

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

GABRIEL MOSER TORRES

**CONSTRUÇÃO COLABORATIVA DE
ONTOLOGIAS PARA DOMÍNIOS VISUAIS
UTILIZANDO FUNDAMENTAÇÃO
ONTOLÓGICA**

Dissertação apresentada como requisito parcial
para a obtenção do grau de Mestre em Ciência
da Computação

Prof. Dr^a. Mara Abel
Orientadora

Porto Alegre, Junho de 2012.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

CIP - Catalogação na Publicação

Torres, Gabriel Moser
CONSTRUÇÃO COLABORATIVA DE ONTOLOGIAS PARA
DOMÍNIOS VISUAIS UTILIZANDO FUNDAMENTAÇÃO ONTOLÓGICA
/ Gabriel Moser Torres. -- 2012.
119 f.

Orientadora: Mara Abel.

Dissertação (Mestrado) -- Universidade Federal do
Rio Grande do Sul, Instituto de Informática,
Programa de Pós-Graduação em Computação, Porto Alegre,
BR-RS, 2012.

1. Engenharia de Conhecimento. 2. Ontologias. 3.
Colaboração. 4. Groupware. I. Abel, Mara, orient. II.
Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da UFRGS com os
dados fornecidos pelo(a) autor(a).

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Pós-Graduação: Prof. Aldo Bolten Lucion

Diretor do Instituto de Informática: Prof. Luis da Cunha Lamb

Coordenador do PPGC: Prof. Álvaro Freitas Moreira

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina B. Haro

AGRADECIMENTOS

Primeiramente gostaria de agradecer à Deus por me dar a força e motivação necessárias para concluir este projeto.

Sou grato de coração à minha namorada, Katiucia Capiotti, pelo seu incentivo para que eu fizesse o Mestrado, e por todo amor, confiança, apoio e paciência durante esta empreitada. Sem este carinho, certamente o caminho teria sido diferente.

Agradeço à minha mãe, Lilian Moser, por todas as boas palavras e por toda educação que recebi, me transmitindo a tranquilidade necessária para passar por cada ciclo sem preocupações e com a certeza de seu apoio e amor incondicionais. Ao meu pai, Nelson Torres, e minha irmã, Bruna Moser Torres, por toda confiança que sempre depositaram em mim e nas minhas decisões.

Gostaria de agradecer com muita alegria à minha orientadora, Mara Abel, que me cativou desde o início com sua competência e profissionalismo. Obrigado pelos elogios e pelos puxões de orelha, ambos são cruciais para o crescimento. Com certeza é uma grande inspiração para todos que trabalham contigo. Agradeço ao professor Rafael Port da Rocha, por sua competência, dinamicidade e pelas suas boas idéias e visões inovadoras, tão necessárias para este trabalho.

Aos bolsistas Vítor Fortes Rey, Guilherme Schievelbein por participaram com dedicação e qualidade na construção deste projeto, e por estarem sempre dispostos a aprender. Aos bolsistas Luan Fonseca e Júlia Weber, por também estarem nos auxiliando na continuidade deste trabalho.

Aos colegas do Grupo de Banco de Dados Inteligentes, Sandro Fiorini, Joel Carbonera e Alexandre Lorenzatti, muito obrigado por todo conhecimento que me transmitiram com toda boa vontade, pelas boas conversas filosóficas e pelo apoio que trocamos nestes meses de convivência. Agradeço também ao pessoal da área de Geologia, Ariane Bernardes e Eduardo Espindola pela sua dedicação neste projeto.

Aos queridos colegas e amigos que me acompanharam nesta jornada, interessando-se pelo meu trabalho e auxiliando-me nas mais diversas formas. Sem sombra de dúvida são parte crucial desta vitória.

Agradeço especialmente aos meus amigos e colegas de trabalho, Roberto Heller, Tagline Treichel, e toda a nossa equipe, por toda paciência, tolerância e incentivo que me foi dado neste tempo em que estive dedicado ao Mestrado.

Por fim, agradeço à UFRGS que tanto nos orgulha e nos possibilita crescer como pessoas e profissionais, sempre estimulando nossa inteligência e nossa força de vontade, nos desafiando a vencer nossos limites.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	6
LISTA DE FIGURAS	7
LISTA DE TABELAS	9
RESUMO	10
COLABORATIVE ONTOLOGY CONSTRUCTION FOR VISUAL DOMAINS BASED ON FOUNDATIONAL ONTOLOGY	11
ABSTRACT	11
1 INTRODUÇÃO	12
2 MODELAGEM CONCEITUAL E ONTOLOGIAS	17
2.1 Metapropriedades Ontológicas	19
2.2 Ontologias de Fundamentação	21
2.2.1 Isomorfismo de Modelos Conceituais.....	21
2.2.2 Unified Foundational Ontology.....	22
2.3 Conhecimento Visual	26
3 ENGENHARIA DE ONTOLOGIAS	29
3.1.1 Metodologias para Construção de Ontologias.....	30
Guia 101.....	30
CommonKADS.....	30
On-to-Knowledge.....	31
TOVE.....	32
METHONTOLOGY.....	33
OntoKEM.....	34
3.2 Desenvolvimento Colaborativo de Ontologias	34
3.2.1 Modelo 3C: Comunicação, Coordenação e Cooperação.....	36
3.2.2 Ferramentas de Desenvolvimento Colaborativo de Ontologias.....	37
3.2.3 Avaliação das Ferramentas para Construção Colaborativa de Ontologias.....	40
4 MODELAGEM DA FERRAMENTA PARA CONSTRUÇÃO COLABORATIVA DE ONTOLOGIAS	44
4.1 Estrutura de Metadados	45
4.2 Modelagem da Colaboração com Metaontologias	45
4.2.1 Ontologia de Representação.....	48
4.2.2 Ontologia de Colaboração.....	55
4.3 Formalização do Ambiente de Colaboração	64

5	DESCRIÇÃO DO PORTAL OBAITÁ	67
5.1	Arquitetura	68
5.2	Persistência de Dados	70
5.2.1	Uso de Tripas	71
5.3	Interação com a Linguagem OWL	73
5.4	Funcionalidades do Sistema	74
5.4.1	Detalhes do Conceito	74
6	VALIDAÇÃO	81
6.1	Ontologia de Domínio do Estudo de Caso	81
6.2	Avaliação	82
6.2.1	Uso da Ferramenta pelos Usuários de Teste	83
6.2.2	Comparação da Ferramenta	84
6.2.3	Construção do Sistema de Validação	85
7	CONCLUSÃO	92
	REFERÊNCIAS	96
	ANEXO EXEMPLO DE CONSULTAS SQL COM TRIPLAS	100
	ANEXO META-ONTOLOGIA DE REPRESENTAÇÃO E COLABORAÇÃO EM OWL	102
	ANEXO AMOSTRA DO HISTÓRICO DA COLABORAÇÃO	115

LISTA DE ABREVIATURAS E SIGLAS

IBIS	<i>Issue-based Information System</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
MVC	<i>Model View Controller</i>
OWL	<i>Ontology Web Language</i>
RDF	<i>Resource Description Framework</i>
SPARQL	<i>Protocol and RDF Query Language</i>
SQL	<i>Structured Query Language</i>
UFO	<i>Unified Foundational Ontology</i>
W3C	<i>World Wide Web Consortium</i>

LISTA DE FIGURAS

Figura 2.1 Hieraquia de metaconceitos do tipo Universal, extraído de Guizzardi (2005).	23
Figura 2.2 Estrutura dos Universais de Qualidade, extraído de Guizzardi (2005)......	25
Figura 2.3 Exemplos de componentes visuais em áreas como Biologia, Medicina e Geologia, respectivamente.....	26
Figura 2.4 Extensão do triângulo de Ullmann considerando o Conhecimento Visual como um tipo especial de conceitualização que pode ter uma representação simbólica ou pictórica.....	27
Figura 3.1 Etapas da metodologia On-to-Knowledge, adaptado de York Sure et al. (1999).....	32
Figura 3.2 Modelo de Colaboração 3C.....	36
Figura 4.1 Estrutura das ontologias do sistema.....	47
Figura 4.2 Estrutura das ontologias de metadados e suas instâncias.....	48
Figura 4.3 Associação de construtos visuais a conceitos.....	49
Figura 4.4 Associação de construtos visuais a valores de propriedades.....	49
Figura 4.5 Especialização do meta-conceito <code>OntologyComponent</code>	50
Figura 4.6 Estrutura em alto-nível da Ontologia de Representação.....	51
Figura 4.7 Especialização do meta-conceito <code>OntologyConcept</code>	52
Figura 4.8 Especialização do meta-conceito <code>OntologyConceptMetaProperty</code>	52
Figura 4.9 Interação do especialista com as metapropriedades para inferir o metatipo do conceito.....	53
Figura 4.10 Especialização do meta-conceito <code>OntologyProperty</code>	54
Figura 4.11 Especialização do meta-conceito <code>OntologyRelation</code>	54
Figura 4.12 Estrutura em alto-nível da Ontologia de Colaboração.....	56
Figura 4.13 Especialização do conceito de mudança <code>Change</code>	56
Figura 4.14 Especialização do meta-conceito <code>ConceptChange</code> para mudanças em conceitos.....	57
Figura 4.15 Especialização do meta-conceito <code>PropertyChange</code> para mudanças em propriedades.....	58

Figura 4.16 Especialização do meta-conceito PropertyValueChange para mudanças em valores de propriedades.	58
Figura 4.17 Especialização do meta-conceito RelationChange para mudanças em relações.	59
Figura 4.18 Exemplo de estrutura de colaboração com construtos visuais e de fundamentação.	60
Figura 4.19 Subtipos de anotação com relações semânticas.	62
Figura 5.1 Foco da ferramenta em relação às etapas da metodologia Methontology. ...	67
Figura 5.2 Arquitetura em camadas da ferramenta.	68
Figura 5.3 Árvore de conceitos da ontologia.	74
Figura 5.4 Tela de detalhes do conceito.	75
Figura 5.5 Painel para alteração do termo linguístico de um conceito.	76
Figura 5.6 Painel para seleção de um ícone para o conceito.	76
Figura 5.7 Painel para seleção de fotos de exemplos de instâncias do conceito.	77
Figura 5.8 Painel de propriedades do conceito.	77
Figura 5.9 Painel de relações do conceito.	78
Figura 5.10 Painel de edição de uma relação.	78
Figura 5.11 Painel de metapropriedades do conceito.	79
Figura 5.12 Inferência do metatipo através da classificação de metapropriedades.	79
Figura 5.13 Classificação manual do metatipo do conceito.	80
Figura 5.14 Escolha entre os metatipos disponíveis para o conceito (oriundos da UFO-A).	80
Figura 6.1 Taxonomia de estruturas deposicionais, extraído de Lorenzatti et al. (2011).	82

LISTA DE TABELAS

Tabela 2.1 Classificação dos formalismos de representação de conhecimento de acordo com o uso de primitivas (Guarino 1995)	18
Tabela 2.2 Tabela de inferência de metatipos a partir de metapropriedades (Guizzardi 2005).	24
Tabela 3.1 Quadro comparativo das ferramentas de construção colaborativa de ontologias.....	41
Tabela 5.1 Triplas geradas ao criar um novo conceito.	72
Tabela 5.2 Triplas geradas ao alterar o metatipo de um conceito.	72
Tabela 5.3 Triplas geradas ao alterar o ícone de um conceito.....	73
Tabela 5.4 Correspondência dos componentes da ontologia de representação com construtos de OWL.	73
Tabela 6.1 Análise quantitativa da colaboração efetuada sobre a ontologia de domínio.	83
Tabela 6.2 Quadro comparativo entre as ferramentas de construção coletiva de ontologias.....	84
Tabela 6.3 Resultado das questões de avaliação da ferramenta (dados hipotéticos).....	89
Tabela 6.4 Somatório dos pontos das questões de avaliação da ferramenta, comparando diferentes abordagens/ferramentas de construção de ontologia (dados fictícios).	90
Tabela 6.5 Resultado quantitativo de objetos ontológicos e mudanças, comparando as diferentes abordagens/ferramentas de construção de ontologia (dados fictícios).	90

RESUMO

Ontologias são modelos conceituais explícitos do conhecimento compartilhado por um grupo de usuários focados em determinado domínio de interesse. Seu principal objetivo é facilitar a comunicação através da explicitação das relações semânticas entre conceitos da realidade. O problema conceitual de classificação dos entes do mundo é complexo, pois cada indivíduo possui a sua própria abstração da realidade. Quando o modelo é compartilhado, surgem mais divergências de idéias e conceitualizações. Os usuários podem dar diferentes nomes para o mesmo conceito (sinonímia) ou podem utilizar a mesma palavra para se referir a conceitos diferentes (falsa concordância). Dessa forma, a colaboração se tornou uma parte importante do processo de desenvolvimento de ontologias, auxiliando na explicitação do conceito por trás do vocabulário e na evolução do vocabulário para seus novos significados.

Uma ferramenta de software torna o processo de construção de ontologias mais eficiente e produtivo, facilitando a comunicação entre os usuários e o armazenamento do conhecimento envolvido no processo. O uso de uma ontologia de fundamentação tem papel importante na obtenção do consenso comum, pois reduz as possibilidades de interpretação sobre o domínio através da categorização semântica dos elementos da ontologia. Ela tem o objetivo de estabelecer uma base para obter-se coerência nas negociações de significado sobre um modelo conceitual, guiando a construção do modelo e ajudando a estabelecer a classificação taxonômica e as relações entre os conceitos. Dessa forma, obtém-se uma redução de ambiguidade e um aumento da precisão e consistência do modelo. Além disso, existem domínios imagísticos onde a representação simbólica linguística nem sempre é suficiente para explicitar certos conhecimentos, ou seja, onde o conhecimento visual é crucial para capturar informação e dar suporte à resolução de problemas.

Este trabalho apresenta um modelo baseado em metaontologias para permitir a especificação e colaboração de ontologias. A proposta inclui um sistema Web para a construção colaborativa de ontologias, baseado em metadados que fornecem construtos precisos. Alguns construtos são utilizados para embasar as escolhas ontológicas através da expressividade semântica de uma ontologia de fundamentação, enquanto outros permitem a associação de imagens e ícones para permitir maior entendimento do domínio. A colaboração foca na importância da fundamentação ontológica e do suporte ao conhecimento visual.

Palavras-Chave: Colaboração em Ontologias, Conhecimento Visual, Fundamentação Ontológica.

COLABORATIVE ONTOLOGY CONSTRUCTION FOR VISUAL DOMAINS BASED ON FOUNDATIONAL ONTOLOGY

ABSTRACT

Ontologies are explicit conceptual models of consensual knowledge of a community regarding some particular interest. The main goal of ontologies is improving the communication inside a group of people making use of the explicitation of semantic relationships that express the meaning of concepts. The correct classification of the entities of the world is a complex conceptual problem, since each individual retain his/her own abstraction of reality. When people collaborate to build the same conceptual model, many different opinions and divergences arise. Even restricting the domain of problem, the vocabulary exchange and ontology construction can bring unexpected complexity. The users can give distinct names to the same concept (synonym) or use the same word referring distinct concepts (false agreement). Therefore, collaboration has become an important part of the process of ontology development, supporting the explicitation of the concepts behind the vocabulary and the vocabulary evolution to their new meanings.

The support of a software tool improves the efficiency of the ontology building process, helping the communication and the structured knowledge capturing. By its side, the use of a foundational ontology has an important role in achieving the common sense by reducing the interpretation possibilities of the domain objects when restricts the semantic characterization of the ontology elements. It provides a framework to achieve coherence in the meaning negotiation during the collaboration process. In this way, the model to be built has reduced its ambiguity and increased the precision and consistence of the ontology representation. This is more important when we consider domains where the visual knowledge plays an important role in filling the semantic gaps of the concepts. In these imagistic domains, the symbolic representation through a language are not enough to express the knowledge content, so visual pattern recognition is crucial to capture information and support problem resolution.

This work presents a metaontologias based model to allow ontology specification and collaboration. The approach includes a Web system for the collaborative ontology building based in a set of metadata that provides specialized constructs to create the domain ontology elements. Some constructs are applied to support the ontological choices supported by the semantic expressivity of the foundational ontology primitives. Other constructs allow the association of images and icons to allow higher domain understanding. This work considers the collaboration as a supporting instrument for building ontologies, taking in consideration the importance of the ontological foundation and visual knowledge support.

Palavras-Chave: Ontology Collaboration, Visual Knowledge, Ontological Foundation.

1 INTRODUÇÃO

Este trabalho foca no desenvolvimento de ontologias de metadados que dão suporte à representação de domínios visuais e à colaboração dos usuários na especificação e evolução do modelo de conhecimento de domínio. A comunidade de usuários ou especialistas pode definir conceitos, propriedades e valores de propriedades, tornando explícito o seu significado intencional através do uso de primitivas de ontologia de fundamentação, de ícones visuais e de imagens ilustrativas. Os conflitos oriundos da escolha de nomes, propriedades, ícones e imagens são solucionados pelas mudanças efetuadas nos modelos e negociados através de discussões entre os usuários. As mudanças são justificadas pela definição ontológica e são armazenadas para referência futura. A ontologia de domínio resultante é armazenada em um formato compreensível e processável por computador.

Ontologia (em grego *ontos* e *logoi*, "conhecimento do ser") é a parte da filosofia que trata da natureza do ser, da realidade, da existência dos entes e das questões metafísicas em geral (Wikipedia 2012). Embora seja um termo moderno, seu estudo iniciou-se com Aristóteles, filósofo grego que explorou meticulosamente a divisão do ser em existência e essência. As ontologias têm o objetivo de explicar e formalizar a existência de uma maneira sistemática, lidando com tipos e estruturas de objetos, propriedades, eventos, processos e relações pertencentes a certa parte da realidade, em um determinado domínio de interesse. Dessa forma, atuam como infraestrutura na facilitação e criação de um entendimento comum e compartilhado (Uschold & Gruninger 1996).

Desde os tempos antigos, a descrição e compartilhamento do significado dos entes do mundo pelas pessoas dá suporte a comunicação dentro de uma comunidade e se desenvolveram como um sofisticado mecanismo de sobrevivência da espécie. A comunicação permite a transferência de experiência entre os seres e a construção coletiva de soluções ampliando a capacidade domínio do ambiente.

Ontologias podem ser utilizadas de diversas maneiras como integração inteligente de informação, processamento de linguagem natural, gerenciamento de conhecimento, e-commerce, etc.(Gómez-Pérez et al. 2007). Além disso, têm sido reconhecidas como uma camada essencial na Web Semântica emergente (Berners-Lee et al. 2001). A informação global está crescendo em grande velocidade, logo, a necessidade de reuso de ontologias é crucial para auxiliar o desenvolvimento e manutenção de sistemas eficientes e soluções baseadas em conhecimento.

Como as ontologias são utilizadas em diversos propósitos distintos, Jasper (1999) propôs uma nova definição para a palavra *ontologia* para que esse conceito seja mais difundido por outras comunidades que não somente as de engenharia de conhecimento, engenharia de software ou banco de dados. A definição dada por Jasper foi "Uma ontologia pode ter diversas formas, porém necessariamente inclui um vocabulário de termos e alguma especificação do seu significado. Isso inclui definições e a indicação de como os conceitos estão inter-relacionados o que coletivamente impõe uma estrutura no domínio e restringe a possibilidade de interpretação de termos". Na verdade, existem di-

versas definições ontológicas, algumas mais independentes e outras mais dependentes do processo de desenvolvimento. Entretanto, cada definição proposta busca complementar pontos de vista de uma mesma realidade. Na verdade, o objetivo final de conceitualizar e definir corretamente o termo ontologia é estabelecer o conhecimento consensual de uma forma genérica e formal (Corcho et al. 2003).

De acordo com (Studer et al. 1998), uma ontologia altamente informal não será uma ontologia pois não pode ser compreendida por um computador. Por esta razão, uma ontologia precisa ser representada por uma linguagem formal, permitindo o uso de raciocínio automático (inferência) e comunicação. A representação por uma linguagem permite também que a ontologia possa ser compartilhada entre seus usuários, principalmente na Web, visto que os membros da comunidade que partilha o conhecimento geralmente encontram-se separados fisicamente.

Existem diversas linguagens de formalização disponíveis, cada uma com benefícios e limitações. Entretanto, algumas mais modernas são sugeridas e recomendadas pelo *World Wide Web Consortium (W3C)* e formam a base da Web Semântica. *Resource Description Framework (RDF)* é, na verdade, uma linguagem para modelar metadados que representam descrições conceituais sobre recursos. *Web Ontology Language (OWL)* é uma linguagem para representação de conhecimento baseada em RDF (pode ser vista como uma camada de metadados de mais alto nível sobre RDF) e é caracterizada por possibilitar a expressão de semântica formal. De acordo com o objetivo de uso, pode-se usar uma das especializações da linguagem disponíveis OWL Lite, OWL DL, OWL Full.

No processo de Engenharia de Conhecimento, a preocupação geralmente gira em torno da coleta de conhecimento compartilhado por uma comunidade específica e no armazenamento desse conhecimento como uma ontologia formal que possa ser reutilizada para vários propósitos. Portanto, uma ontologia é definida como uma especificação formal de uma conceitualização compartilhada (Borst 1997). O conhecimento pode ser enriquecido se as comunidades comunicarem-se entre si, vencendo as barreiras das diferentes representações e focando em converter a conceitualização compartilhada em ontologias de domínio válidas e eficientes. Se o compartilhamento e reuso de ontologias aumenta sua qualidade (Hartmann et al. 2006), existe uma necessidade de possibilitar o compartilhamento do significado dos termos em um determinado domínio (Jasper et al. 1999).

Entretanto, inclusive em um mesmo domínio, o compartilhamento de vocabulários e a construção da ontologia pode trazer dificuldades inesperadas. Os usuários podem dar diferentes nomes para o mesmo conceito (sinonímia) ou podem utilizar a mesma palavra para se referir a conceitos diferentes (falsa concordância). Além disso, domínios de informação não são fixos ou estáticos: eles evoluem quando elementos inexistentes se tornam parte do domínio ou quando alguns elementos se tornam obsoletos (De León & Antonio 2009). Como o conhecimento comum evolui constantemente, essas mudanças precisam ser adaptadas ao modelo de domínio, atualizando a ontologia através da adição ou remoção de conceitos ou relações. Dessa forma, a colaboração se tornou uma parte importante do processo de desenvolvimento de ontologias, auxiliando na explicitação

do conceito por trás do vocabulário e na evolução do vocabulário para seus novos significados.

Com essa visão, notamos a extração de conhecimento e a sua formalização em uma ontologia é um processo árduo. A dificuldade de classificação dos entes do mundo é um problema conceitual, onde cada indivíduo possui a sua própria abstração da realidade. Essa dificuldade se agrava mais ainda quando diversas pessoas discutem sobre um mesmo modelo conceitual compartilhado, onde surgem muita divergência de idéias e conceitualizações. Baseado nessa dificuldade, diversas ferramentas para colaboração de ontologias surgiram nos últimos anos, como o Collaborative Protégé, que será discutido mais adiante nesse trabalho. Essas abordagens buscam disponibilizar um modelo compartilhado e registrar as mudanças feitas pelos especialistas, buscando reduzir o tempo necessário para a construção de uma ontologia eficiente. Entretanto, as soluções atuais não fornecem construtos suficientes para uma captura semântica mais completa do processo de conceitualização e dos próprios conceitos, ou seja, não possuem precisão suficiente. Com isso, a discussão de significado é dificultada por não existirem todos os construtos necessários para as representações desejadas. Além disso, algumas ferramentas são diretamente relacionadas com linguagens de representação, como a OWL, dificultando o uso por usuários que não conhecem a linguagem.

A construção de consenso ou resolução de problemas colaborativos é um processo que permite a membros de um time com interesses em comum trabalharem em conjunto para desenvolver uma solução aceitável por todos. Quando todos concordam e estão satisfeitos com a proposta final é dito que o consenso foi atingido. Todos os membros devem possuir direitos iguais de voto, veto e opinião e devem estar envolvidos cooperativamente no processo. A colaboração eleva a qualidade dos modelos pois as soluções produzidas incorporam diferentes perspectivas dos membros do time.

De acordo com (Guizzardi 2005), a máxima conversacional para a construção de modelos conceituais é que a mensagem deve ser relevante, completa, clara, não-ambígua, breve, sem excesso de informação e verdadeira do ponto de vista do conhecimento do emissor. Entretanto, para capturar o significado correto de um conceito, é necessário permitir que o indivíduo expresse seu entendimento sobre este conceito através do uso de propriedades que têm significado concreto para ele. Esse é o papel das Ontologias de Fundamentação: expressar as propriedades inerentes que provêm identidade aos objetos em algum mundo. Este trabalho propõe-se a prover suporte aos conceitos de fundamentação para usufruir de mais precisão na formalização do significado.

Além disso, entendemos que o ser humano possui a visão como o primeiro e mais importante sentido para capturar informações do mundo externo e gerar as suas conceitualizações. Isso é ainda mais verdade em domínios imagísticos, como a Geologia Sedimentar, onde desenvolvemos nosso estudo, no qual o reconhecimento de padrões visuais é o processo inicial para capturar informação e dar suporte à resolução de problemas. Dessa forma, este trabalho também propõe-se a prover suporte ao especialista que necessita construir um modelo conceitual de conhecimento visual, tirando vantagem da representação visual e de imagens para ajudar a expressar o significado integral dos conceitos.

Em nossa abordagem, a construção colaborativa de ontologias é baseada em informações sobre os conceitos, representados através de metadados. Esses metadados provêm a base pela qual os usuários podem expressar seu entendimento sobre o significado dos conceitos utilizando um vocabulário específico, provendo uma fundamentação ontológica homogênea ao modelo. O uso de metadados para descrever metaconstrutos ontológicos e representações visuais independentes de domínio nos permite criar um ambiente para a criação e evolução de ontologias de domínio sem exigir que os usuários tenham qualquer conhecimento prévio sobre linguagens formais de representação (o que ocorre com algumas soluções colaborativas atuais), mas fornecendo mecanismos para manipular informação visual e simbólica e para expressar modelos semânticos ricos.

As ontologias de metadados assim desenvolvidas auxiliam a representação de conceitos em domínios visuais através da construção colaborativa dos significados suportado pela ferramenta. A ontologia de fundamentação oferece uma orientação para fazer o ancoramento simbólico dos termos de forma não ambígua, permitindo que os usuários negociem a melhor forma de representação daquela significação através de discussões entre si, registradas na ferramenta. Espera-se que essa forma coletiva de construção crie modelos melhor ancorados na realidade a ser capturada.

A principal contribuição desse trabalho consiste em uma abordagem baseada em metadados que especificam os construtos ontológicos e a colaboração de ontologias de forma independente de linguagens formais de representação, através de um ambiente colaborativo para construção e evolução de ontologias com interface intuitiva para o usuário.

Os objetivos desse trabalho são:

- a) disponibilizar um sistema web, acessível via navegador, que possibilite aos especialistas alterarem diretamente a ontologia, registrando as mudanças automaticamente.
- b) fornecer construtos ontológicos de fundamentação para embasar as escolhas ontológicas através da expressividade semântica de uma ontologia de fundamentação;
- c) fornecer construtos ontológicos visuais para representar conhecimento visual e dar suporte a domínios imagísticos;
- d) fornecer construtos de colaboração que permitam um mapeamento das mudanças ontológicas e discussões organizadas entre os membros da comunidade.

Como objetivo desta proposta, buscamos possibilitar gerar, ao final de um processo de colaboração, uma ontologia de domínio robusta, com maior precisão na especificação do significado dos conceitos e com maior potencialidade de integração com outras aplicações. As capacidades do modelo de metadados e da aplicação de colaboração propostos são avaliados através de um exemplo de aplicação prático através da construção colaborativa de uma ontologia para descrição de fácies sedimentares no domínio da Estratigrafia Sedimentar em Geologia.

A seguir descrevemos a estrutura deste trabalho. No Capítulo 2 introduzimos uma série de conceitos básicos necessários para o entendimento das soluções apresentadas neste trabalho, abordando assuntos referentes principalmente à modelagem conceitual, fun-

damentação ontológica e conhecimento visual. No Capítulo 3 apresentamos algumas das principais metodologias existentes para engenharia e construção de ontologias, assim como conceitos de base para o desenvolvimento colaborativo de ontologias. Além disso, apresentamos uma avaliação comparativa das principais ferramentas colaborativas que representam o estado da arte nesta área. No Capítulo 4 apresentamos a arquitetura do modelo de conhecimento baseado em metadados, o qual embasa o ambiente colaborativo através de metaontologias que formalizam a representação dos construtos ontológicos e a estrutura da colaboração. No Capítulo 5 descrevemos em detalhes a implementação e os recursos do ambiente para construção colaborativa de ontologias desenvolvido neste trabalho, o *Portal Obaitá*. Por fim, no Capítulo 7 apresentamos as conclusões e antecipamos alguns trabalhos futuros.

2 MODELAGEM CONCEITUAL E ONTOLOGIAS

A modelagem conceitual é responsável por identificar, analisar e descrever os conceitos de um domínio, bem como suas relações semânticas, com a ajuda de uma linguagem de modelagem baseada em um conjunto de metaconceitos (formando um metamodelo). A modelagem ontológica procura identificar os conceitos de um domínio e especificá-los através de uma ontologia, com uma linguagem de especificação baseada em categorias ontológicas independentes de domínio (também chamadas de ontologias *upper-level*) (Guizzardi et al. 2004).

Uma ontologia busca capturar o conhecimento consensual que é aceito por um grupo (Gómez-Pérez et al. 2007). Em uma visão prática, uma ontologia é uma especificação explícita de uma conceitualização. A conceitualização é uma visão abstrata e simplificada do mundo que se deseja representar. É importante, e foco deste trabalho, que os membros do grupo estejam alinhados com um mesmo compromisso ontológico, o que significa que estão de acordo em utilizar um vocabulário compartilhado de termos de maneira coerente e consistente (Gruber. 1995). Guarino expande a definição de Gruber fazendo o paralelo com o uso pela Inteligência Artificial, introduzindo que uma ontologia é um artefato de engenharia, constituído de um vocabulário específico utilizado para descrever certa realidade, além de suposições explícitas relacionadas ao significado intencional das palavras do vocabulário.

Segundo Guarino, uma ontologia é uma teoria lógica que leva em conta o significado intencional de um vocabulário formal, isto é, o compromisso ontológico com uma conceitualização particular do mundo. Os modelos intencionais de uma linguagem lógica que utilize tal vocabulário são restringidos pelo seu compromisso ontológico. Uma ontologia reflete indiretamente esse compromisso (e a conceitualização subjacente) pela aproximação desses modelos intencionais (Guarino 1998).

Segundo Guarino (Guarino 1998) ambas as definições de ontologias, da filosofia e computacional, estão relacionadas porém possuem diferenciação terminológica. A visão filosófica define a *conceitualização* enquanto a visão computacional define a *ontologia*, como objeto da engenharia de conhecimento. Dessa forma, ontologias que possuam vocabulários diferentes podem ainda assim compartilhar da mesma conceitualização.

Guarino define as relações conceituais em um espaço de domínio (*domain space*), enfatizando que o foco está na representação do significado das relações conceituais, também chamadas de relações intencionais. Dessa forma, o espaço conceitual pode ser definido como:

$$\langle D, W \rangle$$

Onde D é um domínio de informação e W é o conjunto de mundos possíveis (conjunto de estados possíveis que o domínio pode assumir). A conceitualização é o conjunto de relações conceituais definidas sobre um espaço de domínio, e pode ser representa-

da como abaixo, onde R é o conjunto de relações conceituais para o espaço de domínio $\langle D;W \rangle$.

$$C = \langle D, W, R \rangle$$

O objetivo da modelagem de conhecimento é elaborar modelos que representem corretamente o domínio em questão. Dessa forma, embasar o modelo ontologicamente evita o desvio e auxilia na criação de ontologias consistentes. As escolhas ontológicas devem ficar claras e explícitas para permitir o entendimento mútuo, logo, é necessário explicitar as relações intencionais e o significado das primitivas da linguagem de representação a ser utilizada para especificar o conhecimento. Em (Guarino 1995) é estabelecido um nível de classificação dos formalismos de acordo com os tipos de primitivas em que se baseiam. A Tabela 2.1 mostra a classificação dos formalismos de forma detalhada.

Tabela 2.1 Classificação dos formalismos de representação de conhecimento de acordo com o uso de primitivas (Guarino 1995)

Nível	Primitivas	Interpretação	Principal característica
Lógico	Predicados e funções	Arbitrária	Formalização
Epistemológico	Relações estruturais	Arbitrária	Estrutura
Ontológico	Relações ontológicas	Restrita	Significado
Conceitual	Relações conceituais	Subjetiva	Conceitualização
Lingüístico	Termos lingüísticos	Subjetiva	Dependência de linguagem

No nível *lógico*, as primitivas básicas são os predicados e funções, que dão semântica formal em termos das relações entre os objetos do domínio. As relações aqui são genéricas e independente de conteúdo. Esse nível formaliza as primitivas, porém a interpretação das mesmas é totalmente arbitrária.

No nível *epistemológico* busca-se diminuir o espaçamento entre o nível lógico, onde as primitivas são extremamente genéricas, e o nível conceitual, onde é dado significado às primitivas. Logo, enquanto o nível lógico lida com predicados abstratos e o nível conceitual com conceitos específicos, as primitivas declaradas no nível epistemológico permitem que um conceito seja visto como uma primitiva de estruturação de conhecimento. Um conceito no nível epistemológico (correspondente à um predicado unário no nível lógico) possui estrutura interna em termos de outros conceitos ou relações binárias (papéis).

No nível *ontológico*, os compromissos ontológicos associados às primitivas de linguagem são explicitamente especificados através da restrição semântica das primitivas ou através de postulados de significados introduzidos diretamente na linguagem. O ob-

jetivo é restringir a possibilidade de interpretações divergentes. Logo, o nível ontológico foca-se em estabelecer o significado do domínio em termos das primitivas. Pode-se dizer que uma linguagem é ontologicamente adequada se, no nível sintático, ela possuir granularidade suficiente e capacidade de reificação para expressar os postulados de significado de suas próprias primitivas, ou se é possível dar interpretação ontológica formal às suas primitivas no nível semântico.

No nível *conceitual*, as primitivas possuem interpretação cognitiva definida, correspondendo a conceitos independentes de linguagem. O esqueleto da estrutura de domínio é definido e o conhecimento é especificado como uma especialização deste esqueleto. Definições nesse nível podem vir acompanhadas de definições no nível ontológico.

No nível *linguístico*, as primitivas têm o objetivo de se referir diretamente a termos como verbos e substantivos.

Essas definições são importantes pois o nível de abstração do conhecimento deve ser definido antes da etapa de modelagem. Um conhecimento pode ser especificado no nível conceitual e nunca ser implementado em uma linguagem de representação formal. Entretanto, para fins de comunicação e colaboração, os modelos mais interessantes no contexto deste trabalho são aqueles que podem ser formalizados em uma linguagem.

2.1 Metapropriedades Ontológicas

Em 1990 foi proposta por Guarino uma metodologia chamada OntoClean (Guarino 1992), focada na análise do significado dos conceitos e propriedades de um domínio em relação à maneira como eles se relacionam na realidade. Essa metodologia guia a construção de modelos de conhecimento ao fornecer uma base ontológica para representar os conceitos de forma a evitar interpretações incorretas, além de otimizar o processo de construção de ontologias. A OntoClean inicia o processo de modelagem através da escolha das primitivas que irão representar cada um dos conceitos com base na análise de metapropriedades. As metapropriedades buscam explicitar restrições ontológicas no nível do conhecimento, facilitando a identificação do conceito através de sua *rigidez, identidade, unidade e dependência*. De acordo com as metapropriedades definidas, são impostas restrições à construção taxonômica da ontologia, facilitando a escolha dos construtos de representação e gerando um modelo mais coeso com a realidade.

A metapropriedade *rigidez* pode assumir os valores *rígida, semirrígida e antirrígida*, representadas respectivamente por **+R**, **-R** e **~R**. Uma propriedade é classificada como rígida se ela é essencial para todos os indivíduos que a possuem, ou seja, caso algum indivíduo deixe de possuir a propriedade, ele também passará a não existir mais. Por exemplo, a propriedade *SER MACACO* poderia ser aplicada à indivíduos primatas. Um indivíduo desse grupo, ao perder essa propriedade essencial, não poderia existir, logo é caracterizada como uma propriedade *rígida*. Entretanto, a propriedade *COME BANANA* pode não ser compartilhada por todos do grupo, logo não é essencial e portanto não é rígida, no caso, *semirrígida*. Propriedades que são essenciais para algumas de seus indivíduos mas não para outros são classificadas como *semirrígidas*. Propriedades que não são essenciais para todos os seus indivíduos são classificadas como *antirrígidas*. Uma restrição taxonômica imposta por essa propriedade é que propriedades rígidias não po-

dem ser especializações (herança) de propriedades não rígidas. Por exemplo, a propriedade *SER MACACO* não pode especializar a propriedade *SER MACACO DE CIRCO*, pois essa última é semirrígida enquanto a primeira é rígida. O indivíduo que perder a propriedade *SER MACACO* deixará de existir, enquanto o indivíduo que perder a propriedade *SER MACACO DE CIRCO* permanecerá sendo macaco. Dessa forma, o correto é que as propriedades não rígidas sejam especializações de propriedades rígidas, para evitar inconsistências.

A metapropriedade *identidade* é focada em permitir identificar um indivíduo específico entre demais indivíduos. Isso é feito através da definição de quais características tornam um indivíduo único. Um conceito pode fornecer identidade (+**O**) ou apenas carregá-la (+**I**). Por exemplo, uma *PESSOA* fornece identidade (+**O**), pois possui um conjunto de características que permitem a distinção entre outra pessoa. Entretanto, um *ESTUDANTE* não fornece identidade, pois ela é fornecida pela *PESSOA* à qual se aplica a propriedade de estudante. Logo, *ESTUDANTE* carrega identidade (+**I**), mas não a fornece. Propriedades que na verdade são atributos não possuem identidade alguma, como a propriedade *BONITA*, e possuem o rótulo (-**I**). Da mesma forma que a rigidez, a identidade também restringe a taxonomia, impedindo que propriedades com o rótulo (+**I**) sejam declaradas como especializações de propriedades com o rótulo (-**I**), por exemplo, uma *PESSOA* não pode ser uma subclasse de *BONITA*. Ou seja, as noções de *identidade* e *rigidez* são utilizadas para definir as possibilidades de estruturação hierárquica entre objetos representados, garantindo que pertençam a hierarquia objetos que compartilhem os mesmos princípios de identidade. Assim, apenas *HOMEM* e *MULHER* são efetivamente subclasses de *PESSOA*, porque carregam identidade e são especializações reais da conceito *PESSOA*. *ESTUDANTE* recebe a identidade mas não é uma especialização da classe.

A metapropriedade *unicidade* busca identificar se o objeto pode ser reconhecido a partir de suas partes e limites. Isso é feito através da análise da composição do objeto identificando se é unitário ou uma soma de objetos unitários. O conjunto de condições necessárias para determinar se as propriedades possuem instâncias ou indivíduos unitários é chamado de critério de unicidade. As propriedades cujos indivíduos são unitários e possuem o mesmo critério de unicidade são rotulados como (+**U**). Por exemplo, indivíduos de *SER UMA FAIXA DE AREIA* possuem rótulo (+**U**) pois é possível identificar os limites do objeto. Propriedades cujos indivíduos são unitários porém possuem diferentes critérios de unicidade são rotuladas como (-**U**), ou seja, não carregam unicidade. Por exemplo, a propriedade *SER QUADRÚPEDE*, pode possuir diferentes critérios de unicidade entre seus indivíduos, cada um podendo possuir um critério de unicidade distinto. Propriedades cujos indivíduos não são unitários são rotuladas com (~**U**), ou seja, carregam antiunicidade. Por exemplo, *SER (UMA QUANTIDADE DE) AREIA* possui indivíduos não unitários, pois não é possível estabelecer limites bem definidos. Essa propriedade também restringe a taxonomia de tal forma que propriedades (+**U**) não podem especializar propriedades (-**U**).

A metapropriedade *dependência*, rotula com (+**D**) as propriedades em que seus indivíduos dependam da existência de indivíduos de outra propriedade para existirem. Dessa forma, pode-se dizer que uma propriedade X pode ser externamente dependente de uma propriedade Y. Caso essa dependência não exista, a propriedade é rotulada com (-

D) Por exemplo, a propriedade *MÃE* só pode existir caso a propriedade *FILHO* exista, ou seja, uma é externamente dependente da outra. Essa propriedade restringe a taxonomia impedindo que propriedades com rótulo (+**D**) sejam especializações de propriedades com rótulo (-**D**).

2.2 Ontologias de Fundamentação

Uma ontologia de fundamentação tem o objetivo de estabelecer uma base para obter-se coerência nas negociações de significado oriundas da colaboração de indivíduos sobre um modelo conceitual (Guizzardi et al. 2009). As ontologias de fundamentação têm sido utilizadas no processo de avaliação e (re)engenharia de linguagens de modelagem conceitual pois descrevem categorias que são usadas para a construção da conceitualização que representa determinada parte da realidade.

A base fornecida pela fundamentação traz vantagens à modelagem guiando a construção do modelo de conhecimento, ajudando a estabelecer a classificação taxonômica e as relações entre os conceitos. Dessa forma, se obtém uma redução de ambiguidade e um aumento da precisão e consistência do modelo, criando modelos melhor ancorados na realidade. A seguir veremos alguns conceitos que embasam a ontologia de fundamentação proposta pelo Guizzardi, a qual é objeto de estudo deste trabalho.

2.2.1 Isomorfismo de Modelos Conceituais

De acordo com (Guizzardi 2005), a máxima conversacional para a construção de modelos conceituais é que a mensagem deve ser relevante, completa, clara, não-ambígua, breve, sem excesso de informação e verdadeira do ponto de vista do conhecimento do emissor. Isso é feito comparando-se o nível de homomorfismo entre um objeto concreto da realidade e a uma representação formal e explícita em uma conceitualização. Para isso, pode-se analisar as propriedades que classificam o isomorfismo dos modelos conceituais: Lucidez, Correção, Laconicidade e Completeza. Essas propriedades serão utilizadas neste trabalho e portanto são descritas a seguir:

2.2.1.1 Lucidez

Uma especificação (representação através de construtos) *S* é chamada de lúcida em relação a um modelo conceitual *M* se um mapeamento (de representação) de *M* para *S* é injetivo. Um mapeamento de *M* para *S* é injetivo se e somente se todas as entidades da especificação *S* representam **no máximo uma** (ou nenhuma) entidade do modelo *M*. Diagramas não-lúcidos ocorrem quando há uma sobrecarga nos construtos de especificação, ou seja, quando mais de um conceito do modelo mapeiam para o mesmo construto, gerando ambiguidade.

2.2.1.2 Correção

Uma especificação *S* é chamada de correta em relação à um modelo *M* se há um mapeamento (de representação) sobrejetivo de *M* para *S*. Um mapeamento de representação de *M* para *S* é sobrejetivo se e somente se o mapeamento de interpretação correspondente de *S* para *M* é total, isto é, se e somente se todos os construtos da especificação *S* representam **pelo menos** uma entidade do modelo *M* (talvez mais de uma). Dia-

gramas não-corretos ocorrem quando existem construtos da especificação que não mapeiam para entidades na conceitualização, gerando um excesso de construtos.

2.2.1.3 *Laconicidade*

Uma especificação S é chamada de lacônica em relação à um modelo M se há um mapeamento de S para M que seja injetivo, isto é, se e somente se todas as entidades do modelo M são representadas por **no máximo** uma (ou nenhuma) entidade na representação S . Diagramas não-lacônicos ocorrem quando existem conceitos do modelo que são mapeados para mais de um construto na especificação, gerando redundância de construtos e uma complexidade desnecessária para a representação.

2.2.1.4 *Completeza*

Uma especificação S é chamada de completa em relação à um modelo M se há um mapeamento de S para M que seja sobrejetivo. Um mapeamento de S para M é sobrejetivo se e somente se o mapeamento de representação correspondente de M para S é total, isto é, se e somente se todas as entidades do modelo são representadas por **pelo menos uma** (ou mais de uma) entidade na representação S . Diagramas incompletos ocorrem quando existem conceitos do modelo que não são mapeados para construtos, ou seja, falta expressividade de especificação para representar determinadas entidades da conceitualização.

Os metaconstrutos propostos neste trabalho buscam fornecer uma base que mantenha, tanto quanto possível, o isomorfismo entre a realidade modelada e o modelo produzido pela ferramenta de colaboração. Dessa forma, a ferramenta possibilita uma modelagem mais completa, através de construtos suficientes para expressar o domínio em questão.

2.2.2 **Unified Foundational Ontology**

Em 2005, foi proposta uma ontologia de fundamentação unificada que provê fundamentação ontológica na construção de modelos conceituais (Guizzardi & Wagner 2004). A *Unified Foundational Ontology* ou *UFO* é dividida em em três fragmentos denominados UFO-A (*Ontology of Endurants*), UFO-B (*Ontology of Perdurants*) e UFO-C (*Ontology of Social and Intentional Entities*). A UFO-A define o núcleo dessa ontologia, constituindo uma teoria estável, formalmente caracterizada com o aparato de uma lógica modal de alta expressividade e possuindo forte suporte empírico promovido por experimentos em psicologia cognitiva (Guizzardi 2005). A UFO-A introduz conceitos estruturantes sobre os objetos físicos da realidade para oferecer mais semântica às linguagens de modelagem conceitual, os quais estão descritos a seguir.

Universais (*Universals*) representam conceitos abstratos e de alto nível que caracterizam diferentes classes de indivíduos. Exemplos de Universais são: carro, avião, planta, livro, médico, peso. Os Universais se dividem em Universais Perenes (*Perdurant Universals*) e Universais Duradouros (*Endurant Universals*). A hierarquia de conceitos pode ser vista na Figura 2.1.

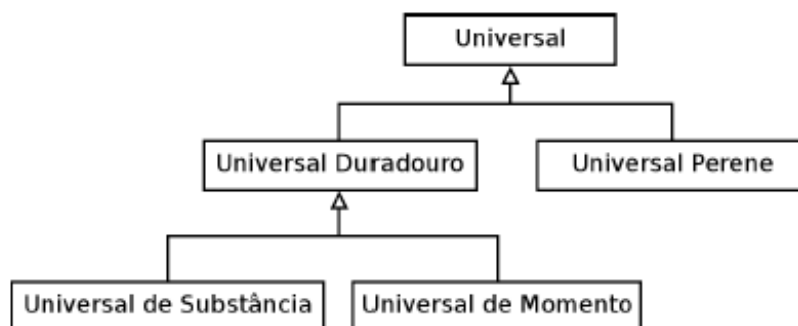


Figura 2.1 Hierarquia de metaconceitos do tipo Universal, extraído de Guizzardi (2005).

Universais Perenes representam conceitos cujos indivíduos são compostos de partes temporais, eles acontecem no tempo, se estendendo ao longo do tempo e acumulando partes temporais, como por exemplo: corrida, tempestade, discussão, festa.

Universais Duradouros representam conceitos cujos indivíduos estão sempre presentes por completo, ou seja, estão sempre compostos pelas suas partes e sua identidade não varia ao longo do tempo. Universais Duradouros se dividem em duas classes, Universais de Momento (*Moment Universals*) e Universais de Substância (*Substantial Universals*) (Guizzardi 2005).

Universais de Momento são conceitos que existirão somente durante a existência de outro indivíduo concreto. Exemplos desse tipo de universal são peso, altura, comprimento, sentimento (Guizzardi 2005). Universais de Substância representam conceitos cujos indivíduos são concretos, persistem no tempo e mantêm sua identidade, como por exemplo: pessoa, carro, gato, cavalo. Esses universais também são independente entre si, ou seja, a existência um não depende do outro, com exceção daqueles que possuem relação de paronímia essencial (relações que condicionam a existência de um indivíduo à pré-existência de outro). A estrutura desses universais pode ser vista na Figura 2.2.

De acordo com Guizzardi (2005), Universais de Substância são especializados em diferentes categorias, também vistas como metatipos de conceitos, por caracterizarem com mais precisão os conceitos da ontologia baseado em suas metapropriedades. São elas:

Tipo (*kind*): representa a classe de indivíduos cujas instâncias são complexos funcionais. Exemplo: *pessoa, cachorro, árvore, cadeira, televisão*.

SubTipo (*subkind*): representa a classe de indivíduos rígidos que carregam o princípio de identidade fornecido pelo conceito de nível superior (*kind*). Exemplo: *Homem* pode ser um subtipo do conceito *Pessoa*.

Fase (*phase*): representa as classes de indivíduos particionadas em relação a determinada fase de existência. Exemplo: *Lagarta* e *Borboleta* são fases do conceito *Lepdopterum*.

Papel (*role*): representa a classe de indivíduos com dependência relacional de um *kind*, especificando o papel que este pode representar no domínio. Exemplo: o papel *Estudante* pode ser representado pelo conceito *Pessoa*.

Quantidade (*quantity*): representa indivíduos que referem-se à porções de substâncias. Essas substâncias estão relacionadas a termos incontáveis da linguagem natural tais como *terra*, *água*, *sal* e *fermento*. *Rocha* é um exemplo de Quantidade.

Categoria (*category*): representa classes de indivíduos distintos do tipo *kind* que compartilham determinadas propriedades essenciais. Exemplo: a categoria *Ser Vivo* engloba os conceitos *Animal* e *Pessoa*, ambos do tipo *kind*.

Mistura de Papéis (*roleMixin*): representa classes de indivíduos distintos do tipo *role* que compartilham determinadas propriedades essenciais. Exemplo: o conceito *Cliente*, do tipo *roleMixin*, engloba os conceitos *ClientePessoaFisica* e *ClientePessoaJuridica*, ambos do tipo *role*.

Mistura (*mixin*): representa classes de indivíduos nos quais algumas de suas instâncias possuem propriedades essenciais enquanto outras possuem propriedades acidentais. Exemplo: o *mixin Sentável* representa uma propriedade que pode ser essencial para *Cadeira* ou *Sofá*, porém acidental para *Caixa* ou *Pedra*.

Para facilitar a classificação dos Universais de Substância, Guizzardi propôs uma tabela que possibilita a identificação de alguns dos principais metatipos a partir das metapropriedades dos conceitos, a qual pode ser vista na tabela 2.2.

Tabela 2.2 Tabela de inferência de metatipos a partir de metapropriedades (Guizzardi 2005).

Metatipo	Provê- Identidade (O)	Carrega- Identidade (I)	Rigidez (R)	Dependência (D)
SORTAL	-	+	+/-	+/-
kind	+	+	+	-
subkind	-	+	+	-
role	-	+	-	+
Phase	-	+	-	-
NÃO-SORTAL	-	-	+/~	+/-
category	-	-	+	-
roleMixin	-	-	-	+
mixin	-	-	~	-

Também existem os chamados Universais de Qualidade (*Quality Universal*) e Universais de Modo (*Mode Universal*). O primeiro refere-se a conceitos que representam qualidades e são inerentes à um único indivíduo, enquanto o segundo refere-se a conceitos que representam qualidades mas que são inerentes à outros universais. Por exemplo, *sensação* é um Universal de Modo, enquanto *cor* é um Universal de Qualidade. Universais de Qualidade estão associados à uma Estrutura de Qualidade. Estruturas de Qualidade representam um conjunto de possíveis valores (Qualia) que o Universal de Qualidade pode assumir. Por exemplo, no caso de *altura*, a Dimensão de Qualidade ao qual o Universal se relacionada é o conjunto de reais positivos (R^+). Cada posição em uma dimensão de qualidade, ou valor possível, é chamado de Quale (Guizzardi 2005). Essa estrutura pode ser melhor vista na Figura 2.3.

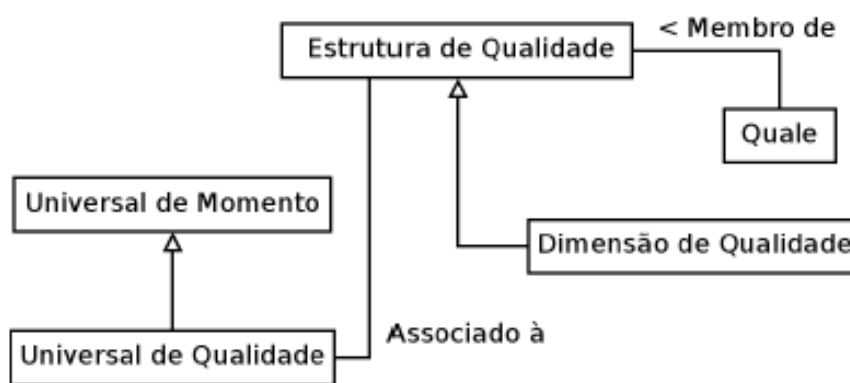


Figura 2.2 Estrutura dos Universais de Qualidade, extraído de Guizzardi (2005).

A UFO-A também possui classificação para as relações existentes entre conceitos. Na ontologia de fundamentação, essas relações podem ser do tipo *Material* ou *Formal*. As relações Formais ocorrem entre duas entidades diretamente, sem nenhum indivíduo que os relacione, como “5 é maior que 3” ou “esse dia é parte de outro mês”. As relações estruturantes que expressam hierarquia (é-um), partonomia (parte-de e seus subtipos discutidos no Capítulo 4), entre outras são relações formais. As relações Materiais, por outro lado, possuem estrutura material própria e incluem exemplos como empregos, beijos, conexões aéreas e compromissos. As relações Materiais são intermediadas por um indivíduo que têm o poder conectar duas entidades, o chamado *Relator*. Por exemplo, o relator *emprego* conecta um *empregado* com uma *empresa*.

Neste trabalho, além do interesse na fundamentação ontológica há um forte interesse no suporte à domínios visuais. Optamos por utilizar a UFO-A pois essa ontologia de fundamentação provê os construtos necessários para trabalhar com Universais de Substância, que são os objetos que têm representação visual, ou seja, objetos concretos e visíveis.

Os construtos propostos por Guizzardi guiam a construção do modelo de conhecimento, auxiliando na classificação taxonômica e das relações entre os conceitos. Dessa forma, diminui-se a ocorrência de ambiguidades e aumenta-se a precisão do modelo de domínio.

2.3 Conhecimento Visual

Em (Lorenzatti et al. 2011), conhecimento visual é descrito como a seguir:

"Conhecimento visual é o conjunto de modelos mentais (conceitos) de cenas reais ou imaginárias manipuladas pelo cérebro para lidar com tarefas baseadas em imagens como interpretação de imagens ou reconhecimento de padrões ou formas na realidade."

Em outras palavras, conhecimento visual é o conjunto de modelos mentais que dá suporte ao processo de raciocínio sobre informações oriundas de aspectos visuais das entidades do domínio.

Como sabemos, existem domínios de informação onde o conhecimento visual é parte crucial do processo de resolução de problemas. Áreas como a Medicina, Biologia, Engenharia ou Geologia fazem uso constante de componentes visuais para facilitar a aquisição e disseminação de conhecimento que, se fosse tratado somente de forma textual, dificultaria muito o entendimento pelas partes, como pode ser visto na Figura 2.4, a qual exhibe, em ordem, imagens que exibem um tecido celular (Biologia), um raio-x (Medicina) e uma estrutura rochosa (Geologia).



Figura 2.3 Exemplos de componentes visuais em áreas como Biologia, Medicina e Geologia, respectivamente.

Na Geologia, por exemplo, o especialista analisa informações visuais de formações rochosas para identificar feições e correlacioná-las com a ocorrência de certos fenômenos geológicos (Abel 2001). Logo, a construção de ontologias em domínios visuais exige mais do que descrições proposicionais para explicar os conceitos. Esse domínios demandam construtos específicos para expressar conteúdo que não pode ser expressado textualmente.

Através da construção de modelos ontológicos alternativos é possível representar o conhecimento visual em domínios imagísticos. Diversas abordagens anteriores buscam captar o conhecimento visual como redes neurais, raciocínio baseado em casos e processamento de imagens (Lorenzatti 2009). É importante ressaltar que o foco da aquisição e disseminação de conhecimento visual não está nas imagens, mas sim no modelo mental criado pelo especialista para expressar a carga teórica e prática adquirida ao longo dos anos.

É possível extrair o conhecimento de especialistas através de técnicas de aquisição de conhecimento, porém compartilhá-lo muitas vezes é complicado pois cada indivíduo

cria seu próprio modelo mental para a realidade. Para entender essa questão, pode-se analisar o triângulo de Ullmann (Guizzardi 2005), que define a relação entre um *Objeto*, um *Conceito* e um *Símbolo*. Um *Objeto* pode ser encontrado na realidade porém é dependente da abstração do agente observador, um *Conceito* é a representação interna mental criada pelo indivíduo para abstrair o *Objeto*. Por fim, o *Símbolo* é uma representação linguística do *Conceito* em uma linguagem. O *Símbolo* é parte essencial do triângulo pois permite a externalização do conhecimento e conseqüentemente o seu compartilhamento por indivíduos de um mesmo grupo.

Na falta de símbolos próprios na linguagem para a representação de todos os aspectos do conceito relevantes ao observador, representações pictóricas complementam as representações simbólicas. Dessa forma, símbolos e as representações pictóricas mantêm ambos uma relação com o conceito a ser externalizado de modo que a representação pictórica *ancora* o símbolo, como representado na extensão do triângulo de Ullman proposto por Lorenzatti e apresentado na Figura 2.5.

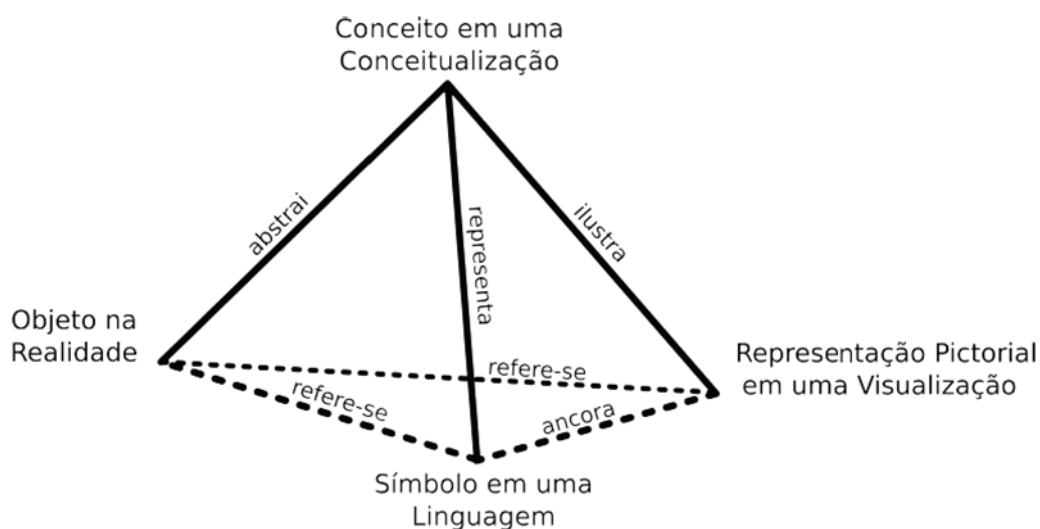


Figura 2.4 Extensão do triângulo de Ullmann considerando o Conhecimento Visual como um tipo especial de conceitualização que pode ter uma representação simbólica ou pictórica.

Ícones são representações gráficas que abstraem informações não-relevantes ao processo de reconhecimento visual, omitindo os aspectos irrelevantes à intenção do observador. Um ícone busca representar exclusivamente o conceito em questão, tentando evitar interpretações ambíguas ou incorretas. Dessa forma, os ícones são perceptualmente similares ao que eles representam. Seu significado é capturado através do mesmo processo de percepção utilizado para reconhecer o objeto ou evento representado. Em outras palavras, seu significado pode ser compreendido através da observação da representação.

Imagens são fotografias de conceitos, que podem ser associadas aos símbolos da linguagem para prover exemplos de instâncias de um conceito, apoiando o esclarecimento do seu significado.

Este trabalho busca estudar como a colaboração entre membros de um grupo pode auxiliar na obtenção de uma conceitualização compartilhada melhor ancorada na reali-

dade em domínios imagísticos, onde a disseminação do conhecimento é mais difícil devido a falta de símbolos adequados para representar os aspectos visuais dos objetos modelados.

A área de aplicação da proposta desenvolvida é a Estratigrafia Sedimentar, uma sub-área de Geologia que busca interpretar a sucessão dos depósitos de sedimentos e sua relação com o processo gerador e com o tempo geológico. Os depósitos sedimentares possuem enorme importância econômica, uma vez que definem as condições para a formação das mais importantes jazidas econômicas, tais como ouro, prata, carvão e, especialmente, petróleo. A investigação dos depósitos sedimentares se dá através da análise do tipo de rocha e do arranjo espacial entre os grãos desta rocha (chamado de *estrutura sedimentar*), que definem um aspecto visual único para cada processo gerador e condicionam a ocorrência das jazidas. O reconhecimento e descrição destes aspectos constituem na habilidade básica para a exploração mineral.

Apesar da importância da descrição dos aspectos visuais de rochas e estruturas como base científica para exploração, a Geologia como Ciência mostra-se ainda imatura na definição da terminologia formal de descrição. Em especial, na Estratigrafia Sedimentar, muitos aspectos diagnósticos das rochas são referidos na literatura por diferentes termos técnicos, enquanto um mesmo termo pode descrever aspectos geneticamente distintos. Os problemas são ainda agravados pela multiplicidades de idiomas onde os termos são inicialmente definidos e posteriormente traduzidos. Muitos esforços têm sido realizados pela academia e indústria para estabelecer a taxonomia e tesauro descritivo das rochas sedimentares, com o objetivo de melhorar a comunicação científica e permitir o desenvolvimento de sistemas para correlação estratigráfica.

No próximo capítulo apresentamos algumas das principais metodologias existentes para engenharia e construção de ontologias, assim como conceitos de base para o desenvolvimento colaborativo de ontologias. Por fim, apresentamos uma avaliação comparativa das principais ferramentas colaborativas que representam o estado da arte nesta área.

3 ENGENHARIA DE ONTOLOGIAS

A Engenharia de Ontologias estuda as atividades envolvidas no processo de desenvolvimento de ontologias, analisando quais métodos de construção, gerenciamento, manipulação e integração estão envolvidos nos seus ciclos de vida. Essa área originou algumas metodologias que possuem o intuito de definir “o que”, “quem” e “quando” uma atividade deve ser efetuada. Uma metodologia consiste de processos distintos e cada processo é composto por atividades que por sua vez podem ser uma ou mais tarefas. Uma tarefa pode ser vista como a menor unidade de trabalho em um processo de construção de ontologias (Gómez-Pérez et al. 2007).

Para permitir que times cooperativos em locais separados geograficamente interajam na construção de ontologias, foi identificado o processo de desenvolvimento de ontologias (Fernandez et al. 1997), que se baseia nos padrões definidos pelo IEEE (IEEE 1996). Esse processo identifica quais atividades estão envolvidas na construção de uma ontologia. Como definido em (Gómez-Pérez et al. 2007), esse processo se divide em três categorias distintas de atividades:

Atividades de gerenciamento da ontologia : inclui cronograma e escalonamento de tarefas e auditoria de controle e qualidade.

Atividades orientadas ao desenvolvimento da ontologia: divididas, em três etapas, pré-desenvolvimento, desenvolvimento e pós-desenvolvimento. A primeira etapa contempla principalmente um *estudo do ambiente*, avaliando quais plataformas utilizarão a ontologia, quais aplicações devem ser integradas, etc, e um *estudo de exequibilidade*, avaliando se é possível contruir a ontologia. A segunda etapa envolve as atividades de *especificação*, definindo o porquê de construir e quais as intenções da ontologia, *conceitualização*, estruturando o domínio de informação com modelos de significado no nível de conhecimento (Newell, 1982), *formalização*, transformando o modelo conceitual em um modelo formal semi-computável, e *implementação*, construindo modelos computáveis em uma linguagem ontológica, como OWL. Na etapa final, as atividades envolvidas são *manutenção*, onde são efetuadas atualizações e correções na ontologia, caso necessárias, e *uso*, onde a ontologia já pode ser utilizada ou reutilizada por algumas aplicações.

Atividades de suporte: envolvem as atividades que ocorrem em paralelo ao desenvolvimento da ontologia, incluindo *aquisição de conhecimento*, focada em extrair o conhecimento dos especialistas em determinado domínio ou através de processos automáticos, *avaliação*, onde é feito um julgamento técnico da ontologia em relação à ambientes de software, *integração*, envolvendo esforço de integração quando a ontologia é desenvolvida a partir de ontologias pré-existentes, *agrupamento*, quando a ontologia é desenvolvida a partir de diversas ontologias pré-existentes, *alinhamento*, quando é necessário descrever mapeamentos entre ontologias, *documentação*, clarificando detalhes de implementação e *gerência de configuração*, armazenando os registros de todas as versões de documentação e da ontologia ao longo do processo de desenvolvimento.

As atividades citadas são abordadas de forma diferenciada de acordo com a metodologia escolhida para a engenharia da ontologia. Algumas das principais metodologias são vistas a seguir.

3.1.1 Metodologias para Construção de Ontologias

Como citado, algumas metodologias foram desenvolvidas para guiar as atividades envolvidas no ciclo de vida de uma ontologia. Abaixo estão algumas delas:

Guia 101

Definido em (Noy et al. 2001), a metodologia Guia 101 propõe a construção da ontologia através de sete passos fundamentais.

O primeiro passo procura determinar o escopo de atuação e o domínio de informação alvo da ontologia. Corresponde na verdade à etapa de *especificação* do Processo de Desenvolvimento de Ontologias. É definido nesse passo qual tipo de informação a ontologia deve fornecer a seus usuários. O segundo passo consiste em buscar por ontologias já existentes, evitando o retrabalho de conceitos já formalizados no mesmo domínio. No próximo passo, é construída uma lista de termos iniciais da ontologia, sem definir a priori as relações entre os termos e as propriedades. Também não é levada em conta a duplicidade de termos ou existência de ambiguidade. No quarto passo, são criadas classes da ontologia com os termos que representam conceitos. Esse processo pode ser feito de forma *top-down*, definindo a priori os conceitos mais genéricos, de forma *bottom-up*, definindo a priori os conceitos mais específicos ou de forma *híbrida*, definindo a priori os conceitos intermediários. O próximo passo separa os termos restantes em propriedades dos conceitos e relações. O sexto passo define as cardinalidades e os tipos de valores que as propriedades podem assumir. O sétimo passo consiste na criação de instâncias dos conceitos, definindo os valores dos atributos.

O Guia 101 é uma abordagem interessante pois facilita a obtenção de uma ontologia a partir de métodos práticos de extração de conhecimento, porém o especialista do domínio é consultado somente ao final de todo o processo. A consulta tardia, que gera problemas, já que erros podem ser propagados ao longo do ciclo de construção da ontologia, dificulta a correção quando não são refeitas todas as etapas. Dessa forma, o Guia recomenda avaliar a ontologia posteriormente, utilizando-a em sistemas ou discutindo-a com especialistas do domínio.

CommonKADS

A metodologia CommonKADS é na verdade uma especificação voltada à engenharia de software estruturada com foco em sistemas de conhecimento. O objetivo principal da metodologia é converter a arte de engenharia de conhecimento em uma disciplina científica real. Apesar de não focar na construção de ontologias, a CommonKADS é mais abrangente pois auxilia desde a análise do problema organizacional até a construção de modelos de conhecimento que podem auxiliar a obtenção de ontologias. A metodologia

oferece uma linguagem de descrição *CML*, que descreve o modelo de conhecimento com declarações textuais. Entretanto, não é uma linguagem totalmente adequada para descrever uma ontologia, pois embora suas primitivas de construção possuam mais expressividade semântica, não são formalmente definidas para permitir o processamento automático da ontologia, dificultando a criação de modelos formais.

A metodologia propõe três passos distintos, cada um composto por um conjunto de atividades, para a construção do modelo de conhecimento (Schreiber 2000). O primeiro passo é chamado de *identificação* e consiste na preparação da especificação do modelo de conhecimento. Nessa etapa o engenheiro busca familiarizar-se com o domínio através da busca de fontes de informação como livros, especialistas, documentos, etc, da construção de um glossário de termos e da análise de cenários possíveis. Além disso, são identificados quais componentes de conhecimento têm potencial para reutilização. A etapa de *especificação* consiste na identificação do modelo de tarefa, definido qual problema deve ser resolvido com o modelo de conhecimento. Nessa fase também é feita a conceitualização inicial do domínio, definido os tipos principais de informação presentes no domínio. É então feita a definição completa do domínio, buscando os conceitos mais adequados para compô-lo. O modelo de conhecimento é construído a partir das partes de conhecimento reutilizáveis (identificadas na fase de *identificação*) juntamente das partes de conhecimento introduzidas pelo engenheiro. Ao final dessa etapa, se obtém a especificação completa do modelo de conhecimento. A terceira e última fase, *refinamento*, consiste em validar o modelo de conhecimento gerado através da simulação em papel ou de um protótipo de sistema de inferência. O modelo gerado deve ser capaz de resolver o problema definido na especificação da tarefa. Caso sejam encontradas inconsistências, o modelo é então refinado e finalizado.

As três fases não precisam ser executadas em ordem e podem se interseccionar. Muitas vezes várias repetições são necessárias para se obter um modelo de qualidade (Schreiber 2000).

On-to-Knowledge

Essa metodologia foca em guiar a construção e validação de ontologias voltadas a aplicações práticas (York Sure et al. 1999) (D. Fensel et al. 2000). Na verdade ela implementa uma extensão da metodologia CommonKADS, buscando especificar o processo de desenvolvimento de ontologias através da aquisição, representação e acesso do conhecimento.

O processo é composto de diversas fases. A primeira fase apenas estuda a *viabilidade* da construção da ontologia, verificando os objetivos finais. Na fase seguinte, *kick-off*, são especificados os requisitos da ontologia, produzindo-se um documento que indica o objetivo da ontologia, o domínio de informação, o escopo da abordagem, fontes de conhecimento que darão embasamento à engenharia do conhecimento, possíveis aplicações da ontologia e ontologias existentes que possam ser reutilizadas. A fase de *refinamento* divide-se em construir a taxonomia básica dos termos da ontologia (adquiridos na etapa anterior), seguida da aquisição de conhecimento dos especialistas, baseada na taxonomia gerada e construindo uma ontologia já no nível epistemológico (Lorenzatti 2009).

A fase seguinte, *representação*, consiste na especificação da ontologia em uma linguagem de representação formal como RDF ou OWL. A fase de *avaliação* busca validar se a ontologia desenvolvida é verdadeiramente útil. O engenheiro de conhecimento compara a ontologia construída com os requisitos definidos na fase de *refinamento* e testa a ontologia de forma prática em uma aplicação real. Podem ocorrer diversas repetições da fase de *avaliação* e *refinamento* até que a ontologia atinja a qualidade definida na fase de *especificação*. A última fase, de *manutenção*, possui atividades envolvidas em manter a ontologia válida, buscando a atualização da mesma sempre que algum requisito de especificação mudar. A Figura 3.1 exibe todas as etapas envolvidas no processo de desenvolvimento de ontologias utilizando a metodologia On-to-Knowledge.

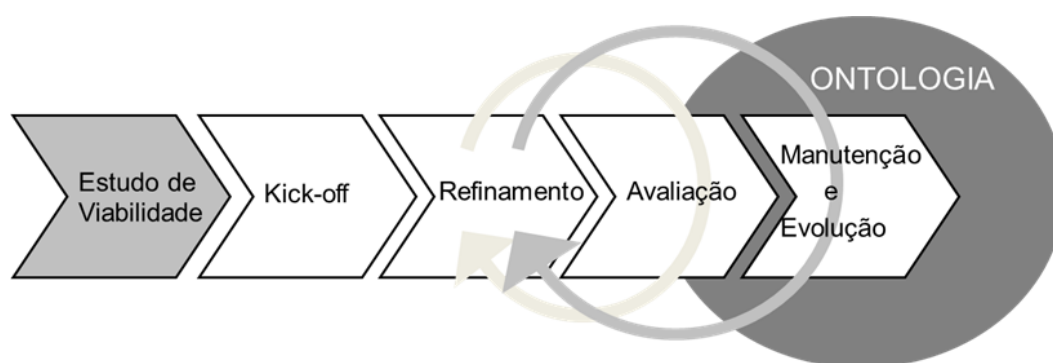


Figura 3.1 Etapas da metodologia On-to-Knowledge, adaptado de York Sure et al. (1999).

TOVE

A metodologia TOVE (Fox 1992) propõe a engenharia de ontologias baseada fortemente em cenários de motivação, buscando extrair ontologias a partir de problemas reais. São definidas a seguintes etapas a serem executadas durante o processo (Jones et al. 1998):

(1) *cenários de motivação*: busca encontrar os problemas existentes em determinada organização, que geralmente são armazenados como registros históricos ou exemplos.

(2) *questões informais de competência*: definição de requisitos da ontologia baseados na etapa (1), descritos como questões de competência que a ontologia deve ser capaz de responder. Isso pode ser visto como uma validação do compromisso ontológico estabelecido na etapa (1).

(3) *especificação da terminologia*: os conceitos, atributos e relações da ontologia são formalmente especificados utilizando lógica de primeira ordem.

(4) *questões formais de competência*: os requisitos da ontologia são formalizados em termos da terminologia definida na etapa (2).

(5) *especificação de axiomas*: axiomas são regras que definem a semântica do domínio, especificando as definições dos termos e as restrições impostas para sua interpretação. Também são formalizados em lógica de primeira ordem, guiados pelos requisitos formais definidos na etapa (4).

(6) *teoremas de completude*: é uma etapa de avaliação onde se define sobre quais condições a ontologia é competente para solucionar as questões de competência definidas em (4) com completude.

A metodologia TOVE enfatiza a avaliação da ontologia, principalmente por definir os teoremas de completude, permitindo a avaliação do compromisso ontológico e da ontologia em si, permitindo identificar necessidades de manutenção ou de extensão da ontologia.

METHONTOLOGY

A metodologia METHONTOLOGY busca definições dos processos de desenvolvimento de software para os processos de desenvolvimento de ontologias. Definida em (Fernandez et al. 1997), a metodologia permite criar ontologias tanto a partir de rascunhos, quanto a partir de ontologias reutilizáveis ou a partir de um processo de reengenharia dessas ontologias. A metodologia inclui a identificação do *processo de desenvolvimento da ontologia*, um *ciclo de vida* baseado na evolução de protótipos e *técnicas* definidas para executar cada atividade. O *processo de desenvolvimento da ontologia* identifica quais tarefas devem ser executadas em cada uma das atividades envolvidas na construção de ontologias (agendamento, controle, garantia de qualidade, especificação, aquisição de conhecimento, conceitualização, integração, formalização, implementação, avaliação, manutenção, documentação e gerência de configuração). O *ciclo de vida* identifica os estágios pelos quais a ontologia passa durante o processo de desenvolvimento, identificando as interdependências entre eles. As *técnicas* utilizadas em cada atividade também são descritas pela metodologia. (Corcho et al. 2003)

As principais atividades envolvidas no processo descrito pela metodologia são descritas a seguir (Jones et al. 1998):

(1) *especificação*: identificar o propósito da ontologia, incluindo os usuários alvo, cenários de uso, grau de formalidade e o escopo da ontologia, incluindo os termos a serem representados, suas características e o nível de granularidade. A saída dessa atividade é um documento especificando a ontologia em linguagem natural.

(2) *aquisição de conhecimento*: ocorre em paralelo com a etapa (1). Nessa etapa é feita a aquisição de conhecimento de forma livre, através de fontes de conhecimento como entrevistas com especialistas, análise de documentos, etc.

(3) *conceitualização*: é a etapa principal da metodologia, onde os termos do domínio são definidos como conceitos, instâncias, relações e propriedades e cada um é representado utilizando uma representação informal.

(4) *integração*: a fim de obter uniformidade entre as ontologias, as definições presentes em outras ontologias devem ser incorporadas.

(5) *implementação*: a ontologia é representada formalmente por uma linguagem de representação formal.

(6) *avaliação*: etapa muito importante pois avalia se a ontologia está de acordo com o esperado, através de técnicas de validação e verificação.

(7) *documentação*: organização dos documentos oriundos das saídas de outras atividades.

Essa metodologia está fortemente vinculada à engenharia de software, dada a grande quantidade de documentação requerida. Logo, consiste em uma metodologia valiosa pois gera artefatos que podem ser consultados em desenvolvimentos futuros.

OntoKEM

A metodologia OntoKEM (Rautenberg et al. 2008) na verdade é uma ferramenta desenvolvida utilizando definições propostas por mais de uma metodologia para construção de ontologias.

A ferramenta permite a construção de ontologias através do processo de sete etapas proposto pelo Guia 101 (Noy et al. 2001), utilizando as questões de competência propostas pela metodologia TOVE (Fox 1992) e fazendo uso de alguns dos artefatos de documentação propostos pela METHONTOLOGY (Fernandez et al. 1997).

3.2 Desenvolvimento Colaborativo de Ontologias

Compartilhamento é um dos objetivos mais básicos do conceito de ontologia. Attingir o conhecimento consensual em um grupo de pessoas é necessário para garantir que o modelo gerado represente a porção do mundo de acordo com o ponto de vista do grupo, ou seja, não deve haver possibilidade de má interpretação dos conceitos e de suas relações. Algumas aplicações recentes utilizam fortemente o compartilhamento de informações como meio de gerar mais conteúdo, como a Wikipedia (uma enciclopédia digital construída on-line : <http://en.wikipedia.org>). Essa ideia se baseia fortemente em utilizar o espaço da web como um espaço livre para participação. Na verdade, o uso de comportamento emergente é uma tendência conhecida como Web 2.0, onde indivíduos utilizam sua criatividade para compor sistemas baseados em internet e que envolvem principalmente compartilhamento e colaboração (Richards 2009).

A formalização de conhecimento é uma das etapas cruciais do processo de engenharia de conhecimento. Acreditamos que utilizar o fenômeno da colaboração coletiva como ferramenta de apoio à construção conjunta de ontologias é uma evolução em relação à construção de ontologias através de metodologias tradicionais. As metodologias tradicionais são úteis quando se foca em um domínio estático, restrito e bem definido. Sabemos que, hoje, muitos domínios de informação são dinâmicos e se modificam o tempo todo. Dessa forma, o refinamento do conhecimento por usuários de diferentes capacidades e pontos de vista pode ter papel fundamental na obtenção de um modelo de conhecimento de qualidade. O uso conjunto de metodologias para a criação das ontologias de domínio iniciais seguido de uma etapa de colaboração entre os especialistas de domínio talvez traga um resultado mais coerente e que convirja de forma mais acelerada, evitando longos processos de extração de conhecimento.

Nessa visão, um ambiente colaborativo que permita a um grupo de pessoas evoluir um modelo de conhecimento é a ferramenta ideal para a construção de consenso em

domínios dinâmicos. Entretanto, para isso ocorrer, cada membro deve estar ciente dos modos de produção e manipulação de conhecimento (Adamczyk & Twidale 2007).

Idealmente, uma ferramenta desse tipo deve poder ser configurável para cada organização que a utilizar, permitindo que mudanças sejam feitas na ontologia, que fluxos de trabalho e processo sejam utilizados para aquisição de conhecimento e com uma interface de uso que permita interação ao longo do ciclo de vida do sistema.

De acordo com John & Melster (2004), um ambiente para construção compartilhada de modelos de conhecimento deve:

- (1) possuir um repositório compartilhado que possibilite o entendimento comum, o compartilhamento de recursos e o reuso de artefatos;
- (2) permitir que as pessoas criem conteúdo de sua própria maneira, utilizando seus próprios termos e conceitos;
- (3) desenvolver um mapa de conhecimento top-down em forma de uma ontologia ou mapa de conceitos para auxiliar as pessoas a definirem e estruturarem seus próprios conceitos em uma amplitude maior;
- (4) permitir a geração de conhecimento tanto *bottom-up* quanto *top-down* de maneira espontânea e imediata.

Richards (2009) complementou essa lista de requisitos, adicionando:

- (5) suporte a diversos níveis de especialização, visões do conhecimento e direitos de acesso;
- (6) prover um processo de revisão no qual os usuários podem registrar suas aprovações ou rejeições;
- (7) testar e manter registro da consistência entre os elementos do sistema de conhecimento, notificando os usuários quando conflitos surgem;
- (8) ser compatível com uma ampla gama de sistemas e fontes de conhecimento;
- (9) prover um ciclo intuitivo, simples e bem estruturado para manutenção do conhecimento;
- (10) suportar a edição do modelo de conhecimento pelos usuários do domínio e não somente por terceiros, como engenheiros de conhecimento;
- (11) ser configurável, permitindo mudanças na ontologia, no processo de aquisição de conhecimento e na interface do sistema, ao longo de seu ciclo de vida.

3.2.1 Modelo 3C: Comunicação, Coordenação e Cooperação

Groupware é um sistema de computador que suporta um grupo de usuários engajado em uma tarefa ou objetivo comum e que provê uma interface para um ambiente compartilhado. Dessa forma, podemos dizer que um sistema colaborativo para construção de ontologias é um Groupware. Esse tipo de sistema demanda forte interação entre os participantes e geralmente necessita de perfis de usuários bem definidos e um ambiente compartilhado acessível via Internet. Em nossa proposta, construímos uma ferramenta de Groupware focada na formalização de uma ontologia de domínio, através de uma estrutura bem definida de colaboração. Para embasar e formalizar as interações no ambiente colaborativo, utilizamos como base conceitual de modelagem o modelo de colaboração 3C (Fuks et al. 2003).

Para colaborar, os indivíduos devem trocar informações (**comunicar-se**), organizar-se (**coordenar-se**) e operar em conjunto num espaço compartilhado (**cooperar-se**). As trocas ocorridas durante a comunicação geram compromissos que são gerenciados pela coordenação, que por sua vez organiza e dispõe as tarefas que são executadas na cooperação. Ao cooperar os indivíduos têm necessidade de se comunicar para renegociar e para tomar decisões sobre situações não previstas inicialmente. Isto mostra o aspecto cíclico da colaboração. Através da percepção, que é um processo de adquirir informação por meio dos sentidos, um indivíduo se informa sobre o que está acontecendo, sobre o que as outras pessoas estão fazendo e adquire informações necessárias para seu trabalho (Fuks et al. 2003). A Figura 3.2 mostra em detalhes a interação entre os pilares da colaboração, de acordo com o modelo 3C.

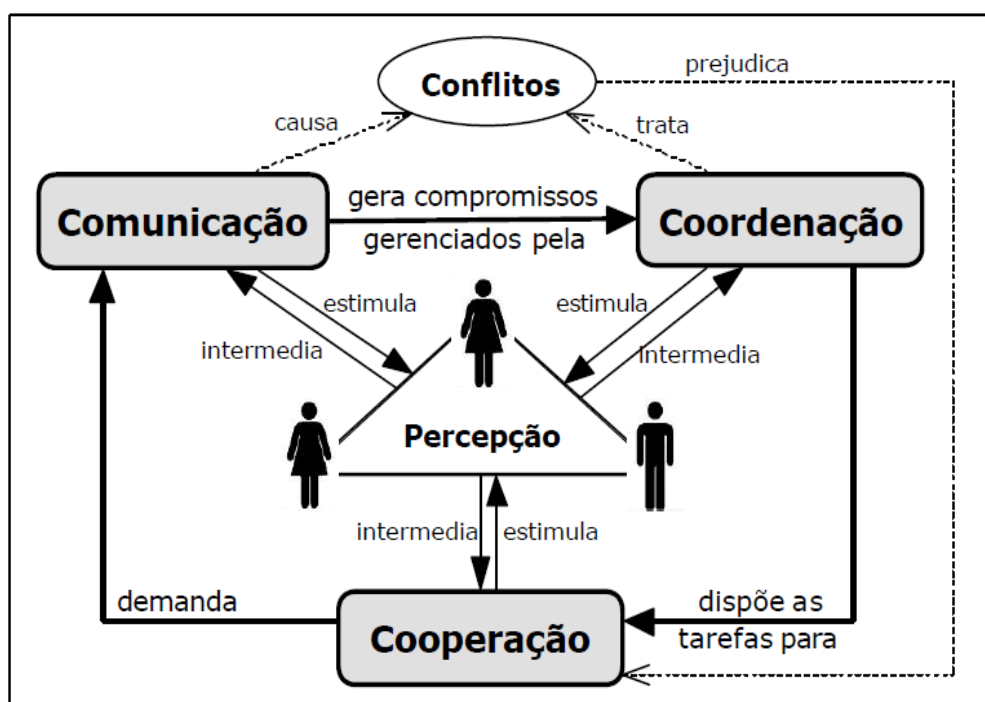


Figura 3.2 Modelo de Colaboração 3C.

Em nossa proposta, entendemos que para estabelecer a colaboração entre os usuários, alguns papéis bem definidos são necessários para manter a organização dos dados e estimular a motivação do grupo no ambiente. Existem três papéis bem definidos: Especialistas do Domínio, Engenheiros de Conhecimento e Visitantes. Especialistas de Domínio devem poder propor mudanças nos elementos da ontologia. Engenheiros de Conhecimento podem efetuar mudanças que tragam mais formalização ao modelo, assim como propor discussões aos especialistas, alertar ou banir usuários com mau comportamento. Visitantes devem poder atuar livremente no ambiente sem efetuar nenhuma mudança no modelo de conhecimento.

Todos os tipos de usuários devem ter permissão para redigir comentários sobre elementos da ontologia (esse tipo de comentário também é chamado de “anotação”), gerando discussões.

Concluindo, o acesso ao ambiente de construção coletiva deve ser disponibilizado aos usuários, propiciando a negociação e o progressivo refinamento das definições e estruturação dos conceitos pela comunidade do projeto até que um modelo consolidado seja obtido.

3.2.2 Ferramentas de Desenvolvimento Colaborativo de Ontologias

Recentemente, o foco das ferramentas para construção de ontologias têm sido a colaboração. Como já vimos, os domínios de informação são muito dinâmicos, necessitando de muitas opiniões para formar um vocabulário comum. A seguir apresentamos algumas das principais ferramentas de construção de ontologias que abordam aspectos de colaboração para construir modelos de conhecimento.

3.2.2.1 Collaborative Protégé – Stanford University

O Protégé é um dos principais editores de ontologias, consistindo de um projeto de código aberto. Seu framework de base de conhecimento foi desenvolvido pela *Stanford Medical Informatics*. Esse framework permite a representação de conhecimento em diversos formalismos como o tradicional *frames* e em linguagens mais recentes como RDF e OWL. Recentemente, o Collaborative Protégé introduziu funcionalidades para permitir a construção colaborativa entre os usuários. O sistema é baseado na arquitetura cliente-servidor, onde diversos usuários podem editar a ontologia simultaneamente (Noy & Tudorache 2008).

O Collaborative Protégé permite que os usuários comentem sobre os componentes da ontologia, discutam mudanças e atinjam consenso de forma interativa através de decisões de modelagem. Isso é controlado através do registro de anotações e mudanças efetuadas sobre os componentes da ontologia. Dentre as funcionalidades fornecidas pelo sistema, podemos citar:

- Anotação de elementos da ontologia como classes, propriedades e instâncias
- Anotação de mudanças da ontologia como criação de classe, deleção, troca de nome, etc.
- Suporte para propostas de mudanças e propostas de votação
- Suporte para filtragem de anotações existentes

- Suporte para chat

Um aspecto interessante da ferramenta é que ela utiliza metadados de ontologias para guiar a interação colaborativa. Em Tudorache et al. (Tudorache et al. 2010) foi proposta a ontologia CHAO (*Change and Annotation Ontology*), a qual fornece construtos para representar os componentes da ontologia de domínio, as mudanças e anotações feitas na ontologia e os tipos distintos de usuários. Dessa forma, a ferramenta cria instâncias dessa metaontologia para representar as mudanças efetuadas sobre a ontologia de domínio.

Entretanto, a ferramenta não dá suporte para fundamentação ontológica ou domínios visuais e para se construir adequadamente uma ontologia utilizando-a é necessário certo conhecimento prévio de OWL ou outro formalismo de representação, pois a interface é orientada aos construtos formais da linguagem.

3.2.2.2 *NeOn Toolkit*

O NeOn Toolkit é um ambiente multi-plataforma para engenharia de ontologias. Baseado em metodologia de construção de ontologias, o ambiente busca prover entendimento e suporte para todo o ciclo de desenvolvimento de uma ontologia. A ferramenta é construída sobre a plataforma Eclipse (plataforma de desenvolvimento Java) e possui arquitetura modular, fornecendo funcionalidades como repositório de ontologias, componentes distribuídos, inferência e colaboração. O foco da ferramenta está voltado ao ciclo de vida da ontologia (Haase et al. 2008).

A comunidade alvo do NeOn Toolkit inclui pesquisadores de web semântica, gestão de conhecimento e áreas correlatas, assim como profissionais de mercado.

O ambiente é fortemente baseado em *plugins*, que agregam funcionalidades ao sistema. O *plugin* de discussão Cicero (Dellschaft et al. 2008) permite que elementos da ontologia e mudanças tenham discussões associadas de forma mais organizada. A ideia principal é fornecer um ambiente que suporte a colaboração em uma comunidade ampla.

3.2.2.3 *WebODE*

O ambiente WebODE (Arpírez et al. 2001) foca em todo o ciclo de vida da ontologia, fornecendo não somente funcionalidades de colaboração, mas também provendo uma infra-estrutura escalável para o desenvolvimento de outras ferramentas de desenvolvimento de ontologias ou aplicações baseadas em ontologias.

O WebODE não é apenas um ambiente de desenvolvimento, mas uma estrutura completa com diversos serviços relacionados à ontologia, focado em suprir carências das ferramentas atuais, buscando ampliar a aceitação do uso de ontologias no meio empresarial.

A ferramenta permite a colaboração no nível de conhecimento, possuindo recursos para evitar problemas de edição concorrente. Entretanto, a colaboração não é o foco do ambiente e é abordada de forma parcial, não permitindo a visualização do histórico de colaboração ou o uso de anotações vinculadas à ontologia. O modelo de conhecimento

que embasa a colaboração é baseado na metodologia METHONTOLOGY, descrita em detalhes na seção 4.1.1.

O WebODE permite efetuar um pós-processamento da ontologia, utilizando a metodologia OntoClean para identificar relações de hierarquia (*is-a*) incorretas.

3.2.2.4 *OntoEdit*

O OntoEdit (Y. Sure et al. 2002) é um ambiente de engenharia de ontologias, o qual foca-se em combinar o desenvolvimento de ontologias baseado em metodologias com a capacidade de colaboração e inferência.

De acordo com a metodologia, o OntoEdit se baseia em três passos principais para o desenvolvimento de ontologias: especificação de requisitos, refinamento e avaliação. Os requisitos para a construção da ontologia são coletados para que engenheiros de conhecimento e especialistas de domínio unam-se em um trabalho conjunto para descrever o domínio e o objetivo da ontologia, definir guias de projeto, verificar fontes de conhecimento existentes (livros, artigos, ontologias reutilizáveis, tesouros, etc.), definir usuários potenciais, casos de uso e aplicações que a ontologia pode suportar. Essa fase gera uma descrição semi-formal da ontologia como saída. A fase de refinamento estende essa descrição semi-formal através de diversas iterações, formalizando a ontologia em uma linguagem de representação. Essa fase gera como saída uma ontologia mais madura, já formalizada. Após, a terceira fase, de avaliação, busca avaliar a ontologia de acordo com os requisitos que foram especificados na primeira fase. Essa fase valida se a ontologia desenvolvida é útil ao time de engenheiros de conhecimento e também aos usuários alvo da ontologia ou da aplicação que a utilizará. Essa fase gera como saída a ontologia validada, pronta para entrar em um ambiente produtivo.

Na atual versão do sistema, há suporte para o desenvolvimento colaborativo, o qual busca atender as necessidades oriundas de conflitos gerados pela construção conjunta da ontologia. Os clientes conectam-se com o servidor, o qual possui a versão global da ontologia. Os clientes são imediatamente informados sobre qualquer alteração no modelo global.

Da mesma forma que o WebODE, o OntoEdit também fornece a funcionalidade de pós-processamento da ontologia, utilizando a metodologia OntoClean para identificar relações de hierarquia (*is-a*) incorretas.

3.2.2.5 *Fabrico*

O ambiente Fabrico (Oliveira et al. 2010) foi desenvolvido por pesquisadores da UFRGS e tem o intuito de fornecer um ambiente operacional em Web 2.0 para que palestrantes, pesquisadores e alunos desenvolvam de forma colaborativa a Ciência da Informação através de fragmentos de textos publicados em diversos formatos. Esses fragmentos são organizados através de metadados baseados em ontologias e produzidos por membros da comunidade. Dessa forma, o ambiente pode ser visto como direcionado à Web Semântica.

A ferramenta é composta por diversas funcionalidades, a citar: extrator de fragmentos de texto da web, repositório de documentos, ambiente cooperativo de escrita (wiki),

sistema de folksonomias, sistema de anotação baseado em tesouro e sistema de favoritos. Através dessas ferramentas, os membros da comunidade do Fabrico produzem textos wiki, registram os documentos produzidos, geram registros de favoritos da web, descrevem conteúdo de acordo com um tesouro e definem descritores (palavras-chave ou etiquetas) para esses conteúdos. Entretanto, Oliveira et al. (Oliveira et al. 2010) mostra como o sistema também pode ser utilizado para construção colaborativa de ontologias.

O Fabrico foca na incrementação e disseminação de informação e conhecimento relacionadas a um determinado curso de graduação, permitindo que o mesmo seja formalmente descrito e disseminado através da construção colaborativa de informações. A ferramenta é, portanto, um ambiente colaborativo que busca categorizar a informação através de metadados. Dessa forma, como a saída da etapa de colaboração é informação estruturada de acordo com metadados, caso esses metadados sejam construídos de forma a representar uma ontologia, o Fabrico pode ser visto como um ambiente para a criação de ontologias a partir de folksonomias.

3.2.3 Avaliação das Ferramentas para Construção Colaborativa de Ontologias

Para possibilitar a análise das ferramentas, foram elencados critérios de comparação bem definidos, assumindo a necessidade anteriormente citada de possibilitar a construção colaborativa de ontologias utilizando construtos de fundamentação ontológica e componentes pictóricos para domínios visuais. Dessa forma, elencamos critérios relacionados a funcionalidades de colaboração, funcionalidades de sistemas web, suporte a fundamentação ontológica e suporte a domínios visuais:

- a) Discussão Estruturada (Anotações)**
Possibilidade dos usuários visualizarem comentários de outros e de discutirem entre si aspectos relevantes à colaboração, formando opiniões e motivando novas mudanças na ontologia.
- b) Histórico de Colaboração**
Armazenamento e consulta do histórico de mudanças efetuadas na ontologia de domínio,
- c) Suporte à OWL**
Possibilidade de importar e exportar a ontologia de domínio em OWL, possibilitando a troca de dados com outro sistema.
- d) Suporte à Inferência**
Mecanismos para efetuar raciocínio automatizado sobre a ontologia de domínio, através das restrições de significado inseridas pelos usuários.
- e) Detecção de Inconsistências**
Possibilidade de detectar conflitos e inconsistências na ontologia de domínio.
- f) Perfis de Usuários Distintos**

Suporte a grupos de usuários com diferentes características e permissões de acesso às funcionalidades do sistema.

g) Interface Intuitiva Web

Interface de uso do sistema com usabilidade adequada para que seja utilizada tanto por usuários experientes quanto por usuários leigos.

h) Acesso Centralizado via Navegador Web

Acesso ao sistema único através de um navegador web e não através da instalação de um software *desktop*.

i) Possibilita utilização sem conhecimento de ontologia formal

Capacidade dos usuários de utilizarem a ferramenta, alterando a ontologia de domínio mesmo sem conhecimentos prévios de formalização ontológica, ou seja, a interface deve utilizar termos leigos e de fácil entendimento pelo usuário.

j) Uso em Domínios Visuais

Capacidade da construção de ontologias para domínios visuais, com adição de ícones e fotografias vinculados aos elementos da ontologia.

k) Uso da Metodologia OntoClean

Utilização da metodologia OntoClean para validação taxonômica e classificação de metapropriedades ontológicas, dentro do contexto da colaboração.

l) Uso de Fundamentação Ontológica

Utilização de primitivas de fundamentação ontológica para enriquecer a classificação de conceitos, propriedades e relações.

As ferramentas foram classificadas de acordo com o suporte que provém para cada um dos critérios. Ferramentas que implementam funcionalidades suficientes para atender ao critério foram classificadas com “SIM”, enquanto ferramentas que não possuem ou possuem funcionalidades insuficientes para atender ao critério foram classificadas com “NÃO”. O quadro comparativo das ferramentas descritas neste capítulo é apresentado na Tabela 3.1.

Tabela 3.1 Quadro comparativo das ferramentas de construção colaborativa de ontologias.

Critério	Fabrico	OntoEdit	NeOn Toolkit	Collaborative Protégé	WebODE
Discussão Estruturada (Anotações)	NÃO	NÃO	SIM	SIM	NÃO
Histórico de Colaboração	NÃO	NÃO	NÃO	SIM	NÃO

Suporte à OWL	NÃO	NÃO	SIM	SIM	SIM
Suporte à Inferência	NÃO	SIM	SIM	SIM	SIM
Deteção de Inconsistências	NÃO	SIM	SIM	SIM	SIM
Perfis de Usuários Distintos	NÃO	NÃO	NÃO	NÃO	NÃO
Interface Intuitiva Web	SIM	NÃO	NÃO	SIM	NÃO
Acesso Centralizado via Navegador Web	SIM	SIM	NÃO	SIM	SIM
Possibilita utilização sem conhecimento de ontologia formal	SIM	SIM	SIM	NÃO	NÃO
Uso em Domínios Visuais	NÃO	NÃO	NÃO	NÃO	NÃO
Uso da Metodologia OntoClean	NÃO	SIM	NÃO	NÃO	SIM
Uso de Fundamentação Ontológica	NÃO	NÃO	NÃO	NÃO	NÃO

Analisando as ferramentas disponíveis, chegamos à conclusão que a maior parte delas não fornece construtos suficientes para uma captura semântica mais completa, ou seja, não possuem precisão suficiente para especificar o conhecimento em sua completude e podem gerar modelos incompletos (como visto na seção 2.2.1). Essa carência de construtos dificulta também a discussão de significado, pois muitas vezes os conceitos não são modelados de forma a evitar ambiguidade e redundância. Além disso, muitas dessas ferramentas também são orientadas às linguagens formais de representação, como a OWL, tanto em implementação quanto em interface. Isso dificulta o uso da ferramenta por usuários que não possuem esse conhecimento técnico.

Pesquisas feitas na área de ontologias colaborativas não somente testemunharam o fato de que a autoria colaborativa de ontologias fortifica o processo de engenharia de ontologias, mas também indicaram que o desenvolvimento e aperfeiçoamento colaborativos de ontologias não é bem atendido por nenhuma das ferramentas atuais de autoria de ontologias (Mangione et al. 2011). Além disso, as ferramentas atuais também não contemplam Fundamentação Ontológica ou Domínios Visuais.

Um aspecto importante é que as ferramentas não oferecem suporte adequado ao problema da *escolha ontológica*, ou seja, qual a melhor primitiva para representar o conceito em análise de modo a criar modelos melhor ancorados na realidade. A fraca correção, laconicidade e lucidez das linguagens de representação de ontologias permitem a produção de uma diversidade de especificações de um mesmo modelo conceitual e consequentes interpretações distintas desses modelos por diferentes usuários da informação. A utilização de metadados baseados em ontologias de fundamentação orienta a escolha das representações e permite a redução das variantes na especificação dos modelos. Considera-se que modelos mais uniformes sejam de, igual forma, mais uniformemente

interpretados. Nenhuma das ferramentas descritas nessa seção trata a orientação à escolha ontológica na abordagem.

Da mesma forma, o compartilhamento de vocabulários está cada vez mais associado à utilização de imagens e construtos visuais, graças a disseminação de equipamentos de captura de imagens e distribuição delas em meios digitais. A própria construção do conceito na mente do emissor é fortemente embasada em conhecimento visual, cujo tratamento ainda é insipiente pelas ferramentas de construção de ontologias, como as acima descritas.

Sistemas interativos não devem ser limitados à sua utilidade, mas também garantir que a mesma é provida de uma maneira ágil. Dois aspectos devem ser levados em consideração: se não for provido um ambiente colaborativo para desenvolvimento de ontologias, as ontologias produzidas jamais serão produto de um processo social. Por outro lado, se a ferramenta não provê boa usabilidade, o processo de engenharia de ontologias não pode esperar espalhar suas ideias para um público não-especialista (Henke 2006).

Após essa análise, foi possível direcionamos o foco deste trabalho para resolver um dos principais problemas apresentados pelas soluções atuais: a carência de um ambiente colaborativo intuitivo com construtos adequados para uma representação completa das ontologias de domínio melhor ancoradas na realidade e que considerem também a captura de conhecimento visual.

No próximo capítulo apresentamos a arquitetura do modelo de conhecimento baseado em metadados, o qual embasa o ambiente colaborativo através de metaontologias que formalizam a representação dos construtos ontológicos e a estrutura da colaboração.

4 MODELAGEM DA FERRAMENTA PARA CONSTRUÇÃO COLABORATIVA DE ONTOLOGIAS

A principal motivação deste trabalho surgiu da existência de grande dificuldade no compartilhamento de conhecimento formal entre especialistas. As distâncias geográficas e barreiras de entendimento muitas vezes impedem a obtenção de consenso sobre determinado domínio, mesmo após uma extração de conhecimento eficiente. O desenvolvimento de ontologias consome tempo, é caro e envolve profissionais altamente especializados. Logo, o processo precisa ser otimizado de modo a acompanhar a necessidade de baixos custos de desenvolvimento e a lidar com a quantidade de informações com as quais vivemos hoje.

Com base nisso, notamos uma carência na área de Engenharia de Conhecimento por uma ferramenta para construção colaborativa de ontologias com suporte à domínios visuais e fundamentação ontológica, a qual permitisse a especialistas e engenheiros de conhecimento cooperar de forma intuitiva sobre determinado domínio de informação na web, gerando ontologias de domínio com representação adequada. Ou seja, desenvolvemos um sistema online para desenvolvimento colaborativo de ontologias, com foco em permitir especificar ontologias com mais expressividade e precisão, através de construtos mais bem organizados através da fundamentação ontológica e conhecimento visual.

Em nosso trabalho, os componentes ontológicos básicos (conceito, propriedade e relação), foram estendidos de acordo com a Ontologia de Fundamentação UFO-A, a qual possibilita especializá-los com construtos mais ricos, aumentando a expressividade do modelo, diminuindo ambiguidades e facilitando a classificação taxonômica. Adicionalmente, o uso de construtos e componentes visuais permite que especialistas ancorem o conhecimento simbólico de forma pictórica, clarificando e facilitando o compartilhamento da informação entre os usuários do sistema, principalmente nos domínios visuais como a Geologia e a Medicina.

Para obter um melhor resultado, buscamos também aspectos de sistemas de colaboração ou *groupwares*, para construir uma ferramenta realmente eficaz no processo de construção de ontologias de domínio através de uma interface de fácil uso e entendimento pelos usuários, orientada a ícones e botões intuitivos.

Com base nesses requisitos, modelamos conceitualmente um ambiente de colaboração baseado em metadados na forma de ontologias de mais alto nível para definir a estrutura da colaboração (referenciadas aqui como metaontologias), onde a expressividade e a precisão desejadas são atendidas através do uso de construtos de ontologias de fundamentação e conhecimento visual.

4.1 Estrutura de Metadados

Metadados geralmente são descritos na literatura como informação sobre informação, mas na verdade são informações estruturadas que descrevem, explicam, encontram ou ajudam na recuperação ou gerenciamento de algum recurso de informação (Press 2004). Um registro de metadados contém elementos pré-definidos que representam atributos específicos de algum recurso ativo, com cada atributo possuindo um ou mais valores. Metadados provêm uma maneira sistemática de categorizar informação através de declarações informativas sobre determinado tipo de dados. Entretanto, para recuperar de forma eficiente essa informação, os dados precisam ser armazenados de maneira correta, permitindo controles para redução de ambiguidades ou redundâncias. Dessa forma, podemos utilizar vocabulários controlados, taxonomias, tesouros e ontologias.

Neste trabalho, construímos um modelo de conhecimento em uma camada de metadados, a qual embasa toda a aplicação colaborativa, definindo os elementos manipuláveis pelo sistema e pelos usuários. O modelo de conhecimento tem dois objetivos distintos: representação da ontologia de domínio e representação da colaboração. O modelo proposto especifica metaconstrutos que possibilitam ao sistema entender os componentes da ontologia de domínio, manipulando-os corretamente e registrando as mudanças e discussões ocorridas durante o processo de colaboração.

O modelo de conhecimento utiliza os conceitos de duas ontologias desenvolvidas para permitir a manipulação de objetos ontológicos em um ambiente colaborativo:

- **Metaontologia de Representação:** especifica a estrutura dos componentes da ontologia de domínio.

- **Metaontologia de Colaboração:** especifica a estrutura dos componentes de colaboração, a fim de obter uma conceitualização consensual e compartilhada por uma comunidade.

Essas metaontologias estruturam a manipulação dos objetos ontológicos da ontologia de domínio em colaboração pelos especialistas. A ontologia de domínio especifica a estrutura do domínio na forma de conceitos, propriedades e relações, com construtos precisos. Essa ontologia é o artefato manipulável pelos usuários e toda mudança sobre ela é feita através da estrutura definida pelas metaontologias.

As duas metaontologias fornecem construtos de suporte à fundamentação ontológica e conhecimento visual, de forma a atingir maior completude nas ontologias de domínio geradas pelo sistema, buscando aproximar ainda mais os modelos gerados da realidade modelada. A seguir, o uso das metaontologias é explicado em detalhes.

4.2 Modelagem da Colaboração com Metaontologias

A modelagem ontológica foca em identificar os conceitos de um domínio e especificá-los em uma ontologia utilizando uma linguagem de especificação baseada em categorias ontológicas independentes de domínio (também chamadas de ontologias *upper-level* ou ontologias de topo) (Guizzardi et al. 2004).

Nesse trabalho, introduzimos uma ontologia de topo com metadados independentes de domínio para especificar os componentes da ontologia de domínio e os eventos de colaboração. Nossa principal contribuição foi desenvolver uma camada de metadados que permite construir uma ontologia de domínio de uma forma independente de linguagem de representação, porém incorporando elementos da ontologia de fundamentação UFO-A e fornecendo suporte a domínios visuais. Dessa forma, construtos específicos foram criados para fornecer mais precisão à ontologia de domínio, os quais foram construídos através dos questionamentos a seguir:

- a) quais componentes da ontologia devem ser representados e com qual estrutura?
- b) quais desses componentes da ontologia podem ser objetos de colaboração pelos usuários?
- c) quais eventos de colaboração devem ser representados e com qual estrutura?

Um evento de colaboração ocorre a partir da interação do usuário com o sistema, podendo ser uma mudança em algum componente da ontologia de domínio ou uma anotação (comentário, pergunta, resposta, etc.) sobre algum componente da ontologia de domínio. O conjunto de mudanças é armazenado de maneira estruturada para permitir o mapeamento da evolução da ontologia e a extração de modelos intermediários durante as diversas interações que são efetuadas no processo de colaboração para obter-se uma conceitualização compartilhada.

O sistema colaborativo é estruturado em três camadas distintas: fundamentação, metadados e dados. A camada de fundamentação não é implementada, porém serve como base ontológica para as demais camadas. Como vimos anteriormente, o sistema trabalha diretamente com três ontologias (duas metaontologias e a ontologia de domínio). A ontologia de domínio é compartilhada diretamente pelos usuários e encontra-se na camada de dados, pois contém os dados inseridos pela comunidade atuante na colaboração. As metaontologias de Representação e Colaboração fazem parte da camada de metadados, estruturando a ontologia de domínio e a colaboração, respectivamente. Ambas possuem construtos precisos e expressivos pois se baseiam na Ontologia de Fundamentação UFO-A (Guizzardi & Wagner 2004), a qual faz parte da camada de fundamentação. A estrutura completa das ontologias envolvidas no processo de colaboração pode ser vista na Figura 4.1.

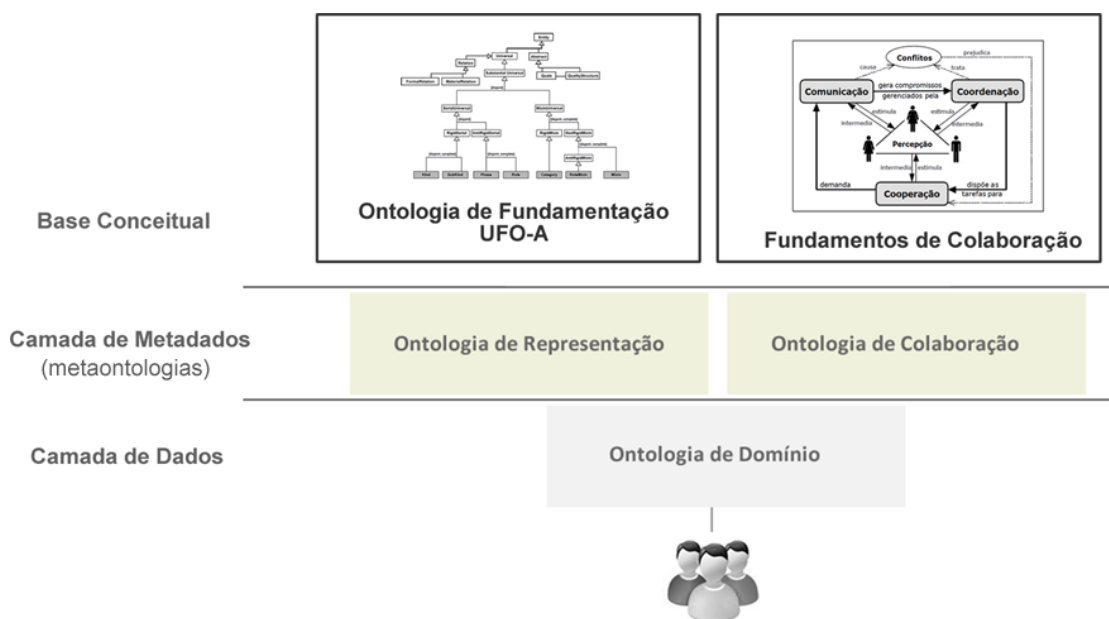


Figura 4.1 Estrutura das ontologias do sistema.

É importante ressaltar que o usuário do sistema manipula diretamente somente a ontologia de domínio, ou seja, as meta-ontologias de Representação e Colaboração não são objeto de colaboração, pois são utilizadas para fins de organização e estruturação da informação, atuando apenas na camada de base do sistema.

A metaontologia de Representação define o que pode ser criado na ontologia de domínio, enquanto a Ontologia de Colaboração define o que pode ser mudado na ontologia de domínio. Em outras palavras, a O.R. define componentes ontológicos como “*Conceito*”, “*Propriedade*” e “*Relação*” enquanto a O.C. define eventos de colaboração como “*Conceito Criado*”, “*Propriedade Excluída*” e “*Relação Criada*”.

Nessa estrutura, a ontologia de domínio é criada através da instanciação da ontologia de representação, assim como o histórico de colaboração (conjunto de eventos de mudanças efetuadas sobre a ontologia de domínio) é criado através da instanciação da ontologia de colaboração. Por exemplo: a Ontologia de Representação possui um meta-conceito chamado “*Conceito*” enquanto a Ontologia de Colaboração possui um meta-conceito chamado “*Conceito Criado*”. Quando um usuário utiliza o sistema para criar um novo conceito na Ontologia de Domínio, esse conceito deverá ser inserido na Ontologia de Domínio e o evento de criação deverá ser registrado no histórico de colaboração. Isso é feito criando-se uma instância do meta-conceito “*Conceito*” e uma instância do meta-conceito “*Conceito Criado*”, gerando os registros necessários para representação da ontologia de domínio e da atividade de colaboração efetuada. A estrutura de interação entre as ontologias de metadados e suas instâncias pode ser vista na Figura 4.2.

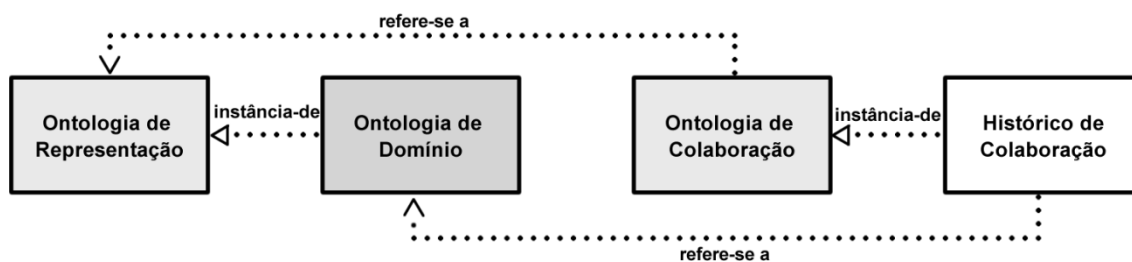


Figura 4.2 Estrutura das ontologias de metadados e suas instâncias.

A seguir, nós detalhamos as metaontologias de Representação e Colaboração.

4.2.1 Ontologia de Representação

Para permitir a definição de ontologias de domínio, propomos uma ontologia de metadados de mais alto nível para distinguir os componentes da ontologia e representá-los de forma que sistemas computacionais possam entender e manipular esses dados. Focamos em uma maneira simplificada de representação através da disponibilização de construtos específicos para cada elemento da ontologia, utilizando metadados para representar o modelo de domínio e distanciando o usuário final de quaisquer formalismos de representação.

Ao lidar com ontologias, busca-se representar seus construtos básicos: *Conceitos*, descrevendo objetos da realidade, *Propriedades*, descrevendo atributos que caracterizam conceitos e *Relações*, descrevendo mapeamentos relacionais entre conceitos. Neste trabalho estamos interessados em expandir o conjunto de construtos básicos, fornecendo construtos mais expressivos e que permitam:

- a representação dos construtos ontológicos básicos;
- a utilização de representações pictoriais, permitindo a representação de domínios visuais (Lorenzatti et al. 2011) através da associação de imagens e ícones a conceitos e propriedades;
- a utilização de metapropriedades de conceitos definidos na metodologia OntoClean (Guarino 1992);
- a utilização de metatipos dos conceitos e relações partonômicas definidos na UFO-A (Guizzardi 2005);

Ao utilizarmos os conceitos e construtos visuais no ambiente, pode-se associar a um conceito **um** ícone ou **uma ou mais** imagens, além da tradicional representação simbólica através de um termo linguístico, como pode ser visto na Figura 4.3. Uma foto não pode ser utilizada por mais de um conceito.

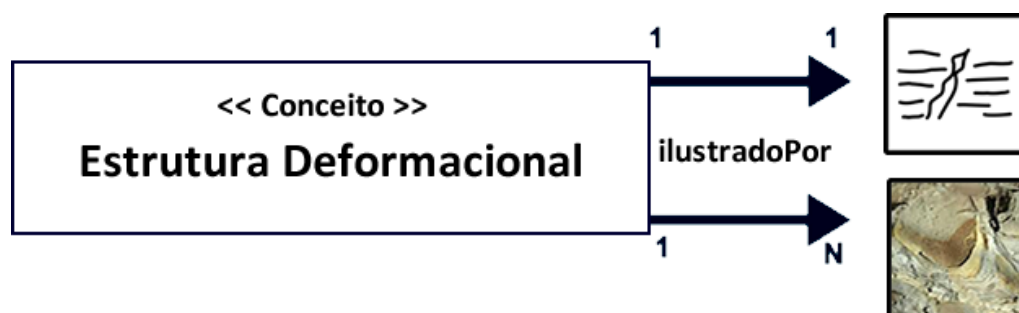


Figura 4.3 Associação de construtos visuais a conceitos.

Propriedades que possuem um conjunto de valores definidos também podem ter associado, para cada valor, um ícone que o represente. Por exemplo, a propriedade *Arredondamento* presente na ontologia de Estratigrafia Sedimentar (Geologia) possui três valores definidos: *sub-arredondado*, *arredondado*, *bem arredondado*. Cada valor pode possuir um ancoramento visual através de um ícone, como visto na Figura 4.4.

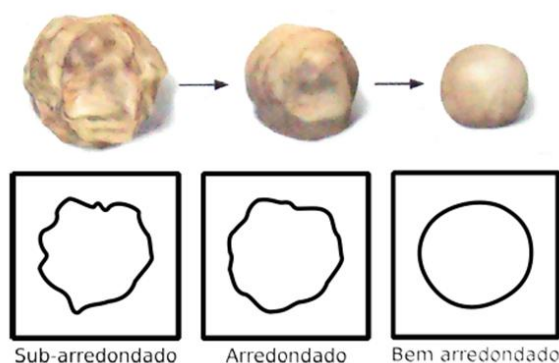


Figura 4.4 Associação de construtos visuais a valores de propriedades.

Adicionalmente, selecionamos construtos da UFO-A procurando equacionar o nível da especialização semântica com a capacidade de interação do usuário com esses construtos e focando no domínio de interesse de Estratigrafia Sedimentar. Dessa forma, alguns construtos como os Universais de Momento não foram utilizados nesse trabalho. Dentre os construtos selecionados estão os metatipos de conceitos definidos como subclasses dos Universais de Substância (objetos concretos da realidade) (*sortal*, *kind*, *subkind*, *quantity*, etc.), tipos de propriedades (*quality universal properties*) e relações partonômicas (*component-of*, *member-of*, etc.). Esses construtos, quando utilizados da forma correta, impõem restrições semânticas ao modelo, auxiliando os usuários a detectar falhas de significado e erros de representação na ontologia de domínio.

A O.R. descreve os componentes estruturantes da ontologia de domínio (Conceito, Propriedade, Relações, etc.), representando-os através de seus conceitos. Portanto, o conceito raiz dessa meta ontologia é *OntologyComponent*, possuindo como sub-conceitos *OntologyConcept*, *OntologyProperty*, *OntologyPropertyValue*, *OntologyRelation*, *OntologyImage* e *OntologyConceptMetaProperty*, os quais representam respecti-

vamente os objetos Conceito, Propriedade, Valor de Propriedade, Relação, Imagem e Metapropriedade de Conceito (oriundas da OntoClean), como visto na Figura 4.5.



Figura 4.5 Especialização do meta-conceito OntologyComponent.

A Ontologia de Representação define como a aplicação interpreta os componentes ontológicos, auxiliando o sistema colaborativo a detectar se os mesmos foram corretamente definidos pelos usuários. Logo, a ontologia de domínio criada pelos usuários é formalizada em termos da ontologia de representação através da instanciação de seus conceitos. Para exemplificar, um conceito de domínio hipotético *Rocha* seria representado como uma instância do meta-conceito *RigidSortal*, uma subclasse do meta-conceito *OntologyConcept* (visto logo a seguir). Da mesma forma, uma propriedade de domínio *Angularidade* seria representada como uma instância do meta-conceito *OntologyProperty*, assim como a relação *compostaPor* seria representada como uma instância do meta-conceito *OntologyRelation*. Os (meta)conceitos da Ontologia de Representação possuem suas próprias propriedades, que precisam ser atribuídas no momento da instanciação.

Assumimos que todos os conceitos da Ontologia de Representação possuem duas propriedades para representar o símbolo linguístico em cada idioma. Elas são implementadas como valores textuais (string): *labelEN* para Inglês e *labelPT* for Português. Mais propriedades podem ser adicionadas para dar suporte a outros idiomas.

Podemos ver os principais conceitos da Ontologia de Representação (sem suas especializações), na Figura 4.6.

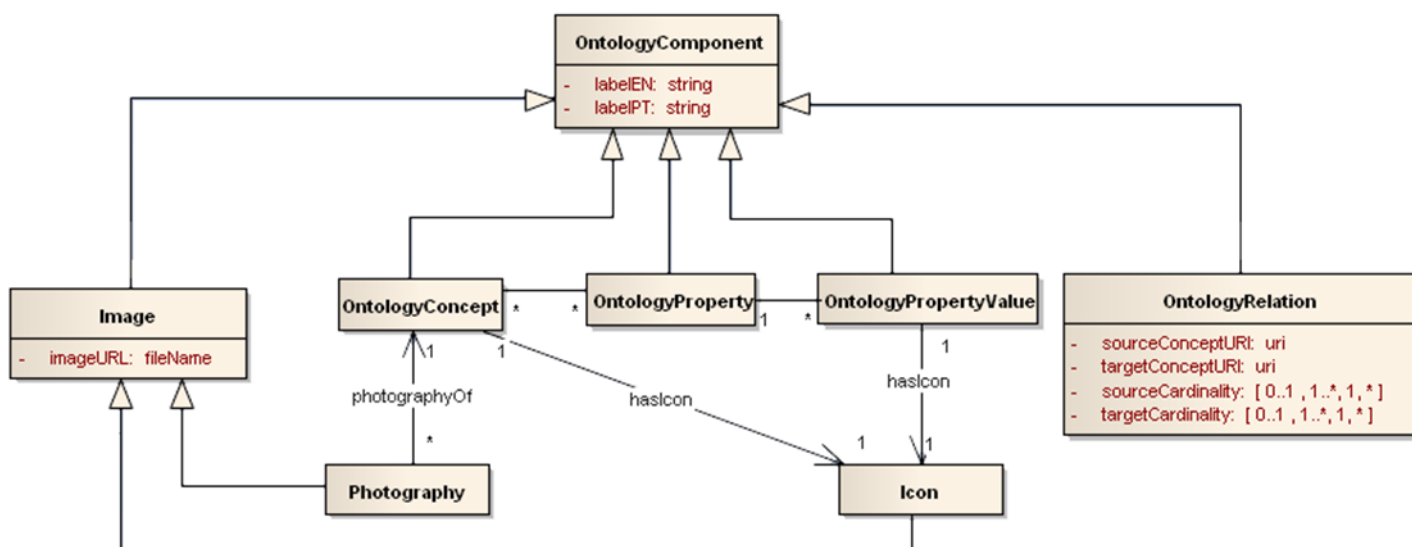


Figura 4.6 Estrutura em alto-nível da Ontologia de Representação.

Para prover suporte visual (como descrito na seção 2.3), utilizamos o meta-conceito *OntologyImage*, o qual é especializado em dois sub-conceitos: *Photography* (para representar fotografias de instâncias e *Icon* (para representar ícones pictoriais simbólicos) (Lorenzatti et al. 2011). A O.R. contém algumas relações que conectam o meta-conceito *OntologyConcept* ao meta-conceito *Icon*, permitindo o vínculo de até um ícone, e ao meta-conceito *Photography*, permitindo o vínculo de fotografias, de acordo com a formalização descrita na seção 3.1. Esses artefatos correspondem à conexão entre um conceito e sua representação pictórica, descrita no triângulo de Ullmann.

Para adicionar fundamentação ontológica ao modelo, alguns dos meta-conceitos da Ontologia de Representação foram especializados em alguns dos construtos propostos na ontologia de fundamentação UFO-A (Guizzardi & Wagner 2004). A seguir, detalhamos como foi feita cada uma dessas especializações.

Neste trabalho focamos em representar apenas as entidades concretas do mundo, pois a ontologia de domínio utilizada para validação (Estratigrafia Sedimentar) representa principalmente esse tipo de objeto. Guizzardi propôs uma hierarquia desse tipo de objeto, chamados de Universais de Substância (Guizzardi 2005). Os tipos desses Universais também são descritos como metatipos de conceitos. Utilizando-se esses metatipos, é possível representar um conceito da ontologia com construtos que representam melhor a sua relação com a realidade. Portanto, o meta-conceito *OntologyConcept* é especializado nos sub-conceitos vistos na Figura 4.7:

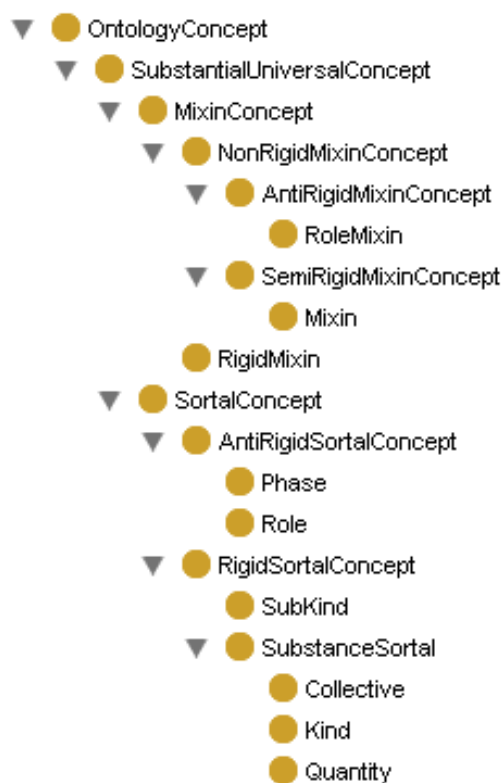


Figura 4.7 Especialização do meta-conceito **OntologyConcept**.

Com visto na seção 2.2, a metodologia OntoClean (Guarino 1992), analisa o significado dos conceitos de um domínio em relação à maneira como eles se relacionam na realidade. A OntoClean inicia o processo de modelagem através da escolha das primitivas que irão representar cada um dos conceitos com base na análise de metapropriedades. As metapropriedades buscam explicitar restrições ontológicas no nível do conhecimento, facilitando a identificação do conceito através de sua *essência*, *identidade*, *unicidade* e *dependência*. O uso das metapropriedades auxilia a criação de modelos mais consistentes, principalmente se forem utilizadas em conjunto com uma ontologia de fundamentação.

Para fins de interação com o usuário, especificamos a metapropriedade *identidade* dividindo-a em duas *carrega identidade* e *provê identidade*. As demais são diretamente mapeadas para os conceitos da Ontologia de Representação. Logo, o meta-conceito *OntologyConceptMetaProperty* foi especializado como visto na Figura 4.8.

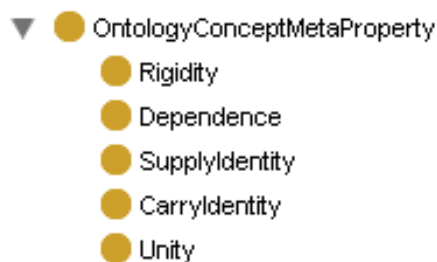


Figura 4.8 Especialização do meta-conceito **OntologyConceptMetaProperty**.

Definimos as metapropriedades e seus valores possíveis como abaixo:

Rigidity (Rigidez) : Rígida (+R), Semi-Rígida (-R), Anti-Rígida (~R)

Dependence (Dependência): Dependente (+D), Não-Dependente (-D)

SupplyIdentity (Provê Identidade): Provê (+O), Não-Provê (-O)

CarryIdentity (Carrega Identidade): Carrega (+I), Não Carrega (-I)

Unity (Unicidade): Unitária (+U), Não-Unitária (-U) , Anti-Unitária (~U)

A fim de proporcionar a construção de uma ontologia mais precisa, as relações estabelecidas na tabela descrita na seção 2.3.2 foram incorporadas ao sistema para que fosse possível inferir o metatipo do conceito a partir da classificação de suas metapropriedades. Dessa forma, o usuário especialista pode interagir diretamente com as metapropriedades, as quais são mais fáceis de classificar do que os metatipos, para aqueles que não possuem conhecimento consolidado sobre fundamentação ontológica. A classificação de metatipos exige um conhecimento prévio e aprofundado da UFO-A. Como os usuários do sistema não terão este conhecimento a priori, o entendimento e classificação das metapropriedades é suficiente para que o sistema atribua os metatipos adequados aos conceitos. Com isso, acreditamos que este aspecto de fundamentação da ontologia de domínio ocorrerá de maneira natural e com pouca ou nenhuma interferência do engenheiro de conhecimento, como visto na Figura 4.9.

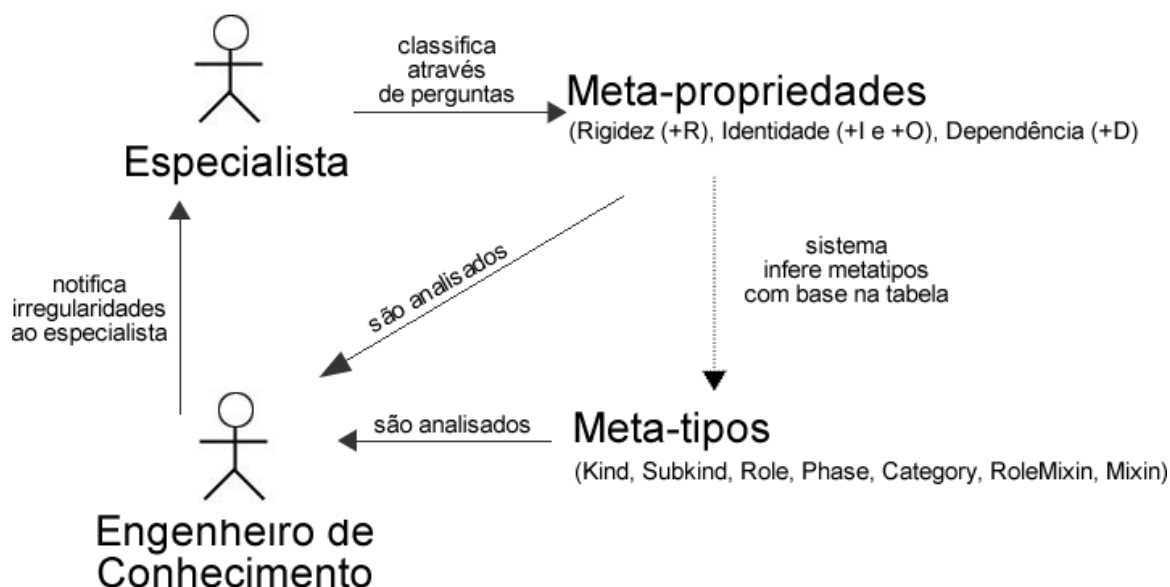


Figura 4.9 Interação do especialista com as metapropriedades para inferir o metatipo do conceito.

Seguindo a ideia de Guizzardi, as propriedades são Universais de Qualidade. Como visto anteriormente, Universais de Qualidade estão associados à Estruturas de Qualidade.

Portanto, o meta-conceito *OntologyProperty* foi especializado em Estrutura de Qualidade (*QualityStructure*) e este especializado em Domínio de Qualidade (*QualityDomain*). As Estruturas de Qualidade representam um conjunto de possíveis valores (Qualia) que a propriedade pode assumir. Cada posição em uma dimensão de qualidade, ou valor possível, é chamado de Quale (Guizzardi 2005). Logo, uma estrutura possui um ou mais Quales. Algumas estruturas podem ter um conjunto de valores bem definido, como ALTO, MÉDIO, BAIXO, outras podem estar relacionadas com um conjunto infinito de valores como os reais positivos (R^+) ou inteiros positivos (I^+). Essa estrutura pode ser melhor vista na Figura 4.10.

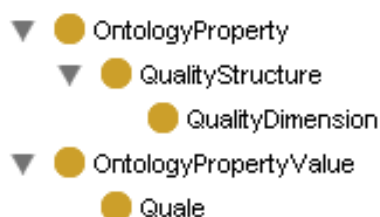


Figura 4.10 Especialização do meta-conceito *OntologyProperty*.

Como vimos, as relações na UFO-A podem ser do tipo Material ou Formal. As relações parte-todo são formais, assim como a relação de hierarquia entre dois conceitos, também chamada de “*subsumption*” ou “*is-a*” também é uma relação formal. As relações parte-todo são classificadas em relação aos tipos de indivíduos que conectam e são divididas em quatro tipos: *subQuantityOf*, a qual relaciona indivíduos que são quantidades (metatipo *Quantity*), *subCollectionOf*, a qual relaciona indivíduos que são coletivos (metatipo *Collective*), *memberOf*, a qual relaciona indivíduos que são complexos funcionais (metatipo *Kind*) ou coletivos e *componentOf*, a qual relaciona indivíduos que são complexos funcionais. As relações materiais são intermediadas por um indivíduo que têm o poder conectar duas entidades. Com base nessas definições, o meta-conceito *OntologyRelation* foi especializado como na Figura 4.11.

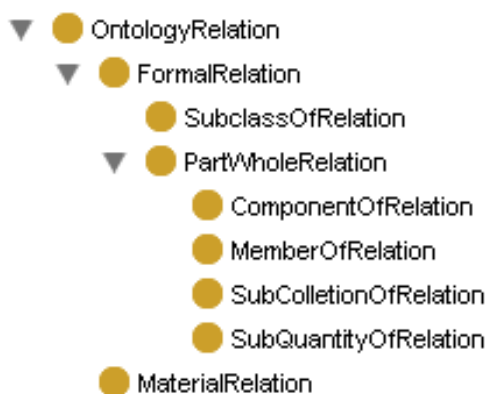


Figura 4.11 Especialização do meta-conceito *OntologyRelation*.

Essas especializações aumentam a expressividade dos modelos gerados, pois geram construtos mais precisos para descrever os objetos da ontologia de domínio. A classificação fundamentada de conceitos, propriedades e relações evita ambiguidades, erros de

modelagem e facilita as escolhas ontológicas que ocorrem a cada passo da colaboração. Através desses construtos mais precisos, o significado real dos objetos ontológicos pode ser inferido a partir de sua classificação pelos usuários, gerando uma ontologia de domínio bem fundamentada ao final do processo de colaboração.

A seguir, descrevemos em detalhes a outra metaontologia, a Ontologia de Colaboração.

4.2.2 Ontologia de Colaboração

A Ontologia de Colaboração (C.O.) define quais atividades de colaboração podem ser feitas na ontologia de domínio. O processo de colaboração é focado no gerenciamento e armazenamento de mudanças feitas sobre conceitos, propriedades ou relações e também em anotações vinculadas aos componentes da ontologia. Através da aplicação web, os especialistas podem criar mudanças ou anotações diretamente sobre o modelo de domínio, adicionando, alterando ou removendo componentes da ontologia. Colaborativamente, eles podem ver o histórico de alterações de outros usuários e discutir sobre elas através das anotações, efetuando novas mudanças quando enquanto necessário até obter-se um modelo de domínio consolidado.

Internamente, a aplicação armazena os eventos de colaboração instanciando os conceitos da Ontologia de Colaboração. Esses conceitos são metadados importantes para definir o conjunto de modificações possíveis na ontologia de domínio. Um evento de mudança (instância da O.C.) possui propriedades que definem o seu significado: *domainComponent* relaciona a mudança a um componente da ontologia de domínio; *autor* armazena o nome de usuário de quem fez a mudança; *date* armazena a data e hora de quando a mudança ocorreu; *value* armazena o novo valor definido pelo usuário para o componente (novo nome, valor de propriedade, etc.); *language* armazena o idioma no qual a propriedade *value* está.

Os (meta)conceitos da O.C. descrevem os eventos de colaboração, os quais são divididos em *Change* (Mudança) e *Annotation* (Anotação).

Podemos ver os principais conceitos da Ontologia de Colaboração (sem suas especializações) na Figura 4.12 e a seguir explicamos em detalhes como foi feita a modelagem de cada um dos eventos de colaboração.

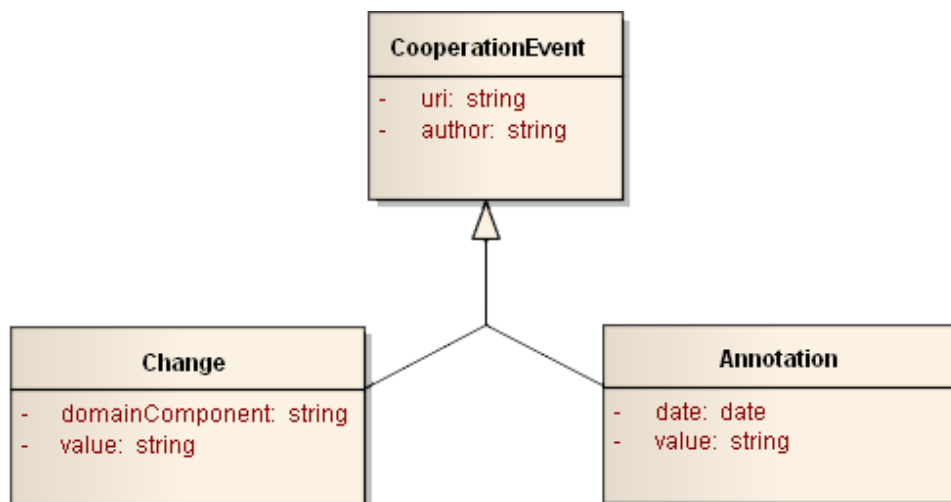


Figura 4.12 Estrutura em alto-nível da Ontologia de Colaboração.

4.2.2.1 Modelagem das Mudanças

Uma mudança pode ocorrer em um conceito, uma propriedade, um valor de propriedade, uma relação ou um termo (símbolo linguístico) de um desses componentes. Portanto, o meta-conceito *Change* foi especializado como mostra a Figura 4.13.

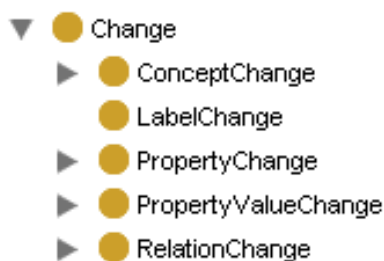


Figura 4.13 Especialização do conceito de mudança *Change*.

Vamos detalhar agora cada uma das mudanças possíveis no sistema. A mudança de termos de componentes é feita utilizando-se o meta-conceito *LabelChange*. As demais mudanças exigem especializações para permitir um rastreamento completo das atividades de colaboração.

Para representar mudanças envolvendo conceitos, especializamos o meta-conceito *ConceptChange* como mostra a Figura 4.14.

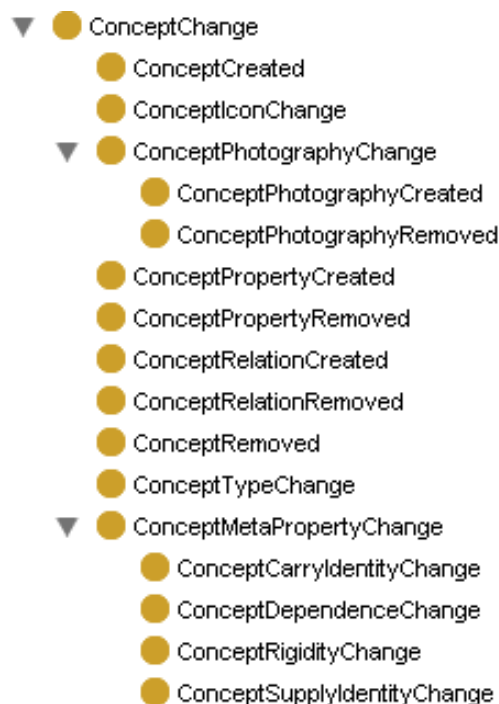


Figura 4.14 Especialização do meta-conceito **ConceptChange** para mudanças em conceitos.

Logo, os eventos de mudanças que são relacionados a algum conceito da ontologia de domínio são:

- ***ConceptCreated*** : criação de um novo conceito ou sub-conceito;
- ***ConceptIconChange*** : alteração do ícone pictórico do conceito;
- ***ConceptPhotographyChange*** : alteração de uma das fotografias de instâncias do conceito, podendo ser a adição de uma nova foto (*ConceptPhotographCreated*) ou a exclusão de uma foto existente (*conceptPhotographyRemoved*);
- ***ConceptPropertyCreated*** : novo vínculo de uma propriedade da ontologia ao conceito em questão;
- ***ConceptPropertyRemoved*** : remoção do vínculo de uma propriedade da ontologia com o conceito em questão;
- ***ConceptRelationCreated*** : novo vínculo de uma relação da ontologia ao conceito em questão;
- ***ConceptRelationRemoved*** : remoção do vínculo de uma relação da ontologia com o conceito em questão;
- ***ConceptRemoved*** : remoção do conceito em questão.
- ***ConceptTypeChange*** : alteração do metatipo do conceito;
- ***ConceptMetaPropertyChange*** : alteração do valor de uma das metapropriedades do conceito, classificadas em *ConceptSupplyIdentityChange*, *ConceptCarryIdentityChange*, *ConceptDependenceChange*, *ConceptRigidityChange* ou *ConceptUnityChange*;

Da mesma forma, as mudanças envolvendo propriedades foram especializadas como mostra a Figura 4.15

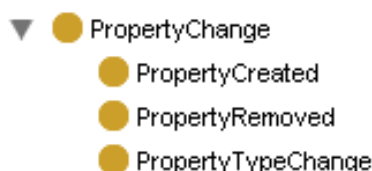


Figura 4.15 Especialização do meta-conceito **PropertyChange** para mudanças em propriedades.

Dessa forma, podemos explicitar as seguintes mudanças sobre propriedades:

- **PropertyCreated** : criação de uma nova propriedade na ontologia;
- **PropertyRemoved** : remoção de uma propriedade da ontologia;
- **PropertyTypeChange** : alteração do tipo de valor da propriedade (domínio), por exemplo, inteiros (int), texto (string), reais (float) ou valores pré-definidos (qualia);

No caso de propriedades que possuem um conjunto de valores finito e pré-definidos, especializamos um tipo de mudança específico para cada valor desse conjunto, como visto na Figura 4.16. Cada valor do conjunto de valores (*Qualia*) da propriedade é também chamado de *Quale* pela UFO-A, como visto nos capítulos anteriores.



Figura 4.16 Especialização do meta-conceito **PropertyValueChange** para mudanças em valores de propriedades.

As seguintes mudanças podem ocorrer com valores de propriedade:

- **PropertyValueCreated** : criação de um novo valor da propriedade;
- **PropertyValueRemoved** : remoção de um valor da propriedade;
- **PropertyValueIconChanged** : alteração do ícone relacionado ao valor da propriedade;

A relação hierárquica *É-UM* é tratada como uma relação formal não-diferenciada. Embora exista farto material na literatura sobre os subtipos com semântica distinta de relações taxonômica, estas não foram tratadas neste trabalho e serão objeto de uma dissertação de mestrado específica.

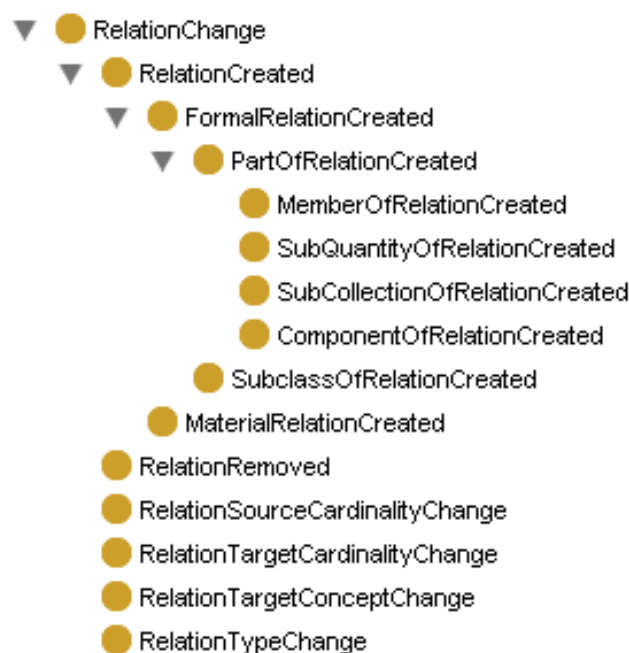


Figura 4.17 Especialização do meta-conceito **RelationChange** para mudanças em relações.

Dessa forma, podemos explicitar as seguintes mudanças sobre relações:

- **RelationCreated** : criação de uma nova relação na ontologia, especializada nos seguintes meta-conceitos, para melhor rastreamento de qual relação foi criada:
 - **FormalRelationCreated** : relações diretas
 - **SubClassOfRelationCreated** : relações de hierarquia
 - **PartOfRelationCreated** : relações partonômicas
 - **MemberOfRelationCreated**
 - **SubQuantityOfRelationCreated**
 - **SubCollectionOfRelationCreated**
 - **ComponentOfRelationCreated**
 - **MaterialRelationCreated** : relações com intermediador
- **RelationRemoved** : remoção de uma propriedade da ontologia;
- **RelationSourceCardinalityChange** : alteração do valor de cardinalidade do conceito de origem [0..1 ; 0..n ; 1..n ; 1];
- **RelationTargetCardinalityChange** : alteração do valor de cardinalidade do conceito de origem [0..1 ; 0..n ; 1..n ; 1];
- **RelationTargetConcept** : alteração do conceito destino da relação;
- **RelationTypeChange** : alteração do tipo da relação, possibilitando alterá-la, por exemplo, de Formal para Material, de MemberOf para SubCollection, etc.

Por fim, temos uma estrutura que possibilita a criação de elementos na ontologia de domínio através da instanciação de meta-conceitos específicos em uma camada de metadados. Essa camada fornece componentes visuais necessários para a colaboração em

domínios visuais e componentes com semântica bem definida necessários para a aplicação de fundamentação ontológica no processo de colaboração. Um exemplo que mostra uma visão completa da arquitetura de colaboração pode ser visto na Figura 4.18.

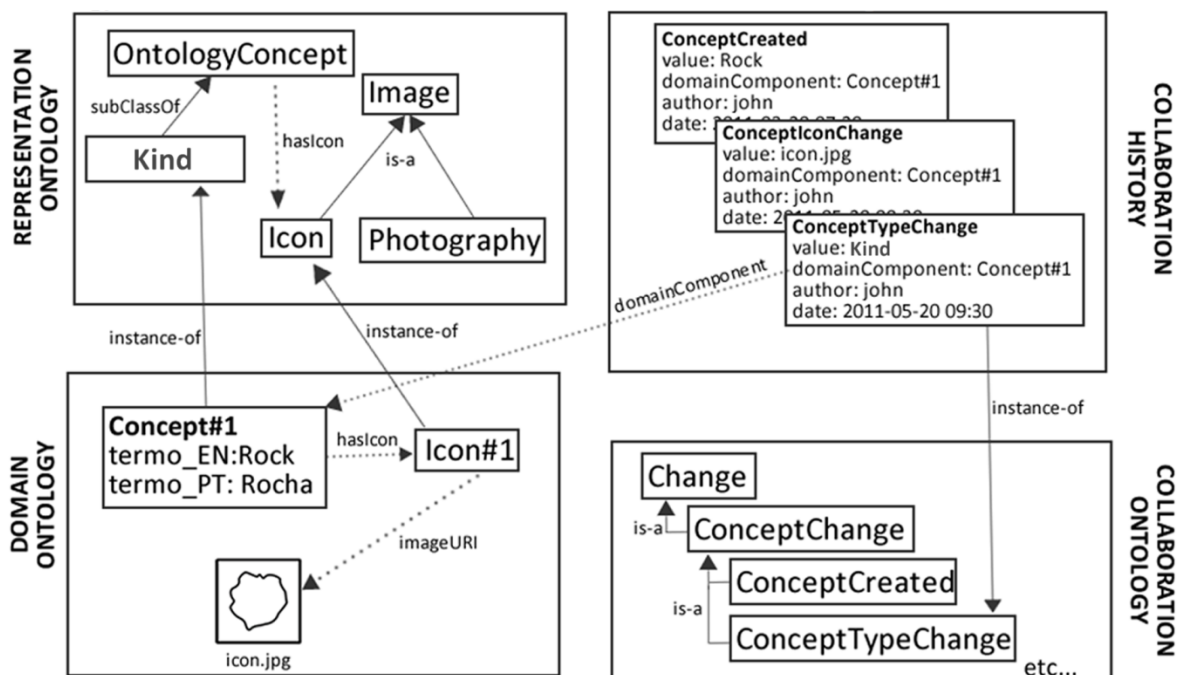


Figura 4.18 Exemplo de estrutura de colaboração com construtos visuais e de fundamentação.

Nessa figura, vemos uma série de três atividades de colaboração: *ConceptCreated*, *ConceptIconChange* e *ConceptTypeChange*. A criação do conceito “Rocha” gerou um registro desse evento no Histórico de Colaboração (nova instância do meta-conceito *ConceptCreated* da Ontologia de Colaboração) e um novo componente na ontologia de domínio, *Concept#1* (o qual é uma instância do meta-conceito *OntologyConcept* da Ontologia de Representação). Após, foi efetuada uma alteração de ícone do conceito, indicando que o novo ícone é “icon.jpg”. Da mesma forma, foi criada uma instância do meta-conceito da O.R. *Icon*, gerando o componente de domínio *Icon#1*, vinculado à imagem “icon.jpg”, e uma instância da O.C. *ConceptIconChange*, gerando o registro desse evento no histórico. Dessa forma, na ontologia de domínio, é criado um relacionamento do conceito com o novo ícone. Por fim, é efetuada uma alteração de metatipo do conceito, através da instanciação do conceito da O.C. *ConceptTypeChange*, o qual indica que o conceito agora não é mais uma instância de *OntologyConcept*, mas sim de uma de suas especializações, o *Kind*. Essa é uma das principais mudanças do ambiente, pois permite que o conceito seja definido em termos de construtos mais precisos como os metatipos definidos na UFO-A. Inicialmente os conceitos podem ser especificados de forma mais genérica, utilizando o construto *OntologyConcept* e posteriormente serem alterados para definir melhor a sua semântica, sendo especializados como um dos possíveis metatipos: *Kind*, *Role*, *Quantity*, *Collective*, etc.

4.2.2.2 Modelagem das Anotações

Como citado anteriormente, os usuários podem discutir entre si e gerar comentários sobre os componentes da ontologia através de *anotações*.

A IBIS (Issue-based Information System) é uma metodologia de argumentação e tem com objetivo suportar a coordenação e o planejamento de processos de decisões políticas, através da identificação, estruturação e definição de questões oriundas dos grupos de resolução de problemas, além de prover informações pertinentes ao discurso (Kunz & Rittel 2010). A metodologia IBIS fornece os seguintes elementos para construir a argumentação:

Questão (*Issue*) : define um novo tópico de discussão de uma perspectiva conceitual.

Idéia (*Idea*) : responde uma questão e faz referência com a formalização na visão conceitual.

Justificativa (*Justification*) : argumentos que defendem uma questão ou uma ideia e são divididos em Avaliação (*Evaluation*), para prover medidas ou experimentos e Exemplo (*Example*), para aumentar a confiança do argumento.

Desafio (*Challenge*) : argumentos contrários à uma questão ou ideia, divididos em Contra-Exemplo (*Counter-Example*), provendo um exemplo contrário e Alternativa (*Alternative*), provendo uma comparação.

Apesar de modelos de argumentação proverem um modelo conceitual para a interação de questões, ideias e argumentos, eles não diferenciam com precisão os diferentes tipos de argumentos que surgem em uma discussão oriunda de um processo colaborativo. Em nosso trabalho, propomos um modelo de comunicação baseado nos conceitos da metodologia IBIS, porém sem o compromisso de utilizar a nomenclatura ou semântica dos elementos existentes. Dessa forma, introduzimos diferentes elementos e relações entre eles para compor a discussão no ambiente colaborativo de construção de ontologias. Entretanto alguns dos elementos de nosso modelo podem ser vistos como equivalentes à alguns elementos da IBIS.

Na aplicação desenvolvida, construímos uma estrutura de anotações buscando aumentar a rastreabilidade das decisões tomadas sobre o modelo conceitual através da criação de novos componentes argumentativos e com vínculos aos componentes da ontologia de domínio ou às mudanças efetuadas sobre a ontologia.

Em nosso modelo de metadados, uma anotação é uma especialização do conceito *CooperationEvent* da ontologia de colaboração. Ela é um comentário efetuado sobre algum artefato do ambiente, seja ele um componente da ontologia, como um conceito ou propriedade, uma mudança ou até mesmo outra anotação.

Para identificar a semântica da discussão e evitar que os argumentos consistam apenas de texto puro relacionados aos artefatos da ontologia, construímos subtipos de anotação específicos, com relações semânticas entre eles, possibilitando extrair mais informação e relacionar os eventos de colaboração de forma mais rica, como mostra a Figura 4.19, na qual a relação *applyTo* descreve o objeto onde a anotação pode ser aplicada. Esses vínculos permitem uma discussão estruturada, vinculando as interações entre si e criando uma rede de conversação inter-relacionada com a ontologia de domínio.

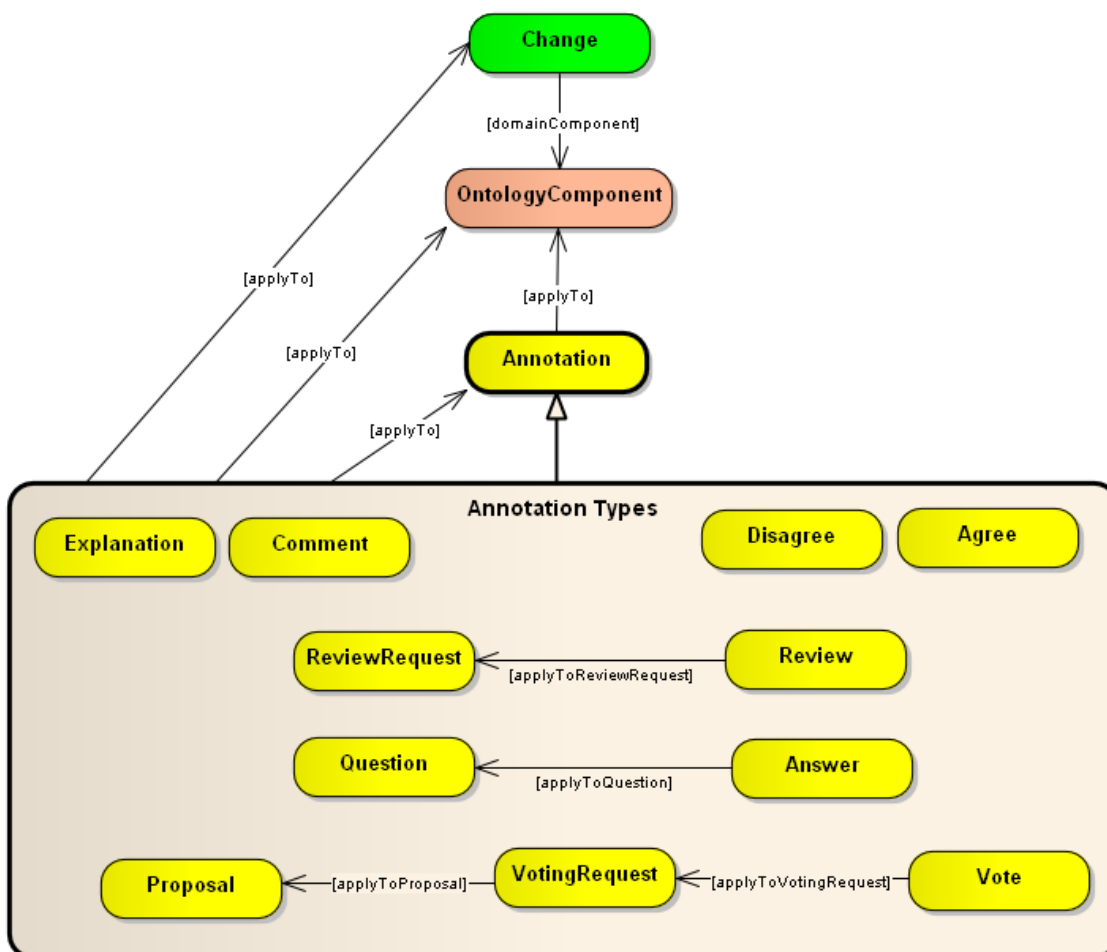


Figura 4.19 Subtipos de anotação com relações semânticas.

Dessa forma, uma anotação pode ser classificada em um dos tipos abaixo, de acordo com o contexto do ambiente:

Explanation (Explicação): comentário opcional inserido ao se justificar alguma mudança na ontologia. Possui relação semântica *applyToChange* relacionando-o a uma mudança. Corresponderia ao elemento *Justification* da IBIS.

Comment (Comentário): comentário livre inserido a qualquer momento. Possui relação semântica *applyTo* relacionando-o a uma mudança, uma anotação ou a um componente da ontologia.

Agree (Concordância): comentário inserido ao se concordar com algum componente da ontologia, mudança ou anotação. Possui relação semântica *applyTo*, relacionando-o a uma mudança, uma anotação ou a um componente da ontologia.

Disagree (Discordância): comentário inserido ao se discordar de algum componente da ontologia, mudança ou anotação. Possui relação semântica *applyTo*, relacionando-o a uma mudança, uma anotação ou a um componente da ontologia. Corresponderia ao elemento *Challenge* da IBIS.

ReviewRequest (Solicitação de Revisão): comentário inserido ao se desejar que outros usuários revisem determinado artefato. Possui relação semântica *applyTo*, relacionando-o a uma mudança, uma anotação ou a um componente da ontologia.

Review (Revisão): comentário inserido ao se revisar determinado artefato, após uma solicitação de revisão. Possui relação semântica *applyToReviewRequest*, relacionando-o a uma anotação do tipo *ReviewRequest*.

Question (Pergunta): comentário inserido ao se perguntar algo a outros usuários. Possui relação semântica *applyTo*, relacionando-o a uma mudança, uma anotação ou a um componente da ontologia. Corresponderia ao elemento *Issue* da IBIS.

Answer (Resposta): comentário inserido ao se responder determinado assunto, após alguma pergunta. Possui relação semântica *applyToQuestion*, relacionando-o a uma anotação do tipo *Question*. Corresponderia ao elemento *Idea* da IBIS.

Proposal (Proposta): comentário inserido ao se propor algo a outros usuários. Possui relação semântica *applyTo*, relacionando-o a uma mudança, uma anotação ou a um componente da ontologia.

VotingRequest (Solicitação de Votação): comentário inserido ao se solicitar que os usuários votem sobre determinada proposta. Possui relação semântica *applyToProposal*, relacionando-o a uma anotação do tipo *Proposal*.

Vote (Voto): comentário inserido ao se votar em determinado assunto, após alguma solicitação de votação. Possui relação semântica *applyToVotingRequest*, relacionando-o a uma anotação do tipo *VotingRequest*.

Esse modelo proposto direciona as discussões, criando fluxos de comunicação bem estruturados e com semântica bem definida, fazendo das anotações um recurso muito importante para complementar a colaboração no desenvolvimento da ontologia de domínio. Os diferentes tipos de anotação, com significados distintos, evita discussões não-estruturadas. Dessa forma, a discussão é direcionada aos tipos utilizados, por exemplo, se um usuário faz uma anotação do tipo Pergunta, outros usuários só poderão gerar anotações do tipo Resposta, assim como quando um usuário cria uma Solicitação de Votação, os demais usuários somente poderão criar anotações do tipo Voto, relacionadas à anotação de solicitação. Os diferentes tipos de anotação e possíveis anotações subsequentes podem ser vistos na Tabela 4.1.

Tabela 4.1 Tipos de anotações e possíveis anotações subsequentes

Anotação	Anotações Subsequentes Possíveis
Pergunta	Responder
Responder	Concordar / Discordar / Perguntar / Propor Mudança
Solicitar Revisão	Revisar / Perguntar

Revisar	Concordar / Discordar / Perguntar / Propor Mudança
Propor Mudança	Solicitar Votação / Concordar / Discordar / Perguntar
Solicitar Votação	Votar / Perguntar
Votar	Concordar / Discordar / Perguntar
Comentar	Concordar / Discordar / Perguntar / Propor Mudança
Concordar	Concordar / Discordar / Perguntar / Propor Mudança
Discordar	Concordar / Discordar / Perguntar / Propor Mudança

Acreditamos que a classificação de anotações em diferentes tipos faz com que cada processo de discussão, ao chegar ao final, gere mudanças na ontologia de domínio, atingindo o objetivo final da colaboração.

4.3 Formalização do Ambiente de Colaboração

Nesta seção, ambiente de colaboração é descrito de forma estrutural a fim de definir os componentes que fazem parte do mesmo e suas relações. O ambiente inclui os tipos de usuários que acessam o sistema e os construtos disponíveis na camada de metadados para representar a ontologia de domínio e a colaboração. Dessa forma, podemos definir formalmente o ambiente de colaboração como a seguir:

$$S = [R, C, D_1..D_n, H_1..H_n, U, P]$$

Assumindo que:

- **R** representa a metaontologia de representação;
- **C** representa a metaontologia de colaboração;
- **D₁..D_n** representa uma ou mais ontologias de domínio;
- **H₁..H_n** representa um ou mais históricos de colaboração, um para cada ontologia de domínio;
- **U** representa o conjunto de tipos de usuário permitidos no sistema;
- **P** representa o conjunto de operações de mudança definidas na ontologia de colaboração **C** que os usuários do tipo **U** tem permissão para efetuar no ambiente. Assim, **P** é o mapeamento $U \times C \rightarrow P$.

Assumindo que as mudanças podem ser feitas por diferentes usuários, algumas mudanças só podem ser efetuadas por determinados perfis. Por exemplo, um usuário do perfil “Engenheiro de Conhecimento” pode efetuar uma mudança que envolve fundamentação ontológica (MudancaDeMetaPropriedade), enquanto um usuário do perfil “Especialista” não. A seguir, os elementos apresentados são detalhados:

$$\begin{aligned}
\mathbf{R} &= [\mathbf{R}_1.. \mathbf{R}_n] \\
\mathbf{C} &= [\mathbf{C}_1.. \mathbf{C}_n] \\
\mathbf{D} &= [\mathbf{I}^{\mathbf{R}}_1.. \mathbf{I}^{\mathbf{R}}_n] \\
\mathbf{H} &= [\mathbf{I}^{\mathbf{C}}_1.. \mathbf{I}^{\mathbf{C}}_n] \\
\mathbf{U} &= [\mathbf{U}^{\text{ENGENHEIRO}}, \mathbf{U}^{\text{ESPECIALISTA}}, \mathbf{U}^{\text{VISITANTE}}] \\
\mathbf{P} &= [\\
&\quad \mathbf{U}^{\text{ENGENHEIRO}} \times \mathbf{C} \text{ MudancaDeMetaPropriedade} \rightarrow \text{true}, \\
&\quad \mathbf{U}^{\text{ENGENHEIRO}} \times \mathbf{C} \text{ MudancaDeMetaTipo} \rightarrow \text{true}, \\
&\quad \mathbf{U}^{\text{ESPECIALISTA}} \times \mathbf{C} \text{ MudancaDeMetaPropriedade} \rightarrow \text{false}, \\
&\quad \mathbf{U}^{\text{ESPECIALISTA}} \times \mathbf{C} \text{ MudancaDeMetaTipo} \rightarrow \text{false}, \\
&\quad \mathbf{U}^{\text{VISITANTE}} \times \mathbf{C} \text{ MudancaDeMetaPropriedade} \rightarrow \text{false}, \\
&\quad \mathbf{U}^{\text{VISITANTE}} \times \mathbf{C} \text{ MudancaDeMetaTipo} \rightarrow \text{false}, \\
&\quad \mathbf{U}^{\text{ESPECIALISTA}} \times \mathbf{C} \text{ CriacaoConceito} \rightarrow \text{true}, \\
&\quad \mathbf{U}^{\text{VISITANTE}} \times \mathbf{C} \text{ CriacaoAnotacao} \rightarrow \text{true}, \\
&\quad \text{etc...} \\
&]
\end{aligned}$$

Onde para a ontologia de representação \mathbf{R} , $\mathbf{R}_1.. \mathbf{R}_n$ são os meta-conceitos que definem os componentes possíveis de serem criados na ontologia de domínio (Conceito, Propriedade, Relação, etc.) e para a ontologia de colaboração \mathbf{C} , $\mathbf{C}_1.. \mathbf{C}_n$ são os eventos de mudança possíveis de serem criados no ambiente de colaboração. Para a ontologia \mathbf{D} , $\mathbf{I}^{\mathbf{R}}_1.. \mathbf{I}^{\mathbf{R}}_n$ representa um ou mais componentes da ontologia de domínio (que são, na verdade, instâncias da ontologia de representação, por isso representados com $\mathbf{I}^{\mathbf{R}}$). O histórico de colaboração \mathbf{H} é o conjunto de instâncias da ontologia de colaboração \mathbf{C} , contendo todos os eventos de colaboração (mudanças e anotações) que ocorreram no processo, representados por $\mathbf{I}^{\mathbf{C}}_1.. \mathbf{I}^{\mathbf{C}}_n$. O conjunto de tipos de usuários contém os tipos Engenheiro de Conhecimento, $\mathbf{U}^{\text{ENGENHEIRO}}$, Especialista, $\mathbf{U}^{\text{ESPECIALISTA}}$ e Visitante, $\mathbf{U}^{\text{VISITANTE}}$. As permissões \mathbf{P} relacionam a combinação de operações de colaboração e os tipos de usuários com a possibilidade de efetuar a operação (valor *true*) ou não (valor *false*).

Para representar os atributos visuais dos conceitos e propriedades ontológicas descritos na seção 2.3, formalizamos esses componentes ontológicos definidos na ontologia de representação como a seguir:

$$\begin{aligned}
\mathbf{R}^{\mathbf{C}} &= (\mathbf{T}_1.. \mathbf{T}_n, \mathbf{P}_0.. \mathbf{P}_1, \mathbf{F}_0.. \mathbf{F}_n) \\
\mathbf{R}^{\mathbf{P}} &= (\mathbf{T}_1.. \mathbf{T}_n, \mathbf{V}_0.. \mathbf{V}_n) \\
\mathbf{V} &= (\mathbf{N}_1.. \mathbf{N}_n, \mathbf{P}_0.. \mathbf{P}_1)
\end{aligned}$$

Onde $\mathbf{R}^{\mathbf{C}}$ são os conceitos ontológicos, definidos pelo conjunto $\mathbf{T}_1.. \mathbf{T}_n$ representando o nome (termos linguísticos) do conceito em cada idioma, $\mathbf{P}_0.. \mathbf{P}_1$ representando no máximo um ícone pictórico vinculado ao conceito e $\mathbf{F}_0.. \mathbf{F}_n$ representando zero ou mais fotografias pictóricas para exemplificar as instâncias do conceito e $\mathbf{R}^{\mathbf{P}}$ são as propriedades ontológicas, definidas pelo conjunto $\mathbf{T}_1.. \mathbf{T}_n$ representando o nome (termos linguísticos) da propriedade em cada idioma e também por zero ou mais valores possíveis $\mathbf{V}_0.. \mathbf{V}_n$ que

a propriedade pode assumir para determinada instância. Cada valor possível V possui termos linguísticos $T_1..T_n$ e no máximo um ícone pictórico $P_0..P_1$.

No próximo capítulo descrevemos em detalhes a implementação e os recursos do ambiente para construção colaborativa de ontologias desenvolvido neste trabalho, a partir dos conceitos introduzidos neste Capítulo, no *Portal Obaitá*.

5 DESCRIÇÃO DO PORTAL OBAITÁ

A ferramenta proposta consiste em um sistema online, acessível pela Internet, focado em atender aos requisitos da proposta desse trabalho, disponibilizando uma ontologia de domínio a todos os usuários (especialistas e engenheiros de conhecimento) de forma que seus componentes possam ser alterados e comentados, ficando o sistema responsável por documentar tudo que ocorre no processo de colaboração em uma interface simples de alta usabilidade.

Na nossa visão, as atividades da metodologia METHONTOLOGY (descrita na seção 3.1.1) expressam bem as etapas existentes no processo de desenvolvimento de ontologias. Dentre as suas etapas, podemos dizer que a ferramenta proposta nesse trabalho se enquadra nas etapas (3) *conceitualização*, (4) *integração* e (5) *implementação*. Idealmente, no processo de colaboração, um modelo inicial da ontologia de domínio é definido e disponibilizado para que os especialistas o critiquem e validem através da ferramenta. Logo, as etapas (1) *especificação* e (2) *aquisição de conhecimento* ocorrem antes do uso do sistema se iniciar. O foco da ferramenta está na etapa (3) *conceitualização*, onde os termos dos conceitos são escolhidos e organizados em uma taxonomia, com adição de relações e propriedades. As etapas (4) *integração*, (5) *implementação*, (6) *avaliação* e (7) *documentação* ocorrem após a etapa de colaboração e não são abordadas pela ferramenta. A Figura 5.1 mostra as etapas da metodologia METHONTOLOGY, destacando a etapa na qual a ferramenta desse trabalho concentra seu foco.

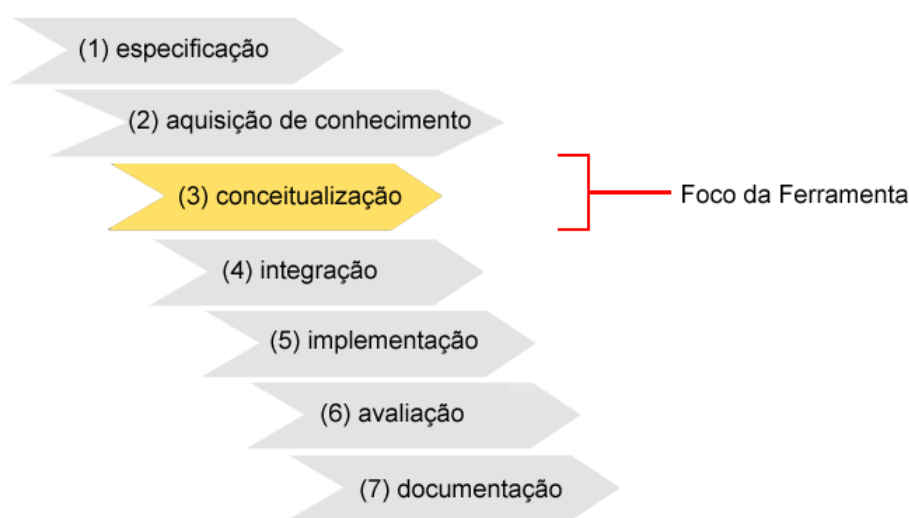


Figura 5.1 Foco da ferramenta em relação às etapas da metodologia Methontology.

A seguir, descrevemos em detalhes a arquitetura da ferramenta, baseada em camadas e projetada para suportar o uso de metadados e elencamos as principais funcionalidades do sistema.

5.1 Arquitetura

A aplicação foi desenvolvida no modelo de três camadas (MVC – *Model View Controller*): Camada de Interface, Camada de Modelo e Camada de Negócio. Utilizamos uma camada de acesso a dados adicional para abstrair a manipulação de triplas e outras funções de baixo nível de manipulação da ontologia. A arquitetura completa do sistema pode ser vista na Figura 5.1. A seguir, explicaremos cada componente da arquitetura em detalhe.

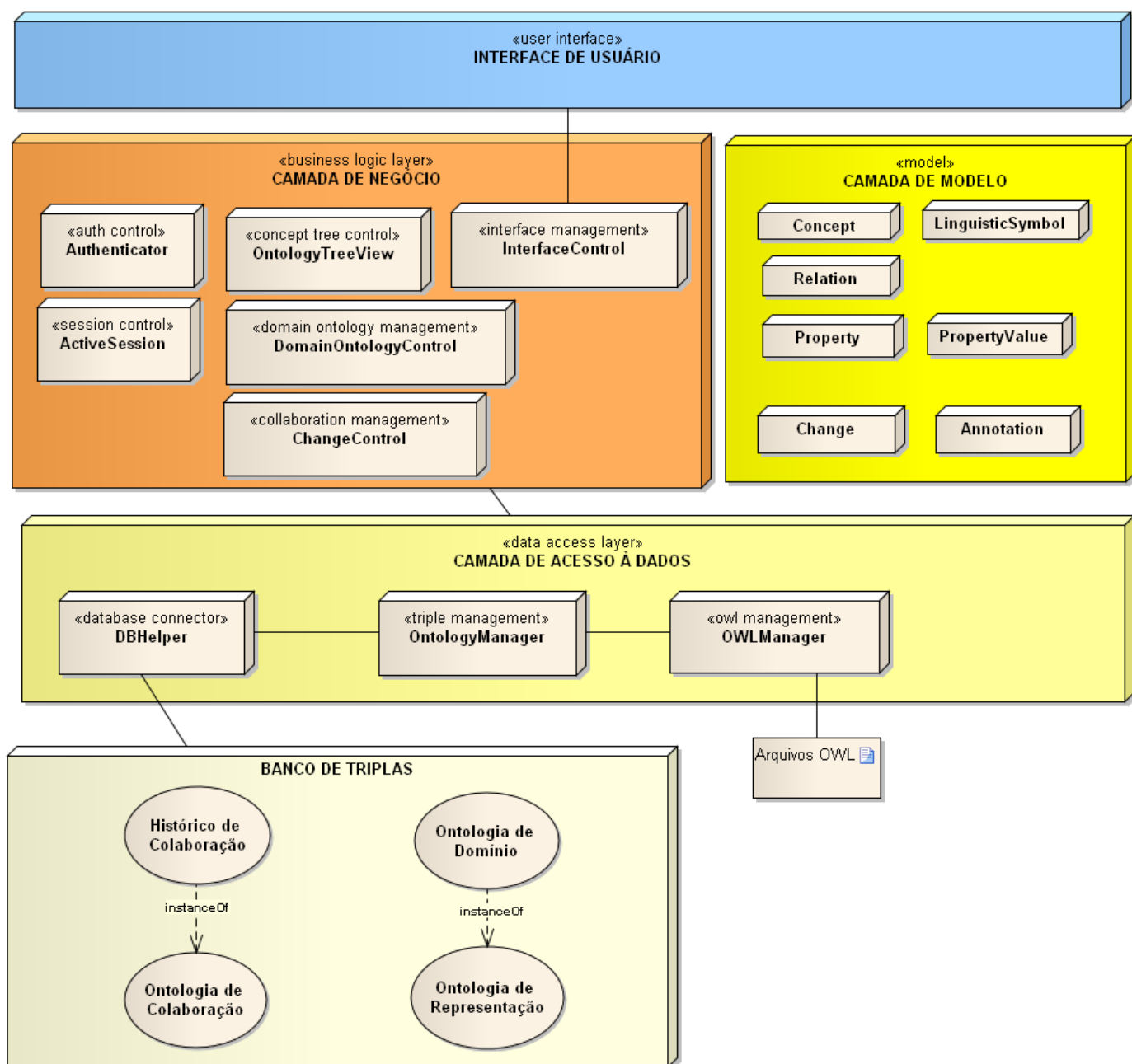


Figura 5.2 Arquitetura em camadas da ferramenta.

A camada de **Interface de Usuário** é responsável pelos componentes visuais com os quais o usuário fará a interação, como botões, listas e formulários. A maior parte desses componentes é fornecido pelo framework que utilizamos, o JBoss Seam, porém integrados com os componentes da camada de negócio.

A **Camada de Modelo** inclui classes que definem os componentes básicos do sistema. No caso, são definidos os componentes da ontologia de domínio, baseados na sua estrutura, especificada na metaontologia descrita anteriormente como **Ontologia de Representação**. O atributo URI está presente em quase todos os componentes e é o identificador único do componente, após ser instanciado na ontologia de domínio. Os componentes definidos nessa camada são:

Concept (Conceito) : define o componente **conceito**.

LinguisticSymbol (Símbolo Linguístico) : define o componente **termo linguístico** em um ou mais idiomas.

Relation (Relação) : define o componente **relação**.

Property (Propriedade) : define o componente **propriedade**.

PropertyValue (Valor de Propriedade) : define o componente **valor de propriedade**.

Change (Mudança) : define o componente **mudança**, a qual é vinculada a um componente ontológico (componente da ontologia de domínio que sofreu a mudança).

Annotation (Anotação) : define o componente **anotação**, a qual é vinculada a um componente da ontologia de domínio, uma mudança ou outra anotação.

A **Camada de Negócio** inclui as classes que contemplam as regras de negócio que são responsáveis pela funcionalidade do sistema. A “inteligência” do sistema encontra-se nessa camada. Essa camada utiliza os modelos definidos na Camada de Modelo. A seguir estão as classes que compõem esse módulo de software em detalhes:

Authenticator : classe de infra-estrutura, responsável pelo controle de autenticação de usuários através de usuário e senha;

ActiveSession : classe de infra-estrutura, responsável pelo controle de sessão ativa de usuário, ou seja, pelo gerenciamento dos dados utilizados em um determinado acesso ao sistema;

InterfaceControl : responsável pela interação da camada de interface com as demais classes da camada de negócios. Possui métodos que são chamados, em sua maioria, diretamente pela interface.

OntologyTreeView : responsável pelo controle de carregamento e montagem da árvore de conceitos;

DomainOntologyControl : responsável pelo gerenciamento da ontologia de domínio, definindo como será carregada a ontologia a ser utilizada no momento da colaboração.

ChangeControl : responsável pelo controle da colaboração, incluindo métodos para a criação e remoção de componentes da ontologia.

A **Camada de Acesso a Dados** é responsável por abstrair o acesso ao banco de dados ou outros artefatos de armazenamento, como arquivos OWL. Essa camada é composta por três classes principais:

DBHelper : responsável pelo gerenciamento de conexões com o banco de dados.

OntologyManager : responsável pelo gerenciamento de triplas no formato sujeito, predicado, objeto. Essa classe é muito utilizada pela classe ChangeControl, da camada de negócio, permitindo a abstração da criação de triplas através do seu uso em alto nível.

OWLManager : responsável pela leitura ou geração de arquivos OWL.

O **Banco de Triplas** é o armazenamento efetivo dos dados, o qual é explicado em detalhes na seção a seguir.

5.2 Persistência de Dados

Dentro do contexto de sistemas de informação, uma ontologia é um artefato de engenharia que utiliza um vocabulário específico para descrever certa realidade. Sabemos que uma ontologia de domínio é uma especificação formal de um conhecimento compartilhado por um grupo sobre um determinado domínio de interesse (Borst 1997). De acordo com (R. Studer et al. 1998), uma ontologia altamente informal não será uma ontologia pois não pode ser compreendida por um computador. Por esta razão, uma ontologia precisa ser representada por uma linguagem formal como RDF ou OWL. O *Resource Description Framework (RDF)* é uma linguagem que consiste na verdade de um modelo de dados de metadados utilizado para modelar descrições conceituais sobre recursos. A *Web Ontology Language (OWL)* é uma linguagem para representação de conhecimento baseada em RDF e caracterizada por possibilitar a expressão de semântica formal e através de lógica de primeira ordem. Ambas são indicadas para o armazenamento de ontologias pelo W3C (Berners-Lee et al. 2001).

Entretanto, também é possível armazenar metadados de RDF ou OWL em banco de dados utilizando-se as chamadas *Triple Stores*. As *Triple Stores* armazenam todas as informações somente com triplas de objeto-atributo-valor, utilizando uma ou mais tabelas e recuperando as informações com linguagens de consulta estendidas de SQL, como o SPARQL (SPARQL Protocol and RDF Query Language) ou diretamente via SQL (Structured Query Language).

Escolhemos o formato em banco de triplas para armazenar as informações das ontologias de metadados e de domínio pois essa forma de armazenamento manipula os dados de forma genérica, permitindo a expansão do modelo de conhecimento sem exigir mudanças na estrutura da base de dados.

Em nosso trabalho utilizamos uma tabela com três colunas (objeto-atributo-valor) para armazenar as triplas, com suas URIs (identificadores únicos). Uma quarta coluna

foi adicionada ao modelo padrão de triplas para indicar a qual grafo (modelo conceitual) a tripla pertence. Dessa forma, triplas idênticas poderão coexistir, desde que pertençam a ontologias de domínio diferentes. Por questões de performance, um índice foi atribuído para cada coluna da tabela de triplas.

Para fins de implementação, foi criado um grafo (modelo conceitual) para armazenar as metaontologias de Representação e Colaboração de forma unificada. A URI dada a esse grafo foi "**http://localhost/obaita/#MetaData**". Dessa forma, todas as triplas que descrevem as metaontologias são armazenadas na tabela de triplas referenciando esse grafo.

As ontologias de domínio criadas pelo sistema, como visto na seção 3.2, são instâncias da Ontologia de Representação. Dessa forma, um grafo distinto é criado para cada ontologia de domínio (Ex: "**http://localhost/obaita/#dominio_estratigrafia_versao_1**"). As triplas são inseridas na tabela de triplas, referenciando o respectivo grafo da ontologia de domínio. Dessa forma é possível diferenciar componentes idênticos de diferentes ontologias de domínio.

Os eventos de colaboração ocorridos no sistema, como visto na seção 3.2, são instâncias da Ontologia de Colaboração. Dessa forma, um grafo distinto é criado para a colaboração de cada ontologia de domínio, a fim de armazenar separadamente os componentes ontológicos dos eventos de colaboração (Ex: "**http://localhost/obaita/#dominio_estratigrafia_versao_1_Collaboration**"). As triplas são inseridas na tabela de triplas, referenciando o respectivo grafo.

5.2.1 Uso de Triplas

Para exemplificar o uso de triplas, vamos supor que desejamos fazer as seguintes modificações em uma ontologia de domínio:

- a) Criar um conceito cujos termos linguísticos são “Sedimentary Facies” em inglês e “Facies Sedimentar” em português;
- b) Mudar o metatipo do conceito para Kind;
- c) Associar um ícone ao conceito;

Primeiro, criamos o conceito através do método *addConceptCreatedChange* (*conceptUri*, *changeDescription*) da classe **ChangeControl**, o qual irá instanciar o meta-conceito da ontologia de representação *OntologyConcept* e o meta-conceito da ontologia de colaboração *ConceptCreated*, gerando as triplas exibidas na Tabela 5.1. Como podemos ver, três triplas são criadas no grafo da ontologia de domínio (O.D.), pois descrevem componentes da ontologia e quatro triplas são criadas no grafo da ontologia de colaboração (O.C.), pois são eventos de colaboração.

Tabela 5.1 Triplas geradas ao criar um novo conceito.

<i>Grafo</i>	<i>Sujeito</i>	<i>Predicado</i>	<i>Objeto</i>
O.D.	#Concept_1	rdf-syntax-ns#type	OntologyConcept
O.D.	#Concept_1	labelEN	Sedimentary Facies
O.D.	#Concept_1	labelPT	Facies Sedimentar
O.C.	#Change_1	rdf-syntax-ns#type	ConceptCreated
O.C.	#Change_1	domainComponent	#Concept_1
O.C.	#Change_1	date	2011-03-26 20:31:26
O.C.	#Change_1	author	gabriel

Supondo que, baseado nas metapropriedades, o sistema inferiu que o metatipo do conceito é um *Kind*, podemos então indicar formalmente que esse é o metatipo correto para o conceito (em alguns casos, as metapropriedades não estão classificadas corretamente, levando a um metatipo incorreto, portanto há uma classificação formal do metatipo de cada conceito), como vimos na descrição das funcionalidades da ferramenta. Nessa operação, o sistema chama o método *addConceptTypeChange(conceptUri, new-Type)* da classe **ChangeControl**, o qual gera as triplas vistas na Tabela 5.2. Note que nesse caso nenhuma tripla é criada no grafo de domínio, somente no grafo de colaboração.

Tabela 5.2 Triplas geradas ao alterar o metatipo de um conceito.

<i>Grafo</i>	<i>Sujeito</i>	<i>Predicado</i>	<i>Objeto</i>
O.C.	#Change_3	rdf-syntax-ns#type	ConceptTypeChange
O.C.	#Change_3	domainComponent	#Concept_1
O.C.	#Change_3	value	Kind
O.C.	#Change_3	date	2011-03-26 20:34:31
O.C.	#Change_3	author	gabriel

Por último, ao associar um ícone ao conceito, a imagem é enviada para o servidor e é chamado o método *addConceptIcon(conceptUri, iconURL)* da classe **ChangeControl**, o qual gera as triplas vistas na Tabela 5.3.

Tabela 5.3 Triplas geradas ao alterar o ícone de um conceito.

<i>Grafo</i>	<i>Sujeito</i>	<i>Predicado</i>	<i>Objeto</i>
O.D.	#Icon_1	rdf-syntax-ns#type#type	Icon
O.D.	#Icon_1	imageURL	http://onto/icon1.jpg
O.C.	#Change_2	rdf-syntax-ns#type#type	ConceptIconChange
O.C.	#Change_2	domainComponent	#Concept_1
O.C.	#Change_2	value	#Icon_1
O.C.	#Change_2	date	2011-03-26 20:35:21
O.C.	#Change_2	author	gabriel

5.3 Interação com a Linguagem OWL

Visando compatibilidade com padrões atuais, pode-se estabelecer uma correspondência entre os principais componentes da ontologia de representação e primitivas da linguagem OWL, possibilitando a geração da ontologia de domínio no sistema a partir de ontologias já existentes em OWL, ou a geração de uma ontologia em OWL a partir da ontologia de domínio no sistema.

No caso de gerar a ontologia de domínio a partir de uma ontologia OWL, os componentes ontológicos são importados de forma genérica (por exemplo, toda *Class* em OWL transforma-se em um *OntologyConcept* na ontologia de domínio) para depois ser especializada em algum dos seus subtipos (Ex: *Sortal*, *Mixin*, *Role*, etc.) durante a etapa de colaboração.

No caso de gerar a ontologia OWL a partir da ontologia de domínio, é importante ressaltar que toda informação semântica já vinculada aos componentes ontológicos através do sistema será perdida nessa operação, pois a linguagem OWL não possui primitivas com semântica precisa o suficiente para armazenar conhecimento visual ou fundamentação ontológica, não permitindo, por exemplo, especificar relações partonômicas, metatipos dos conceitos ou vincular imagens aos componentes.

A Tabela 5.4 abaixo mostra a correspondência possível dos componentes da ontologia de domínio com as primitivas da linguagem OWL.

Tabela 5.4 Correspondência dos componentes da ontologia de representação com construtos de OWL.

Ontologia de Representação	Primitiva OWL DL
Conceito	<i>Class</i>
Relação	<i>ObjectProperty</i>
Propriedade	<i>DatatypeProperty</i>
Ícone	[Sem correspondência]
Fotografia	[Sem correspondência]

5.4 Funcionalidades do Sistema

O acesso à ferramenta é baseado em autenticação através de usuário e senha. Existem diferentes tipos de usuário, *Especialista*, *Engenheiro de Conhecimento* e *Visitante*. Após o *login*, o usuário é levado até a área de colaboração, onde pode escolher o idioma de edição da ontologia. Neste trabalho, abordamos os idiomas Português e Inglês, porém mais idiomas podem ser adicionados sem maiores modificações.

O sistema comporta mais de uma ontologia de domínio. O usuário seleciona a ontologia de domínio que deseja editar para exibir a árvore de conceitos.

Novos conceitos podem ser adicionados na raiz da ontologia através do botão com o sinal de +. Para cada mudança efetuada, o sistema solicita diferentes informações do usuário, porém sempre solicita uma Justificativa de Mudança, a qual consiste de uma anotação do tipo *Explanation*, a qual fica vinculada a esta mudança.

Entretanto, conceitos podem ser adicionados em um nível hierárquico inferior a outro conceito já existente, ou seja, possuindo uma relação *SubClassOf* em relação a esse conceito. Para isso, o usuário utiliza um menu acessado através do clique com o botão direito em um conceito já existente. Também é possível excluir o conceito, como visto na Figura 5.2.

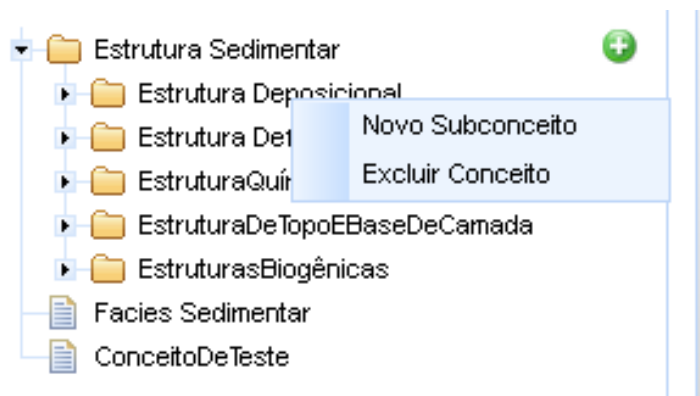


Figura 5.3 Árvore de conceitos da ontologia.

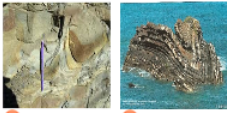
5.4.1 Detalhes do Conceito

Ao exibir os detalhes de um conceito, são exibidos diversos painéis, possibilitando mudanças sobre todos os componentes do mesmo, como visto na Figura 5.3. Um ícone específico acompanha cada um dos componentes, possibilitando abrir um painel de anotações referente ao componente em questão. A seguir explicamos cada painel de componente em detalhes.


Conceito: Estrutura Deformacional

URL: http://localhost/obaiba/#imported-485992548



Fotos de Instâncias



Ícone












Termos(Símbolo Linguístico)

Linguagem	Termo	
Portugues	Estrutura Deformacional	
English	DeformationStructure	









Estatísticas

Data de Criação: 07/06/2011 16:56:25
Última Atualização: 14/05/2012 11:53:52
Nº de Mudanças: 7
Nº de Anotações: 0


Propriedades

Nome	Tipo	 
(SentidoDeMergulho)	Conjunto de Valores	 
(FormaDaLaminação)	Conjunto de Valores	 
(Paleocorrente)	Conjunto de Valores	 
(Angularidade)	Conjunto de Valores	 
(Espessura)	Conjunto de Valores	 


Relações

Relação	Tipo	Conceito Destino	 
creates_DepositionalProcess_SedimentaryStructure	Componente de	MarcasOnduladasLineares	  
(creates_DepositionalProcess_SedimentaryStructure)	Componente de	MarcasOnduladasLineares	 
(indicatesFacies_Process)	SubQuantidade de	Estrutura Depositional	 
(hasPart)	SubColecao de	Estrutura Depositional	 


Meta-Dados


Meta-Tipo Atual: Kind 


Meta-Propriedades:

Rigidez: Esse conceito pode deixar de existir (e passar ou não a ser outro) em algum determinado momento? Exemplo: Um "Professor", pois algum exemplar desta classe pode deixar de ser professor e se tornar um "Diretor".
 

Dependência: Um exemplar desse conceito só pode existir no mundo real, caso exista um exemplar que dê identidade a ele e algum outro conceito, relacionado à este, que o defina como este exemplar? Exemplo: Um "Estudante" só existe caso esteja relacionado a uma "Instituição de Ensino" (que o define como estudante) e a uma Pessoa (que lhe dá identidade).
 

Provê Identidade: Um exemplar desse conceito possui propriedades essenciais que permitam diferenciá-lo de outro, mesmo com a passagem do tempo? Exemplo: Um "Cachorro" provê identidade pois mantém suas propriedades ao longo do tempo, enquanto "Filhote de Cachorro" não, pois perde algumas propriedades ao longo do tempo. Da mesma forma, um Estudante não provê identidade, pois suas propriedades mudam ao longo do tempo. O conceito "Pessoa" provê a identidade que é carregada pelo conceito "Estudante".
 

Carrega Identidade: Um exemplar desse conceito carrega a identidade de algum conceito que defina suas propriedades essenciais? Exemplo: Um "Professor" ou "Estudante" não provê identidade, apenas carrega a identidade provida pelo conceito "Pessoa".
 

Unicidade: Um exemplar desse conceito carrega unicidade?
 Exemplos:
 Uma "FAIXA DE AREIA" é unitária, é formado por diversos objetos, porém POSSUI limites definidos e todos objetos são do mesmo critério de unicidade.
 Um "GRÃO DE AREIA" é unitário, pois POSSUI limites definidos e é formado por um objeto único.
 Um "QUANTIDADE DE AREIA" é anti-unitário, é formado por diversos objetos, porém NÃO POSSUI limites definidos.
 Um "QUADRÚPEDE" é não-unitário, é formado por diversos objetos mas NEM TODOS os objetos são do mesmo critério de unicidade.
 

Meta-Tipo Sugerido: - (sugerido com base nas Meta-Propriedades)

Figura 5.4 Tela de detalhes do conceito.

5.4.1.1 Termo Linguístico

Cada conceito possui um termo linguístico para cada idioma, os quais podem ser alterados pelo painel visto na Figura 5.4.



Termos(Símbolo Linguístico)		
Linguagem	Termo	
Portugues	Estrutura Deformacional	
English	DeformationStructure	

Figura 5.5 Painel para alteração do termo linguístico de um conceito.

5.4.1.2 Ícone

Como já vimos anteriormente, um ícone pode ser associado a um conceito, de modo a ancorar a representação pictórica do mesmo, ampliando a comunicação e o entendimento dos conceitos pelos especialistas. Cada conceito pode possuir no máximo um ícone, selecionado através do painel visto na Figura 5.5.



Figura 5.6 Painel para seleção de um ícone para o conceito.

5.4.1.3 Fotos

Como também já vimos, fotos podem ser associadas a um conceito, de modo a exemplificar as instâncias do mesmo na realidade. O painel de fotos pode ser visto em detalhes na Figura 5.6.

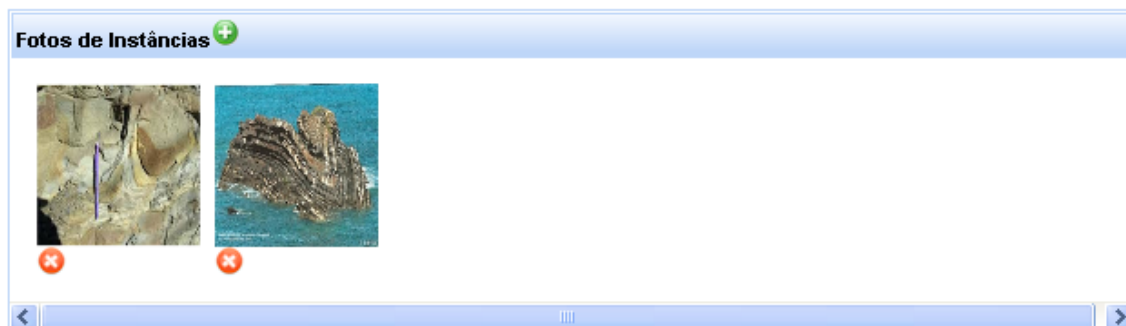


Figura 5.7 Painel para seleção de fotos de exemplos de instâncias do conceito.

5.4.1.4 Propriedades

O painel de propriedades exibe as propriedades vinculadas ao conceito, permitindo a criação de novas propriedades ou o vínculo de propriedades já existentes ao conceito, assim como é possível desvincular as propriedades do conceito. Esse painel pode ser visto na Figura 5.7.

Propriedades		
Nome	Tipo	
(SentidoDeMergulho)	Conjunto de Valores	 
(FormaDaLaminação)	Conjunto de Valores	 
(Paleocorrente)	Conjunto de Valores	 
(Angularidade)	Conjunto de Valores	 
(Espessura)	Conjunto de Valores	 

Figura 5.8 Painel de propriedades do conceito.

Ao editar uma propriedade, pode-se escolher o nome (termo para o idioma de edição selecionado no início da colaboração), o tipo da propriedade (Valores Inteiros, Valores com Vírgula, Conjunto de Valores (*qualia*), etc.). Como visto na etapa de modelagem, os valores de propriedades que possuem um conjunto finito de valores podem ter um ícone associado.

5.4.1.5 Relações

O painel de relações exibe as relações vinculadas ao conceito, permitindo a criação de novas relações ou o vínculo de relações já existentes ao conceito, assim como é possível desvincular as relações do conceito. Esse painel pode ser visto na Figura 5.8.

Relações			
Relação	Tipo	Conceito Destino	
creates_DepositionalProcess_SedimentaryStructure	Componente de	MarcasOnduladasLineares	  
(creates_DepositionalProcess_SedimentaryStructure)	Componente de	MarcasOnduladasLineares	 
(indicatesFacies_Process)	SubQuantidade de	Estrutura Deposicional	 
(hasPart)	SubColecao de	Estrutura Deposicional	 

Figura 5.9 Painel de relações do conceito.

Ao editar uma relação, pode-se escolher o nome (termo para o idioma de edição selecionado no início da colaboração), o tipo da relação (classificação pela UFO-A), o conceito destino, a cardinalidade de origem e a cardinalidade de destino. O campo *Cardinalidade* define a multiplicidade da relação, tanto para o conceito origem quanto para o conceito destino. O conceito de origem é automaticamente selecionado como o conceito que está sendo visualizado em detalhes no sistema. O painel de edição de uma relação pode ser visto na Figura 5.9.

Editar Relação

Alterar Nome:

Tipo de Relação: Membro de ▼

Conceito Origem: Estrutura Sedimentar
 Cardinalidade: Um ▼

Conceito Destino: ————>> Laminação Trativa ▼

Cardinalidade: Um ou Mais ▼

Justificativa da Mudança:

Salvar

Cancelar

Figura 5.10 Painel de edição de uma relação.

5.4.1.6 Metapropriedades

O painel de metadados exibe as informações de fundamentação ontológica do conceito. Como as metapropriedades devem ser classificadas pelo especialista, para cada metapropriedade definida pela metodologia *OntoClean* (Guarino 1995) foi associada uma pergunta ou explicação que facilita o entendimento do significado dessa metapro-

priedade pelo mesmo, como visto na Figura 5.10. Dessa forma, ele pode classificar as metapropriedades sem compreender profundamente os aspectos teóricos que as caracterizam. Através dessa classificação, como vimos anteriormente, é possível inferir o meta-tipo do conceito, como visto na Figura 5.11.

Meta-Propriedades:	
Rigidez:	<p>Esse conceito pode deixar de existir (e passar ou não a ser outro) em algum determinado momento? Exemplo: Um "Professor", pois algum exemplar desta classe pode deixar de ser professor e se tornar um "Diretor".</p> <p>SIM, TODOS OS EXEMPLARES PODEM DEIXAR DE EXI <input type="button" value="v"/></p>
Dependência:	<p>Um exemplar desse conceito só pode existir, no mundo real, caso exista um exemplar que dê identidade a ele e algum outro conceito, relacionado à este, que o defina como este exemplar? Exemplo: Um "Estudante" só existe caso esteja relacionado a uma "Instituição de Ensino" (que o define como estudante) e a uma Pessoa (que lhe dá identidade).</p> <p>NÃO = Não Dependente <input type="button" value="v"/></p>
Provê Identidade:	<p>Um exemplar desse conceito possui propriedades essenciais que permitam diferenciá-lo de outro, mesmo com a passagem do tempo? Exemplo: Um "Cachorro" provê identidade pois mantém suas propriedades ao longo do tempo, enquanto "Filhote de Cachorro" não, pois perde algumas propriedades ao longo do tempo. Da mesma forma, um Estudante não provê identidade, pois suas propriedades mudam ao longo do tempo. O conceito "Pessoa" provê a identidade que é carregada pelo conceito "Estudante".</p> <p>SIM = Provê <input type="button" value="v"/></p>
Carrega Identidade:	<p>Um exemplar desse conceito carrega a identidade de algum conceito que defina suas propriedades essenciais? Exemplo: Um "Professor" ou "Estudante" não provê identidade, apenas carrega a identidade provida pelo conceito "Pessoa".</p> <p>SIM = Carrega <input type="button" value="v"/></p>
Unicidade:	<p>Um exemplar desse conceito carrega unicidade? Exemplos: Uma "FAIXA DE AREIA" é unitária, é formado por diversos objetos, porém POSSUI limites definidos e todos objetos são do mesmo critério de unicidade. Um "GRÃO DE AREIA" é unitário, pois POSSUI limites definidos e é formado por um objeto único. Um "QUANTIDADE DE AREIA" é anti-unitário, é formado por diversos objetos, porém NÃO POSSUI limites definidos. Um "QUADRÚPEDE" é não-unitário, é formado por diversos objetos mas NEM TODOS os objetos são do mesmo critério de unicidade.</p> <p>Selecione <input type="button" value="v"/></p>

Figura 5.11 Painel de metapropriedades do conceito.

Meta-Tipo Sugerido: Kind (sugerido com base nas Meta-Propriedades)

Figura 5.12 Inferência do metatipo através da classificação de metapropriedades.

Após essa inferência, o **Engenheiro de Conhecimento** deve avaliar se a sugestão está correta e então classificar o metatipo manualmente, pois em alguns casos a classificação incorreta das metapropriedades pode induzir a um metatipo também incorreto. Dessa forma, o metatipo do conceito também pode ser classificado de forma independente das metapropriedades, como visto na Figura 5.12.



Figura 5.13 Classificação manual do metatipo do conceito.

Ao editar o meta-tipo, o usuário pode selecionar entre os tipos disponíveis, vistos na Figura 5.13.

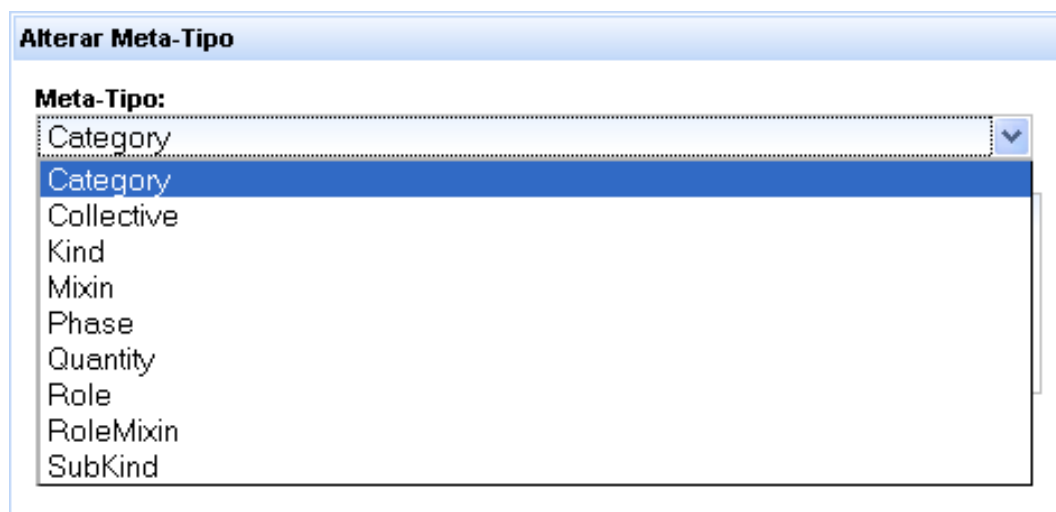


Figura 5.14 Escolha entre os metatipos disponíveis para o conceito (oriundos da UFO-A).

O sistema desenvolvido possui arquitetura expansível, o que possibilita a sua expansão de forma livre, através da criação de novas funcionalidades e do aprimoramento de sua camada de metadados. A seguir, mostramos alguns resultados obtidos após uma etapa de colaboração em uma ontologia de domínio real e a proposta de um sistema de validação detalhado para avaliar a abordagem.

6 VALIDAÇÃO

Esse capítulo descreve a abordagem de validação do sistema proposto neste trabalho e o planejamento do instrumento de avaliação pelos usuários finais, concebido para identificar as limitações do ambiente proposto.

A ferramenta foi utilizada para construção de uma ontologia de domínio para descrição de fácies em Estratigrafia Sedimentar. O instrumento será aplicado em um grupo de testes de geólogos convidados do meio acadêmico, empresas e sociedades científicas brasileiras envolvidos na negociação da construção dos conceitos em estratigrafia sedimentar. Por restrições de prazo, o instrumento ainda não foi aplicado e está sendo avaliado pelos geólogos da própria equipe de desenvolvimento.

6.1 Ontologia de Domínio do Estudo de Caso

Para validar o sistema em um ambiente real, buscamos um estudo de caso com o domínio de Estratigrafia Sedimentar, área da Geologia especializada no estudo dos estratos ou camadas de rochas, buscando determinar os processos e eventos que as formaram.

A Estratigrafia Sedimentar especifica os corpos rochosos que formam a crosta da Terra e distinguem as suas unidades através da identificação de cada tipo de rocha. Para identificar a porção de rocha, o especialista muitas vezes necessita de apoio visual para vincular os padrões da rocha com alguma categoria de rocha conhecida. Por isso se justifica o uso de componentes visuais como ícones e fotografias na modelagem do conhecimento do domínio em forma de ontologia.

Esse domínio foi escolhido principalmente porque possui alguns aspectos importantes para o nosso foco:

- a) é fortemente fundamentado em conhecimento visual;
- b) sua estruturação é complexa, envolve diversos problemas de estrutura e discordâncias entre os especialistas;
- c) tem importância científica e econômica, pois estuda as condições de geração e atual distribuição espacial de depósitos minerais importantes como o carvão e o petróleo.

Em (Lorenzatti et al. 2011), uma ontologia de domínio foi construída com o auxílio de especialistas na área de Estratigrafia Sedimentar, buscando também associar ícones a conceitos e valores de propriedades. Essa ontologia serviu como base para dar início às etapas de colaboração do sistema, ou seja, os conceitos, propriedades e relações foram importados no sistema de colaboração, propiciando que um grupo maior de especialistas julgue as informações, discutindo sobre elas de forma organizada e propondo mudanças para se obter um modelo estável da ontologia, refletindo a conceitualização compartilhada do domínio.

Os conceitos de fácies sedimentar, estrutura sedimentar, de processo deposicional e ambiente sedimentar são fundamentais para as atividades do domínio. Um ambiente sedimentar corresponde a uma área da superfície terrestre com características físicas, químicas e biológicas que a distinguem das áreas adjacentes. Os ambientes sedimentares deposicionais também são conhecidos como ambientes de sedimentação ou ambientes deposicionais e são caracterizados e classificados em função dos diversos processos deposicionais que ocorrem neles. A taxonomia de estruturas deposicionais pode ser vista na Figura 6.1. Fácies Sedimentar (do latim *facies*, que significa face) corresponde à menor unidade que compõem um pacote de rochas sedimentares e corresponde ao conjunto de aspectos litológicos e paleontológicos de um depósito sedimentar (Fávera 2001).

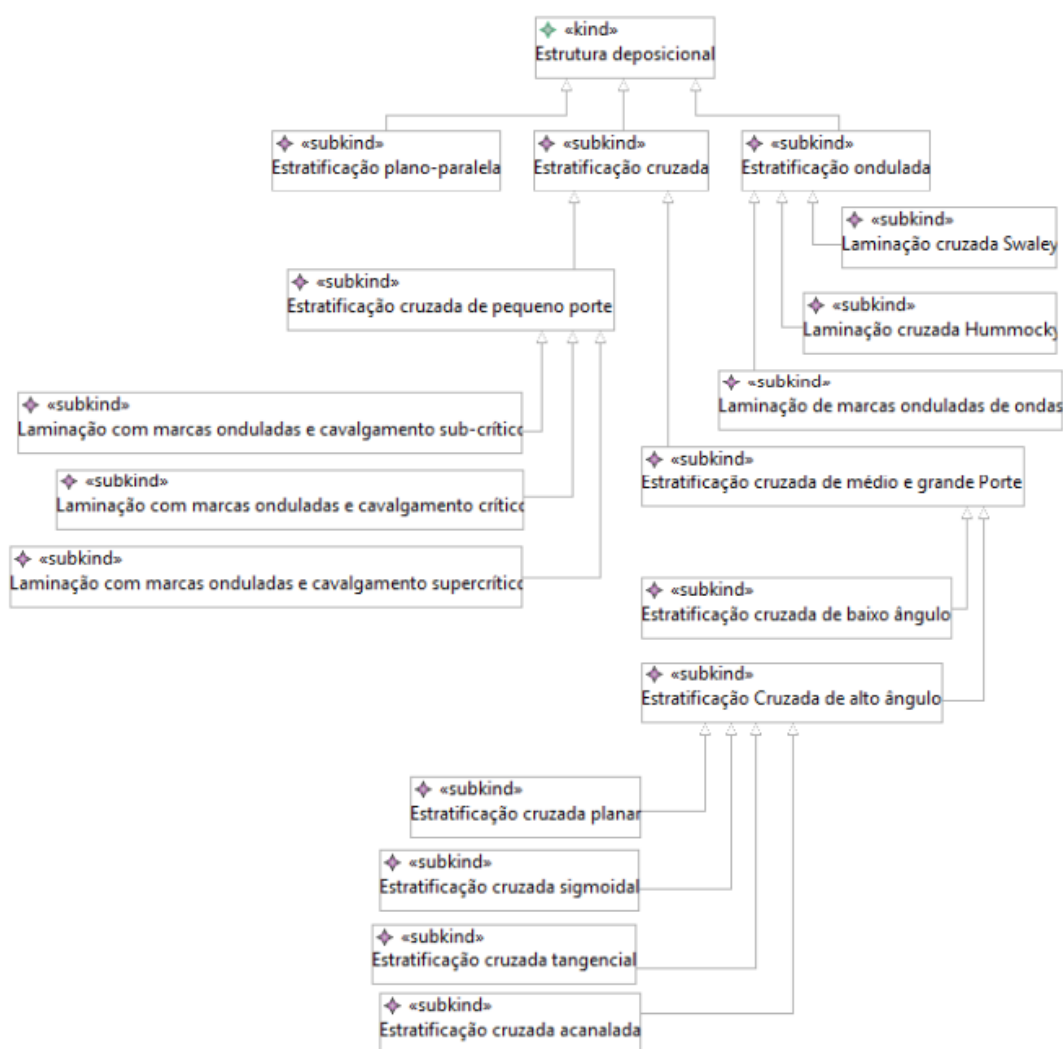


Figura 6.1 Taxonomia de estruturas deposicionais, extraído de Lorenzatti et al. (2011).

6.2 Avaliação

Como avaliação prévia do ambiente, disponibilizamos o mesmo para três usuários (dois geólogos e 1 engenheiro de conhecimento) que a utilizarem por aproximadamente

6 meses evoluindo uma ontologia de domínio previamente construída através de um processo de entrevistas com especialistas. Adicionalmente, propomos um sistema de validação para que seja possível avaliar esta ferramenta colaborativa com mais qualidade e abrangência em trabalhos futuros.

6.2.1 Uso da Ferramenta pelos Usuários de Teste

A ontologia de domínio construída em (Lorenzatti et al. 2011) estava disponível em formato OWL, o que permitiu a sua importação para o sistema colaborativo. Isso foi feito através da conversão de primitivas OWL para os construtos do sistema desenvolvido, gerando instâncias da Ontologia de Representação. Após, convidamos um grupo de usuários, estudantes de Geologia, para analisar a ontologia e propor novas mudanças, a fim de consolidá-la através do ambiente de colaboração. Os ícones definidos no trabalho de Lorenzatti não estavam vinculados à ontologia pois não existem construtos adequados para representá-los em OWL. Portanto, os ícones foram formalmente vinculados à ontologia de domínio no processo de colaboração.

Os usuários de teste utilizaram o sistema ao longo de 3 meses, gerando modificações nos conceitos, propriedades e relações da ontologia, além de introduzir ícones já desenvolvidos no trabalho de Lorenzatti (2009) e ícones novos, criados durante o processo de colaboração. O cenário quantitativo da colaboração sobre a ontologia de domínio pode ser visto na Tabela 6.1.

Tabela 6.1 Análise quantitativa da colaboração efetuada sobre a ontologia de domínio.

ATRIBUTO	ANTES DA COLABORAÇÃO	APÓS COLABORAÇÃO
Nº de Total de Mudanças Efetuadas	Não Disponível	1485
Nº de Total de Conceitos	121	183
Nº de Total de Propriedades	10	22
Nº de Total de Relações	28	15
Nº de Total de Ícones	Nenhum	125
Nº de Conceitos Adicionados	Não Disponível	93
Nº de Conceitos Excluídos	Não Disponível	31
Nº de Conceitos Alterados	Não Disponível	84
Nº de Propriedades Adicionadas	Não Disponível	25
Nº de Propriedades Excluídas	Não Disponível	13

Nº de Relações Adicionadas	Não Disponível	16
Nº de Relações Excluídas	Não Disponível	29
Nº de Ícones vinculados a Conceitos	Nenhum	26
Nº de Ícones vinculados a Valores de Propriedade	Nenhum	99
Nº de Alterações nas Metapropriedades dos Conceitos	Não Disponível	63
Nº de Metatipos classificados manualmente pelos usuários	Não Disponível	5

Após a análise, pode-se perceber que o número de conceitos e propriedades aumentou após a colaboração, devido à expansão da ontologia pelos usuários. O número de relações reduziu pois a ontologia existente possuía muitas propriedades incorretamente definidas como relações pela dificuldade de se formalizar propriedades com valores pré-definidos (*QualityStructure*) na ferramenta onde a ontologia foi modelada, o Protégé. Pode-se notar que diversas modificações foram feitas em relação à elementos visuais (ícones) e elementos de fundamentação (metapropriedades e metatipos), pela capacidade que a ferramenta tem em se trabalhar com esses artefatos. Ressaltamos que na contabilização de relações, não foram consideradas as relações de hierarquia (*subclassOf*), por se tratarem de relações inerentes à taxonomia e não à semântica do domínio.

6.2.2 Comparação da Ferramenta

Para efetuar uma análise comparativa com as ferramentas existentes para colaboração de ontologias, com base nos critérios de colaboração definidos na seção 3.2.3, construímos um quadro comparativo das soluções atuais e da solução proposta nesse trabalho, visto na Tabela 6.2.

Tabela 6.2 Quadro comparativo entre as ferramentas de construção coletiva de ontologias.

Critério	Fabrico	OntoEdit	NeOn Toolkit	Collaborative Protégé	WebODE	Obaitá
Discussão Estruturada (Anotações)	NÃO	NÃO	SIM	SIM	NÃO	SIM
Histórico de Colaboração	NÃO	NÃO	NÃO	SIM	NÃO	SIM
Suporte à OWL	NÃO	NÃO	SIM	SIM	SIM	SIM
Suporte à Inferência	NÃO	SIM	SIM	SIM	SIM	NÃO

Detecção de Inconsistências	NÃO	SIM	SIM	SIM	SIM	NÃO
Perfis de Usuários Distintos	NÃO	NÃO	NÃO	NÃO	NÃO	SIM
Interface Intuitiva Web	SIM	NÃO	NÃO	SIM	NÃO	SIM
Acesso Centralizado via Navegador Web	SIM	SIM	NÃO	SIM	SIM	SIM
Possibilita utilização sem conhecimento de ontologia formal	SIM	SIM	SIM	NÃO	NÃO	SIM
Uso em Domínios Visuais	NÃO	NÃO	NÃO	NÃO	NÃO	SIM
Uso da Metodologia OntoClean	NÃO	SIM	NÃO	NÃO	SIM	SIM
Uso de Fundamentação Ontológica	NÃO	NÃO	NÃO	NÃO	NÃO	SIM

Após a comparação, nota-se que o portal *Obaitá* atende aos seus objetivos ao prover suporte a domínios visuais e fundamentação ontológica, além de prover interface intuitiva web e acesso via navegador de internet. Reconhece-se que a ferramenta não possui suporte à inferência automática, apesar de permitir que a ontologia construída seja aplicada à algum motor de inferência já existente. Com base nisso, assumimos que a ferramenta cumpre seu papel em relação à seus objetivos principais.

6.2.3 Construção do Sistema de Validação

Para verificar de forma imparcial a abordagem e a ferramenta, construímos um método de validação incluindo indicadores, métricas e questões. Este instrumento de validação pode ser utilizado para avaliar tanto o sistema proposto quando outras abordagens similares.

Inicialmente, definimos indicadores para avaliar a qualidade da abordagem e da ferramenta de uma forma mais precisa.

Abaixo estão os indicadores de qualidade intrínseca da abordagem:

1. Suporte à escolha da primitiva de representação
2. Modelagem com maior detalhamento
3. Convergência do modelo construído coletivamente
4. Aplicabilidade da ontologia construída

Abaixo estão os indicadores de qualidade intrínseca da ferramenta:

5. Facilidade de modificar a ontologia já existente
6. Usabilidade

Para possibilitar uma medição eficiente, definimos métricas concretas de avaliação para cada um dos indicadores. Cada uma das métricas abaixo está relacionada com o indicador de mesmo número.

1. Número de construtos fornecidos para representar os objetos ontológicos.
2. Número de construtos em níveis hierárquicos inferiores aos construtos ontológicos básicos (conceito, propriedade, relação).
3. Tempo decorrido para que o número de mudanças efetuadas na ontologia seja reduzido até certo limite (o limite pode variar de acordo com o número de usuários, tamanho da ontologia, etc.).
4. Número de aplicações ou projetos mapeados pela comunidade de usuários que poderão utilizar direta ou indiretamente a ontologia de domínio desenvolvida.
5. Número de objetos ontológicos alterados em determinado período, comparado com o número de objetos alterados em um processo de desenvolvimento manual de uma ontologia de tamanho equivalente.
6. Número de cliques necessários para acessar as funcionalidades mais comuns de modificação da ontologia pelos usuários.

Para efetuar a medição destas métricas, desenvolvemos questões que devem ser respondidas pelo grupo de usuários que utilizar a ferramenta em um processo colaborativo, buscando obter uma ideia inicial e objetiva da aplicabilidade e utilidade do sistema de forma imparcial. Algumas perguntas referem-se especificamente à ferramenta proposta enquanto outras comparação a tarefa de construção da ontologia de domínio através da documentação manual (ou utilizando editores de texto e planilhas eletrônicas) com o uso da ferramenta proposta neste trabalho. Dessa forma, chegamos às questões abaixo:

1) Em relação à associação dos conceitos do domínio com as primitivas de representação adequadas, é possível dizer que: (Essa questão está relacionada com a métrica 1)

- 1 – A ferramenta inviabiliza a tarefa de modelagem
- 2 – A ferramenta dificulta a tarefa de modelagem
- 3 – Equivalente a fazer a tarefa manualmente
- 4 – A ferramenta auxilia a tarefa de modelagem
- 5 – A ferramenta viabiliza modelagens que não seriam possíveis de fazer manualmente

- 2) **Os componentes ontológicos básicos (conceito, propriedade, relação) possuem construtos especializados para representá-los na proporção:** (Essa questão está relacionada com a métrica 2)
- 1 – Insuficiente / Somente é possível representar os componentes básicos
 - 2 – Abaixo do Esperado / Faltam construtos básicos para representar diversos componentes
 - 3 – Razoável / Alguns componentes não possuem construtos adequados para representá-los com expressividade
 - 4 – Adequada / Alguns componentes não possuem construtos adequados, porém sem perda representativa de expressividade
 - 5 – Perfeita / Todos os componentes possuíam construtos adequados para representá-los com expressividade
- 3) **O tempo decorrido desde o início da colaboração até que a ontologia chegasse a um senso comum, possibilitando utilizá-la para outros fins foi:** (Essa questão está relacionada com a métrica 3)
- 1 – Exagerado / Ultrapassou muito a expectativa dos usuários
 - 2 – Elevado / Ultrapassou levemente a expectativa dos usuários
 - 3 – Adequado / Ficou dentro da expectativa dos usuários
 - 4 – Reduzido / Levou menos tempo do que o esperado pelos usuários
 - 5 – Muito Reduzido / Levou muito menos tempo do que o esperado pelos usuários
- 4) **Como você classificaria a ontologia gerada APÓS o processo de colaboração em relação à sua aplicabilidade no mundo real, se comparada com a ontologia existente ANTES da colaboração?** (Essa questão está relacionada com a métrica 4)
- 1 – Muito Ruim / A ontologia poderia servir apenas como consulta informal
 - 2 – Ruim / A ontologia pode servir como entrada para sistemas ou como consulta informal, porém com a possibilidade gerar diversos erros
 - 3 – Adequada / A ontologia pode servir como entrada para sistemas ou como consulta informal, porém com a possibilidade gerar poucos erros
 - 4 – Boa / A ontologia pode servir como entrada para alguns sistemas ou como consulta formal
 - 5 – Muito Boa / A ontologia pode servir como entrada para diversos sistemas ou como consulta formal

5) O número de objetos ontológicos alterados desde o início da colaboração até que a ontologia chegasse a um senso comum se comparado com o número de objetos ontológicos alterados na construção da ontologia utilizando o processo manual foi: (Essa questão está relacionada com a métrica 5)

1 – Muito Reduzido / O processo manual gerou alterações em muito mais objetos do que no processo colaborativo

2 – Reduzido / O processo manual gerou alterações em mais objetos do que no processo colaborativo

3 – Equivalente / O processo manual alterou praticamente o mesmo número de objetos que o processo colaborativo

4 – Elevado / O processo colaborativo gerou alterações em mais objetos do que no processo manual

5 – Muito Elevado / O processo colaborativo gerou alterações em muito mais objetos do que no processo manual

6) Como você classificaria a interface de uso da ferramenta, em relação à facilidade no uso (facilidade para acessar as funcionalidades desejadas, clareza da interface, facilidade da interação com a interface) ? (Essa questão está relacionada com a métrica 6)

1 - Muito Ruim / Impossível utilizar algumas funcionalidades

2 – Ruim / Difícil de utilizar algumas funcionalidades

3 – Adequada / Funcionalidades básicas acessadas facilmente porém pouco claras e objetivas

4 – Boa / Todas as funcionalidades acessadas facilmente porém pouco claras

5 - Muito Boa / Todas as funcionalidades são claras, acessadas facilmente e objetivas

7) Como você classificaria a interface de uso da ferramenta em relação a outras ferramentas de edição de ontologias que já utilizou? (Essa questão está relacionada com a métrica 6)

1 - Muito Pior / As outras ferramentas são extremamente mais fáceis de utilizar

2 – Pior / Algumas ferramentas são mais fáceis de utilizar

3 – Equivalente / As outras ferramentas possuem interface similar ou com mesmo nível de dificuldade de interação

4 – Melhor / As outras ferramentas são mais difíceis de utilizar do que o sistema proposto

5 – Muito Melhor / O sistema proposto facilita a edição da ontologia, se comparado à interface de outras ferramentas

Alguns indicadores necessitam de um período extenso de uso da ferramenta para que possam ser medidos, ou seja, sua aplicabilidade só é possível a médio e longo prazo, após diversas interações colaborativas dos usuários com o sistema. Portanto, como o período de validação da ferramenta foi relativamente reduzido em relação ao tempo de construção da abordagem e da ferramenta, as métricas dos indicadores 3 e 4 não puderam ser medidas com efetividade. Por outro lado, o indicador 6 necessita de uma comparação com um desenvolvimento manual da ontologia, dados indisponíveis no momento da validação deste trabalho. Por conta destes obstáculos, acreditamos que o sistema de validação proposto seja de grande utilidade para que trabalhos futuros possam validar em detalhes esta proposta.

Com as respostas das perguntas, é possível construir uma tabela comparativa. As alternativas das questões estão relacionadas a um peso que, somados, leva a um valor total. Quanto maior o valor, melhor o desempenho da ferramenta proposta neste trabalho, em comparação ao processo manual. Os pesos são equivalentes aos números das alternativas. Por exemplo, em todas as perguntas, a alternativa 1 (um) possui peso 1 (um) e beneficia o processo manual, enquanto a alternativa 5 (cinco) possui peso 5 (cinco) e beneficia o processo colaborativo através da ferramenta. Dessa forma, para proceder a validação, sugerimos o uso da tabela 6.3, com dados hipotéticos, exibida abaixo:

Tabela 6.3 Resultado das questões de avaliação da ferramenta (dados hipotéticos).

Pontos	Usuário 1	Usuário 2	Usuário 3	Total
Pergunta 1	1	2	3	6
Pergunta 2	2	5	5	12
Pergunta 3	3	2	2	7
Pergunta 4	4	3	3	10
Pergunta 5	5	4	1	10
Pergunta 6	3	1	2	6
Pergunta 7	1	3	5	9

Nesse exemplo, o somatório total de pontos para as perguntas é 60. O total mínimo seria de 21 (assumindo que os três usuários responderam 1 para todas as perguntas), enquanto o total máximo seria de 105 (assumindo que os três usuários responderam 5 para todas as perguntas). Quanto mais próximo o total estiver do número 105, maior o benefício do uso da ferramenta colaborativa. Sugerimos também a criação de um limiar den-

tro desse intervalo (por exemplo: 80), para facilitar a identificação de que quando a ferramenta atingir este limiar, a mesma atende com qualidade os requisitos da construção colaborativa de ontologias.

Como experimento comparativo, também sugerimos a comparação da construção de uma ontologia no processo manual, com a ferramenta desenvolvida neste trabalho e com três outras ferramentas colaborativas atuais (por exemplo: Collaborative Protégé, WebODE e OntoEdit). Neste processo, seriam definidos 15 (quinze) usuários, divididos em grupos de 3 (três) usuários para cada abordagem/ferramenta. Cada grupo seria responsável por modelar uma ontologia, formalizando um mesmo problema conceitual proposto a todos os grupos (por exemplo, modelar a operação em uma livraria), dispondo do mesmo período de tempo para a tarefa. Após a modelagem, os grupos seriam submetidos às perguntas de avaliação, para a montagem da tabela comparativa 6.4:

Tabela 6.4 Somatório dos pontos das questões de avaliação da ferramenta, comparando diferentes abordagens/ferramentas de construção de ontologia (dados fictícios).

Pontos	Processo Manual	Obaitá	Collaborative Protégé	WebODE	OntoEdit
Total de Pontos	80	80	80	80	80

Adicionalmente, pode ser construída uma tabela comparativa para medir quantitativamente, em cada abordagem, os objetos ontológicos definidos na seção 6.2.1. Esta comparação pode ser vista na Tabela 6.5:

Tabela 6.5 Resultado quantitativo de objetos ontológicos e mudanças, comparando as diferentes abordagens/ferramentas de construção de ontologia (dados fictícios).

Pontos	Processo Manual	Obaitá	Collaborative Protégé	WebODE	OntoEdit
Nº de Total de Mudanças Efetuadas	1	1	1	1	1
Nº de Total de Conceitos	1	1	1	1	1
Nº de Total de Propriedades	1	1	1	1	1
Nº de Total de Relações	1	1	1	1	1
Nº de Total de Ícones	1	1	1	1	1
Nº de Conceitos Adicionados	1	1	1	1	1
Nº de Conceitos Excluídos	1	1	1	1	1
Nº de Conceitos Alterados	1	1	1	1	1

Nº de Propriedades Adicionadas	1	1	1	1	1
Nº de Propriedades Excluídas	1	1	1	1	1
Nº de Relações Adicionadas	1	1	1	1	1
Nº de Relações Excluídas	1	1	1	1	1
Nº de Ícones vinculados a Conceitos	1	1	1	1	1
Nº de Ícones vinculados a Valores de Propriedade	1	1	1	1	1
Nº de Alterações nas Metapropriedades dos Conceitos	1	1	1	1	1
Nº de Metatipos classificados manualmente pelos usuários	1	1	1	1	1

Após essas avaliações, será possível identificar com mais clareza os benefícios da ferramenta proposta neste trabalho, em comparação com as propostas atuais nesta área. No capítulo a seguir, apresentamos as conclusões deste trabalho e antecipamos alguns trabalhos futuros, como expansão dos objetivos já alcançados.

7 CONCLUSÃO

Ontologias são modelos conceituais explícitos oriundos do conhecimento compartilhado por um grupo de usuários focados em determinado domínio de interesse. O objetivo principal das ontologias é facilitar a comunicação de conhecimento entre os indivíduos de uma comunidade através da explicitação das relações semânticas entre conceitos da realidade, ou seja, estabelecer o conhecimento consensual de uma forma genérica e formal (Corcho et al. 2003).

Por se tratar de um artefato compartilhado e focado no consenso, uma ontologia somente é construída após diversas interações entre os participantes, contemplado diferentes pontos de vista e níveis de conhecimento dos especialistas. Em alguns casos, uma ontologia pode levar anos até chegar em um estado coeso o suficiente para ser utilizada como consulta formal ou para embasar sistemas computacionais. Esse tempo é devido à dificuldade de se encontrar consenso em uma comunidade, pois quanto mais conhecimento e restrições são impostas ao modelo pelos usuários, maior a necessidade por uma comunicação eficiente entre os mesmos. A comunicação permite a transferência de experiência entre os seres e a construção coletiva de soluções ampliando a capacidade domínio do ambiente.

Além disso, como vimos anteriormente, os domínios de informação não são estáticos: eles evoluem quando elementos inexistentes se tornam parte do domínio ou quando alguns elementos se tornam obsoletos (De León & Antonio 2009). Desta forma, o problema conceitual de classificação dos entes do mundo torna-se ainda mais complicado, pois cada indivíduo possui a sua própria abstração da realidade (a qual pode estar em constante atualização). Essa dificuldade agrava-se ainda mais quando diversas pessoas discutem sobre um mesmo modelo conceitual compartilhado, onde surgem muita divergência de ideias e conceitualizações.

O suporte por uma ferramenta de software torna o processo de construção de ontologias mais eficiente e produtivo, facilitando a comunicação entre os usuários e o armazenamento do conhecimento envolvido no processo. Entendemos que, com o advento da Web 2.0, a colaboração é uma tendência e que a união dos aspectos da cooperação em massa com a formalização semântica das ontologias é um passo importante a ser dado na área de Engenharia de Conhecimento. Dessa forma, propomos neste trabalho a construção de um sistema colaborativo para a criação de ontologias na web.

De acordo com Guizzardi (2005), a máxima conversacional para a construção de modelos conceituais é que a mensagem deve ser relevante, completa, clara, não-ambígua, breve, sem excesso de informação e verdadeira do ponto de vista do conhecimento do emissor. Dessa forma, não basta a criação de um modelo de conhecimento, mas este modelo deve ser semanticamente válido para que possa ser utilizado com

eficiência. Isso é feito comparando-se o nível de homomorfismo entre um objeto concreto da realidade e a uma representação formal e explícita em uma conceitualização, através da análise de propriedades que classificam o isomorfismo dos modelos conceituais: Lucidez, Correção, Laconicidade e Completeza.

Vimos na seção 2.2.1 que há problemas que surgem quando criamos diagramas com carência dessas propriedades: a) diagramas não-lúcidos ocorrem quando há uma sobrecarga nos construtos de especificação, ou seja, quando mais de um conceito do modelo mapeiam para o mesmo construto, gerando ambiguidade; b) diagramas não-corretos ocorrem quando existem construtos da especificação que não mapeiam para entidades na conceitualização, gerando um excesso de construtos; c) diagramas não-lacônicos ocorrem quando existem conceitos do modelo que são mapeados para mais de um construto na especificação, gerando redundância de construtos e uma complexidade desnecessária para a representação; d) diagramas incompletos ocorrem quando existem conceitos do modelo que não são mapeados para construtos, ou seja, falta expressividade de especificação para representar determinadas entidades da conceitualização.

Buscando orientar a criação de diagramas mais coesos, notamos que o uso de uma ontologia de fundamentação tem papel importante na obtenção do consenso comum, pois reduz as possibilidades de interpretação sobre o domínio através da categorização semântica dos conceitos e propriedades da ontologia. Uma ontologia de fundamentação tem o objetivo de estabelecer uma base para obter-se coerência nas negociações de significado oriundas da colaboração de indivíduos sobre um modelo conceitual. Em (Guizzardi & Wagner 2004) foi proposta uma ontologia de fundamentação unificada que provê fundamentação ontológica na construção de modelos conceituais. Os construtos propostos por Guizzardi guiam a construção do modelo de conhecimento, auxiliando na classificação taxonômica e das relações entre os conceitos. Dessa forma, diminui-se a ocorrência de ambiguidades e aumenta-se a precisão do modelo de domínio.

Abordamos o estado da arte dos editores de ontologia e introduzimos aspectos de colaboração que guiam o desenvolvimento de sistemas de colaboração. De fato, os editores de ontologias colaborativos atuais possuem diversas funcionalidades importantes como detecção de inconsistências, inferência, documentação, validação, consultas, integração, etc. Entretanto, muitos deles não podem ser acessados via navegador web, o que dificulta muito o processo colaborativo nos dias de hoje. Algumas abordagens utilizam aspectos da metodologia *OntoClean*, porém nenhuma faz uso de ontologias de fundamentação, o qual amplia a completude do modelo através do uso de construtos mais ricos, permitindo expressar melhor a realidade. Além disso, algumas aplicações são fortemente orientadas à linguagem de representação ou ao formalismo ontológico. Nesses casos, as ferramentas não abstraem informações técnicas para o especialista ou disponibilizam uma interface não intuitiva o suficiente para evitar que o usuário tenha receio de alterar a ontologia livremente. Isso cria barreiras muitas vezes intransponíveis na construção colaborativa de ontologias, dificultando a criação de modelos corretos.

Em nossa proposta, implementamos um sistema web colaborativo, acessível via navegador, o qual possibilita aos especialistas alterarem diretamente a ontologia, registrando as mudanças automaticamente. Utilizamos como base uma camada de metadados construída para fornecer construtos ontológicos de fundamentação para embasar as escolhas ontológicas através da expressividade semântica de uma ontologia

de fundamentação, além de artefatos de colaboração para permitir um mapeamento das mudanças ontológicas e discussões organizadas entre os membros da comunidade.

Um modelo de discussão estruturada também foi proposto, com base em soluções consolidadas como o DILIGENT e IBIS, tomando como base o uso de anotações vinculadas aos elementos da ontologia, às mudanças efetuadas no processo de colaboração ou à outras anotações. O mapeamento das discussões possibilita a consulta detalhada do histórico de conversação que acompanha as mudanças na ontologia de domínio e a detecção com precisão dos momentos onde são tomadas as decisões ontológicas. Essa é uma evolução em relação aos editores existentes, pois introduzimos uma maior gama de tipos de interações na discussão e restringimos o seu uso em relação à interação anterior, vinculando os argumentos de acordo com a sua semântica.

Em nossa proposta, a ferramenta possui funcionalidades que permitem que os usuários ou especialistas da comunidade definam conceitos, propriedades e relações, tornando explícito o seu significado intencional através do uso de primitivas de ontologia de fundamentação. Entretanto, a construção de ontologias é uma tarefa que demanda conhecimento sobre modelagem conceitual, o que traz dificuldade para que especialistas de domínio construam sozinhos ontologias bem fundamentadas. Dessa forma, em nossa proposta, é possível introduzir formalmente o papel de engenheiro de conhecimento como usuário atuante na colaboração, focado em monitorar, corrigir e garantir o melhor ancoramento simbólico entre os conceitos modelados e as primitivas de representação, gerando então modelos mais convergentes.

Existem domínios de informação onde o conhecimento visual é crucial para a sua completude. Nesses domínios imagísticos, o reconhecimento de padrões visuais é o processo inicial para capturar informação e dar suporte à resolução de problemas. Percebemos que nenhuma das aplicações para construção de ontologias analisadas aborda o uso de componentes imagísticos, o que mostramos ser uma grande barreira para a construção de ontologias em domínios visuais. O uso de primitivas visuais como imagens e ícones permite um maior entendimento do domínio onde a representação simbólica linguística não é suficiente para explicitar certos conhecimentos. Um dos objetivos da ferramenta aqui proposta foi fornecer construtos ontológicos para representar conhecimento visual e dar suporte a domínios imagísticos na construção colaborativa de ontologias. Dessa forma, a ferramenta busca prover suporte ao especialista que necessita construir um modelo conceitual de conhecimento visual e, por outro lado, tirar vantagem da representação visual e de imagens para ajudar a expressar o significado integral dos conceitos.

Este trabalho focou em aproveitar o fenômeno da colaboração coletiva como ferramenta de apoio à construção de ontologias, levando em consideração a importância da fundamentação ontológica e de suporte a conhecimento visual. Para embasar e formalizar as interações no ambiente colaborativo, utilizamos como base conceitual de modelagem o modelo de colaboração 3C (Fuks et al. 2003). Este modelo defende que, para colaborar, os indivíduos devem trocar informações (**comunicar-se**), organizar-se (**coordenar-se**) e operar em conjunto num espaço compartilhado (**cooperar-se**). Esses conceitos serviram de base conceitual para a implementação de ferramentas específicas no sistema e para a abordagem colaborativa como um todo.

Devido à diversos fatores, alguns aspectos não puderam ser levados em consideração neste trabalho. Sugerimos a abordagem desses aspectos como trabalhos futuros. Acreditamos que é possível otimizar o suporte à seleção de construtos, auxiliando os usuários a selecionar os construtos corretos de representação bem fundamentada, diminuindo ainda mais a necessidade de conhecimento conceitual e formal. Também almejamos expandir a abrangência dos construtos de fundamentação, contemplando também a ontologia de fundamentação UFO-B, a qual visa sistematizar conceitos como estados, processos, eventos, relações temporais, entre outros. A ferramenta em si também permite muitas expansões, como a implementação de fluxos de trabalho (workflow), para organizar e coordenar os usuários nas tarefas de modificação da ontologia, revisão de mudanças, votação estruturada, etc. Essas tarefas podem ser coordenadas através de um módulo que delegue tarefas aos responsáveis de acordo com critérios bem definidos. Também acreditamos ser interessante a criação de agentes computacionais para identificação de conflitos, duplicidades, inconsistências, envio de mensagens motivacionais aos usuários, etc. Outra evolução interessante na ferramenta seria a criação de um nível de confiança para cada usuário, estimado através do nº de mudanças efetuadas na ontologia sem mudanças posteriores, dividido pelo nº total de mudanças efetuadas pelo usuário. Dessa forma seria possível efetuar um ranking dos usuários mais “confiáveis” dentro do sistema, assumindo que os usuários que modificaram componentes que não foram mais alterados posteriormente, geraram modificações que alcançaram o consenso geral da comunidade. No que se refere à validação, alguns fatores dificultaram uma avaliação extensa da ferramenta, apesar de seus benefícios já ficarem explícitos através da própria abordagem conceitual e colaborativa. Dessa forma, propomos um instrumento organizado de validação para verificar se a construção de ontologias de domínio com o uso da ferramenta Obaitá é realmente mais eficiente do que utilizando outras ferramentas atuais ou através do processo manual

Por fim, a ferramenta desenvolvida traz importantes diferenciais em relação às ferramentas existentes, principalmente porque provê um ambiente colaborativo para a Web, focado na construção de modelos explícitos de conhecimento compartilhado, ou seja, ontologias de domínio, através do uso de construtos de representação precisos e expressivos, oriundos de conceitos de fundamentação ontológica e componentes visuais. Dessa forma, pode-se dizer que este trabalho contribuiu com uma abordagem conceitual e uma ferramenta inovadoras, focadas em permitir a construção de ontologias cada vez mais ricas e aplicáveis ao uso comercial.

REFERÊNCIAS

ABEL, M. Estudo da perícia em petrografia sedimentar e sua importância para a engenharia de conhecimento. 2002. 239 f. Tese de Doutorado (Programa de Pós-Graduação em Computação) - Instituto de Informática - Universidade Federal do Rio Grande do Sul, Porto Alegre.

ADAMCZYK, P.D.; TWIDALE, M.B. Supporting multidisciplinary collaboration: requirements from novel HCI education. Proceedings of the SIGCHI conference on Human factors in computing systems, ACM, New York, USA, p. 1073–1076, 2007.

ARPÍREZ, J.C. et al. WebODE: a scalable workbench for ontological engineering. Proceedings of the 1st international conference on Knowledge capture., ACM, K-CAP '01, New York, USA, p. 6–13, 2001.

BERNERS-LEE, T. et al. The semantic web. Scientific American, vol. 279, no. 5, p. 35-43, 2001 [S.l.].

BORST, W.N. Construction of Engineering Ontologies for Knowledge Sharing and Re-use. Ph.D. Dissertation, Centre for Telematics and Information Technology, University of Twente, Enschede, The Netherlands, 1997.

CORCHO, O.; FERNANDEZ-LOPEZ, M.; GÓMEZ-PÉREZ, A. Methodologies, tools and languages for building ontologies. Where is their meeting point?. Data & Knowledge Engineering, 46(1), p.41–64, 2003 [S.l.].

DAVIES, J., VAN HARMELEN, F., FENSEL, D. Towards the semantic web: ontology-driven knowledge management, John Wiley & Sons, Inc, Chichester, England, 2002. 328 p.

DELLSCHAFT, K. et al. Tracking design rationale in collaborative ontology engineering. The Semantic Web: Research and Applications, Springer. Berlin Heidelberg, 2008. p.782–786.

FÜRST, F.; TRICHET, F. Heavyweight ontology engineering. Proceedings of the 5th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE'2006). Lecture Notes in Computer Science - LNCS 4277. Springer-Verlag, Berlin Heidelberg. p. 38–39.

FÁVERA, J.C.D. Fundamentos de estratigrafia moderna, 1.ed. Rio de Janeiro: EDUERJ, 2001. 264 p.

FENSEL, D. et al. On-to-knowledge: Ontology-based tools for knowledge management. Proceedings of the eBusiness and eWork, EMMSEC 2000, Madrid, Spain, 2000, p. 18–20.

FENSEL, D. Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce, 2nd ed. Springer Verlag, 2004. 162p.

FERNANDEZ-LOPEZ, M.; GÓMEZ-PÉREZ, A.; JURISTO, N. Methontology: from ontological art towards ontological engineering. Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering, Stanford, USA, 1997, p. 33–40.

FOX, M.S. The TOVE Project Towards a Common-Sense Model of the Enterprise. Lecture Notes in Computer Science, IEA/AIE, Springer, 1992, p.25–34.

FUKS, H., GEROSA, M., PIMENTEL, M. Projeto de Comunicação em Groupware: Desenvolvimento, Interface e Utilização. XXII Jornada de Atualização em Informática, Anais do XXIII Congresso da Sociedade Brasileira de Computação, 2003. p. 295–338.

GÓMEZ-PÉREZ, A.; FERNANDEZ-LOPEZ, M.; CORCHO, O. Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. 1rd ed. (Advanced Information and Knowledge Processing), Springer-Verlag New York, Inc, 2004. 420 p.

GRUBER, T.R. Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human Computer Studies, 43(5) , 1995, p.907–928 [S.I.].

GUARINO, N. Concepts, attributes and arbitrary relations: Some linguistic and ontological criteria for structuring knowledge bases. Data & Knowledge Engineering, 8(3) , 1992, p.249–261 [S.I.].

GUARINO, N., Formal Ontology in Information Systems. Proceedings of FOIS'98, Trento, Italy, 1998, p. 6-8.

GUARINO, N. Formal ontology, conceptual analysis and knowledge representation. International Journal of Human Computer Studies, 43(5), 1995, p.625–640 [S.I.].

GUIZZARDI G. et al. Ontologias de Fundamentação e Modelagem Conceitual, II Seminário de Pesquisa em Ontologia no Brasil, IME, Rio de Janeiro, 2009.

GUIZZARDI, G. et al. An Ontologically Well-Founded Profile for UML Conceptual Models. A. Persson, J. Stirna (eds.) Advanced Information Systems Engineering, Proceedings of 16th CAiSE Conference, Riga, Springer, 2004. p. 1–122.

GUIZZARDI, G. Ontological Foundations for Structural Conceptual Models. Enschede, The Netherlands: Universal Press, v.05-74. 2005. 410 p. (CTIT PhD Thesis Series), 2005.

GUIZZARDI, G.; WAGNER, G. A unified foundational ontology and some applications of it in business modeling. Workshop on Enterprise Modeling and Ontologies for

Interoperability, 16th International Conference on Advanced Information Systems Engineering (CAISE), Riga, 2004. p 129-143.

HAASE, P. et al. The Neon Ontology Engineering Toolkit. WWW 2008 Developers Track, 2008 [S.1].

HARTMANN, J. et al. DEMO - Design Environment for Metadata Ontologies. Proceedings of the 3rd European Semantic Web Conference, ESWC 2006. Volume 4011. Budva, Montenegro. Springer Berlin, 2006. p.427–441.

HENKE, J. Towards a Usable Group Editor for Ontologies. *The Semantic Web-ISWC 2006*, 2006, p.978–979 [S.1].

INSTITUTE OF ELECTRICAL AND ELECTRONIC ENGINEERING. IEEE Std 1074-1995: IEEE Standard for Developing Software Life Cycle Processes. IEEE Computer Society. New York, 1996.

USCHOLD, M.; JASPER, R. A framework for understanding and classifying ontology applications. Proceedings 12th Int. Workshop on Knowledge Acquisition, Modelling, and Management KAW'99, Banff, Alberta, Canada, 1999. p. 16–21.

JOHN, M.; MELSTER, R. Knowledge networks—managing collaborative knowledge spaces. *Advances in Learning Software Organizations*, 6th International Workshop, LSO 2004, Banff, Canada, Springer, 2004. p.165–171.

JONES, D.; BENCH-CAPON, T.; VISSER, P. Methodologies for ontology development. Proceedings ITi and KNOWS Conference of the 15th IFIP World Computer Congress, Chapman-Hall, Sydney, Australia, 1998. pp. 62–75.

KUNZ, W.; RITTEL, H.W.J. Issues as Elements of Information Systems. Working Paper No. 131, Studiengruppe für Systemforschung, Heidelberg, Germany, July 1970.

PALMA DE LEON, R. A. Ontology metadata management in distributed environment. Doctoral thesis. Universidad Politécnica de Madrid, Madrid, Spain, 2009.

LORENZATTI, A. Ontologia para Domínios Imagísticos: Combinando Primitivas Textuais e Pictóricas. 2009. 117 f. Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul, Porto Alegre.

LORENZATTI, A. et al. Ontological Primitives for Visual Knowledge. *Advances in Artificial Intelligence SBIA 2010*. São Bernardo do Campo: Springer, v.6404, 2011. p.1–10.

MANGIONE, G.R.et al. A Pedagogical Approach for Collaborative Ontologies Building. *Technology-Enhanced Systems and Tools for Collaborative Learning Scaffolding*, 350/2011, 2011. p. 135-166.

NOY, N.F.; MCGUINNESS, D.L. Ontology development 101: A guide to creating your first ontology, KSL Technical Report, 2001 [S.1].

NOY, N.F.; TUDORACHE, T. Collaborative ontology development on the (semantic) web. AAAI Spring Symposium on Semantic Web and Knowledge Engineering (SWKE), Stanford, CA, 2008.

OLIVEIRA, L. et al. Collective construction of Information Science through a Web 2.0 environment. International Workshop on Metamodels, Ontologies and Semantic Technologies, ONTOBRAS-MOST 2011, Gramado, Rio Grande do Sul, 2011.

PRESS, N. Understanding Metadata. NISO Press Booklets, US, ISBN, 1-880124-62-9, 2004 [S.I.].

RAUTENBERG, S. et al. ontoKEM: uma ferramenta para construção e documentação de ontologias. Seminário de Pesquisa em ontologia no Brasil, Niterói, Rio de Janeiro, 2008.

RICHARDS, D. A social software/Web 2.0 approach to collaborative knowledge engineering. Information Sciences, 179(15) , 2009. p.2515-2523.

SCHREIBER, G. Knowledge Engineering and Management: The CommonKADS Methodology, MITPress, ISBN 0262193000. 2000.

STUDER, R.; BENJAMINS, V.R.; FENSEL, D. Knowledge engineering: principles and methods. Data & knowledge engineering, 25(1-2), 1998, p.161-197.

SURE, Y. et al. OntoEdit: Collaborative ontology development for the semantic web. The Semantic Web—ISWC 2002, 2002, p.221-235.

SURE, Y. et al. On-To-Knowledge Methodology. Handbook on Ontologies, International Handbooks on Information Systems, Springer, 2003 [S.I]. p. 117-132.

TUDORACHE, T. et al. Supporting collaborative ontology development in Protégé. The Semantic Web-ISWC 2008, 2008, p.17-32.

USCHOLD, M.; GRUNINGER, M. Ontologies: Principles, methods and applications. The Knowledge Engineering Review, 11(02) , 1996, p.93-136.

ANEXO EXEMPLO DE CONSULTAS SQL COM TRIPLAS

Como exemplo, podemos buscar todas as triplas da ontologia de domínio “**dominio_estratigrafia_versao_1**” com a consulta SQL abaixo:

```
SELECT *
FROM triples
WHERE
graph = "http://localhost/obaita/#dominio_estratigrafia_versao_1"
```

Da mesma forma, podemos buscar todas as triplas que descrevem a colaboração efetuada sobre a ontologia de domínio “**dominio_estratigrafia_versao_1**” com a consulta SQL abaixo:

```
SELECT *
FROM triples
WHERE
graph = "http://localhost/obaita/#dominio_estratigrafia_versao_1_Collaboration"
```

Podemos, por exemplo, buscar todas as URIs de mudanças que foram efetuadas por determinado usuário com a consulta abaixo:

```
SELECT *
FROM triples
WHERE
graph = "http://localhost/obaita/#dominio_estratigrafia_versao_1_Collaboration"
```

Em alguns casos, a junção da tabela de triplas com ela mesma é necessária. Podemos, por exemplo, buscar todas as URIs de mudanças que foram efetuadas por determinado usuário com a consulta SQL abaixo:

```
SELECT n1.subject
FROM triples n1,triples n2
WHERE n1.subject = n2.subject
AND n1.predicate = 'http://localhost/obaita/#domainComponent'
AND n2.predicate = 'http://localhost/obaita/#author'
AND n2.object = 'gabriel'
```

Da mesma forma, podemos buscar quais os tipos de mudanças e a data de cada mudança efetuada por determinado usuário. Para isso, necessitamos de mais dois JOINS entre as tabelas. A consulta ficaria como a seguir:

```

SELECT n1.subject as change_uri, n3.object as change_type,n4.object as change_date
FROM triples n1, triples n2, triples n3, triples n4
WHERE n1.subject = n2.subject
AND n2.subject = n3.subject
AND n3.subject = n4.subject
AND n1.predicate = 'http://localhost/obaita/#domainComponent'
AND n2.predicate = 'http://localhost/obaita/#author'
AND n2.object = 'gabriel'
AND n3.predicate = 'http://www.w3.org/1999/02/22-rdf-syntax-ns#type'
AND n4.predicate = 'http://localhost/obaita/#ValidDate'
AND n1.graph = 'http://localhost/obaita/#dominio_vn1_Collaboration'

```

Um exemplo do resultado dessa consulta pode ser visto na Figura abaixo.

change_uri text	change_type text	change_date text
http://localhost/obaita/#Change_32bcc692-3058-4ca7-95bb-5c62561ad61a	http://localhost/obaita/#ConceptPropertyRemoved	2011-12-11 20:07:43
http://localhost/obaita/#Change_abca211a-ec11-4e35-91ee-ea1fd257625e	http://localhost/obaita/#SubclassOfRelationCreated	2011-12-11 20:08:53
http://localhost/obaita/#Change_d4120a77-21ce-4562-b7ea-a59e0f17955f	http://localhost/obaita/#ConceptRelationCreated	2011-12-11 20:08:54
http://localhost/obaita/#Change_b8b517f4-9ff7-4f00-b265-49ef2f770c17	http://localhost/obaita/#SubclassOfRelationCreated	2011-12-11 20:13:16
http://localhost/obaita/#Change_a217c29c-1fae-40c3-b476-2e59dcc04854	http://localhost/obaita/#ConceptRelationCreated	2011-12-11 20:13:17
http://localhost/obaita/#Change_08d2c607-cf08-4950-8639-5acf60a60715	http://localhost/obaita/#ConceptRelationRemoved	2011-12-11 20:26:09
http://localhost/obaita/#Change_9b172b8a-cfc8-4c9c-9b72-e30bde436694	http://localhost/obaita/#ConceptRelationRemoved	2011-12-11 20:26:09
http://localhost/obaita/#Change_e0ddd3a-c64c-4929-a24d-3b44aedf756f	http://localhost/obaita/#ConceptRelationRemoved	2011-12-11 20:26:09
http://localhost/obaita/#Change_51edd389-99fd-46a7-b4c1-37c23ef96fbf	http://localhost/obaita/#ConceptRelationRemoved	2011-12-11 20:26:10
http://localhost/obaita/#Change_bab9e3f4-d701-49b6-a2a3-ef0d51be8af7	http://localhost/obaita/#ConceptRelationRemoved	2011-12-11 20:26:10
http://localhost/obaita/#Change_0b32a8ef-ce5d-448b-8523-6f6eaff1f687	http://localhost/obaita/#ConceptRelationRemoved	2011-12-11 20:26:10
http://localhost/obaita/#Change_383536bc-9969-43f2-8d56-fe673098be8a	http://localhost/obaita/#SubclassOfRelationCreated	2011-12-11 20:26:10
http://localhost/obaita/#Change_e7afe6c4-9dad-4914-bdf7-6834798f3cd6	http://localhost/obaita/#ConceptRelationCreated	2011-12-11 20:26:10
http://localhost/obaita/#Change_9e02aa48-c7e2-4465-a6ad-053f5f51f9cf	http://localhost/obaita/#ConceptRelationRemoved	2011-12-11 20:32:22
http://localhost/obaita/#Change_a5048645-4570-4ac1-9a42-ad8fe2d2525a	http://localhost/obaita/#ConceptRelationRemoved	2011-12-11 20:32:22
http://localhost/obaita/#Change_17955393-62bb-4a41-91e7-53b47c32443d	http://localhost/obaita/#ConceptRelationRemoved	2011-12-11 20:32:23
http://localhost/obaita/#Change_91e3d946-a9db-43ac-92db-f555fb728839	http://localhost/obaita/#ConceptRelationRemoved	2011-12-11 20:32:23
http://localhost/obaita/#Change_e30c9613-00f2-4605-bee0-d77d1a9788c0	http://localhost/obaita/#ConceptRelationRemoved	2011-12-11 20:32:23
http://localhost/obaita/#Change_19c6f1f4-7b97-46d4-a1ca-fa863394a09b	http://localhost/obaita/#ConceptRelationRemoved	2011-12-11 20:32:23

ANEXO META-ONTOLOGIA DE REPRESENTAÇÃO E COLABORAÇÃO EM OWL

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
  xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
  xmlns:swrl="http://www.w3.org/2003/11/swrl#"
  xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns="http://localhost/obaita/#"
  xml:base="http://localhost/obaita/">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="MemberOfRelation">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="PartWholeRelation"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="DataTypePropertyCreated">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="PropertyCreated"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="CarryIdentity">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="OntologyConceptMetaProperty"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="RigidSortalConcept">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="SortalConcept"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="OntologyPropertyValue">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="OntologyComponent"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="ComponentOfRelation">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#PartWholeRelation"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="SemiRigidMixinConcept">
    <rdfs:subClassOf>

```

```

    <owl:Class rdf:ID="NonRigidMixinConcept"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#PartWholeRelation">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="FormalRelation"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Review">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Annotation"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="ConceptDependenceChange">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="ConceptMetaPropertyChange"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="RelationTargetCardinalityChange">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="RelationChange"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Mixin">
  <rdfs:subClassOf rdf:resource="#SemiRigidMixinConcept"/>
</owl:Class>
<owl:Class rdf:ID="Disagree">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Annotation"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#ConceptMetaPropertyChange">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="ConceptChange"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="PropertyRemoved">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="PropertyChange"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Quantity">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="SubstanceSortal"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="SubCollectionOfRelationCreated">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="PartOfRelationCreated"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="PropertyTypeChange">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#PropertyChange"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Change">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="CooperationEvent"/>
  </rdfs:subClassOf>

```

```

</owl:Class>
<owl:Class rdf:ID="RigidMixin">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="MixinConcept"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="AntiRigidMixinConcept">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#NonRigidMixinConcept"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class>
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:ID="User"/>
    <owl:Class rdf:about="#PropertyChange"/>
  </owl:unionOf>
</owl:Class>
<owl:Class rdf:ID="Kind">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#SubstanceSortal"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="RelationSourceCardinalityChange">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#RelationChange"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="PropertyValueChange">
  <rdfs:subClassOf rdf:resource="#Change"/>
</owl:Class>
<owl:Class rdf:ID="Unity">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#OntologyConceptMetaProperty"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="ConceptSupplyIdentityChange">
  <rdfs:subClassOf rdf:resource="#ConceptMetaPropertyChange"/>
</owl:Class>
<owl:Class rdf:about="#SortalConcept">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="SubstantialUniversalConcept"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Visitor">
  <rdfs:subClassOf rdf:resource="#User"/>
</owl:Class>
<owl:Class rdf:ID="ConceptCarryIdentityChange">
  <rdfs:subClassOf rdf:resource="#ConceptMetaPropertyChange"/>
</owl:Class>
<owl:Class rdf:ID="Agree">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Annotation"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="RoleMixin">
  <rdfs:subClassOf rdf:resource="#AntiRigidMixinConcept"/>
</owl:Class>
<owl:Class rdf:ID="PropertyDataTypeChange">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#PropertyChange"/>

```



```

</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="RelationTargetConceptChange">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#RelationChange"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="ReviewRequest">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Annotation"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="ConceptIconChange">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#ConceptChange"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Proposal">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Annotation"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="QualityStructure">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="OntologyProperty"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="KnowledgeEngineer">
  <rdfs:subClassOf rdf:resource="#User"/>
</owl:Class>
<owl:Class rdf:ID="VotingRequest">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Annotation"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="SubQuantityOfRelationCreated">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#PartOfRelationCreated"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="QualityUniversalPropertyCreated">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#PropertyCreated"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="OntologyImage">
  <rdfs:subClassOf rdf:resource="#OntologyComponent"/>
</owl:Class>
<owl:Class rdf:about="#PartOfRelationCreated">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="FormalRelationCreated"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="RelationRemoved">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#RelationChange"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="SubclassOfRelationCreated">
  <rdfs:subClassOf>

```

```

    <owl:Class rdf:about="#FormalRelationCreated"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Collective">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#SubstanceSortal"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="QualeIconChanged">
  <rdfs:subClassOf rdf:resource="#PropertyValueChange"/>
</owl:Class>
<owl:Class rdf:ID="Vote">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Annotation"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Mediator">
  <rdfs:subClassOf rdf:resource="#User"/>
</owl:Class>
<owl:Class rdf:ID="Quale">
  <rdfs:subClassOf rdf:resource="#OntologyPropertyValue"/>
</owl:Class>
<owl:Class rdf:about="#PropertyCreated">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#PropertyChange"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#SubstanceSortal">
  <rdfs:subClassOf rdf:resource="#RigidSortalConcept"/>
</owl:Class>
<owl:Class rdf:ID="Rigidity">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#OntologyConceptMetaProperty"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="ComponentOfRelationCreated">
  <rdfs:subClassOf rdf:resource="#PartOfRelationCreated"/>
</owl:Class>
<owl:Class rdf:about="#NonRigidMixinConcept">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#MixinConcept"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="ConceptPhotographyCreated">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="ConceptPhotographyChange"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Phase">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="AntiRigidSortalConcept"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#FormalRelation">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="OntologyRelation"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="SubKind">
  <rdfs:subClassOf rdf:resource="#RigidSortalConcept"/>

```

```

</owl:Class>
<owl:Class rdf:about="#OntologyProperty">
  <rdfs:subClassOf rdf:resource="#OntologyComponent"/>
</owl:Class>
<owl:Class rdf:ID="SubclassOfRelation">
  <rdfs:subClassOf rdf:resource="#FormalRelation"/>
</owl:Class>
<owl:Class rdf:ID="DomainExpert">
  <rdfs:subClassOf rdf:resource="#User"/>
</owl:Class>
<owl:Class rdf:ID="Icon">
  <rdfs:subClassOf rdf:resource="#OntologyImage"/>
</owl:Class>
<owl:Class rdf:ID="Dependence">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#OntologyConceptMetaProperty"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="QualeRemoved">
  <rdfs:subClassOf rdf:resource="#PropertyValueChange"/>
</owl:Class>
<owl:Class rdf:ID="RelationTypeChange">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#RelationChange"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="SubQuantityOfRelation">
  <rdfs:subClassOf rdf:resource="#PartWholeRelation"/>
</owl:Class>
<owl:Class rdf:ID="ConceptPropertyRemoved">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#ConceptChange"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Question">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Annotation"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#FormalRelationCreated">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="RelationCreated"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="ConceptRelationRemoved">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#ConceptChange"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="ConceptCreated">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#ConceptChange"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Photography">
  <rdfs:subClassOf rdf:resource="#OntologyImage"/>
</owl:Class>
<owl:Class rdf:ID="SupplyIdentity">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#OntologyConceptMetaProperty"/>

```

```

</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#ConceptPhotographyChange">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#ConceptChange"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="MaterialRelation">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#OntologyRelation"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Comment">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Annotation"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Answer">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Annotation"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#RelationCreated">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#RelationChange"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#ConceptChange">
  <rdfs:subClassOf rdf:resource="#Change"/>
</owl:Class>
<owl:Class rdf:about="#RelationChange">
  <rdfs:subClassOf rdf:resource="#Change"/>
</owl:Class>
<owl:Class rdf:ID="QualityDimension">
  <rdfs:subClassOf rdf:resource="#QualityStructure"/>
</owl:Class>
<owl:Class rdf:about="#OntologyConceptMetaProperty">
  <rdfs:subClassOf rdf:resource="#OntologyComponent"/>
</owl:Class>
<owl:Class rdf:ID="ConceptRigidityChange">
  <rdfs:subClassOf rdf:resource="#ConceptMetaPropertyChange"/>
</owl:Class>
<owl:Class rdf:ID="MaterialRelationCreated">
  <rdfs:subClassOf rdf:resource="#RelationCreated"/>
</owl:Class>
<owl:Class rdf:about="#OntologyRelation">
  <rdfs:subClassOf rdf:resource="#OntologyComponent"/>
</owl:Class>
<owl:Class rdf:about="#AntiRigidSortalConcept">
  <rdfs:subClassOf rdf:resource="#SortalConcept"/>
</owl:Class>
<owl:Class rdf:ID="LabelChange">
  <rdfs:subClassOf rdf:resource="#Change"/>
</owl:Class>
<owl:Class rdf:ID="ConceptPhotographyRemoved">
  <rdfs:subClassOf rdf:resource="#ConceptPhotographyChange"/>
</owl:Class>
<owl:Class rdf:ID="MemberOfRelationCreated">
  <rdfs:subClassOf rdf:resource="#PartOfRelationCreated"/>
</owl:Class>

```

```

<owl:Class rdf:ID="Role">
  <rdfs:subClassOf rdf:resource="#AntiRigidSortalConcept"/>
</owl:Class>
<owl:Class rdf:about="#SubstantialUniversalConcept">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="OntologyConcept"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#MixinConcept">
  <rdfs:subClassOf rdf:resource="#SubstantialUniversalConcept"/>
</owl:Class>
<owl:Class rdf:about="#OntologyConcept">
  <rdfs:subClassOf rdf:resource="#OntologyComponent"/>
</owl:Class>
<owl:Class rdf:ID="QualeCreated">
  <rdfs:subClassOf rdf:resource="#PropertyValueChange"/>
</owl:Class>
<owl:Class rdf:about="#PropertyChange">
  <rdfs:subClassOf rdf:resource="#Change"/>
</owl:Class>
<owl:Class rdf:ID="SubColletionOfRelation">
  <rdfs:subClassOf rdf:resource="#PartWholeRelation"/>
</owl:Class>
<owl:Class rdf:ID="ConceptTypeChange">
  <rdfs:subClassOf rdf:resource="#ConceptChange"/>
</owl:Class>
<owl:Class rdf:ID="Explanation">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Annotation"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="ConceptRelationCreated">
  <rdfs:subClassOf rdf:resource="#ConceptChange"/>
</owl:Class>
<owl:Class rdf:ID="ConceptRemoved">
  <rdfs:subClassOf rdf:resource="#ConceptChange"/>
</owl:Class>
<owl:Class rdf:ID="ConceptPropertyCreated">
  <rdfs:subClassOf rdf:resource="#ConceptChange"/>
</owl:Class>
<owl:Class rdf:about="#Annotation">
  <rdfs:subClassOf rdf:resource="#CooperationEvent"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="Label_pt">
  <rdfs:domain rdf:resource="#OntologyComponent"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="changes">
  <rdfs:range rdf:resource="#Change"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="applyToChange">
  <rdfs:range rdf:resource="#Change"/>
  <rdfs:domain rdf:resource="#Explanation"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="anchors">
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Quale"/>
        <owl:Class rdf:about="#OntologyConcept"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
</owl:ObjectProperty>

```

```

    </owl:Class>
  </rdfs:range>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="qualeOf">
  <rdfs:domain rdf:resource="#Quale"/>
  <rdfs:range rdf:resource="#QualityStructure"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="ObjectProperty_11"/>
<owl:ObjectProperty rdf:ID="Label_en">
  <rdfs:domain rdf:resource="#OntologyComponent"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasPhotos">
  <rdfs:range rdf:resource="#Photography"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="targetConcept">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain rdf:resource="#OntologyRelation"/>
  <rdfs:range rdf:resource="#OntologyConcept"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="annotations">
  <rdfs:range rdf:resource="#Annotation"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="photographyOf">
  <rdfs:domain rdf:resource="#Photography"/>
  <rdfs:range rdf:resource="#OntologyConcept"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasIcon">
  <rdfs:range rdf:resource="#Icon"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#OntologyConcept"/>
        <owl:Class rdf:about="#Quale"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="properties">
  <rdfs:range rdf:resource="#OntologyProperty"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="date">
  <rdfs:domain rdf:resource="#Annotation"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#dateTime"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="sourceCardinality">
  <rdfs:domain rdf:resource="#OntologyRelation"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="confidenceLevel">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Agree"/>
        <owl:Class rdf:about="#Disagree"/>
        <owl:Class rdf:about="#Answer"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>

```

```

    </owl:unionOf>
  </owl:Class>
</rdfs:domain>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="uri">
  <rdfs:domain rdf:resource="#CooperationEvent"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="valid_date">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#dateTime"/>
  <rdfs:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="targetCardinality">
  <rdfs:domain rdf:resource="#OntologyRelation"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="imageURI">
  <rdfs:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Photography"/>
        <owl:Class rdf:about="#OntologyImage"/>
        <owl:Class rdf:about="#User"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
</owl:DatatypeProperty>
<owl:FunctionalProperty rdf:ID="applyToVotingRequest">
  <rdfs:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:range rdf:resource="#VotingRequest"/>
  <rdfs:domain rdf:resource="#Vote"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="language">
  <rdfs:domain rdf:resource="#LabelChange"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="password">
  <rdfs:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#User"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="author">
  <rdfs:domain rdf:resource="#CooperationEvent"/>
  <rdfs:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:range rdf:resource="#User"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="username">
  <rdfs:domain rdf:resource="#User"/>
  <rdfs:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="type">
  <rdfs:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="applyToReviewRequest">
  <rdfs:domain rdf:resource="#Review"/>

```

```

<rdfs:range rdf:resource="#ReviewRequest"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="supplyIdentity">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="rangeConcept">
<rdfs:range rdf:resource="#OntologyConcept"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="rejected">
<rdfs:domain rdf:resource="#Change"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="applyToQuestion">
<rdfs:domain rdf:resource="#Answer"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="domainComponent">
<rdfs:domain rdf:resource="#Change"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
<rdfs:range rdf:resource="#OntologyComponent"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="hasDescription">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="applyTo">
<rdfs:range>
<owl:Class>
<owl:unionOf rdf:parseType="Collection">
<owl:Class rdf:about="#OntologyComponent"/>
<owl:Class rdf:about="#Annotation"/>
<owl:Class rdf:about="#Change"/>
</owl:unionOf>
</owl:Class>
</rdfs:range>
<rdfs:domain>
<owl:Class>
<owl:unionOf rdf:parseType="Collection">
<owl:Class rdf:about="#Comment"/>
<owl:Class rdf:about="#Question"/>
<owl:Class rdf:about="#Proposal"/>
<owl:Class rdf:about="#ReviewRequest"/>
<owl:Class rdf:about="#Agree"/>
<owl:Class rdf:about="#Disagree"/>
</owl:unionOf>
</owl:Class>
</rdfs:domain>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="score">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
<rdfs:domain rdf:resource="#Vote"/>
<rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="datatype">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>

```



```

<rdfs:range>
  <owl:DataRange>
    <owl:oneOf rdf:parseType="Resource">
      <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >int</rdf:first>
      <rdf:rest rdf:parseType="Resource">
        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >float</rdf:first>
        <rdf:rest rdf:parseType="Resource">
          <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >string</rdf:first>
          <rdf:rest rdf:parseType="Resource">
            <rdf:rest rdf:parseType="Resource">
              <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
              >date</rdf:first>
              <rdf:rest rdf:parseType="Resource">
                <rdf:rest rdf:parseType="Resource">
                  <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
                  >time</rdf:first>
                  <rdf:rest rdf:parseType="Resource">
                    <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
                    <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
                    >qualia</rdf:first>
                  </rdf:rest>
                </rdf:rest>
              </rdf:rest>
            </rdf:rest>
          </rdf:rest>
        </rdf:rest>
      </rdf:rest>
    </owl:oneOf>
  </owl:DataRange>
</rdfs:range>
<rdfs:domain rdf:resource="#QualityStructure"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="value">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Change"/>
        <owl:Class rdf:about="#Annotation"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="applyToProposal">
  <rdfs:domain rdf:resource="#VotingRequest"/>
  <rdfs:range rdf:resource="#Proposal"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:FunctionalProperty>
<owl:DataRange>
  <owl:oneOf rdf:parseType="Resource">
    <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >true</rdf:first>
  </owl:oneOf>
</owl:DataRange>

```

```

<rdf:rest rdf:parseType="Resource">
  <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
  <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
  >false</rdf:first>
</rdf:rest>
</owl:oneOf>
</owl:DataRange>
<owl:DataRange>
  <owl:oneOf rdf:parseType="Resource">
    <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >true</rdf:first>
    <rdf:rest rdf:parseType="Resource">
      <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
      >false</rdf:first>
      <rdf:rest rdf:parseType="Resource">
        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
        >false</rdf:first>
        <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
      </rdf:rest>
    </rdf:rest>
  </owl:oneOf>
</owl:DataRange>
<owl:DataRange>
  <owl:oneOf rdf:parseType="Resource">
    <rdf:rest rdf:parseType="Resource">
      <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
      >false</rdf:first>
      <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
    </rdf:rest>
    <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >true</rdf:first>
  </owl:oneOf>
</owl:DataRange>
<owl:DataRange>
  <owl:oneOf rdf:parseType="Resource">
    <rdf:rest rdf:parseType="Resource">
      <rdf:rest rdf:parseType="Resource">
        <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
        >false</rdf:first>
      </rdf:rest>
      <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
      >false</rdf:first>
    </rdf:rest>
    <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
    >true</rdf:first>
  </owl:oneOf>
</owl:DataRange>
</rdf:RDF>

```

ANEXO AMOSTRA DO HISTÓRICO DA COLABORAÇÃO

Uma porção do histórico de colaboração foi anexado para exibir as interações efetuadas na ontologia de domínio. O namespace das URIs “<http://localhost/obaita/#>” foi omitido por questões de espaço. As URIs também foram comprimidas para facilitar a visualização da tabela.

Data da Mudança	Autor	Tipo de Mudança	Conceito Destino	Novo Valor da Mudança
2011-08-16 09:18:40	ariane	LabelChange	Quale_b76b036d-0067-444a-80dc-8e11fcf527d9	Planar
2011-08-16 09:18:40	ariane	LabelChange	Quale_f960da76-2fd6-4015-a307-b892c644067a	Wavy lamination
2011-08-16 09:18:40	ariane	LabelChange	Quale_29fd47cc-0256-4ebd-aecb-6604ff04687a	Truncated Wavy Lamination
2011-08-16 09:18:41	ariane	LabelChange	Quale_e6a3f820-82a8-4f28-9506-35569d45f764	Sigmoidal Cross-strata
2011-08-16 09:18:41	ariane	LabelChange	Quale_0ea1617f-5b9f-4e6a-8b19-572c591fb965	Trough Cross-strata
2011-08-16 09:18:41	ariane	LabelChange	Quale_d7343a42-34a8-40fb-87aa-d4f3ddc1fa50	Horizontal Lamination
2011-08-30 14:19:58	ariane	LabelChange	Quale_b76b036d-0067-444a-80dc-8e11fcf527d9	Planar
2011-08-16 09:23:33	ariane	LabelChange	Quale_b76b036d-0067-444a-80dc-8e11fcf527d9	Planar
2011-08-16 09:23:33	ariane	LabelChange	Quale_f960da76-2fd6-4015-a307-b892c644067a	Wavy lamination
2011-08-16 09:23:34	ariane	LabelChange	Quale_29fd47cc-0256-4ebd-aecb-6604ff04687a	Truncated Wavy Lamination
2011-08-16 09:23:34	ariane	LabelChange	Quale_e6a3f820-82a8-4f28-9506-35569d45f764	Sigmoidal Cross-strata
2011-08-16 09:23:34	ariane	LabelChange	Quale_0ea1617f-5b9f-4e6a-8b19-572c591fb965	Trough Cross-strata
2011-08-16 09:23:34	ariane	LabelChange	Quale_d7343a42-34a8-40fb-87aa-d4f3ddc1fa50	Horizontal Lamination
2011-08-16 09:26:36	ariane	LabelChange	Quale_b76b036d-0067-444a-80dc-8e11fcf527d9	Planar
2011-08-16 09:26:36	ariane	LabelChange	Quale_f960da76-2fd6-4015-a307-b892c644067a	Wavy lamination
2011-08-16 09:26:36	ariane	LabelChange	Quale_29fd47cc-0256-4ebd-aecb-6604ff04687a	Truncated Wavy Lamination
2011-08-16 09:26:37	ariane	LabelChange	Quale_e6a3f820-82a8-4f28-9506-35569d45f764	Sigmoidal Cross-strata
2011-08-16 09:26:37	ariane	LabelChange	Quale_0ea1617f-5b9f-4e6a-8b19-572c591fb965	Trough Cross-strata
2011-08-16 09:26:37	ariane	LabelChange	Quale_d7343a42-34a8-40fb-87aa-d4f3ddc1fa50	Horizontal Lamination
2011-08-30 14:19:58	ariane	PropertyValueIConChanged	Quale_b76b036d-0067-444a-80dc-8e11fcf527d9	Icon_4240b038-1fba-43fd-83fa-8884284a64cf
2011-08-30 14:19:58	ariane	LabelChange	Quale_f960da76-2fd6-4015-a307-b892c644067a	Wavy lamination
2011-08-30 14:19:58	ariane	LabelChange	Quale_29fd47cc-0256-4ebd-aecb-6604ff04687a	Truncated Wavy Lamination

2011-08-16 10:02:31	ariane	LabelChange	Quale_b76b036d-0067-444a-80dc-8e11fc527d9	Planar
2011-08-16 10:02:31	ariane	LabelChange	Quale_f960da76-2fd6-4015-a307-b892c644067a	Wavy lamination
2011-08-16 10:02:31	ariane	PropertyValueIconChanged	Quale_f960da76-2fd6-4015-a307-b892c644067a	Icon_16c6a193-8a6a-4465-80fc-492c147a7c5b
2011-08-16 10:02:31	ariane	LabelChange	Quale_29fd47cc-0256-4ebd-aecb-6604ff04687a	Truncated Wavy Lamination
2011-08-16 10:02:31	ariane	LabelChange	Quale_e6a3f820-82a8-4f28-9506-35569d45f764	Sigmoidal Cross-strata
2011-08-16 10:02:31	ariane	LabelChange	Quale_0ea1617f-5b9f-4e6a-8b19-572c591fb965	Trough Cross-strata
2011-08-16 10:02:31	ariane	LabelChange	Quale_d7343a42-34a8-40fb-87aa-d4f3ddc1fa50	Horizontal Lamination
2011-08-30 14:19:58	ariane	LabelChange	Quale_e6a3f820-82a8-4f28-9506-35569d45f764	Sigmoidal Cross-strata
2011-08-16 10:43:10	ariane	LabelChange	Quale_b76b036d-0067-444a-80dc-8e11fc527d9	Planar
2011-08-16 10:43:11	ariane	LabelChange	Quale_f960da76-2fd6-4015-a307-b892c644067a	Wavy lamination
2011-08-16 10:43:11	ariane	PropertyValueIconChanged	Quale_f960da76-2fd6-4015-a307-b892c644067a	Icon_16c6a193-8a6a-4465-80fc-492c147a7c5b
2011-08-16 10:43:11	ariane	LabelChange	Quale_29fd47cc-0256-4ebd-aecb-6604ff04687a	Truncated Wavy Lamination
2011-08-16 10:43:11	ariane	LabelChange	Quale_e6a3f820-82a8-4f28-9506-35569d45f764	Sigmoidal Cross-strata
2011-08-16 10:43:11	ariane	LabelChange	Quale_0ea1617f-5b9f-4e6a-8b19-572c591fb965	Trough Cross-strata
2011-08-16 10:43:11	ariane	LabelChange	Quale_d7343a42-34a8-40fb-87aa-d4f3ddc1fa50	Horizontal Lamination
2011-08-30 14:19:58	ariane	LabelChange	Quale_0ea1617f-5b9f-4e6a-8b19-572c591fb965	Trough Cross-strata
2011-08-30 14:19:58	ariane	LabelChange	Quale_d7343a42-34a8-40fb-87aa-d4f3ddc1fa50	Horizontal Lamination
2011-08-16 15:07:33	ariane	LabelChange	Quale_b76b036d-0067-444a-80dc-8e11fc527d9	Planar
2011-08-16 15:07:33	ariane	LabelChange	Quale_f960da76-2fd6-4015-a307-b892c644067a	Wavy lamination
2011-08-16 15:07:33	ariane	PropertyValueIconChanged	Quale_f960da76-2fd6-4015-a307-b892c644067a	Icon_16c6a193-8a6a-4465-80fc-492c147a7c5b
2011-08-16 15:07:33	ariane	LabelChange	Quale_29fd47cc-0256-4ebd-aecb-6604ff04687a	Truncated Wavy Lamination
2011-08-16 15:07:33	ariane	LabelChange	Quale_e6a3f820-82a8-4f28-9506-35569d45f764	Sigmoidal Cross-strata
2011-08-16 15:07:33	ariane	LabelChange	Quale_0ea1617f-5b9f-4e6a-8b19-572c591fb965	Trough Cross-strata
2011-08-16 15:07:34	ariane	LabelChange	Quale_d7343a42-34a8-40fb-87aa-d4f3ddc1fa50	Horizontal Lamination
2011-08-30 14:21:19	ariane	LabelChange	Quale_b76b036d-0067-444a-80dc-8e11fc527d9	Planar
2011-06-14 22:49:00	eduardo	ConceptPropertyRemoved	imported1649620127	imported907493114
2011-06-14 22:57:31	eduardo	ConceptPropertyCreated	imported1649620127	Property_8841424c-73e5-4931-a166-a148cfd098da
2011-06-14 23:05:26	eduardo	LabelChange	Quale_e7fe8354-3831-411f-bd5e-b789342697b8	Low Angle
2011-06-14 23:05:26	eduardo	LabelChange	Quale_f142196c-c608-4d6b-92a1-0a9070086fb4	High Angle
2011-06-14 23:05:26	eduardo	LabelChange	Quale_ec261066-9f9c-423c-b387-267523f2db22	Horizontal
2011-08-16 15:21:34	ariane	LabelChange	Quale_b76b036d-0067-444a-80dc-8e11fc527d9	Planar
2011-06-17 11:28:10	eduardo	ConceptPropertyRemoved	imported1649620127	imported-850224997
2011-06-17 11:28:59	eduardo	ConceptPropertyRemoved	imported1649620127	Property_417db8bd-b5c2-4a96-aa31-74191746aff4
2011-06-17 20:16:17	ariane	LabelChange	Concept_b7ec6f3e-fc1c-497c-bcde-75a35a5a250d	Test
2011-08-16 15:21:34	ariane	LabelChange	Quale_f960da76-2fd6-4015-a307-b892c644067a	Wavy lamination

2011-08-16 15:21:34	ariane	PropertyValuelconChanged	Quale_f960da76-2fd6-4015-a307-b892c644067a	Icon_16c6a193-8a6a-4465-80fc-492c147a7c5b
2011-08-16 15:21:34	ariane	LabelChange	Quale_29fd47cc-0256-4ebd-aecb-6604ff04687a	Truncated Wavy Lamination
2011-08-16 15:21:34	ariane	LabelChange	Quale_e6a3f820-82a8-4f28-9506-35569d45f764	Sigmoidal Cross-strata
2011-08-16 15:21:34	ariane	LabelChange	Quale_0ea1617f-5b9f-4e6a-8b19-572c591fb965	Trough Cross-strata
2011-06-27 16:14:08	eduardo	TypeChange	imported2134121411	MixinConcept
2011-08-16 15:21:34	ariane	LabelChange	Quale_d7343a42-34a8-40fb-87aa-d4f3ddc1fa50	Horizontal Lamination
2011-08-30 14:21:19	ariane	PropertyValuelconChanged	Quale_b76b036d-0067-444a-80dc-8e11fcf527d9	Icon_c606b97b-4634-461d-8869-575c7dab624e
2011-08-30 14:21:19	ariane	LabelChange	Quale_f960da76-2fd6-4015-a307-b892c644067a	Wavy lamination
2011-07-06 11:46:12	eduardo	ConceptMetaPropertyRigidityChange	Concept_3627ff19-69f6-4a3e-8b04-f97216eec1b4	1
2011-07-06 11:47:02	eduardo	ConceptMetaPropertyDependenceChange	Concept_3627ff19-69f6-4a3e-8b04-f97216eec1b4	0
2011-07-06 11:47:09	eduardo	ConceptMetaPropertySupplyIdentityChange	Concept_3627ff19-69f6-4a3e-8b04-f97216eec1b4	1
2011-07-06 11:47:16	eduardo	ConceptMetaPropertyCarryIdentityChange	Concept_3627ff19-69f6-4a3e-8b04-f97216eec1b4	1
2011-08-16 17:41:05	ariane	LabelChange	Quale_b76b036d-0067-444a-80dc-8e11fcf527d9	Planar
2011-08-16 17:41:05	ariane	LabelChange	Quale_f960da76-2fd6-4015-a307-b892c644067a	Wavy lamination
2011-08-16 17:41:05	ariane	PropertyValuelconChanged	Quale_f960da76-2fd6-4015-a307-b892c644067a	Icon_10089d4d-a5b2-4f8d-b2b8-e8bf3c4966d3
2011-07-06 12:11:37	eduardo	ConceptMetaPropertyRigidityChange	imported1649620127	1
2011-07-19 14:28:48	eduardo	LabelChange	imported-1638108575	TractiveCurrentStructure
2011-07-19 14:29:25	eduardo	LabelChange	Quale_83e98a02-3716-410f-84a3-377ac9987fb5	Medium
2011-07-19 14:29:25	eduardo	LabelChange	Quale_9808c3ec-6b6a-42e8-a7ea-dfed36fff156	Low
2011-07-19 14:29:25	eduardo	LabelChange	Quale_848292a7-3996-4b4e-9962-26a54a6f191f	High
2011-07-19 14:29:25	eduardo	LabelChange	Quale_b2002985-d90a-48eb-b859-d28202b89b4f	Very Low
2011-07-19 14:29:37	eduardo	LabelChange	Quale_9808c3ec-6b6a-42e8-a7ea-dfed36fff156	Low
2011-07-19 14:29:37	eduardo	LabelChange	Quale_848292a7-3996-4b4e-9962-26a54a6f191f	High
2011-07-19 14:29:37	eduardo	LabelChange	Quale_b2002985-d90a-48eb-b859-d28202b89b4f	Very Low
2011-07-19 14:29:37	eduardo	LabelChange	Quale_83e98a02-3716-410f-84a3-377ac9987fb5	Medium
2011-07-19 14:33:26	eduardo	ConceptMetaPropertyRigidityChange	imported-1638108575	0
2011-08-16 17:41:05	ariane	LabelChange	Quale_29fd47cc-0256-4ebd-aecb-6604ff04687a	Truncated Wavy Lamination
2011-07-20 16:47:01	eduardo	LabelChange	Quale_4831f302-4cfd-4482-88c3-fff77e16bd8c	NE
2011-07-20 16:47:01	eduardo	LabelChange	Quale_972c2797-075c-4e20-8855-ba7be263af44	S
2011-07-20 16:47:01	eduardo	LabelChange	Quale_ce08b2db-9127-4357-9bba-818608c92553	SE
2011-07-20 16:47:01	eduardo	LabelChange	Quale_bb636d55-ddf3-464c-8e7f-bc43c13b3943	E
2011-07-20 16:47:01	eduardo	LabelChange	Quale_66422410-5cde-401e-98b3-fd40f5c18aec	N
2011-07-20 16:47:01	eduardo	LabelChange	Quale_5509c2f9-8981-4c57-8178-e8fe242e1791	SW
2011-07-20 16:47:01	eduardo	LabelChange	Quale_16f35822-29b3-4ef5-8e88-b87b1ae7c76b	W
2011-07-20 16:47:01	eduardo	LabelChange	Quale_c6fba24d-769e-4052-97d2-e10bad165856	NW

2011-08-16 17:41:05	ariane	LabelChange	Quale_e6a3f820-82a8-4f28-9506-35569d45f764	Sigmoidal Cross-strata
2011-08-16 17:41:05	ariane	LabelChange	Quale_0ea1617f-5b9f-4e6a-8b19-572c591fb965	Trough Cross-strata
2011-07-26 16:43:22	eduardo	LabelChange	Quale_a404b034-24f7-4c78-a37c-972a0ff906b4	Moderately Sorted
2011-07-26 16:43:22	eduardo	LabelChange	Quale_1b552e07-34f6-4eab-a5bd-97f24deff831	Well Sorted
2011-07-26 16:43:22	eduardo	LabelChange	Quale_3ab1c3e9-bf8f-41ee-bf93-4ee080b67171	Very Poorly Sorted
2011-07-26 16:43:22	eduardo	LabelChange	Quale_0461d1d3-ea53-4aa1-9548-66c1847cc58a	Very Well Sorted
2011-07-26 16:43:22	eduardo	LabelChange	Quale_76e6818b-2ef9-4220-9636-e2f40fb98a16	Poorly Sorted
2011-07-26 16:44:05	eduardo	LabelChange	Quale_a404b034-24f7-4c78-a37c-972a0ff906b4	Moderately Sorted
2011-07-26 16:44:05	eduardo	LabelChange	Quale_3ab1c3e9-bf8f-41ee-bf93-4ee080b67171	Very Poorly Sorted
2011-07-26 16:44:05	eduardo	LabelChange	Quale_0461d1d3-ea53-4aa1-9548-66c1847cc58a	Very Well Sorted
2011-07-26 16:44:05	eduardo	LabelChange	Quale_76e6818b-2ef9-4220-9636-e2f40fb98a16	Poorly Sorted
2011-07-26 16:44:06	eduardo	LabelChange	Quale_1b552e07-34f6-4eab-a5bd-97f24deff831	Well Sorted
2011-08-16 17:41:05	ariane	LabelChange	Quale_d7343a42-34a8-40fb-87aa-d4f3ddc1fa50	Horizontal Lamination
2011-08-01 16:04:15	eduardo	LabelChange	Property_5645e22e-7965-4838-8289-2d27831f29e3	temFormatoLamina
2011-08-01 16:04:16	eduardo	LabelChange	Quale_b76b036d-0067-444a-80dc-8e11fc527d9	Planar(PT)
2011-08-01 16:04:16	eduardo	LabelChange	Quale_f960da76-2fd6-4015-a307-b892c644067a	Wavy lamination(PT)
2011-08-01 16:04:16	eduardo	LabelChange	Quale_29fd47cc-0256-4ebd-aecb-6604ff04687a	Truncated Wavy Lamination(PT)
2011-08-01 16:04:16	eduardo	LabelChange	Quale_e6a3f820-82a8-4f28-9506-35569d45f764	Sigmoidal Cross-strata(PT)
2011-08-01 16:04:16	eduardo	LabelChange	Quale_0ea1617f-5b9f-4e6a-8b19-572c591fb965	Trough Cross-strata(PT)
2011-08-01 16:04:16	eduardo	LabelChange	Quale_d7343a42-34a8-40fb-87aa-d4f3ddc1fa50	Horizontal Lamination(PT)
2011-08-18 18:10:12	eduardo	ConceptMetaPropertyRigidityChange	Concept_3627ff19-69f6-4a3e-8b04-f97216eec1b4	2
2011-08-01 16:40:05	eduardo	LabelChange	imported2134121411	Estrutura Deposicional
2011-08-30 14:21:19	ariane	LabelChange	Quale_29fd47cc-0256-4ebd-aecb-6604ff04687a	Truncated Wavy Lamination
2011-08-09 14:35:20	ariane	LabelChange	Quale_b76b036d-0067-444a-80dc-8e11fc527d9	Planar
2011-08-09 14:35:20	ariane	LabelChange	Quale_f960da76-2fd6-4015-a307-b892c644067a	Wavy lamination
2011-08-09 14:35:20	ariane	LabelChange	Quale_29fd47cc-0256-4ebd-aecb-6604ff04687a	Truncated Wavy Lamination
2011-08-09 14:35:20	ariane	LabelChange	Quale_e6a3f820-82a8-4f28-9506-35569d45f764	Sigmoidal Cross-strata
2011-08-09 14:35:20	ariane	LabelChange	Quale_0ea1617f-5b9f-4e6a-8b19-572c591fb965	Trough Cross-strata
2011-08-09 14:35:20	ariane	LabelChange	Quale_d7343a42-34a8-40fb-87aa-d4f3ddc1fa50	Horizontal Lamination
2011-08-18 19:00:00	eduardo	ConceptMetaPropertyDependenceChange	Concept_3627ff19-69f6-4a3e-8b04-f97216eec1b4	1
2011-08-09 14:40:31	ariane	LabelChange	Quale_b76b036d-0067-444a-80dc-8e11fc527d9	Planar
2011-08-09 14:40:31	ariane	LabelChange	Quale_f960da76-2fd6-4015-a307-b892c644067a	Wavy lamination
2011-08-09 14:40:32	ariane	LabelChange	Quale_29fd47cc-0256-4ebd-aecb-6604ff04687a	Truncated Wavy Lamination
2011-08-09 14:40:32	ariane	LabelChange	Quale_e6a3f820-82a8-4f28-9506-35569d45f764	Sigmoidal Cross-strata
2011-08-09 14:40:32	ariane	LabelChange	Quale_0ea1617f-5b9f-4e6a-8b19-572c591fb965	Trough Cross-strata

2011-08-09 14:40:32	ariane	LabelChange	Quale_d7343a42-34a8-40fb-87aa-d4f3ddc1fa50	Horizontal Lamination
2011-08-18 19:04:23	eduardo	ConceptMetaPropertyDependenceChange	Concept_3627ff19-69f6-4a3e-8b04-f97216eec1b4	0
2011-08-18 19:04:29	eduardo	ConceptMetaPropertySupplyIdentityChange	Concept_3627ff19-69f6-4a3e-8b04-f97216eec1b4	0
2011-08-18 19:04:36	eduardo	ConceptMetaPropertyCarryIdentityChange	Concept_3627ff19-69f6-4a3e-8b04-f97216eec1b4	0
2011-08-30 14:21:19	ariane	LabelChange	Quale_e6a3f820-82a8-4f28-9506-35569d45f764	Sigmoidal Cross-strata
2011-08-09 14:44:52	ariane	LabelChange	Quale_a8326732-d3df-4142-8a46-21849ae79b2d	Small
2011-08-09 14:44:52	ariane	LabelChange	Quale_ea8b167b-be3d-4410-abfc-d06c152ad538	Midsized
2011-08-09 14:44:52	ariane	LabelChange	Quale_746e8f5f-6ea1-48f5-8180-a7e506e50390	Large
2011-08-09 14:46:27	ariane	LabelChange	Quale_ecc52c24-0e7d-4a23-9a03-6f827550dd91	NE-SW
2011-08-09 14:46:27	ariane	LabelChange	Quale_3c11779c-13f5-4908-9fab-fa5d8434ce09	NW-SE
2011-08-09 14:46:28	ariane	LabelChange	Quale_f269a9ff-324d-43cb-a2da-f494f3ea7638	N-S
2011-08-09 14:46:28	ariane	LabelChange	Quale_5052293b-ffff-4722-b78a-fbc2645e9e40	E-W
2011-08-09 14:49:25	ariane	LabelChange	Quale_ecc52c24-0e7d-4a23-9a03-6f827550dd91	NE-SW
2011-08-09 14:49:25	ariane	LabelChange	Quale_3c11779c-13f5-4908-9fab-fa5d8434ce09	NW-SE
2011-08-09 14:49:25	ariane	LabelChange	Quale_f269a9ff-324d-43cb-a2da-f494f3ea7638	N-S
2011-08-09 14:49:25	ariane	LabelChange	Quale_5052293b-ffff-4722-b78a-fbc2645e9e40	E-W
2011-08-09 15:09:47	ariane	LabelChange	Quale_a8326732-d3df-4142-8a46-21849ae79b2d	Small
2011-08-09 15:09:47	ariane	LabelChange	Quale_ea8b167b-be3d-4410-abfc-d06c152ad538	Midsized
2011-08-09 15:09:47	ariane	LabelChange	Quale_746e8f5f-6ea1-48f5-8180-a7e506e50390	Large
2011-08-18 19:10:51	eduardo	TypeChange	Concept_3627ff19-69f6-4a3e-8b04-f97216eec1b4	Mixin
2011-08-18 19:14:26	eduardo	LabelChange	Quale_621fd5ce-8022-4571-be9c-bd97757faca0	White
2011-08-18 19:14:26	eduardo	LabelChange	Quale_326b56e0-24aa-48a5-a2d2-17dbe5d5aac9	Yellow
2011-08-18 19:14:26	eduardo	LabelChange	Quale_96d5770d-da75-4d70-aa92-096ed1841e1f	Red
2011-08-18 19:14:26	eduardo	LabelChange	Quale_879dd065-c6d6-483a-b7a6-a380823eb24c	Gray
2011-08-09 15:19:44	ariane	LabelChange	Quale_b76b036d-0067-444a-80dc-8e11fcf527d9	Planar
2011-08-09 15:19:44	ariane	LabelChange	Quale_f960da76-2fd6-4015-a307-b892c644067a	Wavy lamination
2011-08-09 15:19:44	ariane	LabelChange	Quale_29fd47cc-0256-4ebd-aecb-6604ff04687a	Truncated Wavy Lamination
2011-08-09 15:19:44	ariane	LabelChange	Quale_e6a3f820-82a8-4f28-9506-35569d45f764	Sigmoidal Cross-strata
2011-08-09 15:19:45	ariane	LabelChange	Quale_0ea1617f-5b9f-4e6a-8b19-572c591fb965	Trough Cross-strata
2011-08-09 15:19:45	ariane	LabelChange	Quale_d7343a42-34a8-40fb-87aa-d4f3ddc1fa50	Horizontal Lamination
2011-08-18 19:14:26	eduardo	LabelChange	Quale_1da2e363-16ed-41c2-b554-0e56e0625c09	Black
2011-08-30 14:21:19	ariane	LabelChange	Quale_0ea1617f-5b9f-4e6a-8b19-572c591fb965	Trough Cross-strata
2011-08-30 14:21:19	ariane	LabelChange	Quale_d7343a42-34a8-40fb-87aa-d4f3ddc1fa50	Horizontal Lamination
2011-08-09 19:40:50	ariane	LabelChange	Quale_b76b036d-0067-444a-80dc-8e11fcf527d9	Planar
2011-08-09 19:40:51	ariane	LabelChange	Quale_f960da76-2fd6-4015-a307-b892c644067a	Wavy lamination

2011-08-09 19:40:51	ariane	LabelChange	Quale_29fd47cc-0256-4ebd-aecb-6604ff04687a	Truncated Wavy Lamination
2011-08-09 19:40:52	ariane	LabelChange	Quale_e6a3f820-82a8-4f28-9506-35569d45f764	Sigmoidal Cross-strata
2011-08-09 19:40:52	ariane	LabelChange	Quale_0ea1617f-5b9f-4e6a-8b19-572c591fb965	Trough Cross-strata