



Universidade Federal do Amazonas
Instituto de Ciências Exatas
Programa de Pós-Graduação em Informática

Construção de Ontologias de Domínio a Partir de Mapas Conceituais

Gretchen Torres de Macedo

Manaus – Amazonas
Março de 2007

Gretchen Torres de Macedo

Construção de Ontologias de Domínio a Partir de Mapas Conceituais

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Instituto de Ciências Exatas da Universidade Federal do Amazonas, como requisito parcial para obtenção do Título de Mestre em Informática. Área de concentração: Inteligência Artificial.

Orientadora: Profa. Virgínia V. B. Biris Brilhante, Ph.D.

Gretchen Torres de Macedo

Construção de Ontologias de Domínio a Partir de Mapas Conceituais

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Instituto de Ciências Exatas da Universidade Federal do Amazonas, como requisito parcial para obtenção do Título de Mestre em Informática. Área de concentração: Inteligência Artificial.

Banca Examinadora

Prof. Virgínia V. B. Biris Brillhante, Ph.D. – Orientadora
Departamento de Ciência da Computação – UFAM/PPGI

Prof. Alberto Nogueira de Castro Jr., Ph.D.
Departamento de Ciência da Computação – UFAM/PPGI

Prof. Dr. Crediné Menezes
Departamento de Ciência da Computação – UFES

Manaus – Amazonas

Maio de 2007

Ao meu esposo Claudemir Ivan, com amor.

Agradecimentos

A conclusão deste trabalho foi algo que, muito longe de depender somente de mim mesma, somente foi possível diante da colaboração de muitas pessoas. Aproveito este espaço para agradecer a todos que possibilitaram a sua realização.

Serei eternamente grata à Professora Virginia Brilhante, que foi muito mais que uma orientadora e superou de longe todas as minhas possíveis expectativas: me pegou pela mão e me conduziu em direção à realização deste trabalho, me fazendo perceber que a graça está em percorrer o caminho.

Tive mais uma vez a sorte de poder contar com Andréa Pereira Mendonça, amiga e professora desde a graduação, que continua me mostrando que nossas maiores limitações nos são impostas por nós mesmos.

Agradeço aos professores do curso, em especial Edleno Silva de Moura, Alberto Nogueira de Castro Jr e Ilias Biris, que transformaram meus conhecimentos, praticamente nulos em ciência da computação, em algo de que eu possa me orgulhar.

Foi muito bom poder contar com meu amigo Bruno Gadelha, sempre presente, que me deu dicas preciosíssimas durante todo o curso.

Agradeço a Sabatta Macedo, pela amizade e grande contribuição nas tarefas do projeto. Agradeço também às colegas da secretaria, em especial à Elienai, sempre eficiente e prestativa.

Agradeço ao CEFET-AM pela oportunidade dada de realizar o experimento com mapas conceituais e aos alunos Cícero, Marcelle, Miriam e Reginaldo, pela colaboração e

dedicação na realização das atividades solicitadas.

Sou muito grata à colaboração da Fundação Paulo Feitoza, que possibilitou a conciliação entre mestrado e trabalho e também a meus amigos de trabalho, que colaboraram para que eu conseguisse realizar estas duas atividades.

Com amor, agradeço ao meu esposo Ivan, de paciência infinita, que soube ser compreensivo diante do adiamento dos planos de crescimento da nossa família.

“The essence of things is not only determined by the things themselves, but also by the contribution of whoever perceives and understands them.”

Emmanuel Kant

Resumo

Ontologias têm sido construídas e utilizadas em diversas aplicações como um modelo de representação de conhecimento compartilhável entre agentes de software e usuários. Mapas conceituais, por sua vez, são um modelo de representação do conhecimento que, em relação às ontologias, é informal, menos complexo e, portanto, de fácil elaboração. Entretanto, a liberdade permitida na definição de conceitos e relações nos mapas dificulta a transcrição direta desses modelos em representações formais que possam ser utilizadas em aplicações baseadas em conhecimento.

Este trabalho apresenta um processo de transcrição de mapas conceituais em ontologias especificadas em OWL (Web Ontology Language), tornando possível o aproveitamento da facilidade de elaboração oferecida por mapas conceituais no processo de construção de ontologias de domínio. O processo de transcrição consiste de duas etapas principais: tradução e mesclagem. A etapa de tradução consiste na obtenção de ontologias intermediárias a partir de um conjunto de mapas conceituais tendo sido realizada mediante o desenvolvimento de uma ferramenta de software. A etapa de mesclagem, responsável por unificar as ontologias intermediárias obtidas na primeira etapa, foi realizada através da utilização de uma ferramenta de mesclagem existente. Foram ainda realizadas experiências de produção de mapas conceituais, os quais foram submetidos às ferramentas mencionadas, de forma a avaliar o processo apresentado.

Palavras-chave: Mapas conceituais; Ontologias *OWL*; Transcrição; Mesclagem; Heurísticas.

Abstract

Ontologies have been built and used in a variety of applications as a form of knowledge representation that is meant for software systems, or agents, as well as for human users. In relation to ontologies, conceptual maps are a more informal, simple and, thus, accessible form of knowledge representation. However, the freedom enjoyed in defining concepts and their links makes it difficult to directly draw formal representations from conceptual maps.

This work presents a transcription process that is able to transform conceptual maps into ontologies specified in OWL (Web Ontology Language). In this way, the ease of construction of conceptual maps can be taken advantage of to alleviate the knowledge acquisition bottleneck that is inherent in ontology engineering. The translation process consists of two main stages: translation and merging. In the translation stage a group of conceptual maps about the same knowledge domain is transformed into a set of preliminary ontologies by means of a translator module software. In the merging stage, ontology merging techniques are applied to the set of preliminary ontologies so as to yield a single unified ontology. This phase has been achieved by means of an available merging tool. Experiments for building conceptual maps have also been done and submitted to the two phases of the translation process, in order to evaluate it.

Keywords: Conceptual maps; OWL Ontologies ; Transcription; Merging; Heuristics.

Sumário

1	Introdução	2
1.1	Questões Envolvendo a Aquisição de Conhecimento	3
1.2	De Mapas Conceituais a Ontologias de Domínio: um Processo de Transcrição	5
1.2.1	Objetivos	6
1.2.2	Metodologia	7
1.3	Organização da Dissertação	8
2	Fundamentação Teórica e Revisão Bibliográfica	10
2.1	Mapas Conceituais	10
2.1.1	Mapas Conceituais Bem Construídos	13
2.1.2	CMapTools: um Ambiente para Construção de Mapas Conceituais .	13
2.2	Ontologias	14
2.2.1	Metodologias de Construção de Ontologias	16
2.2.2	OWL: Uma Linguagem para Representação de Ontologias	17
2.2.3	Integração de Ontologias	19
2.2.3.1	Métodos para Identificação de Similaridade	21
2.3	Trabalhos Relacionados	25
2.3.1	Trabalho Desenvolvido	26
3	De Mapas Conceituais a Ontologias	28
3.1	Processo de Transcrição	28

3.2	Mesclagem de Ontologias Intermediárias	30
3.2.1	Chimaera	30
3.2.2	OntoMerge	31
3.2.3	PROMPT	32
4	Heurísticas de Tradução	35
4.1	Linguagem Adotada na Representação das Heurísticas	36
4.2	Instâncias de Conceitos	37
4.3	Conceitos	39
4.4	Relações de Classificação	40
4.5	Relações de Composição	43
4.6	Relações Bidirecionais	49
4.7	Demais Relações	50
5	Ferramenta de Tradução	51
5.1	Componente <i>Parser</i>	52
5.2	Componente Mapeador	55
5.2.1	Composição de Heurísticas	56
5.3	Componente Gerador de OWL	58
6	Estudo de Caso	62
6.1	Preparação e Realização de Atividades de Construção de Mapas Conceituais	62
6.2	Preparação dos Mapas para Submissão ao Processo de Transcrição	64
6.3	Tradução	69
6.4	Mesclagem	79
6.4.1	Ontologia Resultante	82
6.4.2	Ontologia Alvo	83
6.4.2.1	Mesclagem Adicional de Classes e Propriedades	84
6.4.2.2	Ajuste de Relações Hierárquicas e de Composição	89

6.5	Avaliação	90
6.5.1	Consistência	90
6.5.2	Concisão	91
6.5.3	Completeness	91
6.6	Requisitos para Ferramenta de Mesclagem	91
7	Discussão e Conclusões	93
7.1	Heurísticas	94
7.2	Tradução e Tradutor	95
7.3	Mesclagem	96
7.4	Trabalhos Futuros	96
	Referências Bibliográficas	98

Lista de Figuras

2.1	Mapa conceitual sobre plantas (Traduzido de: servidor CMapTools IHMC internal)	12
3.1	Processo de construção de ontologia de domínio a partir de mapas conceituais	29
3.2	Plug-in PROMPT durante processo de mesclagem de ontologias	32
3.3	Algoritmo PROMPT	33
4.1	Relação que denota os conceitos Public, Private e Package como instâncias de Qualifiers	38
4.2	Relação hierárquica entre conceitos identificada pela expressão <i>can be</i>	41
4.3	Composição entre conceitos identificada pela expressão <i>are composed of</i>	44
5.1	Interface de usuário da ferramenta de tradução	52
5.2	Arquitetura do módulo de tradução	52
5.3	Mapa conceitual em formato Proposições como Texto	53
5.4	Fatos em linguagem Prolog gerados pelo componente <i>Parser</i>	55
5.5	Especificação em Prolog da heurística de identificação de hierarquia	56
5.6	Resolução de possíveis relações de composição e decisão entre subclasses e indivíduos	58
5.7	Trecho de especificação em OWL da classe Flowers	59
5.8	Trecho de especificação em OWL da classe Plants	60
5.9	Ontologia obtida do mapa conceitual sobre Plantas aberta no Protégé	61

6.1	Mapa original do aluno 1	64
6.2	Mapa original do aluno 2	65
6.3	Mapa original do aluno 3	66
6.4	Mapa original do aluno 4	67
6.5	Mapa do aluno 1 em inglês	68
6.6	Mapa do aluno 2 em inglês	69
6.7	Mapa do aluno 3 em inglês	70
6.8	Mapa do aluno 4 em inglês	71
6.9	Ontologia intermediária obtida do mapa do aluno 1: hierarquia (a), propriedades (b), composições (c) e instâncias (d)	73
6.10	Ontologia intermediária obtida do mapa do aluno 2: hierarquia (a), propriedades (b) e composição (c)	75
6.11	Ontologia intermediária obtida do mapa do aluno 3: hierarquia (a), propriedades (b) e composição (c)	77
6.12	Ontologia intermediária obtida do mapa do aluno 4: hierarquia (a), propriedades (b) e composição (c)	78
6.13	Ordem de mesclagem das ontologias intermediárias	79
6.14	Hierarquia de classes da ontologia resultante	83
6.15	Propriedades de objeto da ontologia resultante	84
6.16	Restrições (a) e instâncias (b) da ontologia resultante	85
6.17	Hierarquia de classes da ontologia alvo	86
6.18	Propriedades de objeto da ontologia alvo	87
6.19	Restrições (a) e instâncias (b) da ontologia alvo	88

Lista de Tabelas

2.1	Variações do termo atividade enzimática (adaptado de [13])	22
2.2	Classes similares segundo o método estrutural	23
2.3	Comparação de sentenças geradas a partir de diferentes ontologias	24
6.1	Possíveis hierarquias identificadas na ontologia do aluno 1	72
6.2	Possíveis composições identificadas na ontologia do aluno 1	74
6.3	Diferenciação entre subclasses e instâncias da ontologia do aluno 1	74
6.4	Possíveis hierarquias identificadas na ontologia do aluno 2	74
6.5	Possíveis composições identificadas na ontologia do aluno 2	74
6.6	Possíveis hierarquias identificadas na ontologia do aluno 3	76
6.7	Possíveis composições identificadas na ontologia do aluno 3	76
6.8	Possíveis hierarquias identificadas na ontologia do aluno 4	77
6.9	Possíveis composições identificadas na ontologia do aluno 4	78
6.10	Classes e propriedades mescladas entre as ontologias 1 e 2	80
6.11	Classes e propriedades mescladas entre as ontologias 1-2 e 3	81
6.12	Classes e propriedades mescladas entre as ontologias 1-2-3 e 4	82

Capítulo 1

Introdução

Ontologias são definidas como “uma especificação explícita de uma conceituação” [22]. A especificação do conhecimento na forma de ontologias torna possível que o conhecimento relativo a um determinado domínio seja utilizado por sistemas baseados em conhecimento na realização de diferentes tarefas, tais como permitir a troca de informações entre agentes de software de diferentes aplicações [31] ou inferir informações não explicitamente especificadas [44].

Para viabilizar este tipo de tarefa, ontologias de domínio devem capturar corretamente a essência dos conceitos que compõem o domínio representado, bem como as relações existentes entre estes conceitos e as teorias e princípios elementares que os regem. Isto é necessário para viabilizar a reutilização desta ontologia em aplicações semelhantes dentro do mesmo domínio. Deve também ser estabelecido um compromisso ontológico [21], ou seja, um acordo sobre o significado pretendido dos termos adotados na ontologia, para que estes possam ser formalizados de maneira que sejam somente interpretados de acordo com o significado pré-estabelecido. Estas circunstâncias fazem da construção de ontologias um processo demorado, complexo e, por vezes, repetitivo.

Uma outra ferramenta de representação de conhecimento, os mapas conceituais[36], são utilizados para expressar graficamente o conhecimento de uma pessoa ou grupo sobre

os conceitos que formam um domínio e as relações existentes entre os mesmos. Esta ferramenta é mais informal, menos complexa e, portanto, de fácil elaboração. Entretanto, a liberdade permitida na definição de conceitos e relações faz com que a mesma informação possa ser representada de muitas formas diferentes quando descrita por diferentes pessoas. Além disso, não há formas especiais de representação de relações específicas, como hierarquias entre conceitos. Estas características dificultam a transcrição direta de mapas conceituais em representações formais.

Este trabalho propõe o aproveitamento da flexibilidade oferecida por mapas conceituais na especificação de conceitos e relações de um domínio específico como uma contribuição para a solução do problema do gargalo de aquisição de conhecimento (*knowledge acquisition bottleneck*) [26], que é característico da construção de ontologias.

Este capítulo inicia-se com uma explanação sobre os problemas inerentes ao processo de construção de ontologias de consenso (Seção 1.1). Em seguida, na Seção 1.2, é apresentado o processo desenvolvido de transcrição de mapas conceituais em ontologias de consenso. Finalmente, na seção 1.3 é apresentada a organização da dissertação.

1.1 Questões Envolvendo a Aquisição de Conhecimento

A construção de ontologias é o processo de traduzir o conhecimento da mente de especialistas para uma representação rigorosa que especifique uma interpretação inequívoca dos conceitos e de suas relações pertinentes a um domínio. Este processo tradicionalmente envolve o trabalho em grupo com especialistas de domínio e engenheiros do conhecimento, várias entrevistas, construção e reconstrução da ontologia, além da resolução de conflitos resultantes das diferentes estruturas cognitivas dos especialistas do domínio. A estrutura cognitiva de um indivíduo é a forma como os conceitos e relações estão organizados em sua mente [36].

Elaborar uma ontologia é um processo onde são identificados e descritos formalmente

as classes e indivíduos do domínio abordado, como também as relações que ocorrem entre indivíduos destas classes. É necessário definir os limites do domínio de conhecimento com objetividade, descrever o conhecimento essencial ao domínio e definir um vocabulário que minimize interpretações ambíguas [22]. Esta definição deve, ainda, ser obtida mediante o consenso de todos os envolvidos na especificação formal do domínio. As dificuldades inerentes a este processo são características do chamado “*knowledge acquisition bottleneck*” [26].

Mapas conceituais, por sua vez, mostram-se uma poderosa ferramenta de eliciação do conhecimento devido a sua facilidade de construção. Isto acontece porque, durante sua elaboração, somente é necessário concentrar-se na identificação dos conceitos de um domínio e nas relações existentes entre os mesmos. Sua construção não depende da definição completa de conceitos, da sua identificação de maneira inequívoca, nem tampouco do consenso de um grupo de especialistas de domínio.

No contexto deste trabalho, uma das características mais relevantes de mapas conceituais é a facilidade com que podem ser elaborados. Isto faz com que seu processo de construção seja menos complexo, podendo ser construídos por pessoas conhecedoras do domínio sem intervenção de engenheiros de conhecimento. Além disso, por representarem o conhecimento de um único indivíduo (ou grupo de indivíduos) trabalhando cooperativamente sobre um domínio específico, não dependem da resolução de conflitos ideológicos durante sua concepção. Ambientes de autoria de mapas conceituais, como o CMapTools [7] ou o CMTTool [42], facilitam ainda mais a atividade de elaboração, oferecendo ao usuário ferramentas que facilitam a disposição de conceitos e relações.

Embora a liberdade oferecida na definição de conceitos e relações nos mapas conceituais permita que a mesma informação possa ser representada de diferentes maneiras em diferentes mapas, estes constituem um valioso corpo de conhecimento, mesmo que informalmente representado. Tal corpo de conhecimento pode ser aproveitado como fonte inicial para a construção de uma ontologia de domínio formalmente especificada.

Na seção seguinte, é apresentada uma visão geral do processo elaborado neste trabalho, que utiliza mapas conceituais na construção de ontologias de domínio específico. Tal abordagem visa aliviar o gargalo de aquisição de conhecimento inerente às metodologias tradicionais de construção de ontologias de domínio.

1.2 De Mapas Conceituais a Ontologias de Domínio: um Processo de Transcrição

Diante da possibilidade de aproveitamento do conhecimento disponível na forma de mapas conceituais na elaboração de ontologias de domínio, foi desenvolvido um processo onde, a partir de um conjunto de mapas conceituais de um mesmo domínio, pode-se extrair e unificar o conhecimento do domínio ali representado, em conceitos e relações de diversas formas, para obtenção de uma ontologia de grupo.

O processo é formado por duas etapas. Na primeira, cada um dos mapas conceituais é traduzido para uma linguagem de representação de ontologia, constituindo-se assim em uma ontologia intermediária do domínio. Estas ontologias são, na segunda etapa do processo, mescladas através da utilização de técnicas de mesclagem de ontologias, resultando assim em uma ontologia de grupo.

Estabelecida esta abordagem para obtenção de ontologias de domínio a partir de mapas conceituais, surgem as seguintes questões de pesquisa:

- Como mapear conceitos e relações de mapas conceituais em elementos de ontologias?
- É possível usar abordagens de mesclagem de ontologias em ontologias criadas a partir de mapas conceituais?
- O que faz com que o resultado gerado seja uma ontologia e não apenas uma representação de um mapa conceitual de grupo?

Das questões de pesquisa apresentadas acima, foram levantadas as hipóteses a seguir.

- **Hipótese 1:** O uso de heurísticas é uma boa abordagem na identificação de proposições especiais que são representadas em diferentes construções na forma de ontologia.
- **Hipótese 2:** Uma vez que um conjunto de mapas conceituais é transformado em ontologias intermediárias, as técnicas de mesclagem que podem ser utilizadas são limitadas apenas pelos elementos disponíveis nas ontologias em si, e não pela origem das informações ali especificadas.
- **Hipótese 3:** A transcrição de proposições de um mapa conceitual em elementos específicos de ontologias não explicitamente representados na mesma torna possível sua utilização em sistemas inteligentes baseados em conhecimento onde a ontologia construída é utilizada na realização de inferências, o que não pode ser feito com mapas conceituais diretamente.

1.2.1 Objetivos

Esta dissertação tem como objetivo principal investigar a possibilidade de utilização de um conjunto de mapas conceituais relativos ao mesmo domínio para obtenção de uma ontologia de consenso. Este objetivo pode ser alcançado através dos seguintes objetivos específicos:

- identificar heurísticas capazes de mapear conceitos e relações de mapas conceituais em elementos de ontologias;
- implementar uma ferramenta de tradução de cada mapa conceitual em uma ontologia intermediária, utilizando as heurísticas identificadas;

- identificar técnicas de mesclagem de ontologias que possam ser aplicadas às ontologias intermediárias, originárias dos mapas conceituais, para a construção de uma ontologia de consenso;
- pesquisar e selecionar uma ferramenta de mesclagem que implemente as técnicas selecionadas para construção da ontologia de consenso;
- avaliar o processo de tradução como um todo através da análise da ontologia obtida, segundo os critérios de consistência, coesão e completude, sendo este último avaliado em relação ao conteúdo inicialmente definido nos mapas conceituais.

1.2.2 Metodologia

Com o objetivo de obter mapas que sirvam como material para realização das etapas de tradução e de mesclagem do processo de transcrição, foram realizadas atividades de busca e seleção de mapas nos servidores da ferramenta CMapTools [7]. Foram pesquisados mapas em inglês, uma vez que foi previsto o uso de bancos de dados léxicos, disponíveis apenas nesse idioma, no processo de transcrição. Além disso, foram selecionados mapas de tamanho médio, que contivessem entre quinze e vinte conceitos, visando minimizar a complexidade dos experimentos de tradução manual de mapas. Foi ainda um critério de seleção que as relações contidas nos mapas formassem proposições acerca do domínio representado.

Os mapas selecionados foram então analisados, buscando identificar características diferenciadas nas proposições representadas em linguagem natural, que pudessem ser representadas em construções de ontologias. Para isso, os mapas foram inicialmente transcritos manualmente. Foram então elaboradas heurísticas que possibilitassem a identificação destas características e o consequente mapeamento das mesmas em elementos de ontologias.

Foi desenvolvida uma ferramenta de tradução que, utilizando as heurísticas estabelecidas, é capaz de traduzir cada mapa em uma ontologia intermediária.

A segunda etapa do processo de transcrição, que consiste na mesclagem das ontologias intermediárias, foi realizada através da utilização de uma ferramenta de mesclagem disponível. Para verificar se a ferramenta poderia ser utilizada na mesclagem de ontologias obtidas a partir de mapas conceituais, foi selecionada uma ferramenta que implementasse técnicas de mesclagem aplicáveis a este tipo de ontologia.

Como estudo de caso, foi realizado um experimento para produção de mapas conceituais em um contexto de aprendizagem. Tais mapas foram então submetidos à ferramenta de tradução, gerando um conjunto de ontologias intermediárias. Estas ontologias foram então mescladas, gerando-se assim uma ontologia do domínio representado.

A ontologia resultante foi avaliada segundo critérios pré-estabelecidos, sendo por fim construída, a partir desta, uma outra ontologia, representando a que poderia ter sido potencialmente obtida dos mapas conceituais. A comparação entre estas duas ontologias permitiu a avaliação da ontologia resultante segundo os critérios de consistência, completude e coesão.

1.3 Organização da Dissertação

Esta dissertação está organizada em mais cinco capítulos. O Capítulo 2 apresenta a fundamentação teórica das duas formas de representação do conhecimento abordadas nesta dissertação, mapas conceituais e ontologias. No Capítulo 3, o processo de tradução de mapas conceituais em ontologias de consenso é apresentado, juntamente com as heurísticas que possibilitam tal mapeamento. O Capítulo 5 apresenta considerações importantes relativas à construção do módulo e tradução, juntamente com o estudo de características desejáveis em uma ferramenta de mesclagem de ontologias obtidas a partir de mapas conceituais. No Capítulo 6, são apresentados os resultados dos experimentos de obtenção de ontologias de consenso a partir de mapas conceituais, nos quais mapas foram submetidos ao processo apresentado e posteriormente avaliados. Finalmente, no Capítulo 7, são

apresentadas uma discussão e conclusões sobre os resultados do processo apresentado.

Capítulo 2

Fundamentação Teórica e Revisão

Bibliográfica

Este trabalho descreve um processo de transformação de um modelo de comunicação de conhecimento informal, mapa conceitual, em uma representação ontológica formal. Ambas as formas de representação de conhecimento são descritas nas seções 2.1 e 2.2. Em seguida são apresentados alguns trabalhos existentes, que utilizam abordagens relacionadas à nossa para construir ontologias.

2.1 Mapas Conceituais

Mapas conceituais são representações gráficas do conhecimento de uma pessoa sobre os conceitos de um domínio e das relações existentes entre os mesmos. Podem também representar o conhecimento de um grupo quando elaborados cooperativamente. Conceitos podem ser vistos como regularidades percebidas em eventos ou objetos [36]. Mapas conceituais são formados por um conjunto de conceitos, especificados em caixas, e ligações entre os mesmos, especificados como linhas. Ambos são identificados por rótulos, podendo também serem utilizados símbolos para descrever conceitos.

A representação de conhecimento na forma de mapas conceituais tem origem na teoria

de Ausubel [3] sobre aprendizagem significativa. Segundo Ausubel, a estrutura cognitiva é, no contexto de um determinado assunto, o conteúdo e organização das idéias na mente de um indivíduo sobre aquela área particular de conhecimento. Para Ausubel, a estrutura cognitiva de cada indivíduo é extremamente organizada e hierarquizada, no sentido em que as várias idéias se encadeiam de acordo com as relações que se estabelecem entre elas.

Novak [36] desenvolveu pesquisas de ensino-aprendizagem com mapas conceituais em 1984, utilizando-os como ferramenta de representação, construção e comunicação de conhecimento. Para Novak, o propósito de mapas conceituais é representar relações relevantes entre conceitos na forma de proposições, formadas por conjuntos *conceito - relação - conceito*. Assim, proposições são afirmações sobre os conceitos envolvidos.

Baseando-se nos princípios definidos por Ausubel, Novak propõe o uso de mapas conceituais como ferramenta para externalizar o conhecimento de um indivíduo, no sentido em que torna possível a expressão de forma gráfica da estrutura cognitiva do mesmo através da representação de conceitos e das diversas relações existentes entre os mesmos. Novak [36] afirma ainda que mapas conceituais devem ser hierárquicos, de forma a refletir mais fielmente a hierarquia da organização de idéias na mente de um indivíduo. Assim, conceitos mais gerais e mais inclusivos devem estar localizados no topo do mapa, com conceitos progressivamente mais específicos e menos inclusivos localizados na parte inferior. Na Figura 2.1, é apresentado um mapa conceitual sobre plantas, onde esta característica pode ser percebida.

Novak ressalta ainda que a característica gráfica de mapas conceituais os torna uma ferramenta de negociação de significado, possibilitando diálogo, discussão e compartilhamento dos significados expostos por diferentes pessoas, levando-os a identificar ambigüidades e inconsistências e a chegar a um acordo sobre o domínio discutido.

O princípio teórico fundamental sobre mapas conceituais é o de estrutura hierárquica [36]. Segundo este princípio, cada novo conceito é mais facilmente agregado à estrutura cognitiva de um indivíduo se for alocado abaixo de conceitos mais abrangentes existentes.

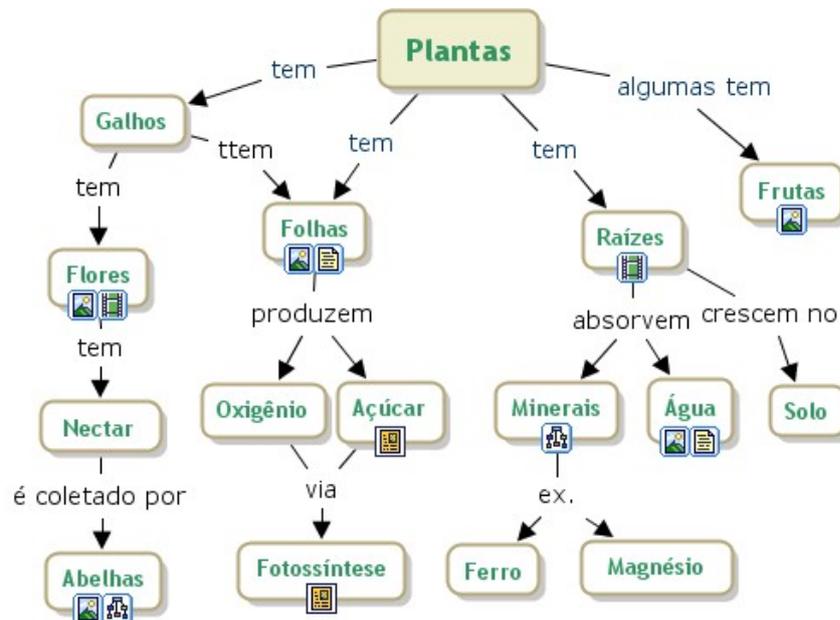


Figura 2.1: Mapa conceitual sobre plantas (Traduzido de: servidor CMapTools IHMC internal)

Agregados a este princípio estão os de Diferenciação Progressiva e o de Reconciliação Integrativa. O primeiro baseia-se na idéia de que o aprendizado é o resultado da mudança no significado dos conceitos conhecidos, e que novos conceitos ganham significados maiores e mais complexos conforme novas proposições vão sendo adquiridas, sendo assim progressivamente diferenciados. Já a ação de agregar novas proposições ao conhecimento existente é chamada de reconciliação integrativa.

Finalmente, é importante salientar o caráter idiossincrático dos mapas conceituais, que refletem a estrutura cognitiva de um único indivíduo acerca de um domínio. Esta característica faz com que mapas conceituais sobre o mesmo tema construídos por indivíduos diferentes sejam necessariamente diferentes. Cada mapa conceitual é apenas uma das representações possíveis de uma estrutura conceitual, e esta última pode, portanto, ser representada por várias hierarquias válidas.

2.1.1 Mapas Conceituais Bem Construídos

O uso de direcionamentos para a construção de mapas conceituais facilita sua utilização por ferramentas inteligentes que possam, a partir do seu conteúdo, prover fontes de informação para aplicações baseadas em conhecimento. Em [9], embora seja defendido o uso de mapas conceituais sem restrições quanto ao uso de frases de ligação ou de conceitos em taxonomias pré-definidas, assume-se o uso de algumas diretrizes na construção dos mapas conceituais, a fim de torná-los processáveis por ferramentas de apoio ao seu processo de elaboração. Tais diretrizes são definidas por Novak [36] como características de mapas conceituais bem construídos, quais sejam:

- o conceito principal define o tema do mapa;
- cada tripla conceito - relação - conceito forma uma proposição que faz sentido se lida separadamente;
- possuem estrutura hierárquica, com conceitos mais gerais definidos na parte superior do mapa e conceitos mais específicos na parte inferior;
- possuem contexto bem definido, abordando conceitos de um único domínio;
- rótulos dados a conceitos e frases de ligação são os mais curtos possíveis.

Dentre tais características, são de fundamental relevância para este trabalho aqueles mapas que formam proposições independentes a cada tripla conceito - frase de ligação - conceito e cujos rótulos de conceitos e frases de ligação são os mais curtos possíveis.

2.1.2 CMapTools: um Ambiente para Construção de Mapas Conceituais

Ferramentas de apoio à construção de mapas conceituais, como CMTool [42], Inspiration [29], MindMan [35] e Decision Explorer [46], evitam que sua forma gráfica de representação

torne-se um empecilho à recuperação das informações neles representada. Dentre estas ferramentas, destaca-se o CMapTools [9], desenvolvido pelo IHMC - Institute for Human Machine Cognition da University of West Florida.

Uma característica importante deste ambiente gratuito, ao tratarmos mapas conceituais como ferramenta de aquisição do conhecimento, é a possibilidade de exportação de conteúdo em proposições como texto ou em formato XML. Esta funcionalidade possibilita o processamento do mapa por outras aplicações, tais como ferramentas de apoio ao processo de construção de mapas de maneira colaborativa [9].

O CMapTools permite também a construção e disponibilização de mapas conceituais em servidores Web, oferecendo assim acesso a vastas bases de conhecimento especificado na forma de mapas conceituais.

2.2 Ontologias

Ontologia é um termo emprestado da filosofia e refere-se à ciência que descreve os tipos de entidades existentes no mundo e de que maneira estão relacionadas. Em Ciência da Computação, é um termo utilizado para representar especificações formais dos conceitos pertinentes a um certo domínio de conhecimento e das relações existentes entre os mesmos.

Gruber [22] define ontologias como uma especificação explícita de uma conceituação. Ontologias têm sido utilizadas em sistemas nas mais diversas áreas de conhecimento, com propósitos que vão desde a formalização de conceitos básicos até a realização de inferências em sistemas inteligentes. Como exemplo de ontologia podemos destacar [44], que descreve a construção de uma ontologia sobre estruturas físicas básicas de organismos vivos. Tal ontologia tem por objetivo servir como base para o desenvolvimento de ontologias mais específicas, que podem ser utilizadas em diferentes aplicações sobre ciências biomédicas, uma vez que diferentes organismos compartilham as mesmas estruturas básicas no nível genético. [5] trata do desenvolvimento de uma ontologia de base que especifica termos

jurídicos levando em consideração o senso comum para identificar relações causais entre eventos. Em [45] é apresentada uma ontologia para anotação de imagens com dois níveis de representação de conhecimento que descreve mais explicitamente o conhecimento contido nas mesmas. Estes níveis baseiam-se, respectivamente, nas características fundamentais de imagens como um todo e no conteúdo de cada imagem.

Dada a grande variedade de sistemas onde ontologias são utilizadas, Kavouras [30] apresenta uma classificação de ontologias conforme o nível de especificidade. Nesta classificação, destacam-se as ontologias *de domínio*, que definem conceitos associados a um domínio específico. Tais ontologias, em oposição às de *uso geral*, formalizam conceitos tomando-se por base um contexto específico, consistindo em um entendimento comum e compartilhado do mesmo e possibilitando a troca de informações em nível semântico por sistemas baseados em conhecimento cujos domínios se sobrepõem. Quanto ao grau de formalidade, [21] classifica ontologias em *leves*, quando contém apenas uma taxonomia de conceitos e definições dos mesmos em linguagem natural; e ontologias *pesadas*, que compreendem também indivíduos, propriedades e axiomas em linguagem formal.

Construir uma ontologia consiste em identificar a conceituação relativa ao domínio abordado e formalizar explicitamente esses conceitos. Esta formalização permite que o conhecimento relativo ao domínio possa ser compartilhado e que possa ser interpretado por pessoas e *software*, por este último, em particular, através do processamento, possivelmente utilizando padrões de inferência, do conhecimento especificado. É um processo onde são identificados e descritos formalmente as classes e, opcionalmente, indivíduos do domínio abordado, como também as relações que podem ocorrer entre indivíduos das classes. É necessário definir os limites do domínio de conhecimento com objetividade, descrever o conhecimento essencial ao domínio e definir um vocabulário que minimize interpretações ambíguas [22]. Deve ainda existir um acordo sobre o significado pretendido de cada um dos termos utilizados na ontologia, estabelecido mediante o consenso de todos os envolvidos na especificação formal do domínio.

2.2.1 Metodologias de Construção de Ontologias

Existem diversos métodos de desenvolvimento de ontologias [47, 1, 14], que podem ser classificados em três tipos [10]: métodos *bottom-up*, que começam com descrições do domínio e obtém uma classificação de conceitos; métodos *top-down*, que partem de uma visão abstrata do domínio; e métodos *middle-out*: dos conceitos mais importantes aos mais específicos ou gerais.

Em [47], primeiro método para construção de ontologias desenvolvido, foi proposta uma estrutura de metodologia para construção de ontologias através de um processo manual cujas etapas são: identificar propósito, escopo e questões de competência, identificar de conceitos-chave e relacionamentos, codificar ontologia sendo definidos os termos adotados, integrar a ontologia com outras existentes, avaliar e documentar a ontologia.

O método 101 de desenvolvimento de ontologias foi criado por autores de ambientes de edição de ontologias, como Protégé e Chimaera [1]. Este método segue um processo de design iterativo, onde os passos gerais consistem em: determinar o domínio e escopo da ontologia, considerar o reuso de ontologias existentes, enumerar termos importantes na ontologia, definir as classes e hierarquia de classes, definir propriedades, ajustar valores de propriedades (domínio, range, cardinalidade) e criar instâncias.

A metodologia de desenvolvimento de ontologias Methontology [14], além de determinar um fluxo de atividades que devem ser seguidas na elaboração da ontologia, estabelece também as tarefas que envolvem o gerenciamento (planejamento, controle e garantia da qualidade) e o suporte à construção da ontologia (aquisição de conhecimento, avaliação, integração e documentação). As atividades de desenvolvimento da ontologia são: a especificação, que determina o propósito, a justificativa e os usuários finais da ontologia; a conceituação, onde são elaboradas especificações completas dos termos, hierarquias entre conceitos, fórmulas e regras, ainda não formalizados, de forma a verificar sua correta especificação com usuários finais; a formalização, onde a especificação conceitual é formalizada em uma linguagem de representação de conhecimento e; a implementação, onde o modelo

formal é escrito em uma linguagem de implementação.

De maneira geral, os métodos iniciam por uma fase onde é definido o propósito e escopo da ontologia. Esta fase é seguida pela de conceituação propriamente dita, onde são identificados os conceitos-chave e propriedades, criadas especificações em linguagem natural, seguidas de especificação formal dos conceitos e identificação de instâncias, e é definida a hierarquia existente entre os mesmos. A fase final é a de codificação, onde a ontologia é escrita em uma linguagem de representação de ontologias para que possa ser efetivamente utilizada em sistemas baseados em conhecimento.

2.2.2 OWL: Uma Linguagem para Representação de Ontologias

Existem diversas linguagens para especificação de ontologias, como lógicas de ordem superior ou Lógicas de Descrição. A OWL (*Web Ontology Language*) [4] é uma linguagem para definição e instanciação de ontologias recomendada pelo W3C (World Wide Web Consortium) tendo sido concebida para viabilizar a utilização de ontologias pela Web Semântica. OWL permite a derivação de conseqüências lógicas, ou seja, conhecimento não explicitamente presente na ontologia pode ser deduzido das relações semânticas explícitas e implícitas existentes. Alguns tipos de inferência, como classificação e verificação de consistência, podem ser realizadas com o uso de raciocinadores (*reasoners*), como RACER [23] e FaCT [27].

As principais bases de OWL são as Lógicas de Descrição (Description Logics) e a linguagem RDF (Resource Description Framework) [32]. OWL herda de RDF uma sintaxe semelhante à de XML (eXtensible Markup Language). RDF foi criada inicialmente para descrever significados de conteúdos de documentos ou recursos Web identificados por URIs (*Universal Resource Identifiers*). Através da explicitação de propriedades (relações ou características) dos recursos e de valores para tais propriedades. Uma variação de RDF, RDF *Schema*, fornece primitivas para padronizar a criação dos esquemas de descrição,

visando facilitar, a aplicações Web, a localização e interpretação do conhecimento descrito. Das Lógicas de Descrição, OWL herdou a estrutura semântica que permite a utilização de mecanismos de inferência. Tal estrutura semântica, fundamentada na Teoria de Modelos e na Lógica de Primeira Ordem, é formada por conceitos, propriedades e indivíduos. Desta forma, uma ontologia em OWL pode ser entendida como uma base de conhecimento em DL na qual possível descrever recursos disponíveis na Web.

Uma ontologia OWL é constituída de classes, propriedades e indivíduos, onde classes representam conceitos e podem ser organizadas em hierarquias de classes e indivíduos representam instâncias de classes. Propriedades representam relações, ou pares ordenados, entre indivíduos de classes. É possível determinar para cada propriedade as classes de domínio e de contradomínio, que definem as classes dos indivíduos que participam, respectivamente, como origem e destino das relações. Propriedades podem também ser organizadas em hierarquias e ainda serem definidas como funcionais (cada indivíduo da classe de domínio relaciona-se somente com um indivíduo correspondente no contradomínio), transitivas, simétricas e reflexivas. Para cada propriedade pode ser também definida uma propriedade inversa, que ocorre entre as mesmas classes, mas na direção oposta (do contradomínio para o domínio da propriedade original).

Além disso, é possível criar restrições sobre classes. Restrições são compostas por uma classe, uma propriedade e um quantificador, descrevendo o conjunto de indivíduos que possuem relações através desta propriedade com os indivíduos desta classe da maneira descrita pelo quantificador. Dois quantificadores são especialmente importantes para este trabalho. O quantificador existencial (\exists) é usado para definir que cada indivíduo pertinente à classe deve possuir pelo menos uma relação através da propriedade utilizada na restrição com indivíduos da classe utilizada na definição da restrição. O quantificador universal (\forall) é usado para definir que todas as relações dos indivíduos da classe em questão estabelecidas através da propriedade devem ser com indivíduos da classe utilizada na restrição. Por exemplo, a restrição existencial \exists *temAtividade Caminhada* sobre a classe *ParqueFlorestal*

indica que todo parque florestal deve possuir pelo menos uma atividade do tipo caminhada. A mesma restrição, com quantificador universal, significaria que todas as atividades de parque florestal devem ser do tipo *caminhada*. Existem ainda outros tipos de restrições que podem ser estabelecidas, como as de cardinalidade, que determinam quantos indivíduos de uma classe relacionam-se com os da outra classe através da propriedade especificada, ou de valor, que define valores explícitos em vez de uma classe no contradomínio da relação.

Restrições são, em outras palavras, classes anônimas de indivíduos da classe que satisfazem a condição especificada, sendo que a classe onde as restrições são definidas é formada pelos indivíduos que atendem a todas as restrições ou à interseção destas classes anônimas.

2.2.3 Integração de Ontologias

A integração de ontologias é um problema que foi previsto por engenheiros do conhecimento: conforme fossem sendo criadas especificações ontológicas para diferentes aplicações, seriam produzidas diversas representações formais para sistemas que operam sobre o mesmo domínio mas utilizam diferentes vocabulários por se tratarem de diferentes contextos.

Deste cenário, resulta a necessidade de integrar ontologias, permitindo que sistemas cujos domínios se sobreponham ou estejam relacionados possam trocar informações, ou que se obtenha uma especificação mais completa e abrangente de um dado domínio. De acordo com [24], a tendência natural é a de que ontologias se proliferem e divirjam entre si, em vez de convergirem para um grupo de ontologias padrão de alto nível. Ainda que esta última hipótese se confirmasse, ainda assim seria necessário fazer o alinhamento do conhecimento contido nas diversas versões destas ontologias, conforme as mesmas fossem evoluindo e novas versões fossem criadas.

Existem dois principais paradigmas quando se trata de integração de ontologias, de acordo com o resultado que se deseja obter: alinhamento e mesclagem. O alinhamento é

utilizado normalmente quando duas ontologias representam domínios complementares ou fazem parte de sistemas baseados em conhecimento que eventualmente precisam trocar informações [40]. Durante o alinhamento, é feita uma especificação semântica, através de axiomas, de como conceitos em ontologias diferentes estão relacionados. Exemplos de tipos de relações são equivalência e similaridade entre classes e relações de classe-subclasse. Neste tipo de integração, as ontologias originais permanecem inalteradas, e apenas conceitos com correspondências em ambas as ontologias fazem parte do mapeamento. Na mesclagem, por outro lado, duas (ou mais) ontologias são unificadas em uma ontologia resultante [37]. As ontologias originais podem se tornar indisponíveis, sendo substituídas pela ontologia mesclada, ou podem continuar sendo utilizadas, por questões de compatibilidade, sendo neste último caso estabelecidos mapeamentos entre os conceitos das ontologias originais e seus correspondentes na ontologia mesclada. No caso em que uma única ontologia é obtida, é esperado que algumas modificações na mesma sejam necessárias [40].

Tanto no alinhamento quanto na mesclagem, a primeira questão a ser resolvida por ferramentas de integração é a identificação de conceitos similares nas ontologias, o que envolve a resolução de problemas terminológicos como, por exemplo, o uso de diferentes termos para representar o mesmo conceito. Na mesclagem, onde as ontologias originais são substituídas pela ontologia mesclada, novos problemas devem ser tratados, entre os quais a mesclagem de instâncias, hierarquias e axiomas de restrição, bem como propriedades reconhecidas como equivalentes. As hierarquias mescladas podem ainda possuir diferentes níveis de granularidade e a ontologia resultante pode conter ambigüidades ou inconsistências, que devem ser identificadas e resolvidas.

As correspondências entre conceitos em ambas as representações e o tipo e grau de relacionamento entre os mesmos serve para guiar o processo de mesclagem ou para ser transformada em mapeamento durante o alinhamento. Tal atividade pode ser dividida em outras duas: encontrar similaridades e, dentre estas, decidir quais serão especificadas

através de mapeamentos, uma vez que o processo de encontrar classes semelhantes pode conter pares de classes que não são realmente equivalentes.

2.2.3.1 Métodos para Identificação de Similaridade

Diversas técnicas têm sido utilizadas para encontrar correspondências entre conceitos entre ontologias, que vão desde a simples comparação de caracteres até a verificação semântica dos conceitos. Sua aplicação depende dos objetos, do contexto em que se encontram e algumas vezes, de sua semântica externa.

A seguir, são apresentados os principais métodos existentes para solucionar o problema de estabelecer relações de similaridade entre conceitos. Os mesmos se dividem em quatro tipos: terminológicos, estruturais, extensivos e semânticos.

Métodos Terminológicos

Os métodos terminológicos baseiam-se na comparação de caracteres [13], sendo aplicados ao nome ou aos comentários de um objeto. Existem dois tipos principais de métodos terminológicos: o baseado em caracteres e o baseado em linguagem. O primeiro deles utiliza técnicas de comparação direta de caracteres para estabelecer relações de similaridade. Esta técnica identificaria, por exemplo, Aeronave e aeronave como conceitos semelhantes, mas não Aeronave e Avião. Os métodos baseados em caracteres realizam uma etapa antes da comparação de caracteres, que é a etapa de normalização. Esta etapa serve para preparar as cadeias de caracteres que serão comparadas, a fim de eliminar diferenças irrelevantes que possam atrapalhar a comparação, tais como a transformação dos caracteres para letras minúsculas, a supressão de sinais ou dígitos, a normalização de caracteres brancos, supressão de dígitos, pontuação e palavras de ligação. As formas de comparação de caracteres mais utilizadas são as seguintes:

- Igualdade de cadeias de caracteres, que simplesmente compara duas cadeias de caracteres, sem verificar o quanto as cadeias são diferentes.

Tipo de Variação	Exemplo
Morfológica	Atividades enzimáticas
Sintática	Atividade de enzima, atividade bactericida e enzimática
Morfossintática	Atividade de enzimas
Morfológica	Atividades enzimáticas
Semântica	Fermentação
Multi-linguagem	Enzyme activity

Tabela 2.1: Variações do termo atividade enzimática (adaptado de [13])

- Teste de *substrings*, onde se estabelece um grau de similaridade entre as cadeias de caracteres, quando uma cadeia está contida na outra.
- Distância de edição, técnica que mede o grau de similaridade entre dois objetos pelo número de edições (inserções, remoções e trocas de caracteres) necessárias em uma cadeia de caracteres para transformá-la na outra. A cada operação é associado um custo, que implica na redução do grau de similaridade resultante.
- Comparação de caminhos, que além de comparar conceitos diretamente, esta técnica faz uma composição do nome de um conceito com o nome da sua superclasse.

O segundo método terminológico é o baseado em linguagem, que usa técnicas de processamento de linguagem natural para encontrar associações entre instâncias de conceitos e classes [13]. Esta técnica baseia-se na variação de termos, considerando que os mesmos podem ser expressos de diferentes maneiras. Na Tabela 2.1 são apresentados exemplos de variações do termo atividade enzimática.

Os métodos baseados em linguagem se dividem em intrínsecos e extrínsecos. Métodos intrínsecos fazem uso de análise sintática e morfológica para normalizar termos. Este método é normalmente utilizado em recuperação de informação e funciona removendo variações de sufixos e prefixos de palavras, reduzindo-as ao radical durante a normalização. Métodos extrínsecos fazem uso de dicionários externos, como WordNet [34], para encontrar

Pessoa	Humano
Nome	Nome
Endereço	Gênero
Estado civil	Peso
Peso	Altura
Altura	Idade

Tabela 2.2: Classes similares segundo o método estrutural

termos sinônimos. Uma variação desse método é o de métodos multi-linguagem, que fazem uso de dicionários multi-linguagem para comparar termos em diferentes idiomas.

Métodos Estruturais

Os métodos estruturais comparam a estrutura dos conceitos, em vez de simplesmente seus nomes [13]. Dividem-se em métodos internos e externos. Os métodos internos comparam a estrutura interna de conceitos, como propriedades e restrições. Uma abordagem mais completa destes métodos utiliza também outras características associadas aos conceitos, como cardinalidade, unicidade e restrições de integridade semântica. Na Tabela 2.2, o grau de similaridade entre os conceitos Pessoa e Humano seria dado pela quantidade dos atributos comuns às duas classes (Nome, Peso e Altura).

Métodos estruturais internos são usados normalmente em combinação com métodos terminológicos, extensivos ou semânticos, para reduzir o número de sugestões.

Os métodos externos verificam a similaridade de conceitos pelas relações taxonômicas que os mesmos possuem com os outros conceitos em sua ontologia, ou seja, pela posição que ocupam em suas respectivas hierarquias de conceitos. Os critérios adotados por esta abordagem consideram conceitos pais, irmãos, filhos e conceitos ascendentes ou descendentes indiretos para estabelecer graus de similaridade. Estes métodos se tornam inadequados quando as duas ontologias possuem diferentes subclassificações para o mesmo conceito. Por exemplo, em uma ontologia tem-se o conceito *Humano*, com subclasses *Homem* e *Mulher*. Na segunda, tem-se novamente o conceito *Humano*, porém com as subclasses *Jovem* e *Adulto*. Utilizando a comparação de conceitos filhos, não será estabelecido um

Primeira Ontologia	Segunda Ontologia
éUma(jornal, publicaçãoPeriodica)	éUma(jornal, publicação)

Tabela 2.3: Comparação de sentenças geradas a partir de diferentes ontologias

grau de similaridade entre os conceitos *Humano*, visto que seus filhos não são similares.

Métodos Extensivos

Os métodos extensivos comparam as instâncias dos conceitos, verificando se dois conceitos compartilham a mesma instância [13]. Por exemplo, considerando duas ontologias, onde na primeira possuímos a classe *Avião*, que tem como instâncias os objetos *Boeing 747* e *Fokker 100*. Na segunda ontologia, temos a classe *Aeronave*, que possui as instâncias *Airbus A320* e *Boeing 747*. Comparando as instâncias de ambos os conceitos, pode-se obter uma medida de similaridade entre *Avião* e *Aeronave*, que por métodos terminológicos não seria possível estabelecer. Além de comparar diretamente as instâncias, podem ser usadas técnicas terminológicas baseadas em linguagem na comparação entre os valores de instâncias, para calcular o grau de similaridade de acordo com a distância entre os termos (sufixo, prefixos ou substrings, por exemplo).

Métodos Semânticos

Métodos semânticos transformam as ontologias em sentenças em lógica de primeira ordem, mapeando pares de conceitos e relações em asserções lógicas [13], como no exemplo *Jornal - é uma - Publicação*, que é traduzido como *éUma(jornal, publicação)*.

É então feita uma avaliação de sentenças geradas, para verificar sua validade. As relações entre conceitos são comparadas entre as diferentes ontologias, e uma vez que um conceito possui a mesma interpretação em ambas, o mesmo é alinhado. Entretanto, a simples comparação de relações não é suficiente para permitir o estabelecimento de similaridade entre conceitos, como visto no exemplo da Tabela 2.3.

Se não for indicado anteriormente que os termos *publicação* e *publicaçãoPeriódica* são equivalentes nas duas ontologias, a aplicação do método semântico por si só não seria

capaz de identificar sua equivalência. Por este motivo, métodos semânticos necessitam, antes de serem executados, de uma fase de preparação, na qual conceitos são indicados como equivalentes, através da interferência do usuário ou da aplicação de métodos terminológicos.

2.3 Trabalhos Relacionados

Em [20] é utilizada uma abordagem semelhante de aproveitamento de mapas conceituais para contornar o gargalo de aquisição de conhecimento. O mesmo trata de um sistema de inferências baseadas em caso (*Case-Based Reasoning*), paradigma no qual experiências anteriores são utilizadas para resolver problemas correntes. Para que as experiências anteriores possam ser utilizadas, estas devem primeiramente ser formalizadas, neste trabalho, em Lógicas de Descrição (*Description Logics*). Os usuários do sistema, entretanto, evitaram seu uso devido à complexidade enfrentada na conceituação e formalização de casos, caracterizando uma situação onde ocorre o gargalo de aquisição de conhecimento. Mapas conceituais foram então introduzidos como um passo intermediário que antecede a especificação de casos em lógicas de descrição. [18] também utilizou mapas conceituais para facilitar o desenvolvimento de ontologias feito por usuários dispersos geograficamente.

Em seu trabalho, [9] argumenta que mapas conceituais foram concebidos para serem construídos e entendidos por humanos, e que qualquer tentativa de formalização de seu conteúdo ou de limitar as frases de ligação ou conceitos utilizados em sua construção é uma maneira de transformar mapas conceituais em algo que eles não são. São apresentadas diversas ferramentas que fazem uso de técnicas de inteligência artificial para auxiliar na elaboração de mapas conceituais. Uma destas ferramentas, Soup [8], permite que um participante visualize proposições de mapas conceituais elaboradas e publicadas por outros participantes de um grupo, com base nas proposições publicadas de seu próprio mapa conceitual, de modo que este só tenha acesso às proposições de outros mapas que

tenham semelhança com as proposições que o mesmo publicou.

A medição de aprendizagem significativa é outra área na qual ontologias e mapas conceituais podem ser combinados na resolução de problemas. Em [42], é apresentada uma ferramenta de medição de aprendizagem significativa, através da qual uma ontologia é construída pelo educador contendo conceitos e frases de ligação relativos a um domínio e que serão potencialmente descritos por seus alunos. A ontologia contém também as distâncias semânticas entre as diferentes frases de ligação que podem ser utilizadas na construção do mapa do aluno. Um algoritmo genético é responsável por gerar um conjunto de todos os possíveis mapas conceituais que descrevem o domínio especificado. Sendo assim, tendo como entrada o mapa de um aluno, outro algoritmo verifica se o mesmo equivale a alguma das soluções geradas.

COE [25] é uma ferramenta integrada em Java que permite realizar busca de conceitos, composição, edição e reutilização colaborativa de ontologias em um ambiente gráfico, baseado no CMapTools. Utiliza mapas conceituais por considerar que os mesmos oferecem uma forma natural de mostrar e examinar a estrutura de uma ontologia em um ambiente colaborativo. Entretanto, para evitar a poluição visual durante a exibição de uma ontologia obtida através de um mapa conceitual, exibe a mesma de maneira simplificada, evidenciando relações relevantes e escondendo outros tipos de informações, como notas e comentários. Para possibilitar a construção de ontologias usando apenas os recursos oferecidos na definição de mapas, e, ainda assim, agilizar o processo, faz uso tanto de convenções baseadas nos nomes das relações, como de ontologias-*template*, que direcionam o usuário a utilizar, na construção de sua ontologia, as relações especiais pré-definidas, como relações de subclasse e restrições.

2.3.1 Trabalho Desenvolvido

Este trabalho apresenta uma abordagem para obtenção de ontologias de domínio a partir de mapas conceituais, semelhante à adotada por [18] para integrar o conhecimento de

usuários dispersos graficamente, apresentando também uma solução em software capaz de transformar uma representação na outra.

Embora tal abordagem se oponha ao propósito de mapas conceituais segundo [9], um de seus trabalhos [8] faz uso de heurísticas para identificar proximidades semânticas entre proposições contidas no mapa em elaboração e as publicadas por outros usuários.

A abordagem de duas camadas utilizada neste trabalho, que segmenta o problema de tradução em duas etapas em detrimento de uma abordagem de transformação direta de mapas conceituais em uma ontologia de domínio, foi inspirada pelo processo apresentado em [19], que parte partindo de grafos semânticos e utiliza ontologias na geração de apresentações multimídia.

Não é feita nenhuma restrição ou recomendação durante a definição de conceitos e frases de ligações nos mapas conceituais, em contraste à abordagem apresentada em [25]. Neste trabalho, relações semânticas especiais entre conceitos são identificados com o uso do banco de dados léxico WordNet.

Finalmente, é importante ressaltar algumas aplicações desta abordagem na área de aprendizagem. Primeiramente, a formalização do conhecimento de um grupo em uma ontologia de domínio permite verificar o grau de conhecimento deste grupo e identificar possíveis inconsistências ao serem unificadas diferentes representações do domínio. Diferentes ontologias poderiam também ser geradas em momentos distintos de tempo, para avaliar o progresso do conhecimento adquirido pelo grupo.

Capítulo 3

De Mapas Conceituais a Ontologias

No Capítulo 1 foi defendido que mapas conceituais podem ser utilizados como um ponto de partida no processo de construção de ontologias de domínio. Este capítulo começa com a apresentação do processo de transcrição de ontologias a partir de mapas conceituais. Em seguida, são apresentadas ferramentas de mesclagem pré-selecionadas e a utilizada para realizar a função de módulo de Mesclagem.

3.1 Processo de Transcrição

Como visto no Capítulo 2, mapas conceituais oferecem grande flexibilidade na descrição de conceitos e proposições e, por este motivo, mapas construídos por pessoas diferentes mesmo que retratando o mesmo domínio, são necessariamente diferentes. Desta forma, e também por representar a estrutura cognitiva de um único indivíduo, um mapa conceitual não pode ser adotado como correto ou definitivo na representação de um domínio de conhecimento. Este é o motivo pelo qual, no processo aqui apresentado, parte-se de um conjunto de mapas conceituais sobre o mesmo domínio para a obtenção de uma ontologia de consenso.

Assim, dado um domínio sobre o qual se deseja obter uma especificação formal, cada um dos especialistas do domínio elabora seu próprio mapa conceitual, representando sua



Figura 3.1: Processo de construção de ontologia de domínio a partir de mapas conceituais

estrutura cognitiva sobre o domínio abordado. Esta atividade tem como resultado um conjunto de mapas conceituais sobre o mesmo domínio, mas com diferentes conceitos e relacionamentos entre os mesmos. Sendo assim, a transformação de mapas conceituais em ontologias corrobora para a unificação do conhecimento representado em cada mapa em uma representação ontológica única.

Primeiramente, cada mapa conceitual é submetido a um módulo de tradução, que o transcreve em uma linguagem de representação de ontologias, sendo então obtido um conjunto de ontologias individuais intermediárias. Em uma segunda etapa do processo, são aplicadas técnicas de mesclagem a estas ontologias, gerando-se assim uma ontologia do domínio. O processo completo é apresentado na Figura 3.1. Os módulos de tradução e mesclagem estão descritos, respectivamente, no Capítulo 5 e na Seção 3.2 deste capítulo.

É importante salientar que nenhuma restrição é imposta sobre a definição de conceitos e relações durante a elaboração dos mapas conceituais, visto que uma de suas características mais importantes como ferramenta de elicitação de conhecimento é justamente a flexibilidade oferecida na definição de seus elementos. Em vez disso, são adotadas as diretrizes de mapas conceituais bem construídos na construção dos mapas conceituais.

3.2 Mesclagem de Ontologias Intermediárias

O processo de alinhar e mesclar ontologias, quando feito manualmente, é tedioso e demorado, principalmente quando existem mais de duas ontologias a serem mescladas. No caso de ontologias construídas de maneira distribuída, é necessário também um extensivo processo de negociação entre seus autores, de forma a atingir um consenso de representação na ontologia resultante. A ontologia resultante precisa ainda ser verificada, de forma que sejam encontrados problemas de inconsistência, incompletude ou duplicidade de elementos. Entretanto, o uso de ferramentas de mesclagem agiliza este processo e facilita a identificação de possíveis inconsistências na ontologia resultante.

A segunda etapa do processo de obtenção de uma ontologia a partir de mapas conceituais consiste em mesclar as ontologias individuais, oriundas de mapas, gerando assim uma ontologia de domínio. Para realizar esta etapa, foram avaliadas diversas ferramentas de mesclagem existentes, de forma a selecionar a que melhor se adequasse a mesclar ontologias obtidas a partir de mapas conceituais. Esta atividade foi necessária uma vez que os métodos de mesclagem utilizados em algumas implementações fazem uso de características específicas que as ontologias individuais obtidas na primeira etapa do processo não necessariamente possuem, como, por exemplo, a especificação de instâncias. Primeiramente, foram pré-selecionadas as ferramentas que aceitassem OWL como linguagem de entrada e saída e, dentre as mesmas, a mais adequada às ontologias resultantes do experimento de transcrição realizado. As ferramentas pré-selecionadas foram Chimaera, OntoMerge e PROMPT, tendo esta última sido escolhida pelos motivos apresentados na Seção 3.2.3, juntamente com as características mais relevantes de cada uma destas ferramentas.

3.2.1 Chimaera

O Chimaera [33] é um ambiente Web interativo para mesclagem e teste de ontologias, desenvolvido pelo Laboratório de Sistemas de Conhecimento da Universidade de Stan-

ford. Foi inicialmente concebido para permitir a mesclagem de bases de conhecimento desenvolvidas por diferentes autores, sendo posteriormente adaptada para dar suporte à criação, edição, navegação e teste de ontologias. Utiliza métodos terminológicos para encontrar termos equivalentes e complementar hierarquias e instâncias de classes. Permite que os usuários interfram, a qualquer momento, no processo de mesclagem, indicando semelhanças entre significados de termos.

O Chimaera possui também uma ferramenta de diagnóstico embutida. Entretanto, tal ferramenta somente avalia ontologias parcialmente quanto à corretude e completude, não verificando inconsistências. Além disso, o acesso via Web, mais adequado ao desenvolvimento de ontologias de maneira distribuída, se torna um empecilho por não estar sempre disponível.

3.2.2 OntoMerge

O OntoMerge [12] apresenta abordagem diferenciada para mesclagem de ontologias, a ferramenta faz a união de todos os elementos das duas ontologias em uma só, utilizando-se *namespaces* para diferenciar estes elementos. A mesclagem é realizada, de fato, através do uso de axiomas de ligação (*bridging axioms*) que estabelecem relações entre termos equivalentes.

O uso de *namespaces* torna confuso o uso da ontologia por seus usuários, caso a mesma tenha sido mesclada com o propósito de permitir a comunicação de sistemas cujos domínios se sobrepõem, e não para a obtenção de uma representação única e mais completa do domínio representado, substituindo as ontologias originais. Além disso, a ontologia mesclada é poluída por termos com diferentes *namespaces*, axiomas de ligação e termos equivalentes duplicados, o que dificulta a usabilidade, compartilhamento e reuso da ontologia por outras aplicações. O autor sugere, entretanto, que a ferramenta aplica-se a situações em que é necessária apenas a tradução de termos entre ontologias, chamada por alguns autores de mediação em tempo de execução [11].

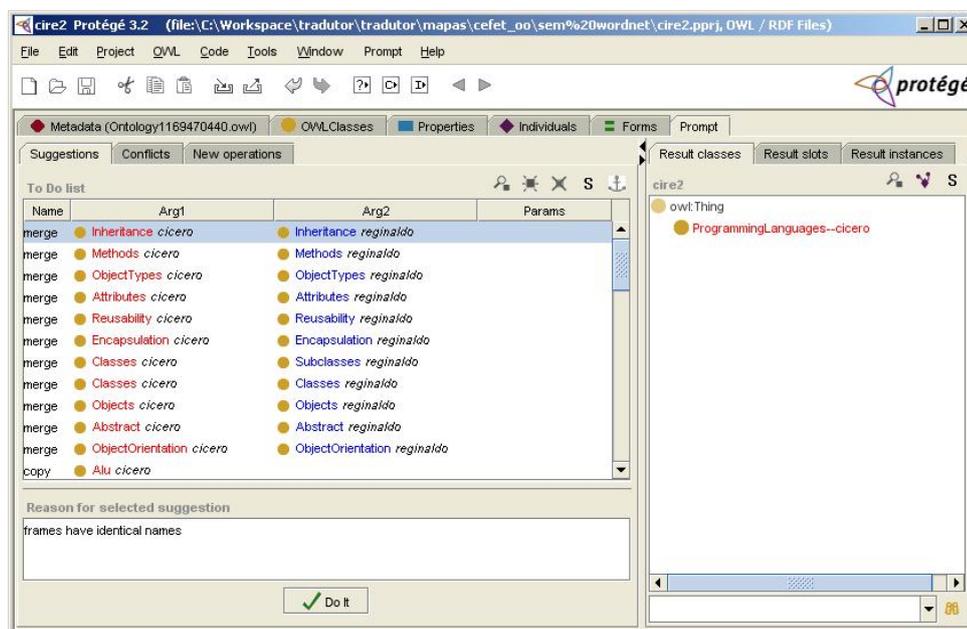


Figura 3.2: Plug-in PROMPT durante processo de mesclagem de ontologias

Esta ferramenta utiliza internamente uma linguagem de representação de ontologias (Web-PDDL), mas permite a importação e exportação em linguagem OWL.

3.2.3 PROMPT

O PROMPT [39], assim como suas extensões (AnchorPROMPT, PROMPTDiff), são baseados no ambiente Protégé de edição de ontologias, aceitando como entrada, por este motivo, ontologias nos formatos permitidos pelo ambiente, inclusive OWL. É uma ferramenta de mesclagem e alinhamento com mecanismos sofisticados de perguntas ao usuário para confirmação de possíveis termos equivalentes.

A ferramenta recebe como entrada um par de ontologias, a partir das quais é elaborado um conjunto de pares de termos equivalentes pertencentes às mesmas, que são identificados com o auxílio de técnicas terminológicas, como as apresentadas na Seção 2.2.3.1. A partir disto, PROMPT constrói um grafo direcionado e rotulado representando a ontologia através da hierarquia de conceitos e relações, onde nós no grafo são conceitos e arcos são relações denotando relacionamentos entre conceitos (sendo os rótulos os nomes das

relações). AnchorPROMPT analisa então os caminhos do sub-grafo e determina quais conceitos freqüentemente aparecem em posições ou caminhos similares. Baseado nessas freqüências, o algoritmo decide se tais conceitos são semanticamente similares. A Figura 3.2 apresenta a lista inicial de sugestões apresentada pela ferramenta após realizadas as etapas descritas em um ontologia intermediária.

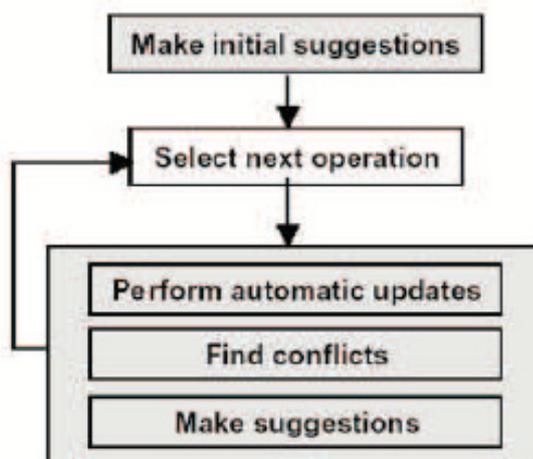


Figura 3.3: Algoritmo PROMPT

O elemento central do PROMPT é o algoritmo que define um conjunto de passos para o processo de mesclagem interativo e iterativo, mostrado na figura 3.3. No primeiro passo do algoritmo são identificados candidatos à mesclagem em potencial, baseando-se em similaridades dos nomes das classes. O resultado é apresentado ao usuário como uma lista de operações de mesclagem em potencial. O segundo passo é iniciado pelo usuário quando o mesmo escolhe uma das operações sugeridas na lista ou especifica diretamente a operação. O sistema executa a ação e automaticamente executa mudanças adicionais derivadas da ação, sendo então reconstruída a lista de sugestões, com base na nova estrutura da ontologia resultante, após determinar conflitos introduzidos pela última ação e encontrar possíveis soluções para estes conflitos.

O processo iterativo característico desta ferramenta torna possível visualizar ações

realizadas durante a mesclagem, além de permitir que sejam feitas intervenções nestas ações. Além disso, o ambiente Protégé suporta ontologias escritas em OWL diretamente, sem necessidade de importação e exportação. Também por ser integrada ao ambiente Protégé, a ferramenta está sempre em contínuo aprimoramento. Por este motivos, a mesma foi escolhida para viabilizar a segunda etapa do processo de transcrição.

Capítulo 4

Heurísticas de Tradução

A primeira etapa do processo apresentado na Seção 3.1 consiste na transcrição de cada um dos mapas conceituais individuais em ontologias em OWL, esta última uma recomendação da W3C. Para que isso pudesse ser feito, foi necessário considerar mapeamentos de conceitos e relações em mapas conceituais em elementos de ontologias em OWL. Como critérios de seleção de mapas dos servidores CMapTools para realizar os experimentos manuais de transcrição, foram escolhidos mapas bem construídos (ou facilmente adaptáveis a bem-construídos), de tamanho médio (cerca de 15 conceitos), em inglês e que abordassem somente um domínio de conhecimento. A restrição a mapas em inglês se deve ao fato de que algumas heurísticas fazem uso do banco de dados léxico WordNet [34], construído neste idioma, para identificar tipos especiais de relação. Após a realização de experiências de transcrição manual de mapas conceituais em ontologias, algumas características relevantes foram percebidas nos mapas conceituais, no que diz respeito à organização dos conceitos e suas relações são mapeadas. Nesta seção, estas características identificadas são apresentadas, como também o modo como as mesmas são mapeadas para OWL, compondo assim ontologias individuais correspondentes cada uma a um mapa conceitual. Tais mapeamentos são descritos como um conjunto de heurísticas [6], uma vez que foram criadas através da observação das características das relações, tendo sido formuladas como

regras em lógica de predicados de primeira ordem, que se baseiam nas características terminológicas das relações entre conceitos nos mapas conceituais. É possível que o mesmo elemento de um mapa possua mais de um mapeamento correspondente em OWL. Isto pode acontecer especificamente nos casos onde são identificadas instâncias de um conceito no mapa conceitual, que podem ser mapeados na ontologia como relações classe-subclasse ou como indivíduos de uma classe. Estas soluções são tratadas na fase de tradução.

As heurísticas se beneficiam do fato de que as relações, nos mapas conceituais, são binárias e direcionadas, características que tornam possível a definição de predicados em lógica de primeira ordem. Para as regras que definem as heurísticas, assume-se um sistema de prova baseado no método de resolução [43]. Todas as variáveis são universalmente quantificadas. As abreviações MC e OWL são utilizadas para distinguir predicados relativos à representação em mapa conceitual ou OWL, respectivamente.

4.1 Linguagem Adotada na Representação das Heurísticas

A linguagem adotada neste trabalho para representação das heurísticas de transcrição de mapas conceituais em ontologias individuais é a lógica de primeira ordem, que se baseia em objetos e relações para descrever relações entre os mesmos. Através da lógica de primeira ordem é possível, ainda, descrever fatos ou regras sobre alguns ou todos os objetos no universo representado. Esta linguagem utiliza-se basicamente de termos, sentenças e quantificadores na definição de fatos e regras.

Um termo é uma expressão lógica que se refere a um objeto. Um termo pode ser simples, quando é uma constante ou variável, ou complexo, quando é uma função composta por termos entre parênteses que são argumentos para a função. O termo complexo $indivOWL(C_{to}, C_{from})$, por exemplo, é uma função que enuncia, segundo a interpretação dada neste trabalho, que C_{to} é um indivíduo de C_{from} na representação OWL, onde C_{to}

e C_{from} são variáveis.

A partir de termos são formadas sentenças, que são utilizadas para enunciar fatos. As mesmas podem ser atômicas, quando são formadas por um só termo, ou complexas, quando são combinadas através de conectivos lógicos. Os conectivos lógicos podem ser:

- negação (\neg) - serve para negar uma sentença.
- conjunção (\wedge) - define uma sentença cuja validade é definida pela validade de todas as sentenças que a compõem.
- disjunção (\vee) - define uma sentença cuja validade é definida pela validade de alguma das sentenças que a compõem.
- implicação (\Rightarrow) - serve para indicar que uma sentença é verdadeira somente se sua premissa o for. Caso contrário, nenhuma afirmação sobre sua validade é feita.
- bicondicional (\Leftrightarrow) - utilizada para indicar que uma sentença é verdadeira somente se a outra o for, e vice-versa.

Na sentença 4.1, por exemplo, são utilizados os conectores de conjunção e implicação para descrever a regra de que C_{to} é um indivíduo da classe C_{from} na representação OWL se existe uma relação entre ambos no mapa conceitual e ainda, se esta relação pertence ao conjunto de frases de ligação que caracterizam relações entre indivíduo e classe.

Finalmente, quantificadores são utilizados em lógica de primeira ordem para descrever sentenças sobre alguns (\exists) ou todos (\forall) os indivíduos do universo descrito.

4.2 Instâncias de Conceitos

Não há uma distinção em mapas conceituais entre a representação de conceitos e de instâncias de conceitos. Entretanto, certas frases de ligação podem indicar uma relação

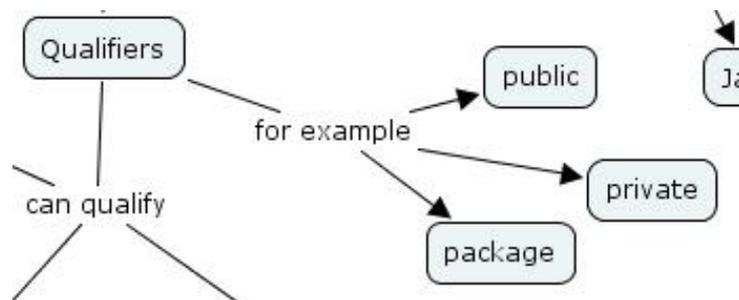


Figura 4.1: Relação que denota os conceitos Public, Private e Package como instâncias de Qualifiers

entre conceito e instância, como, por exemplo, o rótulo *for example* para a relação entre o conceito *Qualifiers* e os conceitos *Public*, *Private* e *Package* na Figura 4.1.

Para identificar instâncias de conceitos nos mapas conceituais, são utilizadas duas heurísticas. A primeira delas possui um conjunto de termos usualmente utilizados para relacionar conceitos às suas instâncias. A segunda, procura verificar se o conceito pode ser identificado com algum tipo de dado (valores numéricos e caracteres entre aspas), o que caracteriza o conceito como uma instância. O conceito em si estará representado como um nó do mapa conceitual relacionado ao conceito que for identificado como instância.

Em OWL, um conceito instanciado, C , torna-se uma classe, e uma das maneiras de representar suas instâncias é através de indivíduos. Como será mostrado na Seção 4.4, as instâncias podem também ser mapeadas como subclasses da classe C . Tal mapeamento, de um conceito instanciado no mapa conceitual como uma classe com indivíduos na ontologia OWL, é formulado como a heurística apresentada a seguir.

C_{to} é um indivíduo de C_{from} na especificação em OWL de um dado mapa conceitual, se R é uma relação no mapa do conceito C_{from} para o conceito C_{to} e R é um membro de um conjunto de frases de ligação que relacionam conceitos com suas instâncias:

$$indivOWL(C_{to}, C_{from}) \leftarrow relationCM(R, C_{from}, C_{to}) \wedge R \in InstPhraseSet \quad (4.1)$$

Onde indivíduos OWL pertencem a classes OWL:

$$\text{classOWL}(C) \leftarrow \text{indivOWL}(I, C) \quad (4.2)$$

Conceitos numéricos ou representados entre aspas também são indivíduos:

C_{to} é um indivíduo de C_{from} na especificação em OWL de um dado mapa conceitual, se R é uma relação no mapa do conceito C_{from} para o conceito C_{to} e C_{to} é um número, então R é uma relação de instâncias:

$$\text{indivOWL}(C_{to}, C_{from}) \leftarrow \text{relationCM}(R, C_{from}, C_{to}) \wedge \text{numericalValue}(C_{to}) \quad (4.3)$$

C_{to} é um indivíduo de C_{from} na especificação em OWL de um dado mapa conceitual, se R é uma relação no mapa do conceito C_{from} para o conceito C_{to} e C_{to} é uma cadeia de caracteres, então R é uma relação de instâncias:

$$\text{indivOWL}(C_{to}, C_{from}) \leftarrow \text{relationCM}(R, C_{from}, C_{to}) \wedge \text{quoted}(C_{to}) \quad (4.4)$$

4.3 Conceitos

Conceitos de mapas conceituais são diretamente mapeados em classes em OWL.

C é uma classe OWL na especificação OWL de um dado mapa conceitual, se C é um conceito no mapa:

$$\text{classOWL}(C) \leftarrow \text{conceptCM}(C) \quad (4.5)$$

Onde,

C_{from} e C_{to} são conceitos em um mapa conceitual se estiverem relacionados no mapa:

$$(\text{conceptCM}(C_{from}) \wedge \text{conceptCM}(C_{to})) \leftarrow \text{relationCM}(R, C_{from}, C_{to}) \quad (4.6)$$

4.4 Relações de Classificação

Em [2], é feita uma distinção entre mapas conceituais associativos e classificatórios. Os primeiros são aqueles que descrevem relações gerais entre conceitos. Os segundos são aqueles onde as relações entre conceitos são principalmente hierárquicas. Relações hierárquicas em mapas conceituais estruturam os mesmos de forma que conceitos em um certo nível do mapa são mais específicos quando comparados a conceitos em níveis superiores.

Uma relação hierárquica em um mapa conceitual é mapeada como uma relação da mesma natureza na ontologia em OWL, ocorrendo entre as classes correspondentes aos conceitos do mapa. As regras 4.24 e 4.10 identificam relações hierárquicas entre os nomes de dois conceitos relacionados em um mapa conceitual através de um conjunto de frases de ligação, como em *is a*, *can be*, *type of*, *e.g.*, etc. e de relações semânticas definidas no banco de dados léxico Wordnet [34].

Frases de ligação que denotam relações de classificação são separadas em dois grupos: um conjunto de frases comumente usadas para denotar relações hierárquicas descendentes, como em *can be* na Figura 4.2, e um conjunto de frases de ligação que denotam relações hierárquicas ascendentes, como em *is a* em *Tree-is a-Plant*.

Hiperônimos e hipônimos são relações léxicas que se aplicam a termos cujos significados estão subordinados uns aos outros: dados dois termos T e T' , T é um hiperônimo de T' se T generaliza T' ; similarmente, T é um hipônimo de T' se T especializa T' . Relações de hiperonímia ou hiponímia no WordNet entre os nomes dos conceitos no mapa conceitual reforça a relação de subclasse entre as classes correspondentes aos conceitos na ontologia.

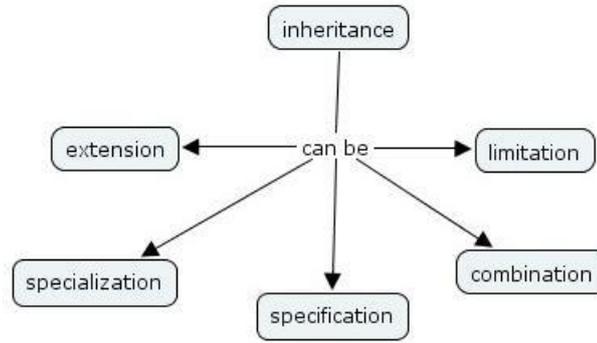


Figura 4.2: Relação hierárquica entre conceitos identificada pela expressão *can be*.

C_{to} é uma subclasse de C_{from} na especificação OWL de um dado mapa conceitual, se R é uma relação do conceito C_{from} para o conceito C_{to} no mapa, e R é membro de um conjunto de frases de ligação que indicam que C_{to} é um subconceito de C_{from} , e C_{from} é um hiperônimo de C_{to} :

$$\begin{aligned} subclassOWL(C_{to}, C_{from}) \leftarrow & \quad (4.7) \\ relationCM(R, C_{from}, C_{to}) \wedge R \in DescHierPhraseSet \wedge hypernym(C_{from}, C_{to}) \end{aligned}$$

Onde,

C e C' são classes OWL se uma relação de subclasse OWL ocorre entre elas:

$$(classOWL(C) \wedge classOWL(C')) \leftarrow subclassOWL(C, C') \quad (4.8)$$

Similarmente,

C_{from} é uma subclasse de C_{to} na especificação OWL de um dado mapa conceitual, se R é uma relação do conceito C_{from} para o conceito C_{to} no mapa, e R é membro de um conjunto de frases de ligação que indicam que C_{from} é um subconceito de C_{to} , e C_{from} é um hipônimo de C_{to} :

$$\begin{aligned} subclassOWL(C_{from}, C_{to}) \leftarrow & \quad (4.9) \\ relationCM(R, C_{from}, C_{to}) \wedge R \in AscHierPhraseSet \wedge hyponym(C_{from}, C_{to}) \end{aligned}$$

O uso de frases de ligação de relações de classificação entre conceitos pode indicar possíveis relações deste tipo, mesmo que não exista uma relação de hiponímia ou hiperonímia correspondente no WordNet entre estes conceitos. O inverso também pode ocorrer quando estes tipos de relação entre os conceitos ocorrerem, mas o rótulo da relação no mapa não pertence ao conjunto de frases de ligação.

Sendo assim, a heurística seguinte é capaz de identificar relações onde esta situação ocorre, tornando possível tratá-las.

C_{to} é uma subclasse de C_{from} em potencial na especificação OWL de um dado mapa conceitual, se R é uma relação que utiliza frase de ligação hierárquica, e R não é uma relação hierárquica de C_{from} para C_{to} no WordNet:

$$\begin{aligned} \text{potentialSubclassOWL}(\text{Subclass}, \text{Superclass}) \leftarrow \\ \text{hierPhraseRelation}(C_{from}, C_{to}) \wedge \neg \text{hierWordNetRelation}(\text{Subclass}, \text{Superclass}) \end{aligned} \quad (4.10)$$

C_{to} é uma subclasse de C_{from} em potencial na especificação OWL de um dado mapa conceitual, se R é uma relação hierárquica de C_{from} para C_{to} no WordNet, e R não é uma relação que utiliza frase de ligação hierárquica:

$$\begin{aligned} \text{potentialSubclassOWL}(\text{Subclass}, \text{Superclass}) \leftarrow \\ \text{hierWordNetRelation}(\text{Subclass}, \text{Superclass}) \wedge \neg \text{hierPhraseRelation}(C_{from}, C_{to}) \end{aligned} \quad (4.11)$$

Onde,

R é uma relação que utiliza frase de ligação hierárquica, se R é uma relação do conceito C_{from} para o conceito C_{to} no mapa, e R é membro de um conjunto de frases de ligação que indicam que C_{from} é um subconceito de C_{to}:

$$\begin{aligned} \text{hierPhraseRelation}(R, C_{from}, C_{to}) \leftarrow \\ \text{relationCM}(R, C_{from}, C_{to}) \wedge R \in \text{AscHierPhraseSet} \end{aligned} \quad (4.12)$$

R é uma relação que utiliza frase de ligação hierárquica, se R é uma relação do conceito C_{from} para o conceito C_{to} no mapa, e R é membro de um conjunto de frases de ligação que indicam que C_{to} é um subconceito de C_{from} :

$$\begin{aligned} & \text{hierPhraseRelation}(R, C_{to}, C_{from}) \leftarrow \\ & \text{relationCM}(R, C_{from}, C_{to}) \wedge R \in \text{DescHierPhraseSet} \end{aligned} \quad (4.13)$$

e, R é uma relação hierárquica de C_{from} para C_{to} no WordNet, se R é uma relação do conceito C_{from} para o conceito C_{to} no mapa, e C_{from} é um hipônimo de C_{to} :

$$\begin{aligned} & \text{hierWordNetRelation}(R, C_{from}, C_{to}) \leftarrow \\ & \text{relationCM}(R, C_{from}, C_{to}) \wedge \text{hyponym}(C_{from}, C_{to}) \end{aligned} \quad (4.14)$$

R é uma relação hierárquica de C_{from} para C_{to} no WordNet, se R é uma relação do conceito C_{from} para o conceito C_{to} no mapa, e C_{from} é um hiperônimo de C_{to} :

$$\begin{aligned} & \text{hierWordNetRelation}(R, C_{to}, C_{from}) \leftarrow \\ & \text{relationCM}(R, C_{from}, C_{to}) \wedge \text{hypernym}(C_{from}, C_{to}) \end{aligned} \quad (4.15)$$

4.5 Relações de Composição

Uma relação de composição, também conhecida como relação *todo-parte*, ocorre entre dois conceitos quando um é uma parte, ou componente, do outro. No trecho de mapa conceitual da Figura 4.3, por exemplo, este tipo de relação está representado: *Objects* são compostos de *Methods* e *Attributes*.

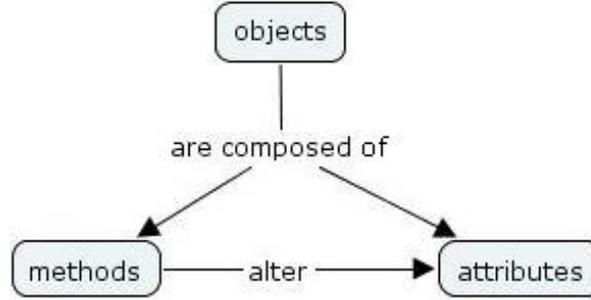


Figura 4.3: Composição entre conceitos identificada pela expressão *are composed of*.

Tais relações são mapeadas em OWL através de heurísticas cujas condições são similares às de relações de classificação (Seção 4.4), com a diferença de que, neste caso, as relações léxicas consultadas no WordNet são as de meronímia (por exemplo, *Attributes* é um meronímio de *Objects*) e holonímia (*Objects* é um holonímio de *Attributes*). As regras são mostradas a seguir.

R é uma subpropriedade de *hasPart*, e restrições existenciais e universais são estabelecidas sobre a propriedade R como condições necessárias na descrição de C_{from} com relação aos indivíduos de C_{to} , se existe uma relação R no mapa de C_{from} para C_{to} , R é membro de um conjunto de frases de ligação que indicam que C_{to} é uma parte de C_{from} , e existe uma relação de holonímia de C_{from} em relação a C_{to} :

$$\left(\begin{array}{l} subPropOfOWL(R, hasPart) \wedge \\ necesCondOWL(C_{from}, rest(\exists, \forall, R, C_{to})) \end{array} \right) \leftarrow \quad (4.16)$$

$$relationCM(R, C_{from}, C_{to}) \wedge R \in DescCompPhraseSet \wedge holonym(C_{from}, C_{to})$$

Onde,

C e C' são classes OWL e R é uma propriedade OWL, se uma condição necessária da forma $rest(\exists, \forall, R, C')$ é estabelecida na descrição de C :

$$\left(\begin{array}{l} classOWL(C) \wedge classOWL(C') \wedge \\ propOWL(R) \end{array} \right) \leftarrow necesCondOWL(C, rest(\exists, \forall, R, C')) \quad (4.17)$$

Do mesmo modo,

R é uma subpropriedade de $isPartOf$, e restrições existenciais e universais são estabelecidas sobre a propriedade R como condições necessárias na descrição de C_{from} com relação aos indivíduos de C_{to} , se existe uma relação R no mapa de C_{from} para C_{to} , R é membro de um conjunto de frases de ligação que indicam que C_{from} é uma parte de C_{to} , e existe uma relação de meronímia de C_{from} em relação a C_{to} :

$$\left(\begin{array}{l} subPropOfOWL(R, isPartOf) \wedge \\ necesCondOWL(C_{from}, rest(\exists, \forall, R, C_{to})) \end{array} \right) \leftarrow \quad (4.18)$$

$$relationCM(R, C_{from}, C_{to}) \wedge R \in AscCompPhraseSet \wedge meronym(C_{from}, C_{to})$$

A regra 1.16 diz que uma relação R em um mapa conceitual, de um todo para uma de suas partes, é estabelecida como subpropriedade de $hasPart$. O mesmo está especificado de forma similar na regra 1.18 com a propriedade $isPartOf$. OWL não provê construções embutidas para hierarquias parte-todo, como o faz para as relações de sub-classe (é-um). Existem, entretanto, padrões de projeto de ontologias em uso¹ onde uma propriedade nomeada $isPartOf$ e sua inversa $hasPart$ são empregadas como padrão na representação de relações parte-todo. A regra 1.16 tem o efeito de estabelecer a relação na ontologia como uma especialização da relação parte-todo genérica $hasPart$, ao mesmo tempo em que preserva seu nome conforme foi especificado no mapa conceitual. Além disso, uma relação de composição entre dois conceitos em um mapa conceitual dá origem na ontologia a condições necessárias, formuladas como restrições sobre propriedades, na descrição da classe que corresponderá ao conceito de origem. Para a relação entre *Objects* e *Attributes* no mapa da Figura 4.3, por exemplo, seriam resultantes as seguintes condições necessárias na descrição da classe *Objects*: $\exists areComposedOf Attributes$ e

¹Veja, por exemplo, Simple part-whole relations in OWL Ontologies, <http://www.w3.org/2001/sw/BestPracticesOEP/SimplePartWhole/simple-part-whole-relations-v1.5.html>.

$\forall areComposedOf\ Attributes$, significando que a classe *Objects* é uma subclasse de uma outra, anônima, de indivíduos que se relacionam, através da propriedade *areComposedOf*, com pelo menos um indivíduo da classe *Attributes*, e somente se relacionam através da propriedade *areComposedOf* com indivíduos da classe *Attributes*.

A restrição existencial significa, em outras palavras, que, para que algo seja um objeto, o mesmo deve ter pelo menos um atributo o que, ontologicamente falando, não é verdade. Entretanto, a classe *Objects*, nesta ontologia, não tem a pretensão de descrever objetos reais, mas sim a noção conceitual de objetos, expressa neste mapa em particular, como algo que tem atributos.

Após todas as relações binárias de composição de uma classe origem serem estabelecidas, um axioma de fechamento (*closure axiom*) sobre a propriedade da composição é especificada como uma condição necessária na descrição da classe de origem. Para o conceito *Objects* e a relação *are composed of* da Figura 4.3, por exemplo, a condição necessária é $\forall is_composed_of\ (Attributes \sqcup Methods)$ na descrição da classe *Objects*, significando que, de acordo com o escopo do mapa conceitual do qual se originou a ontologia, as únicas partes de *Objects* capturadas pela propriedade *isComposedOf* são *Attributes* e *Methods*. Devido à semântica da restrição universal ($\forall x, y \langle x, y \rangle \in isComposedOf \rightarrow y \in Attributes \cup Methods$), esta restrição, sozinha, poderia incluir indivíduos que não se relacionam através da propriedade *isComposedOf* com nenhum indivíduo. As restrições existenciais em *isComposedOf* em relação a *Attributes* e *Methods*, que também fazem parte da descrição de *Objects*, garantem que esta interpretação não se aplica à classe.

Relações de composição são transitivas por definição. Ainda de acordo com boas práticas de representação de relações parte-todo em OWL, o estabelecimento de *R* como uma subpropriedade de *isPartOf* leva à caracterização de ambas na ontologia como transitivas e de *hasPart* como inversa de *isPartOf*.

hasPart e *R* são propriedades transitivas e *hasPart* é a inversa de *isPartOf*, se *R* é uma

subpropriedade de hasPart:

$$\left(\begin{array}{l} transPropOWL(hasPart) \wedge \\ inverseOWL(hasPart, isPartOf) \end{array} \right) \leftarrow subPropOWL(R, hasPart) \quad (4.19)$$

A regra 4.20 abaixo é correspondente à 4.19 para relações de composição da parte para o todo.

$$\left(\begin{array}{l} transPropOWL(isPartOf) \wedge \\ inverseOWL(hasPart, isPartOf) \end{array} \right) \leftarrow subPropOWL(R, isPartOf) \quad (4.20)$$

Sendo que propriedades transitivas OWL são propriedades OWL:

$$propOWL(R) \leftarrow transPropOWL(R) \quad (4.21)$$

É importante notar que as propriedades *hasPart* e *isPartOf*, como permitido em OWL, são estabelecidas como propriedades sem domínio e contradomínio especificados, porque as restrições criadas já restringem os indivíduos às classes estabelecidas nas relações.

Assim como nas relações de classificação, o uso de frases de ligação de relações de composição entre conceitos pode indicar possíveis relações deste tipo, mesmo que não exista uma relação de meronímia ou holonímia correspondente no WordNet entre estes conceitos. O mesmo pode ocorrer no sentido inverso, quando estes tipos de relação entre os conceitos ocorrerem, mas o rótulo da relação no mapa não pertence ao conjunto de frases de ligação de composição.

Sendo assim, a heurística seguinte é capaz de identificar relações onde esta situação ocorre, tornando possível tratá-las.

C_{to} é uma parte de C_{from} em potencial na especificação OWL de um dado mapa conceitual, se R é uma relação que utiliza frase de ligação de composição, e R não é uma relação de

composição de C_{from} para C_{to} no WordNet:

$$\begin{aligned} &potentialCompositionOWL(Part, Whole) \leftarrow \\ &compPhraseRelation(C_{from}, C_{to}) \wedge \neg compWordNetRelation(Part, Whole) \end{aligned} \quad (4.22)$$

C_{to} é uma composição de C_{from} em potencial na especificação OWL de um dado mapa conceitual, se R é uma relação de composição de C_{from} para C_{to} no WordNet, e R não é uma relação que utiliza frase de ligação de composição:

$$\begin{aligned} &potentialCompositionOWL(Part, Whole) \leftarrow \\ &compWordNetRelation(Part, Whole) \wedge \neg compPhraseRelation(C_{from}, C_{to}) \end{aligned} \quad (4.23)$$

Onde,

R é uma relação que utiliza frase de ligação de composição, se R é uma relação do conceito C_{from} para o conceito C_{to} no mapa, e R é membro de um conjunto de frases de ligação que indicam que C_{from} é uma parte de C_{to} :

$$\begin{aligned} &compPhraseRelation(R, C_{from}, C_{to}) \leftarrow \\ &relationCM(R, C_{from}, C_{to}) \wedge R \in AscCompPhraseSet \end{aligned} \quad (4.24)$$

R é uma relação que utiliza frase de ligação de composição, se R é uma relação do conceito C_{from} para o conceito C_{to} no mapa, e R é membro de um conjunto de frases de ligação que indicam que C_{to} é uma parte de C_{from} :

$$\begin{aligned} &compPhraseRelation(R, C_{to}, C_{from}) \leftarrow \\ &relationCM(R, C_{from}, C_{to}) \wedge R \in DescCompPhraseSet \end{aligned} \quad (4.25)$$

e,

R é uma relação de composição de C_{from} para C_{to} no WordNet, se R é uma relação do conceito C_{from} para o conceito C_{to} no mapa, e existe uma relação de meronímia de C_{from} para C_{to} :

$$\begin{aligned} compWordNetRelation(R, C_{from}, C_{to}) \leftarrow \\ relationCM(R, C_{from}, C_{to}) \wedge meronym(C_{from}, C_{to}) \end{aligned} \quad (4.26)$$

R é uma relação de composição de C_{from} para C_{to} no WordNet, se R é uma relação do conceito C_{from} para o conceito C_{to} no mapa, e existe uma relação de holonímia de C_{from} para C_{to} :

$$\begin{aligned} compWordNetRelation(R, C_{to}, C_{from}) \leftarrow \\ relationCM(R, C_{from}, C_{to}) \wedge holonym(C_{from}, C_{to}) \end{aligned} \quad (4.27)$$

4.6 Relações Bidirecionais

Relações bidirecionais em mapas conceituais são aquelas que ocorrem em ambas as direções entre dois conceitos. Por exemplo, a relação de *casamento* entre os conceitos de *Marido* e *Esposa*. Tais relações são mapeadas como propriedades simétricas em OWL, como formulado em 4.28.

Uma propriedade simétrica OWL R é estabelecida entre indivíduos de C_{from} e C_{to} na especificação em OWL de um dado mapa conceitual, se R é uma relação de C_{from} para C_{to} e também de C_{to} para C_{from} no mapa:

$$symPropOWL(R, C_{from}, C_{to}) \leftarrow relationCM(R, C_{from}, C_{to}) \wedge relationCM(R, C_{to}, C_{from}) \quad (4.28)$$

Propriedades simétricas OWL são também propriedades OWL:

$$\text{propOWL}(P, C, C') \leftarrow \text{symPropOWL}(P, C, C') \quad (4.29)$$

4.7 Demais Relações

Além das relações entre conceitos e suas instâncias e relações de classificação e composição, mapas conceituais possuem diversas relações associativas regulares [2] como em *Leaves-produce-Sugar* and *Roots-grow in-Soil* na Figura 2.1. Tais relações são mapeadas como propriedades de objeto na representação em OWL, como formulado em 4.31.

Uma propriedade OWL R é estabelecida de indivíduos de C_{from} para indivíduos de C_{to} na especificação OWL de um dado mapa conceitual, se R é uma relação do conceito C_{from} para o conceito C_{to} no mapa, e C_{to} não é um indivíduo de C_{from} na ontologia, e não existe uma relação OWL de subclasses entre C_{from} e C_{to} , e R não é uma propriedade de composição:

$$\begin{aligned} \text{propOWL}(R, C_{from}, C_{to}) \leftarrow & \quad (4.30) \\ & \text{relationCM}(R, C_{from}, C_{to}) \wedge \neg \text{indivOWL}(C_{to}, C_{from}) \wedge \\ & \neg (\text{subclassOWL}(C_{from}, C_{to}) \vee \text{subclassOWL}(C_{to}, C_{from})) \wedge \neg \text{compPropOWL}(R) \end{aligned}$$

Onde,

*R é uma relação de composição se R for uma subpropriedade de *hasPart* ou de *isPartOf*:*

$$\text{compPropOWL}(R) \leftarrow (\text{subPropOWL}(R, \text{hasPart}) \vee \text{subPropOWL}(R, \text{isPartOf})) \quad (4.31)$$

Finalmente, propriedades OWL ocorrem entre indivíduos de classes OWL:

$$(\text{classOWL}(C) \wedge \text{classOWL}(C')) \leftarrow \text{propOWL}(P, C, C') \quad (4.32)$$

Capítulo 5

Ferramenta de Tradução

As duas etapas que compõem o processo de transcrição de mapas conceituais em uma ontologia de consenso, apresentado na Seção 3.1, são realizadas através dos módulos de software de Tradução e Mesclagem. Como módulo Tradutor, foi desenvolvida uma ferramenta em linguagem Java, que utiliza as heurísticas apresentadas para identificar elementos da ontologia nos mapas conceituais, e gera uma especificação em linguagem OWL do mesmo. O tradutor recebe como entrada um mapa conceitual e, através da aplicação das heurísticas apresentadas na Seção 4, implementadas na linguagem Prolog, retorna uma ontologia individual especificada em linguagem OWL cujo conteúdo heurísticamente corresponde aos conceitos e relações apresentados no mapa. A Figura 5.1 apresenta a interface de usuário da ferramenta, onde é possível informar a localização do mapa conceitual e o nome e localização do arquivo OWL a ser gerado. Na parte inferior, são apresentadas mensagens informativas ao usuário à medida que o processo de tradução é realizado.

A Figura 5.2 ilustra a estrutura interna do módulo Tradutor, formado por três componentes: o *Parser*, o Mapeador e o Gerador OWL. Cada um destes componentes são apresentados em maior detalhe nas seções seguintes.

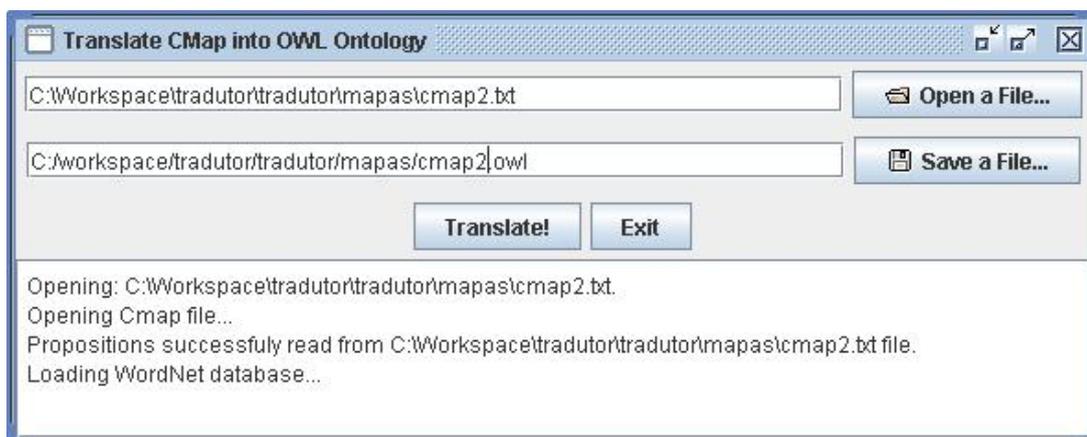


Figura 5.1: Interface de usuário da ferramenta de tradução

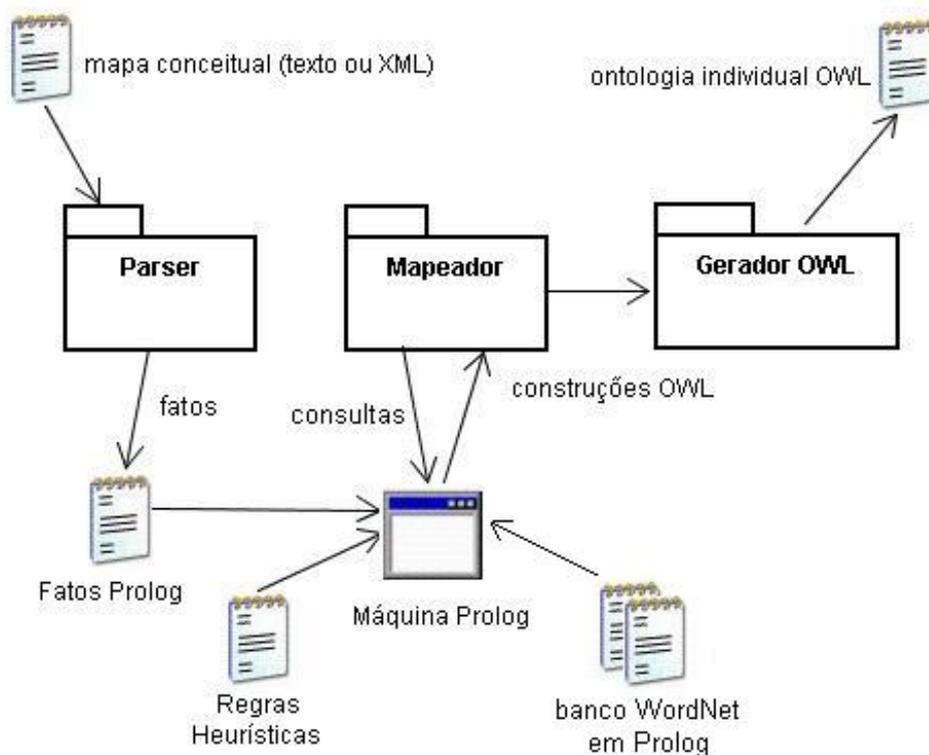


Figura 5.2: Arquitetura do módulo de tradução

5.1 Componente *Parser*

O *Parser* é o componente responsável por analisar sintaticamente o conteúdo do mapa conceitual informado e gerar uma representação deste conteúdo através fatos na linguagem

Prolog. Para que mapas conceituais, que consistem de uma representação gráfica, possam ser analisados pelo *Parser*, os mesmos são primeiramente convertidos em um formato no qual seu conteúdo possa ser capturado. Isto é feito através da funcionalidade de exportação do CMapTools, ambiente onde os mapas são construídos, que permite obter versões dos mapas nos formatos texto ou XML.

Primeiramente, foi desenvolvido um *parser* para analisar os arquivos de mapas conceituais em formato texto. Este arquivo é formado por proposições, uma em cada linha, sendo que cada tripla *conceito - frase de ligação - conceito* possui estes elementos separados por tabulações. A Figura 5.3 ilustra um exemplo de mapa conceitual exportado para formato texto.

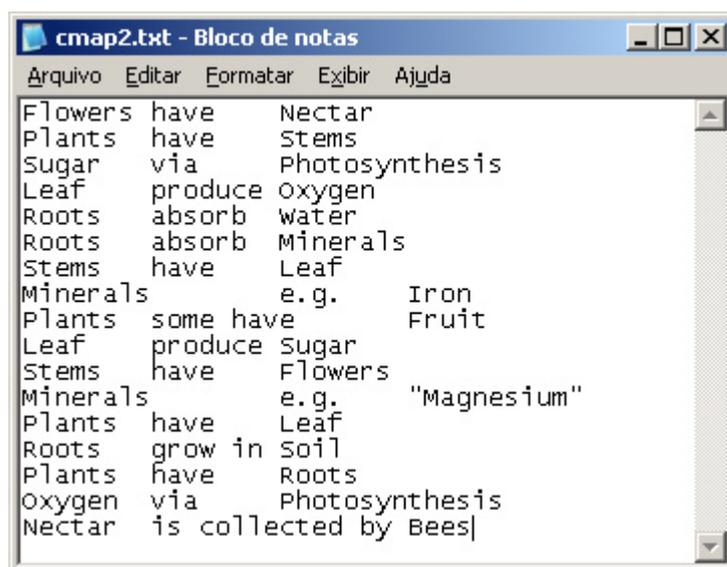


Figura 5.3: Mapa conceitual em formato Proposições como Texto

Com o intuito de oferecer maior flexibilidade no uso da ferramenta, foi desenvolvida posteriormente uma extensão do *parser* que analisa arquivos de mapas conceituais exportados para o formato XML. Para tanto, foi utilizada a API JDom [28], construída para criação e atualização de documentos XML, permitindo também a busca por *tags* específicas em documentos.

O arquivo em formato XML, diferentemente da representação em texto, possui também

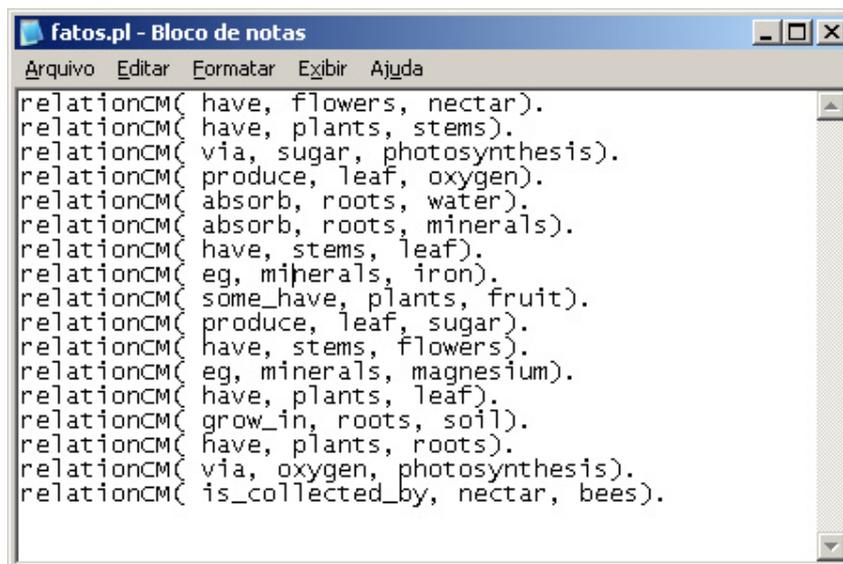
informações gráficas sobre o mapa conceitual, como disposição, tamanho e forma dos conceitos e relações, cores e fontes utilizadas. Ambos os tipos de informação estão igualmente armazenados em *tags* rotuladas *storableObject*. Desta forma, é tarefa do parser seleccionar somente os itens correspondentes a informações sobre conceitos e frases de ligação contidos no mapa, ignorando dados de apresentação.

No arquivo XML, de maneira simplificada, as proposições são formadas por um par ordenado de conexões, onde cada conexão, por sua vez, é formada por um par de nós (*firstNode* e *SecondNode*), que apontam para conceitos ou frases de ligação. Para ilustrar esta estrutura, a proposição *Tree - is a - Plant* seria formada por um par de conexões, Con1 e Con2, onde, em Con1, o valor de *FirstNode* seria *Tree* e de *SecondNode* seria *is a* e, em Con2, o valor de *FirstNode* seria *is a* e o de *SecondNode* seria *Plant*.

Apesar de mais complexo, este formato é mais próximo ao que é representado no mapa, por representar proposições onde vários conceitos participam e, além disso, evita a redefinição de um conceito ou frase de ligação cada vez que os mesmos reaparecem em diferentes proposições.

Em ambas as implementações, após identificadas as proposições, as mesmas são transcritas em fatos na linguagem Prolog, da forma $\text{relationCM}(R, C_{\text{from}}, C_{\text{to}})$, onde *R* é a frase de ligação, *C_{from}* é o conceito de origem da proposição e *C_{to}* é o conceito destino. O arquivo de fatos gerado é utilizado como entrada para as regras heurísticas. A Figura 5.4 apresenta o arquivo de fatos correspondente ao mapa convertido para texto da Figura 5.3.

É importante salientar que, no caso de frases de ligação ou de conceitos rotulados com mais de uma palavra, são criados átomos correspondentes substituindo-se o espaço entre as palavras pelo carácter de sublinha. Esta notação foi adotada para manter a compatibilidade com o padrão adotado na base de dados léxica WordNet, que é utilizada por algumas regras heurísticas, para representar termos com mais de uma palavra. Além disso, é feita a eliminação de caracteres como o ponto e substituição de aspas, que possuem



```
relationCM( have, flowers, nectar).
relationCM( have, plants, stems).
relationCM( via, sugar, photosynthesis).
relationCM( produce, leaf, oxygen).
relationCM( absorb, roots, water).
relationCM( absorb, roots, minerals).
relationCM( have, stems, leaf).
relationCM( eg, minerals, iron).
relationCM( some_have, plants, fruit).
relationCM( produce, leaf, sugar).
relationCM( have, stems, flowers).
relationCM( eg, minerals, magnesium).
relationCM( have, plants, leaf).
relationCM( grow_in, roots, soil).
relationCM( have, plants, roots).
relationCM( via, oxygen, photosynthesis).
relationCM( is_collected_by, nectar, bees).
```

Figura 5.4: Fatos em linguagem Prolog gerados pelo componente *Parser*

significado especial para a máquina de inferência Prolog, pelo predicado `quoted/1`.

5.2 Componente Mapeador

Uma vez gerados os fatos em Prolog contendo as proposições especificadas no mapa conceitual, o componente Mapeador submete à máquina de inferência Prolog os fatos gerados, as heurísticas previamente concebidas e especificação em Prolog da base de dados léxica do WordNet, de forma a obter como resposta construções OWL.

A comunicação entre a implementação em Java e a máquina de inferência Sicstus Prolog é realizada através da interface Jasper [15]. Na Figura 5.5 é mostrada a especificação em linguagem Prolog de uma das heurísticas, a de identificação de hierarquia.

O componente Mapeador, primeiramente, recupera as classes e propriedades de objeto inferidas de consultas à base de dados em Prolog. Para cada uma das classes obtidas são então inferidas a sua superclasse (através de relações de classificação), os indivíduos e as restrições sobre propriedades que descreverão a classe na ontologia (originadas de relações de composição). Finalmente, são estabelecidos os domínios e contra-domínios as propri-

```

% ----- subclass -----
subclassOWL(Cfrom, Cto) :-
    relationCM(R, Cfrom, Cto),
    asHierPhraseSet(R),
    s(Codfrom, _, Cfrom, _, _, _), s(Codto, _, Cto, _, _, _),
    hyp(Codfrom, Codto).

subclassOWL(Cto, Cfrom) :-
    relationCM(R, Cfrom, Cto),
    descHierPhraseSet(R),
    s(Codfrom, _, Cfrom, _, _, _), s(Codto, _, Cto, _, _, _),
    hyp(Codto, Codfrom).

```

Figura 5.5: Especificação em Prolog da heurística de identificação de hierarquia

idades inferidas, bem como as características de simetria, transitividade e propriedade inversa (no caso de relações de composição para as propriedades *hasPart* e *isPartOf*). As classes de domínio e contra-domínio, originadas de conceitos origem e destino nas relações dos mapas conceituais, não são especificadas quando uma propriedade é resultado da utilização da mesma frase de ligação entre diferentes pares de conceitos no mapa conceitual. Como exemplo pode ser citada a propriedade *known-as*, que tem origem nas relações *Combination - known as - Multiple inheritance* e *Operations - known as - Methods* do mapa conceitual da Figura 6.5. Neste caso, o domínio seria formado pela união indivíduos das classes *Combination* e *Operations*, e o contradomínio pela união dos indivíduos das classes *Multiple Inheritance* e *Methods*, o que especifica que indivíduos das classes *Operations* e *Multiple Inheritance* se relacionam por meio da propriedade *known as*, embora não estejam relacionados no mapa conceitual.

5.2.1 Composição de Heurísticas

Nas heurísticas de identificação de relações hierárquicas é possível observar que somente são assim classificadas as relações cujo rótulo pertence ao conjunto frases de ligação utilizadas para identificar este tipo de relação e cujos conceitos estão relacionados no base de fatos WordNet através de relações de hiponímia e hiperonímia. De forma similar, relações de composição somente são assim consideradas se, tanto através do conjunto de

frases de ligação para relações de composição, quanto por meio da existência de uma relação de meronímia ou holonímia no WordNet, as mesmas forem inferidas.

Entretanto, nos casos onde deseja-se representar no mapa tais relações mas os conceitos utilizados na proposição não existem no WordNet ou não estão ligados por meio das relações especiais acima citadas, as mesmas seriam representadas apenas como propriedades de objeto. De outro lado, o mesmo aconteceria quando existissem tais relações no WordNet mas a frase de ligação utilizada na proposição não pertencer ao conjunto de frases de ligação correspondente.

Por outro lado, a decisão de mapear relações hierárquicas (ou de composição) quando qualquer uma das duas condições fosse atendida poderia resultar em representações imprecisas na ontologia gerada, não correspondendo ao que foi representado no mapa conceitual. No caso de hierarquias identificadas somente pela frase de ligação, por exemplo, um termo ambíguo, como “is-a”, pode identificar igualmente um subtipo de um conceito ou uma característica de um conceito. Por outro lado, o fato de existirem relações de hiponímia ou hiperonímia no WordNet entre um par de conceitos relacionados no mapa conceitual não implica que a relação ali representada seja hierárquica: como o mapa conceitual é dependente de contexto, outro tipo de relação entre os conceitos pode estar sendo representada.

Além disso, consideramos que os resultados das heurísticas para identificação de instâncias não são suficientes para determinar quando este tipo de relação deve ser mapeado como uma hierarquia subclasse/classe ou como um indivíduo da classe. Isto ocorre porque as frases de ligação utilizadas na heurística de instâncias servem tanto para denotar instâncias de um conceito quanto tipos do mesmo, como é o caso da frase de ligação *such as*.

Por este motivo, após extrair os elementos OWL, o componente Mapeador identifica as situações acima ilustradas e, com base nos elementos OWL inferidos, oferece ao usuário a opção de decidir sobre possíveis relações de classificação e composição, bem como sobre

Confirmation

Partially identified Compositions

Nectar is a part of Flowers ? Yes No

Stems is a part of Plants ? Yes No

Leaf is a part of Plants ? Yes No

Roots is a part of Plants ? Yes No

Leaf is a part of Stems ? Yes No

Flowers is a part of Stems ? Yes No

Subclasses and Instances

Is Iron an individual or a subclass of Minerals ? Individual Subclass

Is Magnesium an individual or a subclass of Minerals ? Individual Subclass

Solve!

Figura 5.6: Resolução de possíveis relações de composição e decisão entre subclasses e indivíduos

relações com instâncias. A Figura 5.6 ilustra um processo de tradução onde foi encontrada no mapa conceitual, entre outras relações, uma relação entre conceito (*minerals*) e instância (*magnesium*). Neste caso, é perguntado ao usuário se a mesma deve ser mapeada como uma hierarquia classe/subclasse ou como uma classe com indivíduos.

5.3 Componente Gerador de OWL

O terceiro componente da arquitetura interna do módulo Tradutor é o Gerador OWL. O mesmo é responsável por escrever os elementos de ontologia retornados pelo Mapeador

```

<owl:Class rdf:ID="Flowers">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:someValuesFrom rdf:resource="#Nectar"/>
      <owl:onProperty rdf:resource="#have"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:allValuesFrom>
        <owl:Class rdf:about="#Nectar"/>
      </owl:allValuesFrom>
      <owl:onProperty rdf:resource="#have"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

Figura 5.7: Trecho de especificação em OWL da classe Flowers

como construções corretas da linguagem OWL.

Um arquivo OWL é inicialmente formado por um cabeçalho, no qual são definidas a localização da meta-linguagem OWL e do arquivo criado. Em seguida, são definidas as classes que compõem as ontologias e, dentro da especificação das mesmas, a definição de suas superclasses e de duas restrições sobre propriedades. Tanto a especificação de superclasse quanto as das restrições são feitas mediante a definição do elemento *SubClassOf*. Isso acontece porque, de fato, uma restrição sobre propriedade significa dizer que os membros de uma dada classe são subclasses de uma outra, anônima, formada pelos indivíduos que atendem aos critérios especificados na restrição. A Figura 5.7 mostra a especificação da classe Flowers obtida da transcrição do mapa conceitual sobre plantas, onde foi identificada uma relação de composição entre Flowers (todo) e Nectar (parte). Conforme apresentado nas heurísticas, este tipo de relação dá origem a duas restrições sobre a propriedade *have*, uma existencial (\exists), a outra, universal (\forall).

Ainda na tradução das restrições em construções OWL, é feito um tratamento para identificar restrições universais, na mesma classe, sobre a mesma propriedade, de forma a montar adequadamente o axioma de fecho (closure axiom), pelos motivos apresentados na Seção 4.5. Na Figura 5.8, que apresenta a especificação em OWL da classe *Plants*, é

```

<owl:Class rdf:ID="Plants">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:someValuesFrom rdf:resource="#Stems"/>
      <owl:onProperty rdf:resource="#have"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:someValuesFrom rdf:resource="#Leaf"/>
      <owl:onProperty rdf:resource="#have"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:someValuesFrom rdf:resource="#Roots"/>
      <owl:onProperty rdf:resource="#have"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:allValuesFrom>
        <owl:Class>
          <owl:unionOf rdf:parseType="Collection">
            <owl:Class rdf:about="#Stems"/>
            <owl:Class rdf:about="#Leaf"/>
            <owl:Class rdf:about="#Roots"/>
          </owl:unionOf>
        </owl:Class>
      </owl:allValuesFrom>
      <owl:onProperty rdf:resource="#have"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

Figura 5.8: Trecho de especificação em OWL da classe Plants

possível verificar que foi definida uma subclasse anônima para cada restrição existencial (*Stems*, *Leaf* e *Roots*), e que as restrições universais foram agrupadas em uma única definição de subclasse anônima.

No arquivo OWL, seguindo a definição de classes, são definidas as propriedades de objeto, nas quais são ajustados os atributos que definem se a propriedade é simétrica, inversa, transitiva ou subpropriedade de outra. Finalmente, os indivíduos das classe são montados.

A especificação em OWL da ontologia obtida do mapa conceitual informado pode, então, ser aberta diretamente no ambiente de edição de ontologias Protégé [38, 41]. A

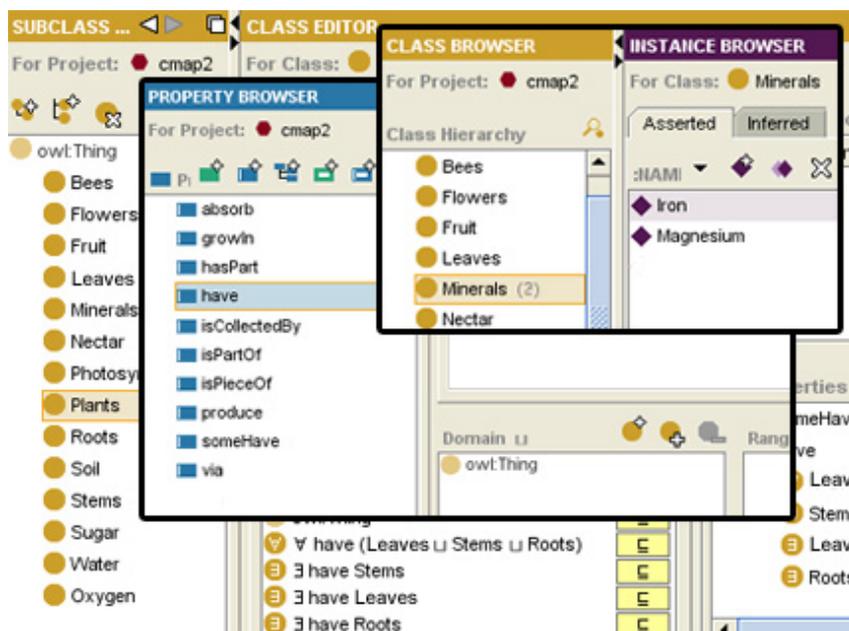


Figura 5.9: Ontologia obtida do mapa conceitual sobre Plantas aberta no Protégé

Figura 5.9 mostra partes do mapa conceitual sobre Plantas da Figura 2.1 transformado em uma ontologia após ser submetido ao módulo Tradutor e aberto no Protégé.

Capítulo 6

Estudo de Caso

Com o intuito de obter um conjunto de mapas conceituais descrevendo o mesmo domínio de conhecimento, foi realizada uma atividade para elaboração de mapas conceituais, em um ambiente acadêmico, com a turma da disciplina de Inovações Tecnológicas do curso de Tecnologia em Processamento de Dados do Centro Federal de Tecnologia do Amazonas (CEFET-AM), ocorrida durante o segundo semestre de 2005. Tal atividade teve por objetivo possibilitar a realização de experimentos de verificação do processo de geração de ontologias de consenso a partir de mapas conceituais. O experimento foi realizado em cinco etapas: preparação e realização das aulas, preparação dos mapas para submissão ao processo, tradução de cada um dos mapas para OWL, mesclagem das ontologias individuais e avaliação do processo. Cada uma destas etapas está descrita nas seções a seguir.

6.1 Preparação e Realização de Atividades de Construção de Mapas Conceituais

No decorrer da disciplina, foram ministradas 7 aulas abordando os temas Orientação a Objetos e Padrões de Projeto, constantes da ementa da disciplina. Em cada aula, foi

solicitado aos alunos que elaborassem Mapas Conceituais, que foram utilizados como ferramenta de avaliação dos alunos na disciplina, o que serviu para estimulá-los a construir os mapas dentro do prazo estabelecido. Cada novo mapa construído tinha como condição ser uma complementação do mapa anteriormente elaborado, agregando os novos conceitos aprendidos. A cada nova versão, conceitos e relações definidos no mapa anterior podiam também ser modificados. Os mapas construídos em cada aula foram avaliados pelo professor segundo as diretrizes de mapas conceituais bem construídos, sendo feitas sugestões e recomendações para construção do mapa seguinte.

A edição dos mapas conceituais foi feita utilizando o software CMAP Tools v. 3.10. Os mapas foram enviados para o e-mail da disciplina até 2 dias após a aula, no formato de arquivo da ferramenta CMapTools (.cmap). Esta limitação de prazo serviu para garantir que cada versão de mapa contivesse somente os conceitos aprendidos na aula correspondente.

A turma era composta por 12 alunos. Na primeira aula, foi apresentada a teoria sobre mapas conceituais e a ferramenta CMapTools, a ser utilizada no experimento. Em seguida, foi apresentada a teoria de orientação a objetos [16]. A segunda aula foi relativa a Padrões de Projeto [17] e, nas aulas subsequentes, foram apresentados os padrões Factory, Abstract Factory, Adapter, Composite, Template e Observer. Para o experimento de tradução de mapas conceituais em ontologias de domínio, foi selecionado o grupo de mapas conceituais sobre Orientação a Objetos por possuírem uma menor quantidade de conceitos: os mapas seguintes foram versões mais completas do mapa inicial. Como alguns alunos faltaram aulas e entregaram mapas com atraso, foram selecionados apenas 4 mapas conceituais deste grupo, sendo estes de alunos que assistiram a todas as aulas, entregaram os mapas com pontualidade e cujo conteúdo estava de acordo com as diretrizes de mapas conceituais bem construídos. Nas Figuras 6.1, 6.2, 6.4 e 6.3, são apresentados os mapas selecionados.



Figura 6.2: Mapa original do aluno 2

software, foi utilizado o banco de dados léxico WordNet, construído no idioma inglês. Por este motivo, tornou-se necessário que os mapas conceituais obtidos através deste experimento fossem traduzidos para este idioma. Alguns ajustes durante a tradução de expressões em português para o inglês foram necessários, visto que, nesta última linguagem, a ordem de sintagmas adotada na construção de orações é diferente.

Como exemplo disso, podemos citar o uso de adjetivos, que ocorrem antes de substantivos na língua inglesa. O mapa conceitual da Figura 6.3 ilustra uma situação onde este problema ocorre na proposição *objetos - representam coisas - reais / abstratas*, onde uma tradução literal levaria a *Objects - represent things - real / abstract*.

Entretanto, este tipo de tradução não respeita a ordem de sintagmas do inglês, onde o adjetivo ou qualidade deve vir antes do nome. Sendo assim, a proposição teve de ser ajustada durante a tradução para o inglês, como pode ser visto na Figura 6.7, onde a palavra *things* foi incorporada em ambos os conceitos, *real* e *abstract*.

Além dos ajustes inerentes à tradução de português para inglês, foram necessários ajustes em proposições mal-formadas, ou seja, que não possuíam a forma *conceito - frase*

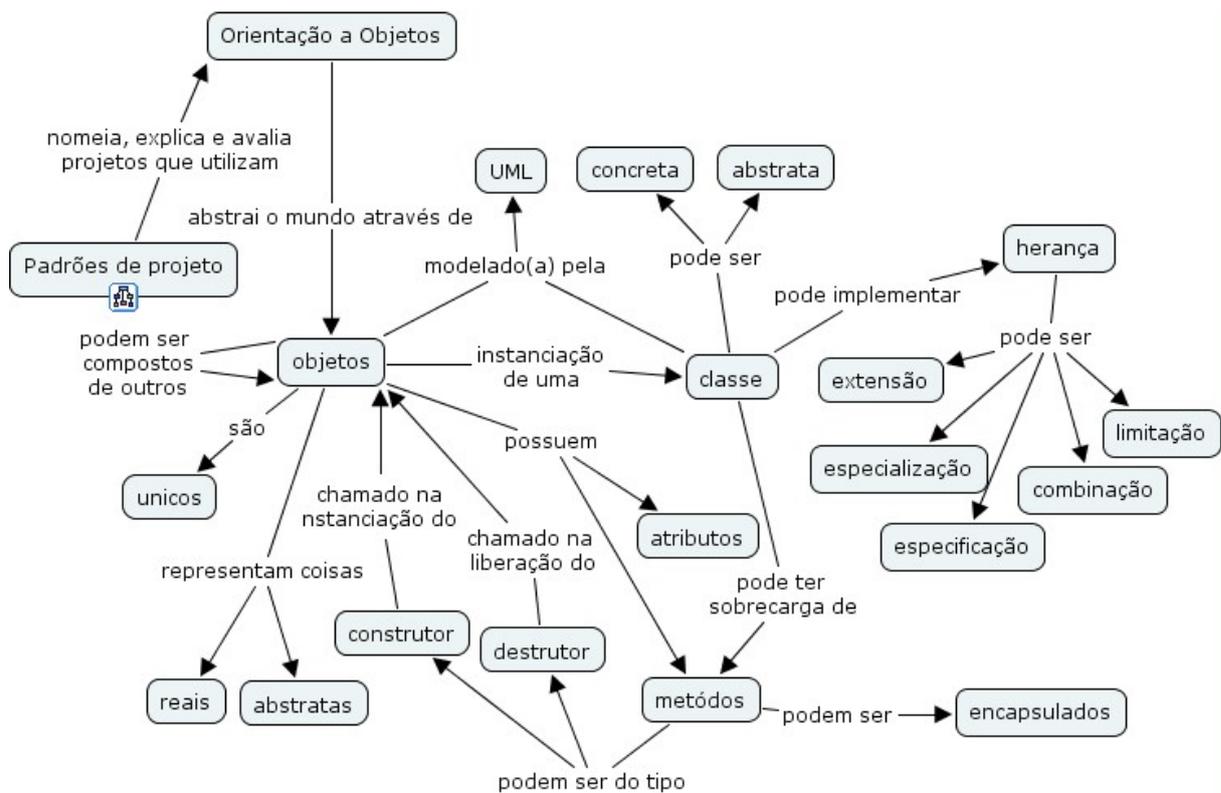


Figura 6.3: Mapa original do aluno 3

de ligação - conceito. Tais proposições foram reescritas, como pode ser visto nas Figura 6.3, na proposição *Padrão de projeto - nomeia, explica e avalia projetos que utilizam - Orientação a objetos*. A frase de ligação desta proposição foi dividida em outras três, sendo decomposta novamente para gerar o conceito *projetos* e a nova frase de ligação *que utilizam*, conforme visto na Figura 6.7. Outra situação semelhante ocorre na Figura 6.6, onde a seguinte proposição foi adaptada durante a tradução para inglês, incorporando-se um verbo à frase de ligação e modificando o conceito *proteção* em tornou *mecanismo de proteção*:

encapsulamento - mecanismo de - proteção

encapsulamento - é um - mecanismo de proteção

encapsulation - is a - protection mechanism

Nesta outro caso, especificamente, trata-se de uma única proposição, com um verbo

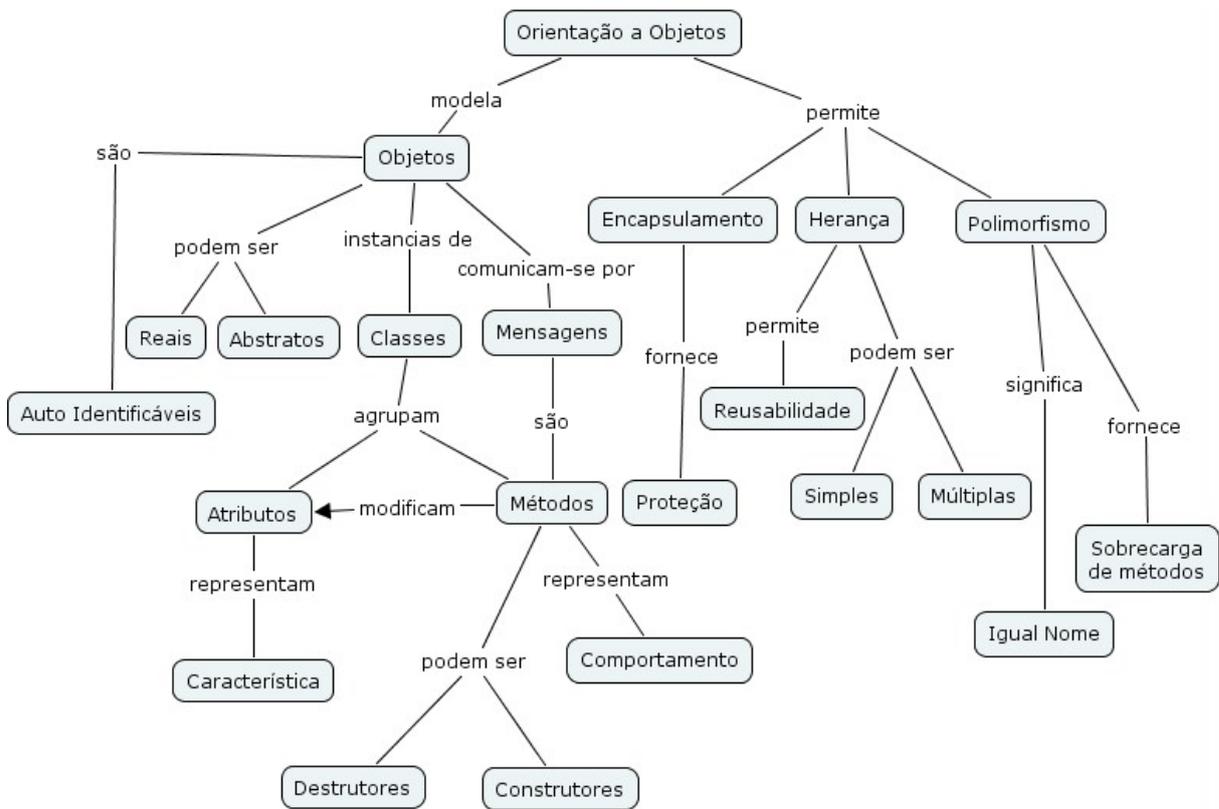


Figura 6.4: Mapa original do aluno 4

transitivo direto e indireto, resultando em um conceito destino formado por dois outros.

abstração - mapeia - objetos reais em objetos abstratos

Para resolver este problema, a proposição foi decomposta em outras duas:

abstração - mapeia - objetos reais

objetos reais - são mapeados em - objetos abstratos

Um outro problema ocorreu com nomes de conceitos que utilizam caracteres especiais, como é o caso do conceito *C++* e das proposições nas quais o mesmo está envolvido, ilustradas pela Figura 6.1. Tal representação resultaria em erro de interpretação das proposições geradas em Prolog, visto que, para esta linguagem, átomos não podem conter caracteres que não sejam letras maiúsculas, minúsculas e números, sempre começando

com uma letra minúscula. Tal conceito foi, então, transcrito para o inglês como *C plus plus*, como ilustrado na Figura 6.5.

Por fim, foi realizada uma correção semântica na proposição *Superclasse - especializa - subclasse*, do mapa conceitual da Figura 6.2, invertendo-se a direção em que a relação ocorre.

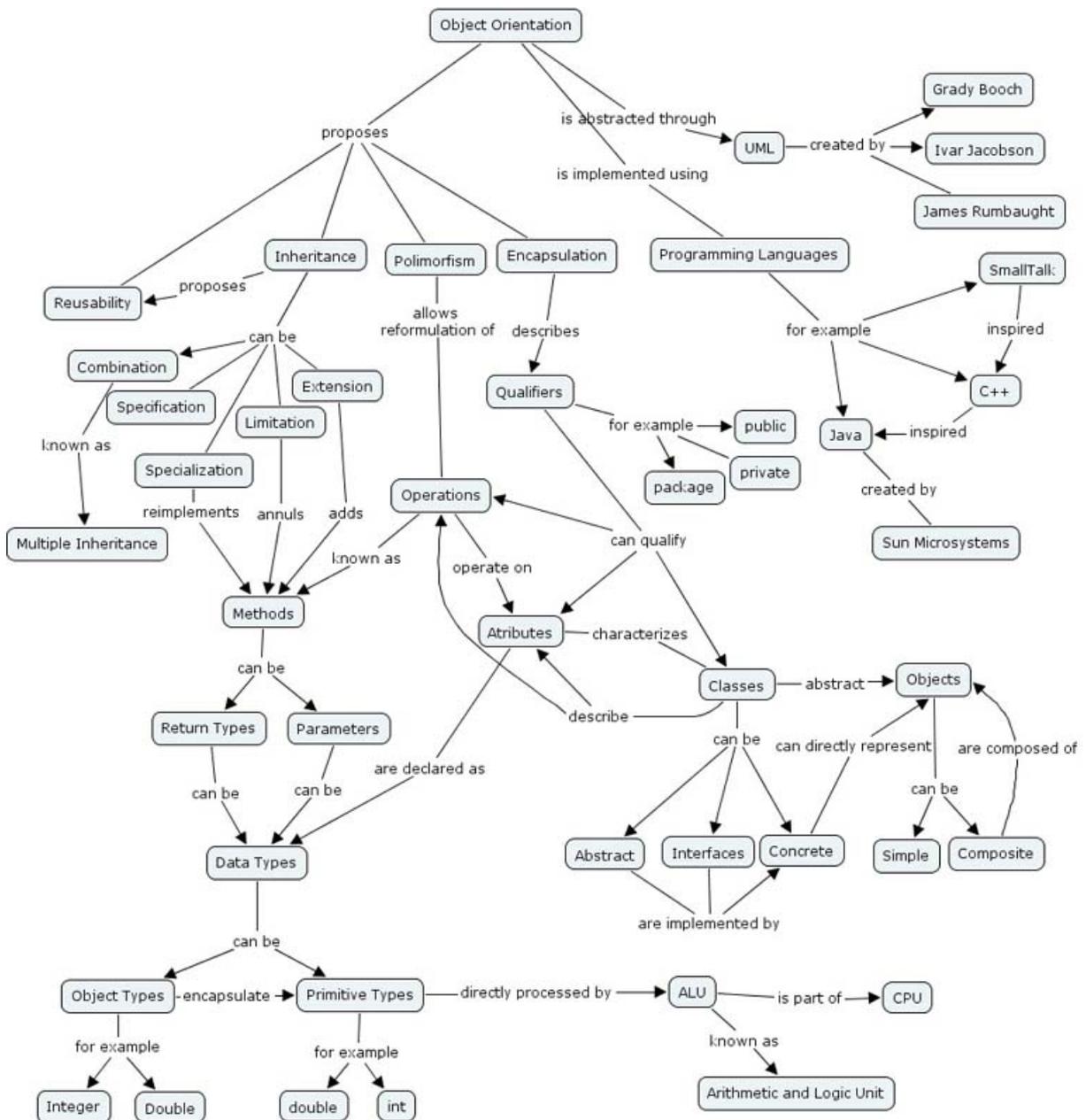


Figura 6.5: Mapa do aluno 1 em inglês

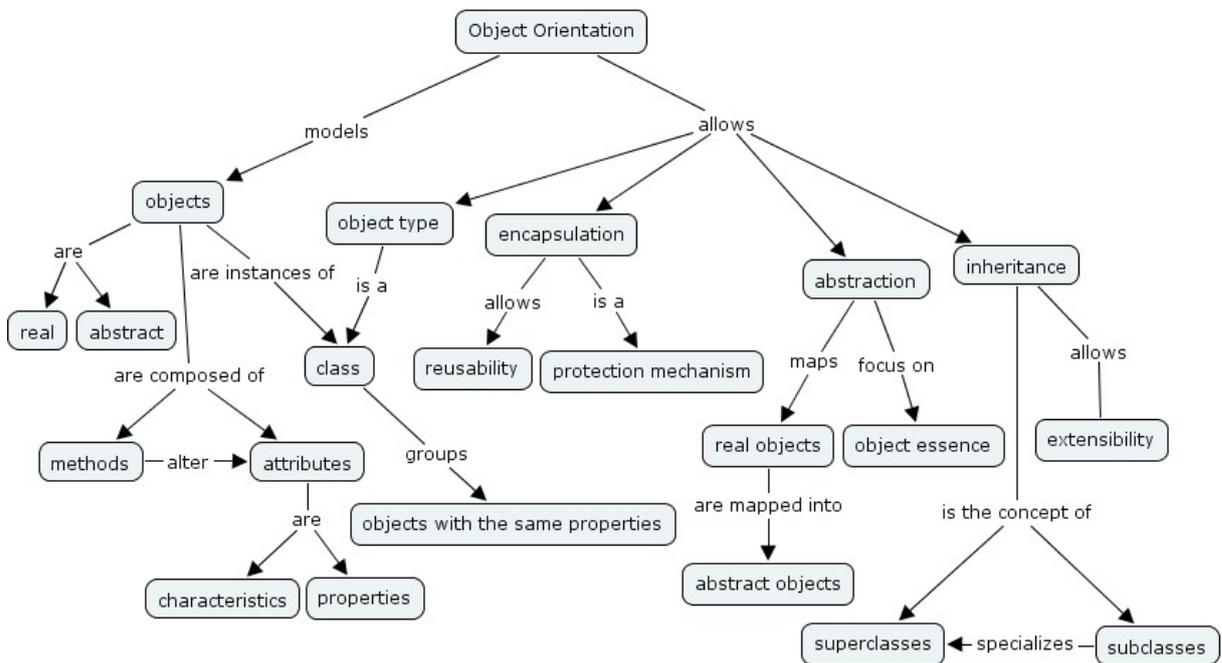


Figura 6.6: Mapa do aluno 2 em inglês

6.3 Tradução

De forma a executar a primeira etapa do processo de transcrição de mapas conceituais em ontologias de domínio, cada um dos quatro mapas selecionados foi submetido à ferramenta de tradução. Durante a tradução dos mapas, nenhuma relação hierárquica ou de composição foi identificada completamente, por frases de ligação e por relações semânticas no WordNet. Foram, entretanto, encontrados diversos casos de relações deste tipo somente pela heurística de uso de frases de ligação.

Durante a tradução do mapa na Figura 6.5, foram apresentadas ao usuário 14 possíveis relações hierárquicas, conforme apresentado na Tabela 6.1. Dentre as sugestões apresentadas, apenas duas não correspondiam realmente a hierarquias de conceitos: as originadas das proposições *return_types - can_be - data_types* e *parameters - can_be - data_types*. Ambas indicam apenas que tanto parâmetros quanto tipos de retorno de um método são tipos de dados, mas não indicam necessariamente uma classificação dos conceitos *parameters* e *return_types*.

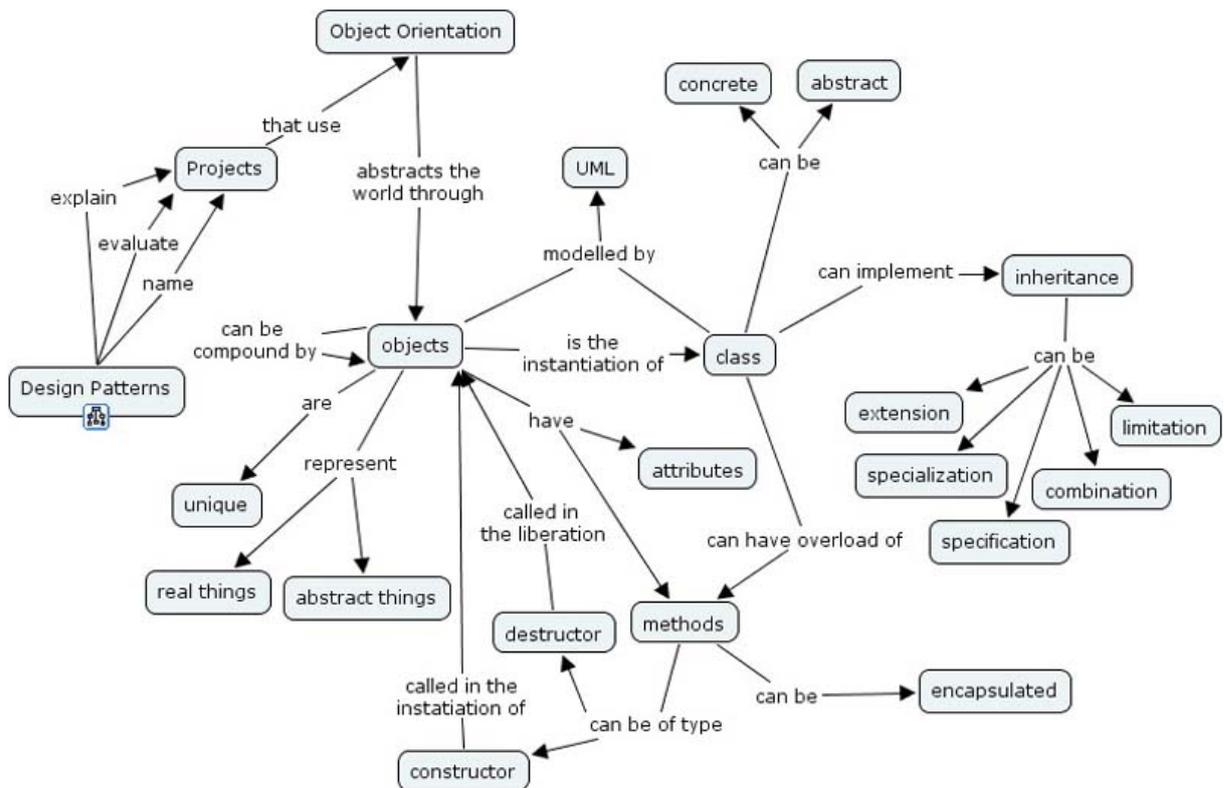


Figura 6.7: Mapa do aluno 3 em inglês

Na tradução deste mapa foram apresentadas também duas possíveis relações de composição, conforme mostra a Tabela 6.2. Ambas as sugestões foram aceitas. Desta forma, a frase de ligação *is_part_of*, da proposição *alu - is_part_of - cpu*, tornou-se subpropriedade da propriedade *isPartOf*. Similarmente, a frase de ligação *are_composed_of* da proposição *compound - are_composed_of - objects* tornou-se uma subpropriedade de *hasPart*. Ainda, de acordo com as heurísticas apresentadas na Seção 4, as propriedades *hasPart* e *isPartOf* são inversas entre si e transitivas. A ontologia intermediária obtida da tradução deste mapa é apresentada na Figura 6.9.

O mapa da Figura 6.5 foi ainda o único onde foram identificadas relações entre objetos e instâncias, perguntando ao usuário quais relações deveriam ser representadas através de indivíduos e quais relações deveriam ser representadas através de hierarquias de classes. Tais relações são apresentadas na Tabela 6.3.

Conforme a interpretação pessoal do usuário do experimento, apenas as relações entre a

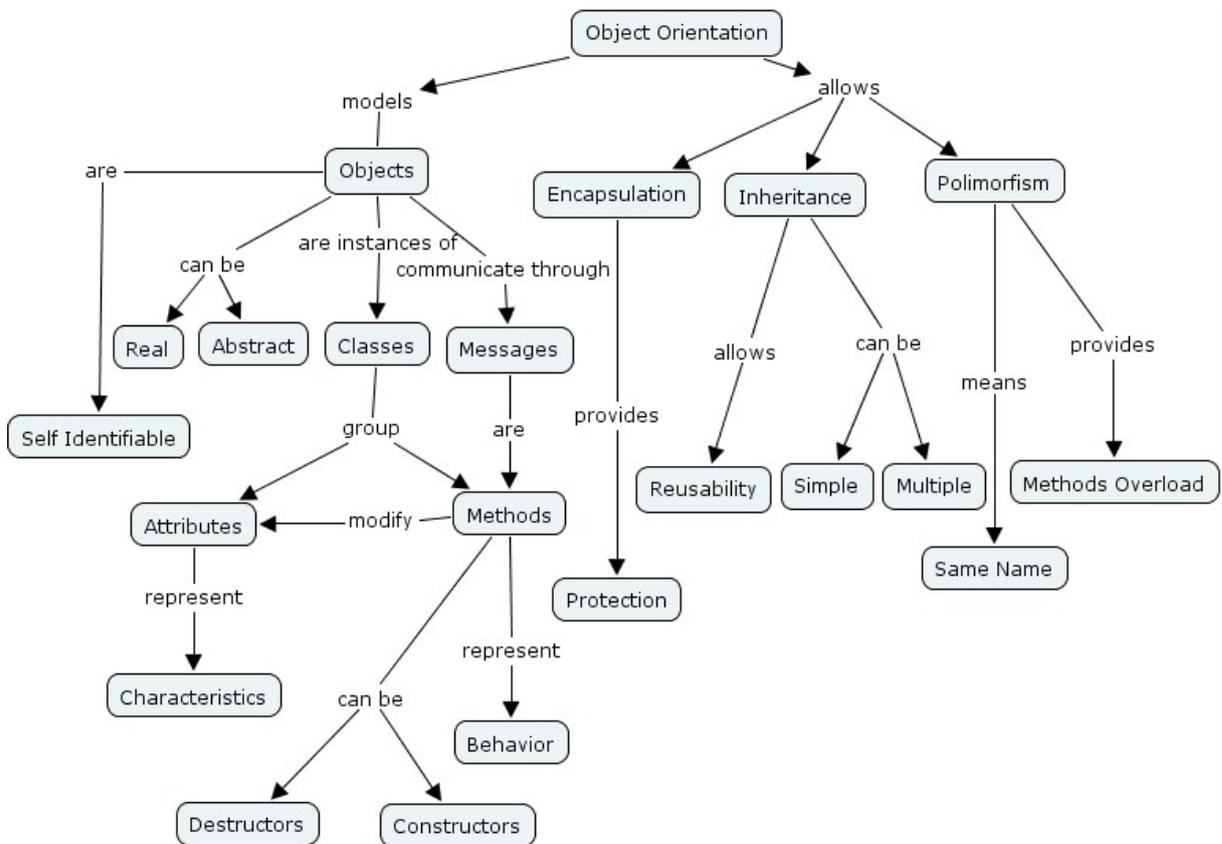


Figura 6.8: Mapa do aluno 4 em inglês

classe *Programming_languages* e as demais *Java*, *C_plus_plus* e *Smalltalk* representava uma relação entre classe e indivíduos, sendo as demais relações identificadas como hierarquias de classes.

O mapa da Figura 6.5 contém o mesmo conceito especificado duas vezes, um iniciando com letra maiúscula (*Double*), o outro não (*double*). A intenção do autor, durante a construção do mapa, era a de fazer distinção entre o tipo primitivo de dados do Java, *double*, e a classe que o representa, *Double*. Entretanto, como na linguagem OWL não há distinção de maiúsculas e minúsculas, apenas uma classe foi criada, durante a tradução do mapa, para representar ambos os conceitos.

No mapa da Figura 6.6, das duas relações hierárquicas apresentadas para decisão do usuário, uma não correspondia a este tipo de relação. A proposição, *encapsulation - is_a - protection_mechanism*, descreve uma característica do conceito encapsulamento e não que

Hierarquias parcialmente identificadas		
Superclasse	Subclasse	Decisão
Data_types	Object_types	sim
	Primitive_types	sim
Classes	Interface	sim
	Concrete	sim
	Abstract	sim
Inheritance	Specialization	sim
	Extension	sim
	Combination	sim
	Limitation	sim
	Specification	sim
Objects	Simple	sim
	Compound	sim
Return_types	Data_types	não
Parameters	Data_types	não

Tabela 6.1: Possíveis hierarquias identificadas na ontologia do aluno 1

encapsulamento é um tipo especial de mecanismo de proteção. As relações hierárquicas identificadas parcialmente são apresentadas nas Tabelas 6.4 e 6.5.

Nas possíveis relações de composição identificadas, foi apresentada uma proposição que também não corresponde a uma relação Todo - Parte. A mesma foi identificada pelo uso da frase de ligação *groups*, na proposição *classes - groups - Objects_with_the_same_properties* do mapa conceitual da Figura 6.6.

Como ocorre com relações do tipo composição, a propriedade *are_composed_of*, da relação entre o conceito *objects* com os conceitos *methods* e *attributes*, foi transcrita como uma subpropriedade de *hasPart*.

A ontologia intermediária obtida da tradução do mapa da Figura 6.6 é apresentada na Figura 6.10.

A ferramenta de tradução apresentou dez possíveis relações hierárquicas e duas composições ao traduzir o mapa na Figura 6.7. As sugestões são mostradas nas Tabelas 6.6 e 6.7. Das sugestões de relações hierárquicas apresentadas, apenas uma não correspondia realmente a uma hierarquia de conceitos, originária da proposição *methods - can_be - en-*

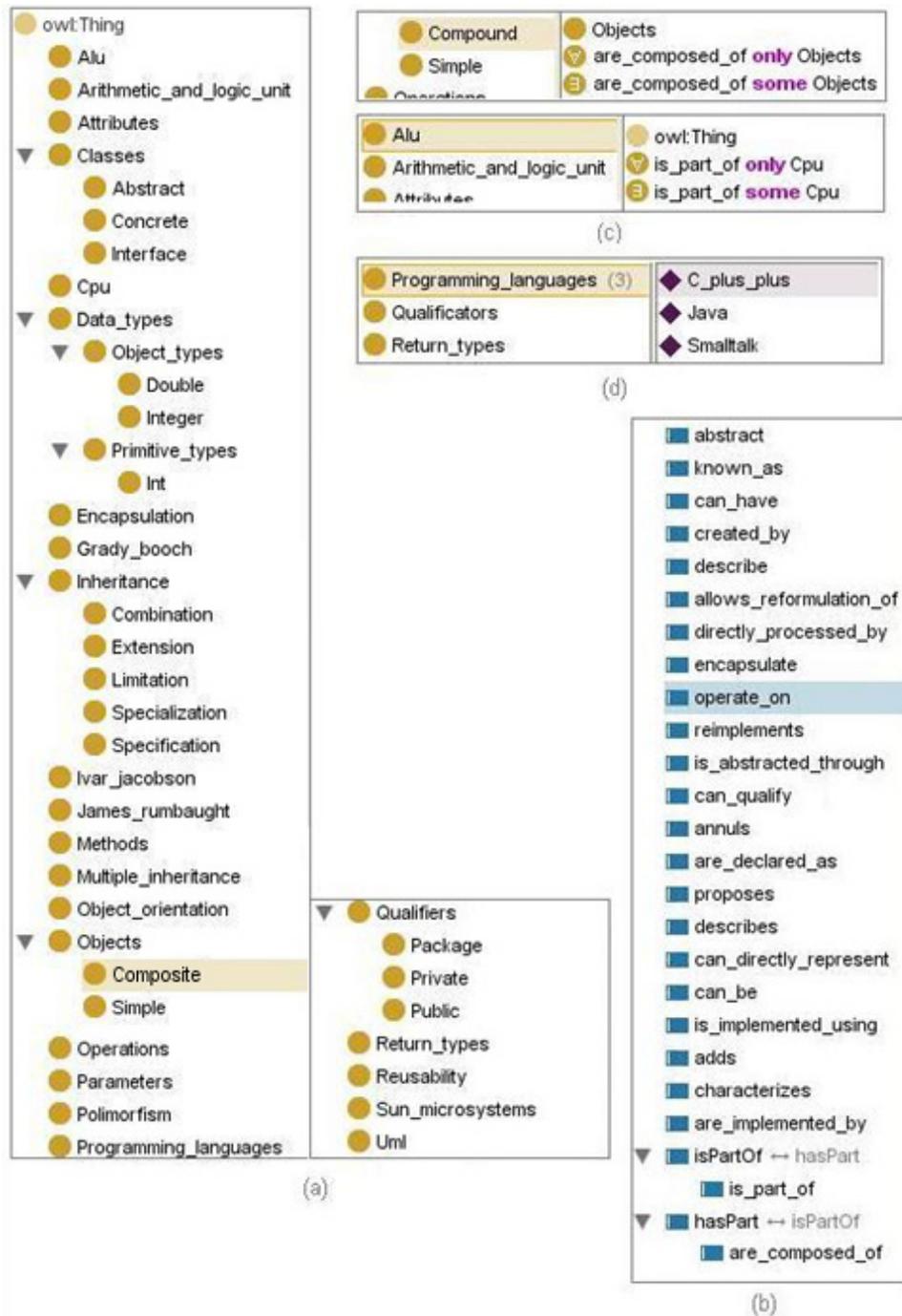


Figura 6.9: Ontologia intermediária obtida do mapa do aluno 1: hierarquia (a), propriedades (b), composições (c) e instâncias (d)

Composições parcialmente identificadas		
Todo	Parte	Decisão
Compound	Objects	sim
Cpu	Alu	sim

Tabela 6.2: Possíveis composições identificadas na ontologia do aluno 1

Subclasses e Instâncias		
Indivíduo / Subclasse	Da Classe	Decisão
Primitive_types	int	subclasse
	double	subclasse
Object_types	Integer	subclasse
	Double	subclasse
Qualifiers	Public	subclasse
	Package	subclasse
	Private	subclasse
Programming_languages	C_plus_plus	indivíduo
	Java	indivíduo
	Smalltalk	indivíduo

Tabela 6.3: Diferenciação entre subclasses e instâncias da ontologia do aluno 1

Hierarquias parcialmente identificadas		
Superclasse	Subclasse	Decisão
Superclasses	Subclasses	sim
Encapsulation	Protection_mechanism	não

Tabela 6.4: Possíveis hierarquias identificadas na ontologia do aluno 2

Composições parcialmente identificadas		
Todo	Parte	Decisão
Objects	Methods	sim
	Attributes	sim
Classes	Objects_with_the_same_properties	não

Tabela 6.5: Possíveis composições identificadas na ontologia do aluno 2

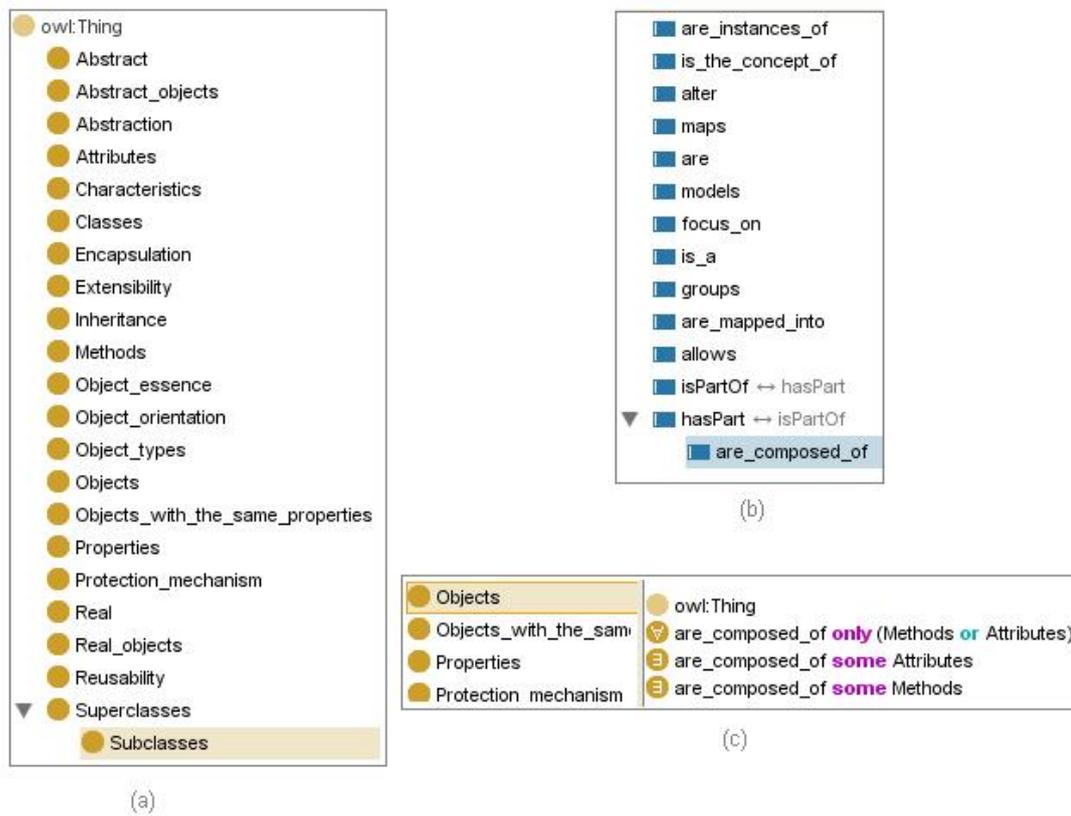


Figura 6.10: Ontologia intermediária obtida do mapa do aluno 2: hierarquia (a), propriedades (b) e composição (c)

capsulated. Esta relação demonstra apenas uma característica do conceito *methods*, não significando uma classificação deste conceito.

Uma vez identificada como relação de composição, a frase de ligação *have*, das proposições *objects - have - attributes* e *objects - have - methods*, tornou-se subpropriedade da propriedade *hasPart*. Foi também definida sua propriedade inversa, *isPartOf*, e ambas foram marcadas como transitivas.

A Figura 6.11 apresenta os elementos da ontologia intermediária obtida da tradução do mapa da Figura 6.7.

Ao ser traduzido, o mapa da Figura 6.8 apresentou seis possíveis relações hierárquicas e duas de composição, tendo todas sido aceitas. Tais relações estão elencadas nas Tabelas 6.8 e 6.9.

Hierarquias parcialmente identificadas		
Superclasse	Subclasse	Decisão
Methods	Constructor	sim
	Destructor	sim
	Encapsulated	não
Class	Concrete	sim
	Abstract	sim
Inheritance	Specialization	sim
	Extension	sim
	Combination	sim
	Limitation	sim
	Specification	sim

Tabela 6.6: Possíveis hierarquias identificadas na ontologia do aluno 3

Composições parcialmente identificadas		
Todo	Parte	Decisão
Objects	Methods	sim
	Attributes	sim

Tabela 6.7: Possíveis composições identificadas na ontologia do aluno 3

Como consequência da relação de composição identificada, a frase de ligação *group*, das proposições *classes - group - attributes* e *classes - group - methods*, tornou-se sub-propriedade da propriedade *hasPart*. Novamente, foi também definida sua propriedade inversa *isPartOf* e ambas foram marcadas como transitivas.

Na Figura 6.12 são apresentados os elementos da ontologia intermediária obtida da tradução do mapa da Figura 6.8.

Todas as quatro ontologias foram verificadas quanto à consistência com o uso do *reasoner* Racer [23].

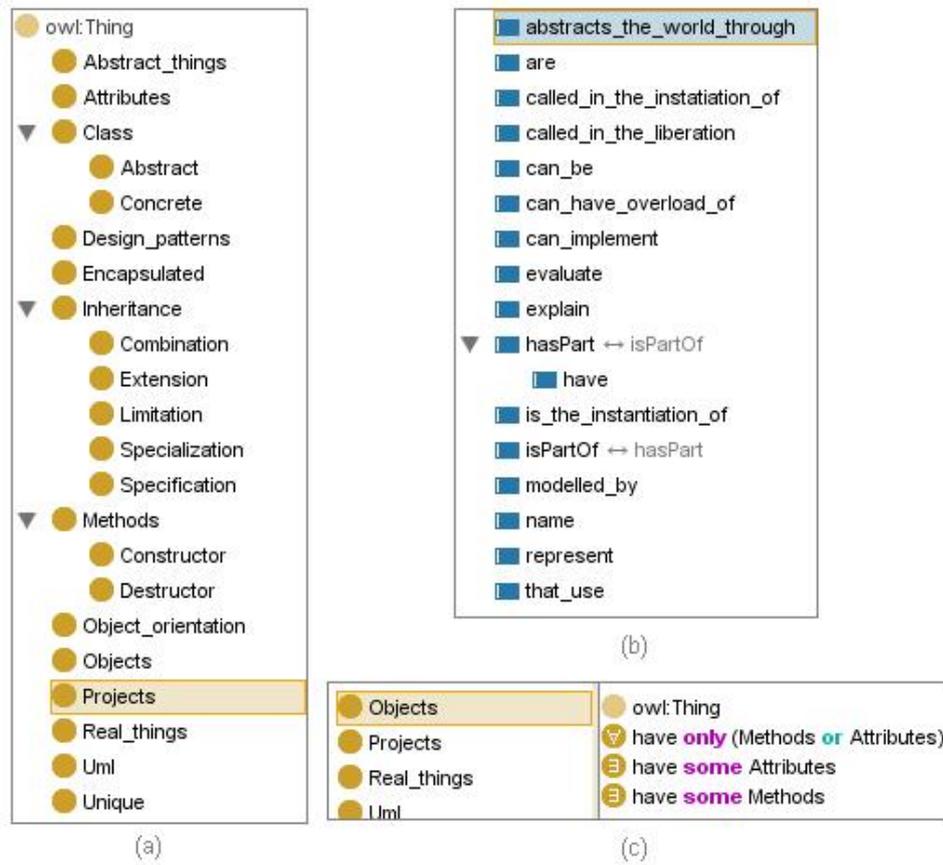


Figura 6.11: Ontologia intermediária obtida do mapa do aluno 3: hierarquia (a), propriedades (b) e composição (c)

Hierarquias parcialmente identificadas		
Superclasse	Subclasse	Decisão
Methods	Constructor	sim
	Destructor	sim
Objects	Real	sim
	Abstract	sim
Inheritance	Multiple	sim
	Simple	sim

Tabela 6.8: Possíveis hierarquias identificadas na ontologia do aluno 4

Composições parcialmente identificadas		
Todo	Parte	Decisão
Classes	Methods	sim
	Attributes	sim

Tabela 6.9: Possíveis composições identificadas na ontologia do aluno 4

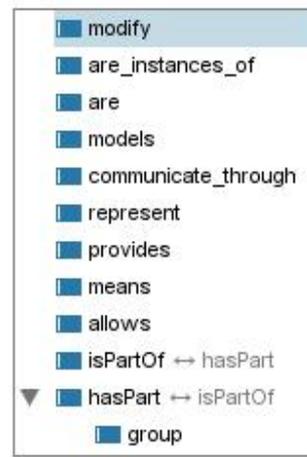
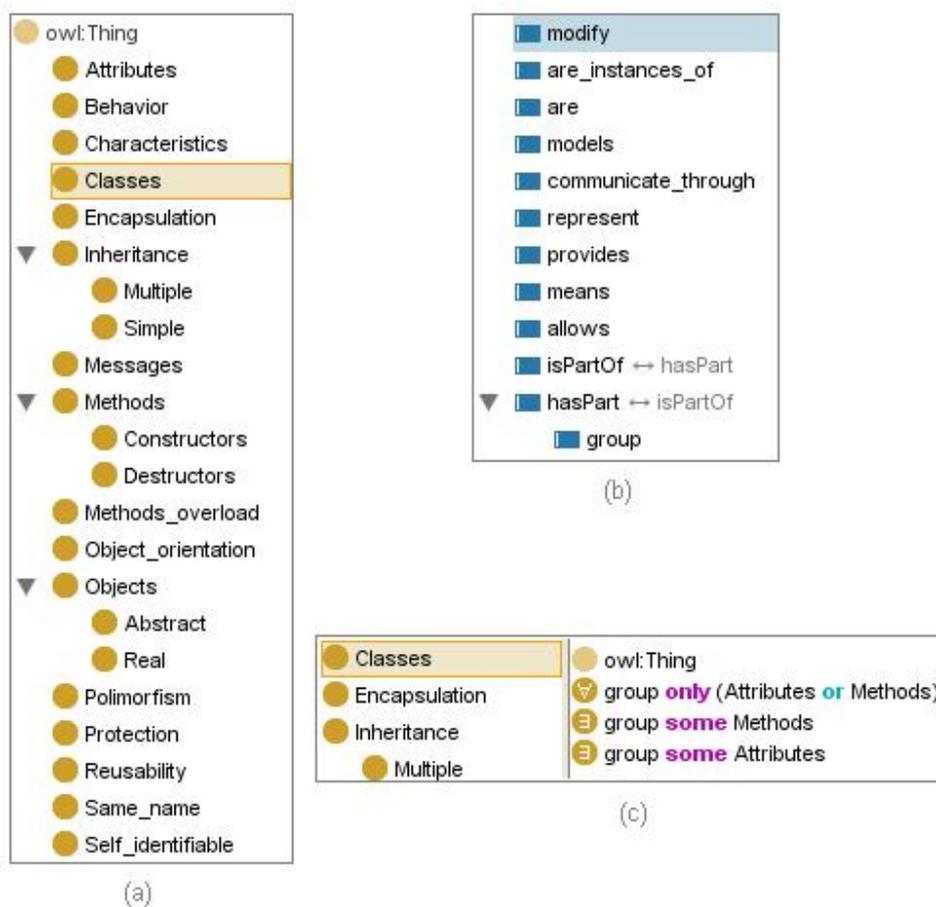


Figura 6.12: Ontologia intermediária obtida do mapa do aluno 4: hierarquia (a), propriedades (b) e composição (c)

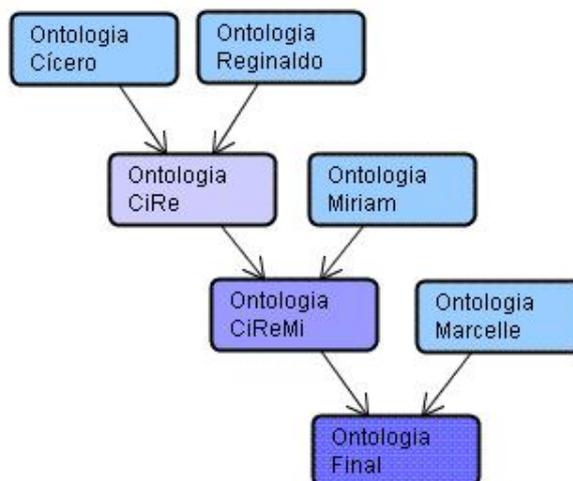


Figura 6.13: Ordem de mesclagem das ontologias intermediárias

6.4 Mesclagem

Uma característica da ferramenta de mesclagem escolhida PROMPT [39] é a de mesclar ontologias em pares. Por este motivo, a segunda fase do processo de transcrição, que corresponde à mesclagem das ontologias traduzidas, foi realizada mesclando-se duas ontologias de cada vez. O resultado da mesclagem de um par de ontologias foi sendo mesclado sucessivamente à ontologia seguinte, conforme ilustra a Figura 6.13. As ontologias foram mescladas começando-se pela mais complexa e indo até a mais simples, tendo sido seguida esta ordem como prioridade na escolha de rótulos de termos mesclados. A complexidade de cada ontologia foi medida pela quantidade de conceitos e de propriedades contida em cada uma.

Na primeira mesclagem realizada, entre as ontologias dos alunos 1 6.9 e 2 6.10, foram mescladas 10 classes e 3 propriedades de objeto. As classes e propriedades equivalentes destas duas ontologias que foram mescladas são apresentadas na Tabela 6.10. Na mesclagem de classes, foram acumuladas na ontologia resultante as características das classes originais. A classe *Inheritance*, por exemplo, é domínio da propriedade *is_the_concept_of*, conforme a ontologia do aluno 2, e que possui como subclasses *Combination*, *Extension*,

Elementos Mesclados - Ontologia 1-2	
Classes	Propriedades
Inheritance	are_composed_of
Methods	hasPart
Attributes	isPartOf
Object_types	
Reusability	
encapsulation	
Classes	
Objects	
Abstract	
Object_orientation	

Tabela 6.10: Classes e propriedades mescladas entre as ontologias 1 e 2

Limitation, *Specialization* e *Specification*, conforme a ontologia do aluno 1.

Além desta, as principais contribuições da ontologia do aluno 2 foram a composição representada na classe *Objects*, através da propriedade *are_composed_of* com as classes *Methods* e *Attributes*, e a relação hierárquica entre *Superclasses* e *Subclasses*.

Na segunda mesclagem, realizada entre a ontologia mesclada 1-2 e a do aluno 3 6.11, foram encontradas 14 classes e 3 propriedades de objeto equivalentes. Estas classes e propriedades, que foram sobrepostas na ontologia mesclada, estão listadas na Tabela 6.11. As subclasses da classe *Inheritance* eram equivalentes em ambas as ontologias, enquanto a classe *Class* possuía apenas duas das três subclasses definidas na ontologia mesclada CiRe. A ontologia do aluno 3 contribuiu também com a hierarquia da classe *Methods*, que tem como subclasses *Constructor* e *Destructor*.

Durante a mesclagem das classes *Objects*, verificou-se que a propriedade utilizada para representar a composição desta classe com as partes *Methods* e *Attributes* era diferente nas duas ontologias: na ontologia mesclada CiRe, foi utilizada *are_composed_of*; na ontologia do aluno 3, a propriedade *are*. Sendo assim, a propriedade utilizada para representar a composição, neste caso, foi *are_composed_of*, em virtude da prioridade dada à ontologia mais completa. A propriedade *have*, entretanto, foi mantida na ontologia mesclada, como

Elementos Mesclados - Ontologia 1-2-3	
Classes	Propriedades
Attributes	can_be
Methods	hasPart
Object_orientation	isPartOf
Uml	
Class	
Abstract	
Concrete	
Inheritance	
Combination	
Extension	
Limitation	
Specialization	
Specification	
Objects	

Tabela 6.11: Classes e propriedades mescladas entre as ontologias 1-2 e 3

subpropriedade de *hasPart*.

Na mesclagem de propriedades, foi seguida a mesma convenção utilizada na tradução para a definição de domínios e contradomínios: nos casos em que ambos estavam especificados nas duas ontologias e as classes de domínio e contra-domínio não eram coincidentes, a propriedade na ontologia resultante foi definida sem a especificação de nenhuma das classes. Isto ocorreu durante a mesclagem da propriedade *can_be* que, na ontologia CiRe, possuía *Return_types* e *Data_types* como domínio e contra-domínio, respectivamente, e, na ontologia do aluno 3, as classes *Methods* e *Encapsulated*. Nota-se que tratam-se de relações entre elementos de diferentes classes que, caso mantidas na definição desta propriedade, poderiam especificar a existência de relações entre elementos de classes que, na representação original, não se relacionam de fato, como *Return_types* - *can_be* - *Encapsulated* ou *Methods* - *can_be* - *Data_types*.

Na terceira e última mesclagem, realizada entre a ontologia resultante CiReMi e a do aluno 4, mais 14 classes foram mescladas, além de 8 propriedades. As classes e propriedades equivalentes que foram sobrepostas nesta mesclagem estão listadas na Tabela

Elementos Mesclados - Ontologia Resultante	
Classes	Propriedades
Attributes	are
Characteristics	are_instances_of
Classes	represent
abstract	models
Encapsulation	allows
Inheritance	groups
Simple	isPartOf
Methods	hasPart
Constructor	
Destructor	
Object_orientation	
Polimorfism	
Real	
Reusability	

Tabela 6.12: Classes e propriedades mescladas entre as ontologias 1-2-3 e 4

6.12.

6.4.1 Ontologia Resultante

A ontologia resultante, que pode ser vista nas Figuras 6.14, 6.15 e 6.16, reúne todos os elementos contidos nas ontologias intermediárias e está consistente de acordo com a verificação de consistência realizada com o uso do raciocinador Racer [23]. Um raciocinador é um software que implementa um método de inferência para obter conclusões. Entretanto, a mesclagem de alguns conceitos resultou na mesclagem de taxonomias distintas. O primeiro caso onde isto ocorre é na mesclagem da classe *Inheritance* apresentada na Figura 6.14, onde as subclasses da classe *Inheritance* da ontologia do aluno 4 descrevem os possíveis tipos de herança a nível de classe, *Simple* ou *Multiple*, e a da ontologia Ci-ReMi, que descreve os tipos de herança com respeito a métodos (*Combination*, *Extension*, *Limitation*, *Specialization* e *Specification*).

A mesclagem da classe *Objects* também resultou na mistura de taxonomias diferentes:

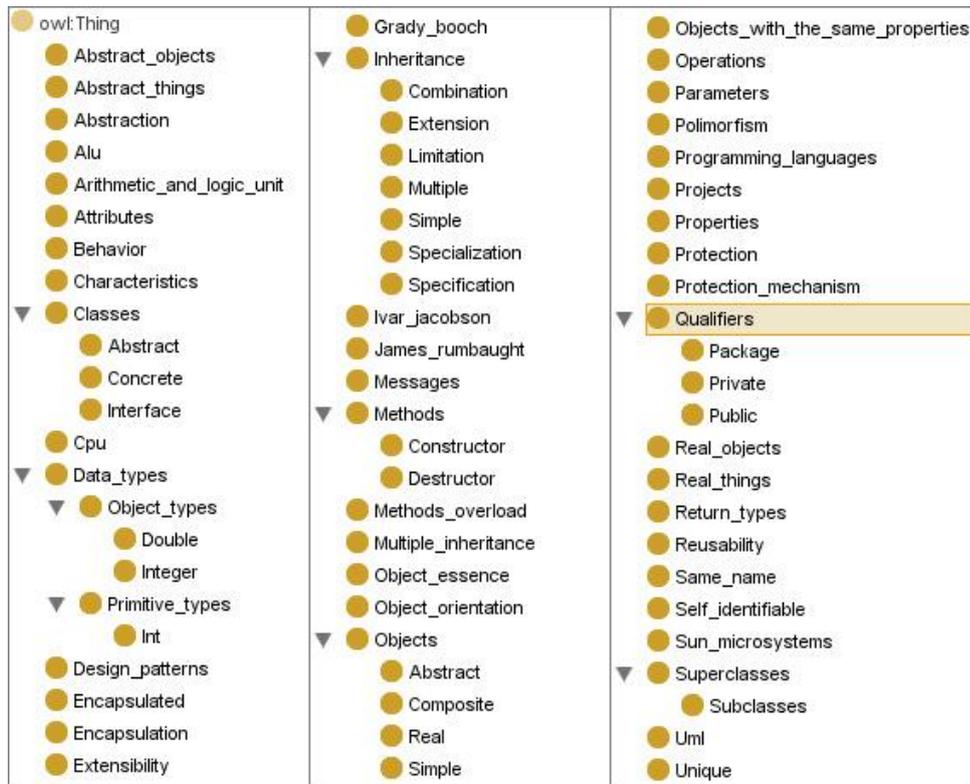


Figura 6.14: Hierarquia de classes da ontologia resultante

de objetos simples e compostos e de objetos reais e abstratos, conforme mostrado na Figura 6.16. Além disso, caracterizou-se uma redundância entre as taxonomias das classes *Objects*, com subclasses *Real* e *Abstract*, e *Classes*, com subclasses *Concrete* e *Abstract*. A composição em atributos e métodos também ficou duplicada nas classes *Objects* e *Classes*.

Ao obter-se a ontologia resultante, nota-se também a existência de classes e propriedades de objeto com nomes semelhantes, como *Protection* e *Protection_mechanism*, ou semanticamente equivalentes, como *Methods* e *Operations*, que poderiam ser mapeados em um único conceito ou propriedade.

6.4.2 Ontologia Alvo

Resultados como os comentados na seção são indícios da ausência de consenso, entre representações ou nomes adotados para representar conceitos, que não puderam ser unifi-

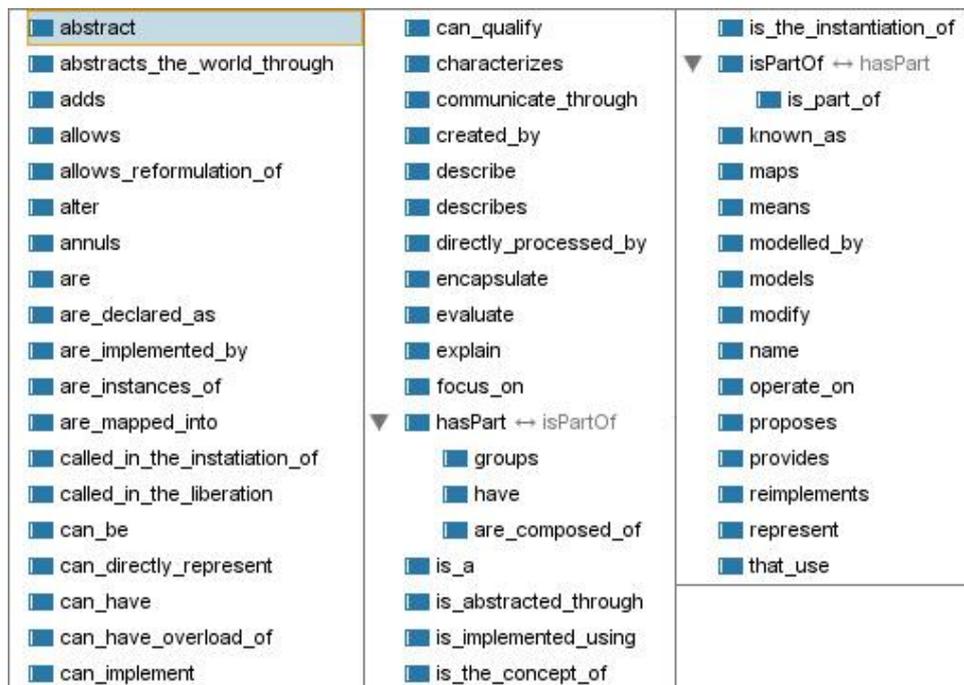


Figura 6.15: Propriedades de objeto da ontologia resultante

cados pela ferramenta de mesclagem utilizada. Para diminuir estes conflitos e obter uma ontologia sem as inconsistências identificadas, que melhor represente o domínio modelado, foi realizada uma nova etapa, manual, de refinamento da ontologia resultante, onde foi produzida uma nova versão da ontologia resultante, aqui chamada de ontologia alvo.

6.4.2.1 Mesclagem Adicional de Classes e Propriedades

Na criação da ontologia em alvo, que poderia ser obtida das ontologias intermediárias, primeiramente foi feita a mesclagem de classes e propriedades cuja equivalência não foi identificada pela a ferramenta de mesclagem adotada. Esta seção reúne classes e propriedades com nomes ou significados semelhantes que foram considerados durante a mesclagem manual.

*Classes **Protection** e **Protection_mechanism***

As classes *Protection* e *Protection_mechanism*, são contradomínio de duas propriedades, ambas com a classe *Encapsulation* no contra-domínio:

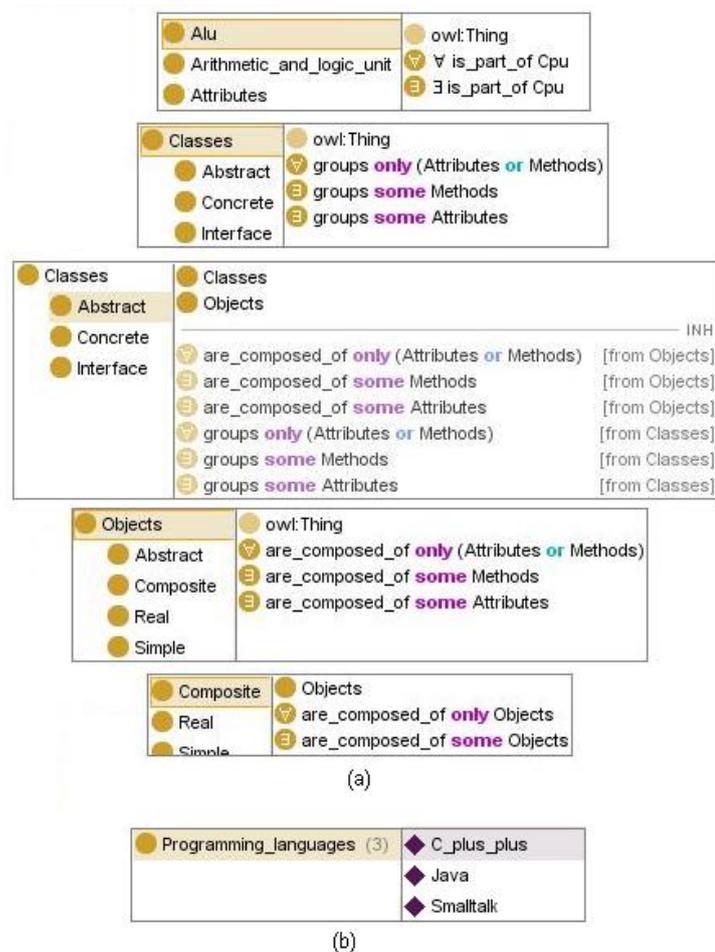


Figura 6.16: Restrições (a) e instâncias (b) da ontologia resultante

is_a , *provides*, *resultado das relações Encapsulation - is a - protection mechanism Encapsulation - protection* *merge estas classes. Por representar um conceito mais específico, Protection_mechanism foi man-*

Classes *Methods* e *Operations*

Conceitualmente, um método é uma implementação de uma operação em uma classe que representa um tipo abstrato de dados. No contexto apresentado, o conceito *Methods* é mais adequado do que o conceito *Operations*, visto que o domínio modelado trata de classes, e não de tipos abstratos de dados. Sendo assim, a classe *Operations* foi removida; *Methods* foi colocada como contradomínio das propriedades *describe*, *can_qualify* e *allows_reformulation* e como domínio da propriedade *operate_on*, em substituição à classe removida.

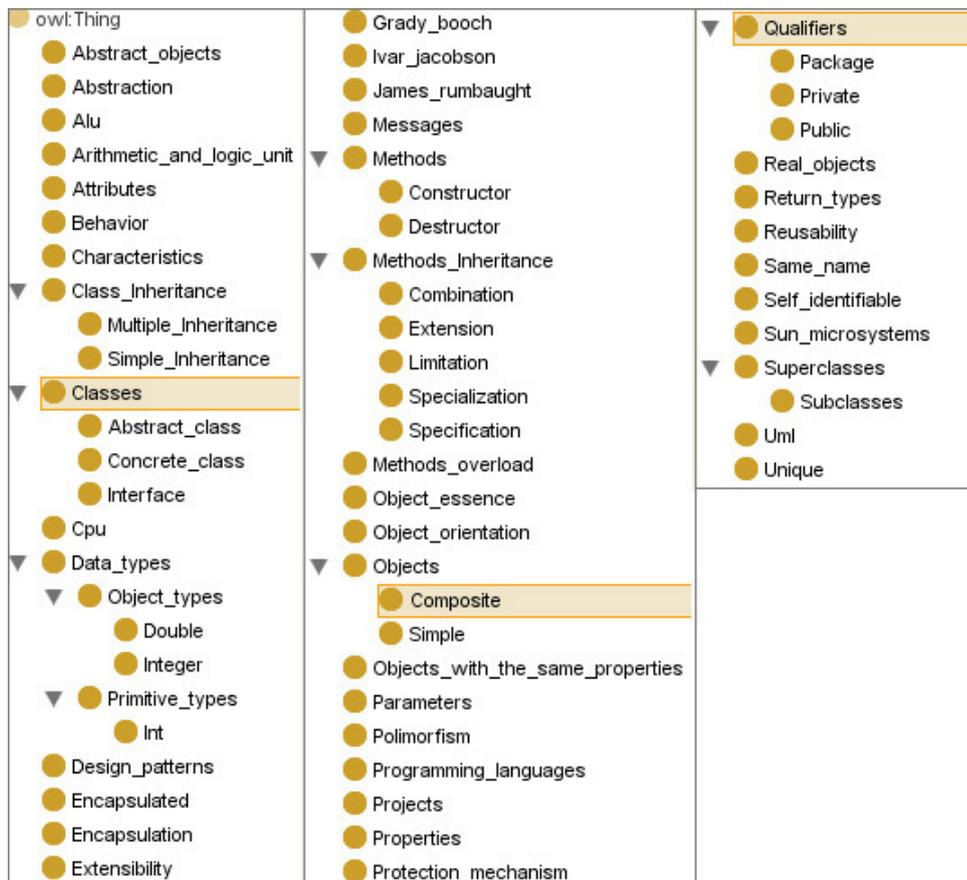


Figura 6.17: Hierarquia de classes da ontologia alvo

Classes *Abstract* e *Abstract_objects*

Embora os conceitos tenham nome semelhantes, os mesmos possuem significados distintos na ontologia. O primeiro deles, *Abstract*, representa um tipo especial de classe, que é implementada por uma (ou mais) classes concretas. Isto pode ser visto pela taxonomia do conceito *Classes* na Figura 6.14 e pelas propriedades *Abstract/Interface* - *are_implemented_by* - *Concrete*, *Objects* - *represent* - *Real_objects/Abstract_objects* e *Real_objects* - *are_mapped_into* - *Abstract_objects*.

Ainda assim, para facilitar a diferenciação destas classes, a primeira delas (*Abstract*) teve seu nome modificado para *Abstract_class* e a classe *Concrete*, para seguir o padrão de nomenclatura e por se referir a classes, foi também modificada para *Concrete_class*.

Classes *Real_objects* e *Real_things*

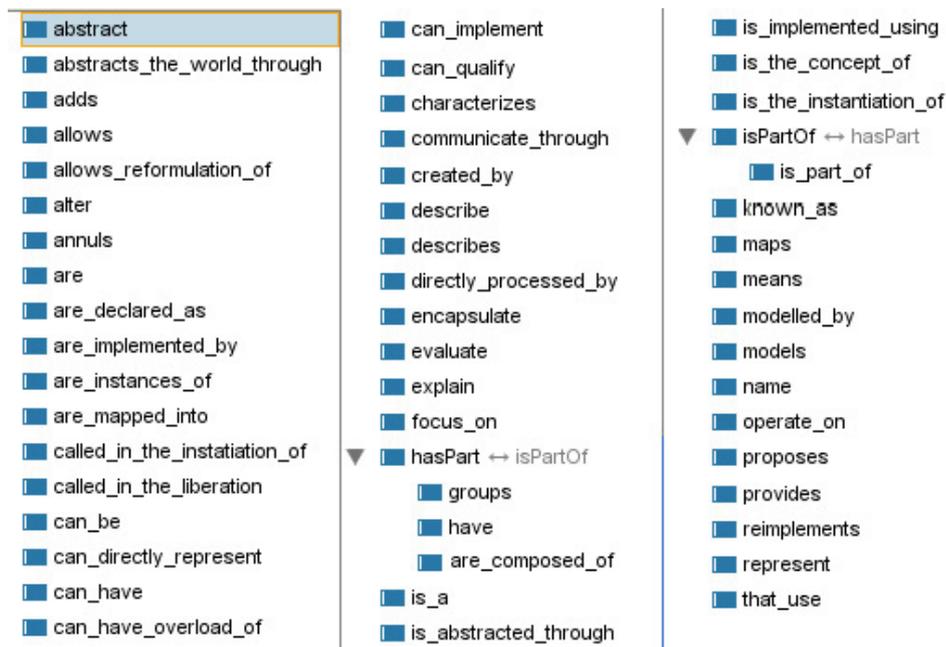


Figura 6.18: Propriedades de objeto da ontologia alvo

Neste caso, mais uma vez, foram analisadas as propriedades em que ambos os conceitos participam como domínio ou contra-domínio. *Real_objects* aparece nas propriedades *abstraction - maps - real_objects* e *Real_objects are_mapped_into Abstract_objects*, enquanto *Real_things* aparece em *Objects - represent - Real_things/Abstract_things*. A relação de indivíduos destas classes com indivíduos da classe *Objects* sugere que trata-se do mesmo conceito. Por ser mais representativa, a classe *Real_objects* foi mantida e colocada como contradomínio da propriedade *represent*, enquanto a classe *Real_things* foi removida.

Classes *Abstract_objects* e *Abstract_things*

Similarmente ao caso anterior, a classe *Abstract_objects* foi mantida e colocada como contradomínio da propriedade *represent*, enquanto a classe *Abstract_things* foi removida.

Classes *Real* e *Real_objects*

A classe *Real* aparece como subclasse de *Objects*, o que sugere que a mesma tem o mesmo significado que *Real_objects*. Por ter um nome mais significativo, a classe *Real_objects* foi mantida, sendo colocada como subclasse de *Objects*, e a classe *Real* foi removida.

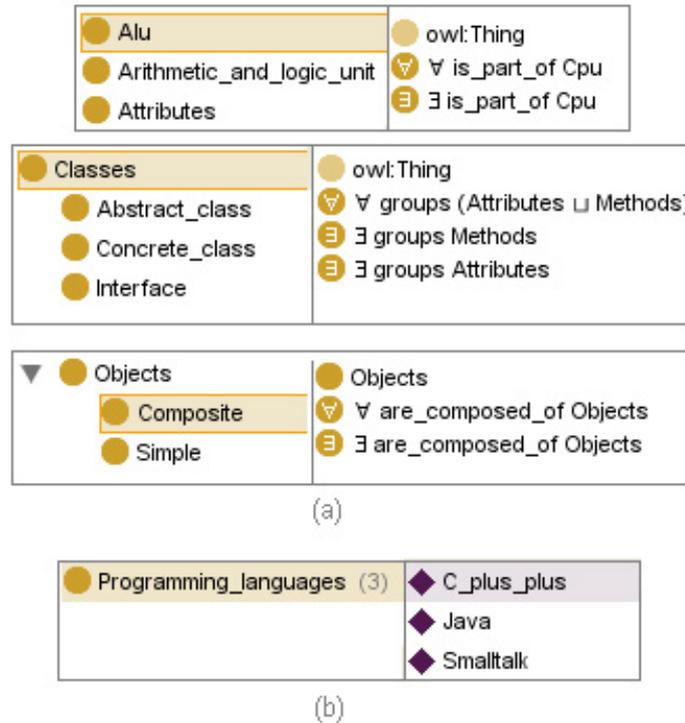


Figura 6.19: Restrições (a) e instâncias (b) da ontologia alvo

Propriedades *describe* e *describes*

As propriedades *describe* e *describes* possuem domínios e contra-domínios distintos: *Classes - describe - Attributes/Methods* e *Encapsulation - describes - Qualifiers*. Por este motivo, considerou-se serem propriedades diferentes.

Propriedades *modelled_by* e *models*

Aqui também as propriedades possuem domínio e contra-domínio distintos, tendo sido mantidas.

Classes/Objects - modelled_by - UML

Object_orientation - models - Objects

Propriedades *alter* e *modify*

As propriedades *alter* e *modify*, além de possuir o mesmo significado, possuem as mesmas classes como domínio e contra-domínio: *Methods* e *Attributes*, respectivamente. Por este motivo, a propriedade *modify* foi removida.

6.4.2.2 Ajuste de Relações Hierárquicas e de Composição

Após a mesclagem de classes e propriedades, as hierarquias da ontologia resultante foram avaliadas e ajustadas na ontologia alvo de forma a tratar situações como a mistura de taxonomias diferentes na definição de um conceito e duplicação da mesma taxonomia, ou composição, em conceitos diferentes.

Hierarquia da classe *Inheritance*

Como visto na Figura 6.14, a classe *Inheritance*, ao ser mesclada, acumulou como subclasses duas taxonomias diferentes: as relativas ao tipo de herança de classe, *Simple* e *Multiple*, e as relativas ao tipo de herança de métodos, *Combination*, *Extension*, *Limitation*, *Specialization* e *Specification*. Para separar estas duas taxonomias, a classe *Inheritance* foi renomeada para *Methods_inheritance*, e foi criada uma nova classe, *Class_inheritance*, para onde foram movidas as subclasses *Simple* e *Multiple*, também renomeadas para *Simple_inheritance* e *Multiple_inheritance*, como pode ser visto na Figura 6.17.

Hierarquia da classe *Objects*

A classe *Objects* também acumulou duas hierarquias distintas após a mesclagem de ontologias intermediárias: *Simple* e *Compound*, contra *Real* e *Abstract*. Estas duas últimas também são subclasses do conceito *Classes*, o que indica uma ausência de consenso sobre os significados de *Classes* e *Objects*. Decidiu-se por manter *Simple* e *Compound* como subclasses de *Objects* e *Real* e *Abstract* como tipos de *Classes*, mais uma vez priorizando o que foi definido na ontologia intermediária mais completa.

Composição em *Attributes* e *Methods*

A composição com as classes *Attributes* e *Methods* aparece na definição tanto da classe *Objects* como na da classe *Classes*. Mais uma vez, seguindo a ordem de ontologia mais complexa, foi adotada esta composição na classe *Objects*, sendo removida da definição de *Classes*.

6.5 Avaliação

A avaliação de ontologias tem por objetivo realizar um julgamento do conteúdo da mesma, analisando os termos definidos, os não definidos e os definidos incorretamente. Esta análise é realizada nas classes, restrições, propriedades e indivíduos da ontologia, sob dois aspectos: verificação, feita para avaliar se ontologia foi construída corretamente de acordo com os requisitos e questões de competência, e validação, que julga se a ontologia realmente modela o domínio para o qual foi criada [21]. Para efeito deste trabalho, o conhecimento do domínio abordado considerado se limita às informações contidas nos mapas, não sendo avaliado o conteúdo do mapa em relação a representações externas do mesmo domínio.

A avaliação da ontologia resultante foi realizada de acordo com os critérios identificados em [21]: consistência, completude e concisão.

6.5.1 Consistência

O aspecto de verificação de consistência de ontologias tem como objetivo verificar se não existem contradições entre definições existentes, resultando na inferência de informações contraditórias.

A consistência das ontologias individuais e da ontologia resultante foi avaliada com o uso do raciocinador RACER [23]. Os arquivos OWL gerados foram abertos e analisados no protege para verificar sua corretude e completude de acordo com as regras heurísticas estabelecidas. De acordo com a verificação realizada pelo raciocinador, todas as ontologias estão consistentes. Foram ainda identificadas algumas inconsistências semânticas na representação de hierarquias de classes, já mencionadas nas seções 6.4.2 e 6.4.2.

No estudo de caso realizado, o processo de transcrição apresentou bons resultados, sendo capaz de identificar relações especiais entre conceitos nos mapas conceituais que serviram de base para compor a hierarquia da ontologia alvo. Além disso, foram identi-

ficadas classes e propriedades equivalentes na ontologia resultante que, após a resolução manual de conflitos, resultaram em uma ontologia representativa do domínio abordado segundo o entendimento do grupo que elaborou os mapas conceituais.

6.5.2 Concisão

A concisão de uma ontologia pode ser avaliada medindo-se o quanto definições desnecessárias ou redundâncias entre especificações de termos [21] são identificadas.

Na ontologia resultante, foram localizadas redundâncias de classes, propriedades e restrições, resultado de termos que não foram identificados como equivalentes durante a mesclagem das ontologias intermediárias, relatadas em maiores detalhes nas seções 6.4.2 e 6.4.2. Tal equivalência foi identificada ao se comparar nomes de classes e propriedades, restrições e significado de propriedades.

6.5.3 Completude

A verificação de completude de uma ontologia tem o propósito de verificar se todos os conceitos e definições relativos ao domínio foram especificados. Para avaliar a ontologia resultante sob este critério, considerou-se que a completude da ontologia seria avaliada medindo-se a quantidade de conceitos e relações modelados nos mapas conceituais que estão de alguma forma representados na ontologia resultante.

Todos os conceitos e relações modelados nos mapas conceituais estão representados em classes, propriedades, restrições e indivíduos na ontologia resultante, mostrando que a mesma é completa segundo o critério de completude adotado.

6.6 Requisitos para Ferramenta de Mesclagem

Como resultado da avaliação da ontologia resultante, foram identificados requisitos de uma ferramenta de mesclagem para ontologias obtidas a partir de mapas conceituais. Tal

ferramenta teria como requisitos:

- Permitir seleção de um conjunto de ontologias para mesclagem.
- Verificar a equivalência de classes e propriedades utilizando-se também de métodos terminológicos extrínsecos, identificando assim termos sinônimos.
- Verificar a equivalência de classes e propriedades utilizando-se também de métodos estruturais, de forma a comparar as restrições, as propriedades das quais participa e as hierarquias às quais pertence uma classe.
- Possuir uma política de mesclagem de classes que, ao identificar subclasses na classe mesclada, decida, de acordo com as subclasses coincidentes, se trata-se da mesma classe ou se são classes diferentes (uso de métodos estruturais externos).
- Ao mesclar classes, mesclar as restrições contidas na mesma, de forma que, se as restrições forem com as mesmas classes em ambas as ontologias, manter propriedade da ontologia preferida; se forem com classes diferentes, adicionar restrições.
- Ao mesclar indivíduos de classes, sobrepor repetidos e adicionar os demais
- Ao mesclar propriedades, se uma das propriedades não possuir nem domínio nem contra-domínio, não especificar estes atributos na propriedade mesclada. Caso contrário, os domínios e contra-domínios devem ser unidos, respeitando a regra de transcrição, onde, caso a propriedade possua mais de um domínio e mais de um contra-domínio, os mesmos não devem ser especificados.

Capítulo 7

Discussão e Conclusões

Ontologias de domínio são uma peça fundamental em sistemas inteligentes baseados em conhecimento, possibilitando a inferência de informações não explicitamente especificadas e a interoperabilidade entre sistemas.

No contexto educacional, a geração de uma ontologia a partir de mapas conceituais produzidos por alunos torna-se interessante quando aplicada com objetivos como:

- Verificação de aprendizagem de um grupo sobre um dado domínio;
- Verificação da evolução da aprendizagem de um grupo, em instantes de tempo distintos (comparação entre ontologias geradas a partir de mapas conceituais, em diferentes momentos);
- Comparação entre ontologias geradas de mapas conceituais, de grupos diferentes, porém sobre o mesmo domínio.

Alguns trabalhos têm sido desenvolvidos utilizando mapas conceituais como parte do processo de construção de ontologias. Tais trabalhos, entretanto, tendem a limitar a forma como mapas conceituais são construídos, sugerindo frases de ligação pré-determinadas.

Neste trabalho, foi apresentado um processo de geração de ontologias de domínio a partir de mapas conceituais, onde os mesmos são primeiramente traduzidos individual-

mente em uma linguagem de representação de ontologias, para que possam ser em seguida mesclados em uma única ontologia de domínio.

A tradução de cada mapa conceitual em uma ontologia intermediária é realizada através da utilização de heurísticas baseadas principalmente nos nomes de conceitos e frases de ligação, com suporte banco de dados léxico WordNet na identificação de relações semânticas entre os conceitos.

Foi desenvolvido um artefato de software para realizar a primeira etapa do processo, de tradução de mapas conceituais em ontologias individuais, utilizando as heurísticas especificadas. A segunda etapa foi realizada mediante a utilização de uma ferramenta de mesclagem existente, que foi selecionada segundo condições impostas por ontologias obtidas de mapas conceituais.

Para avaliar o processo proposto, foi realizado uma atividade de produção de mapas conceituais. Tal experimento teve por objetivo obter um conjunto de mapas conceituais descrevendo um mesmo domínio de conhecimento, de forma que pudessem ser traduzidos em ontologias individuais e estas, por sua vez, posteriormente mescladas.

Nas seções seguintes, são discutidos os pontos relevantes sobre cada uma das etapas envolvidas no desenvolvimento deste trabalho. Em seguida, são apresentadas sugestões para trabalhos futuros.

7.1 Heurísticas

A tradução de cada um dos mapas utilizou-se de heurísticas baseadas principalmente na terminologia adotada na nomenclatura de frases de ligação.

Inicialmente, a especificação das heurísticas foi implementada de forma procedimental no código do módulo Tradutor. Entretanto, conforme novas heurísticas foram sendo incorporadas ao módulo, o código do mesmo foi se tornando muito complexo, aumentando assim a dificuldade de manutenção e teste da aplicação. A especificação das heurísticas

foi então migrada para Prolog, um recurso mais adequado à implementação de regras heurísticas, que facilitou as atividades de manutenção e testes ao isolar a parte lógica e a realização de inferências da parte funcional do componente.

Os conjuntos de frases de ligação, utilizados nas heurísticas de identificação de relações hierárquicas, de composição e de instâncias, podem variar drasticamente de acordo com o vocabulário do domínio representado, e tende a crescer conforme a ferramenta for sendo utilizada em diferentes aplicações. Fornecer um ambiente de manutenção de forma a possibilitar a atualização destes conjuntos com segurança, pois atualmente se encontram especificados no mesmo arquivo que as heurísticas.

7.2 Tradução e Tradutor

Na tradução de mapas conceituais em ontologias intermediárias, a grande maioria (mais de 90%) das sugestões de hierarquias e relações de composição foram aceitas. Isto demonstra duas coisas: primeiro, que uma grande quantidade de relações especiais poderia ter sido mapeada de maneira inadequada caso não fosse oferecido ao usuário a opção de decidir sobre hierarquias e composições parcialmente identificadas. Por outro lado, confirma que a heurística de frases de ligação, por si só, não é suficiente para identificar estes dois tipos de relações, visto que houveram casos em que as sugestões não foram aceitas.

Na seleção de instâncias ou subclasses durante a tradução, também não houve unanimidade: em alguns casos, os conceitos foram identificados como instâncias de classes; em outros, como relações entre classes e subclasses. Isto evidencia, novamente, que as frases de ligação utilizadas podem caracterizar relações de ambos os tipos.

A utilização da linguagem OWL facilitou o trabalho de geração de hierarquias de classes, pois as definições de cada classe são independentes entre si, por não precisarem estar encadeadas umas às outras quando participam de hierarquias.

7.3 Mesclagem

Na ontologia resultante, obtida após a mesclagem de ontologias intermediárias, foram encontrados casos de inconsistência e duplicidade na definição de classes e propriedades, como visto nas seções 6.4.2 e 6.4.2. Estes conflitos foram resolvidos após a realização de uma etapa manual de mesclagem de propriedades e classes, além do ajuste de hierarquias e restrições. Isto mostra que a ontologia resultante é uma versão preliminar do domínio modelado, e não uma versão definitiva. Tal ontologia pode ainda ser aproveitada para facilitar negociação de conceitos em atividades de negociação de ontologias, pois reúne, em uma única representação, uma transcrição formalizada da estrutura cognitiva de um grupo de indivíduos.

7.4 Trabalhos Futuros

O processo de transcrição aqui apresentado possui espaço para que novos trabalhos sejam desenvolvidos, dentre os quais destacam-se:

- a avaliação de mais ferramentas de mesclagem disponíveis, que utilizem métodos diferenciados na identificação de conceitos equivalentes, com o intuito de identificar métodos de mesclagem adicionais que possam ser implementados em uma ferramenta de mesclagem de ontologias obtidas a partir de mapas conceituais;
- o desenvolvimento de uma ferramenta de mesclagem que atenda aos requisitos identificados na Seção 6.6 que seja mais adequada à mesclagem de ontologias obtidas a partir de mapas conceituais;
- a revisão e complementação das heurísticas de tradução, de modo a mapear características diferenciadas das relações nos mapas conceituais ainda não mapeadas em elementos específicos em linguagem OWL;

-
- a realização de um novo experimento de transcrição utilizando mapas obtidos de especialistas de domínio, com o objetivo de comparar os resultados obtidos da mesclagem de uma ontologia resultante do conhecimento de especialistas, com a ontologia obtida de mapas conceituais de alunos em um ambiente de ensino-aprendizagem;
 - a utilização de bancos de dados léxicos mais específicos sobre o domínio representado, uma vez que, neste experimento especificamente, nenhuma relação hierárquica ou de composição dos mapas foi encontrada na base do WordNet.

Referências Bibliográficas

- [1] M. S. Amoretti. Conceptual map: analysis of prototypes and categorization levels. In *Proceedings of the CCM Digital Government Symposium*, The University of Alabama, USA, 2003.
- [2] D. P. Ausubel, J. D. Novak, and H. Hanesian. *Educational Psychology: a cognitive view*. Interamericana, Rio de Janeiro, 1980.
- [3] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. McGuinness, P. Patel-Schneider, and A. Stein. Owl web ontology language reference, 2004.
- [4] Joost Breuker. Constructing a legal core ontology: Lri-core. In *Proceeding of the Workshop on Ontologies and their Applications*, September 2004.
- [5] Virgínia Brillhante, Gretchen Macedo, and Sabatta Macedo. Heuristic transformation of well-constructed conceptual maps into owl preliminary domain ontologies. In *Proceedings of the Workshop on Ontologies and Their Applications (WONTO 2006) at the 18th Brazilian Symposium on Artificial Intelligence*, Ribeirão Preto, Brazil, 2006.
- [6] A. J. Cañas, J. D. Novak, and F. M. González. Cmaptools: A knowledge modeling and sharing environment. In *Proceedings of the First International Conference on Concept Mapping*, pages 125–133, Pamplona, Spain, 2004.

-
- [7] A. J. Cañas. Online concept maps: Enhancing collaborative learning by using technology with concept maps. *The Science Teacher*, 68(4):49–51, 2001.
- [8] A. J. Cañas and M. Carvalho. Concept maps and AI: an unlikely marriage? In *Anais do XV Simpósio Brasileiro de Informática na Educação*, pages 1–10, Manaus, Brazil, 2004.
- [9] M. Cristani and R. Cuel. A survey on ontology creation methodologies. *International Journal on Semantic Web Information Systems*, 1(2):49–69, April-June 2005.
- [10] Jos de Bruijn and Axel Polleres. Towards an ontology mapping specification language for the semantic web. Technical report, DERI Technical Report 2004-06-30, 6 2004. <http://www.deri.at/fileadmin/documents/DERI-TR-2004-06-30.pdf>.
- [11] Dejing Dou, Drew McDermott, and Peishen Qi. Ontology translation by ontology merging and automated reasoning. In *Proceedings of the EKAW2002 Workshop on Ontologies for Multi-Agent Systems*, pages 3–18, 2002.
- [12] J. Euzenat et al. State of the art on ontology alignment. knowledge web deliverable d2.2.3. Technical report, INRIA, 2004. <http://www.starlab.vub.ac.be/research/projects/knowledgeweb/kweb-223.pdf>.
- [13] M. Fernández, A. Gómez-Pérez, and N. Juristo. Methontology: From ontological art towards ontological engineering. In *Proceedings of the AAAI Spring Symposium on Ontological Engineering*, The University of Stanford, California, 1997.
- [14] European Research Consortium for Informatics and Mathematics. Jasper - sicstus prolog. <http://www.sics.se/sicstus/docs/3.12.7/html/sicstus/Jasper.html>. Acessado em maio de 2007.
- [15] Martin Fowler and Kendall SCOTT. *UML Essencial: um breve guia para a linguagem padrão de modelagem de objetos*. Bookman, Porto Alegre, 2000.

-
- [16] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley, Reading, Massachusetts, 1994.
- [17] A. Garcia, S. A. Sansone, P. Rocca-Serra, C. Taylor, and M. A. Ragan. The use of conceptual maps for two ontology developments: nutrigenomics, and a management system for genealogies. In *Proceedings of the 8th International Protégé Conference*, Madrid, Spain, 2005.
- [18] J. Geurts, S. Bocconi, J. van Ossenbruggen, and L. Hardman. Towards ontology-driven discourse: From semantic graphs to multimedia presentations. In *Proceedings of the Second International Semantic Web Conference (ISWC 2003)*, 2003.
- [19] H. Gómez-Gauchía, B. Díaz-Agudo, and P. González-Calero. Two-layered approach to knowledge representation using conceptual maps and description logics. In *Proceedings of the First International Conference on Concept Mapping*, Pamplona, Spain, 2004.
- [20] A. Gómez-Pérez, M. Fernández-López, and O. Corcho. *Ontological Engineering*. Springer-Verlag, London, 2004.
- [21] T. R. Gruber. A translation approach to portable ontology specification. *Knowledge Acquisition*, 5(2):199–220, 1993.
- [22] V. Haarslev and R. Möller. *RACER User s Guide and Reference Manual Version 1.7.7*. Concordia University and Univ. of Appl. Sciences in Wedel, 2004.
- [23] Adil Hameed, Alun Preece, and Derek Sleeman. Ontology reconciliation. In *Handbook on Ontologies in Information Systems*, pages 231–250. Springer Verlag, October 2003.

- [24] Pat Hayes, Thomas C. Eskridge, Mala Mehrotra, Dmitri Bobrovnikoff, Thomas Reichherzer, and Raul Saavedra. COE: Tools for collaborative ontology development and reuse. In *Proceedings of the Knowledge Capture Conference (K-CAP)*, 2005.
- [25] F. Hayes-Roth, D. Waterman, and D. Lenat. *Building Expert Systems*. Addison-Wesley, 1983.
- [26] I. Horrocks. FaCT and iFaCT. In *Proceedings of the International Workshop on Description Logics (DL'99)*, pages 133–135, 1999.
- [27] Jason Hunter and Brett McLaughlin. JDOM XML API, 2004. www.jdom.org/.
- [28] Inspiration Software Inc. Inspiration software - ferramenta de apoio à construção de mapas conceituais. <http://www.inspiration.com/>. Acessado em maio de 2007.
- [29] M. Kavouras. A unified ontological framework for semantic integration. In *Proceeding of the International Workshop on Next Generation Geospatial Information*, pages 19–21, Cambridge (Boston) USA, 2003.
- [30] A. Malucelli and E. Oliveira. Towards similarity identification to help in agents negotiation. In *Proceedings of the XVII Brazilian Artificial Intelligence Symposium*, São Luis, MA, Brasil, 2004.
- [31] F. Manola and E. Miller. Rdf primer. w3c recommendation, 2004. <http://www.w3.org/TR/rdf-primer/>.
- [32] D.L. McGuinness, R. Fikes, J. Rice, and S. Wilder. An environment for merging and testing large ontologies. In *Proceeding of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000)*, pages 483–493, Breckenridge, Colorado, April 2000.
- [33] G. A. Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

- [34] MindJet. Mind man - ferramenta de apoio à construção de mapas conceituais. <http://mindman.com/>. Acessado em maio de 2007.
- [35] J. D. Novak and D. B. Gowin. *Learning How to Learn*. Cambridge University Press, 1984.
- [36] N. F. Noy. Semantic integration: a survey of ontology-based approaches. *SIGMOD Rec.*, 33:65–70, December 2004.
- [37] N. F. Noy, R. W. Fergerson, and M. A. Musen. The knowledge model of protégé-2000: combining interoperability and flexibility. In *Proceedings of the 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW04)*, pages 17–32, Berlin, Germany, 2004. Springer-Verlag.
- [38] Natalya F. Noy and Deborah L. McGuinness. Ontology development 101: A guide to creating your first ontology, 2001. http://protege.stanford.edu/publications/ontology_development/ontology101_noy-mcguinness.html.
- [39] Natalya Fridman Noy and Mark A. Musen. PROMPT: Algorithm and tool for automated ontology merging and alignment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 450–455, 2000.
- [40] H. Sofia Pinto, A. Gomez-Perez, and J. P. Martins. Some issues on ontology integration. In *Proceedings of IJCAI99s Workshop on Ontologies and Problem Solving Methods: Lessons Learned and Future Trends*, 1999.
- [41] Protégé. <http://protege.stanford.edu/>. Last accessed on 20 Mar. 2006.

-
- [42] F. E. L. Rocha, J. V. Costa, and E. L. Favero. A new approach to meaningful learning assessment using concept maps: Ontologies and genetic algorithms. In *Proceedings of the First International Conference on Concept Mapping*, Pamplona, Spain, 2004.
- [43] Stuart Russel and Peter Norvig. *Inteligência Artificial*. Campus, São Paulo, 2004.
- [44] Stefan Schulz and Udo Hahn. Ontological foundations for biomedical sciences. In *Proceeding of the Workshop on Ontologies and their Applications*, September 2004.
- [45] Luís A. Lima Silva, Laura S. Mastella, Mara Abel, Renata M. Galante, and Luiz F. De Ros. An ontology-based approach for visual knowledge: Image annotation and interpretation. In *Proceeding of the Workshop on Ontologies and their Applications*, September 2004.
- [46] Banxia Software. Decision explorer - ferramenta de apoio à construção de mapas conceituais. <http://www.scotnet.co.uk/banxia/demain.html>. Acessado em maio de 2007.
- [47] M. Uschold and M. Gruninger. Ontologies: Principles, methods and applications. *The Knowledge Engineering Review*, 11(2):93–136, 1996.