

UNIVERSIDADE FEDERAL DA GRANDE DOURADOS

Wagner Kenedy Waterkemper

Representação e Consulta na Linguagem OWL de Conceitos e Associações

Extraídos de Informação Não Estruturada

DOURADOS – MS

2016

Wagner Kenedy Waterkemper

Representação e Consulta na Linguagem OWL de Conceitos e Associações

Extraídos de Informação Não Estruturada

Trabalho de Conclusão de Curso de graduação apresentado para obtenção do título de Bacharel em Sistemas de Informação pela Faculdade de Ciências Exatas e Tecnologia da Universidade Federal da Grande Dourados.

Orientador: Prof^o Dr. Joinvile Batista Junior

DOURADOS – MS

2016

Wagner Kenedy Waterkemper

Representação e Consulta na Linguagem OWL de Conceitos e Associações

Extraídos de Informação Não Estruturada

Trabalho de Conclusão de Curso aprovado como requisito para obtenção do título de Bacharel em Sistemas de Informação na Universidade Federal da Grande Dourados, pela comissão formada por:

Orientador Prof. Dr. Joinvile Batista Junior

FACET – UFGD

Prof. Me. Murilo Táparo

FACET – UFGD

Prof. Dr. Wellington Lima dos Santos

FACET – UFGD

Dourados, 28 de Abril de 2016.

RESUMO

A linguagem OWL tem sido extensamente utilizada para representar formalmente uma ontologia. Exemplos de ontologias em OWL, reportados na literatura são baseados em uma padronização dos conceitos e relações utilizados para representar a ontologia. No entanto, o mapeamento de relações extraídas de sentenças extraídas de uma fonte não estruturada (texto em linguagem natural) para a linguagem OWL não é uma tarefa simples, em função da variedade e heterogeneidade de conceitos e relações que emergem das sentenças. O objetivo deste trabalho é realizar um estudo de caso de mapeamento de sentenças de um texto em linguagem natural para ser representado formalmente na linguagem OWL. O mapeamento gerado foi representado em OWL, com a utilização da ferramenta Protégé [STANFORD]. A associação do Protégé com uma máquina de raciocínio (*reasoner*) suporta a realização de consultas, a partir da ontologia descrita em OWL. O mapeamento para OWL foi realizado com foco nas consultas a serem realizadas. O *corpus* (texto de entrada) para este estudo de caso foi um texto na área de nutrição humana básica abordando escolhas nutricionais. Este texto é rico em conceitos e relações, caracterizando um desafio significativo para o mapeamento em OWL.

Palavras-Chave: Linguagem OWL, ontologia, Protégé.

SUMÁRIO

	Lista de Figuras.....	vi
	Lista de Tabelas.....	viii
1	Introdução.....	01
1.1	Motivação.....	01
1.2	Oportunidades e Relevância.....	01
1.3	Objetivos do Trabalho.....	02
1.4	Metodologia Adotada.....	02
1.5	Conteúdo do Trabalho.....	03
2	Fundamentação Teórica.....	03
2.1	A Linguagem OWL.....	04
2.1.1	Propriedades OWL.....	08
2.1.2	Classes Primitivas e Definidas (Condições Necessárias e Suficientes).....	14
2.1.3	Padrões de Projeto.....	15
2.1.4	Classes Complexas: Intersecção e União.....	17
2.2	O suporte do ambiente Protégé para a linguagem OWL.....	18
2.2.1	Hierarquia de Classes em Protégé.....	19
2.2.2	Ilustração de Propriedades e Restrições.....	19
2.2.3	Condições Necessárias e Suficientes.....	21
2.2.4	Padrões de Projeto: Axiomas de Cobertura.....	22
2.2.5	Utilização de Raciocinadores.....	22
2.2.6	Pressuposto de Mundo Aberto: Axiomas de Fechamento.....	24
3	Desenvolvimento do Trabalho Proposto.....	26
3.1	Ontologia para aplicação no domínio de Nutrição Humana.....	26
3.2	Processo utilizado e resultados obtidos	28
3.2.1	Reescrita da sentença em linguagem natural para evidenciar as relações relevantes	29
3.2.2	Representação das hierarquias de classes relevantes.....	30
3.2.3	Representação das hierarquias de propriedades relevantes.....	31
3.2.4	Representação dos relacionamentos com foco nas consultas potenciais.....	32
3.2.4.1	Consulta com foco no uso do operador "not".....	33
3.2.4.2	Consulta com foco em respostas contendo um conjunto de entidades 1.....	34
3.2.4.3	Consulta com foco na hierarquia de classes.....	36
3.2.4.4	Consulta com foco na hierarquia de propriedades.....	38
3.2.4.5	Consulta com foco na hierarquia de classes e propriedades.....	40
4	Considerações Finais.....	42
4.1	Conclusões.....	42
4.2	Dificuldades Encontradas.....	43
4.3	Trabalhos Futuros.....	43
	REFERÊNCIAS.....	44

LISTA DE FIGURAS

Figura	Representação	Pág
2.1	Indivíduos.....	06
2.2	Propriedades.....	07
2.3	Classes (incluindo indivíduos e propriedades).....	07
2.4	Propriedade do tipo objeto.....	09
2.5	Propriedade do tipo de dados.....	09
2.6	Propriedade do tipo anotação.....	09
2.7	Propriedade inversa.....	09
2.8	Propriedade funcional.....	10
2.9	Propriedade inversa funcional.....	11
2.10	Propriedade transitiva.....	11
2.11	Propriedade simétrica.....	11
2.12	Propriedade assimétrica.....	12
2.13	Propriedade reflexiva.....	12
2.14	Propriedade irreflexiva.....	12
2.15	Domínio e intervalo de valores.....	13
2.16	Condições necessárias.....	15
2.17	Condições necessárias e suficientes.....	15
2.18	Sem axioma de cobertura.....	17
2.19	Com axioma de cobertura.....	17
2.20	Intersecção de classes.....	17
2.21	União de classes.....	18
2.22	Hierarquia de classes em Protégé.....	19
2.23	Restrição existencial em Protégé.....	20
2.24	Restrição de cardinalidade em Protégé.....	20
2.25	Restrição de valores em Protégé.....	20
2.26	Condições necessárias em Protégé.....	21
2.27	Condições necessárias e suficientes em Protégé.....	21
2.28	Restrição utilizando axioma de cobertura em Protégé.....	22
2.29	Hierarquia afirmada e hierarquia inferida.....	23
2.30	Classe Sonda.....	23
2.31	Inconsistência.....	24
2.32	Axioma de fechamento.....	25
3.1	Consulta: “isBase_toDecideAbout some FoodChoices”.....	34
3.2	Consulta: “not (isBase_toDecideAbout some FoodChoices)”.....	34
3.3	Consulta: “hasInfluenceOn some FoodChoices”.....	36
3.4	Consulta: “hasStrongest_InfluenceOn some FoodChoices”.....	36
3.5	Hierarquia de Classes.....	37
3.6	Hierarquia de Classes Expandidas.....	37
3.7	Consulta: “stimulates some SocialMeals”.....	38

3.8	Consulta: “stimulates some SharingFood”.....	38
3.9	Hierarquia de Propriedades.....	39
3.10	Consulta: “mayInfluenceLittle some People_Health”.....	40
3.11	Consulta: “mayInfluenceMuch some People_Health”.....	40
3.12	Consulta: “mayInfluence some People_Health”.....	40
3.13	Consulta: “hasStrongest_influenceOn some LifeQuality”.....	42

LISTA DE TABELAS

1	Transição do texto original – exemplo 01.....	29
2	Transição do texto original – exemplo 02.....	29
3	Transição do texto original – exemplo 03.....	30
4	Representação em OWL – exemplo 01.....	30
5	Representação em OWL – exemplo 02.....	31
6	Representação em OWL – exemplo 03.....	31
7	Representação em OWL – exemplo 04.....	32
8	Representação em OWL – exemplo 05.....	32
9	Representação em OWL – exemplo 06.....	32
10	Representação em OWL – exemplo 07.....	33
11	Representação em OWL – exemplo 08.....	35
12	Representação em OWL – exemplo 09.....	36
13	Representação em OWL – exemplo 10.....	38
14	Representação em OWL – exemplo 11.....	41

1. Introdução

Uma ontologia permite a extração de conhecimento, a partir de um domínio de interesse, através de relacionamentos entre conceitos, servindo de base para buscas na Web baseadas na semântica dos conceitos e relações extraídos de sentenças, escritas em linguagem natural, em substituição a buscas baseadas em palavras chaves.

"A semântica representa o significado dos dados. E para que esse significado possa ser compreendido pela máquina, é necessária uma detalhada e complexa descrição da representação desses dados através de suas propriedades, conceitos e relacionamentos." (LUSTOSA, 2003).

A linguagem OWL (*Ontology Web Language*) é muito utilizada para a representação formal de ontologias.

Este trabalho tem por finalidade a construção de uma pequena ontologia na linguagem OWL para caracterizar o suporte provido por OWL para a realização de consultas.

1.1 Motivação

Uma ontologia suporta a representação de conceitos e relacionamentos existentes em um domínio de conhecimento. A extração automática de informação para a construção de Ontologias é uma área de pesquisa ativa na comunidade de processamento de linguagem natural.

A construção automática de uma ontologia pode ser avaliada a partir de sua representação, em uma linguagem formal para caracterizar a captura adequada dos conceitos e relacionamentos utilizados na construção da ontologia, bem como avaliar a capacidade de estendê-la a partir de consultas.

1.2 Oportunidades e Relevância

OWL é uma linguagem formal muito utilizada para a representação de ontologias, que incorpora a definição de propriedades a partir das quais são possíveis consultas para extensão da ontologia. É suportada pelo ambiente Protégé que disponibiliza uma interface gráfica para construção e visualização da ontologia em construção, bem como a interligação com motores de raciocínio que suportam consultas sobre a mesma construída em OWL.

O estudo da capacidade de representação, e da realização de consultas em OWL é de grande relevância para posterior integração dos resultados gerados no projeto de pesquisa do orientador deste trabalho, com ontologias construídas automaticamente, dado que a interface gráfica do ambiente Protégé permite: (a) inspecionar a ontologia em construção; (b) realizar consultas sobre o conhecimento representado na ontologia.

Em função do fato de que o referido trabalho de pesquisa, do orientador deste trabalho, utiliza a língua inglesa para a extração de conceitos e relacionamentos, os exemplos e o estudo de caso apresentados neste trabalho também serão representados na língua inglesa.

1.3 Objetivos do Trabalho

O objetivo geral deste trabalho é estudar a capacidade de representação e consultas em uma ontologia na linguagem OWL.

Os objetivos específicos são:

- a) Estudar as construções disponíveis em OWL para a representação de uma ontologia;
- b) Especificar uma pequena ontologia, que sirva como estudo de caso para avaliar o suporte da linguagem OWL para a sua representação e para a realização de consultas.

1.4 Metodologia Adotada

A metodologia adotada pode ser descrita em cinco etapas:

- **Primeira etapa:** Estudo e conceituação da linguagem OWL.
- **Segunda etapa:** Estudo do ambiente Protégé para suportar a utilização de OWL.
- **Terceira etapa:** Estudo de uma pequena aplicação existente para se capacitar na representação de uma ontologia em OWL no ambiente Protégé.

- **Quarta etapa:** Refraseamento do texto original para um texto, ainda em linguagem natural, que melhor represente os conceitos e relacionamentos relevantes para o assunto de interesse.
- **Quinta etapa:** Representação dos conceitos e relacionamentos da ontologia Nutrição, em linguagem OWL, com a utilização do ambiente Protégé, com base nas consultas que possam ser realizadas a partir da representação da ontologia.

1.5 Conteúdo do Trabalho

No capítulo 2, Fundamentação Teórica, são conceituados e exemplificados os principais elementos da linguagem OWL e ilustrada a utilização da ferramenta Protégé ao editar um exemplo de ontologia utilizando alguns dos conceitos da subcessão anterior.

No capítulo 3, Desenvolvimento do Trabalho Proposto, é apresentada a construção de uma pequena ontologia, denominada Nutrição, para caracterizar a capacidade de representação em OWL, bem como, o suporte às consultas.

No capítulo 4, Considerações Finais, são apresentadas as conclusões do trabalho, as dificuldades encontradas e as sugestões para trabalhos futuros.

2. Fundamentação Teórica

OWL é a principal linguagem para criação de ontologias para Web, recomendada pela W3C (*World Wide Web Consortium*), organização responsável por desenvolver protocolos e recomendações para assegurar o crescimento da Web em longo prazo.

A seguir são conceituados os principais elementos da Linguagem OWL, assim como a exemplificação de uma ontologia em um ambiente gráfico chamado Protégé.

Os conceitos e exemplos apresentados neste capítulo são baseados em um tutorial sobre a construção de uma ontologia de ilustração, utilizando a linguagem OWL na ferramenta Protégé (HORRIDGE, 2011). Algumas referências adicionais também são utilizadas.

2.1 A Linguagem OWL

OWL é a linguagem mais atual para o desenvolvimento de ontologias para web recomendada pela W3C. [W3 Org.]

A finalidade principal da Web Semântica é atribuir significados (semântica) em conteúdos on-line que sejam legíveis tanto para computador quanto para o ser humano. Neste contexto, ontologias têm sido construídas a fim de representar o conhecimento em um dado domínio.

OWL surgiu a partir de uma revisão analítica da linguagem "DAML+OIL", linguagem de marcação semântica para web baseada nas linguagens "RDF" e "RDF Schema".

A linguagem RDF (*Resource Description Framework*) de acordo com MANOLA (2004) “é uma linguagem para representar as informações sobre os recursos na *World Wide Web*” e “baseia-se na ideia de que as ocorrências descritas possuem propriedades que têm valores e que os recursos podem ser descritos por meio de declarações”.

Basicamente, RDF é responsável por modelar dados em objetos (recursos), o que facilita o processamento de informações a partir de aplicativos além de garantir uma baixa taxa de perda de significado de tais informações.

A estrutura de modelagem RDF se resume em três elementos: sujeito, predicado e objetos. Imagina-se um exemplo onde se afirma que uma pessoa, chamada João Henrique criou uma página qualquer. Formalmente dizendo: “<http://www.exemplo.org/index.html> tem um criador cujo nome é João Henrique”.

O modelo RDF reconheceria como os seguintes elementos:

- Sujeito = <http://www.exemplo.org/index.html>;
- Predicado = criador;
- Objeto = João Henrique

Já em relação à linguagem RDF Schema, trata-se de um vocabulário para modelagem de dados para dados RDF, além de ser uma evolução da mesma.

É uma extensão semântica do RDF. Prevê mecanismos para descrever grupos de recursos relacionados e as relações entre estes recursos. Esses recursos são usados

para determinar características de outros recursos, como os domínios e os alcances de propriedades. (BRICKLEY e GUHA, 2014).

Percebe-se então, conceitos como construção de classes bem como o relacionamento entre elas, conhecida como hierarquia de classes; domínio e alcance de propriedades além de outras características não suportadas por RDF.

Portanto, como dito anteriormente, RDF e RDF Schema formam a base para a implementação da linguagem OWL, objeto de estudo deste trabalho. Elementos presente em ambas as linguagens, tais como classes, indivíduos e relacionamentos, são descritas neste estudo como sendo pertencentes à linguagem OWL para uma melhor conceituação sobre as particularidades da Linguagem para criação de ontologias à web.

Antes de conceituar os elementos presentes em OWL, é preciso ter o conhecimento sobre algumas restrições atuais na linguagem.

De acordo com *McGuinness e Harmelen* (2004) “OWL fornece três sublinguagens cada vez mais expressivas, projetadas para uso por comunidades específicas de implementadores e usuários”:

- OWL Lite suporta aqueles usuários que necessitam principalmente de uma hierarquia de classificação e restrições simples. Por exemplo, enquanto ele suporta restrições de cardinalidade, ela só permite valores de cardinalidade de ‘0’ ou ‘1’. Deve ser mais simples de fornecer suporte de ferramentas para OWL Lite que seus parentes mais expressivos e OWL Lite oferece um caminho de migração rápida para enciclopédias e outras taxonomias. OWL Lite também tem uma complexidade formal menor do que OWL DL.
- OWL-DL suporta aqueles usuários que querem a expressividade máxima, mantendo a integralidade computacional (todas as conclusões são garantidas para ser computável) e decidibilidade (todas as computações terminarão em tempo finito). OWL DL inclui todas as construções da linguagem OWL, mas eles podem ser usados somente sob certas restrições (por exemplo, enquanto que uma classe pode ser uma subclasse de muitas classes, uma classe não pode ser uma instância de outra classe). OWL DL é assim chamado devido à sua correspondência com as lógicas de descrição, um campo de pesquisa que estuda as lógicas que formam a base formal da OWL.
- OWL Full é destinado a usuários que desejam máxima expressividade e a liberdade sintática do RDF sem garantias computacionais. Por exemplo, em OWL Full uma classe pode ser tratada simultaneamente como uma coleção de indivíduos e como um indivíduo em seu próprio direito. OWL Full permite uma Ontologia aumentar o significado do vocabulário pré-definido (RDF ou OWL). É improvável que qualquer

software de raciocínio seja capaz de suportar o raciocínio completo para todos os recursos do OWL Full. (McGUINNESS e HARMELEN, 2004).

Percebe-se que a principal característica distinta entre as três versões é o grau de expressividade. Os conceitos sobre a linguagem OWL, bem como o exemplo de ontologia utilizados neste trabalho se referem à sublinguagem OWL-DL (*Description Logic*).

“Uma ontologia é uma especificação explícita de uma conceitualização” (GRUBER, 1992). O foco principal da construção de ontologias é representar conhecimento sobre um domínio de interesse a partir de conceitos e relacionamentos. Em OWL, uma ontologia consiste basicamente em três componentes: Indivíduos, Propriedades e Classes.

Indivíduos representam os objetos no domínio de interesse. Uma característica importante em OWL é que dois ou mais indivíduos podem representar o mesmo objeto, ou seja, dois nomes distintos poderão pertencer à mesma pessoa. Por exemplo, *Queen Elizabeth*, *The Queen* e *Elizabeth Windsor* para a linguagem OWL representam três indivíduos, mas que na realidade refere-se à mesma pessoa. Para tanto, faz-se necessário definir explicitamente que dois ou mais indivíduos sejam equivalentes a fim de não causar inconsistências na ontologia.

Na figura 2.1 são ilustrados alguns indivíduos sem qualquer tipo de relacionamento ou propriedades sobre eles.



Figura 2.1 - Indivíduos

Como dito anteriormente, indivíduos representam objetos que por sua vez são conhecidos como instâncias de classes. Basicamente, uma classe representa uma restrição, um delimitador capaz de informar se um indivíduo pertence ou não àquela classe.

“Propriedades são relações binárias entre indivíduos, ou seja, propriedades ligam dois indivíduos” (HORRIDGE, 2011). Basicamente, um indivíduo se relaciona com outro indivíduo através de uma propriedade específica.

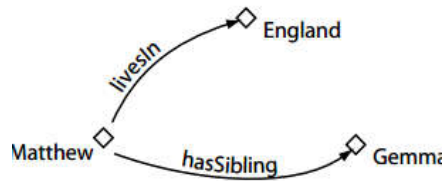


Figura 2.2 - Propriedades

Na figura 2.2 é ilustrada o relacionamento do indivíduo *Matthew* com outros dois indivíduos, *England* e *Gemma* através das propriedades *livesIn* e *hasSibling* respectivamente. Traduzindo os termos, obtêm-se as informações de que *Matthew* mora na Inglaterra e tem uma irmã que se chama *Gemma*.

Em OWL, as propriedades diferem em basicamente três tipos: propriedades de objetos, propriedades do tipo de dados e propriedades de anotação. Estas propriedades e outras mais são conceituadas na subseção 2.1.1 deste trabalho.

Em relação às classes, elas são representadas como sendo conjunto de indivíduos. O fato de ser um delimitador refere-se a como elas são descritas em OWL, de maneira formal, estabelecem os requisitos para um indivíduo pertencer a uma classe específica.

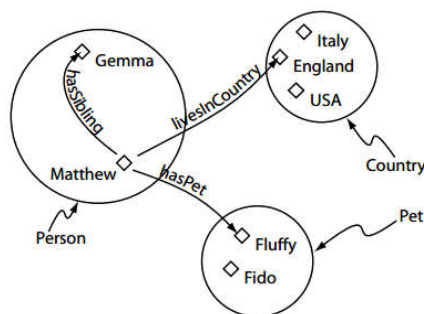


Figura 2.3 - Classes (incluindo indivíduos e propriedades)

A figura 2.3 nos mostra todos os componentes básicos de uma ontologia, conceituados até o momento. As classes, ilustradas como círculos, representam os conjuntos de indivíduos.

Percebe-se que é possível o relacionamento entre indivíduos de mesma classe e também entre classes distintas.

Além disso, é possível em OWL, criar um relacionamento entre classes chamado hierarquia de classes. Esta característica permite especializar uma superclasse em uma ou mais subclasses. Na figura acima, por exemplo, em relação à classe *Person*, poderiam ser criadas duas subclasses, *Man* e *Woman*. Traduzindo e conceituando esta nova característica, significa que "Homens" e "Mulheres" estariam separados por alguma característica que as diferem, mas ainda sim pertenceriam a uma mesma classe, mais genérica, que neste caso, "Pessoas".

Existem casos em que um indivíduo atende a requisitos de classes equivalentes. Em grandes ontologias, é comum um número elevado de classes e pode acontecer de existirem classes semelhantes, com poucas características que as diferem. As classes equivalentes às quais um indivíduo pode participar são conhecidas como classes anônimas.

Além de descrever superclasses nomeadas e superclasses anônimas (restrições), OWL permite a criação de classes a partir de uma lista de indivíduos. Como exemplo disso, podemos criar uma classe chamada "DiasDaSemana" que contém os seguintes indivíduos (e somente eles) {domingo, segunda, terça, quarta, quinta, sexta e sábado}. Essas são conhecidas como classes enumeradas. Isso implica que um indivíduo, membro de "DiasDaSemana", obrigatoriamente estará listado por ela.

2.1.1 Propriedades OWL

Relembrando o breve conceito anterior, as propriedades em OWL são os relacionamentos binários entre indivíduos, ou seja, um indivíduo se relaciona com outro indivíduo através de uma propriedade. Existem basicamente três tipos de propriedades: de objetos, de tipo de dados e de anotação.

As propriedades de objetos representam os relacionamentos entre indivíduos. Na figura 2.4 é ilustrada este tipo de propriedade.



Figura 2.4 - Propriedade do tipo objeto

Por sua vez, as propriedades do tipo de dados descrevem as relações entre valores de dados a um indivíduo. Como é mostrada na figura 2.5, o indivíduo *Matthew* se relaciona com um valor numérico do tipo inteiro via propriedade *hasAge*.

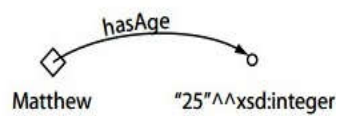


Figura 2.5 - Propriedade do tipo de dados

A propriedade de anotação é usada basicamente para acrescentar informações aos indivíduos e às classes como também para as propriedades de objeto e tipo de dados. Na figura 2.6 é ilustrada uma propriedade de anotação, ligando uma classe a um nome (dado valor do tipo literal).

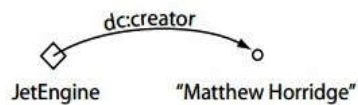


Figura 2.6 - Propriedade do tipo anotação

Os tipos de propriedades em OWL foram conceituados de forma sucinta. Além disso, é possível utilizar propriedades que apresentam características específicas para um dado relacionamento. Na figura 2.7 a seguir, é mostrada uma característica chamada propriedade inversa.

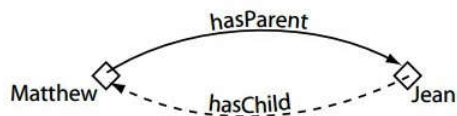


Figura 2.7 - Propriedade inversa

Cada propriedade do tipo objeto pode ter uma propriedade inversa correspondente. A ilustração acima implica que *Matthew* tem um pai chamado *Jean*. O inverso disso, logicamente indica que *Jean* tem um filho chamado *Matthew*. Ou seja, *hasChild* é inverso de *hasParent*.

Além desta característica, as propriedades em OWL do tipo objeto possuem várias outras particularidades, são elas:

- Funcional;
- Inversa Funcional;
- Transitiva;
- Simétrica;
- Assimétrica;
- Reflexiva;
- Irreflexiva.

Como dito anteriormente, em OWL é possível que dois indivíduos distintos sejam o mesmo indivíduo e é a característica funcional que garante isso, ou seja, se um indivíduo "A" se relaciona com dois outros indivíduos "B" e "C" através de uma mesma propriedade funcional, então "B" e "C" são o mesmo indivíduo. Na figura 2.8 é ilustrada esta particularidade.

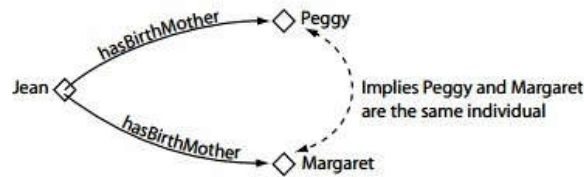


Figura 2.8 - Propriedade funcional

Da mesma forma que existe a propriedade anterior, existe também a inversa da mesma. Formalmente conceituando, se *Jean* tem uma mãe biológica chamada *Peggy*, mas também tem uma mãe biológica chamada *Margaret* e esta propriedade sendo funcional garante que *Peggy* e *Margaret* sejam a mesma pessoa, logicamente é correto afirmar que *Peggy* e *Margaret* é mãe biológica de *Jean*, como podemos ver na figura 2.9.

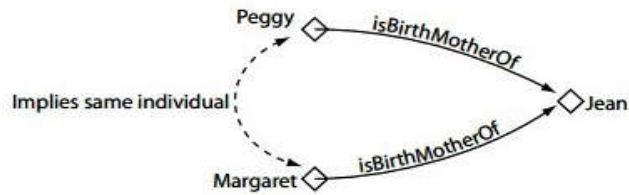


Figura 2.9 - Propriedade inversa funcional

Uma propriedade com característica transitiva garante que, se um indivíduo "A" se relaciona com um indivíduo "B" através de uma propriedade "P" e também que o indivíduo "B" se relaciona com um indivíduo "C" através da mesma propriedade "P", pode-se concluir que "A" se relaciona com "C" através da propriedade "P". Na figura 2.10 é comprovada a transitividade da propriedade *hasAncestor* entre os indivíduos *Matthew*, *Peter* e *William*.

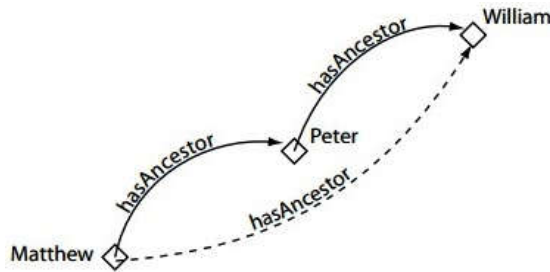


Figura 2.10 - Propriedade transitiva

A característica simetria garante que, se um indivíduo "A" se relaciona com um indivíduo "B" através de uma propriedade "P" então "B" também se relaciona com "A" por meio da mesma propriedade "P". Na figura 2.11, a seguir é ilustrada essa característica.



Figura 2.11 - Propriedade simétrica

Do mesmo modo existe uma maneira de negar a particularidade anterior. É chamada de propriedade assimétrica, que implica que "A" se relaciona a "B" através de "P" mas "B" não pode se relacionar a "A" por meio de "P", como se vê na figura 2.12.

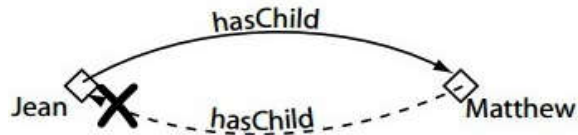


Figura 2.12 - Propriedade assimétrica

É dito que uma propriedade é reflexiva quando ela atua em relacionar um indivíduo a si mesmo, além é claro de relacioná-lo a outro indivíduo através da mesma propriedade. No exemplo a seguir, formalmente dizendo, "George" conhece "Simon" e também que "George" e "Simon" conhecem a si mesmos.

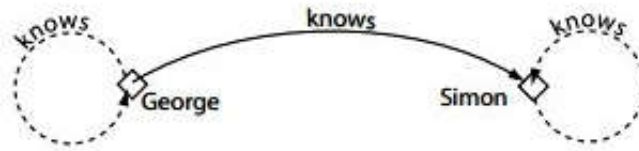


Figura 2.13 - Propriedade reflexiva

Também existe a negação da reflexiva, ou seja, a propriedade irreflexiva.

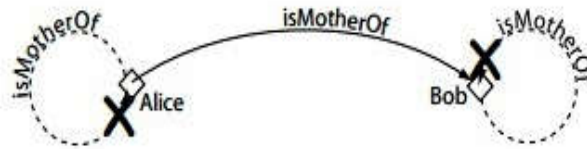


Figura 2.14 - Propriedade irreflexiva

As propriedades OWL garantem o enriquecimento de uma ontologia. Deve-se ter cautela ao definir uma propriedade, pois elas são consideradas axiomas (sentenças verdadeiras) por mecanismos de raciocínio. Por isso, as informações e conhecimentos devem ser precisos.

Outro conceito importante para caracterizar propriedades é chamado de domínio e alcance (intervalo) de propriedades. Observa-se na figura 2.15 que o indivíduo *Matthew* se relaciona com outros três indivíduos sendo que um deles pertence à mesma classe (*Person*). Assuma-se como exemplo a propriedade *livesInCountry* que liga *Matthew* a *England*. No conceito de domínio e intervalo de propriedade temos que a classe *Person* é o domínio e a classe *Country* é o intervalo da propriedade *livesInCountry*.

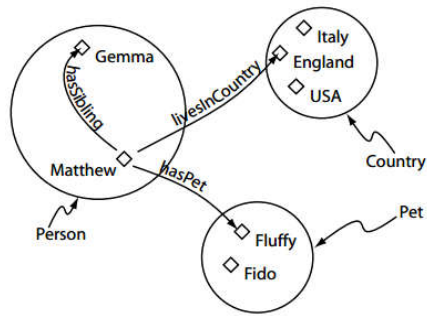


Figura 2.15 - Domínio e intervalo de valores

Propriedades em OWL podem ser usadas para criar restrições. Basicamente, funciona como um requisito para que indivíduo pertença a uma classe. Por exemplo, no caso anterior, *Matthew hasSibling Gemma*. É possível definir uma classe que tem como requisito a propriedade *hasSibling* e portanto só poderão ser membros dessa classe indivíduos que apresentem esta propriedade.

São exemplos de restrições que definem uma classe:

- Indivíduos que participam de pelo menos uma relação *hasSibling*;
- Indivíduos que participam de pelo menos uma relação *hasSibling* a indivíduos que sejam "homens";
- Indivíduos que só tem relações *hasSibling* a indivíduos que sejam "mulheres";
- Indivíduos que tem mais de três relacionamentos *hasSibling*;

As restrições em OWL definem as classes anônimas (não nomeadas) que contém apenas indivíduos que atendam as restrições e podem ser descritas de três maneiras diferentes: de quantificador, de cardinalidade e de valor.

As restrições de quantificador são classificadas em dois tipos: de quantificador existencial e quantificador universal.

- As restrições de quantificador existencial descrevem a classe de indivíduos que participam de **pelo menos um** relacionamento ao longo de uma propriedade específica para indivíduos que são membros de uma classe específica (nomeada). Os exemplos 1 e 2 acima são exemplos de restrições de quantificador existencial.

- As restrições de quantificador universal por sua vez descrevem a classe de indivíduos que participam **apenas de** relacionamentos ao longo de uma propriedade específica para indivíduos que são membros de uma classe única. O exemplo 3 acima atende esta restrição.

Restrições que descrevem classes de indivíduos que tem uma quantidade **mínima**, **máxima** ou **exatamente** de relacionamentos com outros indivíduos ou valores do tipo de dados são conhecidas como restrições de cardinalidade. O exemplo 4 acima representa uma restrição de cardinalidade mínima.

Restrições de valor descrevem o conjunto de indivíduos que se relacionam com um único indivíduo específico de uma classe nomeada específica. Exemplo desta restrição seria: os indivíduos que tem uma relação *livesInCountry* com o indivíduo da classe *Country* chamado *England*.

2.1.2 Classes Primitivas e Definidas (Condições Necessárias e Suficientes)

Na seção anterior, foram apresentados os conceitos de restrições do ponto de vista das propriedades em OWL. Caso as restrições fossem atendidas, um indivíduo faria parte, então, de uma classe anônima (primitiva). O que isso quer dizer? Do ponto de vista da classe, o indivíduo atendeu um requisito necessário para ser membro, mas não pode ser definido com um membro, pois atender somente condições **necessárias** não é o **suficiente**.

Surge então o conceito de restrições de classe. São dois tipos de condições a serem atendidos: Condições Necessárias e Condições Necessárias e Suficientes.

Se uma classe “A” é descrita usando condições necessárias, então, se um indivíduo é um membro da classe “A”, então este indivíduo deve satisfazer as condições de “A”. Porém, não é possível afirmar que qualquer indivíduo que satisfaça tais condições também seja membro da classe. Na figura 2.16 é ilustrada este conceito.

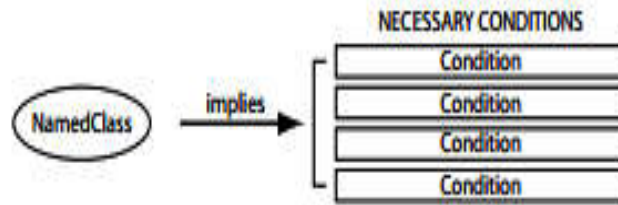


Figura 2.16 - Condições necessárias

No entanto, se uma classe "A" agora é descrita usando condições necessárias e suficientes, então podemos afirmar que um indivíduo, membro da classe "A" satisfaz as condições de "A" e qualquer indivíduo (aleatório) que também satisfaça será membro da classe. Na figura 2.17 é ilustrada esta particularidade.

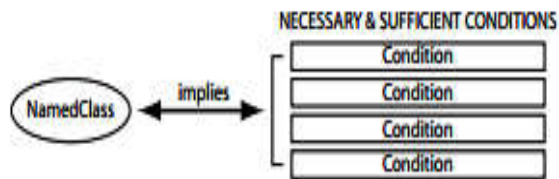


Figura 2.17 - Condições necessárias e suficientes

Classes descritas apenas com condições necessárias são conhecidas como classes primitivas (equivalentes). As classes definidas (nomeadas) são descritas com condições necessárias e suficientes.

2.1.3 Padrões de Projeto

Padrões de projeto utilizam conceitos e propriedades anteriormente citadas basicamente para refinar as descrições das classes, ou seja, não existe uma propriedade exclusiva para sua conceituação. São criados para resolver problemas de modelagem que ocorrem repetidas vezes. Assim como programação orientada a objetos, que soluciona esse caso, os padrões de projeto em OWL obedecem ao mesmo conceito.

Como exemplo, imagina-se uma ontologia intitulada como Pizza, ao qual se cria um padrão que descreve os temperos para as coberturas deste prato. A partição é responsável por restringir o intervalo de valores possíveis para todas as coberturas levando em consideração se

os temperos são leves, médios ou picantes. Esse tipo de padrão de projeto é conhecido como **Partição de Valores**.

Para se criar uma partição de valores, devem ser seguidos os seguintes passos:

- Criar uma classe para representar a partição, exemplo: *SpicinessValuePartition*;
- Criar subclasses representando os valores possíveis: *Mild*, *Medium* e *Hot*;
- Fazer a disjunção das subclasses;
- Fornecer um axioma (sentença verdadeira) de cobertura para inferir dados do tipo de valores;
- Criar uma propriedade de objeto para a partição, exemplo: *hasSpiciness*;
- Fazer da propriedade anterior funcional;
- Definir o intervalo (alcance) da propriedade como a classe *ValuePartition*. Ou seja, a faixa para *hasSpiciness* seria *SpicinessValuePartition*.

Por padrão, em OWL, as subclasses são criadas de tal maneira que elas se sobrepõem, ou seja, significa que um indivíduo pode pertencer a várias classes distintas. E tornar as classes disjuntas garante que isso não ocorra. Em suma, uma pizza não pode ter dois tipos de temperos, nem uma combinação entre eles.

Como parte do padrão de partição de valores, pode-se usar um axioma de cobertura para definir as classes.

Por exemplo, imagina-se o caso anterior e suponha-se que há um axioma de cobertura que especifica: a classe *SpicinessValuePartition* é coberta pelas classes *Mild*, *Medium* e *Hot*.

- Isso significa que um indivíduo, membro de *SpicinessValuePartition*, obrigatoriamente é um indivíduo de *Mild* e/ou *Medium* e/ou *Hot*.

Como foi solicitada, no exemplo anterior, a disjunção das subclasses, então:

- Um indivíduo, membro de *SpicinessValuePartition* é também membro de *Mild* ou *Medium* ou *Hot*.

Nas figuras 2.18 e 2.19 a seguir, ilustra-se o exemplo anterior além de mostrar a diferença básica da utilização ou não de axiomas de cobertura em se definir uma classe. Nota-se que no primeiro caso (sem axioma de cobertura) um indivíduo, membro da classe *SpicinessValuePartition* poderá não ser membro das suas subclasses, ou seja *Mild*, *Medium* e *Hot*.

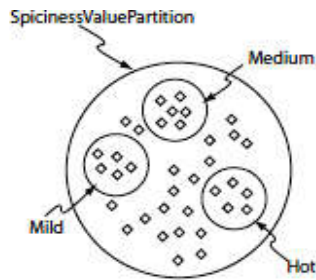


Figura 2.18 - Sem axioma de cobertura

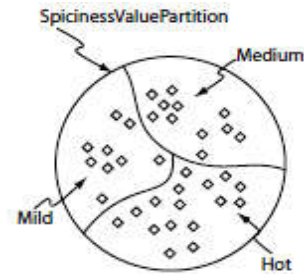


Figura 2.19 - Com axioma de cobertura

2.1.4 Classes Complexas: Intersecção e União

As superclasses também podem assumir descrições complexas para suas construções. Basicamente, utilizam-se operadores lógicos em conjuntos de descrições de classes mais simples para tal processo. Os operadores são denominados intersecção (\cap) e união (\cup).

Uma classe de intersecção é composta pelos indivíduos resultantes da combinação de duas ou mais classes usando o operador lógico “E” (AND). A figura a seguir traz um exemplo e determina que o espaço mais escuro (ao centro) seria a composição da classe resultante.

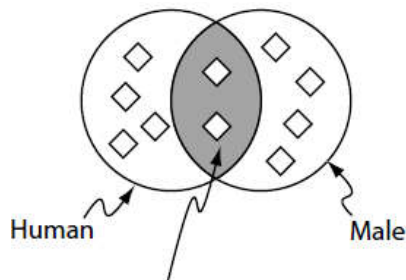


Figura 2.20 - Intersecção de classes

A semântica em OWL para a classe resultante de uma intersecção de classes, nesse caso, significa que a classe anônima descrita é uma subclasse de *Human* e uma subclasse de *Male*.

Já a classe resultante da união é composta pela combinação de indivíduos de duas ou mais classes que utilizam o operador lógico “OU” (OR). O resultado está ilustrado na figura 2.21 a seguir. Percebe-se que a classe resultante é toda a área pontilhada.

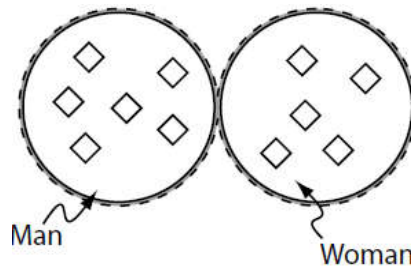


Figura 2.21 - União de classes

A classe anônima descrita a partir da união das classes acima seria composta pelos indivíduos pertencentes à classe *Man* ou a classe *Woman* (ou em ambos). Além disso, poderia também ser usada na descrição de outra classe, por exemplo, *Person* seria a classe equivalente da união das classes *Man* e *Woman*.

2.2 O Suporte do Ambiente Protégé para a linguagem OWL

A seção anterior deste estudo conceituou algumas das propriedades e características mais importantes da linguagem em OWL. Esta seção mostra como é feito este processo em um ambiente de criação e editoração de ontologias utilizando a linguagem OWL.

A ferramenta Protégé é um editor de ontologias, escrito em Java, de código aberto, grátis e um sistema de aquisição de conhecimento. Em sua versão mais recente, Protégé 4.3 apresenta uma interface de usuário que pode ser configurável permitindo um ambiente perfeito para iniciantes e experientes. Ela permite que vários outros projetos sugiram plugins (módulo de extensão usado para adicionar pequenas funções a softwares maiores). Protégé tem recentemente mais de 200.000 usuários registrados e é considerada a ferramenta líder de engenharia ontológica.

Protégé está em constante desenvolvimento na Universidade de Stanford (Estados Unidos) em colaboração com a Universidade de Manchester (Inglaterra).

Usa-se como modelo de conceituação a ontologia "Pizza", brevemente citada na seção 2.1.3 deste trabalho, ilustrando alguns exemplos utilizando a ferramenta Protégé em sua versão 4.3.

2.2.1 Hierarquia de Classes em Protégé

A criação de superclasses e subclasses é ilustrada na figura 2.22.



Figura 2.22 - Hierarquia de classes em Protégé

A disposição das classes de forma indentada e as setas ao lado do nome das classes ajudam a distinguir as superclasses das respectivas subclasses. Uma particularidade importante em OWL é a superclasse *Thing* ao topo da hierarquia. Todas as ontologias criadas no ambiente Protégé terão como a superclasse geral a classe *Thing*, o que quer dizer, então, que tudo é subclasse de *Thing*.

2.2.2 Ilustração de Propriedades e Restrições

Em OWL pode-se usar restrições para descrever uma classe. No exemplo abaixo é ilustrado um caso simples usando restrições.



Figura 2.23 - Restrição existencial em Protégé

Neste caso, é utilizado uma restrição existencial para descrever a classe *CheeseyPizza*. Formalmente está definido que, para que um indivíduo seja uma *CheeseyPizza* é necessário que ele seja uma "Pizza" e "tenha algumas coberturas de queijo". Abaixo é definido que "tem alguma base de pizza", isso acontece porque a classe "Pizza" foi descrita usando uma condição necessária do tipo: Para que algo seja uma Pizza, é necessário que tenha alguma base de pizza, logo, ao afirmar que *CheeseyPizza* é uma "Pizza" automaticamente em sua descrição aparece a classe anônima dos indivíduos que tem alguma base de pizza.



Figura 2.24 - Restrição de cardinalidade em Protégé

Na figura 2.24, implica-se que uma *InterestingPizza* é uma classe equivalente dos indivíduos que são membros da classe *Pizza* e também dos indivíduos que tenham pelo menos três relações via propriedade *hasTopping* com outros indivíduos distintos. Caracteriza-se então uma restrição de cardinalidade mínima.



Figura 2.25 - Restrição de valores em Protégé

Uma restrição de valor (ilustrada na figura 2.25) descreve o conjunto de indivíduos que tem pelo menos um relacionamento via propriedade específica para um indivíduo

específico. Neste caso, a descrição de *MozzarellaTopping* implica que ela faz parte de uma classe anônima de indivíduos que tem pelo menos um relacionamento com o indivíduo *Italy* via propriedade *hasCountryOfOrigin*.

2.2.3 Condições Necessárias e Suficientes

Condições necessárias são as restrições, requisitos mínimos, de classes impostas aos indivíduos poderem ser membros da classe. Lembre-se que apenas condições necessárias não definem que qualquer indivíduo que atenda aos seus requisitos seja um membro da classe.



Figura 2.26 - Condições necessárias em Protégé

A janela de descrição de classe, ilustrada na figura 2.26, descreve restrições para a classe *CheeseyPizza*. O cabeçalho “Superclasses” é utilizado para determinar tais restrições, o que significa que *CheeseyPizza* será a superclasse dos indivíduos que atendam as duas restrições impostas por ela. Implicitamente, uma subclasse anônima será criada com tais indivíduos. Formalmente dizendo, para que algo seja uma *CheeseyPizza* é necessário que seja uma “Pizza” e tenha alguma “cobertura de queijo”.



Figura 2.27 - Condições necessárias e suficientes em Protégé

A descrição de classe agora, ilustrada na figura 2.27, está definida com condições necessárias e suficientes. O cabeçalho *Equivalent classes* indica que qualquer indivíduo que atender a restrição será um membro da classe *CheeseyPizza*.

O cabeçalho *Inherited anonymous classes* representam as restrições e/ou classes herdadas. Como a classe “Pizza” foi descrita usando uma condição necessária, “Para que seja uma pizza é necessário que um indivíduo tenha alguma base de pizza”, automaticamente esta restrição é herdada por *CheesyPizza*.

2.2.4 Padrões de Projeto: Axiomas de Cobertura

Como dito anteriormente, os padrões de projeto são utilizados para refinar as descrições de classes. Na figura abaixo é ilustrada um tipo de restrição que formaliza um padrão de projeto utilizando axioma de cobertura.

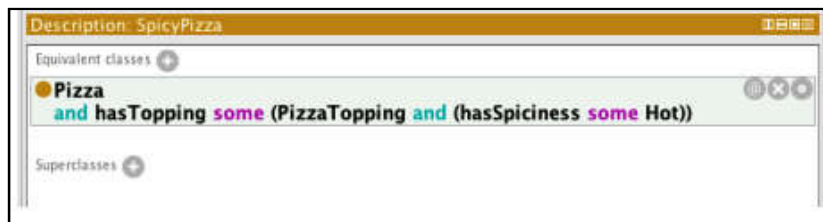


Figura 2.28 - Restrição utilizando axioma de cobertura em Protégé

A descrição de *SpicyPizza* acima define uma condição necessária e suficiente, formalmente dizendo: “Para que algo seja uma *SpicyPizza* é necessário e suficiente que seja uma *Pizza* e tenha pelo menos uma cobertura que tenha algum tempero picante”.

2.2.5 Utilização de Raciocinadores

Uma das principais características das ontologias criadas em OWL-DL são os mecanismos de raciocínio, ou raciocinadores. Eles são responsáveis por processar as informações inseridas na ontologia e verificar inconsistências bem como computar a hierarquia de classes.

A hierarquia de classes construídas manualmente é conhecida como hierarquia afirmada. Por sua vez, a hierarquia computada pelo raciocinador é chamada de hierarquia inferida (deduzida). Na figura abaixo é ilustrada esta diferença.

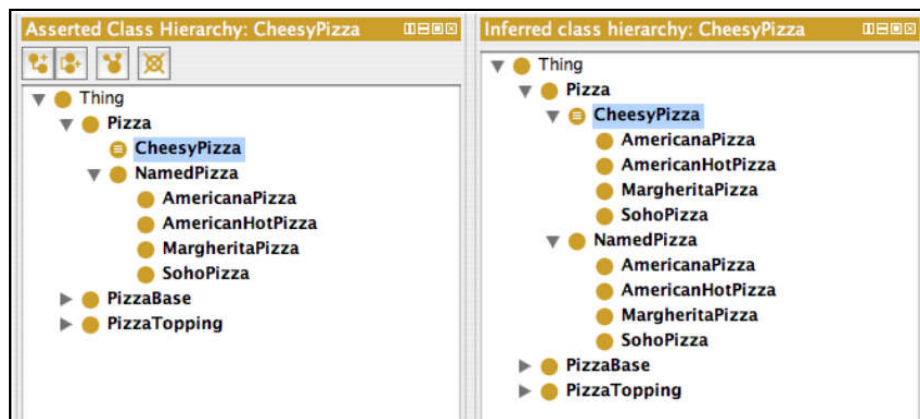


Figura 2.29 - Hierarquia afirmada e hierarquia inferida

Percebe-se, na figura acima, que a utilização do mecanismo de raciocínio encontrou classes que atendem aos requisitos de *CheesyPizza* e como a descrição da mesma estava definida com condições necessárias e suficientes, automaticamente as classes se tornam subclasses de *CheesyPizza*.

Outra tarefa do raciocinador é encontrar inconsistências na ontologia. Uma inconsistência se refere a uma sentença falsa dentro de uma classe. Basicamente o raciocinador verifica, com base nas descrições, se é ou não é possível para as classes ter todas as instâncias.

Na figura 2.30 a seguir, é mostrado um exemplo de inconsistência. Trata-se de uma classe criada exclusivamente para testar o raciocinador. Classes criadas para testar consistência em ontologias são conhecidas como Classes Sondas.



Figura 2.30 - Classe Sonda

Intuitivamente, sabe-se que alguma cobertura não pode, simultaneamente, ser queijo e vegetal. Ao utilizar o raciocinador, teremos o resultado conforme a figura 2.31.

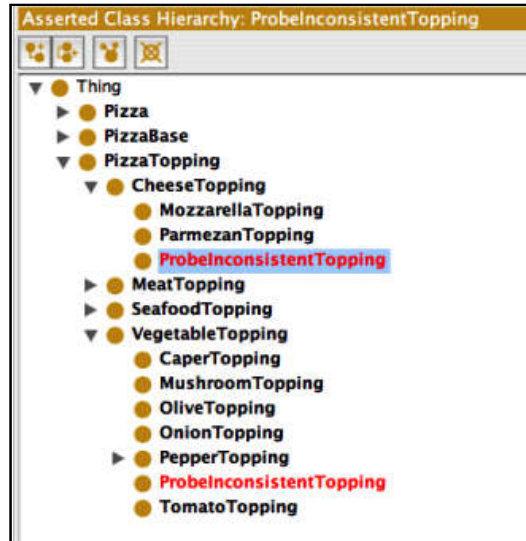


Figura 2.31 – Inconsistência

ProbelInconsistentTopping aparece hierarquicamente como sendo uma subclasse de *CheeseTopping* e também *VegetableTopping* bem como destacado na cor vermelha para provar a inconsistência.

Como dito anteriormente, "intuitivamente", sabe-se que algo não pode ser queijo e vegetal ao mesmo tempo. Mas, esta sentença está sendo considerada com base no nome dos indivíduos. Ontologia, em OWL, não conhece nomes, ou melhor, não leva em consideração nomes de indivíduos para provar uma inconsistência. Então, como foi provada a inconsistência anterior? Simples, *VegetableTopping* e *CheeseTopping* foram definidas para serem disjuntas entre si, o que implica que os membros de uma não pode ser membro da outra.

Este é um exemplo claro da importância em se definir axiomas em ontologia. Por padrão, em OWL, as classes criadas se sobrepõem até que sejam definidas como disjuntas entre si. Resultados inesperados podem surgir caso alguma informação seja definida erroneamente.

2.2.6 Pressuposto de Mundo Aberto: Axiomas de Fechamento

Não se pode assumir, em OWL, que algo não existe até que seja explicitamente confirmado, este é o conceito do Pressuposto de mundo aberto. Raciocinadores em OWL-DL utilizam este conceito para tratar as inconsistências e hierarquia de classes.

Conforme um exemplo em nossa ontologia *Pizza*, a classe *MargheritaPizza* está descrita da seguinte maneira: Para que algo seja uma *MargheritaPizza* é necessário que o indivíduo pertença a classe *NamedPizza*, tenha alguma cobertura de mozzarella e também alguma cobertura de tomate. Por causa do pressuposto de mundo aberto, presume-se que *MargheritaPizza* poderia ter outras coberturas.

Como *MargheritaPizza* está descrita usando condições necessárias, o raciocinador não colocaria nenhuma subclasse para ela caso algum indivíduo atendesse tais requisitos. Para resolvermos este caso, devemos usar axiomas de fechamento para a classe *MargheritaPizza*.



Figura 2.32 - Axioma de fechamento

O axioma de fechamento é caracterizado pelas restrições existenciais e sua respectiva restrição universal abaixo. Como assim? Caso a descrição de *MargheritaPizza* contivesse apenas:

- *NamedPizza*
- *hasTopping only (MozzarellaTopping or TomatoTopping)*

Pode acontecer que um indivíduo, membro da classe *NamedPizza* sem qualquer tipo de cobertura atendesse a restrição acima, portanto pertenceria a uma classe anônima de *MargheritaPizza* o que causaria um resultado inesperado.

Agora se definirmos restrições existenciais juntamente com a universal, afirmamos que é necessário que um indivíduo tenha algum tipo de cobertura de mozzarella e algum tipo de cobertura de tomate e somente eles.

3. Desenvolvimento do Trabalho Proposto

3.1 Ontologias para aplicação no domínio de Nutrição Humana

O conhecimento conceitual de um domínio de estudo é importante para a construção de ontologia, pois favorece a caracterização dos principais elementos relacionados a este domínio. Desta forma, ao buscar uma definição sobre o termo nutrição, ROSSI, 2012 expõe: “Define-se a nutrição como ciência que estuda todos os processos por meios dos quais o organismo recebe, utiliza e elimina os nutrientes ingeridos”. Mais do que isso, também salienta que nutrição envolve:

... Aspectos desde a seleção e escolha dos alimentos, passando pelo contexto de vida do indivíduo até sua relação com a saúde e doença, sem esquecer os aspectos fisiológicos. Dessa forma inclui implicações sociais, econômicas, culturais e psicológicas relacionadas aos alimentos e à alimentação. (ROSSI, 2012)

Pode-se extrair deste conceito, elementos como nutrientes, alimentos, indivíduo, saúde, doença e aspectos fisiológicos: implicações sociais, econômicas, culturais e psicológicas.

A partir disso podem-se enxergar duas principais relações: 1. Nutrientes, alimentos, indivíduo, saúde e doença; 2. Aspectos fisiológicos com escolhas alimentares.

Como fora exemplificado na segunda seção deste presente estudo, os relacionamentos entre os elementos, neste caso entidades, são a chave para a construção de uma ontologia. São a partir deles que podemos realizar consultas em um ambiente sistemático como o Protégé, além do que ele próprio poderá revelar relações implícitas entre as entidades, constituindo, portanto, as inferências.

A fim de construir uma ontologia bem estruturada, o processo de extração dos elementos pode passar por refinamentos. Por exemplo, toma-se como base o relacionamento 1, acima: Nutrientes, alimentos, indivíduo, saúde e doença. Para deixar mais claro esta relação, realoca-se e refina-se da seguinte maneira:

- Doença – Sintoma – Causa – Nutriente – Alimento

Tem-se então uma relação hierárquica entre as entidades que deixa mais claro a intenção de pesquisa. De acordo com o CESAR, a seguir, são mostradas algumas das principais doenças nutricionais, obedecendo a relação hierárquica acima.

1. Anemia, causada por insuficiência do nutriente Ferro no sangue, apresenta sintomas de fadiga, palidez, falta de disposição entre outros. Alimentos como carnes vermelhas, tais como fígado, rim, coração, bem como carne de aves e peixes são as principais fontes de ferro para o organismo e controlam a causa da doença;
2. Raquitismo, causada por insuficiência do nutriente Cálcio e da Vitamina D, apresenta sintomas de ossos fracos e quebradiços. Alimentos como óleo de peixes e derivados do leite controlam a causa da doença;
3. Escorbuto, causada por insuficiência de Vitamina C, apresenta sintomas de hemorragias nas gengivas, dentes soltos, cansaço entre outros. Alimentos como Acerola e Laranja são ricos em Vitamina C e controlam a causa da doença;
4. Hiperlipidemia, causada pelo excesso de Gorduras, tais como Colesterol ruim e Triglicérido, apresenta alguns sintomas, dentre eles forte dor abdominal. Alimentos como o peixe e aveia controlam a quantidade de Lípidios no organismo.
5. Obesidade, causada pelo excesso de gorduras, açúcares e proteínas no organismo. Sintomas estão relacionados ao ganho de peso, dores nas articulações, circunferência da cintura. Alimentos ricos em fibras, tais como os vegetais, legumes e algumas frutas bem como alimentos que diminuam o colesterol ruim combatem as causas da doença.
6. Marasmo, causada pela insuficiência proteico-calórica por parte de crianças recém-nascidas, apresenta sintomas de baixa estatura, perda de peso, tecido muscular e subcutâneo. O leite materno é a principal fonte de alimentação para esta causa.

A partir destes dados pode ser construída uma pequena ontologia, para ser mapeada em construções da linguagem OWL. Os elementos da relação 1 acima nos esclarecem não apenas as entidades em OWL, mas também algumas propriedades importantes nos relacionamentos, são elas: temSintomas, temCausa, temFaltaDe, temExcessoDe, encontraEm:

- Doenças que podem contribuir para outras doenças como é o caso da obesidade que pode ocasionar doenças cardíacas;
- Fatores de risco como, por exemplo, a genética, gravidez, locais que podem influenciar nas manifestações de certas doenças.

Contudo, a proposta deste trabalho é analisar a capacidade de realização de consultas em uma ontologia com informações não estruturadas.

Para isso no presente estudo foi utilizada a seção do livro *Understanding Nutrition* de *Ellie Whitney* e *Sharon Rady Rolfes*, que aborda escolhas nutricionais. Neste livro, que está em sua 12ª Edição, é apresentado um conteúdo bem elaborado no domínio de nutrição humana, discriminando assuntos desde a conceituação dos principais nutrientes, presentes na alimentação básica, até planejamentos de dietas saudáveis para todos os tipos de pessoas, abrangendo todo o ciclo de vida, ou seja, desde a infância até a vida idosa.

Considerando que todo tipo de publicação tem seus conceitos introdutórios notadamente no início e que essas informações normalmente são visões gerais de um determinado domínio, fora abordado então a seção *Food Choices* do primeiro capítulo do livro acima, intitulado *An Overview of Nutrition* para a extração.

O nível de abstração das informações presentes neste capítulo se mostram desafiantes para a construção da ontologia Nutrição, pois uma das principais preocupações ao construir uma ontologia trata-se da inconsistência dos dados.

A seção a seguir mostra a representação das entidades e propriedades presentes na ontologia Nutrição.

3.2 Processo utilizado e resultados obtidos

O processo adotado, após a escolha do texto a ser representado na linguagem OWL, foi baseado nos seguintes passos: (1) reescrita da sentença em linguagem natural para evidenciar as relações relevantes da sentença; (2) representação das hierarquias de classes relevantes; (3) representação das hierarquias de propriedades relevantes; e (4) representação dos relacionamentos com foco nas consultas que poderão ser realizadas. A seguir são descritos os quatro passos do processo.

3.2.1 - Reescrita da sentença em linguagem natural para evidenciar as relações relevantes

A transição da extração do texto original, em linguagem natural para sua representação na linguagem OWL requer uma etapa intermediária de reescrita do texto original em linguagem natural. A seguir, nas tabelas 1, 2 e 3, são apresentados exemplos, nos quais é possível comparar o texto original com o texto adaptado (reescrito). No texto original estão sublinhadas as partes do texto cuja extração de conceitos e relações seria inviável, sem a etapa intermediária de reescrita. Esta dificuldade decorre da grande flexibilidade de representação que existe em uma linguagem natural, como por exemplo, o trecho: “*decide what to eat, when to eat, and even whether to eat*”.

Tabela 1 – Transição do texto original – exemplo 01

Texto Original	<i>People <u>decide what to eat, when to eat, and even whether to eat</u> in highly personal ways, <u>often based on</u> behavioral or social motives <u>rather than on an awareness of nutrition’s importance to health.</u></i>
Texto adaptado	<i>Food Choices is decided based on highly personal ways, behavioral and social reasons. Food Choices isn’t decided based on nutritional performance.</i>

Tabela 2 – Transição do texto original – exemplo 02

Texto Original	<i><u>Among the strongest influences on food choices are ethnic heritage and tradition.</u> <u>Some people cannot eat when they are emotionally upset. Others may eat in response to a variety of emotional stimuli.</u> Sometimes people select certain foods and supplements <u>that they believe will improve their physical appearance and avoid those they believe might be detrimental.</u></i>
Texto Adaptado	<i>Ethnic heritage and Tradition has strongest influence on food choices. Emotions, weight and image of body has influence on food choices.</i>

Tabela 3 – Transição do texto original – exemplo 03

Texto Original	<i>As you might expect, the number one reason people choose foods is taste - they like certain flavors. Two widely shared preferences are for the sweetness of sugar and the savoriness of salt. Liking high-fat foods also appears to be a universally common preference.</i>
Texto Adaptado	<i>Flavors of food are the main reason of food choices. Sweetness of sugar, savoriness of salt and high fat foods is flavors preferences of general cooking.</i>

3.2.2 - Representação das hierarquias de classes relevantes

Neste passo do processo, são definidas as classes raízes das hierarquias de classes de conceitos, bem como as subclasses que fazem parte de cada hierarquia. Nas tabelas 4, 5 e 6, são ilustrados exemplos do texto adaptado e das hierarquias de classes resultantes. As classes de conceitos são selecionadas em função de sua relevância com o tema escolhas nutricionais. É possível observar que, em alguns exemplos, o texto adaptado contém somente as subclasses, a partir do qual são extraídas as superclasses (classes mais genéricas), como por exemplo, “reasons” a partir de “behavioral reasons” e de “social reasons”. O mesmo ocorre na tabela 6, com a classe genérica “social iterations” que não aparece no texto adaptado.

Tabela 4 – Representação em OWL – exemplo 01

Texto adaptado	<i>Food Choices is decided based on highly personal ways, behavioral and social reasons.</i>
Representação OWL	<p>Hierarquia de Classes</p> <ul style="list-style-type: none"> • <i>PersonalWays: Highly_PersonalWays</i> • <i>Reasons: Behavioral_Reasons, Social_Reasons</i>

Tabela 5 – Representação em OWL – exemplo 02

Texto Adaptado	<i>Flavors of food are the main reason of food choices. Sweetness of sugar, savoriness of salt and high fat foods is flavors preferences of general cooking.</i>
Representação OWL	Classes <ul style="list-style-type: none"> • <i>Flavors_Food: SweetnessOf_Sugar, SavorinessOf_Salt, HighFat_Foods, HotPeppers, CurrySpices</i> • <i>Cooking: General_Cooking</i>

Tabela 6 – Representação em OWL – exemplo 03

Texto Adaptado	<i>Companionship is enjoyed with social meals. Socialization stimulates social meals. Go out with friends is fun with social meals. Social events are frequently occurring with social meals. Hospitality stimulates sharing food. Social customs stimulates to accept variety of foods.</i>
Representação OWL	Hierarquia de Classes <ul style="list-style-type: none"> • <i>SocialMeals: SharingFood, VarietyFood;</i> • <i>SocialInteractions: Companionship, GoOut_WithFriends, Socialization, SocialEvents, Hospitality, SocialCustoms.</i>

3.2.3 - Representação das hierarquias de propriedades relevantes

Neste passo do processo, são definidas as propriedades raízes das hierarquias de propriedades, bem como as subpropriedades que fazem parte de cada hierarquia. Nas tabelas 7, 8 e 9, são ilustrados exemplos do texto adaptado e das hierarquias de propriedades resultantes. As propriedades são selecionadas em função de sua relevância com o tema escolhas nutricionais. Da mesma forma que ocorre com as classes de conceitos, em alguns exemplos, o texto adaptado contém somente as subpropriedades, a partir do qual são extraídas as propriedades (mais genéricas), como por exemplo, “*isReasonOf*” e “*mayInfluence*”, nas tabelas 7 e 9. No

tabela 8, ocorre uma situação na qual a propriedade mais genérica “stimulates” aparece no texto adaptado.

Tabela 7 – Representação em OWL – exemplo 04

Texto Adaptado	<i>Flavors of food are the main reason of food choices.</i>
Representação OWL	Propriedades <ul style="list-style-type: none"> • <i>isReasonOf: isTheMain_ReasonOf</i>

Tabela 8 – Representação em OWL – exemplo 05

Texto Adaptado	<i>Companionship is enjoyed with social meals. Socialization stimulates social meals. Go out with friends is fun with social meals. Social events are frequently occurring with social meals. Hospitality stimulates sharing food. Social customs stimulates to accept variety of foods.</i>
Representação OWL	Hierarquia de Propriedades <ul style="list-style-type: none"> • <i>stimulates: isEnjoyedWith, isFunWith, areFrequentlyOccuringWith, stimulatesToAccept;</i>

Tabela 9 – Representação em OWL – exemplo 06

Texto Adaptado	<i>Diary Food Choices may little influence of people health. Food Choices repeated long time may much influence people health.</i>
Representação OWL	Hierarquia de Propriedades <ul style="list-style-type: none"> • <i>mayInfluence: mayInfluenceLittle, mayInfluenceMuch;</i>

3.2.4 - Representação dos relacionamentos com foco nas consultas potenciais

O objetivo de se fazer uma consulta no ambiente Protégé é encontrar uma classe e/ou indivíduo que atenda ao requisito proposto no campo de pesquisa. Pode-se dizer que a consulta em si “é uma classe” e os resultados obtidos “são subclasses” da consulta, pois atendem ao requisito.

Deve-se perceber que um relacionamento em OWL é ilustrado pela seguinte regra de encadeamento: E1 – p – E2 onde, “E1” representa a Entidade 1, que pode ser uma classe ou indivíduo, “p” representa Propriedade, responsável por fazer a ligação entre E1 e E2 e por fim Entidade 2 representada por “E2” que pode ser uma classe mas não um indivíduo.

Desta forma, as consultas encontram “E1” relacionadas por “p – E2”.

A seguir são ilustradas cinco situações distintas de consultas.

3.2.4.1 - Consulta com foco no uso do operador “not”

O operador “not” tem por característica a negação de uma relação e não de apenas um elemento, ou seja, a sentença “not (p – E2)” é verdadeira. Na tabela 10 são mostrados os relacionamentos e consultas, a partir do texto adaptado, caracterizando relacionamento sem negação e com negação. Na figura 3.1 é mostrado o resultado das consultas sem o operador de negação, e na figura 3.2 idem para o operador de negação. O objetivo deste exemplo é caracterizar a utilidade do operador “not” para realizar consultas.

Tabela 10 – Representação em OWL - exemplo 07

Texto adaptado	<i>Food Choices is decided based on highly personal ways, behavioral and social reasons.</i> <i>Food Choices isn't decided based on nutritional performance.</i>
Representação OWL	Relacionamentos <ul style="list-style-type: none"> • <i>Highly_PersonalWays isBase_toDecideAbout some FoodChoices.</i> • <i>{Behavioral_Reasons, Social_Reasons} isBase_toDecideAbout some FoodChoices.</i> • <i>NutritionalPerformance not (isBase_toDecideAbout some FoodChoices)</i>
Consultas	<ul style="list-style-type: none"> • <i>? isBase_toDecideAbout some FoodChoices</i> • <i>? not (isBase_toDecideAbout some FoodChoices)</i>

- Resultados Obtidos

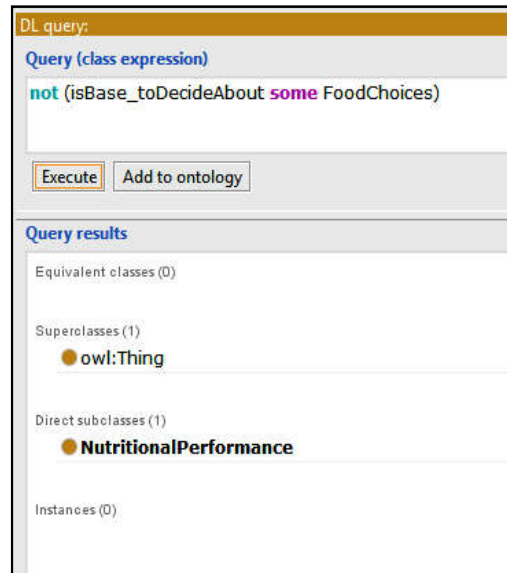
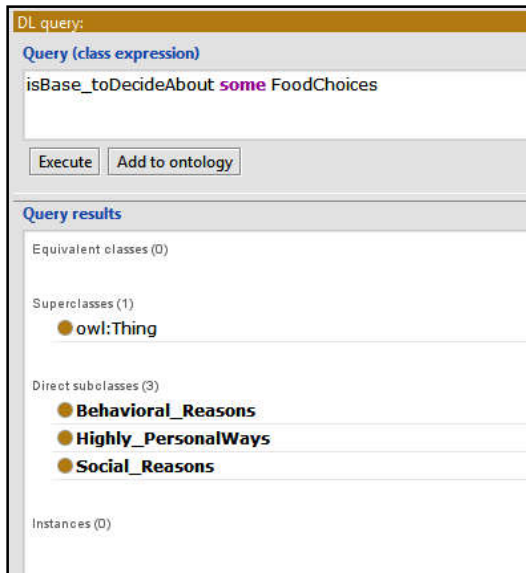


Figura 3.1 - Consulta: "isBase_toDecideAbout some FoodChoices"
 Figura 3.2 - Consulta: "not (isBase_toDecideAbout some FoodChoices)"

Com base nesta característica, as consultas também devem ser escritas desta forma e, portanto apresentam-se como resultados para este tipo de pesquisa:

- Super Classes e Sub Classes se E1 for uma Super Classe;
- Sub Classe se E1 for uma Sub Classe, como é mostrada na figura 3.2 acima.

3.2.4.2 - Consulta com foco em respostas contendo um conjunto de Entidades 1

A representação de um relacionamento genérico é: “Entidade1 Propriedade Entidade2”. Se existirem vários relacionamentos que tem em comum a propriedade e a entidade 2, a resposta retornada será o conjunto de entidades 1 formado a partir da entidade 1 de cada um desses relacionamentos. Na tabela 11, é ilustrada esta situação. A sintaxe “*{EthnicHeritage, Tradition} hasStrongest_influenceOn some FoodChoices*” é uma forma compacta de representar os relacionamentos: (a) *EthnicHeritage hasStrongest_influenceOn some FoodChoices*”, e (b) *Tradition hasStrongest_influenceOn some FoodChoices*”.

As entidades1 que retornam como respostas às consultas dependem do nível hierárquico da propriedade utilizada:

- Se “p” for uma propriedade que tem pelo menos uma subpropriedade, as respostas deverão englobar todas as entidades que tem uma relação com a propriedade e com suas respectivas subpropriedades, como é ilustrada na figura 3.3.
- Se “p” for uma propriedade que não tem nenhuma subpropriedade, as respostas deverão englobar somente as entidades que tem relação com a respectiva propriedade. Na figura 3.4 é mostrado este exemplo.

Tabela 11 – Representação em OWL – exemplo 08

Texto Adaptado	<i>Ethnic heritage and Tradition has strongest influence on food choices. Emotions, weight and image of body has influence on food choices.</i>
Representação OWL	Hierarquia de Propriedades <ul style="list-style-type: none"> • <i>hasInfluenceOn: hasStrongest_influenceOn</i>
	Relacionamentos <ul style="list-style-type: none"> • <i>{EthnicHeritage, Tradition} hasStrongest_influenceOn some FoodChoices;</i> • <i>Emotions hasInfluenceOn some FoodChoices;</i> • <i>{Weight_BodyHuman, Image_BodyHuman} hasInfluenceOn some FoodChoices.</i>
Consultas	<ul style="list-style-type: none"> • <i>? hasInfluenceOn some FoodChoices</i> • <i>? hasStrongest_influenceOn some FoodChoices</i>

- Resultados Obtidos

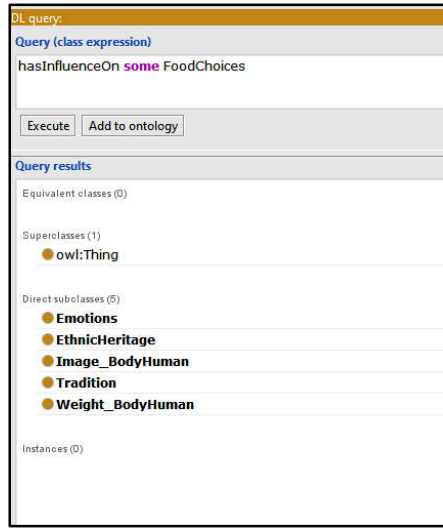


Figura 3.3 – Consulta: “hasInfluenceOn some FoodChoices”

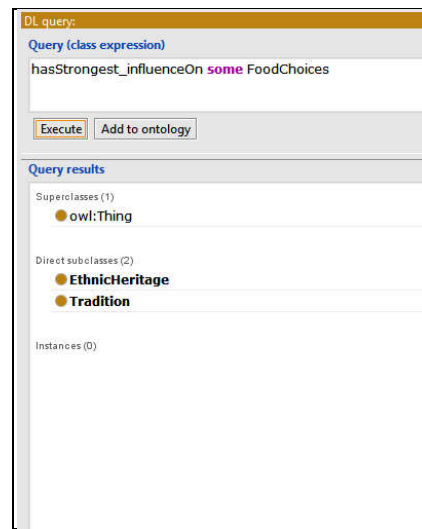


Figura 3.4 – Consulta: “hasStrongest_influenceOn some FoodChoices”

3.2.4.3 - Consulta com foco na hierarquia de classes

Relacionamentos condicionados a uma mesma propriedade, mas que a entidade 2 for uma hierarquia de classes, terá resposta as entidades 1 condicionadas a hierarquia de entidade 2:

- Se entidade 2 for superclasse, as respostas para as consultas serão entidades 1 que tiverem relacionamentos com a superclasse de entidade 2 e suas respectivas subclasses, conforme ilustrado na figura 3.7;
- Se entidade 2 for subclasse, as respostas para as consultas serão entidades 1 que tiverem relacionamentos somente com a subclasse de entidade 2, conforme ilustrado na figura 3.8;

Tabela 12 – Representação em OWL – exemplo 09

Texto Adaptado	<i>Companionship is enjoyed with social meals. Socialization stimulates social meals. Go out with friends is fun with social meals. Social events are frequently occurring with social meals. Hospitality stimulates sharing food. Social customs stimulates to accept variety of foods.</i>
Representação	Hierarquia de Classes

OWL	<ul style="list-style-type: none"> • <i>SocialMeals: SharingFood, VarietyFood;</i> • <i>SocialInteractions: Companionship, GoOut_WithFriends, Socialization, SocialEvents, Hospitality, SocialCustoms</i>
	<p>Hierarquia de Propriedades</p> <ul style="list-style-type: none"> • <i>stimulates: isEnjoyedWith, isFunWith, areFrequentlyOccuringWith, stimulatesToAccept</i>
	<p>Relacionamentos</p> <ul style="list-style-type: none"> • <i>Companionship isEnjoyedWith some SocialMeals;</i> • <i>GoOut_WithFriends isFunWith some SocialMeals;</i> • <i>Socialization stimulates some SocialMeals;</i> • <i>SocialEvents areFrequentlyOccuringWith some SocialMeals;</i> • <i>Hospitality stimulates some SharingFood;</i> • <i>SocialCustoms stimulatesToAccept VarietyFoods.</i>
Consultas	<ul style="list-style-type: none"> • <i>? stimulates some SocialMeals</i> • <i>? stimulates some SharingFood</i>

- Resultados Obtidos

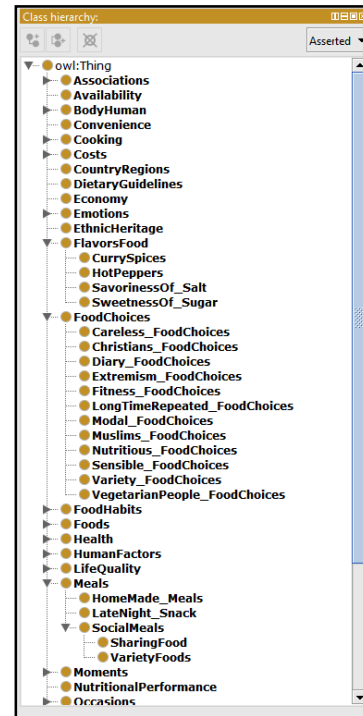


Figura 3.5 – Hierarquia de Classes

Figura 3.6 – Hierarquia de Classes expandida

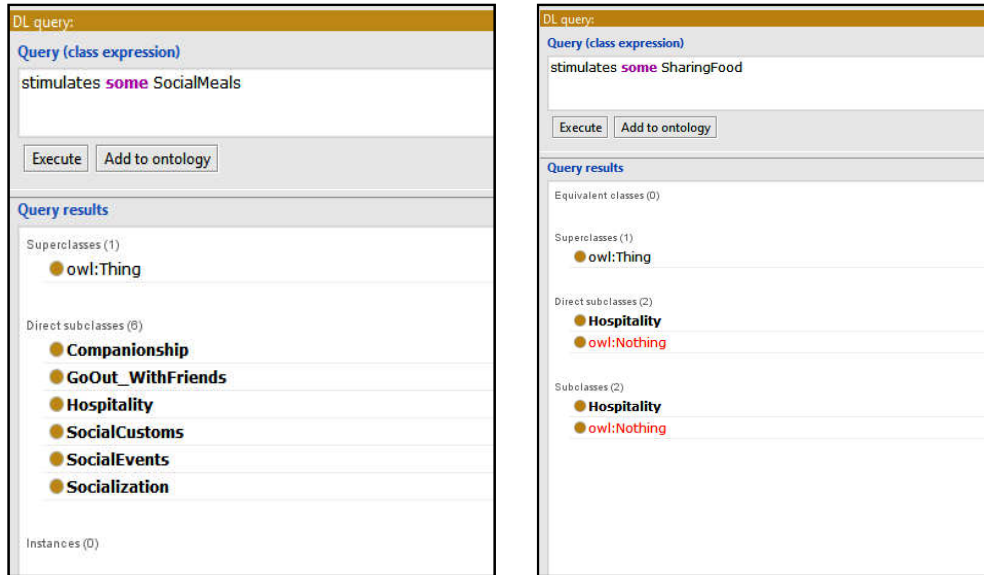


Figura 3.7 – Consulta: “stimulates some SocialMeals”

Figura 3.8 – Consulta: “stimulates some SharingFood”

3.2.4.4 - Consulta com foco na hierarquia de propriedades

Não importa E1, se “p” for uma propriedade que contem subpropriedades vinculadas, todas as superclasses e suas respectivas subclasses e/ou subclasses específicas que tiverem um relacionamento com E2 via propriedade e suas respectivas subpropriedades serão respostas para a consulta. Este caso está representado pela figura 3.12 acima

Agora, se “p” for uma subpropriedade, a resposta para a consulta será apenas E1 que tiver um relacionamento direto com “p”. Se E1 for superclasse, então suas subclasses aparecerão na resposta, se E1 for subclasse, somente ela aparecerá, como é mostrado nas figuras 3.10 e 3.11 acima.

Tabela 13 – Representação em OWL – exemplo 10

Texto Adaptado	<i>Diary Food Choices may little influence of people health. Food Choices repeated long time may much influence people health.</i>
Representação OWL	Propriedades <ul style="list-style-type: none"> • <i>mayInfluence: mayInfluenceLittle, mayInfluenceMuch;</i>
	Relacionamentos <ul style="list-style-type: none"> • <i>Diary_FoodChoices mayInfluenceLittle some People_Health;</i>

	<ul style="list-style-type: none"> • <i>LongTimeRepeated_FoodChoices mayInfluenceMuch some People_Health.</i>
Consultas	<ul style="list-style-type: none"> • <i>? mayInfluenceLittle some PeopleHealth</i> • <i>? mayInfluenceMuch some PeopleHealth</i> • <i>? mayInfluence some PeopleHealth</i>

- Resultados Obtidos



Figura 3.19 – Hierarquia de Propriedades

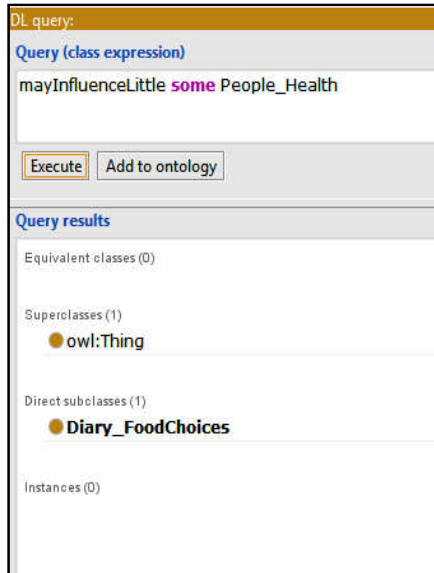


Figura 3.10 - Consulta: “mayInfluenceLittle some Peoplehealth”

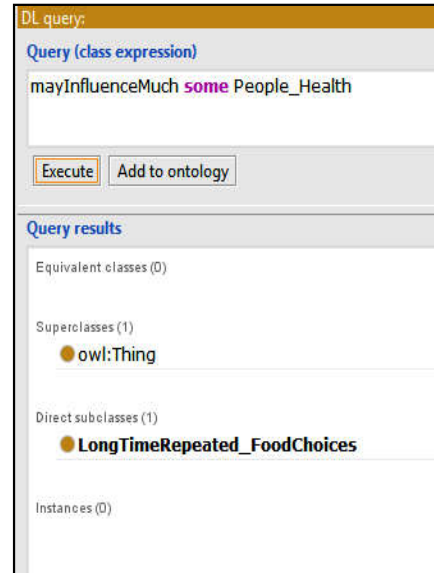


Figura 3.11 - Consulta: “mayInfluenceMuch some Peoplehealth”

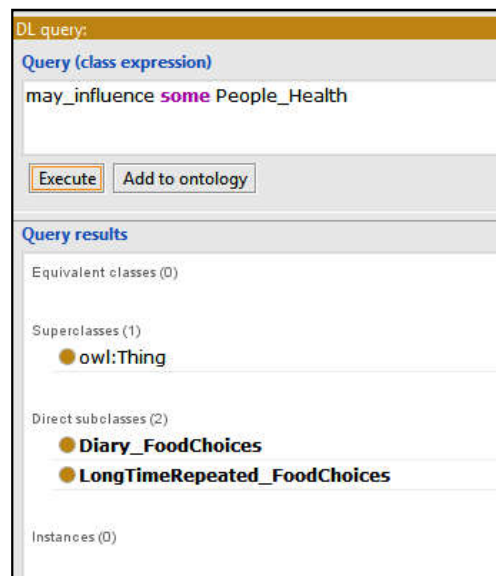


Figura 3.12 - Consulta: “mayInfluence some Peoplehealth”

3.2.4.5 – Consulta com foco na hierarquia de Classes e Propriedades

Neste caso, particiona-se:

- E1 – p, sendo “E1” = superclasse ou subclasse e “p” = superpropriedade. As respostas para consultas via superpropriedade são iguais às impostas na subsecção 3.2.4.4 deste trabalho, mais especificamente na figura 3.12 ilustrada acima.

- E1 – p, sendo “E1” = superclasse e “p” = subpropriedade. As respostas seriam apenas a superclasse e suas respectivas subclasses em E1. As respostas para a consulta ilustrada na figura 3.13 são exemplos para este caso.
- E1 – p, sendo “E1” = subclasse e “p” = subpropriedade. As respostas se condicionam apenas as subclasses de E1, como é mostrada nas figuras 3.10 e 3.11 acima.

Tabela 5 – Representação em OWL – Hierarquia de classes e propriedades

Representação OWL	Hierarquia de Classes <ul style="list-style-type: none"> • <i>FoodChoices: Careless_FoodChoices, Christians_FoodChoices, Diary_FoodChoices, Extremism_FoodChoices, Fitness_FoodChoices, LongTimeRepeated_FoodChoices, Modal_FoodChoices, Muslims_FoodChoices, Nutritious_FoodChoices, Sensible_FoodChoices, Variety_FoodChoices, VegetarianPeople_FoodChoices</i>
	Hierarquia de Propriedade <ul style="list-style-type: none"> • <i>hasInfluenceOn: hasStrongest_influenceOn</i>
	Relacionamentos <ul style="list-style-type: none"> • <i>FoodChoices hasStrongest_influenceOn some LifeQuality</i>
Consultas	<ul style="list-style-type: none"> • <i>? hasStrongest_influenceOn some LifeQuality</i>

- Resultados Obtidos

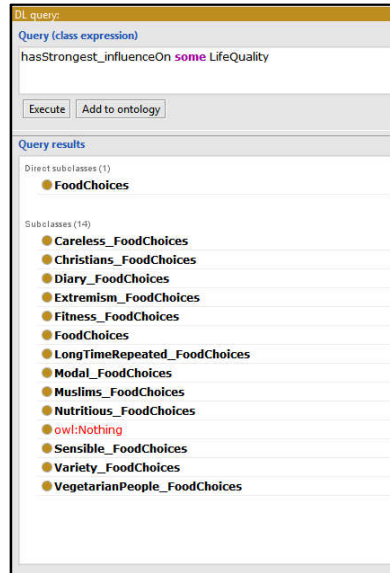


Figura 3.13 - Consulta: “hasStrongest_influenceOn some LifeQuality”

4. Considerações Finais

Nesta seção são apresentadas: as conclusões, as dificuldades encontradas e uma sugestão para trabalhos futuros.

4.1 Conclusões

O texto utilizado para extração da ontologia foi escolhido de forma a dificultar a identificação das relações relevantes e sua representação em OWL. Se o assunto abordado fosse, por exemplo, doenças nutricionais, a identificação dos conceitos e relações de interesse seria muito mais direta: sintomas estão associados a doenças, que são causadas pela deficiência de um dado nutriente, que pode ser encontrado em um dado alimento. No texto escolhido o tema abordado é muito mais abstrato, resultando em uma geração muito mais heterogênea de classes e propriedades, como se pode perceber a partir dos exemplos escolhidos para ilustrar este trabalho.

O mapeamento das sentenças em linguagem natural para OWL resulta em alterações significativas apoiadas em dois passos: (a) do texto original para uma representação intermediária, ainda em linguagem natural, para evidenciar as relações relevantes, e (b) da

geração das construções em OWL, a partir desta representação. O segundo passo é direcionado pelas consultas que poderão ser realizadas a partir da representação em OWL.

É importante observar que a utilização de herança, tanto para a definição de classes de conceitos quanto para a definição de propriedades (relações), é a heurística fundamental para potencializar as consultas realizadas.

4.2 Dificuldades Encontradas

As dificuldades encontradas foram as seguintes:

- Aprendizado dos conceitos da linguagem OWL;
- Aprendizado da utilização da plataforma Protégé para a construção e apresentação de um modelo de ontologia na linguagem OWL;
- Extração de dados em um texto em linguagem estrangeira;
- Análise de dados em um texto com nível de abstração alta;
- Identificação das classes e propriedades heterogêneas presentes no texto;
- Caracterização do mapeamento para OWL com foco em consultas.

4.3 Trabalhos Futuros

A ontologia construída é baseada em relações binárias (em OWL: uma propriedade ligando duas classes). Mas existem construções em linguagem natural que necessitam ser representadas através de relações ternárias, como por exemplo: “boy – gave – gift – to – girlfriend”. Existem indicações de tempo, de maneira, de intensidade que são naturalmente representadas em linguagem natural. OWL não suporta o mapeamento direto destas situações. Uma pesquisa de como abordar a representação destas questões em uma ontologia é bastante pertinente como uma proposta de trabalho futuro.

REFERÊNCIAS

BRICKLEY, Dan; GUHA, R. V. **RDF Schema 1.1**. W3C Editors. W3C Recommendation, 25 February 2014, disponível em:
<http://www.w3.org/TR/rdf-schema/>. Acessado em 24 de Junho de 2014

CESAR, Dr. Danny. **Tratamentos: Doenças nutricionais**. Médico Nutrólogo. Clínica Paris, disponível em: <http://www.clinicaparis.com.br/tratamentos/doencas-nutricionais/>. Acessado em 30 de Abril de 2016

GRUBER, T. **What is an Ontology?** Disponível em:
<http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>. Acesso em 22 de Junho de 2014.

HORRIDGE, M. **A Practical Guide To Building OWL Ontologies Using Protege 4 and CO-ODE Tools**. Edition 1.3, 2011.

LUSTOSA, Pollyane de Almeida. **OWL e Protégé: estudo e aplicação de conceitos para exemplificação da definição da camada de esquema da Web Semântica em um determinado domínio**. Monografia. Curso de Bacharel em Sistemas de Informação. Centro Universitário Luterano de Palmas – CEULP/ULBRA, Palmas, 2003.

MANOLA, Frank; MILLER, Eric; McBRIDE, Brian. **RDF Primer**. W3C Editors. W3C Recommendation, 10 February 2004, disponível em:
<http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>. Acessado em 23 de Junho de 2014.

McGUINNESS, Deborah L.; HARMELEN, Frank V. **OWL Web Ontology Language Overview**. W3C Editors. W3C Recommendation, 10 February 2004, disponível em:
<http://www.w3.org/TR/owl-features/>. Acessado em 22 de Junho de 2014.

STANFORD. **Protege Desktop User Documentation**. Disponível em:
<http://protegewiki.stanford.edu/wiki/Protege4UserDocs>. Acessado em 20 de Abril de 2016

ROSSI, Ana Paula Leão. **Nutrição: Definição**, 29 de Março de 2012, disponível em:
<http://www.portaleducacao.com.br/nutricao/artigos/11570/nutricao-definicao>. Acessado em 10 de Abril de 2016.

WHITNEY, Ellie; ROLFES, Sharon Rady. **Understanding Nutrition**. Edition 12, 2011.

W3 Org. **OWL Web Ontology Language Guide**. W3C Recommendation, 10 February 2004, disponível em: <http://www.w3.org/TR/owl-guide/>. Acessado em 24 de Abril de 2016.