZA, J. F. 2007 (apud LAUSER *et al.*, 2002)], o *framework* adota a metodologia METHONTOLOGY acrescida de modificações em ações específicas para que o processo de criação e evolução de ontologias de domínio seja semi-automático. A construção da ontologia de domínio é feita utilizando duas diferentes abordagens de aquisição de conhecimento:

- **Abordagem de aquisição 1** - criação da ontologia núcleo. Uma pequena ontologia núcleo com os conceitos mais importantes e seus relacionamentos é criada. Este estágio é composto basicamente de três passos das atividades de desenvolvimento da METHONTOLOGY: especificação de requisitos, conceitualização do conhecimento de domínio e formalização do modelo conceitual em linguagem formal.
- **Abordagem de aquisição 2** - Deriva uma ontologia de domínio de um *thesaurus*.
- **Combinação das ontologias** - Combina a ontologia criada manualmente e a ontologia derivada usando os termos do *thesaurus*.

- **Extensão e refinamentos da ontologia** - Os termos mais freqüentes do domínio são usados como possíveis candidatos a conceitos ou relacionamentos para estender a ontologia. Estes termos devem ser avaliados por especialistas do domínio e conferidos pela relevância na ontologia.

3.7 PROCESSO KUP

De acordo com [SOUZA, J. F. 2007 (apud ORLEAN, D. 2003)], o processo KUP (*Knowledge Unified Process*) é um processo unificado para desenvolvimento de ontologias e bases de conhecimento, que deverão ser utilizadas em aplicações voltadas para a Web Semântica. Para garantir a qualidade de desenvolvimento de ontologias, o processo foi baseado nos critérios de avaliação de metodologias para desenvolvimento de ontologias propostas por [FERNÁNDEZ-LÓPEZ, M. GÓMEZ-PÉREZ, A. 2002] e faz a adaptação e integração das melhores práticas de desenvolvimento de software apresentadas pelo RUP (*Rational Unified Process*) (RATIONAL, 1998). O KUP é composto de três fases realizadas de forma iterativa ao longo de todo o processo de desenvolvimento. As fases são as seguintes:

- Fase de Concepção - foco na análise de viabilidade do projeto, na estratégia de desenvolvimento da ontologia, no escopo e levantamento dos requisitos.
- Fase de Construção - foco no projeto, implementação e implantação da ontologia, levando em conta os novos requisitos que possam surgir e mudanças na estratégia.
- Fase de Evolução - foco na integração de novos requisitos ao projeto, incluindo novos conceitos, relações e axiomas na ontologia.

Ainda de acordo com o autor, durante as fases do processo são gerados artefatos a partir de atividades executadas nas disciplinas. Uma disciplina agrupa atividades para integrar um processo específico de um determinado tema e um artefato é tudo o que é produzido, consumido ou modificado por uma atividade, podendo ser um documento, um modelo, um código fonte ou mesmo programas

inteiros. A Figura 24 apresenta as iterações entre as atividades de cada disciplina nas fases propostas pelo KUP.

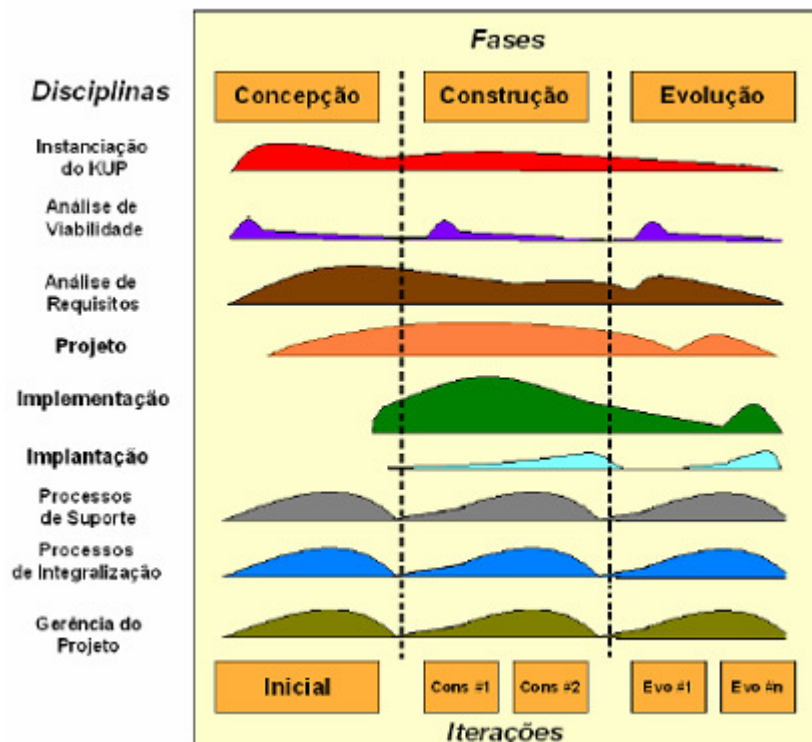


Figura 24. Iterações entre atividades nas fases do processo KUP.

3.8 METODOLOGIA DILIGENT

Segundo [TEMPICH, C. 2006], DILIGENTE (Distributed, Loosely-controlled and evolving Engineering of ontologies) é uma metodologia para construir, desenvolver e conciliar ontologias em configurações DKM (*Distributed Knowledge Management*). Um modelo de processo é definido, orientando os usuários para uma ontologia compartilhada, caracterizando os papéis, especificando o conhecimento adquirido sobre os usuários. A metodologia é apoiada por ferramentas, auxiliando os usuários em cenários de aplicação específica.

A metodologia é constituída por cinco etapas principais. Para todas essas etapas o autor define os papéis principais, as informações de entrada, as decisões, as atividades e as informações de saída. A Figura 25 apresenta a metodologia DILIGENT. As etapas são:

- **Construção:** Para a construção de uma ontologia inicial, participam do processo o especialista do domínio, usuários, engenheiros do conhecimento e engenheiros ontológicos. Inicialmente, a equipe envolvida deve ser relativamente pequena de modo a facilitar a obtenção de uma pequena e consensual primeira versão da ontologia compartilhada.
- **Adaptação local:** uma vez que uma ontologia núcleo está disponível, usuários podem trabalhar com ela e particularmente, podem adaptá-la às suas necessidades locais. Através de um controle central, as adaptações locais são registradas ou enviadas pelos usuários.
- **Análise:** a fim de atualizar a ontologia compartilhada de acordo com os novos requisitos dos usuários, um controle analisa as ontologias locais e as mudanças solicitadas e tenta identificar similaridades na ontologia do usuário. Uma das principais atividades do controle é decidir quais mudanças deveriam ser introduzidas na ontologia compartilhada, visto que nem todas as solicitações serão introduzidas. As mudanças são formalizadas na próxima etapa.
- **Revisão:** O controle revisa regularmente as ontologias locais e a compartilhada. A revisão pode ser considerada como um tipo de desenvolvimento de ontologias guiado por um grupo cuidadosamente equilibrado voltado para os requisitos dos usuários. Engenheiros ontológicos são os responsáveis pela atualização da ontologia, com base nas decisões do controle.
- **Atualização local:** quando uma nova versão da ontologia compartilhada é liberada, usuários podem atualizar as suas ontologias locais. A atualização pode implicar em uma nova reorganização da ontologia local. Os usuários individuais se beneficiam do uso de uma ontologia compartilhada já que ela permite a interoperabilidade com os outros usuários.

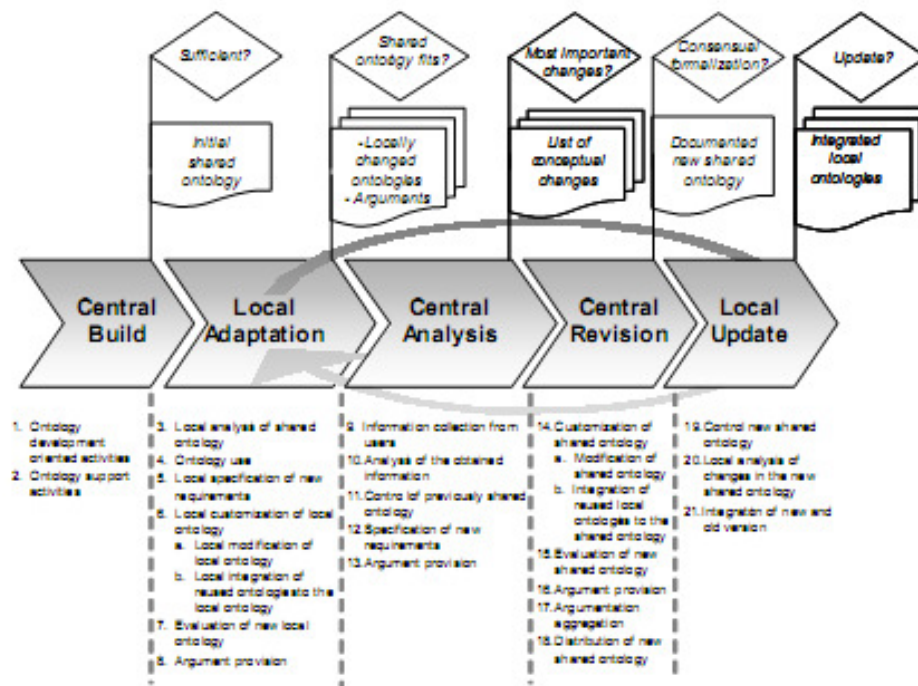


Figura 25. Etapas da metodologia DILIGENT [TEMPICH, C. 2006].

3.9 MÉTODO KACTUS

O método KACTUS é baseado no projeto *Espirit Kactus*, o qual tem como objetivo investigar o reuso do conhecimento em sistemas e o papel da ontologia no processo. O processo de construção de ontologias é fortemente influenciado pela possibilidade de reuso de conceitos definidos em outras ontologias, ou seja, o processo é condicionado pela análise a adaptação de conceitos de ontologias existentes [BREITMAN, K. K. *et al* 2007].

O método KACTUS apresenta as seguintes etapas:

1. Especificação da aplicação: o contexto da aplicação é capturado através da lista dos termos do domínio.
2. Concepção preliminar da ontologia: com base na avaliação de super ontologias disponíveis, categorias relevantes são escolhidas, refinadas, extendidas e incorporada na nova ontologia.
3. Refinamento e estruturação: a ontologia é refinada para reorganizar sua estrutura e para alcançar uma concepção mais modular.

3.10 MÉTODO 101

O método proposto por [NOY, N., McGUINNESS, D. 2001] foi proposto para guiar os usuários na criação de sua primeira ontologia. As etapas que compõe o método serão descritos a seguir:

1. Determinação do domínio e escopo da ontologia:

Para iniciar o desenvolvimento da ontologia, os autores buscam responder questões básicas como:

- Qual é o domínio que a ontologia abrangerá?
- Para que vamos utilizar a ontologia?
- Para que tipos de questões as informações contidas na ontologia devem fornecer respostas?
- Quem usará e manterá a ontologia?

Os autores denotam ainda que as respostas para as questões acima podem mudar durante o processo de desenvolvimento da ontologia, mas num determinado momento podem ajudar a limitar o escopo do modelo.

2. Considerar o reuso de ontologias existentes:

Reutilização de ontologias existentes pode ser um requisito, se o sistema tem que interagir com outras aplicações que fazem uso de ontologias particulares ou vocabulários controlados e muitas ontologias já estão disponíveis em formato eletrônico.

3. Enumeração de termos importantes da ontologia:

É útil para escrever uma lista de todos os termos que se deseja fazer declarações sobre ou para explicar a um usuário. Quais são os termos importantes? Que propriedades esses termos apresentam? O que gostaríamos de dizer sobre esses termos? Inicialmente, é importante para obter uma lista abrangente de termos, sem se preocupar com a sobreposição entre conceitos, as relações entre os termos, ou qualquer propriedade que os conceitos podem ter, ou ainda se os conceitos são classes ou propriedades. As próximas duas etapas – definição de

classes e sua hierarquia e definição das propriedades – estão estreitamente interligados.

4. Definição de classes e sua hierarquia:

Existem várias abordagens possíveis para o desenvolvimento de uma hierarquia de classes. O processo de desenvolvimento *top-down*, inicia com a definição dos termos mais gerais do domínio seguida da especialização dos conceitos. A abordagem *bottom-up*, inicia com a definição das classes mais específicas, com subsequente agrupamento dessas classes em conceitos mais gerais. E finalmente, o processo de combinação, reúne as abordagens *top-down* e *bottom-up*. Primeiro é definido os conceitos se sobressaem e então é feita a generalização e especialização

5. Definição das propriedades

As classes, por si só, não fornecem informações suficientes para responder às questões da competência da etapa 1. Definidas as classes, temos de descrever a estrutura interna dos conceitos. A maioria dos termos criados na etapa 3 são suscetíveis de serem propriedades dessas classes. E para cada propriedade na lista, é preciso determinar qual a classe que descreve.

6. Definição das características das propriedades

Propriedades podem ter diferentes modos para descrever o tipo de valor, valores permitidos, cardinalidade, e outras características que podem ser atribuídas às propriedades. Por exemplo, o valor da propriedade *name* (com em “o nome de um vinho”) é uma *string*. Ou seja, *name* é uma propriedade com o tipo de valor *String*.

7. Criação de instâncias:

O último passo é criar instâncias das classes. Definir uma instância requer (1) escolher uma classe, (2) criar uma instância de que cada classe, e (3) preencher os valores das propriedades.

3.11 CONSIDERAÇÕES SOBRE AS METODOLOGIAS E MÉTODOS PARA O DESENVOLVIMENTO DE ONTOLOGIAS

Como visto nas seções anteriores, as metodologias e métodos para o desenvolvimento de ontologias apresentadas, em geral são divididas em fases. Observa-se, entretanto que estas fases não estão presentes em todas as metodologias e métodos citados. Dessa forma, foi elaborado um quadro comparativo sintetizando uma análise das metodologias e métodos descritos. O quadro 7 a seguir apresenta a análise comparativa.

Quadro 7. Análise comparativa entre metodologias e métodos estudados para a identificação das etapas comuns nos processos de desenvolvimento de ontologias.

	Especificação dos requisitos	Modelagem Conceitual	Formalização	Implementação	Manutenção	Integração	Avaliação	Documentação
USCHOLD E KING	Identifica o propósito e escopo da ontologia	Define os conceitos e relacionamentos textualmente	Ausente	Codificação em linguagem formal	Ausente	Integração com ontologias existentes	Questões de competência	Descreve o processo de construção
TOVE	Cenários motivacionais	Ausente	Especificação dos axiomas	Linguagem de primeira ordem KIF	Ausente	Ausente	Verificação da completude da ontologia	Ausente
METHONTOLOGY	Definição do escopo da ontologia	Atividades envolvendo a conceitualização da ontologia	Modelo conceitual	Ambiente de desenv. adequado à integração	Em todo o ciclo	Integração com ontologias de alto nível	Verificação e validação da ontologia	Em todas as fases.
CYC	Extração manual do conhecimento	Ausente	Ausente	Codificação e extração do conhecimento	Ausente	Ausente	Ausente	Ausente
ON-TO-KNOWLEDGE	Definição dos objetivos, domínio e escopo da ontologia	Construção de uma ontologia modelo	Ausente	Ausente	Aplicação e evolução da ontologia	Utilização de reutilização do conhecimento através de ontologias	Verificação do documento de especificação e questões de competência	Ausente

Framework para Ontologias Multilíngues	Ausente	Construção de uma pequena ontologia núcleo	Definição dos conceitos e relacionamentos	Ausente	Ausente	Ausente	Avaliação dos termos por especialistas	Ausente
KUP	Definição do escopo e levantamento dos requisitos	Ausente	Ausente	Construção do projeto	Adequação à mudanças no projeto	Integração de novos requisitos	Ausente	Ausente
DILIGENT	Especificação do conhecimento	Construção de uma ontologia consensual (1ª versão)	Ausente	Construção de uma ontologia local	Revisão das ontologias locais e a compartilhada	Identificação de similaridades na ontologia local	Verificação de mudanças na ontologia compartilhada	Documentação da ontologia compartilhada
Kactus	Lista de requisitos	Identificação dos termos relevantes	Ausente	Ausente	Ausente	Ausente	Ausente	Ausente
Método 101	Definição de questões de competência	Definição de classes, propriedades, restrições e instâncias	Através de frames	Ferramenta Protégé	Ausente	Considera a reutilização de ontologias	Ausente	Através do protégé

Este capítulo apresentou várias metodologias e métodos existentes que visam o desenvolvimento de ontologias. O quadro 7, tem por objetivo comparar todas as abordagens citadas. Com ele é possível identificar as fases presentes em cada abordagem bem como a forma como a mesma é realizada. Podemos observar que a integração é uma fase que ainda está em amadurecimento, e portanto, muitas vezes é uma etapa ausente ou apenas considerada como atividade às metodologias e métodos.

De acordo com o que foi apresentado, torna-se necessário o desenvolvimento de uma abordagem que faz da Integração uma etapa constituinte. O Capítulo 5 apresenta a abordagem QDAontology que tem como um de seus benefícios a introdução da Integração como uma das etapas de seu processo. Entretanto, antes de apresentar a abordagem, é de fundamental importância mostrar como pode ser aplicada a interoperabilidade entre ontologias na etapa de Integração. Com isso o Capítulo 4 aborda a interoperabilidade de ontologias, apresentando os principais mecanismos e medidas de similaridade, os quais podemos aplicar.

4 INTEROPERABILIDADE ENTRE ONTOLOGIAS

Atualmente a utilização e reuso de informações são fundamentais haja vista que uma quantidade significativa dessas informações está sendo gerada. Entretanto, o uso e reuso dessas informações podem gerar alguns problemas, entre os quais podemos citar como exemplo diferentes formatos de representação do conhecimento e inconsistências semânticas. No sentido de resolver esses problemas, a interoperabilidade entre ontologias auxilia no uso e reuso de informações.

A interoperabilidade semântica pode ser definida como sendo a capacidade de dois ou mais sistemas heterogêneos e distribuídos trabalharem em conjunto, compartilhando as informações entre eles com entendimento comum de seu significado [SOUZA, J. F. 2007 (apud BURANARACH, M. 2001)].

Neste contexto as ontologias têm um papel importante e podem ser reutilizadas de diversas formas: ora resultando na criação de uma ontologia independente a partir dos conceitos de outras (podendo ser estendidas e adaptadas), ora preservando as ontologias originais mas expandindo as mesmas [CAMPOS, M. L. A. *et al.*, 2009].

Diante deste cenário, vários mecanismos têm sido propostos com o objetivo de apoiar a interoperabilidade de ontologias e recebem o nome de *Ontology Mapping* ou Mapeamento de Ontologias. Em [SILVA, N. A. P. 2004] o mapeamento ontológico é definido como sendo o processo de identificação, especificação e aplicação de relações equivalentes entre duas ontologias. Os mecanismos mais utilizados são a combinação, a integração, o mapeamento e o alinhamento de ontologias. O alinhamento de ontologias, foco deste trabalho, geralmente é aplicado em sistemas de domínios complementares e apresenta como resultados as ontologias originais separadas, porém ligações entre os termos equivalentes são adicionadas.

Segundo [SILVA, N. A. P. 2004], alinhamento ou mapeamento de ontologias é um processo que consiste na definição, a nível ontológico, dessas relações semânticas entre entidades de uma ontologia de origem e de uma ontologia de destino. Essas relações são posteriormente aplicadas na transformação das instâncias de uma base de conhecimento, de acordo com a ontologia de origem, em

instâncias da base de conhecimento de acordo com a ontologia de destino. Essa tecnologia é considerada de fundamental importância em cenários em que a troca e partilha de informação sejam essenciais. Interoperabilidade entre sistemas de informação, web semântica, organizações virtuais e negócios eletrônicos, migração de dados entre sistemas e evolução dos modelos de dados subjacentes aos sistemas de informação, são alguns desses cenários de uso.

Alinhamento semântico entre ontologias é uma condição necessária para estabelecer a interoperabilidade entre agentes ou serviços que utilizam diferentes ontologias [EHRIG, M., STAAB, S. 2004]. Em [SILVA, N. A. P. 2004] são apontados requisitos necessários de um sistema de mapeamento de ontologias:

1. Identificação, especificação e representação de relações sintáticas, estruturais e semânticas entre ontologias;
2. Transformação da informação transmitida entre intervenientes na comunicação, de acordo com as relações anteriores;
3. Negociação das relações anteriores;
4. Manutenção das relações anteriores;
5. Integração (mas minimização) da participação do ser humano no processo de mapeamento de ontologias, o que sugere a adoção de um sistema semi-automático de mapeamento de ontologias;
6. Adoção de tecnologia e soluções do contexto da web semântica.

Contudo, aplicar apenas o alinhamento não garante a qualidade da interoperabilidade entre as ontologias. Para tanto, a adoção de medidas de similaridade se faz necessário como meio de se obter resultados mais eficientes e confiáveis. De acordo com [SOUZA JUNIOR, H. C. 2008], quanto mais aprimorado for este mecanismo, melhores resultados poderão ser fornecidos aos mecanismos de interoperabilidade, propiciando maior grau de automatização do processo.

A seguir são apresentados conceitos relacionados à interoperabilidade de ontologias. São descritos os mecanismos de interoperabilidade, bem como tipos de medidas de similaridade e tecnologias para alinhamentos.

No contexto desta dissertação adotamos alinhamento como sendo o conjunto de correspondências entre duas ontologias. Estas ligações permitem que as ontologias alinhadas reusen informações umas das outras. Como resultado temos duas ontologias originais separadas.

4.1 CORRESPONDÊNCIA ONTOLÓGICA

De acordo com [EUZENAT, J., SHVAIKO, P. 2007], a correspondência é uma importante operação em aplicações como integração ontológica, integração de esquemas ou *data warehouses*, já que as mesmas são caracterizadas por modelos estruturais heterogêneos. Essa heterogeneidade pode ocorrer de diversas maneiras, entre as quais, destaca-se a sintática, a terminológica e a conceitual.

Segundo [SHVAIKO, P. EUZENAT, J. 2008], os avanços nos estudos sobre a área de correspondência ontológica (*ontology matching*) nos últimos anos têm acarretado alguns desafios, entre eles: (i) avaliação em larga-escala, (ii) desempenho das técnicas de correspondência semântica, (iii) descoberta de conhecimento do contexto, (iv) incerteza na correspondência ontológica, (v) seleção dos correspondentes e auto-configuração, (vi) usuários envolvidos, (vii) explicação dos resultados, (viii) correspondência ontológica social e colaborativa, (ix) gerenciamento dos alinhamentos: infraestrutura e suporte, e (x) inferências com alinhamentos.

Existem várias maneiras de se classificar as técnicas de correspondência ontológica, nesse estudo nos pautamos nos estudos de [EUZENAT, J., SHVAIKO, P. 2007]. Segundo o autor uma dessas formas seria classificar algoritmos em função (i) da entrada do algoritmo, (ii) das características do processo de correspondência, e (iii) da saída do algoritmo.

Os autores também apresentam outra classificação para a correspondência ontológica. Para isso é utilizada a seguinte classificação sintética [EUZENAT, J., SHVAIKO, P. 2007]:

- *Granularity/Input Interpretation* – Classificação é baseada (i) nos níveis estrutural e do elemento, e (ii) em como as técnicas interpretam a informação de entrada.
- *Kind of Input* – Classificação é baseada no tipo de entrada usada pelas técnicas de correspondência.

A camada ***Granularity/Input Interpretation*** é distinguida de acordo com o seguinte critério de classificação [EUZENAT, J., SHVAIKO, P. 2007]:

- Nível estrutural x Nível do elemento: nível do elemento computa correspondentes através da análise de entidades ou instâncias, ignorando

suas relações com outras entidades ou suas instâncias. Nível estrutural computa correspondentes através da análise de como as entidades ou instâncias aparecem juntas na estrutura.

- Sintática x Externa x Semântica: a característica chave da técnica Sintática é a interpretação da entrada no que diz respeito à sua estrutura única de acordo com o algoritmo. Externas são as técnicas que exploram recursos auxiliares (externos) de um domínio e conhecimento comum, a fim de interpretar a entrada. Estes recursos podem ser entradas ou tesouros expressando relações entre os termos. Semântica utiliza semântica formal para interpretar a entrada e justificar os resultados.

Já a camada ***Kind of Input*** é dividida em dois níveis [EUZENAT, J., SHVAIKO, P. 2007]:

- O primeiro é categorizado de acordo com o tipo de dado que o algoritmo trabalha: strings (terminológica), estrutura (estrutural), modelos (semântica) ou instâncias (extensão). Os dois primeiros são encontrados nas descrições da ontologia. O terceiro exige uma interpretação semântica da ontologia. E o último constitui uma população real de uma ontologia.
- O segundo decompõe adicionais dessas categorias se necessário: métodos terminológicos podem ser baseados em string (considerando os termos como sequências de caracteres) ou com base na interpretação destes termos como objetos linguísticos (linguística). Os métodos estruturais são divididos em dois tipos: os que consideram a estrutura interna das entidades, por exemplo, os atributos e seus tipos (interno), e aqueles que consideram a relação das entidades com outras entidades (relacionais).

A Figura 26 exibe a classificação das técnicas de correspondência ontológica proposta por [EUZENAT, J., SHVAIKO, P. 2007].

O objetivo da correspondência ontológica é descobrir as relações entre entidades expressas em diferentes ontologias. Muitas vezes, essas relações são relações de equivalência que são descobertas através de medidas de similaridade entre as entidades das ontologias [EUZENAT, J., SHVAIKO, P. 2007].

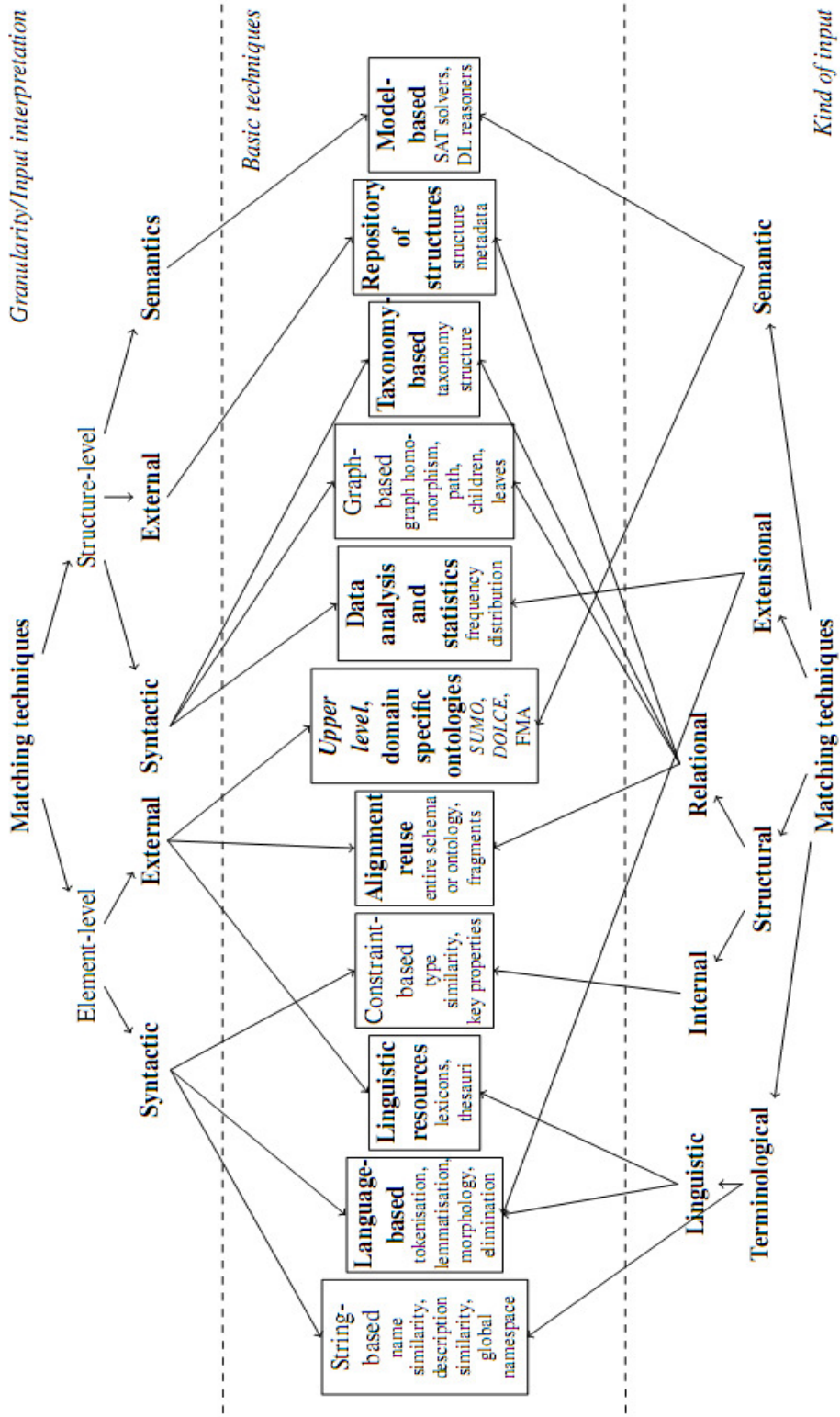


Figura 26. Classificação das técnicas de correspondência ontológica (EUZENAT . J., SHVAIKO. P. 2007).

4.2 MECANISMOS DE INTEROPERABILIDADE

As ontologias são construídas e aplicadas em diversos domínios e de diversas formas. Conseqüentemente, a possibilidade de trocar informações e reutilizar o conhecimento fornecido pelas ontologias tem gerado a necessidade de criação de mecanismos que promovam a interoperabilidade ontológica.

Combinação, alinhamento, integração e mapeamento de ontologias são os principais mecanismos que possibilitam a interoperabilidade. Uma breve descrição destes mecanismos é feita a seguir. Outros mecanismos podem ser encontrados em [EUZENAT, J., SHVAIKO, P. 2007].

4.2.1 Combinação

De acordo com [SOUZA JUNIOR, H. C. 2008] na combinação de ontologias tem-se como resultado a versão das ontologias originais combinadas em uma única ontologia, com todos seus termos juntos e sem a definição clara de suas origens. Geralmente as ontologias originais descrevem domínios similares ou de sobreposição.

Para [EUZENAT, J., SHVAIKO, P. 2007] combinação é a criação de uma nova ontologia a partir de duas ontologias fontes, possivelmente de domínios similares ou sobrepostos, onde as ontologias iniciais permanecem inalteradas. Este conceito está intimamente relacionado com o esquema de integração em bases de dados.

A Figura 27 apresenta um exemplo de combinação de ontologias "O1" e "O2" (Figura 27a), onde os dois conceitos compatíveis, carro da ontologia O1 e veículo da ontologia O2, são combinados (Figura 27b) na ontologia única O (Figura 27c) [SOUZA JUNIOR, H. C. 2008].

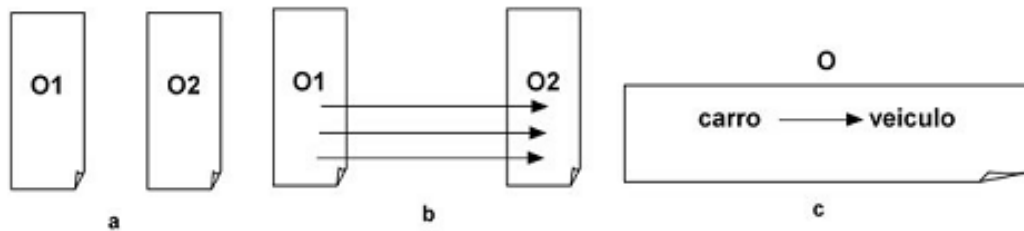


Figura 27. Combinação de ontologias [SOUZA JUNIOR, H. C. 2008].

4.2.2 Alinhamento

É um conjunto de correspondências entre duas ou mais ontologias. O alinhamento é o resultado do processo de correspondência [EUZENAT, J., SHVAIKO, P. 2007].

No alinhamento de ontologias tem-se como resultado as duas ontologias originais separadas, às quais são adicionadas ligações entre seus termos equivalentes. Estas ligações permitem que as ontologias alinhadas reusen informações umas das outras. O alinhamento normalmente é realizado quando as ontologias são de domínios complementares [SOUZA JUNIOR, H. C. 2008].

O alinhamento difere da junção e integração em relação ao seu resultado; em vez de gerar uma ontologia adicional, resultado da combinação das ontologias reutilizadas, o alinhamento mantém as ontologias reutilizadas inalteradas e em seus locais de origem, porém gera um conjunto de vínculos (links) entre essas ontologias. Esses vínculos contêm um conjunto de informações sobre como compatibilizar as ontologias reutilizadas e são expressos em um modelo persistente (que existe fisicamente) em separado. O conjunto de vínculos expressos em um modelo persistente produzido pelo processo de alinhamento é um mapeamento (*mapping*) entre as ontologias. As informações contidas no mapeamento vão depender do tipo de vínculo semântico encontrado entre os elementos e do tipo de formalismo utilizado na ontologia para representar a sua semântica [CAMPOS, M. L. A. *et al* 2009].

A Figura 28 exibe o alinhamento de duas ontologias "O1" e "O2" (Figura 28a), onde seus conceitos são alinhados (Figura 28b) gerando duas ontologias, uma com informações do conceito carro de "O1" alinhado ao conceito veículo de "O2"

(Figura 28c) e outra com o conceito veículo de "O2" alinhado com o conceito carro de "O1" (Figura 28d) [SOUZA JUNIOR, H. C. 2008].

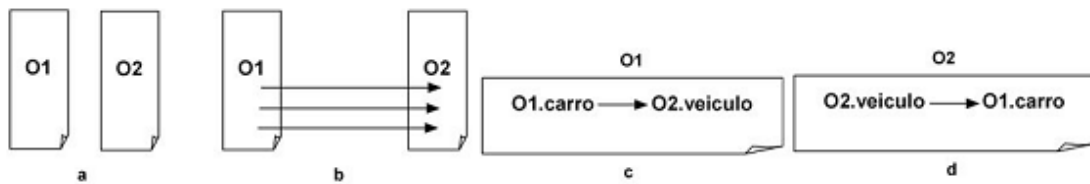


Figura 28. Alinhamento de ontologias [SOUZA JUNIOR, H. C. 2008].

4.2.3 Integração

O conceito integração, segundo [SOUZA JUNIOR, H. C. 2008 (apud PINTO *et al*, 1999)], apresenta três diferentes significados:

- i. Na construção de uma nova ontologia reusando (pela união, extensão, especialização ou adaptação) termos ou conceitos de outras ontologias já existentes;
- ii. Na construção de ontologias utilizando o processo de combinação de várias ontologias, tendo como resultado final uma única ontologia que representa todas que sofreram a fusão. Esse tipo de integração equivale à abordagem já descrita, de combinação de ontologias; e
- iii. Na construção de uma aplicação que usa uma ou mais ontologias.

Resumidamente, podemos definir a integração como sendo a inclusão de uma ontologia em outra ontologia e afirmações expressando a cola entre as ontologias, geralmente como ponte de axiomas. A ontologia integrada contém o conhecimento de ambas as ontologias iniciais. Na integração de ontologias tem-se como resultado uma ontologia única criada pela montagem, extensão, especialização ou adaptação de outras ontologias de assuntos diferentes [SOUZA JUNIOR, H. C. 2008].

A Figura 29 apresenta um exemplo de integração de duas (Figura 29a) ontologias, onde os conceitos são integrados (Figura 29b) em uma única ontologia O (Figura 29c), onde os conceitos veículo de "O2" e carro de "O1" são representados com a indicação de suas origens" [SOUZA JUNIOR, H. C. 2008].

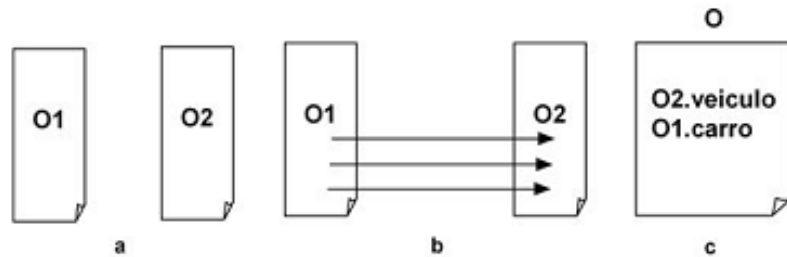


Figura 29. Integração de ontologias [SOUZA JUNIOR, H. C. 2008].

4.2.4 Mapeamento

É considerado a versão orientada de um alinhamento. Ele mapeia as entidades ontológicas para, no máximo, uma entidade de outra ontologia. Um mapeamento pode ser visto como uma coleção de regras de mapeamento todas orientadas na mesma direção, ou seja, a partir de uma ontologia para a outra, os elementos da ontologia fonte aparecem no máximo uma vez [EUZENAT, J., SHVAIKO, P. 2007].

No mapeamento de ontologias tem-se como resultado uma estrutura formal com expressões que ligam os termos de uma ontologia aos termos de outra ontologia. Este mapeamento pode ser usado para transferir instâncias de dados, esquemas de integração e de combinação e outras tarefas similares [SOUZA JUNIOR, H. C. 2008].

Segundo [CHOI, N., SONG, I., HAN, H. 2006], o mapeamento de ontologias é necessário para combinar ontologias distribuídas e heterogêneas. O desenvolvimento de tais mapeamentos tem sido uma questão central para os pesquisadores da área. O autor ainda classifica o mapeamento de ontologias em três categorias: (a) mapeamento entre uma ontologia global integrada e ontologias locais, (b) mapeamento entre ontologias locais, e (c) mapeamento sobre junção de ontologias ou alinhamento.

A Figura 30 apresenta um exemplo de mapeamento de ontologias de duas ontologias "O1" e "O2" (Figura 30a), onde seus conceitos são mapeados gerando um arquivo com informações do mapeamento realizado entre os conceitos carro de "O1" e veículo de "O2" (Figura 30b). Na Figura 30c têm-se as respectivas instâncias destes conceitos, sendo que as instâncias do conceito carro da ontologia "O1" serão copiadas como instâncias do conceito veículo da ontologia "O2", fazendo uso das

informações de mapeamento que foram geradas durante o processo (Figura 30d) [SOUZA JUNIOR, H. C. 2008].

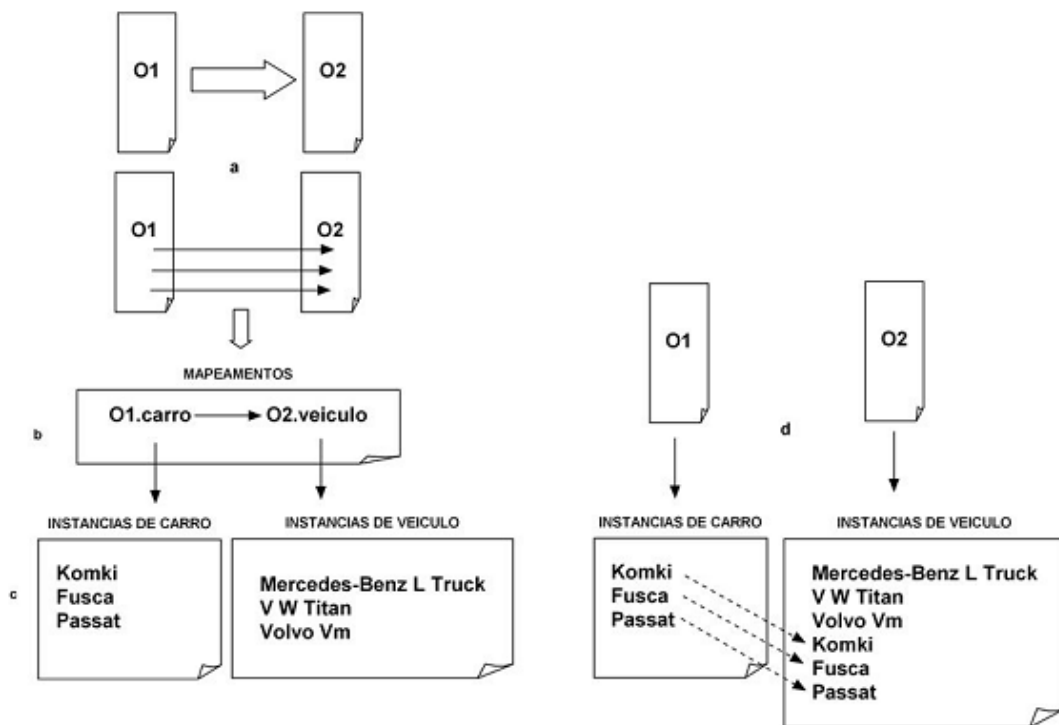


Figura 30. Mapeamento de ontologias [SOUZA JUNIOR, H. C. 2008].

4.3 MEDIDAS DE SIMILARIDADE

Formalmente, a similaridade pode ser definida como [EUZENAT, J., SHVAIKO, P. 2007]:

Uma similaridade $\sigma : o \times o \rightarrow \mathbb{R}$ é uma função de pares de entidades para um número real expressando a similaridade entre dois objetos tais como:

$$\forall x, y \in o, \sigma(x, y) \geq 0 \quad (\text{positividade})$$

$$\forall x \in o, \forall y, z \in o, \sigma(x, y) \geq \sigma(y, z) \quad (\text{maximalidade})$$

$$\forall x, y \in o, \sigma(x, y) = \sigma(y, x) \quad (\text{simetria})$$

Como forma de garantir que os alinhamentos tenham um alto grau de confiabilidade e uma maior eficiência, é necessário adotar medidas de similaridades. Quanto mais aprimoradas forem as técnicas de similaridade, melhores serão os

resultados fornecidos aos mecanismos de interoperabilidade, propiciando assim maior grau de automatização do processo [SOUZA JUNIOR, H. C. 2008].

Na literatura [SOUZA JUNIOR, H. C. 2008], [EUZENAT, J., SHVAIKO, P. 2007]. [SOUZA, J. F. 2007], [FREITAS, J. B. 2007] encontram-se várias propostas que viabilizam o cálculo de similaridades entre termos ontológicos. Para comparar duas ontologias e medir a similaridade entre elas (ou entre parte delas), podem-se considerar diferentes níveis ontológicos, tais como o léxico, onde é investigado como os termos são usados para transmitir significados, e o conceitual, onde são investigadas as relações conceituais existentes entre os termos.

Atualmente, existem métodos com base nos nomes das entidades, estrutura, extensão e semântica. A seguir apresenta-se uma breve descrição de algumas técnicas.

4.3.1 Técnica baseada no nome da entidade

Este tipo de técnica compara *strings* e pode ser aplicada em nomes, rótulos ou em comentários dos termos ontológicos, a fim de encontrar aqueles que são semelhantes. Isto pode ser usado para comparar nomes das classes e/ou URIs (*Uniform Resource Identifier*).

O problema dessa técnica ocorre devido à existência de sinônimos e homônimos. Conseqüentemente, não é possível dizer com certeza se dois termos são similares ou não.

Atualmente, existem muitos métodos que comparam *strings*, dentre os mais utilizados podemos citar: normalização, técnicas de *substring* ou subsequência, distância de edição (*edit distance*) e comparação de caminhos. O uso destes métodos permite a obtenção de resultados mais consistentes.

A normalização é uma técnica utilizada para reduzir *strings* para serem comparados a um formato comum. De acordo com [SOUZA, J. F. 2007], essa fase pode se preocupar em converter letras em caixa alta ou caixa baixa, plurais ou singulares, retirar acentos ou espaços.

A comparação por prefixos e sufixos é um outro método utilizado na comparação de *strings*. Neste método compara-se se uma palavra começa ou

termina como a outra e pode ser realizada após a normalização. Porém, segundo [SOUZA, J. F. 2007], o método convive com problemas como, por exemplo, encontrar alta similaridade entre palavras como “bola” e “bolacha” ou “espaço” e “aço”.

O método denominado distância de edição (*edit distance*) pode ser definido como sendo o custo mínimo de ações a ser aplicado a um dos objetos, a fim de obter o outro. Ele foi projetado para medir a semelhança entre cadeias de caracteres que podem conter erros de ortografia. Segundo [MAEDCHE, A., STAAB, S. 2002], a distância de edição formulada por *Levenshtein* é um método bem estabelecido para a ponderação da diferença entre duas *strings*. Ele mede o número mínimo de *token* de inserções, deleções e substituições necessárias para transformar uma *string* em outra usando um algoritmo de programação dinâmica. A distância de Levenshtein para as palavras “bolha” e “bola” é igual a 1, dado que somente uma operação é necessária para transformar uma palavra na outra: excluindo a letra “h” de bolha ou inserindo a letra “h” após o “l” em bola [SOUZA, J. F. 2007].

E finalmente, a comparação de caminhos consiste em comparar não apenas os nomes das classes, mas a sequência das entidades às quais estão relacionadas. Em uma primeira aproximação, esta pode ser considerada como uma forma particular de agregar *tokens* de forma ordenada. Um simples (e único) exemplo é o que concatena os nomes de todas as superclasses de classes antes de compará-las. Portanto, o resultado depende da comparação da sequência do indivíduo agregada em algum caminho [EUZENAT, J., SHVAIKO, P. 2007].

4.3.2 Técnica baseada na estrutura da entidade

A técnica compara a estrutura das entidades que podem ser encontradas nas ontologias. A comparação utilizada ainda pode ser subdividida em uma comparação interna de uma entidade, ou seja, além do nome e anotações, as propriedades são comparadas, ou, em uma comparação da estrutura relacional das ontologias [EUZENAT, J., SHVAIKO, P. 2007].

Os métodos baseados na estrutura interna das entidades utilizam como critério suas propriedades, o alcance (*range*) das suas propriedades (atributos e

relações), sua cardinalidade ou multiplicidade, transitividade e/ou simetria das suas propriedades para calcular a similaridade entre elas. Geralmente, esses métodos são utilizados em combinação com outros métodos com o objetivo de reduzir o número de candidatos correspondentes. Isso se deve ao fato de que, a comparação de estruturas ou propriedades de duas ontologias com domínios e alcances (*range*) similares pode ser bastante numerosa. O método ainda pode ser usado com outras abordagens como uma etapa de pré-processamento para eliminar a maioria das propriedades claramente incompatíveis. A Figura 31 ilustra um exemplo deste método, onde são consideradas as propriedades associadas a *Product* e *Volume* [EUZENAT, J., SHVAIKO, P. 2007].

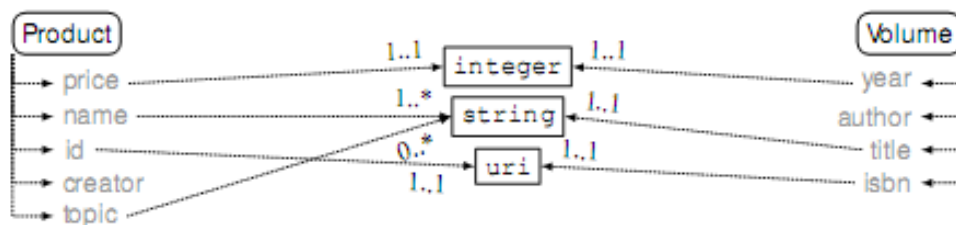


Figura 31. Comparação de dois conjuntos de propriedades [EUZENAT, J., SHVAIKO, P. 2007].

Já os métodos baseados na estrutura relacional, avaliam a organização das classes das ontologias, procurando por pais ou filhos em comum. Neste caso, a ontologia pode ser considerada como um grafo orientado cujas arestas representam as relações entre os termos. Encontrar a correspondência entre os elementos do grafo corresponde a encontrar um subgrafo orientado máximo comum.

4.3.3 Técnica baseada na extensão da entidade

A técnica compara os indivíduos que podem ser encontrados nas entidades da ontologia. Quando ontologias compartilham o mesmo conjunto de indivíduos, a busca por elementos correspondentes é altamente facilitada visto que se duas delas apresentam o mesmo conjunto de indivíduos, então pode haver um alto grau de correspondência entre as mesmas [EUZENAT, J., SHVAIKO, P. 2007].

Os métodos baseados nos indivíduos podem ser divididos em três categorias: aqueles que se aplicam a ontologias com conjuntos de instâncias

comuns, aqueles que propõem técnicas de identificação individual, e aqueles que não necessitam de identificação, ou seja, trabalham em conjuntos heterogêneos de instâncias.

A comparação de indivíduos é muito útil quando um conjunto de indivíduos caracterizados em ambas as ontologias está disponível. Mas há situações, em que as informações das instâncias não estão disponíveis e, portanto, outras técnicas devem ser adotadas.

4.3.4 Técnica baseada na semântica da entidade

A técnica tem grande aplicabilidade na tarefa de junção de duas ontologias, mas também pode ser utilizada para a busca por correspondentes. Ela é considerada um método dedutivo visto que é usada para justificar resultados obtidos por outros métodos. Porém, métodos dedutivos, por sua vez, não funcionam muito bem sozinhos na tarefa de achar correspondentes ontológicos, e, portanto, há a necessidade de se fazer uma fase de pré-processamento que forneça “âncoras”. Dessa forma, devem-se incluir métodos para ligar as ontologias. Tais métodos são baseados no uso de recursos formais existentes para o início de um alinhamento que pode ser ainda considerado um método dedutivo. Como exemplo dos métodos dedutivos tem-se as técnicas proposicional e de descrição lógica [EUZENAT, J., SHVAIKO, P. 2007].

Apesar desse tipo de técnica não poder encontrar os termos correspondentes sozinha, ela é de fundamental importância na geração das correspondências de modo a garantir a integralidade, isto é, encontrar todos os termos similares de forma que se assegure a coerência do alinhamento.

4.4 TECNOLOGIAS DE ALINHAMENTO

A interoperabilidade de ontologias é um fator essencial em sistemas onde o reuso e troca de informações são necessários. Contudo, para a concretização da

interoperabilidade, faz-se necessário o uso de tecnologias para a manipulação de ontologias. Na literatura podem ser encontradas várias tecnologias que apóiam esta atividade. Esta seção descreve algumas destas tecnologias disponíveis para o alinhamento de ontologias, tendo em vista que nosso estudo de caso irá se basear neste mecanismo de interoperabilidade.

4.4.1 NOM – Naive Ontology Mapping

Naive Ontology Mapping – NOM [EHRIG, M., STAAB, S. 2004] constitui uma base de comparações futuras. É definida pelas etapas do processo de modelo abaixo:

- 1) Feature Engineering. Primeiramente, as ontologias devem estar representadas em RDFS.
- 2) Search Step Selection. Todas as entidades da primeira ontologia são comparadas com todas as entidades da segunda ontologia.
- 3) Similarity Computation. A similaridade computacional entre uma entidade de O1 e uma entidade de O2 é feita usando funções de similaridade. Cada função de similaridade é baseada nas características de ambas as ontologias e na respectiva medida de similaridade. A ontologia correspondente é indicada através de índices que são apresentados no Quadro 7.
- 4) Similarity Aggregation. NOM destaca similaridades individuais altas e dá pouco enfoque para as similaridades individuais baixas por ponderação dos resultados de similaridade individual, primeiro com uma função sigmóide seguida da soma dos valores modificados.
- 5) Interpretation. NOM interpreta resultados de similaridade a partir de dois recursos. Primeiro, aplica-se um limite para descartar falsas evidências de similaridade. Em segundo lugar, aplica a objetividade do mapeamento ignorando mapeamentos candidatos que violem esta restrição e favorecendo mapeamentos candidatos com maior similaridade de agregação.
- 6) Iteration. A primeira fase usa somente o método de comparação baseado nos *labels* e similaridade entre as *strings* para computar a

similaridade entre as entidades. Ao fazer o cálculo em várias etapas um pode acessar os pares já computados e usar uma medida de similaridade mais sofisticada. Portanto, na segunda fase e, posteriormente, NOM, pode recorrer a todas as funções de similaridade listadas no Quadro 8.

Quadro 8. Características e Medidas de Similaridade para Diferentes Tipos de Entidade em NOM [EHRIG, M., STAAB, S. 2004].

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

4.4.2 QOM – Quick Ontology Mapping

QOM é uma otimização de NOM pela necessidade de um algoritmo de mapeamento mais eficiente [EHRIG, M., STAAB, S. 2004]. É constituído pelas seguintes etapas:

- 1) Feature Engineering. Segue a abordagem NOM explorando as triplas RDFS.
- 2) Search Step Selection. O número de pares candidatos ao mapeamento pode ser considerado um grande problema de

complexidade, que deve ser comparado para encontrar o melhor mapeamento. QOM resolve isso usando heurísticas para diminuir o número de mapeamentos candidatos. Primeiro tem-se os mapeamentos candidatos que devem ser investigados. Em segundo lugar, uma agenda ordena os mapeamentos candidatos de modo significativo e eficiente, eliminando alguns deles completamente. A análise completa de um mapeamento candidato pode ou não trazer um novo mapeamento candidato na agenda do dia. A estratégia QOM faz uma pesquisa sobre o mais promissor mapeamento candidato

- 3) Similarity Computation. QOM evita uma comparação completa de pares na árvore a favor de uma estratégia *top-down*.
- 4) Similarity Aggregation. Utiliza o mesmo método adotado por NOM.
- 5) Interpretation. Utiliza o mesmo método adotado por NOM.
- 6) Iteration. QOM interage para buscar mapeamentos primeiro com base no conhecimento léxico e depois na estrutura do conhecimento.

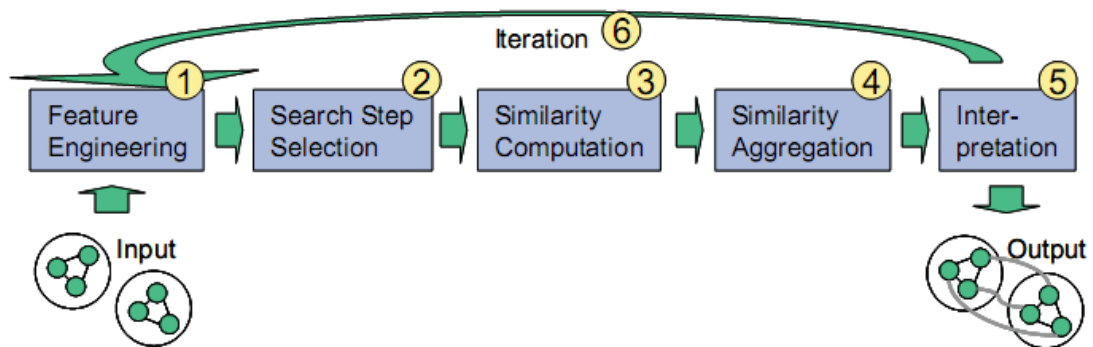


Figura 32. Processo de alinhamento ontológico [EHRIG, M., STAAB, S. 2004].

4.4.3 FOAM – Framework for Ontology Alignment and Mapping

A tecnologia Framework for Ontology Alignment and Mapping – FOAm [EHRIG, M., SURE, Y. 2005] apresenta os seguintes passos:

- Feature Engineering: selecionar trechos da definição da ontologia global para descrever um específico. Isso inclui características individuais, como por exemplo, características estruturais, mas também características mais complexas como usado em OWL, por exemplo, restrições.
- Search Step Selection: escolher duas entidades de duas ontologias para comparar (e1,e2).
- Similarity Assessment: indicar a similaridade para uma dada descrição (ou recurso) de duas entidades (por exemplo, $\text{sim}_{\text{superConcept}}(e1,e2)=1.0$).
- Similarity Aggregation: agregar as várias avaliações de similaridade para um par de entidades em uma única medida.
- Interpretation: usar todos os números agregados, um princípio e uma estratégia de interpretação para propor o alinhamento ($\text{align}(e1) = 'e2'$). Isso também pode incluir uma validação do usuário.
- Iteration: como a similaridade de um alinhamento influencia na similaridade dos pares de entidades vizinhos, a igualdade é propagada através de ontologias.

Finalmente, são recebidos alinhamentos ligando as duas ontologias.

4.4.4 PROMPT

A ferramenta PROMPT [NOY, N. F., MUSEN, M. A. 2003] foi desenvolvida para o gerenciamento de múltiplas ontologias como uma extensão do ambiente Protégé [PROTÉGÉ 2009]. Dessa forma, as ferramentas do PROMPT foram implementadas como um conjunto de plugins do Protégé. O PROMPT é constituído por quatro ferramentas: iPROMPT, AnchorPROMPT, PROMPTDiff e PROMPTFactor. A Figura 33 apresenta a infraestrutura d PROMPT e as interações que ocorrem entre as ferramentas que a compõe.

A ferramenta iPROMPT é destinada para a combinação de ontologias, sugerindo os termos das ontologias que devem ser combinados, identificando

inconsistências e problemas potenciais e sugerindo estratégias para resolvê-los. Porém, as informações entre os conceitos são analisadas somente no contexto local, isto é, apenas as informações das relações entre os conceitos ligados diretamente e aqueles que são referenciados nas restrições destas relações são analisadas

A AnchorPROMPT é uma ferramenta baseada em grafos para alinhamento de ontologias. Ela fornece pares de termos relacionados para ajudar os usuários no mapeamento e alinhamento de ontologias. Como entrada, a ferramenta tem um conjunto de pares de termos relacionados, denominados por âncoras, das ontologias originais. Esses pares podem ser identificados manualmente pelos usuários ou por identificação automática gerada pelo sistema. A partir desse conjunto de âncoras previamente identificado, é produzido um novo conjunto de pares de termos semanticamente similares. AnchorPROMPT trata cada ontologia como um grafo direcionado. Cada classe é representada como um nó e os relacionamentos são as ligações. Os caminhos do sub-grafo limitado pelas âncoras são analisados e são determinados quais os conceitos que freqüentemente aparecem nas mesmas posições dos caminhos similares. Com estas análises e determinações, além do contexto local, o contexto não-local também é analisado. Além disto, utiliza, na detecção de termos similares, a relação estrutural dos termos de ontologias comparadas, medidas de similaridades pré-definidas e grupos de equivalência.

O PROMPTDiff é uma ferramenta para a identificação de diferenças entre duas versões de uma mesma ontologia. Finalmente, a PROMPTFactor é uma ferramenta utilizada na extração de partes da ontologia.

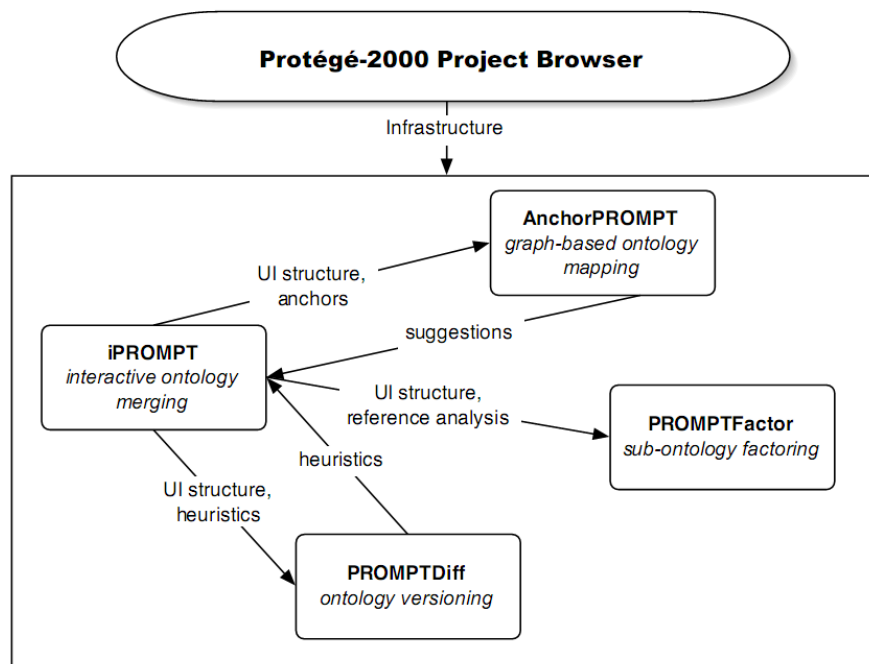


Figura 33. A Infraestrutura do PROMPT e interações entre suas ferramentas.

4.4.5 GLUE

O sistema GLUE [EUZENAT, J., SHVAIKO, P. 2007 (apud DOAN ET AL., 2004)], é um sistema que utiliza várias técnicas de aprendizado de máquina semi-automática para descobrir mapeamentos semânticos entre duas ontologias. A idéia do método é calcular a distribuição conjunta das classes, em vez de se comprometer com uma definição particular da similaridade. Nesta abordagem, dadas duas taxonomias e suas instâncias associadas, para cada nó, o sistema encontra o nó mais similar na outra taxonomia, dada uma medida de similaridade pré-definida. É previsto no sistema a utilização da maioria das medidas práticas de similaridades conhecidas. Trata-se de uma ferramenta de suporte automático que disponibiliza seus resultados para uma outra ferramenta qualquer de interoperabilidade semântica, não necessariamente automática, que fará uso destes resultados.

A abordagem apresenta três módulos na sua arquitetura, conforme ilustrado na Figura 34, a saber [SOUZA JUNIOR, H. C. 2008 (apud DOAN ET AL., 2004)]:

- i. Distribution Estimator: tem como entrada duas taxonomias juntamente com suas instâncias. Este aplica técnicas de aprendizagem para computar cada

DOAN, A., MADHAVAN, J., DOMINGOS, P., HALEVY, A. Ontology Matching: A Machine Learning Approach. In S. Staab and R. Studer, editors, Handbook on Ontologies in Information Systems. Springer-Verlag, 2004.

- par de conceitos com sua distribuição de probabilidade. Utiliza um conjunto de aprendizes e meta-aprendizes;
- ii. Similarity Estimator: aplica funções de similaridade para computar o valor da similaridade de cada par de conceitos. A saída deste módulo é a matriz de similaridade entre os conceitos das duas taxonomias; e
 - iii. Relaxation Labeler: usa a matriz de similaridades e, junto com as restrições de domínio, realiza o mapeamento que melhor satisfaz a estas restrições e as de conhecimento comum. Esta configuração de mapeamento é a saída do GLUE.

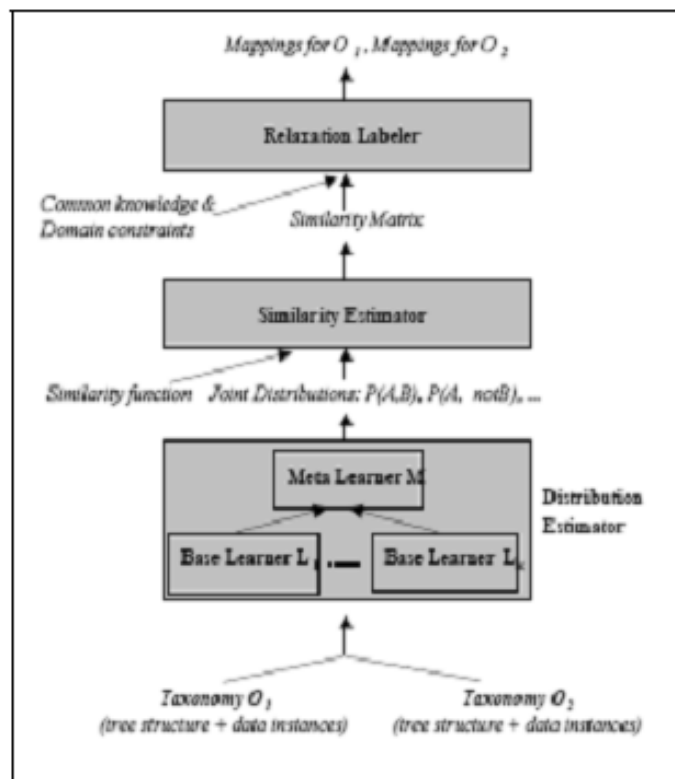


Figura 34. Arquitetura GLUE [SHVAIKO, P., EUZENAT, J. 2007 (apud DOAN *et al.*, 2004)].

4.4.6 CATO

CATO - Componente para Alinhamento Taxonômico de Ontologias [FELICÍSSIMO, C H. 2004] é uma estratégia para alinhamento taxonômico de ontologias. A abordagem tem como entradas pares de ontologias. Para realizar o alinhamento de mais de duas ontologias, deve-se fazer passos seqüenciais, sempre alinhando as

ontologias duas a duas. Conforme ilustrado na Figura 35, o alinhamento é obtido em três etapas executadas seqüencialmente.

A primeira etapa da estratégia faz uma comparação lexical entre os conceitos das ontologias de entrada e utiliza mecanismo de poda estrutural dos conceitos associados como condição de parada. Como resultado, são fornecidas as ontologias de entradas enriquecidas com os alinhamentos conseguidos na etapa. Posteriormente, as ontologias são transformadas em XML, sendo representada apenas suas hierarquias.

A segunda etapa da estratégia compara estruturalmente as hierarquias das ontologias resultantes da primeira, identificando as similaridades entre suas sub-árvores comuns. Os conceitos destas sub-árvores são classificados como similares.

E na terceira etapa da estratégia ocorre um refinamento dos resultados da etapa anterior classificando conceitos identificados como **similares**, **bem similares** ou **pouco similares**, de acordo com um percentual de similaridade pré-fixado. Os resultados desta etapa são os conceitos classificados como bem similares.

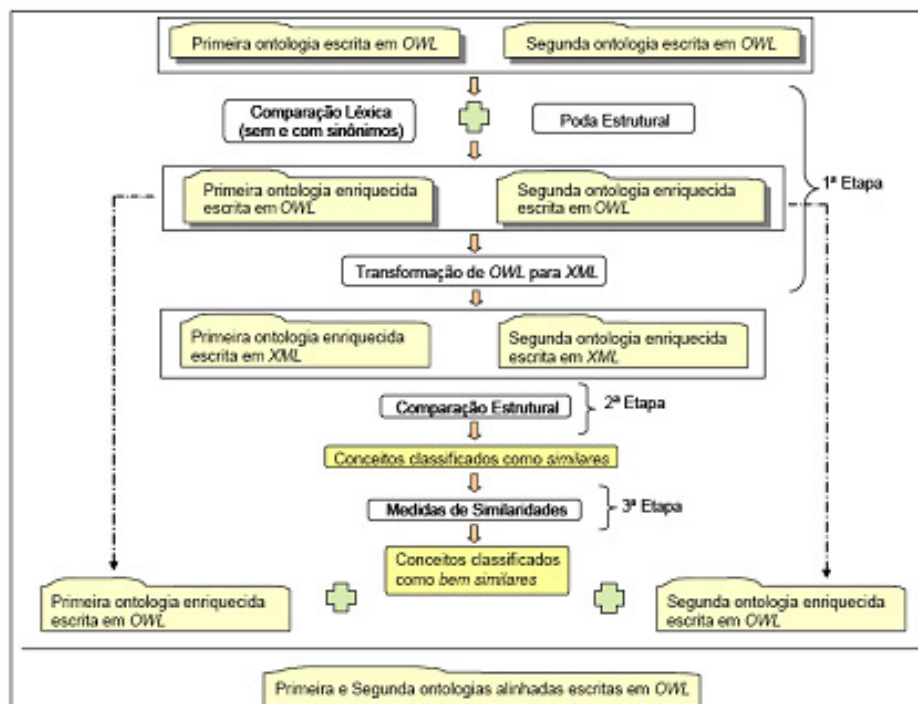


Figura 35. Estratégia para alinhamento taxonômico de ontologias [FELICÍSSIMO, C H. 2004].

Ao final da execução das três etapas da estratégia citadas acima, as informações de equivalência de conceitos bem similares são adicionadas nas

ontologias resultantes da primeira etapa. Após esta adição, as ontologias alinhadas são unidas em uma única ontologia, sendo esta o resultado final da estratégia.

Apesar de essa estratégia resultar em uma única ontologia, e, portanto ferir o conceito de alinhamento descrito acima. A autora diz que isso foi uma decisão de implementação já que as ontologias alinhadas continuam sendo reconhecidas pela identificação de seus *namespaces* e ainda existe a ligação entre os conceitos equivalentes na ontologia única, permitindo a reutilização e o compartilhamento de suas informações comuns, características do mecanismo de alinhamento de ontologias.

4.4.7 Análise das Tecnologias de Alinhamento

Nas seções acima foram apresentadas tecnologias existentes que promovem o alinhamento de ontologias, técnica de correspondência ontológica que será adotada no estudo de caso dessa dissertação. Cada uma dessas tecnologias possui as suas especificidades e adotam diferentes estratégias de alinhamento. É evidente também o uso, em todas elas, de medidas de similaridade. A seguir é feita uma análise das seis tecnologias abordadas nesta dissertação, com o objetivo de identificar características comuns.

QOM e FOAM são duas tecnologias que tem como base a NOM, podendo ser considerada como uma evolução da mesma. Todas elas são divididas nas mesmas seis etapas. A principal diferença entre essas três tecnologias se concentra no modo como os termos das ontologias de entrada são comparados. Diferentemente das outras tecnologias, as ontologias de entrada dessas tecnologias devem estar representadas em triplas RDF. Em NOM todas as entidades da primeira ontologia são comparadas com todas as entidades da segunda. Já QOM, considera que o número de pares candidatos pode gerar um problema de complexidade, e portanto, faz uso de heurísticas para diminuir o número de candidatos. FOAM, por sua vez, escolhe duas entidades, uma de cada ontologia de entrada, para realizar a comparação. Ambas as tecnologias fazem o cálculo de similaridade entre as entidades utilizando funções de similaridade baseadas nas características das ontologias investigadas.

PROMPT, constituída por quatro ferramentas, suporta não apenas o alinhamento, mas também a combinação, identificação da diferença entre duas versões de uma ontologia e a extração de parte de uma ontologia. A AnchorPROMPT, que permite a realização do alinhamento, a partir das ontologias de entrada desenvolvidas na linguagem OWL, fornece pares de termos relacionados. Além da geração desses pares de forma automática, é permitido ao usuário indicar outros pares de termos relacionados. Como medida de similaridade, é adotado a relação estrutural dos termos, medidas de similaridade pré-definidas e grupos de equivalência.

Outra tecnologia abordada é o sistema GLUE. Um sistema que permite descobrir mapeamentos semânticos não entre duas ontologias mas entre duas taxonomias, para isso são utilizadas técnicas de aprendizado de máquina semi-automática. Assim como no PROMPT, para encontrar a similaridade entre as taxonomias, o sistema faz uso de uma medida pré-definida.

Finalmente, a abordagem CATO permite o alinhamento taxonômico entre duas ontologias OWL. Após a comparação entre os conceitos, os mesmo são enriquecidos através de um vocabulário pré-definido. A ferramenta também utiliza medidas de similaridade comparando a estrutura hierárquica das ontologias, a partir da qual os conceitos são identificados como muito ou pouco similares. E ao contrário do conceito de alinhamento, CATO resulta em uma única ontologia, entretanto, as ontologias alinhadas permanecem sendo reconhecidas pelos *namespaces* e ainda é possível reutilizar e compartilhar as informações através da ligação entre os conceitos equivalentes existentes na ontologia única.

5 QDAontology – UMA ABORDAGEM PARA O DESENVOLVIMENTO DE ONTOLOGIAS EM E-SCIENCE

A utilização de ontologias em sistemas computacionais tem se tornando cada vez mais importante e difundida. Entretanto, por ser uma área em constante evolução, não existem modelos de processo de Engenharia Ontológica (EO) consolidados para a construção de uma ontologia. O uso de um modelo de processo no desenvolvimento de uma ontologia torna esta tarefa mais impessoal, menos complexa e mais sistemática. A existência de um processo, independentemente de quem realiza este trabalho e de quais os objetivos se deseja alcançar, permite um elevado grau de precisão e controle sobre o desenvolvimento ontológico a partir da realização de cada atividade e dos artefatos gerados. E assim, como na área de Engenharia de Software (ES) [PFLEEGER, S. L. 2004], nota-se que o uso de processos bem definidos pode ter como conseqüência o aumento da qualidade do produto final e a padronização das tarefas e produtos gerados.

Originalmente, a ontologia CeIO [MATOS, E. E. S. 2008] não seguiu um modelo sistematizado de desenvolvimento para ontologias, mas, diante da atual necessidade de sua expansão, a especificação de uma abordagem que permita a sua reengenharia e sua evolução se torna indispensável.

Baseado na metodologia METHONTOLGY [FERNÁNDEZ, M. *et al.* 1997], em outros modelos descritos nos capítulos anteriores e na experiência do Núcleo de Pesquisa em Qualidade de Software da UFJF (NPQS), a abordagem definida neste trabalho foi elaborada, inicialmente, para fazer a reengenharia da ontologia CeIO e também permitir a sua evolução. Pretende-se, entretanto, que esta abordagem seja usada em outros projetos de desenvolvimento de ontologias para e-Science. Nestas aplicações, as ontologias, em geral, se caracterizam por serem desenvolvidas por equipes multidisciplinares, onde os conteudistas pertencem à área do domínio da aplicação e os construtores são engenheiros ontológicos oriundos da Ciência da Computação.

Ressalta-se ainda que a METHONTOLGY foi a metodologia escolhida como base deste trabalho por ser uma das mais completas e abrangentes metodologias estudadas, já que engloba a maioria dos aspectos do processo de desenvolvimento de software padronizado pelo IEEE. Ela é baseada na idéia de prototipação e

evolução do ciclo de vida da ontologia e favorece o reuso de ontologias existentes [SOUZA, J. F. 2007].

A abordagem QDAontology - Quality Driven Approach for e-Science Ontologies, proposta nesta dissertação para a construção de ontologias no domínio da e-Science, é composta por etapas, atividades, participantes, artefatos e características de qualidade. Ressalta-se que o processo de desenvolvimento adotado é centrado em um modelo evolutivo e, portanto, os ciclos podem se repetir a cada evolução da ontologia.

Neste capítulo é apresentada uma visão geral da abordagem de engenharia ontológica proposta, detalhando as etapas do processo e suas atividades, os participantes envolvidos e os artefatos gerados nas etapas e atividades. As características de qualidade essenciais para uma ontologia são também destacadas.

5.1 QDAontology - QUALITY DRIVEN APPROACH FOR E-SCIENCE ONTOLOGIES

A fim de facilitar o processo de engenharia ontológica, a abordagem QDAontology fornece informações detalhadas a seus usuários e a Figura 36 representa o modelo proposto com todos os seus componentes. Para cada etapa o modelo descreve (i) participantes, (ii) atividades, (iii) artefatos (iv) características de qualidade. Cada etapa é constituída de atividades, em cada atividade são gerados artefatos e os participantes estão relacionados com as etapas e atividades do processo. À partir da evolução dos artefatos ocorre o desenvolvimento da ontologia. Com a implementação da ontologia a mesma pode ser integrada a outras ontologias, através de mecanismos como alinhamento ou mapeamento, e, com base nos resultados obtidos, a ontologia pode sofrer uma evolução e produzir o resultado desejado.

A seguir apresentamos uma descrição detalhada dos componentes da abordagem proposta que é constituída por seis etapas: Especificação, Conceitualização, Formalização, Implementação, Integração e Evolução. Cada ciclo das etapas termina com uma versão da ontologia. As etapas, por sua vez, são compostas por atividades específicas, das quais são extraídos os artefatos gerados,



Figura 36. Componentes da abordagem QDAontology.

como marcos do processo de desenvolvimento. As atividades são definidas de forma ortogonal às etapas, podendo estar presentes em diversas etapas de um mesmo ciclo. Os artefatos são responsáveis por toda a documentação da ontologia e a sua existência garante uma qualidade maior ao processo de desenvolvimento. Além disso, uma descrição dos participantes envolvidos e suas atribuições nas diferentes etapas e atividades é apresentada. As características de qualidade necessárias às ontologias são descritas, de forma que as mesmas permeiem o processo de desenvolvimento e garantam que a versão final da ontologia tenha os atributos desejados.

5.1.1 Descrição das Etapas

ETAPA 1 - ESPECIFICAÇÃO

A etapa de Especificação identifica a motivação pela qual a ontologia está sendo construída e quais são os objetivos de seu uso, e especifica os usuários finais. Tem como proposta elaborar um documento que abrange a principal finalidade da ontologia e também identifica o conjunto de termos a serem representados, as características e as granularidades. Segundo [FERNÁNDEZ-LÓPEZ, M. *et al.* 1999] a especificação de uma ontologia deve ser a mais completa e concisa possível.

Nesta etapa deve ser especificado também um conjunto de perguntas de competência para delimitar o escopo e o propósito do domínio.

Dois participantes principais atuam nesta etapa: o engenheiro ontológico, que analisa a existência de ontologias de domínios similares e/ou complementares e implementa novas ontologias se necessário, e o *stakeholder*, que nessa etapa tem um papel fundamental fornecendo todo o seu conhecimento do domínio.

A Especificação conta com algumas atividades essenciais para seu bom desempenho: aquisição do conhecimento, avaliação, documentação, gerência de configuração, planejamento, qualidade e ambiente.

O artefato gerado na especificação é um documento contendo informações básicas sobre o domínio que está sendo modelado.

O Quadro 9 indica atividades, participantes, artefatos e característica de qualidade presentes na etapa de Especificação.

Quadro 9. Etapa de Especificação do QDAontology.

Especificação	ATIVIDADES	ARTEFATOS	CARACTERÍSTICAS DE QUALIDADE
	Aquisição do conhecimento Avaliação Documentação Gerência de configuração Planejamento Qualidade Ambiente	Doc. De requisitos	Clareza dos termos Unicidade dos termos Compromisso ontológico Relação com a realidade
PARTICIPANTES - Engenheiro ontológico e <i>Stakeholder</i>			

A representação da etapa de Especificação na abordagem QDAontology encontra-se na Figura 37. Essa representação indica que a mesma é evolutiva, podendo se repetir sempre que necessário, além de exibir seus participantes, as características de qualidade fundamentais para o bom desempenho das atividades e o artefato gerado ao final da etapa.

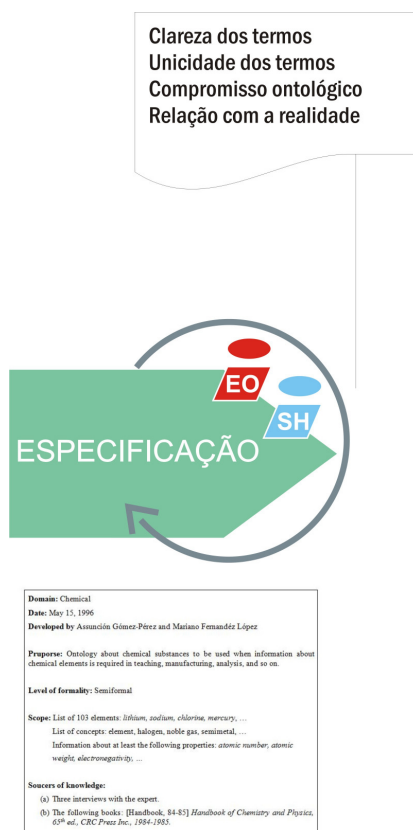


Figura 37. Representação da etapa de Especificação na abordagem QDAontology.

ETAPA 2 - CONCEITUALIZAÇÃO

A atividade de Conceitualização organiza e estrutura o conhecimento do domínio como modelos significativos no nível de conhecimento, usando representações externas, que são independentes da linguagem de implementação e do ambiente de desenvolvimento. Dessa forma, permite o entendimento da estrutura e da dinâmica da organização da ontologia, garantindo que todos os participantes tenham um entendimento comum da mesma e possam derivar os requisitos para o seu desenvolvimento.

Como participantes tem-se o Engenheiro Ontológico e o *Stakeholder*.

Esta etapa apresenta também atividades associadas, que são: Aquisição de Conhecimento, Avaliação, Documentação, Gerência de Configuração, Planejamento, Qualidade e Ambiente.

Como artefatos gerados nesta etapa, são produzidos documentos que permitem a elaboração de uma representação intermediária para a ontologia que se deseja desenvolver. Para tanto, podem ser construídos glossários dos termos e um dicionário contendo todos os conceitos do domínio, instâncias de classes e atributos, além disso, conceitos sinônimos e acrônimos podem ser adicionados. O Quadro 10 abaixo indica atividades, participantes, artefatos e característica de qualidade presentes na etapa de Conceitualização.

Quadro 10. Etapa de Conceitualização do QDAontology.

Conceitualização	ATIVIDADES	ARTEFATOS	CARACTERÍSTICAS DE QUALIDADE
	Aquisição do conhecimento Avaliação Documentação Gerência de configuração Planejamento Qualidade Ambiente	Glossário de termos Dicionário de conceitos	Coerência Corretude Independência da codificação Adequação semântica
PARTICIPANTES - Engenheiro ontológico e <i>Stakeholder</i>			

Uma representação para a etapa de Conceitualização da abordagem QDAontology pode ser visualizado na Figura 38. Assim como na Especificação, essa etapa também é evolutiva. A representação mostra seus participantes, as características de qualidade essenciais e os artefatos gerados na conclusão da etapa.

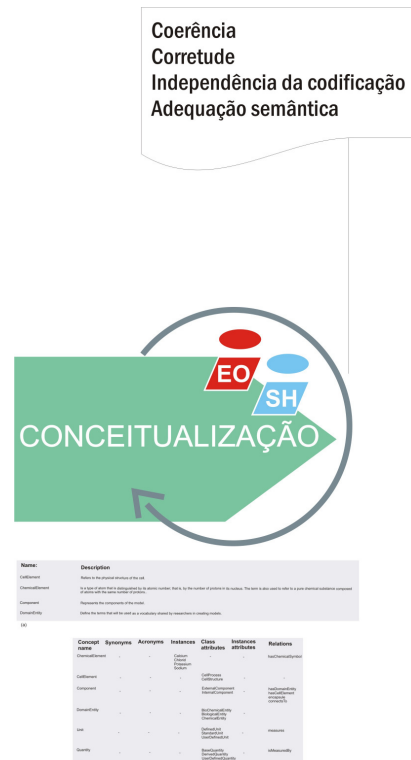


Figura 38. Representação da etapa de Conceitualização na abordagem QDAontology.

ETAPA 3 - FORMALIZAÇÃO

A atividade de Formalização transforma o modelo conceitual em um modelo formal ou semi-computável, o que possibilita a transformação dos requisitos e a definição dos termos em um projeto de ontologia, sugerindo uma arquitetura e adaptando o projeto para o ambiente de implementação

Nesta etapa as atividades contam com o apoio do desenvolvedor. Além disso, o engenheiro ontológico e o *stakeholder*, também participam ativamente das atividades.

Os artefatos gerados nesta etapa são a árvore de classificação dos conceitos e o diagrama de relacionamentos, que permitem elaborar uma representação intermediária do domínio. Nesses diagramas são exibidas as relações e conseqüentemente são definidas as regras do tipo “é uma”, “é um tipo de”, “é parte de” ou outras que melhor representem o domínio.

As atividades que compõe a etapa são: Aquisição de conhecimento, Avaliação, Documentação, Gerência de Configuração, Qualidade e Ambiente.

O Quadro 11 abaixo indica atividades, participantes, artefatos e característica de qualidade presentes na etapa de Formalização.

Quadro 11. Etapa de Formalização do QDAontology.

Formalização	ATIVIDADES	ARTEFATOS	CARACTERÍSTICAS DE QUALIDADE
	Aquisição do conhecimento Avaliação Documentação Gerência de configuração Qualidade Ambiente	Diagrama de relacionamentos Árvore de classificação	Organização Modularidade Estrutura invariante Independência da codificação Adequação semântica Logicamente correta Restrições
PARTICIPANTES - Engenheiro ontológico e <i>Stakeholder</i>			

A Figura 39 apresenta a representação da etapa de Formalização da abordagem QDAontology. Sua representação indica os participantes envolvidos, as características de qualidade e os artefatos produzidos durante a realização das atividades da etapa.



Figura 39. Representação da etapa de Formalização na abordagem QDAontology.

ETAPA 4 - IMPLEMENTAÇÃO

A Implementação constrói modelos computáveis utilizando ferramentas apropriadas para o desenvolvimento de ontologias. Define a organização formal dos termos,

conceitos e relacionamentos de forma a permitir inferência automatizada, através da definição de regras e restrições. Valida os subdomínios desenvolvidos como partes da ontologia e os integra. O resultado desta etapa é a ontologia codificada em uma linguagem formal, ou seja, são produzidas e disponibilizadas versões do produto para uso em diferentes aplicações.

O desenvolvedor e o engenheiro ontológico participam ativamente da etapa de implementação. O desenvolvedor como o responsável pela transformação do conhecimento adquirido com o *stakeholder* para uma linguagem de ontologias e o engenheiro ontológico responsável por todo o gerenciamento do desenvolvimento.

Fazem parte da etapa as seguintes atividades: Avaliação, Documentação, Gerência de Configuração, Qualidade e Ambiente.

Nesta etapa, o artefato gerado é a própria ontologia já codificada em uma linguagem formal.

O Quadro 12 abaixo indica atividades, participantes, artefatos e característica de qualidade presentes na etapa de Implementação.

Quadro 12. Etapa de Implementação do QDAontology.

Implementação	ATIVIDADES	ARTEFATOS	CARACTERÍSTICAS DE QUALIDADE
	Avaliação Documentação Gerência de configuração Qualidade Ambiente	Codificação da ontologia	Propriedades formais Adequação Semântica Logicamente correta Restrições Inferência automatizada
PARTICIPANTES - Engenheiro ontológico e Desenvolvedor			

A Figura 40 representa a etapa de Implementação da abordagem QDAontology. A mesma indica os participantes envolvidos, as características de qualidade e o artefato produzido ao final da etapa.



Figura 40. Representação da etapa de Implementação na abordagem QDAontology.

ETAPA 5 - INTEGRAÇÃO

A etapa de Integração considera a reutilização e o compartilhamento da ontologia desenvolvida com outras ontologias do mesmo domínio. A interoperabilidade entre ontologias é um fator essencial e a realização dessa etapa garante o compartilhamento e troca de informações entre aplicações. Técnicas de correspondência ontológica devem ser usadas, como mapeamento, alinhamento, combinação e integração para garantir a compatibilidade com outras definições existentes. É necessário inspecionar as meta-ontologias (conceitos usados para modelar a ontologia) para selecionar a que melhor se ajusta à conceitualização e tentar encontrar quais outras ontologias contém definições de termos coerentes com os da conceitualização e as que usam as definições mais apropriadas.

Com os termos alinhados obtém-se os termos correspondentes entre as ontologias e, através dessas correspondências é possível reusar as informações umas das outras. Esse processo de correspondência ontológica conta com ferramentas automatizadas ou semi-automatizadas que permitem fazer esse alinhamento ou mapeamento entre as ontologias.

Como nesta etapa há preocupação com a reutilização de partes das ontologias e com utilização de técnicas de correspondência entre elas, a integração conta com o suporte do engenheiro ontológico e do desenvolvedor.

As atividades pertencentes à integração são: Avaliação, Documentação, Gerência de Configuração, Qualidade e Ambiente.

O artefato produzido nesta etapa é um documento contendo as informações essenciais relacionadas à correspondência entre as ontologias.

O Quadro 13 abaixo indica atividades, participantes, artefatos e característica de qualidade presentes na etapa de Integração.

Quadro 13. Etapa de Integração do QDAontology.

Integração	ATIVIDADES	ARTEFATOS	CARACTERÍSTICAS DE QUALIDADE
	Avaliação Documentação Gerência de configuração Qualidade Ambiente	Documento de integração	Extensibilidade Interoperabilidade semântica Similaridade semântica Independência da codificação Inferência automatizada
PARTICIPANTES - Engenheiro ontológico e Desenvolvedor			

A representação da etapa de Integração na abordagem QDAontology encontra-se na Figura 41, onde é possível observar seus participantes, as características de qualidade essenciais para o bom desempenho das atividades e o artefato gerado ao final da etapa.

ETAPA 6 - EVOLUÇÃO

A atividade de Evolução tem por objetivo apoiar o enriquecimento do conhecimento em uma ontologia, acrescentando novos conceitos ou expandindo conceitos existentes com novos conhecimentos adquiridos.

Segundo [FERRARA, A. 2003] a evolução pode ainda fazer o uso de três diferentes políticas: referência, importação e junção. Na política de referência, o conceito de distribuição do conhecimento é enfatizado, onde cada ontologia armazena apenas o seu próprio conhecimento. Na importação, um novo conceito é importado de uma ontologia e uma relação de mapeamento pode ser definida entre os conceitos das ontologias, preservando dessa forma a ontologia original e garantindo o acesso eficiente aos conhecimentos fornecidos. E na política de junção um novo conceito é importado e, em seguida incorporado à ontologia. Neste caso,

um conjunto de novas relações semânticas e propriedades são inseridas na ontologia para a aquisição do novo conceito.



Figura 41. Representação da etapa de Integração na abordagem QDAontology.

Participam desta etapa o engenheiro ontológico e o *stakeholder*.

Esta etapa é apoiada pelas atividades de: Avaliação, Documentação, Gerência de Configuração, Planejamento, Qualidade e Ambiente.

Através do acréscimo ou expansão de conceitos que podem ocorrer nesta etapa, o artefato gerado na Evolução é uma ontologia evoluída. Para representar essa evolução pode-se utilizar representações gráficas que exibem a versão final da ontologia.

Ressalta-se ainda que, como a abordagem proposta apresenta um ciclo de vida evolutivo, a execução dessa etapa implica que o processo é reiniciado, voltando, portanto à primeira etapa da abordagem. Os desenvolvimentos subseqüentes serão o processo de adaptação da ontologia aos requisitos novos ou modificados.

O Quadro 14 abaixo indica atividades, participantes, artefatos e característica de qualidade presentes na etapa de Evolução.

Quadro 14. Etapa de Evolução do QDAontology.

Evolução	ATIVIDADES	ARTEFATOS	CARACTERÍSTICAS DE QUALIDADE
	Avaliação Documentação Gerência de configuração Qualidade Ambiente	Documento de evolução	Organização Modularidade Estrutura invariante Extensibilidade
PARTICIPANTES - Engenheiro ontológico e <i>Stakeholder</i>			

A representação da etapa de Evolução na abordagem QDAontology é apresentada na Figura 42. Essa representação mostra seus participantes, as características de qualidade para a etapa e o artefato gerado na conclusão desta etapa.

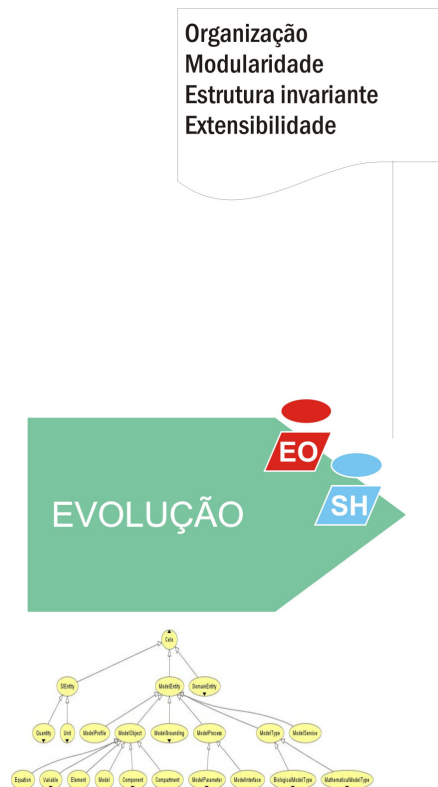


Figura 42. Representação da etapa de Evolução na abordagem QDAontology.

5.1.2 Descrição das Atividades

Aquisição de conhecimento: A Aquisição do conhecimento é uma atividade prioritária no desenvolvimento de uma ontologia, tendo em vista concentrar as ações

de definição do escopo, dos objetivos, dos usuários, dos termos e definições, relacionamentos e restrições. Pode ser considerada uma atividade em espiral, pois o conhecimento é detalhado em ciclos evolutivos, que permeiam as etapas de Especificação, Conceitualização e Formalização da ontologia. Esta evolução permite que os participantes tornem-se cada vez mais familiarizados com o domínio da ontologia.

De acordo com [FERNÁNDEZ-LÓPEZ, M. *et al.* 1999], a aquisição de conhecimento ocorre em três etapas: (1) encontros com pessoas que dominam a área que se deseja construir a ontologia para se ter uma noção geral e não detalhada do domínio; (2) estudo de uma documentação do domínio para o aprendizado do domínio; e (3) obtido algum conhecimento, inicia-se o ciclo de aquisição do conhecimento, começando a olhar para conhecimentos mais gerais e gradualmente em detalhes particular para configuração da ontologia.

Durante a atividade de aquisição de conhecimento pode-se utilizar de um conjunto de técnicas de forma a integrar esse conhecimento. Essas técnicas são: entrevistas não-estruturadas com especialistas do domínio para construir um rascunho preliminar do domínio; análise de textos informais para estudar os principais conceitos; análise de textos formais; entrevistas estruturadas com especialistas do domínio para gerar conhecimento detalhado e específico, propriedades, relacionamentos, entre outros; análise da tabela de domínio que é útil para averiguar valores de conceitos e para identificar regularidades de dados; análise gráfica do domínio; revisão detalhada de especialistas.

Avaliação: Avaliação é utilizada para realizar um julgamento técnico da ontologia, através da análise da documentação gerada ao longo do processo de engenharia ontológica. A avaliação engloba atividades de verificação e validação. A verificação faz referência ao processo técnico garantido a corretude da ontologia. A validação garante que a ontologia corresponde ao domínio que ela representa. Para isso podem ser adotados questionários de validação para verificar o domínio especificado na ontologia. Por sua vez, esses questionários devem ser respondidos por pessoas qualificadas na área de interesse da ontologia, o que permite validar as informações contidas na ontologia. Nesta atividade há uma grande preocupação em encontrar inconsistências, verificar a completude do domínio e possíveis redundâncias na ontologia. O engenheiro ontológico deve documentar a atividade de

avaliação, descrevendo as técnicas usadas, os erros encontrados e ainda as fontes de conhecimento utilizadas.

Documentação: Na Documentação podem ser produzidos uma série de artefatos, podendo ser textuais, visuais (diagramas e imagens) e até mesmo codificação. Por ser uma atividade extremamente importante, a documentação é feita em paralelo com as demais atividades mencionadas a fim de garantir um processo eficiente de controle de qualidade e melhoria das possibilidades de reutilização da ontologia [TEMPICH, C. 2006].

Gerência de Configuração: De acordo com [PRESSMAN, R. S. 2006], gerência de configuração é um conjunto de atividades elaboradas a fim de administrar modificações ao longo do ciclo de vida de um software. No contexto de ontologias, a gerência de configuração exerce uma função importante fornecendo apoio ao seu desenvolvimento já que mudanças podem ser consideradas ao longo de sua evolução. Atividade como controle de versões e mudanças são exemplos de tarefas pertencentes à gerência de configuração. É uma atividade essencial para controlar os diversos artefatos produzidos por pessoas que trabalham em um mesmo projeto. O controle garante que os artefatos resultantes não são conflitantes.

Planejamento: A atividade de Planejamento inclui um conjunto de práticas gerenciais e técnicas que permite aos desenvolvedores a definição de um roteiro enquanto as metas vão sendo concluídas [PRESSMAN, R. S. 2006]. Entender a razão pela qual a ontologia está sendo construída e o domínio ao qual ela pertence são tarefas importantes para o planejamento do desenvolvimento da ontologia. O planejamento [PFLEEGER, S. L. 2004] também contribui para relacionar os participantes presentes no processo de desenvolvimento, como eles estão organizados e qual a sua função dentro do processo.

Qualidade: A engenharia de software tem como meta produzir software de alta qualidade. No desenvolvimento de ontologias isso não é diferente. Com o objetivo de se criar ontologias que sejam as mais concisas possível, a qualidade abrange atividades específicas de garantia e controle de qualidade, uso de ferramentas e métodos, controle sobre todos os processos, produtos e modificações, além de

garantir a satisfação do escopo da ontologia. Dessa forma, a atividade de qualidade visa avaliar a realização de todas as tarefas constituintes do processo garantindo assim que os produtos resultantes das atividades sejam satisfatórios. A atividade também visa a verificação contínua da qualidade, e pode estar focada tanto na qualidade de produto quanto na qualidade do processo de desenvolvimento de ontologias. Dessa forma um conjunto de características e subcaracterísticas de qualidade de ontologias foram definidas de acordo com o que consideramos essencial para o desenvolvimento de ontologias científicas. Essas características devem estar presentes nos diferentes artefatos gerados ao longo do processo de engenharia ontológica de forma a garantir um produto final de alta qualidade.

Ambiente: Provê a organização do ambiente de desenvolvimento da ontologia incluindo os processos e as ferramentas a serem utilizados.

5.1.3 Participantes do Processo

A abordagem QDAontology conta com o apoio de três participantes fundamentais: o engenheiro ontológico, o *stakeholder* e o desenvolvedor. A seguir apresentamos uma breve descrição de cada um desses participantes:

Engenheiro Ontológico : tem como função inicial analisar se as ontologias existentes satisfazem as necessidades do *stakeholder* ou se a solução é implementar o conhecimento adquirido pelo *stakeholder* como uma nova ontologia [JANOWICZ, K. *et al.* 2008]. Além disso, o engenheiro ontológico toma decisões sobre como representar os termos, os conceitos e as relações, dentro de uma ordem lógica, levando a um modelo de nível ontológico cada vez mais detalhado e computacionalmente implementável.

Stakeholder: também conhecido como o especialista de domínio, é a pessoa que detém o conhecimento específico ou habilidade em uma determinada área. É a pessoa responsável pela construção da base de conhecimento, ou seja, contribui

com o conhecimento a ser implementado na ontologia. Na abordagem o termo *stakeholder* pode significar um grupo de especialista do conhecimento.

Desenvolvedor: no contexto de ontologias, é o responsável por construir a ontologia, ou seja, transformar todo o conhecimento extraído em um modelo entendível por máquina. Portanto, constrói a ontologia em uma linguagem formal própria, implementando o que foi especificado.

5.1.4 Artefatos de documentação

A cada nova etapa ou atividade da abordagem são gerados artefatos que objetivam documentar todo o processo de desenvolvimento da ontologia. Os principais artefatos da proposta são:

Documento de requisitos: o documento gerado na etapa de Especificação deve conter informações essenciais para o entendimento geral do domínio, além disso, informações sobre quem elaborou o documento, bem como as fontes de conhecimento utilizadas devem ser mencionadas como forma de garantir a proveniência das informações. Um modelo de documento da especificação de uma ontologia pode ser visualizado na Figura 43 [FERNÁNDEZ-LÓPEZ, M. *et al.* 1999].

Glossário de termos: na etapa de Conceitualização, há organização e conversão da visão do domínio em uma especificação semi-informal, para isso pode ser adotado um conjunto de representações intermediárias em que as pessoas que possuem o conhecimento do domínio e as que desenvolvem a ontologia possam entender. Constrói-se primeiro um glossário de termos do domínio da ontologia que inclui todos os termos como conceitos, instâncias, atributos, verbos [FERNÁNDEZ-LÓPEZ, M. *et al.* 1999]. A Figura 44 exhibe o modelo deste documento.

<p>Domain: Chemical</p> <p>Date: May 15, 1996</p> <p>Developed by Assunción Gómez-Pérez and Mariano Fernández López</p> <p>Pruporse: Ontology about chemical substances to be used when information about chemical elements is required in teaching, manufacturing, analysis, and so on.</p> <p>Level of formality: Semiformal</p> <p>Scope: List of 103 elements: <i>lithium, sodium, chlorine, mercury, ...</i> List of concepts: element, halogen, noble gas, semimetal, ... Information about at least the following properties: <i>atomic number, atomic weight, electronegativity, ...</i></p> <p>Soucers of knowledge:</p> <p>(a) Three interviews with the expert.</p> <p>(b) The following books: [Handbook, 84-85] <i>Handbook of Chemistry and Physics, 65th ed., CRC Press Inc., 1984-1985.</i></p>

Figura 43. Modelo de documento de requisitos para a ontologia *Chemicals* [adaptado de FERNÁNDEZ-LÓPEZ, M. *et al.* 1999].

Name:	Description
Atomic weight	The relative mass of an atom of an element against the mass of the twelfth part of the carbon-12 isotope.
Crystalline structure	A 3D skeleton containing the set of points or ionic positions of a crystal that have an identical environment.
Element	A substance made up of atoms with the same number of points.

Figura 44. Modelo de documento do glossário de termos [adaptado de FERNÁNDEZ-LÓPEZ, M. *et al.* 1999].

Dicionário de conceitos: outro recurso que pode ser adotado para documentar a Conceitualização é o dicionário de conceitos, que contém todos os conceitos do domínio, instâncias, classes e atributos das instâncias, e opcionalmente, conceitos sinônimos e acrônimos [FERNÁNDEZ-LÓPEZ, M. *et al.* 1999]. A Figura 45 exhibe um modelo de representação do dicionário de conceitos.

Concept name	Synonyms	Acronyms	Instances	Class attributes	Instances attributes	Relations
Element	-	Elm.	-	-	Atomic number Atomic weight Chemical group Chemical period Density at 20°C Electronegativity Melting point ...	Has structure
Third transition series	Sixth-period transition series	3TS	Gold Hafnium Mercury Osmium Iridium Platinum Rhenium Tantalum Wolfram	-	-	-

Figura 45. Modelo de documento do dicionário de conceitos [adaptado de FERNÁNDEZ-LÓPEZ, M. *et al.* 1999].

Diagrama de relacionamentos: a partir do glossário de termos e do dicionário de conceitos podemos construir os diagramas de relacionamentos binários entre os conceitos classificados [FERNÁNDEZ-LÓPEZ, M. *et al.* 1999]. A Figura 46 exibe um modelo de representação do diagrama de relacionamentos.

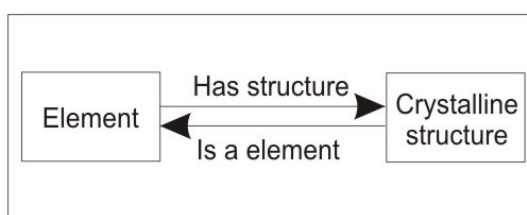


Figura 46. Modelo do diagrama de relacionamentos [adaptado de FERNÁNDEZ-LÓPEZ, M. *et al.* 1999].

Árvore de classificação: outro documento que pode ser gerado na etapa de Conceitualização é a árvore de classificação dos conceitos usando relações como por exemplo *subclass-of*, a partir dos termos do glossário [FERNÁNDEZ-LÓPEZ, M. *et al.* 1999]. A Figura 47 exibe a árvore de classificação obtida através do glossário de termos.

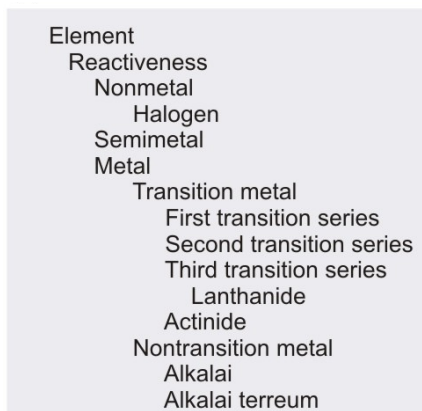


Figura 47. Árvore de classificação dos conceitos [adaptado de FERNÁNDEZ-LÓPEZ, M. *et al.* 1999].

Codificação da ontologia: o artefato gerado é a ontologia já codificada na linguagem específica utilizando-se um editor de ontologias. Esse ambiente permite uma maior facilidade no desenvolvimento da ontologia bem como a geração de sua codificação. A Figura 48, apresenta a geração do código de uma ontologia utilizando um editor de ontologias.

```

</owl:Class>
<owl:Class rdf:ID="xylem_element">
  <rdfs:subClassOf rdf:resource="#axial_cell"/>
  <rdfs:subClassOf rdf:resource="#transporting_cell"/>
</owl:Class>
<owl:Class rdf:ID="xylem_fiber_cell">
  <rdfs:subClassOf rdf:resource="#structural_cell"/>
  <rdfs:subClassOf rdf:resource="#xylem_element"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    >A fiber of the xylem tissue, Two types are recognized in the secondary xylem: fiber tracheids
  </rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="yolk_cell">
  <rdfs:subClassOf rdf:resource="#stuff_accumulating_cell"/>
  <rdfs:subClassOf rdf:resource="#animal_cell"/>
</owl:Class>
<owl:Class rdf:ID="zoospore">
  <rdfs:subClassOf rdf:resource="#sporangiospore"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    >A sporangiospore that is motile, having flagella.</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="zygospore">
  <rdfs:subClassOf rdf:resource="#spore"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    >A resting spore resulting from the conjugation of isogametes or from the fusion of like gametes
  </rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="zygote">
  <rdfs:subClassOf rdf:resource="#parenchymal_cell"/>
  <rdfs:subClassOf rdf:resource="#plant_cell"/>
  <rdfs:subClassOf rdf:resource="#cell_by_lineage"/>
  <rdfs:subClassOf rdf:resource="#animal_cell"/>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    >Diploid cell produced by the fusion of sperm cell nucleus and egg cell.</rdfs:comment>
</owl:Class>
</rdf:RDF>
  
```

Figura 48. Documento da implementação. Código gerado através do ambiente gráfico para edição de ontologias.

Documento de Integração: Na Integração, [FERNÁNDEZ-LÓPEZ, M. *et al.* 1999], como resultado desta atividade propõe-se elaborar um documento contendo informações como: qual(is) será(ão) a(s) ontologia(s) utilizada(s) e para cada termo utilizado indicar o nome do termo no modelo conceitual, o nome da ontologia de onde foi extraído o termo, e, finalmente, a definição e os argumentos da ontologia. A Figura 49 apresenta uma representação deste documento integrado.

Meta-Ontologia	Ontologias	
Termo em nossa conceitualização	Ontologia utilizada	Nome do termo na ontologia
Kilometer	Standard-Units em Ontololingua	Kilometer
Centimeter	Standard-Units em Ontololingua	Undefined
Exponent	KIF-Numbers em Ontololingua	Expt

Figura 49. Exemplo de documento de integração [adaptado de FERNÁNDEZ-LÓPEZ, M. *et al.* 1999].

Documento de evolução: na Evolução pode acontecer tanto o enriquecimento quanto a expansão do domínio. Pelo fato da abordagem apresentar um ciclo evolutivo, neste momento o processo é reiniciado tendo como ponto inicial o que se deseja na evolução da ontologia. Como as etapas serão executadas novamente, todos os artefatos serão produzidos e conseqüentemente, ao final será produzida uma ontologia modificada ou expandida. Para representar essa alteração, exibimos graficamente a versão final da ontologia. A Figura 50 exibe uma possibilidade dessa representação, onde são apresentadas as classes e subclasses de uma ontologia.

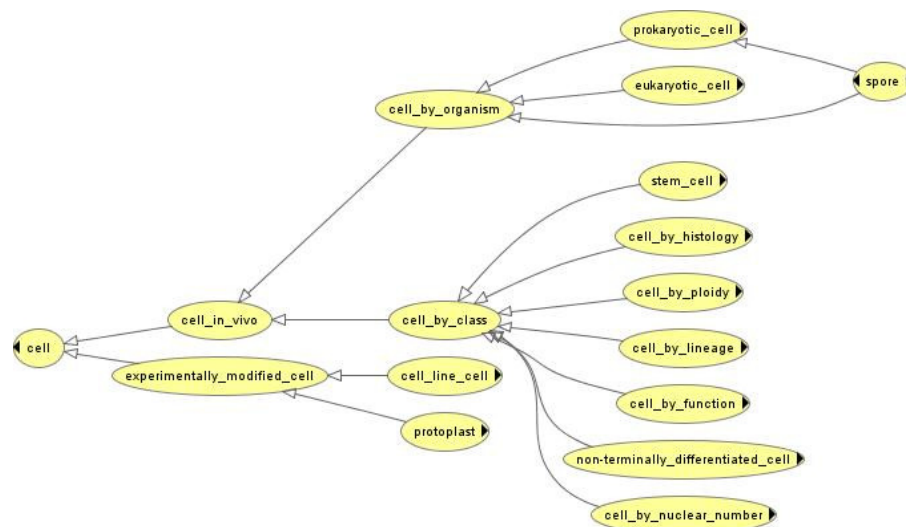


Figura 50. Exemplo de representação gráfica de ontologias [CELL TYPE 2010].

5.1.5 Características de Qualidade

Qualidade de ontologias é ainda um tema novo e na área de e-Science são necessários estudos que descrevam não só essas características, mas, também, suas aplicações no processo de engenharia ontológica e nas ontologias como produtos. Autores como [SURE *et al.* 2002] distinguem três tipos de avaliação para as ontologias: focada na tecnologia, focada no usuário e focada na ontologia. Essa dissertação trabalha com qualidade da ontologia.

Segundo [PFLEEGER 2004] os profissionais devem ser capazes de avaliar os seus produtos e o modo como os mesmos são construídas e para tal existem diferentes técnicas de avaliação e é importante identificar a mais adequada para o que se quer saber. No caso da abordagem proposta nessa dissertação adotamos a categoria de análise de características para avaliação da ontologia. Utilizada para atribuir um valor e classificar os atributos de qualidade de um produto, essa técnica, apesar da subjetividade, é um primeiro passo para a definição de métricas e medidas para a adoção de modelos mais complexos para a avaliação de ontologias.

O desenvolvimento de uma ontologia produz um grande número de artefatos e para cada um deles podemos avaliar se este produto tem os atributos de qualidade desejáveis. Como esses artefatos são gerados ao longo do processo de desenvolvimento da ontologia podemos associar diferentes atributos de qualidade, representantes de diferentes etapas e atividades, que contribuem para a qualidade do produto final. Tomando por base a literatura estudada e a prática do NPQS em desenvolver ontologias, o Quadro 15 apresenta uma hierarquia de características e subcaracterísticas, em que cada uma delas contribui para a qualidade da ontologia como um todo.

Quadro 15. Características e Subcaracterísticas de Qualidade para Ontologias.

CARACTERÍSTICAS E SUBCARACTERÍSTICAS DE QUALIDADE PARA ONTOLOGIAS	
CARACTERÍSTICA	SUB-CARACTERÍSTICA
CONFIABILIDADE Característica da ontologia representar um determinado domínio o mais próximo possível da realidade. A confiabilidade inclui definições claras, objetivas, corretas e fiéis ao domínio	Clareza dos termos Significados claros dos termos e definições objetivas.
	Coerência Inferências consistentes com as definições.
	Corretude Definições abrangentes, corretas, ricas e fiéis ao domínio.
	Unicidade de termos

modelado.	<p>Identificação clara, objetiva e única de cada termo, sem ambigüidade.</p> <p>Compromisso ontológico Compartilhamento de conhecimento e atendimento mínimo das hipóteses sobre o mundo que está sendo modelado.</p> <p>Relação com a realidade Termos e definições relacionados com o mundo real.</p>
<p>MODELAGEM CONCEITUAL Característica da ontologia representar os conceitos de um domínio. A modelagem conceitual inclui identificação, análise, descrição e restrições dos elementos de um domínio do mundo real, a fim de que os mesmos sejam incorporados a sistema computacionais.</p>	<p>Logicamente correta Modelagem adequada para inferência e uso em sistemas computacionais, na recuperação e na troca de informações.</p> <p>Restrições Descrição de restrições para eliminar modelos não pretendidos e clarear conceitualizações.</p> <p>Propriedades formais Capacidade de ser representada por modelos formais, passíveis de processamento automático.</p> <p>Adequação Semântica Captura da semântica pretendida pelo usuário.</p>
<p>ESTRUTURA Característica da ontologia apresentar níveis hierárquicos. A estrutura inclui a organização visual, a modularização do domínio e a inclusão dos invariantes.</p>	<p>Organização Visualização estruturada e hierárquica de termos e relacionamentos.</p> <p>Modularidade Estruturação modular da ontologia em subdomínios, possibilitando a integração com outras ontologias do domínio.</p> <p>Estrutura invariante Captura e especificação dos termos e relacionamentos invariantes do domínio.</p>
<p>INTEROPERABILIDADE Característica da ontologia trocar informações com outras ontologias. Na interoperabilidade há o compartilhamento e a reusabilidade entre ontologias.</p>	<p>Interoperabilidade semântica Capacidade de compartilhamento e troca de informações entre diferentes aplicações.</p> <p>Independência de codificação Conceitualização especificada a nível de conhecimento, sem depender de um determinado tipo de codificação.</p> <p>Similaridade semântica Sobreposição de termos, conceitos e usuários de aplicações similares às definições encontradas em outras ontologias.</p> <p>Inferência automatizada Capacidade de permitir inferência automatizada, através dos modelos formais definidos.</p>
<p>MANUTENIBILIDADE Característica da ontologia ser modificada. A manutenibilidade inclui correção, melhorias ou inclusão de novos conceitos à ontologia.</p>	<p>Extensibilidade Ontologia projetada para o uso de vocabulário compartilhado e inclusão de novos conceitos.</p>

5.2 TRABALHOS RELACIONADOS

A busca por modelos de processo para a Engenharia Ontológica vem ocorrendo desde que a Ciência da Computação começou a trabalhar em conjunto com outras áreas, tais como, Ciência da Informação, Engenharias, Biologia, Química e Física. Essa multidisciplinaridade faz aumentar a necessidade por processos abrangendo todos os domínios envolvidos e garantindo que a troca de informações seja mais qualificada.

Ao longo deste trabalho foram descritas várias metodologias e métodos para o desenvolvimento de ontologias. Entretanto, nota-se que os mesmos, além de ter o seu processo associado a alguma atividade ou objetivos próprios, não relacionam o processo de desenvolvimento de ontologias com características de qualidade que são fundamentais para a garantia de um resultado satisfatório. Apesar dos mecanismos de correspondência ontológica serem muito difundidos, a maioria das metodologias não os explicita como uma etapa do processo de desenvolvimento. Quanto aos participantes, apesar de algumas metodologias definirem as suas participações, em geral elas não são associadas às atividades ou etapas do processo.

Na METHONTOLOGY [FERNÁNDEZ, M. *et al* 1997], apesar da existência de uma atividade relacionada à garantia de qualidade durante o processo de desenvolvimento de ontologias, não são definidos atributos de qualidade. Qualidade das ontologias tem custos, mas é um fator de requisitos essencial para algumas aplicações em e-Science. Na QDAontology, há preocupação com a qualidade da ontologia como um todo. Dessa forma, diversos atributos de qualidade foram definidos, incluindo características fundamentais em todas as etapas, da aquisição de conhecimento à documentação final. A METHONTOLOGY incluiu a etapa de integração como uma atividade, na nossa proposta, devido à sua importância para aplicações científicas a mesma constitui uma etapa da abordagem. Em ambas as propostas constam as atividades de Aquisição do conhecimento, Avaliação, Documentação e Gerência de configuração. Outro fator que difere a QDAontology da METHONTOLOGY são os participantes envolvidos, tendo em vista que, na nossa proposta, eles são explicitados no processo de desenvolvimento da ontologia. Em

comum, ambas apresentam flexibilidade quanto à adoção de diferentes documentos já que ao final de cada etapa são gerados os artefatos de documentação.

A metodologia ON-TO-KNOWLEDGE [SURE, Y. *et al.* 2002] apresenta um processo de desenvolvimento focado no conhecimento, além disso, os participantes envolvidos não são bem definidos. A integração com outras ontologias também não faz parte da metodologia. A QDAontology, define os papéis associados às etapas e trabalha com equipes multidisciplinares permitindo uma maior facilidade de consenso da ontologia final. Outro fator diferencial em nossa proposta é a reutilização do conhecimento, o que é proposto na etapa de integração. Observa-se ainda que a fase denominada de Meta-Processo de Conhecimento na ON-TO-KNOWLEDGE, se enquadra, em grande parte, nas duas primeiras etapas da QDAontology – Especificação e Conceitualização. A Avaliação que na ON-TO-KNOWLEDGE é uma etapa, na nossa abordagem é uma atividade que ocorre durante todo o processo e, portanto, garante que a ontologia produzida seja mais próxima do domínio que representa.

Por ser mais centrada, a metodologia DILIGENT [TEMPICH, C. 2006] torna a sua utilização mais restrita, diferentemente, a QDAontology, que apesar de ter sido elaborada para a criação de ontologias científicas, pode ser aplicada sem comprometimento à todos os domínios. Assim como na abordagem proposta, DILIGENT também evidencia quais são seus participantes e o papel que os mesmos representam no processo.

O processo KUP [ORLEAN 2003] em [SOUZA, J. F. 2007] é constituído por três etapas, entretanto, apesar da diferença do número de etapas, é uma metodologia bem próxima à QDAontology. Se compararmos as etapas do KUP e QDAontology, temos que, a etapa de Concepção engloba as etapas de Especificação e Conceitualização, tendo em vista que o foco está no levantamento de requisitos e elaboração da estrutura da ontologia e portanto, tratam da modelagem da ontologia. A etapa da Construção é semelhante às de Formalização e Implementação definidas na QDAontology, todas elas trabalham com a parte de codificação da ontologia. E por último, tem-se que a etapa de Evolução presente nas duas metodologias é equivalente pelo fato de levarem em consideração a inclusão de novos conceitos e relações à ontologia. Contudo, o processo KUP não insere entre as suas etapas a Integração, o que na abordagem proposta é uma etapa extremamente importante porque sugere a reutilização de outras ontologias.

6 APLICAÇÃO DA ABORDAGEM QDAontology: ESTUDO DE CASO EM BIOLOGIA

Esta seção apresenta a avaliação da abordagem QDAontology proposta nesta dissertação. Para isso será utilizada a ontologia CelO [MATOS, E. E. S 2008], cuja escolha se justifica não só por pertencer ao grupo de pesquisa NPQS ao qual esta pesquisa está vinculada, mas também pela existência de outras ontologias no domínio da área da biologia, como as do OBO Foundry [OBO FOUNDRY 2010], e da disponibilidade de especialistas na UFJF. A fim de tornar a aplicação da abordagem QDAontology mais dinâmica, dividimos o estudo de caso em dois ciclos – Reengenharia da CelO e Expansão da CelO para o domínio de doenças humanas, resultando na CelO-Human Disease. O primeiro com a finalidade de se fazer a reengenharia da CelO e gerar os artefatos para sua documentação e o segundo com o objetivo de expandir o domínio da CelO. Para a concretização dessa expansão, o estudo de caso foi apoiado pelo mecanismo de interoperabilidade entre ontologias denominado alinhamento

6.1 CICLO 1 – REENGENHARIA DA ONTOLOGIA CelO

Etapa 1: Especificação

Nesta etapa há a preocupação em identificar os objetivos para a construção da ontologia e, para tanto, o domínio da ontologia deve ser especificado, assim como a delimitação do escopo. No caso da CelO, o escopo é a eletrofisiologia cardíaca e sua 1ª versão teve como objetivo a representação de modelos biológicos com uma linguagem lógica e modelos descritos em CellML, além de permitir a criação de um repositório desses modelos e promover o reuso de componentes existentes.

A Especificação contou com o suporte de dois participantes: o engenheiro ontológico, no caso um estudante do Mestrado em Modelagem Computacional, e o *stakeholder*, um professor experiente no domínio de CellML. Entre as atividades presentes na etapa estão: aquisição do conhecimento, avaliação, documentação, gerência de configuração, planejamento, qualidade e ambiente.

Na aquisição do conhecimento, foram adotadas técnicas como, entrevistas com especialista do domínio e, principalmente, análise de textos formais e informais para estudar os principais conceitos do domínio. A avaliação foi feita através de validação com o stakeholder. As diferentes versões geradas foram armazenadas e o ambiente de software e hardware foi definido pelo engenheiro ontológico. O planejamento não seguiu planos rígidos de desenvolvimento quando do desenvolvimento da primeira versão.

Consideramos quatro características fundamentais à Especificação: clareza dos termos, unicidade dos termos, compromisso ontológico e relação com a realidade. Essas características se fizeram presentes através da geração de artefatos como marcos do processo de desenvolvimento.

O artefato gerado na etapa de especificação é exibido no Quadro 16. Este documento foi elaborado visando estabelecer os objetivos para o desenvolvimento da ontologia CeIO, bem como identificar os termos que a constituem.

Quadro 16. Especificação da ontologia CeIO.

Domínio	<i>Cardiac electrophysiology</i>
Data	2008
Desenvolvida por	Ely Edison da Silva Matos
Objetivo	<i>Provides representation of biological models with a logic language and models described in CellML, and it allows the creation of a model repository and promote the reuse of components.</i>
Nível de formalidade	<i>Semiformal</i>
Escopo	<i>List of 128 items: Buffer, Channel, Exchanger, Gate, Membrane, Mitochondria, Myoplasm, Nucleus, Pump, SarcoplasmicReticulum, DiadicSpace, ExtraCellularSpace, IntraCellularSpace, AtrialCell, AtrioVentricularNode, His-PurkinjeFibre, SinoAtrialNode, VentricularCell, ATP, Calcium, Chlorid, Potassium, Sodium, Aton, Ion, Molecule, HodgkinHuxleyModel, MarkovModel, MulticompartmentModel, NetworkModel, SynapticModel, ElectricCurrent, Length, LuminousIntensity, Mass, Substance, Temperature, Time, Acceleration, Área, Capacitance, Density, ElectricConductance, EletricInductance, EletricResistence, Frequency, Inductance, MagneticFlux, Velocity, Voltage, Volume, ADimensional, CapacitancePerArea, Concentration, ConcentrationPerTime, Conduitivity, ElectricChargePerAmount, ElectricChargePerConcentration, ElectricConductancePerArea, ElectricConductancePerCapacitance, ElectricCurrentPerCapacitance, ElectricCurrentPerConcentration, EletricCurrentDensity, MolarEntropy, PerConcentration, PerLength, PerTime, PerVoltage, PerVoltageTime, VoltagePerTime, dimensionless, ampere, candela, kelvin, kilogram, meter, metre, mole, second, cubicmeter, farad, hertz, litre, siemens, squaremeter, volt, weber, cm2, concentration_units, coulomb_per_millimole, coulomb_per_mole, faradays_constant_units, gas_constant_units, joule_per_kilomole_kelvin, joule_per_mole_kelvin, microA_per_cm2, microF, microF_per_cm2, micrometre, micrometre3, millimolar, millimolar_per_millisecond, milliS_per_cm2, millisecond, millivolt, mM, mm2, mM_per_ms, mm_per_ms, ms, mS_per_mm2, mV, mV_per_ms, nanoA, nanoA_per_millimolar, nanoS_per_picoF, per_concentration_units, per_millisecond, per_millivolt,</i>

	<p><i>per_millivolt_millisecond, per_mm, per_ms, per_mV, per_mV_ms, picoA, picoA_per_picoF, uA_per_mm2, uF_per_mm2.</i></p> <p><i>List of 80 concepts: Celo, DomainEntity, BioChemicalEntity, Protein, CellElement, CellProcess, GatingProcess, MembraneTransport, PolarizationProcess, CellStructure, CellPart, CellSpace, CellType, Myocyte, CardiacMyocyte, ChemicalEntity, ChemicalCompound, Nucleotide, ChemicalElement, ChemicalObject, ModelEntity, ModelGrounding, CellMLModel, ModelObject, Compartment, Component, ExternalComponent, InternalComponent, Element, Equation, Model, Variable, ComponentVariable, InterfaceVariable, LocalVariable, DomainVariable, ActionPotential, IonChannelGate, PotassiumIonChannel, SodiumIonChannel, ModelVariable, AlgebraicVariable, DependentVariable, IndependentVariable, ParameterVariable, ModelProcess, ModelInterface, ModelParameter, ModelParameterIn, ModelParameterOut, ModelProfile, ModelService, ModelType, BiologicalModelType, PhysiologicalModel, ElectrophysiologicalModel, MathematicalModelType, MathModelDeterministicType, DeterministicModel, StochasticModel, MathModelDiscretizationType, ContinuousModel, DiscreteModel, MathModelLinearityType, LinearModel, NonLinearModel, MathModelTimeDependencyType, StationaryModel, TimeDependentModel, SEntity, Quantity, BaseQuantity, DerivedQuantity, UserDefinedQuantity, Unit, DefinedUnit, StandardUnit, SIBaseUnit, SIDerivedUnit, UserDefinedUnit.</i></p> <p><i>List of 32 properties: hasDomainEntity, hasMeasure, hasVariableUnit, isContainedIn, contains, modelCompartment, hasCellElement, presents, refersTo, hasBiologicalEntity, hasVariable, composedOf, isInterfaceVariableOf, measures, isMeasuredBy, hasInterface, encapsules, describedBy, hasEquation, connectsTo, hasName, hasMath, hasPrivateInterface, hasURI, hasPublicInterface, hasInitialValue, hasSymbol, hasDetail, hasComponentURI, hasDescription, hasChemicalSymbol.</i></p>
Fontes de conhecimento	<i>Laboratory of Computational Physiology (FISIOCOMP), CellML website, technical reports and journals.</i>

Etapa 2: Conceitualização

Nesta etapa o conhecimento do domínio é organizado e estruturado através de modelos significativos no nível de conhecimento. Para isso, utiliza-se representações externas independentes da linguagem de implementação e do ambiente de desenvolvimento.

Participam da Conceitualização o engenheiro ontológico e o *stakeholder*. A etapa conta com o suporte das seguintes atividades: aquisição do conhecimento, avaliação, documentação, gerência de configuração, planejamento, qualidade e ambiente.

Neste estudo foram adotadas as representações intermediárias – glossário de termos e dicionário de conceitos – como forma de organizar e estruturar o conhecimento adquirido na etapa de Especificação. As Figuras 51 e 52 exibem, respectivamente, essas representações intermediárias possibilitando uma visão parcial da documentação (Apêndice 1).

Características de qualidade, tais como, coerência, corretude, independência da codificação e semântica conceitual, são características importantes para a etapa, e foram identificadas na etapa através da geração de modelos e artefatos.

Name	Description
CellElement	Refers to the physical structure of the cell.
ChemicalElement	Is a type of atom that is distinguished by its atomic number; that is, by the number of protons in its nucleus. The term is also used to refer to a pure chemical substance composed of atoms with the same number of protons.
Component	Represents the components of the model.
DomainEntity	Define the terms that will be used as a vocabulary shared by researchers in creating models.

Figura 51. Representação intermediária para a ontologia CeIO, parte do glossário de termos.

Concept name	Synonyms	Acronyms	Instances	Class attributes	Instances attributes	Relations
ChemicalElement	-	-	Calcium Chlorid Potassium Sodium	-	-	hasChemicalSymbol
CellElement	-	-	-	CellProcess CellStructure	-	-
Component	-	-	-	ExternalComponent InternalComponent	-	hasDomainEntity hasCellElement encapsule connectsTo
DomainEntity	-	-	-	BioChemicalEntity BiologicalEntity ChemicalEntity	-	-
Unit	-	-	-	DefinedUnit StandardUnit UserDefinedUnit	-	measures
Quantity	-	-	-	BaseQuantity DerivedQuantity UserDefinedQuantity	-	isMeasuredBy

Figura 52. Representação intermediária para a ontologia CeIO, dicionário parcial de conceitos.

Etapa 3: Formalização

A Formalização tem por objetivo transformar o modelo conceitual construído até o momento em um modelo formal. Dessa forma, um projeto de ontologia pode ser elaborado a partir desta formalização.

O engenheiro ontológico, o *stakeholder* e o *desenvolvedor participam ativamente desta etapa*. A Formalização apresenta as atividades de: aquisição do conhecimento, avaliação, documentação, gerência de configuração, qualidade e ambiente.

Nesta etapa adotamos a árvore de classificação dos conceitos e os diagramas de relacionamento binário como representações intermediárias do domínio, as quais possibilitam organizar e estruturar o conhecimento adquirido até o momento. A Figura 53 exibe essas representações intermediárias possibilitando uma visão parcial da documentação.

A documentação completa desta etapa encontra-se no Apêndice 1.

As características de qualidade envolvidas na etapa são: organização, modularidade, estrutura invariante, independência da codificação, semântica conceitual, logicamente correta e restrições, que podem ser identificadas nos modelos e artefatos gerados.

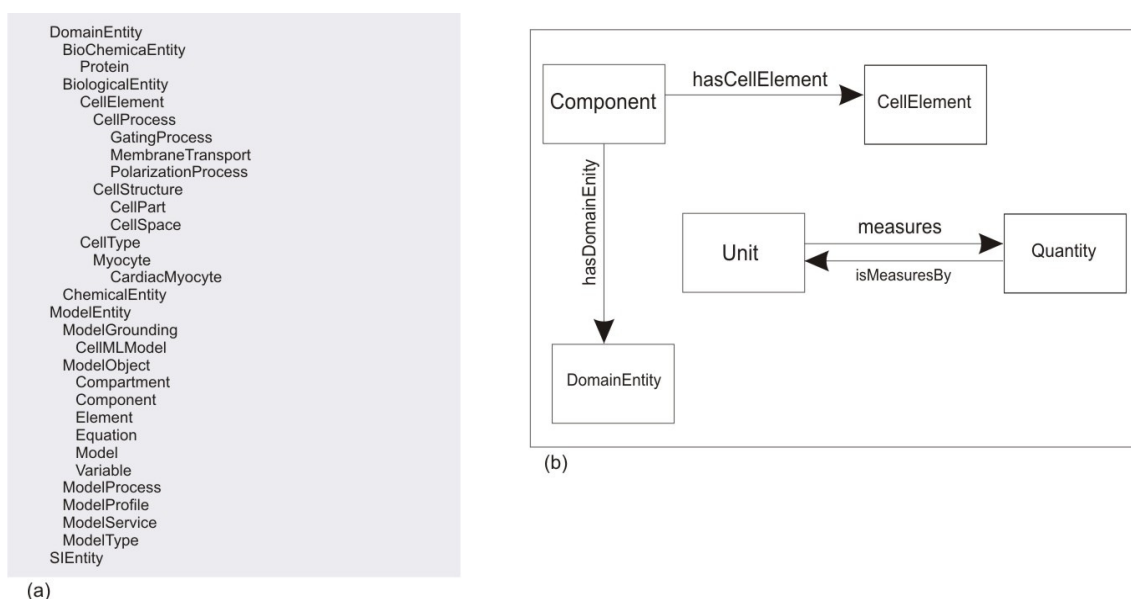


Figura 53. Representação intermediária para a ontologia CeIO: (a) parte de uma árvore de classificação dos conceitos; (b) um diagrama de relacionamento binário de conceitos.

Etapa 4: Implementação

Na implementação ocorre o processo de criação da ontologia. Como ferramenta de desenvolvimento, utilizamos o Protégé [PROTÉGÉ 2010] possibilitando assim a geração do código-fonte da ontologia, no caso OWL. Atualmente a ferramenta Protégé vem sendo muito utilizada, por ser uma ferramenta com boa interface, de fácil manuseio e também por apresentar *plugins* para diversas visualizações e funcionalidades. As Figuras 54 e 55, apresentam a construção da ontologia CeIO usando o ambiente Protégé. A Figura 54 exibe a criação de uma subclasse a partir de outra já criada. E a Figura 55 apresenta a tela de edição de uma classe, na qual é permitido adicionar propriedades e restrições relacionadas à classe. A codificação completa encontra-se em www.ufjf.br/npqs.

A etapa conta com o suporte do engenheiro ontológico e do desenvolvedor. As atividades presentes na Implementação são: avaliação, documentação, gerência de configuração, qualidade e ambiente.

Vale ressaltar que, na etapa de Implementação, também é gerado o código da ontologia. Apesar do desenvolvimento ser feito por meio de um ambiente gráfico, é possível obter a codificação da ontologia na linguagem OWL.

Propriedades formais, semântica conceitual, logicamente correta, restrições e inferência automatizada são as características de qualidade essenciais na Implementação, obtidas pela escolha do ambiente de desenvolvimento adequado e geração dos artefatos.

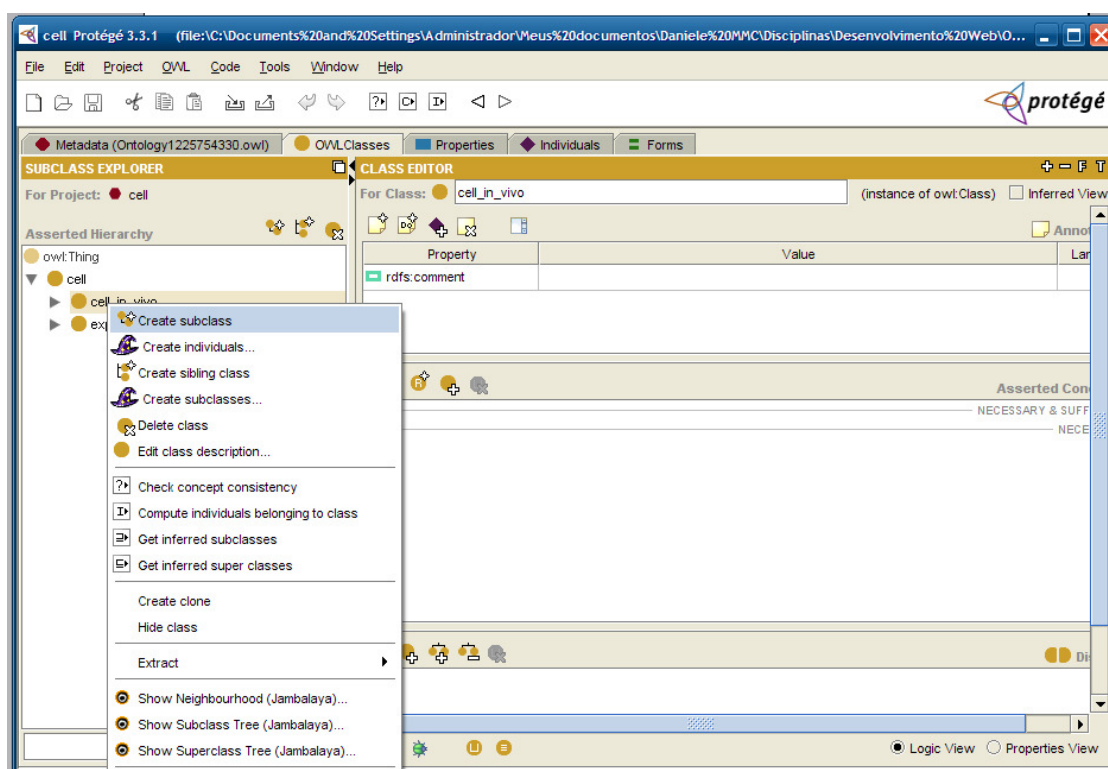


Figura 54. Tela da implementação, criação de subclasse no ambiente Protégé [PROTÉGÉ 2010].

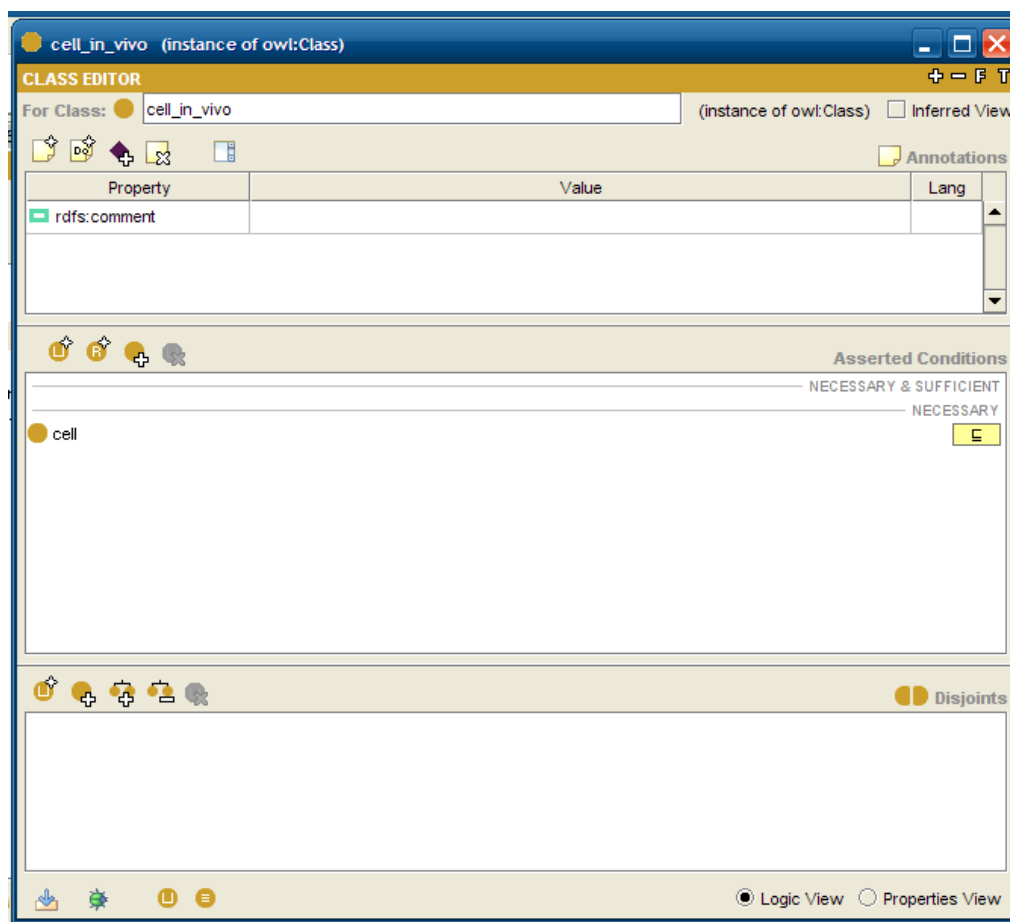


Figura 55. Tela da implementação, edição de classes no ambiente Protégé [PROTÉGÉ 2010].

Etapa 5: Integração

A atividade de integração considera o reuso e a compatibilidade com outras definições existentes. Para isso, verificamos a existência de ontologias de domínio complementares e inspecionamos os conceitos usados na modelagem da ontologia para selecionar a que melhor se ajustava à conceitualização da CelO. Através dessa atividade buscamos nas ontologias selecionadas definições de termos coerentes com os da CelO.

Na integração ocorreu o alinhamento entre ontologias, processo de interoperabilidade entre ontologias adotado neste trabalho, como meio de se obter uma relação entre os termos correspondentes entre duas ontologias.

A Integração conta com o suporte do engenheiro ontológico e do desenvolvedor. As atividades pertencentes à Integração são: Avaliação, Documentação, Gerência de Configuração, Qualidade e Ambiente.

As ontologias de domínio complementares utilizadas, pertencentes ao grupo OBO *Foundry* [OBO FOUNDRY 2010] foram citadas no capítulo 2.

Como resultado desta etapa foi elaborado um documento contendo informações das ontologias utilizadas, indicando para cada termo o nome do termo no modelo conceitual, o nome da ontologia de onde foi extraído o termo e a definição do mesmo na ontologia. O Quadro 17 apresenta a representação deste documento integrado.

Quadro 17. Documento de Integração do primeiro ciclo.

Meta-Ontologia	Ontologias	
Termo na CelO	Ontologia utilizada	Nome do termo na ontologia utilizada
<i>CellProcess</i>	GO in OBO	<i>cellular_process</i>
<i>CellPart</i>	GO in OBO	<i>cell_part</i>
<i>MembraneTransport</i>	GO in OBO	<i>transmembrane_transport</i>
<i>CellStructure</i>	Cell in OBO	<i>structural_cell</i>
<i>CellSpace</i>	Cell and CARO in OBO	<i>cell_space_location</i>
<i>CardiacMyocyte</i>	Cell in OBO	<i>cardiac_muscle_cell</i>

Com a finalização desta etapa, observamos a necessidade de alterar alguns termos da CelO, o que possibilitaria uma melhora na qualidade das informações, já que o grupo OBO *Foundry* é uma referência no desenvolvimento de ontologias do domínio biológico. Portanto, esse processo deixa a CelO mais próxima dos padrões de nomenclatura estabelecidos pela OBO *Foundry* e amplia sua capacidade de reutilização e interoperabilidade.

Apesar de terem ocorridos poucas modificações, elas foram determinantes para a evolução da CelO. Por exemplo, a classe que antes era definida por “*CellProcess*” foi renomeada para “*cellular_process*”, o mesmo aconteceu com os termos “*CellStructure*”, “*CellSpace*”, “*CardiacMyocyte*”, que foram substituídos respectivamente por “*structural_cell*”, “*cell_space_location*” e “*cardiac_muscle_cell*”.

A Figura 56 apresenta o diagrama da ontologia CelO após a sua integração com as ontologias da OBO [MATOS, E. E. S. *et al*, 2009].

Após a etapa de Integração a ontologia CelO apresenta características de qualidade, como: extensibilidade, interoperabilidade semântica, similaridade semântica independência da codificação e inferência automatizada.

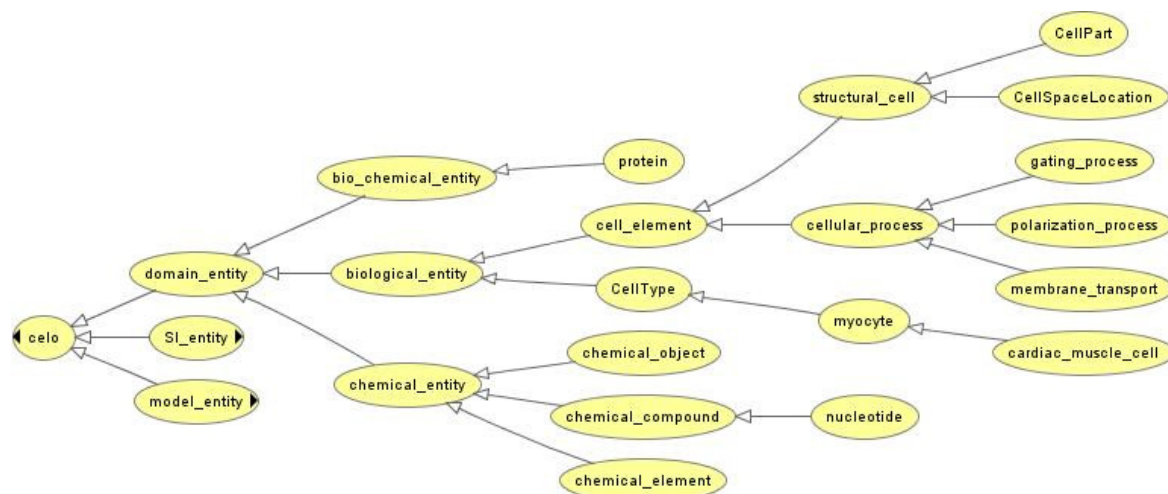


Figura 56. Ontologia CelO após a etapa de Integração.

Etapa 6: Evolução

Até o momento estávamos focados em realizar a reengenharia da CelO e prover a sua documentação. Esta última etapa tem como objetivo apoiar o enriquecimento do conhecimento da ontologia, para tanto, novos conceitos podem ser adicionados ou expandidos. Neste ponto voltamos à etapa inicial da metodologia, o que nos permitirá evoluir a CelO.

6.2 CICLO 2 – EXPANSÃO DA CelO EM CelO – Human Disease

Etapa 1: Especificação

Na expansão do domínio da CelO, antes de mais nada, foi necessário identificar todos os possíveis caminhos existentes para então definir o novo subdomínio. Foram então contactados especialistas de domínio que poderiam contribuir com o seu conhecimento.

Para chegar ao domínio expandido neste trabalho, foram feitas reuniões com alguns especialistas de áreas distintas no domínio da Biologia. Foram realizadas entrevistas com professores das áreas de Medicina, Química e Biologia. Após essas entrevistas verificou-se que o domínio das células, especificamente Doenças Humanas, poderia ser expandido através da CelO.

Antes de iniciar as atividades envolvidas na etapa de Especificação, foi realizada uma reunião com a *stakeholder* com a finalidade de se apresentar a ontologia CelO e os conceitos básicos sobre ontologias.

A partir disso, novas entrevistas foram realizadas com o objetivo de extrair todo o conhecimento da especialista. A primeira reunião resultou na definição do escopo e conseqüentemente na montagem de uma taxonomia. A segunda possibilitou, além da definição dos termos que constituiriam a ontologia, também uma comparação com as ontologias pertencentes ao grupo OBO *Foundry* [OBO FOUNDRY 2010].

O resultado desses encontros foi a obtenção de diferentes versões da ontologia. A primeira versão é apresentada no Quadro 18, observa-se que o número de termos que a constituem é reduzido se comparado com a última versão da ontologia apresentada no Quadro 19. As principais versões que representam a evolução da extração do conhecimento para a representação da ontologia constam no Apêndice 2.

Antes que se chegasse a uma versão final dos termos da ontologia, foi feita uma validação com seis especialistas do domínio, todos mestre e doutores, incluindo a *stakeholder*. A validação foi composta de um questionário onde os avaliadores respondiam se as perguntas constituintes eram verdadeiras ou falsas, se os diagramas que exibiam a hierarquia das classes estavam corretos, além de possuir um campo onde os especialistas poderiam sugerir novas ligações, termos e sinônimos. O instrumento completo submetido aos especialistas encontra-se no Apêndice 3. As propriedades de “é parte de” e “tem” foram definidas.

Observa-se que a Especificação teve a participação do Engenheiro Ontológico e do *stakeholder*, que foram fundamentais para o desenvolvimento das atividades de aquisição do conhecimento, avaliação, documentação, gerência de configuração, planejamento, qualidade e ambiente, presentes na etapa.

A etapa, através dos modelos e artefatos gerados, apresenta ainda as características de qualidade: clareza dos termos, unicidade dos termos, compromisso ontológico e relação com a realidade.

Quadro 18. Especificação da expansão da ontologia CelO (versão 1).

Domínio	<i>Human Disease</i>
Data	2009
Desenvolvida por	Elaine Coimbra

Objetivo	<i>Expand the domain represented in the ontolgy CelO to infectious human disease.</i>
Nível de formalidade	Semiformal
Escopo	<i>List of 19 concepts: Cell, Disease, Human_Disease, Infection_and_Parasitic_Disease, Bacterial, Parasitc, Viral, Yest, Helmenths, Protozoan, Filarials, Intestinal_Helminthes, Schistosomiase, Amebiase, Chagas_Disease, Giardiases, Leishmaniose, Malaria, Toxoplasnose</i>
Fontes de conhecimento	http://www.cdc.gov/ncidod/dpd/parasites/ http://www.who.int/topics/en/ http://www.dpd.cdc.gov/dpdx/Default.htm <i>One stakeholder</i>

Quadro 19. Especificação da expansão da ontologia CelO (versão 2).

Domínio	<i>Human Disease</i>
Data	2009
Desenvolvida por	Elaine Coimbra
Objetivo	<i>Expand the domain represented in the ontolgy CelO from infectious human disease.</i>
Nível de formalidade	<i>Semiformal</i>
Escopo	<i>List of 48 concepts: Acanthamoeba infection, African trypanosomiasis, Amebiasis, Angiostrongyliasis, Anisakiasis, Ascariasis, Babesiosis, Bacterial Diseases, Balantidiasis, Capillariasis, Cell, Chagas Disease, Clonorchiasis, Cryptosporidiosis, Cutaneous Larva Migrans, Cyclosporiasis, Cysticercosis, Diphyllbothriasis, Disease, Dracunculiasis, Ectoparasite, Enterobiasis, Fascioliasis, Fungal Diseases, Giardiasis, Helminths, Hookworm Infection, Human Disease, Hydatidosis, Hymenolepiasis, Infectious Disease, Visceral Larva Migrans, Leishmaniasis, Malaria, Microsporidiosis, Naegleria Infection, Onchocerciasis, Paragonimiasis, Parasitic Disease, Protozoa, Schistosomiasis, Strongyloidiasis, Taeniasis, Toxoplasmosis, Trichinellosis, Trichomoniasis, Trichuriasis, Viral Disease.</i> <i>List of 12 properties: isDisease, isDiseaseCausedBy, isParasitic, isTypeOf, hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasprognostic, hasProtozoa,, hasTransmission.</i>
Fontes de conhecimento	http://www.cdc.gov/ncidod/dpd/parasites/ http://www.who.int/topics/en/ http://www.dpd.cdc.gov/dpdx/Default.htm <i>6 stakeholders</i>

Etapa 2: Conceitualização

Esta etapa tem por objetivo organizar e estruturar o conhecimento do domínio, para isso, representações externas que independem da linguagem de implementação e do ambiente de desenvolvimento são adotados.

O engenheiro ontológico e o *stakeholder* são os principais participantes da Conceitualização. A etapa tem o apoio das atividades de: aquisição do conhecimento, avaliação, documentação, gerência de configuração, planejamento, qualidade e ambiente.

As Figuras 57 e 58 representam, respectivamente, o glossário de termos e o dicionário de conceitos parciais produzidos na Conceitualização durante a ciclo de expansão da CeIO. No dicionário de conceitos foram acrescentados os sinônimos em português. No Apêndice1 encontra-se o glossário de termos e o dicionário de conceitos completos.

Coerência, corretude, independência da codificação e semântica conceitual são as características de qualidade presentes na Conceitualização, identificadas nos modelos e artefatos gerados.

Name	Description
Disease	A condition of the living animal or plant body or of one its parts that impairs normal functioning. Illness and sickness are generally used as synonyms for disease.
Human_Disease	Is an abnormal condition of an organism human that impairs bodily functions, associated with specific symptoms and signs.
Infections_Disease	Is a clinically evident disease resulting from the presence of pathogenic microbial agents, including pathogenic viruses, pathogenic bacteria, fungi, protozoa, multicellular parasites, and aberrant proteins known as prions.
Bacterial_Disease	Infectious diseases caused by pathogenic bacteria. Examples of bacterial diseases: tuberculosis (<i>Mycobacterium tuberculosis</i>), pneumonia (which can be caused by <i>Streptococcus</i> and <i>Pseudomonas</i>) and foodborne illnesses (caused by bacteria such as <i>Shignella</i> , <i>Campylobacter</i> and <i>Salmonella</i>). Pathogenic bacteria also cause infections such as tetanus, typhoid fever, diphtheria, syphilis and leprosy.
Fungal_Disease	Caused by several fungi species, are called mycoses. Examples of some fungal diseases: aspergillosis, blastomycosis, candidiasis, coccidioidomycosis, cryptococcosis, histoplasmosis, and sporotrichosis.
Parasitic_Disease	Diseases caused by parasite (any organism that lives in or on another organism without benefiting the host organism; commonly refers to pathogens, most commonly in reference to protozoans, helminthes and extparasites).
Viral_Disease	Caused by several viruses. Viruses are the smallest life-form existing, since they are not even a single cell. It is almost like they are not alive at all. They are small strands of DNA-like cell material. A virus consists mostly of RNA and cannot survive without host cells.
Ectoparsites	A classification of animals that includes those with hard, segmented bodies and jointed appendages, such as insects. Ectoparasites are usually arthropods which parasitize the skin.
Helminths	A large, multicellular organism that is generally visible to the naked eye in its adult stages. Helminths can be free-living or parasitic.
Protozoa	One-celled organism microscopic that can be free-living or parasitic in nature. They are able to multiply in humans which contributes to their survival and also permits serious infections to develop from just a single organism.

Figura 57. Glossário de termos produzido na Conceitualização do ciclo de expansão da CeIO.

Concept name	Synonyms	Class attributes	Relations
Taeniasis	Tapeworm infection Teníase	-	hasCause hasDefinition hasDiagnostic hasHelminths hasPrevention hasPrognostic hasTransmission
Parasitic_Disease	Doença parasitária	Ectoparasite Helminths Protozoa	hasParasitic
Infectious Disease	Doença infecciosa	Bacterial Disease Fungal Disease Parasitic Disease Viral Disease	-
Chagas Disease	American Trypanosomiasis Doença de Chagas	-	hasCause hasDefinition hasDiagnostic hasProtozoa hasPrevention hasPrognostic hasTransmission

Figura 58. Dicionário de conceitos produzido na Conceitualização do ciclo de expansão da CelO.

Etapa 3: Formalização

Nas etapas anteriores realizamos tarefas voltadas principalmente para delimitação do escopo e aquisição dos termos que constituíram a ontologia. Com isso é necessário transformar o modelo conceitual construído em um modelo mais formal e que permita projetar a ontologia para o ambiente de implementação. A Formalização tem como objetivo fazer essa transformação, utilizando para tanto, como representações intermediárias do domínio em questão, a árvore de classificação dos conceitos e os diagramas de relacionamento binário. As Figuras 59 e 60 exibem respectivamente a árvore de classificação e o diagrama de relacionamento parciais da CelO – Human Disease. As regras, que representam os relacionamentos, foram formalizadas. Na Figura 61 são apresentadas algumas dessas regras obtidas na etapa.

O engenheiro ontológico, o *stakeholder* e o desenvolvedor participam ativamente desta etapa. A Formalização apresenta as atividades de: aquisição do conhecimento, avaliação, documentação, gerência de configuração, qualidade e ambiente.

As características de qualidade envolvidas na etapa são: organização, modularidade, estrutura invariante, independência da codificação, semântica conceitual, logicamente correta e restrições.

A documentação completa desta etapa encontra-se no Apêndice 1.

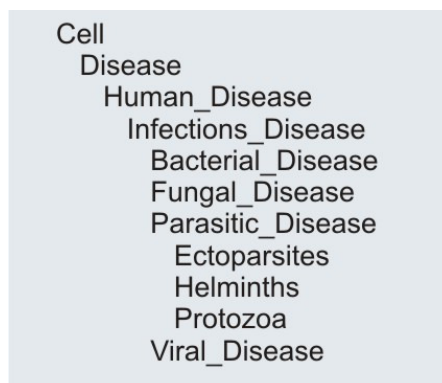


Figura 59. Representação intermediária para a ontologia CeLO Human Disease: árvore de classificação parcial dos conceitos.

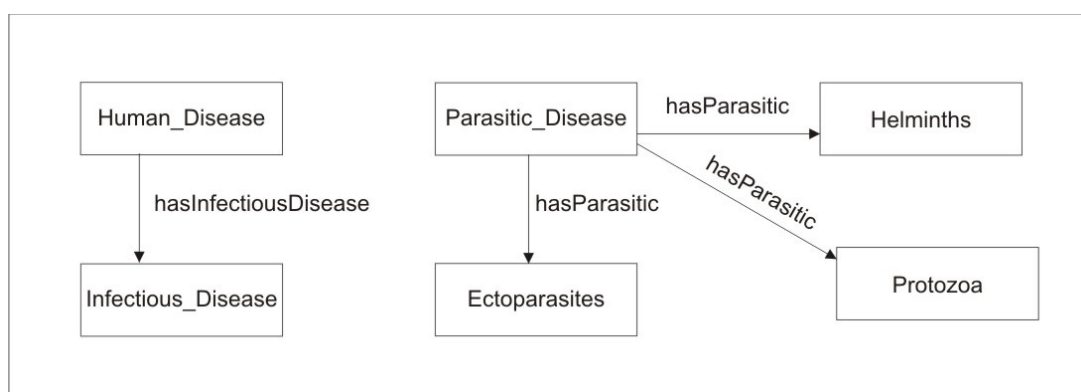


Figura 60. Representação intermediária para a ontologia CeLO-Human Disease: diagrama de relacionamento parcial dos conceitos.

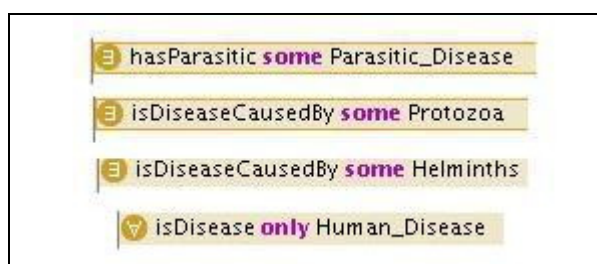


Figura 61. Regras obtidas durante a etapa de Formalização.

Etapa 4: Implementação

Na Implementação construímos a ontologia em um ambiente próprio, a partir dos artefatos produzidos nas etapas anteriores, dessa forma obtemos a ontologia codificada em uma linguagem específica. Foi utilizado o ambiente Protégé como

ferramenta de desenvolvimento da ontologia. As Figuras 62 e 63 apresentam respectivamente a construção da nova ontologia usando o ambiente Protégé e seu código OWL. Na figura 62 são exibidas todas as informações relacionadas à classe *Helminths* como a definição e sinônimos do termo, classes disjuntas, além de sua hierarquia. A Figura 63 apresenta parcialmente a codificação da ontologia. A codificação completa encontra-se em www.ufjf.br/npqs.

O engenheiro ontológico e o desenvolvedor são os participantes envolvidos na etapa. A Implementação é apoiada pelas atividades de: avaliação, documentação, gerência de configuração, qualidade e ambiente.

Propriedades formais, semântica conceitual, logicamente correta, restrições e inferência automatizada são as características de qualidade importantes ao final da etapa de Implementação.

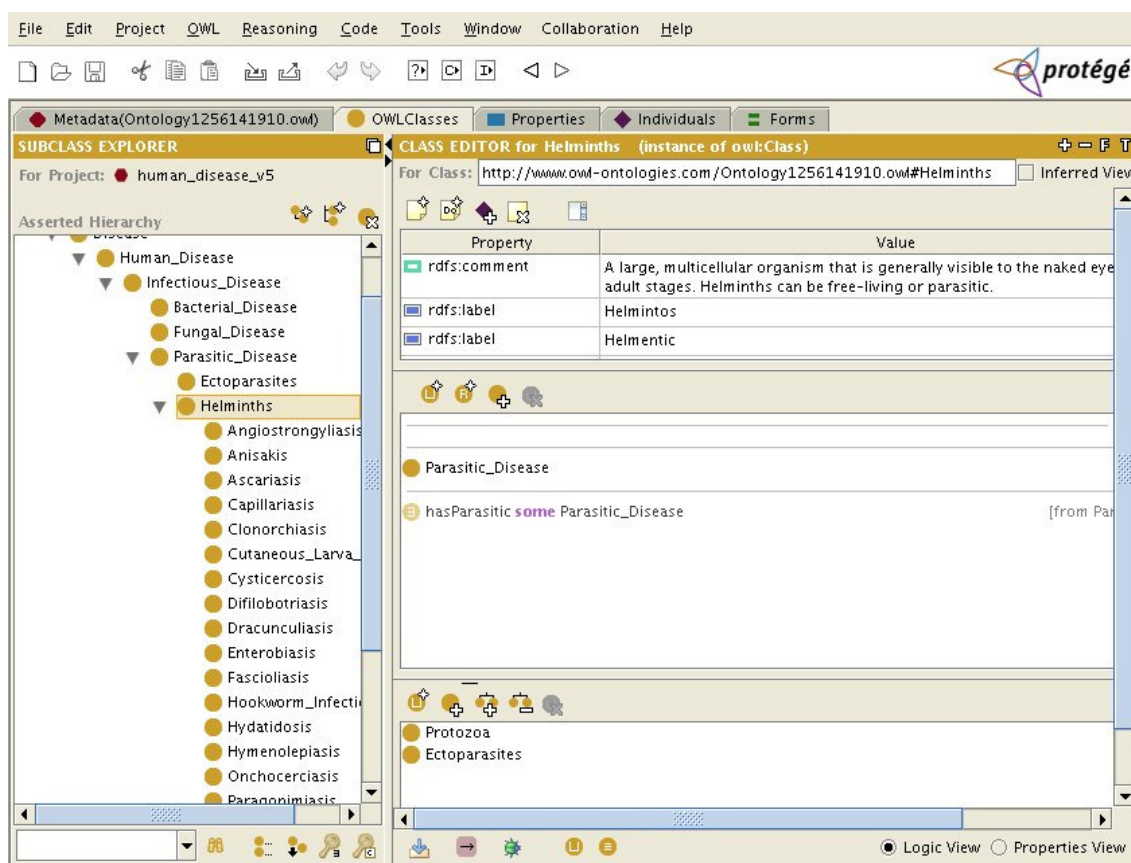


Figura 62. Construção da ontologia CeLO – Human Disease através do ambiente Protégé apresentando todas as informações relacionadas à classe *Helminths*.

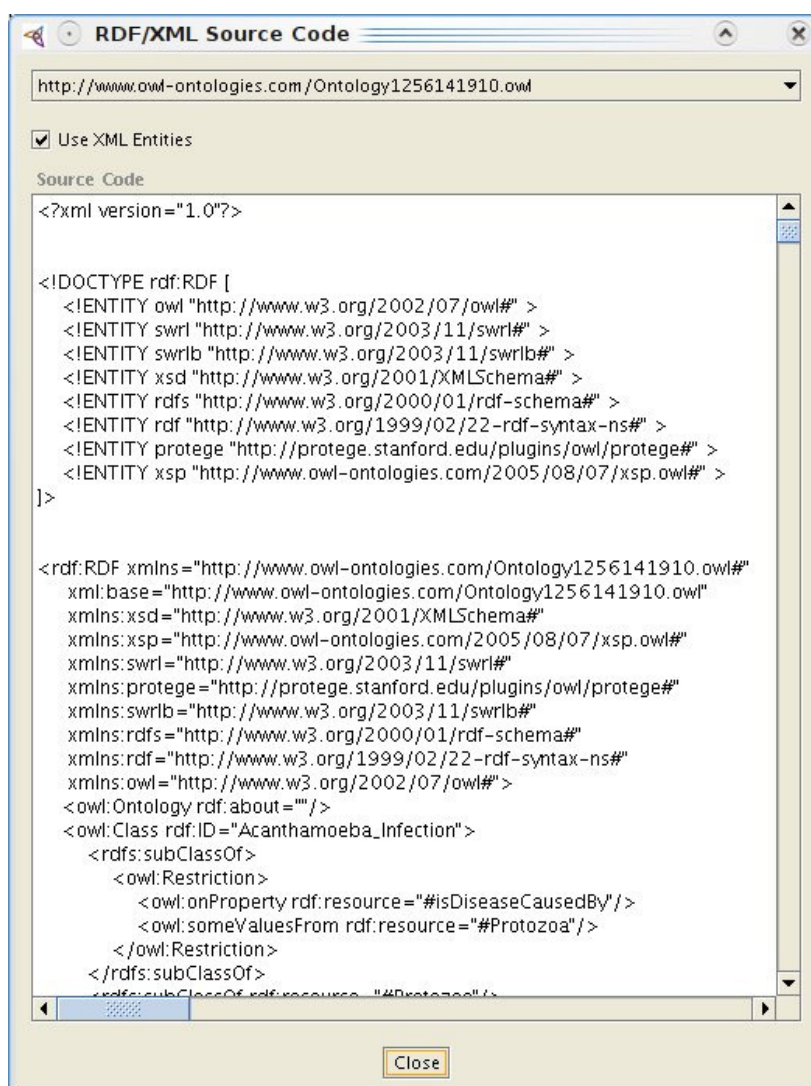


Figura 63. Codificação parcial da ontologia CelO - Human Disease.

Etapa 5: Integração

Com a preocupação de garantir a compatibilidade com outras definições existentes, a Integração busca por outras ontologias que possuem domínios complementares a fim de que se possa encontrar definições próximas às da nossa conceitualização.

A Integração conta com a participação do engenheiro ontológico e do desenvolvedor. As atividades pertencentes à etapa são: avaliação, documentação, gerência de configuração, qualidade e ambiente.

O alinhamento entre ontologias é o recurso adotado nesta etapa. Assim como no primeiro ciclo, o alinhamento também foi feito com uma ontologia do grupo OBO *Foundry* denominada *human_disease*.

O documento de integração exibe as informações da ontologia utilizada, indicando, para cada termo utilizado, nome do termo no nosso modelo conceitual, o

nome da ontologia de onde foi extraído o termo e a definição na ontologia. O Quadro 20 apresenta uma representação deste documento integrado, obtido pelo uso da ferramenta A3O especificada na seção 6.3.

Quadro 20. Documento de Integração do segundo ciclo.

Meta-Ontologia	Ontologias	
Termo na CelO – Human Disease	Ontologia utilizada	Nome do termo na ontologia utilizada
<i>Disease</i>	DOID in OBO	<i>disease</i>
<i>Infectious_Disease</i>	DOID in OBO	<i>infectious disease</i>
<i>Parasitic_Disease</i>	DOID in OBO	<i>parasitic disease</i>
<i>Viral_Disease</i>	DOID in OBO	<i>virus disease</i>
<i>Helminths</i>	DOID in OBO	<i>helminthiasis</i>
<i>Ectoparasites</i>	DOID in OBO	<i>ectoparasitic infestation</i>
<i>Toxoplasmosis</i>	DOID in OBO	<i>toxoplasmosis</i>
<i>Chagas_Disease</i>	DOID in OBO	<i>chagas' disease</i>
<i>Amebiasis</i>	DOID in OBO	<i>amebiasis</i>
<i>Cyclosporiasis</i>	DOID in OBO	<i>cyclosporiasis</i>
<i>Leishmaniasis</i>	DOID in OBO	<i>leishmaniasis</i>
<i>Cryptosporidiosis</i>	DOID in OBO	<i>cryptosporidiosis</i>
<i>Giardiasis</i>	DOID in OBO	<i>giardiasis</i>
<i>Babesiosis</i>	DOID in OBO	<i>babesiosis</i>
<i>Microsporidiosis</i>	DOID in OBO	<i>microsporidiosis</i>
<i>Trichomoniasis</i>	DOID in OBO	<i>trichomoniasis</i>
<i>Balantidiasis</i>	DOID in OBO	<i>balantidiasis</i>
<i>Malaria</i>	DOID in OBO	<i>Malaria</i>
<i>Difilobotriasis</i>	DOID in OBO	<i>Diphyllobothriasis</i>
<i>Clonorchiasis</i>	DOID in OBO	<i>clonorchiasis</i>
<i>Fascioliasis</i>	DOID in OBO	<i>fascioliasis</i>
<i>Hookworm_Infection</i>	DOID in OBO	<i>Hookworm infection</i>
<i>Onchocerciasis</i>	DOID in OBO	<i>onchocerciasis</i>
<i>Hymenolepiasis</i>	DOID in OBO	<i>hymenolepiasis</i>
<i>Capillariasis</i>	DOID in OBO	<i>capillariasis</i>
<i>Cysticercosis</i>	DOID in OBO	<i>cysticercosis</i>
<i>Schistosomiasis</i>	DOID in OBO	<i>schistosomiasis</i>
<i>Enterobiasis</i>	DOID in OBO	<i>enterobiasis</i>
<i>Paragonimiasis</i>	DOID in OBO	<i>paragonimiasis</i>
<i>Dracunculiasis</i>	DOID in OBO	<i>dracunculiasis</i>
<i>Visceral_Larva_Migrans</i>	DOID in OBO	<i>Visceral larva migrans</i>
<i>Taeniasis</i>	DOID in OBO	<i>teniasis</i>
<i>Ascariasis</i>	DOID in OBO	<i>ascariasis</i>
<i>Anisakis</i>	DOID in OBO	<i>anisakiasis</i>
<i>Strongyloidiasis</i>	DOID in OBO	<i>strongyloidiasis</i>

Após a etapa de Integração a ontologia CelO apresenta características de qualidade, como: extensibilidade, interoperabilidade semântica, similaridade semântica independência da codificação e inferência automatizada.

Etapa 6: Evolução

A etapa de evolução permite não só o enriquecimento da ontologia mas, também, a expansão de seu domínio e a sua manutenção quando necessário. Neste sentido, o

estudo de caso se encerra ficando a execução desta última etapa para os próximos trabalhos.

Ao atingirmos essa etapa o artefato gerado é uma versão final da ontologia evoluída, que além das características de qualidade, organização, modularidade, estrutura invariante e extensibilidade, deve acumular os atributos definidos nas etapas anteriores do processo de engenharia ontológica.

6.3 DESCRIÇÃO DA FERRAMENTA A3O PARA ALINHAMENTO ENTRE OBO E OWL

A abordagem QDAontology, proposta neste trabalho, na sua etapa de Integração, sugere que sejam aplicados mecanismos que promovam a interoperabilidade entre ontologias. No estudo de caso foram utilizadas ontologias representadas nas linguagens OBO e OWL, sendo que as ontologias em OBO são pertencentes ao renomado grupo *OBO Foundry* e as ontologias em OWL, são ontologias desenvolvidas durante os trabalhos do Mestrado em Modelagem Computacional. Para a realização de alinhamentos entre ontologias de linguagens distintas foi necessário o desenvolvimento de uma ferramenta que permitisse tal mecanismo.

A ferramenta A3O, desenvolvida para este estudo de caso, tem como objetivo proporcionar o alinhamento entre ontologias desenvolvidas nas linguagens OBO e OWL. Na estratégia proposta, o alinhamento é obtido em três passos, executados sequencialmente. Inicialmente a ferramenta compara os termos das ontologias no nível léxico e em seguida aplica uma medida de similaridade, que será detalhada adiante, para garantir os seus resultados. O resultado do processo de alinhamento é um arquivo contendo os relacionamentos entre os termos selecionados. A Figura 64 ilustra o diagrama com a sequência de passos proposta para a ferramenta A3O.

O primeiro passo faz a normalização das ontologias. Para aplicar essas transformações são utilizadas duas APIs, a Jena [JENA 2010] e OBO [API OBO 2010] que manipulam respectivamente as ontologias OWL e OBO de entrada. Para garantir que o cálculo de similaridade que será aplicado posteriormente seja o mais preciso possível, aplicamos a normalização dos termos. A normalização faz com que

os termos fiquem todos padronizados, para tanto, as letras são transformadas em caixa-alta, retiram-se caracteres especiais e espaços em branco. A Figura 65 apresenta a interface da ferramenta A3O, onde é mostrada a tela inicial para a escolha das duas ontologias de entrada para a realização do alinhamento.

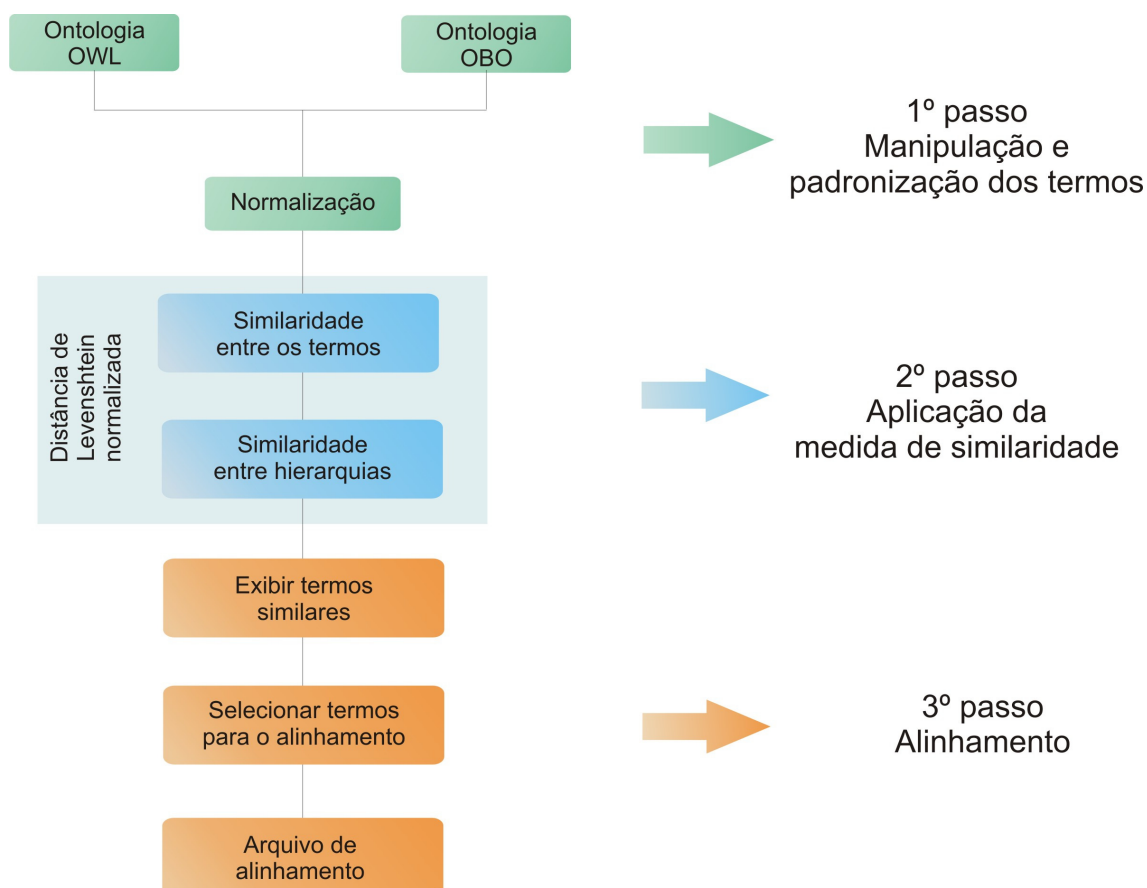


Figura 64. Diagrama da ferramenta A3O.

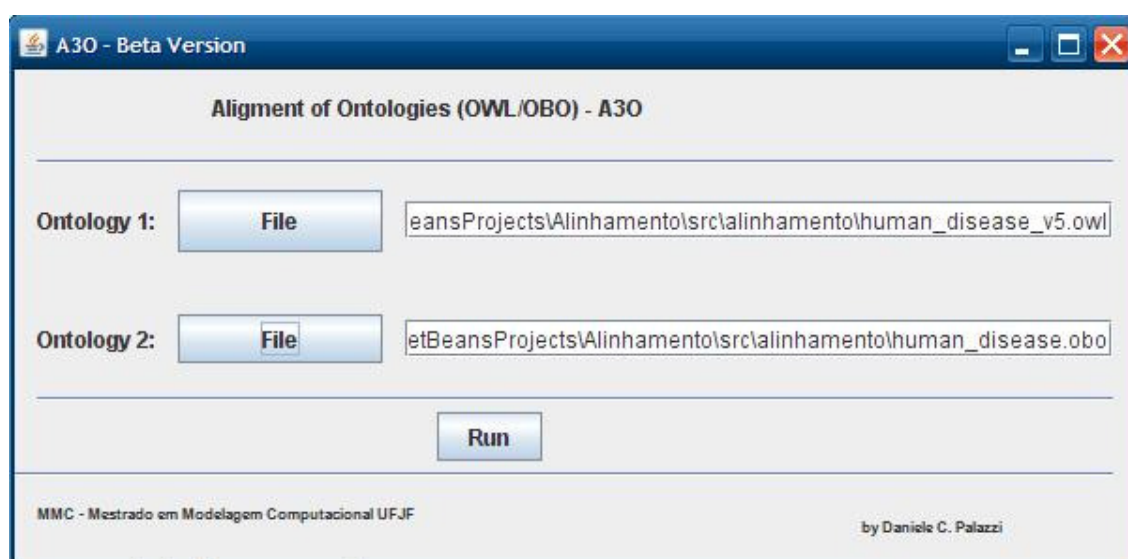


Figura 65. Interface da ferramenta A3O para a escolha das ontologias para o alinhamento.

O segundo passo aplica a medida de similaridade entre os pares de termos e entre níveis hierárquicos. A medida de similaridade adotada no desenvolvimento da ferramenta é a distância de *Levenshtein* ou distância de edição [EUZENAT, J., SHVAIKO, P. 2007]. Este método é uma técnica bem estabelecida para a ponderação da diferença entre duas *strings*. Ele mede o número mínimo de operações, tais como inserções, deleções e substituições necessárias para transformar uma palavra em outra. Por exemplo, dada as palavras “*test*” e “*tesl*”, a distância de *Levenshtein* é igual a 0, uma vez que nenhuma operação é necessária para transformar uma palavra na outra. Para as palavras “*tesl*” e “*tenl*”, a distância de *Levenshtein* é igual a 1, visto que a operação de substituição (mudar “s” para “t”) é suficiente para transformar uma palavra na outra. Outra medida adotada é a comparação das hierarquias dos termos, onde são comparados os caminhos, nesse caso considerando as superclasses dos termos até dois níveis.

Porém a distância de *Levenshtein* aplicada é aqui denominada por distância de *Levenshtein* normalizada, baseada no trabalho de [SOUZA, J. F. 2007], na qual o valor obtido é dividido pelo tamanho do maior termo envolvido na avaliação. Sendo assim a medida de similaridade obtida por essa técnica retorna valores no intervalo [0,1], no qual valores mais baixos demonstram um maior grau de similaridade entre os termos.

E finalmente, o terceiro e último passo, apresenta similaridade entre os termos das ontologias e entre as suas hierarquias dois níveis acima com base na distância de *Levenshtein* normalizada. Além disso, um cálculo denominado por SOMA é realizado com a finalidade de se obter a soma entre similaridades encontradas entre os termos e as hierarquias das ontologias. Para este cálculo, consideramos que a similaridade entre os termos tem um peso maior se comparado com a medida entre as hierarquias, de forma que, para cada par de termos o resultado é o somatório entre 70% da similaridade do termo e 30% da similaridade da hierarquia.

A fim de simplificar a exibição desses resultados, é necessário que o usuário escolha um termo da primeira ontologia que deseja obter a similaridade, através de uma árvore da mesma. Feita a escolha, são apresentados os resultados em ordem decrescente de similaridade comparando com todos os termos da segunda ontologia. As Figuras 66 e 67 exibem exemplos de similaridades encontradas para cada termo em relação aos termos da outra ontologia.

Alignment Ontologies (OWL/OBO) - A3O - Result

Ontology 1: Cell > Disease > Human_Disease > Infectious_Disease > Fungal_Disea > Parasitic_Disea > Protozoa > Helminth > Ectopara > Viral_Diseas > Bacterial_Disea

Term	T.S.	H.S.	Sum
disease	0.0	0.7	0.21
eye disease	0.3	0.906666666666...	0.482
Ear disease	0.3	0.948148148148...	0.494444444444...
Jaw disease	0.3	0.917647058823...	0.485294117647...
Lip disease	0.3	0.941666666666...	0.492499999999...
lyme disease	0.363636363636...	0.912499999999...	0.528295454545...
pica disease	0.363636363636...	1.0	0.554545454545...
bone disease	0.363636363636...	0.899999999999...	0.524545454545...
Lung disease	0.363636363636...	0.906666666666...	0.526545454545...
Nose disease	0.363636363636...	0.939130434782...	0.536284584980...
Iris disease	0.363636363636...	0.883333333333...	0.519545454545...
Lens disease	0.363636363636...	0.929999999999...	0.533545454545...
anus disease	0.363636363636...	0.892307692307...	0.522237762237...
Hair disease	0.363636363636...	0.946153846153...	0.538391608391...
Foot disease	0.363636363636...	0.946153846153...	0.538391608391...
nail disease	0.363636363636...	0.946153846153...	0.538391608391...

T.S.: Term Similarity, H.S.: Hierarchical Similarity, Sum: 0.3*H.S.+0.7*T.S

Buttons: Add Equivalent Term, Clear, Finish

Figura 66. Similaridades encontradas utilizando a ferramenta A3O para o termo DISEASE da ontologia CelO – Human Disease em relação à ontologia Human Disease da linguagem OBO.

Alignment Ontologies (OWL/OBO) - A3O - Result

Helminths > Fascioliasis

Term	T.S.	H.S.	Sum
fascioliasis	0.0	0.75	0.224999999999...
fascioloidiasis	0.2	0.612171052631...	0.323651315789...
fasciolopsiasis	0.2	0.654761904761...	0.336428571428...
ascariasis	0.333333333333...	0.844473684210...	0.486675438596...
Fasciitis	0.333333333333...	0.847916666666...	0.487708333333...
ascariadisis	0.416666666666...	0.844473684210...	0.545008771929...
Vasculitis	0.416666666666...	0.726666666666...	0.509666666666...
candidiasis	0.416666666666...	0.831363636363...	0.541075757575...
mansonelliasis	0.428571428571...	0.705625	0.5116875
capillariasis	0.461538461538...	0.541471571906...	0.485518394648...
Schistosomiasis	0.461538461538...	0.844473684210...	0.576419028340...
Enterobiasis	0.461538461538...	0.781045751633...	0.557390648567...
Facial paralysis	0.466666666666...	0.706862745098...	0.538725490196...
trombiculiasis	0.5	0.722499999999...	0.566675
necatoriasis	0.5	0.734117647058...	0.570235294117...
anisakiasis	0.5	0.844473684210...	0.603342105263...

T.S.: Term Similarity, H.S.: Hierarchical Similarity, Sum: 0.3*H.S.+0.7*T.S

Buttons: Add Equivalent Term, Clear, Finish

Figura 67. Similaridades encontradas utilizando a ferramenta A3O para o termo FASCIOLIASIS da ontologia CelO – Human Disease em relação à ontologia Human Disease da linguagem OBO.

Ao escolher dois termos similares, é possível gerar o arquivo de alinhamento contendo essas informações. Para tanto, o usuário deve escolher esses termos, clicando primeiro no termo exibido na árvore e depois no termo da segunda ontologia apresentada na tabela, em seguida clicar no botão *Add* para adicionar essa ligação. Num campo específico são apresentados todos os alinhamentos indicados e que ao término da execução da A3O serão salvos no arquivo de alinhamento. O arquivo de alinhamento contém expressões do tipo ONTO1.TermoX

⇔ ONTO2.TermoY os nomes das ontologias e seus termos equivalentes, por exemplo,

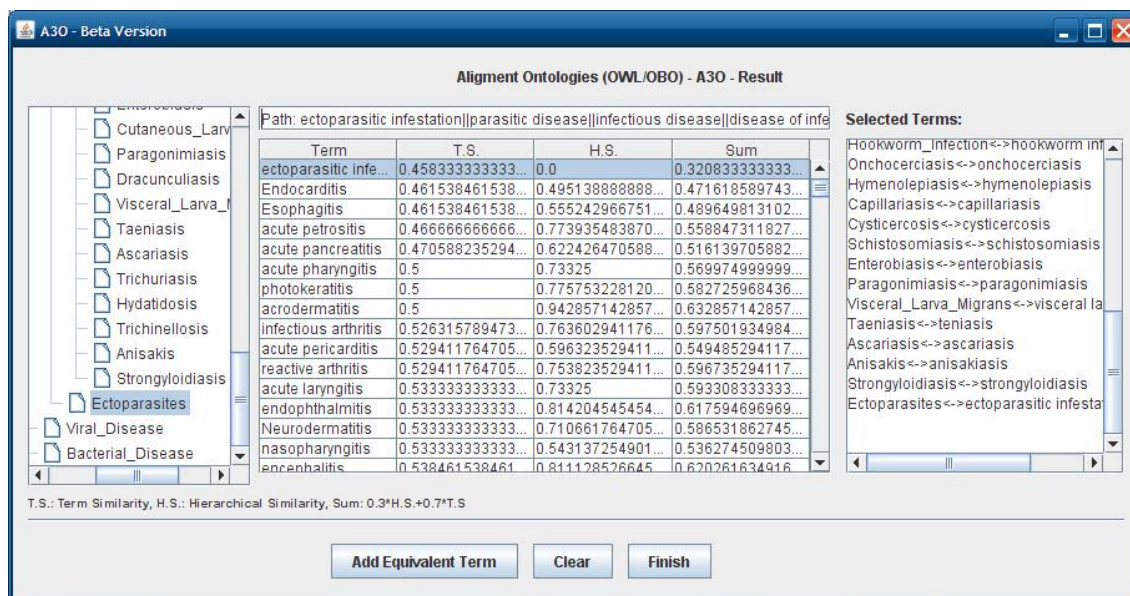


Figura 68. Adicionando termos equivalentes entre as ontologia CeLO – Human Disease e Human Disease da linguagem OBO.

6.3.1 Implementação

Para o desenvolvimento da ferramenta A3O foi adotada a linguagem de programação Java. Esta escolha deve-se principalmente por esta apresentar características fundamentais como portabilidade, escalabilidade e multiplataforma, além de permitir o reuso de códigos através de bibliotecas disponíveis. A implementação fez uso de duas dessas bibliotecas. A API – *Application Programming Interface* – Jena [JENA 2010], específica para manipulação de ontologias escritas em OWL e a API OBO [API OBO 2010], para manipulação de ontologias desenvolvidas na linguagem OBO.

6.3.2 Documentação da ferramenta A3O

De acordo com [LACHTIM, F. A. *et al.*, 2008], podemos documentar e avaliar ferramentas que envolvem interoperabilidade entre ontologias através de um

conjunto de características funcionais, consideradas fundamentais no processo de interoperabilidade entre ontologias. Seguindo esse conjunto de características, a ferramenta A3O é descrita como segue:

- i. Linguagens das ontologias suportadas: o alinhamento é realizado entre ontologias representadas nas linguagens OBO e OWL;
- ii. Conteúdo de entrada: utiliza duas ontologias como entrada, uma em OBO e outra em OWL ou duas em OBO ou duas em OWL, para a especificação de relações e posteriormente para a transformação dos dados;
- iii. Usabilidade: possui uma interface que facilita a realização dos alinhamentos. A ferramenta sugere ainda alinhamentos seguindo um grau de similaridade de acordo com as similaridades encontradas;
- iv. Técnicas de similaridades aplicadas: utiliza similaridade léxica dos termos e das hierarquias da ontologia, especificamente a distância de *Levenshtein* normalizada;
- v. Conteúdo de saída: as informações de saída constam de um relatório com as ligações e os termos equivalentes, correspondentes ao processo de alinhamento entre as duas ontologias;
- vi. Arquitetura: o módulo de alinhamento está dividido em três passos. O primeiro responsável pela transformação das informações contidas nas ontologias em modelo orientado a objetos. O segundo onde é aplicada a medida de similaridade de *Levenshtein* entre os termos e as hierarquias. E por último, o terceiro passo exibe os possíveis alinhamentos encontrados e gera os arquivos com os termos similares;
- vii. Extensibilidade: adoção de linguagem de software livre que permite incluir ou modificar seu código;
- viii. Linguagem de desenvolvimento: Java;
- ix. Sistema operacional: pode ser executado independente da plataforma/SO;
- x. Licença de uso: não há necessidade de aquisição de licenças, pois é *opensource*;
- xi. Nível de automação: a partir do uso da ferramenta é possível calcular automaticamente o grau de similaridade entre termos e hierarquias.

6.4 AVALIAÇÃO DA APLICAÇÃO DA ABORDAGEM QDAontology

Com relação à abordagem proposta nessa dissertação, a sua aplicação no processo de reengenharia da ontologia CelO e na sua evolução no subdomínio *Human Disease* mostrou a sua validação e a sua viabilidade em projetos na área de *e-Science*.

A divisão da abordagem QDAontology em seis etapas se mostrou adequada e ficou visível que o processo de engenharia ontológica proposto é composto de três blocos distintos: o primeiro focado na aquisição e formalização do conhecimento, que inclui as etapas de Especificação, Conceitualização e Formalização, o segundo bloco focado no desenvolvimento e correspondência ontológica, e como último o bloco que inclui a etapa de Evolução, onde o ciclo se completa. A opção por um processo evolutivo se mostrou viável para projetos que contam com equipes multidisciplinares. Percebeu-se, entretanto, que as etapas e as atividades muitas vezes ocorrem paralelamente e não de forma tão sequencial.

Em relação às atividades, apesar de existirem outras que poderiam ser incluídas na proposta, as sete escolhidas representaram ações ocorridas ao longo das etapas e geraram os principais documentos propostos. A aquisição do conhecimento representou a atividade com maior destaque, tendo em vista que a definição do domínio, dos termos, das relações e restrições exigiu muitas reuniões com especialistas e uso de técnicas de validação que garantissem a corretude da ontologia.

A avaliação ocorreu ao longo de todo o processo, mas a validação com os especialistas nas etapas de Especificação, Conceitualização e Formalização sintetizou esta atividade.

A documentação se mostrou uma atividade fundamental no processo evolutivo e ocorreu ao longo de todas as etapas gerando diversos artefatos. Os artefatos garantiram o armazenamento de diversas versões dos documentos, possibilitaram a validação contínua em todas as etapas e permitiram a aderência de todo processo ontológico às características de qualidade definidas.

Em qualquer processo de desenvolvimento de engenharia ontológica, a análise, projeto e estudo de viabilidade são fundamentais para o planejamento de uma ontologia na área científica mesmo que em muitas vezes esse planejamento

sofre alterações ao longo de todo o processo de desenvolvimento. Na área científica essa atividade se mostrou mais informal.

A atividade de gerência de configuração, apesar de não ter sido utilizada nenhuma ferramenta, se mostrou importante porque é visível a evolução da ontologia à medida que os conceitos vão sendo definidos conforme mostrado no Apêndice 2.

Em relação à atividade de qualidade, mesmo não tendo sido definido um plano de controle de qualidade, as características definidas para o processo e para o produto se mostraram viáveis de serem atingidas. A falta de métricas para medi-las não comprometeu a proposta da abordagem, mas é necessário defini-las para que outros as utilizem. O ambiente de desenvolvimento de uma ontologia inclui uma grande variedade de ferramentas automatizadas e semi-automatizadas. Portanto, sua definição e disponibilidade ao longo de todo o processo se mostraram muito importantes. O domínio das mesmas pelo engenheiro ontológico e desenvolvedor é imprescindível para a geração dos artefatos e conseqüente avaliação dos mesmos.

A abordagem propôs três participantes principais e de certa forma a equipe envolvida no estudo de caso ficou bem representada por essas personagens. Os *stakeholders* variaram de uma a seis pessoas dependendo da etapa do processo. Contamos com dois Engenheiros Ontológicos e um desenvolvedor.

Quanto ao uso da abordagem QDAontology em projetos de *e-Science* a mesma se mostrou adequada tendo em vista que a participação da equipe multidisciplinar se caracterizou pela atuação dos especialistas do domínio nas etapas iniciais de extração e formalização do conhecimento, deixando para os profissionais da Ciência da Computação a construção da ontologia e sua interoperabilidade com outras complementares. A aderência dos artefatos gerados ao longo do processo de desenvolvimento às características de qualidade da ontologia garantiu uma versão final evoluída e pronta para ser utilizada em aplicações científicas onde a troca semântica de informações é necessária. A flexibilidade da seleção do ambiente e das ferramentas para geração da documentação garantiu a sua adequação ao projeto de *e-Science*.

7 CONSIDERAÇÕES FINAIS

Uma ontologia é uma especificação de uma conceitualização [GRUBER, T. R. 1995], ou seja, é uma descrição formal dos conceitos e relações de um domínio. A construção de uma ontologia implica em adquirir o conhecimento do domínio sendo tratado e coletar as informações apropriadas que definam, com consistência, os termos usados formalmente para descrever tal domínio. Ela representa computacionalmente o conhecimento humano e permite o compartilhamento e o reuso desse conhecimento. Desta forma, as ontologias vêm sendo empregadas em diversos domínios e aplicações e tem como objetivo compartilhar um conhecimento comum de uma estrutura de informações entre pessoas ou aplicações.

O uso de ontologias em soluções computacionais tem passado por progressos significativos. Elas podem ser empregadas em um grande número de aplicações e com diversas finalidades. Aquisição do conhecimento, processamento da linguagem natural, e-commerce, integração e recuperação de informação, integração de dados, bioinformática, reuso de informação, são algumas das áreas onde é adotado o conceito de ontologias.

Assim como na Engenharia de Software são definidos modelos de processos para o desenvolvimento de softwares, no desenvolvimento de ontologias também se faz necessário a aplicação de uma metodologia para a sua construção. A Engenharia Ontológica ainda é uma área de pesquisa recente, e, portanto, devido à sua constante evolução, ainda não existem metodologias para o seu desenvolvimento que sejam largamente utilizadas e aceitas como padrão.

Como as ontologias podem ser desenvolvidas por vários grupos diferentes e pertencerem a um mesmo domínio, elas podem apresentar estruturas e padrões de nomenclatura de termos diferentes [SILVA, V. S. *et al.*, 2008]. Desta forma, são necessários mecanismos que garantam a interoperabilidade das ontologias, permitindo a integração das informações e a evolução das mesmas.

A correspondência semântica é uma solução para os problemas envolvendo a heterogeneidade. Estas correspondências podem ser utilizadas para várias tarefas, tais como a combinação de ontologias, alinhamento de ontologias, entre outras [SHVAIKO, P., EUZENAT, J. 2008].

Neste contexto, este trabalho propôs a abordagem intitulada QDAontology, para o desenvolvimento de ontologias em projetos de *e-Science*, com as etapas necessárias para desenvolvimento de ontologias científicas, juntamente com um mecanismo que promova a recuperação e integração das informações presentes em outras ontologias do mesmo domínio ou de domínios complementares. A proposta tem enfoque na aquisição do conhecimento e utiliza o mecanismo de alinhamento como técnica de interoperabilidade entre ontologias, incluindo ainda características de qualidade essenciais para se obter uma ontologia completa e correta em relação ao domínio que representa.

A abordagem proposta possibilitou a reengenharia da ontologia CelO bem como a sua evolução, estendendo dessa forma o seu domínio. Os resultados obtidos revelaram que as etapas que compõem o processo são adequadas para o desenvolvimento de ontologias e todos os artefatos gerados são suficientes para a elaboração de toda a documentação. Além disso, a definição de atributos de qualidade para ontologias se mostrou indispensável para a construção de uma ontologia correta. Por meio dos resultados obtidos, ficou claro que a interoperabilidade entre ontologias, a qual é realizada durante a etapa de Integração, é fundamental quando se deseja trocar informações com outras ontologias.

Foram apresentadas algumas das principais metodologias e métodos para o desenvolvimento de ontologias que apresentam características e propósitos diferentes. A abordagem proposta neste trabalho possui um diferencial em relação às outras metodologias, pois possui como uma de suas etapas a Integração, trabalha com equipes multidisciplinares e explicita características de qualidade passíveis de se alcançar em todas as etapas. Esse diferencial é muito importante quando se trabalha em projetos de *e-Science*, tendo vista que há necessidade de um processo abrangente para todos os domínios e que garanta a troca de informações qualificada. Isso fica ainda mais evidente quando a equipe que desenvolve a ontologia é multidisciplinar, ou seja, quando os participantes são oriundos de áreas diferentes, já que a visão de um domínio pode ser expressa por diversas maneiras. Apesar da abordagem ter sido desenvolvida para atender a *e-Science*, o processo pode ser especializado para uma ampla classe ontologias, para diferentes áreas de aplicação, tipos de equipes, níveis de competência e tamanhos de projetos.

O resultado mais significativo deste trabalho foi obtido por meio dos ciclos realizados na aplicação da abordagem. O primeiro ciclo permitiu não somente a elaboração de toda a documentação da ontologia CelO, mas através da execução da etapa de Integração, permitiu que a ontologia ficasse mais próxima de outras ontologias já consagradas por grupos internacionais. O segundo ciclo possibilitou a expansão do domínio da CelO, resultando na CelO - Human Disease.

Além das metodologias e métodos de desenvolvimento de ontologias, para a elaboração e execução da abordagem proposta, foi feita uma revisão da literatura sobre os diversos mecanismos de correspondência ontológica, medidas de similaridade e ferramentas que realizam o alinhamento entre ontologias. Além disso, uma pesquisa sobre as ontologias que poderiam ser utilizadas na etapa de Integração foi feita, e devido à sua importância perante o domínio biológico, algumas das ontologias pertencentes ao grupo OBO *Foundry* foram as escolhidas para esta tarefa. Como consequência, foi necessário um estudo sobre os conceitos relacionados às ontologias abordadas por esse grupo, além do estudo sobre a linguagem OBO adotada por eles.

O alinhamento entre ontologias foi o mecanismo adotado pelo fato de mesmo ser o mais adequado em sistemas onde o reuso e troca de informações são necessários. Contudo, para a concretização da interoperabilidade foi necessário o desenvolvimento de uma ferramenta, denominada A3O, que possibilitasse tal mecanismo, já que não estava disponível uma ferramenta capaz de realizar o alinhamento entre as linguagens OWL e OBO. Como medida de similaridade, a A3O adota técnicas baseadas no nome da entidade e compara o nome e hierarquia dos conceitos.

Neste contexto, podem ser destacadas as seguintes contribuições desse trabalho:

- Proposição de uma abordagem evolutiva para o desenvolvimento de ontologias em projetos de *e-Science*;
- Explicitação dos usuários que compõe a equipe multidisciplinar;
- Adoção da Integração como uma etapa constituinte do processo e não apenas como uma atividade;
- Atribuição de características de qualidade às etapas do processo de desenvolvimento de ontologias;

- Desenvolvimento e uma ferramenta para a realização do alinhamento entre ontologias descritas nas linguagens OWL e OBO;
- Disponibilização da CeLO e da Celo-Human Disease em padrões interoperáveis.

Como trabalhos futuros, é necessário adotar ferramentas para a atividade de gerência de configuração de ontologias, a fim de administrar as modificações ao longo do ciclo de vida, já que mudanças podem ser consideradas ao longo de sua evolução e para controlar os diversos artefatos produzidos por pessoas que trabalham em um mesma ontologia.

No contexto de infraestruturas para *e-Science* podemos destacar como trabalhos futuros o estudo de técnicas de alinhamento entre ontologias que melhorem a busca por serviços Web semânticos relacionados a um dado domínio.

Outro aspecto que pode ser explorado é a adoção de métricas em relação à atividade de qualidade, tendo em vista que as métricas são fundamentais para o gerenciamento, melhoria e desenvolvimento de processos e produtos.

Em relação à ferramenta de alinhamento entre ontologias, desenvolvida neste trabalho, no futuro pode-se melhorar o cálculo da medida de similaridade, acrescentando técnicas baseadas não somente pelo nome, mas também em relação a sua estrutura, extensão e semântica da entidade. Conseqüentemente, o alinhamento se estenderia às propriedades, indivíduos e todos os conceitos que compõem uma ontologia. E finalmente, agregar à ferramenta A3O, outros mecanismos de interoperabilidade.

REFERÊNCIAS

- API OBO. 2010. API OBO. Disponível em <http://oboedit.org/api/obo/>. Acesso em 05 jan. 2010.
- ASHBURNER, M. et al. Gene Ontology: tool for the unification of biology. 2000. In Nature America, nature genetics, volume 25. Disponível em <http://www.geneontology.org/cgi-bin/biblio.cgi> , Acesso em 19 ago. 2008
- BARD, J., RHEE, S. Y. AND ASHBURNER, M. An ontology for cell types. In Genome Biology 2005, 6:R21. Disponível em <http://genomebiology.com/2005/6/2/R21>. Acesso em 19 ago. 2008
- BRAGA, R. M. M. ; CAMPOS, F. C. A. . SBS-AGRO:Sistema de Busca Utilizando Mapeamentos Ontológicos e Retorno do Usuário. Revista Brasileira de Agroinformática, Viçosa, MG, 2004.
- BRANDÃO, A. A. F., LUCENA, C. J. P. Uma Introdução à Engenharia de Ontologias no contexto da Web Semântica. (2002) . *Technical Report* (PUC-RioInf. MCC29/02), Departamento de Informática, PUC-Rio, 2002
- BREITMAN, K. K., CASANOVA, M. A, TRUSZKOWSKI, W. “Ontology in Computer Science”. In: _____. Semantic Web – Concepts, Technologies and Applications. London: Springer, 2007.
- CAMPOS, M. L. A., CAMPOS, M. L.M., DÁVILA, A. M. R., GOMES, H. E., CAMPOS, L. M., LIRA, L. Aspectos Metodológicos no Reuso de Ontologias: um estudo a partir das anotações genômicas no domínio dos tripanosomatídeos. In RECIIS – R. Eletr. de Com. Inf. Inov. Saúde. Rio de Janeiro, v.3, n.1, p.64-75.
- CARDOSO, J. Semantic Web Services: Theory Tools and Applications. EUA: IGI Global, 2007. 372 p. ISBN 978-1-59904-047-9. Ebook.
- CELL TYPE. 2010. Cell Type Ontology. Disponível em <https://lists.sourceforge.net/lists/listinfo/obo-cell-type>. Acesso em 14 jan. 2010.
- CELLML. 2010. Cell Markup Language. Disponível em <http://www.cellml.org>. Acesso em 10 jan. 2010.
- CHENINE, M., KABILAN, V., LOZANO, M. G. A Pattern for Designing Distributed Heterogeneous Ontologies for Facilitating Application Interoperability, 3rd Open INTEROP Workshop On “Enterprise Modeling and Ontologies for Interoperability” (EMOI 2006), co-located with CAISE 2006. Disponível em

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.2394&rep=rep1&type=pdf>

CHOI, N., SONG, I., HAN, H. 2006. A Survey on Ontology Mapping. In SIGMOD Record, Vol. 35, No 3, Sep 2006, pp 34-41. DOI = [Http://doi.acm.org/10.1145/1168092.1168097](http://doi.acm.org/10.1145/1168092.1168097)

CORCHO, O., FERNÁNDEZ-LÓPEZ, M., GÓMEZ –PÉREZ, A., LÓPEZ-CIMA, A. Building legal ontologies with METHONTOLOGY and WebODE. In: Law and the Semantic Web, number 3369 in LNAI, 2005, Springer-Verlag.

CORCHO, O., FERNANDEZ-LOPEZ, M., GOMEZ-PEREZ, A. “Ontological Engineering: What are ontologies and how can we build them?”. In: CARDOSO, J. *Semantic Web Services: Theory, Tools and Applications*. New York: Information Science Reference. 2007. Cap. 3, p. 44-70.

CYC .2010. Cyc. Disponível em <http://www.cyc.com/>. Acesso em 12 jan. 2010.

DEGTYARENKO, K., MATOS, P., ENNIS, M., HASTINGS, J., ZBINDEN, M., MCNAUGHT, A., ALCÁNTARA, R., DARSOW, M., GUEDJ, M. AND ASHBURNER, M. “ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic Acids Res*”. 2007. 36, D344–D350. Disponível em http://nar.oxfordjournals.org/cgi/content/abstract/36/suppl_1/D344, acesso em 19 ago. 2008

DISEASE ONTOLOGY. 2010. Disponível em http://do-wiki.nubic.northwestern.edu/index.php/Main_Page. Acesso em 13 jan. 2010.

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

EHRIG, M., SURE, Y. (2005). Framework for Ontology Alignment and Mapping. Disponível em <http://oaei.inrialpes.fr/doc/intont2005proceedings.pdf#page=74>

EILBECK, K., LEWIS, S., MUNGALL, C. J., YANDELL, M., STEIN, L., DURBIN, R., ASHBURNER, M. (2005). The Sequence Ontology: A tool for the unification of genome annotations. *Genome Biology* (2005) 6:R44. Disponível em <http://genomebiology.com/2005/6/5/R44>. Acesso em 19 ago. 2008

FERRARA, A. Methods and Techniques for Ontology Matching and Evolution in Open Distributed Systems. 2003. Tese Ph.D. Università degli Studi di Milano, Milano, Italy, 2003

EUZENAT, J., SHVAIKO, P, *Ontology Matching*, Springer-Verlag, Berlin Heidelberg (DE), 2007.

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

FERNÁNDEZ LÓPEZ, M. *Overview Of Methodologies For Building Ontologies*. In *Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5)* Stockholm, Sweden, August 2, 199. (1999)

FERNÁNDEZ-LÓPEZ, M. GÓMEZ-PÉREZ, A. *Overview and analysis of methodologies for building ontologies*. *Knowledge Engineering Review (KER)*, v. 17, n. 2 (2002), pp. 129-156.

FERNÁNDEZ, M., GÓMEZ-PÉREZ, A., JURISTO, N. *METHONTOLOGY: From Ontological Art Towards Ontological Engineering*. 1997. In *AAAI Technical Report SS-97-06*. Disponível em <http://www.aaai.org/Papers/Symposia/Spring/1997/SS-97-06/SS97-06-005.pdf>

FMA. 2010. “Foundational Model of Anatomy ontology”. Disponível em <http://sig.biostr.washington.edu/projects/fm/AboutFM.html>. Acesso em 09 jan. 2010.

FREITAS, J. B. (2007). *SiSe: Medida de similaridade semântica entre ontologias em português*. Dissertação (mestrado) – Fac. De Informática PUCRS, Rio Grande do Sul, 2007

GENE ONTOLOGY. 2010. “The Gene Ontology”. Disponível em <http://www.geneontology.org/GO.doc.shtml?all>. Acesso em 09 jan, 2010.

GOMEZ PEREZ, A., BENJAMINS, V. R. *Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods (1999)*. Disponível em <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-18/1-gomez.pdf>

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

GUARINO, N. 1998. “Formal ontology and information systems”. 1998. In: *Proceedings of the First International Conference on Formal Ontologies in Information Systems, FOIS'98, Trento, Italy*, pp. 3-15.. Disponível em <http://www.loa-cnr.it/Papers/FOIS98.pdf>

GUARINO, N. 1997. “Understanding, building, and using ontologies: a commentary to using explicit ontologies”. 1997. *International Journal of Human and Computer*

Studies, v. 46, p. 293-310. Disponível em
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.54.2583>

HAENDEL, M. A., NEUHAUS, F., OSUMI-SUTHERLAND, D., MABEE, P.M., MEJINO JR, J. L. V., MUNGALL, C. J., AND SMITH, B CARO - The Common Anatomy Reference Ontology. 2006 Cap 16. Disponível em
http://www.bioontology.org/wiki/index.php/CARO:Main_Page . Acesso em 19 ago. 2008

HARRIS, M. A. et al. The Gene Ontology (GO) project in 2006. 2006. Nucleic Acids Res. 36, D322–D326. Disponível em <http://www.geneontology.org/cgi-bin/biblio.cgi> , Acesso em 19 ago. 2008

JANOWICK, K., MAUÉ, P. WILKES, M., SCHADE, S., SCHERER, F., BRAUN, M., DUPKE, S., KUHN, W. (2008). Similarity as a Quality Indicator in Ontology Engineering. In FOIS 2008 - 5th international conference on formal ontology in information systems

JARRAR, M. Towards Methodological Principles for Ontology Engineering. 2005. Tese (Doutorado) – Vrije Universiteit Brussel.

JENA. 2010. Jena 2 Ontology API. Disponível em
<http://jena.sourceforge.net/ontology>. Acesso em 20 jan. 2010.

LAMBRIX, P., TAN, H., JAKONIENE, V., STRÖMBÄCK, L. “Biological Ontologies”. In: *Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences*. New York: Information Science Reference. 2007. Cap. 4, p. 85-99.

LAUSCHNER, T; Especificação formal de agentes para Web; Projeto pesquisa para doutorado PUC-RJ, 2005

LEONTIS, N. B. *et al.* The RNA Ontology Consortium: An open invitation to the RNA community. 2006. Disponível em
<http://rnajournal.cshlp.org/cgi/content/abstract/12/4/533> , Acesso em 19 ago. 2008.

LUDASCHER, B., ALTINTAS, I., BERKLEY, C., HIGGINS, D., JAEGER-FRANK, E., JONES, M., LEE, E., TAO, J., AND ZHAO, Y. (2006). Scientific workflow management and the Kepler system. *Concurrency and Computation: Practice and Experience*, 18(10):1039–1065.

MAEDCHE, A., STAAB, S. (2002). Measuring Similarity between Ontologies. Disponível em <http://www.aifb.uni-karlsruhe.de/WBS/sst/Research/Publications/ekaw2002-compare.pdf>. Acesso em 12 nov. 2009.

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

MATOS, E. E., MENDES, L. F., CAMPOS F., BRAGA, R. (2009) "ASOW-Science: a service oriented framework to support e-Science Applications". In: IEEE IRI 2009, Las Vegas. IEEE Proceedings of Internacional on Information Reuse and Integration, 2009.

MATOS, E. E., NOCELLI, C., BRAGA, R., CAMPOS F. (2007) "MathWS: Broker de Serviços Web para e-Science". Disponível em <http://www.lbd.dcc.ufmg.br/bdbcomp/servlet/Trabalho?id=7301>.

NATALE, D. A., ARIGHI, C. N., BARKER, W., BLAKE, J., CHANG, T., HU, Z. Z., LIU, H., SMITH, B., WU, C. H. (2007) Framework for a protein ontology. BMC bioinformatics, 8 Suppl 9(Suppl 9):S1+, 2007. Disponível em <http://www.biomedcentral.com/1471-2105/8/S9/S1>. Acesso em 19 ago. 2008.

NOY, N., MCGUINNESS, D. (2001). Ontology Development 101 – A guide to creating your first ontology. KSL Technical Report, Stanford University, Stanford, CA, USA.

NOY, N. F.; MUSEN, M. A. The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping. Stanford University. International Journal of Human-Computer Studies v. 59, a. 6, p. 983-1024, 2003.

OBO-Edit. 2010. "The OBO Ontology Editor". Disponível em <http://oboedit.org/>. Acesso em 11 jan. 2010.

OBO-FORMAT. 2010. "The OBO Flat File Format Specification, version 1.2". Disponível em http://www.geneontology.org/GO.format.obo-1_2.shtml. Acesso em 04 ja. 2010.

OBO Foundry. 2010. "The Open Biomedical Ontologies". Disponível em <http://www.obofoundry.org/>. Acesso em 20 jan. 2010.

OKBC. 2010. "Open Knowledge Base Connectivity Working Group". Disponível em <http://www.ai.sri.com/~okbc/>.

OntoClean. 2010. OntoClean Central. Disponível em <http://www.ontoclean.org>. Acessado em 10 de março de 2010.

OWL. 2010. "OWL Web Ontology Language Overview". Disponível em <http://www.w3.org/TR/owl-features/>. Acesso em 03 jan. 2010.

PALAZZI, D. ; SILVA, L. ; Mendes, L. F. ; Gaspar, W. ; MATOS, E. ; CAMPOS, F. C. A. ; Braga, Regina . Uso de Ontologias em Projetos de e-Science. In: II Seminário de Pesquisa em Ontologia, 2009, IME- Instituto Militar de Engenharia, Rio de Janeiro, RJ.

PATO. 2010. "Pato:about". Disponível em <http://www.bioontology.org/wiki/index.php/PATO>About>. Acesso em 10 jan. 2010.

PFLEEGER, S. L. Engenharia de Software: teoria e prática. 2004. 2^o ed. São Paulo: Prentice Hall.

PINTO, H. S., MARTINS, J. P. Ontologies: How can They be Built? In Knowledge and Information Systems (2004) 6: 441–464

PRESSMAN, R. S. Engenharia de Software. 2006. 6^a ed. São Paulo: McGraw-Hill.

PRO. 2010. "The Protein Ontology". Disponível em <http://pir.georgetown.edu/pro/>. Acesso em 10 jan. 2010.

PROTÉGÉ. 2010. "The Protégé Ontology Editor and Knowledge Acquisition System". Disponível em <http://protege.stanford.edu>. Acesso em 08 jan. 2010

RDF. 2010. "Resource Description Framework". Disponível em <http://www.w3.org/TR/rdf-schema/>. Acesso em 16 jan. 2010.

RDF-S. 2010. "RDF Schema". Disponível em <http://www.w3.org/RDF/>. Acesso em 16 jan. 2010.

ROC. 2010. "RNA Ontology Consortium". Disponível em <http://roc.bgsu.edu/>. Acesso em 16 jan. 2010.

ROSSE, C. AND MEJINO, J. L. V. *A reference ontology for biomedical informatics: the Foundational Model of Anatomy*. In *Journal of Biomedical Informatics* 36 (2003) 478–500. Disponível em <http://sigpubs.biostr.washington.edu/archive/00000135/>. Acesso em 19 ago. 2008

ROSSE, C. AND MEJINO, J. L. V. The Foundational Model of Anatomy Ontology, in Burger, A and Davidson, D and Baldock, R. 2007. Eds. *Anatomy Ontologies for Bioinformatics: Principles and Practice*, pages pp. 59-117. Springer. Disponível em <http://sigpubs.biostr.washington.edu/archive/00000204/>. Acesso em 19 ago. 2008

SANTOS, N. ; CAMPOS, F. C. A. ; BRAGA, R. M. M. . Digital Libraries and Ontology. In: Yin-Leng Theng; Schubert Foo; Dion Hoe-Lian Goh; Jin-Cheon Na.. (Org.).

Handbook of Research on Digital Libraries: Design, Development, and Impact. Hershey, Pensilvania:: Idea Group Inc, 2008, v. 1.

SHVAIKO, P, EUZENAT, J. Tem Challenges for Ontology Matching. 2007. In Proceedings of The 7th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE), 2008

SILVA, D. L., SOUZA, R. R., ALMEIDA, M. B. Ontologias e vocabulário controlados: Comparação de metodologias para a construção. 2008. Disponível em <http://revista.ibict.br/index.php/ciinf/article/viewFile/1047/765>

SILVA, L. ; Braga, Regina ; Campos, Fernanda . A FRAMEWORK FOR SEMANTIC COMPOSITION OF SCIENTIFIC WORKFLOWS. In: IADIS International Conference WWW/Internet, 2009, 2009, Roma. Proceedings of IADIS International WWW/Internet, 2009. Roma, 2009.

SILVA, N. A. P. MULTI-DIMENSIONAL SERVICE-ORIENTED ONTOLOGY MAPPING. 2004. Tese (PhD.) - Universidade de Trás-os-Montes e Alto Douro, Vila Real, Portugal. 2004. Disponível em <http://www.dei.isep.ipp.pt/~nsilva/R&D/PhD/PhDThesis.pdf>

SILVA, V. S., SILVA, J. C. P., CAMPOS, M. L. M.: Alinhamento de Ontologias através do algoritmo de Alinhamento Local de Caminhos. In: Seminário de Pesquisa em Ontologia no Brasil, Niterói, Rio de Janeiro, Brasil. (2008).

SMITH, B. Ontology (Science). 2008. In Nature Precedings

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

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

SURE, Y., STAAB, S., STUDER, R. (2002). Methodology for development and employment of ontology based knowledge management applications. *SIGMOD Rec.* 31, 4 (December 2002), 18-23. DOI=10.1145/637411.637414 <http://doi.acm.org/10.1145/637411.637414>

TEMPICH, C. Ontology Engineering and Routing in Distributed Knowledge Management Applications. 2006. Tese (Doutorado) - Universität Fridericiana zu Karlsruhe, Alemanha, 2006.

USCHOLD, M., KING, M. Towards a Methodology for a Building Ontologies. 1995. In Workshop on Basic Ontological Issues in Knowledge Sharing.

XML. 2010. "Extensible Markup Language". Disponível em <http://www.w3.org/XML/>. Acesso em 9 jan 2010.

APÊNDICE

Apêndice 1

Artefatos gerados durante a aplicação da abordagem QDAontology

A1.1. Glossário de termos produzido na etapa de Conceitualização do ciclo de reengenharia da CeLO

Name	Description
<i>CeLO</i>	<i>Provides representation of biological models with a logic language and models described in CellML, and allows the creation of a repository of models and promote the reuse of existing components.</i>
<i>DomainEntity</i>	This class and its sub-classes define the terms that will be used as a vocabulary shared by researchers in creating models. These terms have an associated semantics, allowing the description of the model at a higher level.
<i>ModelEntity</i>	This class and its sub-classes define the concepts that will be used in the biological model represented. The goal is to have a high-level description, with reference the CellML model for the simulation questions.
<i>SIEntity</i>	This class and its sub-classes, which have a generic application, define a dictionary of units and quantities that are used in conjunction with domain concepts and variables of the models.
<i>BioChemicalEntity</i>	Refers to the concepts in the field of Biochemical.
<i>BiologicalEntity</i>	Refers to the concepts in the field of Biology.
<i>ChemicalEntity</i>	Refers to the concepts in the field of Chemical.
<i>Protein</i>	A molecule composed of polymers of amino acids joined together by peptide bonds.
<i>CellElement</i>	Refers to the physical structure of the cell (CellStructure) and the processes associated with the electrophysiology.
<i>CellType</i>	Kind of cells.
<i>CellularProcess</i>	Any process that is carried out at the cellular level, but not necessarily restricted to a single cell.
<i>StructuralCell</i>	A cell that provides structural support (physical).
<i>GatingProcess</i>	Process associated to cardiac electrophysiology of the various types of "gating".
<i>MembraneTransport</i>	Refers to the phenomenon of transport through the membrane.
<i>PolarizationProcess</i>	Process associated at cardiac electrophysiology of the phenomenon of depolarization, polarization and repolarization.
<i>CellPart</i>	Any constituent part of a cell, the basic structural and functional unit of all organisms.
<i>CellSpaceLocation</i>	Refers to the space location of the cell, for exemple, extracellular, intracellular, diadic.
<i>Myocyte</i>	Cells of muscles.
<i>CardiacMuscleCell</i>	Cells of cardiac muscle.
<i>ChemicalCompound</i>	Refers to chemical compounds.
<i>ChemicalElement</i>	Refers to chemical elements.
<i>ChemicalObject</i>	A generic concept for the definition of atoms, ions and molecules.
<i>Nucleotide</i>	Nucleotides are molecules which comprise the structural units of RNA and DNA. It's play central roles in metabolism. In that capacity, they serve as sources of chemical energy, participative in cellular signaling and are incorporated into important cofactors of enzymatic reactions.
<i>ModelGrounding</i>	Specifies the "logical" model associated in the case of a simulation.
<i>ModelObject</i>	Groups the objects that make up a model. It directly related to the type of underlying model.

<i>ModelProcess</i>	Indicates how the model can be used, ie, what are the input parameters (ModelParameterIn) and output (ModelParameterOut) that are associated with the interface (ModelInterface) model.
<i>ModelProfile</i>	To inform what is the model: what are the components of the model, if is associated with a specific compartment of the cell and what are the biological entities (instance BiologicalEntity) associated with the model.
<i>ModelService</i>	Provides a way to organize the vision of a model as a service.
<i>ModelType</i>	Generic class used only to characterize the biological problem addressed by the model and type of mathematical model used.
<i>CellMLModel</i>	Used to store the URI of the CellML model associated.
<i>Compartment</i>	Used to share models.
<i>Component</i>	Represents the components of the model.
<i>Element</i>	Elements used to represent a model.
<i>Equation</i>	Relaciona as equações que implementam matematicamente o modelo, como strings expressas em MathML.
<i>Model</i>	Represents the model.
<i>Variable</i>	Represents the variables of the model.
<i>ExternalComponent</i>	Store URI references to external models.
<i>InternalComponent</i>	Components described in the model itself.
<i>ComponentVariable</i>	Variables of each component that makes up the model.
<i>DomainVariable</i>	Combines the variables of the model with the concepts expressed by classes DomainEntity.
<i>ModelVariable</i>	Describes how the variables involved in the equations of model.
<i>InterfaceVariable</i>	Variables used in the interface of component.
<i>LocalVariable</i>	Local variables to the component.
<i>ActionPotential</i>	Represents the action potential.
<i>IonChannelGate</i>	Represents the ion channel.
<i>PotassiumIonChannel</i>	Represents the potassium ion channel.
<i>SodiumIonChannel</i>	Represents the sodium ion channel.
<i>AlgebraicVariable</i>	Algebraic variable.
<i>DependentVariable</i>	Dependent variable.
<i>IndependentVariable</i>	Independent variable.
<i>ParameterVariable</i>	Parameters of the equations.
<i>ModelInterface</i>	Parameters associated with the interface model.
<i>ModelParameter</i>	Parameters constituent of the model.
<i>ModelParameterIn</i>	Input parameter of the model.
<i>ModelParameterOut</i>	Output parameter of the model.
<i>BiologicalModelType</i>	Type of biological model.
<i>MathematicalModelType</i>	Type of mathematical model.
<i>PhysiologicalModel</i>	Models in Physiology.
<i>ElectrophysiologicalModel</i>	Models in Electrophysiology.
<i>MathModelDeterministicType</i>	Type of deterministic mathematical model.
<i>MathModelDiscretizationType</i>	Type of mathematical model discretization.
<i>MathModelLinearityType</i>	Type of linear mathematical model.
<i>MathModelTimeDependencyType</i>	Type of mathematical dependent on time.
<i>DeterministicModel</i>	Represents a deterministic model.
<i>StochasticMode</i>	Represents a stochastic model.

<i>ContinuousModel</i>	Represents a continuous model.
<i>DiscreteModel</i>	Represents a discrete model.
<i>LinearModel</i>	Represents a linear model.
<i>NonLinearModel</i>	Represents a non linear model.
<i>StationaryModel</i>	Represents a stationary model.
<i>TimeDependentModel</i>	Represents a time-dependent model.
<i>Quantity</i>	Quantity measured by a unit of SI.
<i>Unit</i>	Units used to measure the quantities.
<i>BaseQuantity</i>	Basic Quantity defined by SI.
<i>DerivedQuantity</i>	Derived quantity from the basic units of SI.
<i>UserDefinedQuantity</i>	Derived quantity defined from user.
<i>DefinedUnit</i>	Pre-defined units (not listed in SI).
<i>StandardUnit</i>	Standard units defined in SI.
<i>UserDefinedUnit</i>	Units defined by users that do not use biological models.
<i>SIBaseUnit</i>	Basic units defined in SI.
<i>SIDerivedUnit</i>	Derived units from the basic units.

A1.2 Dicionário de conceitos produzido na etapa de Conceitualização do ciclo de reengenhria da CeIO

Name	Instances	Class attributes	Relations
<i>CeIO</i>		<i>DomainEntity</i> <i>ModelEntity</i> <i>SIEntity</i>	<i>hasDescription</i> <i>hasName</i>
<i>DomainEntity</i>		<i>BioChemicalEntity</i> <i>BiologicalEntity</i> <i>ChemicalEntity</i>	
<i>ModelEntity</i>		<i>ModelGrounding</i> <i>ModelObject</i> <i>ModelProcess</i> <i>ModelProfile</i> <i>ModelService</i> <i>ModelType</i>	
<i>SIEntity</i>		<i>Quantity</i> <i>Unit</i>	
<i>BioChemicalEntity</i>		<i>Protein</i>	
<i>BiologicalEntity</i>		<i>CellElement</i> <i>CellType</i>	
<i>ChemicalEntity</i>		<i>ChemicalCompound</i> <i>ChemicalElement</i> <i>ChemicalObject</i>	
<i>Protein</i>			
<i>CellElement</i>		<i>CellularProcess</i> <i>StructuralCell</i>	
<i>CellType</i>		<i>Myocyte</i>	
<i>CellularProcess</i>		<i>GatingProcess</i> <i>MembraneTransport</i> <i>PolarizationProcess</i>	<i>hasDomainEntity</i> <i>hasMeasure</i>
<i>StructuralCell</i>		<i>CellPart</i> <i>CellSpaceLocation</i>	<i>contains</i> <i>isContainsIn</i>
<i>GatingProcess</i>			
<i>MembraneTransport</i>			
<i>PolarizationProcess</i>			
<i>CellPart</i>	<i>Buffer</i> <i>Channel</i> <i>Exchanger</i> <i>Gate</i> <i>Membrane</i> <i>Mitochondria</i> <i>Myoplasm</i> <i>Nucleus</i> <i>Pump</i> <i>SarcoplasmicReticulum</i>		
<i>CellSpaceLocation</i>	<i>DiadicSpace</i> <i>ExtraCellularspace</i> <i>IntraCellularspace</i>		
<i>Myocyte</i>		<i>CardiacMuscleCell</i>	
<i>CardiacMuscleCell</i>	<i>AtrialCell</i> <i>AtrioVentricularNode</i> <i>His-PurkinjeFibre</i> <i>SinoAtrialNode</i> <i>VentricularCell</i>		
<i>ChemicalCompound</i>		<i>Nucleotide</i>	
<i>ChemicalElement</i>	<i>Calcium</i> <i>Chlorid</i>		<i>hasChemicalSymbol</i>

	Potassium Sodium		
ChemicalObject	Atom Ion Molecule		
Nucleotide			
ModelGrounding		CellMLModel	
ModelObject		Compartment Component Element Equation Model Variable	
ModelProcess		ModelInterface ModelParameter	
ModelProfile			composedOf hasBiologicalEntity modelCompartment
ModelService			describedBy presents refersTo
ModelType		BiologicalModelType MathematicalModelType	
CellMLModel			hasURI
Compartment			
Component		ExternalComponent InternalComponent	connectsTo encapsules hasCellElement hasDomainEntity
Element			
Equation			hasMath
Model			
Variable		ComponentVariable DomainVariable ModelVariable	hasDomainEntity hasMeasure hasVariableUnit hasDetail hasInitialValue
ExternalComponent			hasComponentURI
InternalComponent			hasEquation hasVariable hasDetail
ComponentVariable		InterfaceVariable LocalVariable	
DomainVariable		ActionPotential IonChannelGate PotassiumIonChannel SodiumIonChannel	
ModelVariable		AlgebraicVariable DependentVariable IndependentVariable ParameterVariable	
InterfaceVariable			isInterfaceVariableOf hasPrivateInterface hasPublicInterface
LocalVariable			
ActionPotential			
IonChannelGate			
PotassiumIonChannel			

<i>SodiumIonChannel</i>			
<i>AlgebraicVariable</i>			
<i>DependentVariable</i>			
<i>IndependentVariable</i>			
<i>ParameterVariable</i>			
<i>ModelInterface</i>			<i>hasInterface</i>
<i>ModelParameter</i>		<i>ModelParameterIn</i> <i>ModelParameterOut</i>	
<i>ModelParameterIn</i>			
<i>ModelParameterOut</i>			
<i>BiologicalModelType</i>		<i>PhysiologicalModel</i>	
<i>MathematicalModelType</i>		<i>MathModelDeterministicType</i> <i>MathModelDiscretizationType</i> <i>MathModelLinearityType</i> <i>MathModelTimeDependencyType</i>	
<i>PhysiologicalModel</i>		<i>ElectrophysiologicalModel</i>	
<i>ElectrophysiologicalModel</i>	<i>HodgkinHuxleyModel</i> <i>MarkovModel</i> <i>MulticompartmentModel</i> <i>NetworkModel</i> <i>SynapticModel</i>		
<i>MathModelDeterministicType</i>		<i>DeterministicModel</i> <i>StochasticModel</i>	
<i>MathModelDiscretizationType</i>		<i>ContinuousModel</i> <i>DiscreteModel</i>	
<i>MathModelLinearityType</i>		<i>LinearModel</i> <i>NonLinearModel</i>	
<i>MathModelTimeDependencyType</i>		<i>StationaryModel</i> <i>TimeDependentModel</i>	
<i>DeterministicModel</i>			
<i>StochasticModel</i>			
<i>ContinuousModel</i>			
<i>DiscreteModel</i>			
<i>LinearModel</i>			
<i>NonLinearModel</i>			
<i>StationaryModel</i>			
<i>TimeDependentModel</i>			
<i>Quantity</i>		<i>BaseQuantity</i> <i>DerivedQuantity</i> <i>UserDefinedQuantity</i>	<i>isMeasuredBy</i>
<i>Unit</i>		<i>DefinedUnit</i> <i>StandardUnit</i> <i>UserDefinedUnit</i>	<i>Measures</i> <i>hasSymbol</i>
<i>BaseQuantity</i>	<i>ElectricCurrent</i> <i>Length</i> <i>LuminousIntensity</i> <i>Mass</i> <i>Substance</i> <i>Temperature</i> <i>Time</i>		
<i>DerivedQuantity</i>	<i>Acceleration</i> <i>Area</i> <i>Capacitance</i> <i>Density</i> <i>ElectricConductance</i> <i>ElectricResistance</i> <i>ElectricInductance</i>		

	<i>MagneticFlux Velocity Voltage Volume</i>		
<i>UserDefinedQuantity</i>	<i>ADimensional CapacitancePer Area Concentration ConcentrationPerTime Conductivity EletricChargePerAmount EletricChargePerConcentration EletricConductancePerArea EletricConductancePerCapacita nce EletricCurrentPerCapacitance EletricCurrentPerConcentration EletricCurrentDensity MolarEntropy PerConcentration PerLength PerTime PerVoltage PerVoltageTime VoltagePerTime</i>		
<i>DefinedUnit</i>	<i>dimensionless</i>		
<i>StandardUnit</i>		<i>SIBaseUnit SIDerivedUnit</i>	
<i>UserDefinedUnit</i>	<i>cm2 concentration_units coulomb_per_millimole coulomb_per_imole faradays_constants_units gas_constants_units joule_per_kilomole_kelvin joule_per_mole_kelvin mM mM_per_ms mS_per_mm2 mV mV+per_ms microA_per_cm2 microF microF_per_cm2 micrometer micrometre3 milliS_per_cm2 millimolar millimolar_per_millisecond millisecond millivolt mm2 mm_per_ms ms nanoA nanoA_per_millimolar nanoS_per_picoF per_concentration_units per_mV per_mV_ms per_millisecond per_millivolt per_millivolt_millisecond per_mm per_ms picoA picoA_per_picoF uA_per_mm2 uF_per_mm2</i>		
<i>SIBaseUnit</i>			
<i>SIDerivedUnit</i>			

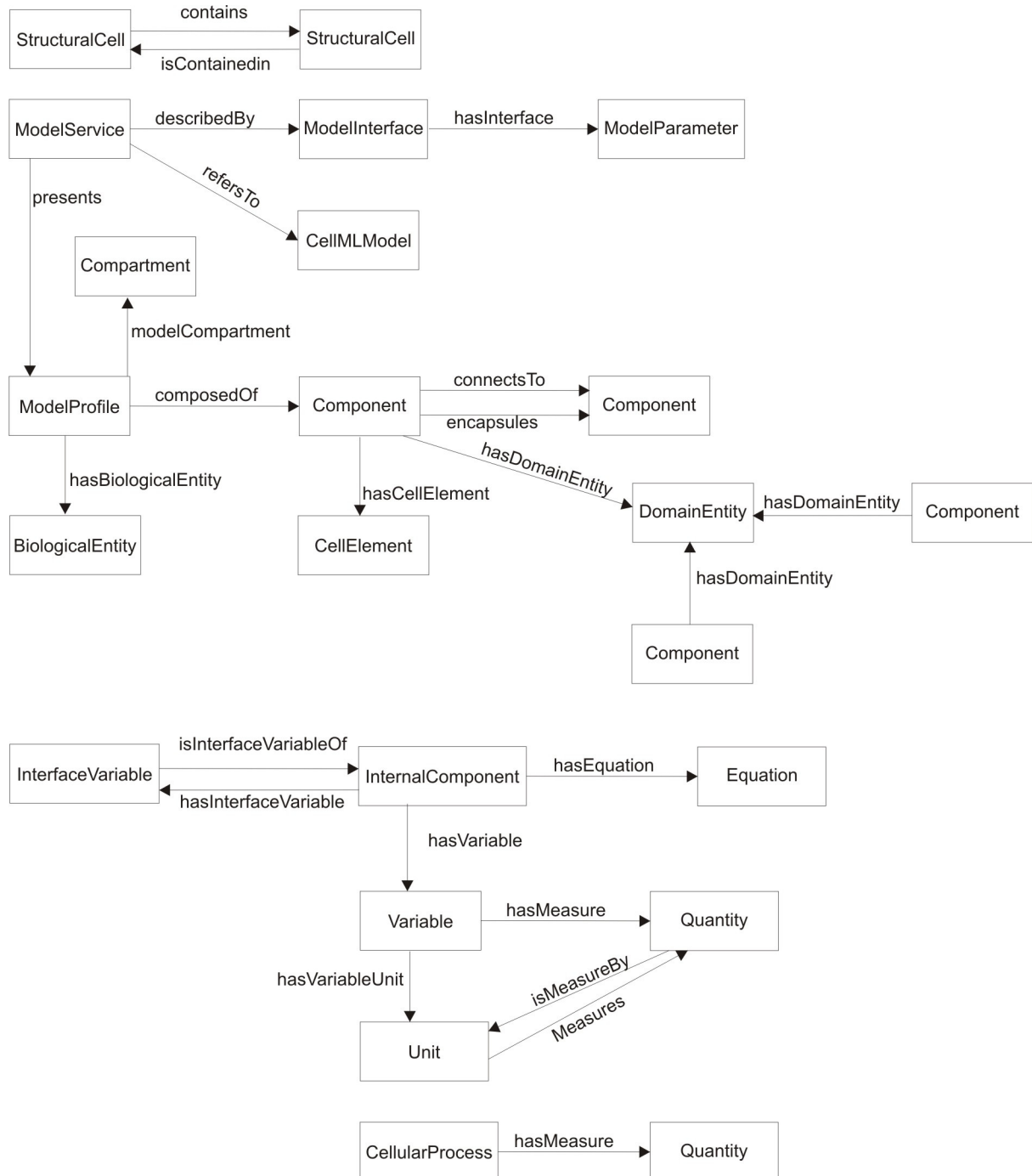
A1.3 Árvore de classificação dos conceitos produzido na etapa de Formalização do ciclo de reengenharia da CeO

CeO

- DomainEntity
 - BioChemicalEntity
 - Protein
 - BiologicalEntity
 - CellElement
 - CellularProcess
 - GatingProcess
 - MembraneTransport
 - PolarizationProcess
 - StructuralCell
 - CellPart
 - CellSpaceLocation
 - CellType
 - Myocyte
 - CardiacMuscleCell
 - ChemicalEntity
 - ChemicalCompound
 - Nucleotide
 - ChemicalElement
 - ChemicalObject
- ModelEntity
 - ModelGrounding
 - CellMLModel
 - ModelObject
 - Compartment
 - Component
 - ExternalComponent
 - InternalComponent
 - Element
 - Equation
 - Model
 - Variable
 - ComponentVariable
 - InterfaceVariable
 - LocalVariable
 - DomainVariable
 - ActionPotential
 - IonChannelGate
 - PotassiumIonChannel
 - SodiumIonChannel
 - ModelVariable
 - AlgebraicVariable
 - DependentVariable
 - IndependentVariable
 - ParameterVariable
 - ModelProcess
 - ModelInterface
 - ModelParameter
 - ModelParameterIn
 - ModelParameterOut
 - ModelProfile
 - ModelService
 - ModelType
 - BiologicalModelType
 - PhysiologicalModel
 - ElectrophysiologicalModel
 - MathematicalModelType
 - MathModelDeterministicType
 - DeterministicModel
 - StochasticMode
 - MathModelDiscretizationType
 - ContinuousModel

- DiscreteModel
- MathModelLinearityType
 - LinearModel
 - NonLinearModel
- MathModelTimeDependencyType
 - StationaryModel
 - TimeDependentModel
- SIEntity
 - Quantity
 - BaseQuantity
 - DerivedQuantity
 - UserDefinedQuantity
 - Unit
 - DefinedUnit
 - StandardUnit
 - SIBaseUnit
 - SIDerivedUnit
 - UserDefinedUnit

A1.4 Diagrama de relacionamentos dos conceitos produzido na etapa de Formalização do ciclo de reengenharia da CeIO



A1.5 Representação gráfica da ontologia produzida na etapa de Integração do ciclo de reengenharia da CeLO



A1.6. Glossário de termos produzido na etapa de Conceitualização do ciclo de expansão da CELO em CELO – Human Disease

Name	Description
<i>Acanthamoeba infection</i>	<i>Acanthamoeba</i> protozoa causes three types of illness involving the eye (keratitis), the brain and spinal cord (Granulomatous Amebic Encephalitis), and infections that can spread from an entry point to the entire body (disseminated infection).
<i>African trypanosomiasis</i>	Is also known as sleeping sickness, caused by the protozoa <i>Trypanosoma brucei rhodesiense</i> (East African trypanosomiasis) and <i>Trypanosoma brucei gambiense</i> (West African trypanosomiasis), which are carried by the tsetse fly.
<i>Amebiasis</i>	Refers to infection caused by the amoeba parasite protozoa <i>Entamoeba histolytica</i> .
<i>Angiostrongyliasis</i>	Is an infection caused by a nematode (roundworm) from the <i>Angiostrongylus</i> genus.
<i>Anisakiasis</i>	Is a zoonotic disease caused by the ingestion of larval nematodes, <i>Anisakis simplex</i> and <i>Pseudoterranova decipiens</i> , in raw seafood dishes such as sushi, sashimi, ceviche, and pickled herring.
<i>Ascariasis</i>	Is a human disease caused by the parasitic roundworm <i>Ascaris lumbricoides</i> , a worm that lives in the small intestine. Related with sanitation and hygiene-related Diseases.
<i>Babesiosis</i>	Is a malaria-like parasitic disease caused by the protozoa <i>Babesia</i> . Human babesiosis is uncommon, but reported cases have risen recently because of expanded medical awareness.
<i>Bacterial Diseases</i>	Infectious diseases caused by pathogenic bacteria. Examples of bacterial diseases: tuberculosis (<i>Mycobacterium tuberculosis</i>), pneumonia (which can be caused by <i>Streptococcus</i> and <i>Pseudomonas</i>) and foodborne illnesses (caused by bacteria such as <i>Shigella</i> , <i>Campylobacter</i> and <i>Salmonella</i>). Pathogenic bacteria also cause infections such as tetanus, typhoid fever, diphtheria, syphilis and leprosy.
<i>Balantidiasis</i>	Is a protozoan infection caused by infection with <i>Balantidium coli</i> .
<i>Capillariasis</i>	Caused by the nematode (roundworm) <i>Capillaria</i> species. <i>C. philippinensis</i> causes human intestinal capillariasis, <i>C. hepatica</i> , which causes in humans hepatic capillariasis, and <i>C. aerophila</i> , which causes in humans pulmonary capillariasis.
<i>Cell</i>	The basic structural and functional unit in people and all living things.
<i>Chagas Disease</i>	Is also referred to as American trypanosomiasis, caused by the parasite <i>Trypanosoma cruzi</i> , which is transmitted to animals and people by insect vectors that are found only in the Americas.
<i>Clonorchiasis</i>	Is an infectious disease caused by the Chinese liver fluke, <i>Clonorchis sinensis</i> .
<i>Cryptosporidiosis</i>	Also known as crypto, is a diarrheal disease caused by microscopic parasites of the genus <i>Cryptosporidium</i> , a protozoan parasite in the phylum Apicomplexa. It affects the intestines of mammals and is typically an acute short-term infection. In immunocompromised individuals, such as AIDS patients, the symptoms are particularly severe and often fatal.
<i>Cutaneous Larva Migrans</i>	Or "creeping eruption," a skin condition typically caused by dog or cat hookworm larvae of the genus <i>Ancylostoma</i> .
<i>Cyclosporiasis</i>	Caused by a microscopic parasite, <i>Cyclospora</i> .
<i>Cysticercosis</i>	Is an infection caused by the pork tapeworm, <i>Taenia solium</i> . Infection occurs when the tapeworm larvae enter the body and form cysticerci (cysts). When cysticerci are found in the brain, the condition is called neurocysticercosis.
<i>Diphyllobothriasis</i>	Is the infection caused by a tapeworm <i>Diphyllobothrium</i> , which can cause <i>Diphyllobothriasis</i> in humans through consumption of raw or undercooked fish. The principal species causing <i>diphyllobothriosis</i> is <i>Diphyllobothrium latum</i> , known as the broad or fish tapeworm, or broad fish tapeworm.
<i>Disease</i>	A condition of the living animal or plant body or of one its parts that impairs normal functioning. Illness and sickness are generally used as synonyms for disease.
<i>Dracunculiasis</i>	More commonly known as Guinea worm disease, <i>Dracontiasis</i> or <i>Medina Worm</i> , caused by the nematode <i>Dracunculus medinensis</i> . In areas stagnant water sources may still host microscopic, fresh-water arthropods known as copepods, which carry the larvae of the Guinea worm.
<i>Ectoparasite</i>	A classification of animals that includes those with hard, segmented bodies and jointed appendages, such as insects. Ectoparasites are usually arthropods which parasitize the skin.
<i>Enterobiasis</i>	Also called as pinworm infection, is caused by the roundworm <i>Enterobius vermicularis</i> .
<i>Fascioliasis</i>	Caused by the trematodes <i>Fasciola hepatica</i> and <i>Fasciola gigantica</i> , parasites of herbivores that can infect humans accidentally.
<i>Fungal diseases</i>	Caused by several fungi species, are called mycoses. Examples of some fungal diseases: aspergillosis, blastomycosis, candidiasis, coccidioidomycosis, cryptococcosis, histoplasmosis, and sporotrichosis.
<i>Giardiasis</i>	Is a diarrheal illness caused by a microscopic parasite, <i>Giardia intestinalis</i> (also known as <i>Giardia lamblia</i> or <i>Giardia duodenalis</i>).
<i>Helminths</i>	A large, multicellular organism that is generally visible to the naked eye in its adult stages. Helminths can be free-living or parasitic.
<i>Hookworm infection</i>	Hookworm is an intestinal parasite of humans that causes mild diarrhea and abdominal pain.

<i>Human Disease</i>	<i>Is an abnormal condition of an organism human that impairs bodily functions, associated with specific symptoms and signs.</i>
<i>Hydatidosis</i>	<i>Results from infection by tapeworm larvae of the genus Echinococcus - notably E. granulosus, E. multilocularis, E. vogeli and E. oligarthrus.</i>
<i>Hymenolepiasis</i>	<i>Is caused by two cestodes (tapeworm) species, Hymenolepis nana (the dwarf tapeworm) and Hymenolepis diminuta (rat tapeworm).</i>
<i>Infectious Disease</i>	<i>Is a clinically evident disease resulting from the presence of pathogenic microbial agents, including pathogenic viruses, pathogenic bacteria, fungi, protozoa, multicellular parasites, and aberrant proteins known as prions.</i>
<i>Visceral Larva Migrans</i>	<i>Is a zoonotic (animal to human) infection caused by the parasitic roundworms commonly found in the intestine of dogs (Toxocara canis) and cats (T. cati).</i>
<i>Leishmaniasis</i>	<i>Is a vector-borne disease that is transmitted by sandflies and caused by obligate intracellular protozoa of the genus Leishmania.</i>
<i>Malaria</i>	<i>Is a mosquito-borne disease caused by a parasite protozoa, Plasmodium.</i>
<i>Microsporidiosis</i>	<i>Also known as microsporidiasis, is caused by infection with microsporidia, which are obligately intracellular, spore-forming parasites that belong to the phylum Microspora and the order Microsporida.</i>
<i>Naegleria infection</i>	<i>Naegleria is an ameba commonly found in the environment in water and soil. Infection of humans is rare but serious, responsible for the primary amebic meningoencephalitis.</i>
<i>Onchocerciasis</i>	<i>Also called River Blindness, is caused by a parasitic worm, Onchocerca volvulus, and is spread by the bite of an infected blackfly.</i>
<i>Paragonimiasis</i>	<i>Is a food-borne parasitic infection caused by the trematodes (flukes) of the genus Paragonimus.</i>
<i>Parasitic disease</i>	<i>Diseases caused by parasite (any organism that lives in or on another organism without benefiting the host organism; commonly refers to pathogens, most commonly in reference to protozoans, helminthes and extparasites).</i>
<i>Protozoa</i>	<i>One-celled organism microscopic that can be free-living or parasitic in nature. They are able to multiply in humans which contributes to their survival and also permits serious infections to develop from just a single organism.</i>
<i>Schistosomiasis</i>	<i>Is a disease caused by parasitic worms, Schistosoma species.</i>
<i>Strongyloidiasis</i>	<i>Caused by the nematode (roundworm) Strongyloides stercoralis.</i>
<i>Taeniasis</i>	<i>Caused by the cestodes (tapeworms) Taenia saginata (beef tapeworm) and T. solium (pork tapeworm). Taenia solium can also cause cysticercosis.</i>
<i>Toxoplasmosis</i>	<i>Is an infection caused by parasite protozoa called Toxoplasma gondii.</i>
<i>Trichinellosis</i>	<i>Also called trichinosis, is caused by eating raw or undercooked meat of animals infected with the larvae of a species of worm nematodes called Trichinella.</i>
<i>Trichomoniasis</i>	<i>Common sexually transmitted disease (STD), caused by Trichomonas vaginalis protozoa, that affects both women and men, although symptoms are more common in women.</i>
<i>Trichuriasis</i>	<i>Caused by the nematode (roundworm) Trichuris trichiura, also called the human whipworm.</i>
<i>Viral disease</i>	<i>Caused by several viruses. Viruses are the smallest life-form existing, since they are not even a single cell. It is almost like they are not alive at all. They are small strands of DNA-like cell material. A virus consists mostly of RNA and cannot survive without host cells.</i>

A1.7. Dicionário de conceitos produzido na etapa de Conceitualização do ciclo de expansão da CELO em CELO – Human Disease

Concept Name	Synonyms	Class Attributes	Relations
<i>Acanthamoeba infection</i>	Infecção por acantoameba		<i>hasCause, hasDefinition, hasDiagnostic, hasProtozoa, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>African trypanosomiasis</i>	<i>Sleeping sickness</i> Tripanossomose Africana		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Amebiasis</i>	<i>Amoebiasis</i> Amebíase		<i>hasCause, hasDefinition, hasDiagnostic, hasProtozoa, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Angiostrongyliasis</i>	Angiostrongilíase		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Anisakiasis</i>	Anisaquíase		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Ascariasis</i>	Ascariíase		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Babesiosis</i>	Babesiose		<i>hasCause, hasDefinition, hasDiagnostic, hasProtozoa, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Bacterial Diseases</i>	Doença Bacteriana		<i>isTypeOfInfectiousDisease</i>
<i>Balantidiasis</i>	Balantidíase		<i>hasCause, hasDefinition, hasDiagnostic, hasProtozoa, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Capillariasis</i>	Capilariíase		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Cell</i>	Célula	<i>Disease</i>	
<i>Chagas Disease</i>	<i>American Trypanosomiasis</i> Doença de		<i>hasCause, hasDefinition, hasDiagnostic, hasProtozoa,</i>

	Chagas		<i>hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Clonorchiasis</i>	Clonorquíase		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Cryptosporidiosis</i>	Crypto Ciptosporidiose		<i>hasCause, hasDefinition, hasDiagnostic, hasProtozoa, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Cutaneous Larva Migrans</i>	Creeping eruption Larva Migrans Cutânea		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Cyclosporiasis</i>	Ciclosporíase		<i>hasCause, hasDefinition, hasDiagnostic, hasProtozoa, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Cysticercosis</i>	Cisticercose		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Difilobotriasis</i>	Difilobotríase		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Disease</i>	Doença	<i>Human Disease</i>	<i>isDisease</i>
<i>Dracunculiasis</i>	Dracunculíase		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Ectoparasite</i>	Ectoparasitos		
<i>Enterobiasis</i>	Pinworm infection Enterobíase		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Fascioliasis</i>	Fasciolíase		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Fungal diseases</i>	Doença fúngica		<i>isTypeOfInfectiousDisease</i>
<i>Giardiasis</i>	Giardíase		<i>hasCause, hasDefinition, hasDiagnostic, hasProtozoa, hasPrevention, hasPrognostic,</i>

			<i>hasTransmission, isDiseaseCausedBy</i>
<i>Helminths</i>	<i>Helmentic Helminthiasis Worms Helmintos</i>	<i>Angiostrongyliasis Anisakiasis Ascariasis Capillariasis Clonorchiasis Cutaneous Larva Migrans Cysticercosis Diphyllobothriasis Dracunculiasis Enterobiasis Fascioliasis Hydatidosis Hookworm infection Hymenolepiasis Onchocerciasis Paragonimiasis Schistosomiasis Strongyloidiasis Taeniasis Trichinellosis Trichuriasis Visceral Larva Migrans</i>	
<i>Hookworm infection</i>	Ancilostomiase		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Human Disease</i>	Doença humana		<i>hasInfectiousDisease</i>
<i>Hydatidosis</i>	Hidatidose		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Hymenolepiasis</i>	Himenolepiase		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Infectious Disease</i>	Doença infecciosa	<i>Bacterial Disease Fungal Disease Parasitic Disease Viral Disease</i>	
<i>Visceral Larva Migrans</i>	Larva migrans visceral		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Leishmaniasis</i>	Leishmaniose		<i>hasCause, hasDefinition, hasDiagnostic, hasProtozoa, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Malaria</i>	Malária		<i>hasCause, hasDefinition, hasDiagnostic, hasProtozoa, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Microsporidiosis</i>	<i>Microsporidiasis Microsporidiose</i>		<i>hasCause, hasDefinition, hasDiagnostic, hasProtozoa, hasPrevention,</i>

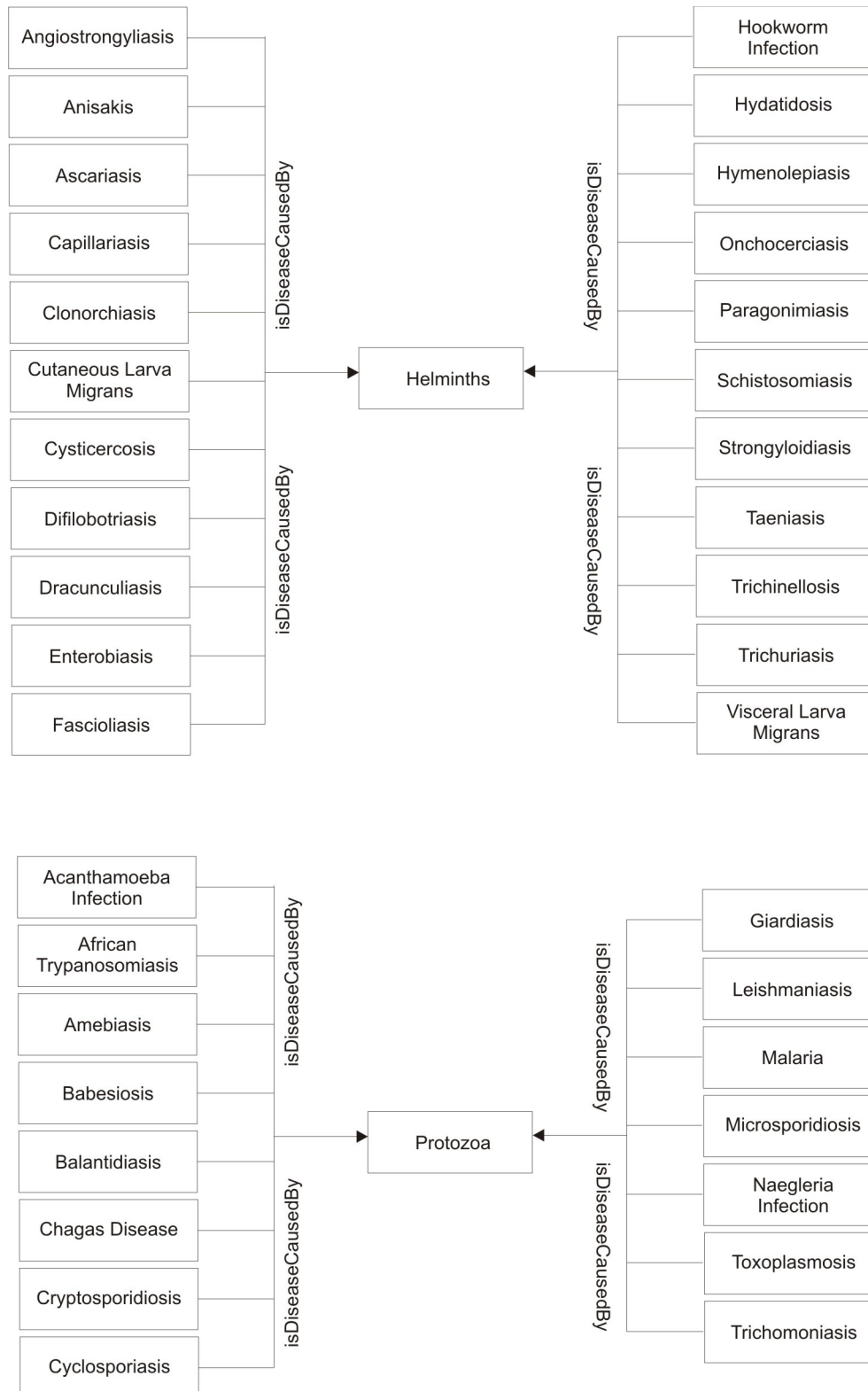
			<i>hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Naegleria infection</i>	Infecção de Naegleria		<i>hasCause, hasDefinition, hasDiagnostic, hasProtozoa, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Onchocerciasis</i>	Oncocercose		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Paragonimiasis</i>	Paragonimíase		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Parasitic disease</i>	Doença parasitária	<i>Ectoparasite Helminths Protozoa</i>	<i>hasParasitic, isTypeOfInfectiousDisease</i>
<i>Protozoa</i>	Protozoário	<i>Acanthamoeba infection African trypanosomiasis Amebiasis Babesiosis Balantidiasis Chagas Disease Cryptosporidiosis Cyclosporiasis Giardiasis Leishmaniasis Malaria Microsporidiosis Naegleria infection Toxoplasmosis Trichomoniasis</i>	<i>isDiseaseCausedBy</i>
<i>Schistosomiasis</i>	Esquistossomose		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Strongyloidiasis</i>	Estrongiloidíase		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Taeniasis</i>	<i>Tapeworm infection</i> Teníase		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Toxoplasmosis</i>	Toxoplasmose		<i>hasCause, hasDefinition, hasDiagnostic, hasProtozoa, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Trichinellosis</i>	<i>Trichinosis</i> Triquinelose		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic,</i>

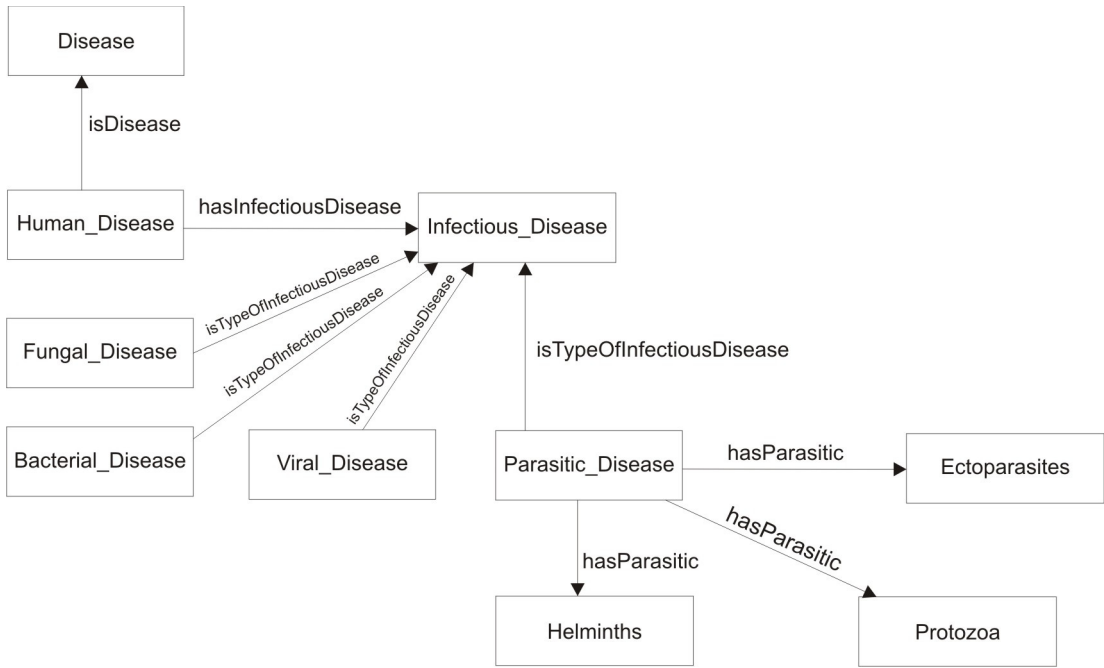
			<i>hasTransmission, isDiseaseCausedBy</i>
<i>Trichomoniasis</i>	Triconomíase		<i>hasCause, hasDefinition, hasDiagnostic, hasProtozoa, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Trichuriasis</i>	Whipworm infection Tricuríase		<i>hasCause, hasDefinition, hasDiagnostic, hasHelminths, hasPrevention, hasPrognostic, hasTransmission, isDiseaseCausedBy</i>
<i>Viral disease</i>	Doença viral		<i>isTypeOfInfectiousDisease</i>

A1.8 Árvore de classificação dos conceitos produzido na etapa de Formalização do ciclo de expansão da CELO em CELO – Human Disease

Cell
Disease
 Human_Disease
 Infectious_Disease
 Bacterial_Disease
 Fungal_Disease
 Parasitic_disease
 Ectoparasites
 Helminths
 Angiostrongyliasis
 Anisakiasis
 Ascariasis
 Capillariasis
 Clonorchiasis
 Cutaneous_Larva_Migrans
 Cysticercosis
 Difilobotriasis
 Dracunculiasis
 Enterobiasis
 Fascioliasis
 Hookworm_infection
 Hydatidosis
 Hymenolepiasis
 Onchocerciasis
 Paragonimiasis
 Schistosomiasis
 Strongyloidiasis
 Taeniasis
 Trichinellosis
 Trichuriasis
 Visceral_Larva_Migrans
 Protozoa
 Acanthamoeba_infection
 African trypanosomiasis
 Amebiasis
 Babesiosis
 Balantidiasis
 Chagas disease
 Cryptosporidiosis
 Cyclosporiasis
 Giardiasis
 Leishmaniasis
 Malaria
 Microsporidiosis
 Naegleria_infection
 Toxoplasmosis
 Trichomoniasis
 Viral_Disease

A1.9 Diagrama de relacionamentos dos conceitos produzido na etapa de Formalização do ciclo de expansão da CELO em CELO – Human Disease

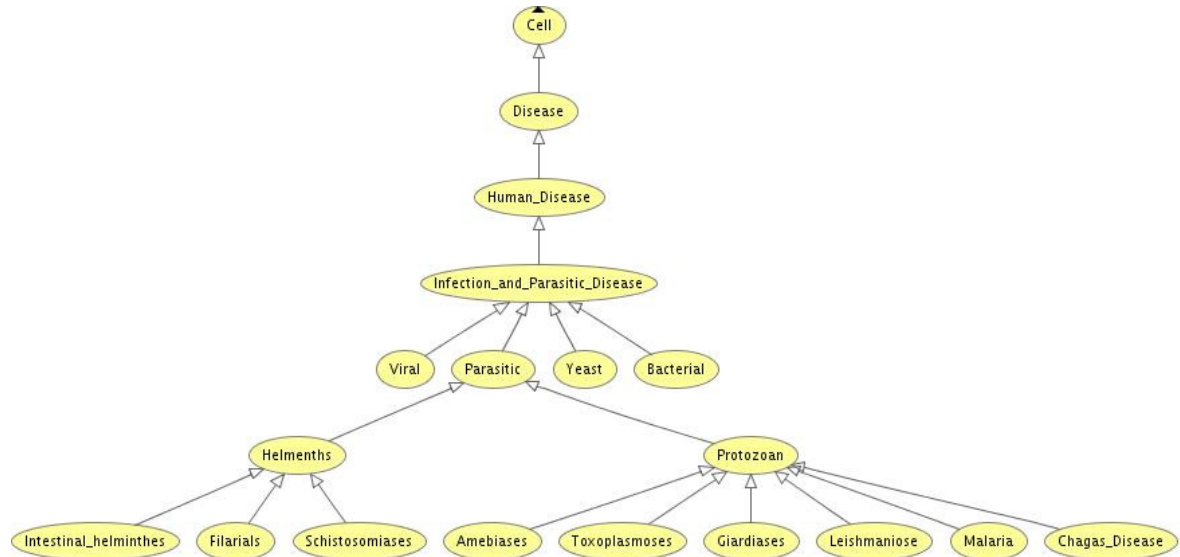




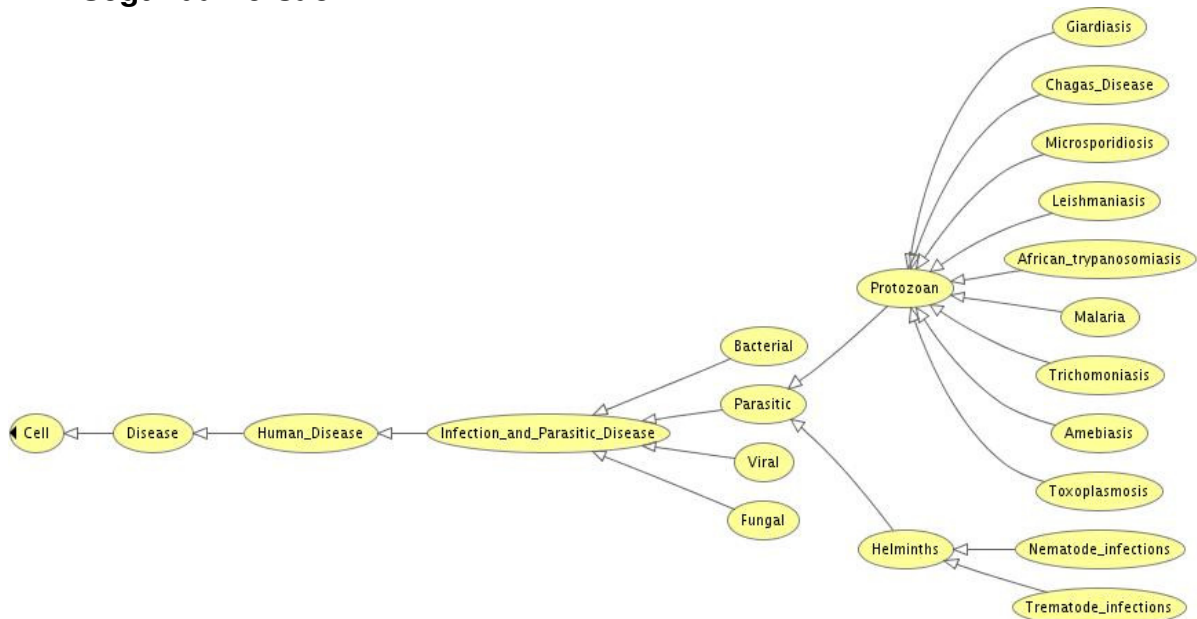
Apêndice 2

Versões das ontologias obtidas durante o ciclo de expansão da ontologia CeLO em CELO – Human Disease

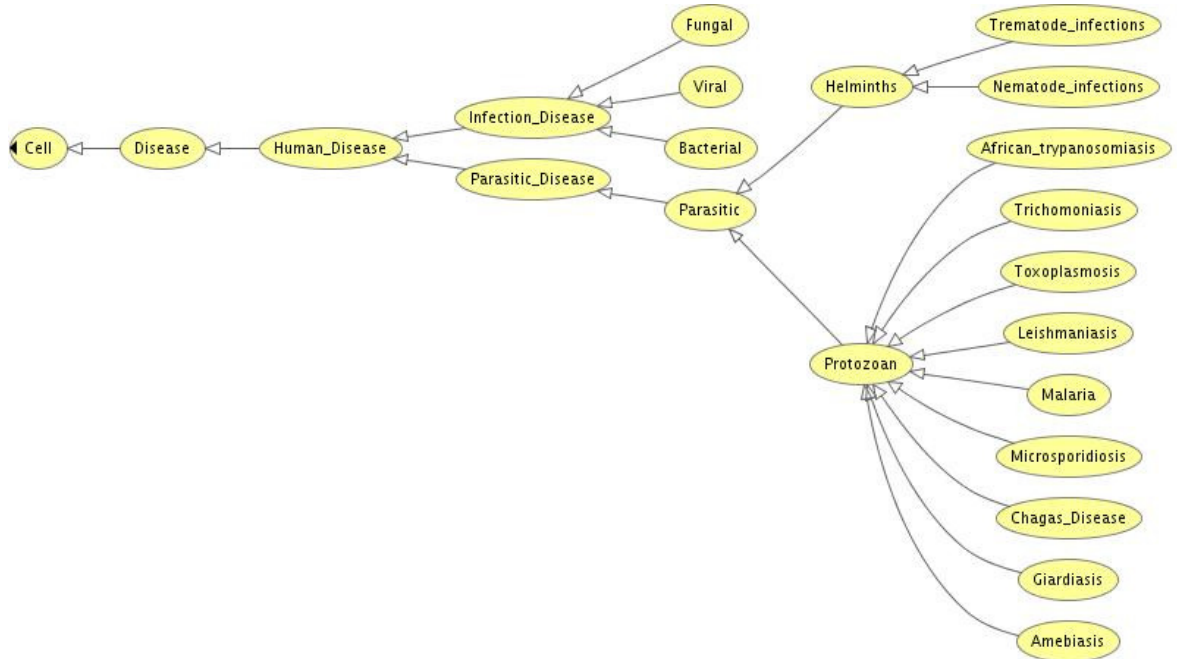
A2.1 Primeira Versão



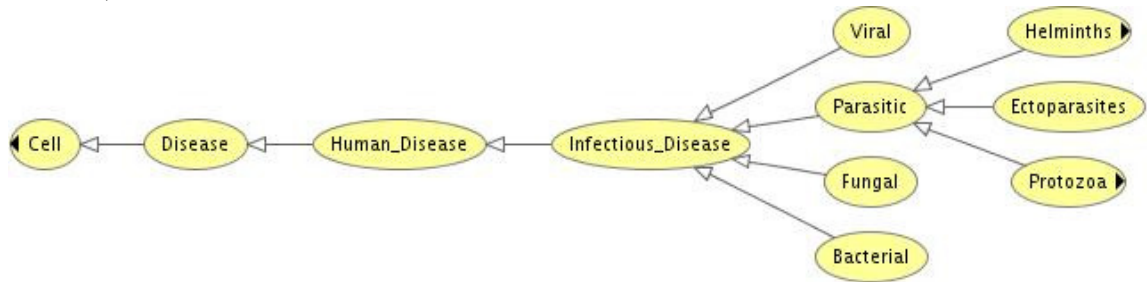
A2.2 Segunda Versão

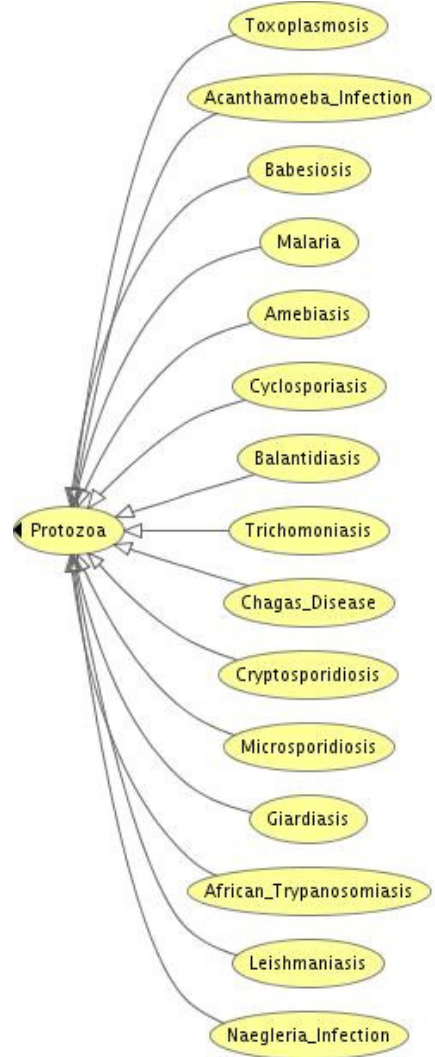
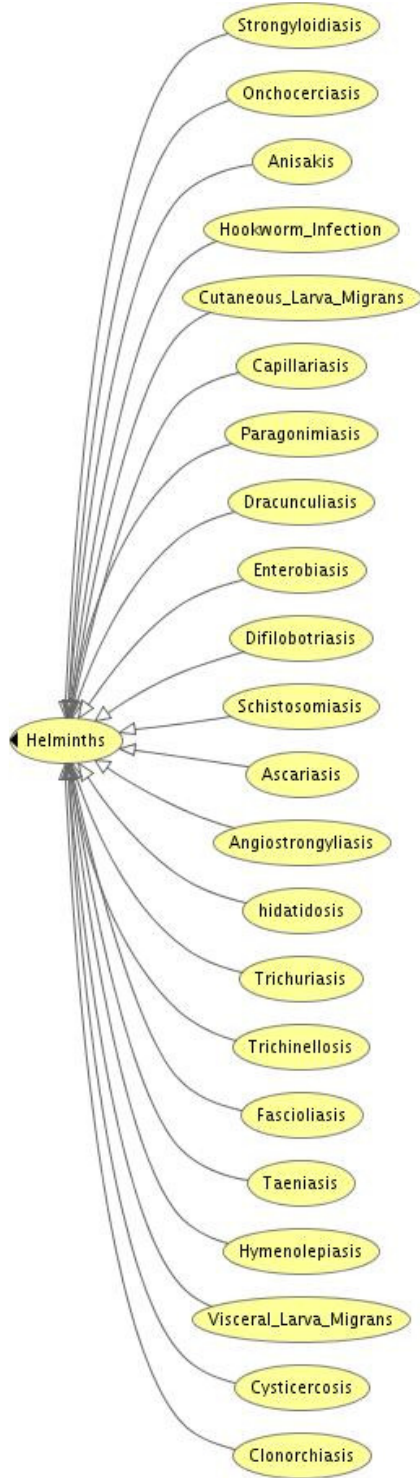


A2.3 Terceira Versão

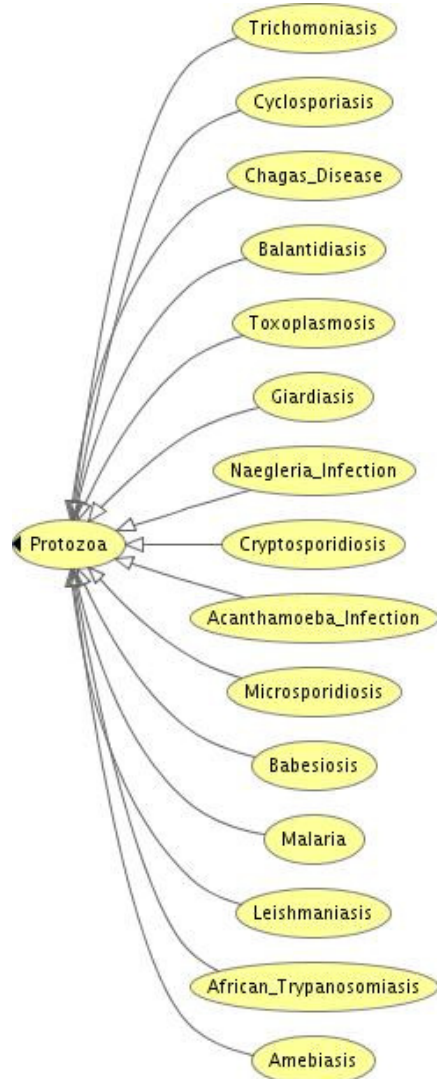
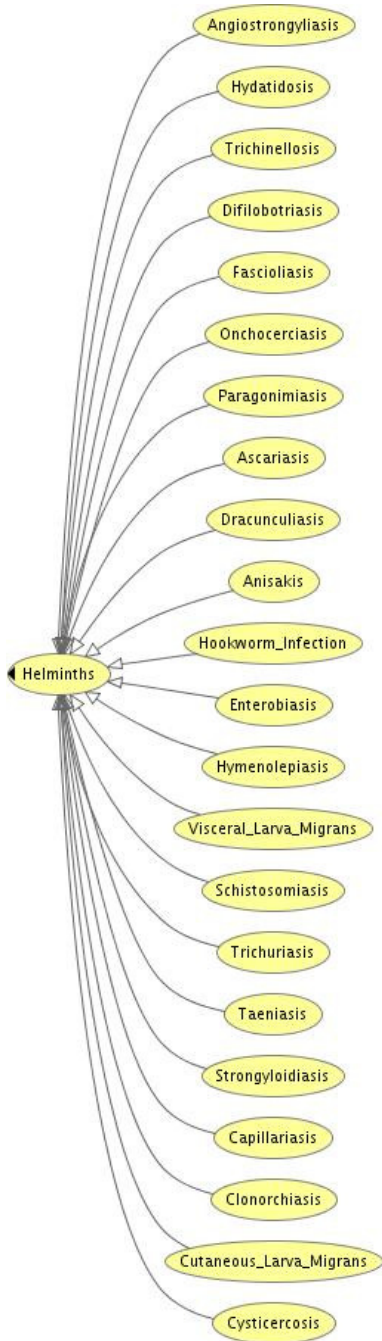
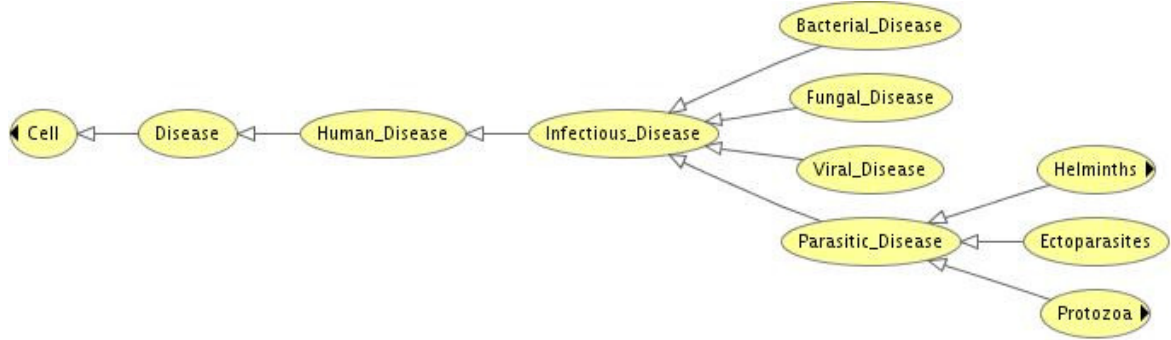


A2.4 Quarta Versão





A2.5 Quinta Versão



Apêndice 3

A3.1 Questionário de validação para a ontologia CELO – Human Disease

Questionário de Validação - Ontologia sobre Doenças Humanas

Formação do avaliador(a):

Assinale a alternativa que representa a sua resposta para a pergunta.

PERGUNTAS	SIM	NÃO
Doenças humanas são doenças?		
Doenças infecciosas são doenças humanas?		
Doenças parasitárias são doenças infecciosas?		
Doenças bacterianas são doenças infecciosas?		
Doenças fúngicas são doenças infecciosas?		
Doenças virais são doenças infecciosas?		
Protozoários são parasitas?		
Helmintos são parasitas?		
Ectoparasitas são parasitas?		
<i>A Amebíase é uma doença causada por protozoários?</i>		
<i>A Babesiose é uma doença causada por protozoários?</i>		
<i>A Balantidíase é uma doença causada por protozoários?</i>		
<i>A Ciclosporíase é uma doença causada por protozoários?</i>		
<i>A Criptosporidiose é uma doença causada por protozoários?</i>		
<i>A Doença de Chagas é uma doença causada por protozoários?</i>		
<i>A Giardíase é uma doença causada por protozoários?</i>		
<i>A Infecção de Naegleria é uma doença causada por protozoários?</i>		
<i>A Infecção por Acantamoeba é uma doença causada por protozoários?</i>		
<i>A Leishmaniose é uma doença causada por protozoários?</i>		
<i>A Malária é uma doença causada por protozoários?</i>		
<i>A Microsporidiose é uma doença causada por protozoários?</i>		
<i>A Toxoplasmose é uma doença causada por protozoários?</i>		
<i>A Tricomoníase é uma doença causada por protozoários?</i>		
<i>A Tripanossomose Africana é uma doença causada por protozoários?</i>		
<i>A Angiostrongilíase é uma doença causada por helmintos?</i>		
<i>A Anisquíase é uma doença causada por helmintos?</i>		
<i>A Ascariíase é uma doença causada por helmintos?</i>		
<i>A Capilaríase é uma doença causada por helmintos?</i>		
<i>O Clonorquíase é uma doença causada por helmintos?</i>		
<i>A Cisticercose é uma doença causada por helmintos?</i>		
<i>A Difilobotríase é uma doença causada por helmintos?</i>		
<i>A Dracunculíase é uma doença causada por helmintos?</i>		
<i>A Hidatidose é uma doença causada por helmintos?</i>		
<i>A Enterobíase é uma doença causada por helmintos?</i>		
<i>A Fasciolíase é uma doença causada por helmintos?</i>		
<i>A Himenolepíase é uma doença causada por helmintos?</i>		
<i>A Ancilostomíase é uma doença causada por helmintos?</i>		
<i>A Oncocercose é uma doença causada por helmintos?</i>		
<i>A Paragonimíase é uma doença causada por helmintos?</i>		
<i>A Esquistossomose é uma doença causada por helmintos?</i>		
<i>A Estrongiloidíase é uma doença causada por helmintos?</i>		
<i>A Teníase é uma doença causada por helmintos?</i>		

A Larva Migrans Visceral é uma doença causada por helmintos?		
A Larva Migrans Cutânea é uma doença causada por helmintos?		
A Triquinelose é uma doença causada por helmintos?		
A Tricuríase é uma doença causada por helmintos?		

Observando a figura abaixo, que representa a ontologia, você sugeriria alguma alteração em relação a:

- novas ligações,
- novos termos,
- sinônimos para os termos?

Escreva aqui as suas sugestões.

