



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

Relatórios Técnicos
do Departamento de Informática Aplicada
da UNIRIO
n° 0002/2008

Gestão de Ontologias

Jairo Souza
Sean Siqueira
Leonardo Azevedo
Fernanda Baião
Mauro Lopes
Flavia Santoro
Claudia Cappelli
Vanessa Nunes
Andréa Magdaleno

Departamento de Informática Aplicada

UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
Av. Pasteur, 458, Urca - CEP 22290-240
RIO DE JANEIRO – BRASIL

Projeto de Pesquisa

Grupo de Pesquisa Participante



Patrocínio



PETROBRAS

Gestão de Ontologias *

Jairo Souza^{1,2}, Sean Siqueira^{2,3}, Leonardo Azevedo^{2,3}, Fernanda Baião^{2,3},
Mauro Lopes², Flavia Santoro^{2,3}, Claudia Cappelli², Vanessa Nunes², Andréa
Magdaleno²

¹Pontifícia Universidade Católica do Rio de Janeiro (PUC-RIO)

²Núcleo de Pesquisa e Prática em Tecnologia (NP2Tec)

³ Departamento de Informática Aplicada (DIA) – Universidade Federal do Estado do Rio de Janeiro (UNIRIO)

jairofsouza@gmail.com, sean@uniriotec.br, azevedo@uniriotec.br,
fernanda.baiao@uniriotec.br, mauro.lopes@uniriotec.br, flavia.santoro@uniriotec.br,
claudia.cappelli@uniriotec.br, vanessa.nunes@uniriotec.br, andrea.magalhaes@uniriotec.br

Abstract. Ontologies have gained popularity, but its promises of being a key point to the solution of real-world problems and mitigating interoperability problems at a large scale have not yet been accomplished. Ontology management is at the kernel of this evolution, and there is a lack of adequate strategies and mechanisms for handling it in such a way to contribute to a better alignment between business and IT. This work proposes an approach for enterprise ontology management as part of an Information Architecture initiative. This approach provides a more complete foundation of the ontology lifecycle while guiding the enterprise in this management, by defining a set of processes, roles and competencies required for ontology management.

Keywords: Ontology management, knowledge management, information architecture.

Resumo. Ontologias vêm ganhando popularidade, mas suas promessas de solução para problemas reais e de interoperabilidade não foram ainda realizados. Gestão de ontologias é um ponto central desta evolução, e existem uma falta de estratégias e mecanismos adequados para tratá-la de forma a contribuir para o alinhamento mais adequado entre negócio e TI. Este trabalho propõe uma abordagem para gestão de ontologia em empresas como parte de uma iniciativa de Arquitetura de Informação. Esta abordagem provê uma fundamentação mais completa do ciclo de vida de ontologia enquanto guia a empresa na sua gestão, pela definição de um conjunto de processos, papéis e competências necessárias para gestão de ontologias.

Palavras-chave: Gestão de ontologias, gestão de conhecimento, arquitetura de informação.

* Trabalho patrocinado pela Petrobras.

Sumário

1	Introdução	7
1.1	Contexto	7
1.2	Motivação	7
1.3	Objetivos	11
1.4	Metodologia	12
1.5	Estrutura do Relatório	13
2	Ontologias	13
2.1	Classificação de Ontologias	14
2.2	Princípios Básicos a Serem Seguidos na Construção de Ontologias	15
2.2.1	Modelagem de Ontologias	16
2.2.2	Passos para Desenvolver uma Ontologia	17
3	Propostas Acadêmicas Relacionadas à Gestão de Ontologias	18
3.1	Metodologias para a Construção de Ontologias	18
3.1.1	Metodologia Uschold e King	19
3.1.2	Metodologia Grüninger & Fox	19
3.1.3	Metodologia 101	20
3.1.4	Método KACTUS	21
3.1.5	Método SENSUS	22
3.1.6	Metodologia On-To-Knowledge	22
3.1.7	METHONTOLOGY	23
3.1.8	Processo KUP	25
3.1.9	Metodologia DILIGENT	28
3.1.10	Metodologia CO4	28
3.1.11	Metodologia Cyc	29
3.1.12	Considerações finais	29
3.2	Linguagens de representação	30
3.3	Linguagens de consulta de ontologias	32
3.3.1	nRQL	33
3.3.2	OWL-QL	34
3.3.3	SPARQL	36
3.3.4	Considerações finais	37
3.4	Reuso de Ontologias	38
3.4.1	Combinação	38
3.4.2	Alinhamento	39
3.4.3	Integração	40
3.4.4	Mapeamento	42
3.4.5	Divergências encontradas durante o reuso de ontologias	44
3.4.5.1	Divergências no nível da linguagem	45
3.4.5.2	Divergências no nível da conceitualização	46
3.5	Avaliação da Qualidade de Ontologias	48

3.6	Evolução de Ontologias	50
3.6.1	Manutenção	52
3.6.2	Versionamento	52
3.7	Ferramentas de apoio	54
4	Conclusão	55
	Referências Bibliográficas	56

1 Introdução

Este documento tem como objetivo apresentar o estudo realizado sobre a área de ontologias e sua gestão, abordando os principais autores e os trabalhos mais recentes encontrados na literatura, a fim de promover o entendimento atual sobre esse domínio.

Cobrir todas as questões com as quais um gestor de ontologias deve lidar não é uma tarefa simples. Além disso, até mesmo o conceito de “gestão de ontologias” não está bem definido na literatura e não há uma metodologia de projeto de ontologias única e correta, e não é o objetivo deste documento definir uma. O objetivo deste documento é apresentar um conjunto das principais idéias que são consideradas úteis para todas as atividades relacionadas à gestão de ontologias.

1.1 Contexto

O presente projeto de pesquisa trata da modelagem de processos da Gestão de Ontologias junto à equipe da Base de Dados Integrada do E&P (BDIEP), subordinada à área de Gestão de Dados Integrada do E&P, na TI Corporativa do E&P, na Petrobras (TIC/TIC-E&P/GDIEP). A motivação principal para este projeto é permitir gerenciar conceitos dentro do domínio da Exploração e Produção de Óleo e Gás na Petrobras, utilizando para isso uma representação explícita, formal e compartilhada que possibilite a exploração de seus relacionamentos, agregando integridade semântica às atividades da gerência de dados do E&P.

Assim, este projeto tem como objetivo definir e modelar processos de gestão de Ontologias junto a GDIEP. Para isso, pretende-se definir, em conjunto com a gerência da GDIEP/BDIEP, um conjunto de objetivos da gestão de ontologias, incluindo os papéis e competências envolvidos, as atividades de manutenção de ontologias dentro do domínio da E&P, e interfaces destes processos com outros processos da Petrobras.

Por gestão de ontologias, entende-se um conjunto de processos que têm como objetivo planejar, organizar, liderar, coordenar e controlar todo o ciclo de vida de uma ontologia, garantindo sua aplicação e utilização, assim como definindo métodos para conduzir as pessoas responsáveis por sua manipulação através da realização de suas atividades.

As próximas seções abordam a motivação para a concepção desta modelagem de processos de gestão de ontologias, os objetivos a serem alcançados, o método utilizado e a estrutura do presente relatório.

1.2 Motivação

Segundo Mike Uschold e Michael Gruninger [1996], pessoas, organizações e sistemas de software devem se comunicar. Entretanto, devido a diferentes necessidades e contextos, pode haver uma variedade de pontos de vistas e suposições diferentes no que se refere essencialmente à mesma questão. Cada um utiliza um jargão diferente; cada um pode ter conceitos, estruturas e métodos diferentes, sobrepostos e/ou não-correspondentes. A conseqüente falta de um entendimento compartilhado leva a uma comunicação deficiente entre pessoas, entre organizações e mesmo dentro de uma mesma organização.

No contexto da construção de um sistema de Tecnologia de Informação, esta falta de entendimento compartilhado leva a dificuldades na identificação de requisitos e na definição de especificações de sistemas e bases de dados. Diversos métodos de modelagem, paradigmas, linguagens e ferramentas de software limitam a interoperabilidade e o potencial para reuso e compartilhamento da informação.

No cenário da GDIEP/BDIEP, estas questões ficam evidentes tanto na criação/manutenção da base integrada como no seu próprio uso. Além disto, o desenvolvimento de novos sistemas, bem como a adequação e atualização dos já existentes implicam muitas vezes em utilizar conceitos e serviços já definidos e implementados. Um entendimento compartilhado do domínio torna-se então fundamental para possibilitar o reuso de tabelas e serviços.

Ainda segundo Uschold e Gruninger [1996], o modo de endereçar estes problemas é reduzir ou eliminar a confusão conceitual e terminológica e chegar a um entendimento compartilhado. Tal entendimento pode funcionar como um *framework* unificador para os diferentes pontos de vista e servir como base para:

- A comunicação entre pessoas com diferentes necessidades e pontos de vista, vindo de seus diferentes contextos;
- A interoperabilidade entre sistemas obtida pela tradução entre diferentes métodos de modelagem, paradigmas, linguagens e ferramentas de software;
- Os benefícios de engenharia de sistemas, em particular para a:
 - Reusabilidade - o entendimento compartilhado é a base para uma codificação formal das entidades, atributos, processos e seus relacionamentos importantes no domínio de interesse. Esta representação formal pode ser (ou se tornar assim pela tradução automática) um componente reutilizável e/ou compartilhado em um sistema de software.
 - Confiabilidade - uma representação formal também torna a automação de verificação de consistência possível, resultando em software mais confiável.
 - Especificação - o entendimento compartilhado pode auxiliar no processo de identificação de requisitos e na definição de uma especificação para um sistema de TI. Isto é especialmente verdadeiro quando os requisitos envolvem diferentes grupos usando diferentes terminologias no mesmo domínio ou em múltiplos domínios.
 - Qualidade da informação - o conhecimento da semântica da informação permite que sejam identificadas informações utilizadas com significados diferentes em diferentes domínios, ou seja, permite identificar se a informação está ou não sendo utilizada com a semântica definida na ontologia.

Noy e McGuinness [2001] apresentam razões igualmente interessantes para o desenvolvimento de uma ontologia:

- Para compartilhar entendimento comum da estrutura da informação entre pessoas e agentes de software: Se diferentes páginas / aplicações compartilham e publicam a mesma ontologia base dos termos que utilizam, então agentes de computadores podem extrair e agregar informação destes diferentes locais e utilizar esta informação agregada para responder consultas de usuários ou como entrada de dados para outras aplicações.

- Para possibilitar reuso do conhecimento do domínio: Se um grupo de pesquisadores / desenvolvedores cria uma ontologia detalhada, outros podem simplesmente reutilizá-la para seus domínios. Adicionalmente, se é necessário criar uma ontologia muito grande, pode-se integrar várias outras existentes, que descrevem partes do domínio maior.
- Para tornar explícitas as suposições do domínio: Se o conhecimento sobre o domínio muda, então será mais fácil mudar estas suposições se estiverem explícitas. Suposições sobre o mundo, quando codificadas no corpo do programa, tornam-se mais difíceis de encontrar e entender, bem como de alterar. Além disto, especificações explícitas do conhecimento de domínio são úteis para novos usuários.
- Para separar o conhecimento de domínio do conhecimento operacional: Pode-se descrever a tarefa independentemente do domínio. Então, pode-se desenvolver uma ontologia do domínio e aplicar o algoritmo para desenvolver a tarefa. O mesmo algoritmo pode ser utilizado para outros domínios.
- Para analisar o conhecimento de domínio: Análise formal de termos é extremamente valiosa ao tentar reutilizar ontologias existentes e estendê-las.

Assim, ontologias podem ser utilizadas em várias áreas, tais como citado em [Damjanovic *et al.*, 2004] e detalhado em [Mcguinness, 2005]:

- Integração de dados: Ao descrever num nível conceitual as informações da organização e utilizar construtos lógicos, ontologias permitem que informações entre diferentes fontes de informação sejam mapeadas em conceitos comuns, facilitando a troca de informações.
- Modelagem de processos de negócio: O compartilhamento de conhecimento através da perspectiva de processos de negócio se tornou um assunto proeminente nas organizações no sentido de promover a aprendizagem organizacional. Dentro desta visão, a representação de conceitos do domínio do negócio e das relações semânticas existentes entre os mesmos torna-se essencial e extremamente importante, apesar de não trivial. Ontologias permitem uma representação de tais abstrações de forma fidedigna e não ambígua, e desta forma atuam na complementação ao modelo de processos de negócio facilitando o entendimento do domínio [Cappelli *et al.*, 2007].
- Projeto de banco de dados: Ontologias são modelos conceituais de dados e podem ser derivadas para um projeto de banco de dados ou, até mesmo, serem utilizadas para validar um projeto de banco de dados, garantindo que os conceitos estão sendo utilizados conforme os especialistas sobre o domínio os entendem.
- Recuperação e extração de informações: A capacidade de inferência de relacionamentos implícitos em ontologias enriquece o processo de mineração de dados, onde existem abordagens principalmente em mineração de textos. Em sistemas para recuperação de informação, ontologias dão suporte tanto para a fase de indexação ao criar anotações em textos (enriquecimento semântico com classes descritas na ontologia) quanto durante a fase de recuperação através de inferências para descobrir novas informações [Freitas *et al.*, 2005].
- Organização e gerência de conhecimento: O conhecimento gerado dentro de uma corporação, na forma de informação útil, *expertise*, melhores práticas e outros tipos de conhecimento, pode ser considerado a melhor ferramenta para re-

solução de problemas. Ontologias podem ser utilizadas para descrever definições e relações precisas e não ambíguas, e restrições que representarão o contexto da rotina diária do funcionário, podendo fazê-los registrar suas experiências e descobertas, colaborar e alcançar a inteligência coletiva. Além disso, os requisitos para sistemas de gerência de conhecimento casam com vários benefícios conseguidos ao utilizar ontologias, como dar suporte à integração dos sistemas e bases de dados já existentes e a habilidade de descobrir informações implícitas [Freitas *et al.*, 2005].

- Busca baseada em ontologias: Consultas podem ser expandidas a fim de permitir que, assim que usuários escolham um conceito para pesquisa, possam ser utilizadas também as propriedades do conceito para tornar as consultas mais eficientes. Assim, ao escolher pesquisar sobre carros, o usuário pode utilizar-se da propriedade “possui sistema de frenagem” para escolher somente os carros com o sistema de frenagem que lhe interessa.
- Modelagem qualitativa: Um modelo qualitativo pode ser pensado como um conjunto de modelos numéricos definidos por algumas restrições (em geral biológicas e físicas). Ontologias auxiliam tais modelos através da representação explícita dos conceitos, bem como a definição de restrições aplicáveis, tais como tempo e espaço.
- Engenharia de linguagem: A aquisição de informação a partir de textos livres e documentos parcialmente estruturados terá um papel fundamental no crescimento e sucesso da web semântica. Por outro lado, a linguística computacional aplicada que enfoca no resultado prático de modelar o uso da linguagem humana poderá se utilizar das tecnologias da web semântica via edição semântica e de ontologias de modo a promover a aquisição de conhecimento.
- Comércio Eletrônico (ex., Amazon.com, Yahoo Shopping, etc.): Compartilhar o entendimento comum de estruturas de informação entre agentes de *software* é uma das vantagens em se utilizar ontologias [Gruber, 1993]. Essa vantagem pode ser aplicável em cenários de comércio eletrônico, onde agentes de *software* trocam informações, negociam e tomam decisões baseadas num conhecimento descrito por ontologias. Além disso, ontologias estão sendo utilizadas para criar padrões de comunicação entre sistemas de comércio eletrônico, como em [Gordijn, 2002], [Blommestein, 2006]
- Suporte à configuração: Classes podem ser definidas para conter descrições de quais tipos de partes um sistema pode ter. Além disso, interações entre propriedades podem definir valores de uma propriedade que podem afetar o valor de outras propriedades. Por exemplo, pode-se gerar uma ontologia com a informação para configurar alguns tipos de *home theaters*. Assim, temos como televisão, caixas de som, amplificador são definidos e devidamente relacionados. Uma classe de HomeTheaterSemFio é definida e, os usuários podem escolher esta classe, um sistema configurador automaticamente preencherá os aparelhos compatíveis com esse tipo de *home theater*, bem como as peças vendidas separadas que são compatíveis, tipos de televisões, formas de ligação, etc.

1.3 Objetivos

Em função da importância do tema, especialmente no contexto de integração de dados, foi desenvolvida uma pesquisa, com o objetivo geral de propor uma forma de estruturação da gestão de ontologias, que esteja integrada aos processos e procedimentos de gestão de dados da BDIEP/GDIEP, bem como esteja alinhada às propostas acadêmicas e viabilidade de implantação no contexto da Petrobras/TIC-E&P.

Os objetivos específicos da pesquisa foram definir, em conjunto com a gerência da GDIEP/BDIEP, papéis e competências envolvidas, as atividades de manutenção de ontologias dentro do domínio da E&P, e interfaces destes processos com outros processos da Petrobras.

O presente relatório apresenta uma revisão conceitual de Gestão de Ontologias, bem como propostas relacionadas, para então permitir uma proposta de macroprocessos de gestão de ontologias e seus desdobramentos aplicados à GDIEP/BDIEP, apresentado no Relatório de Proposta de Gestão de Ontologias para a Petrobras, que apresenta também discussões fruto da aplicabilidade da proposta para, então, indicar conclusões e propostas de estudos futuros.

Os seguintes objetivos foram levantados como primeira proposta do projeto de gestão de ontologias como um todo:

- Prover o entendimento uniforme do domínio por todos os envolvidos
 - Garantir o entendimento dos conceitos
 - Aumentar a compreensão do domínio de forma explícita
 - Melhorar o entendimento do negócio
 - Maximizar a aprendizagem dos conceitos integrados
 - Garantir o uso de um vocabulário comum
 - Aumentar a separação do conhecimento do domínio do conhecimento operacional
 - Permitir a análise do conhecimento do domínio
 - Aumentar eficiência na integração de conceitos
 - Garantir a representação de conceitos de forma não ambígua.
 - Possibilitar a representação de conceitos em diferentes níveis de abstração.
 - Garantir a participação dos analistas de negócio na organização dos conceitos
 - Otimizar as atividades dos analistas de dados na gestão dos conceitos integrados
 - Viabilizar planejamento/desenvolvimento de serviços de acesso a dados de forma integrada
 - Otimizar os recursos de BDIEP
 - Garantir o domínio de tecnologias relacionadas a ontologias
 - Otimizar a realização das atividades especializadas de gestão de conceitos

- Atender aos padrões de qualidade estabelecidos pela organização para gestão de conceitos
- Possibilitar a reutilização do domínio da informação
- Assegurar o uso de conceitos definidos em fóruns da área
- Maximizar a divulgação de conceitos integrados em fóruns da área

1.4 Metodologia

De modo a atingir os objetivos traçados, partiu-se de uma estratégia de etapas ilustradas na figura abaixo.

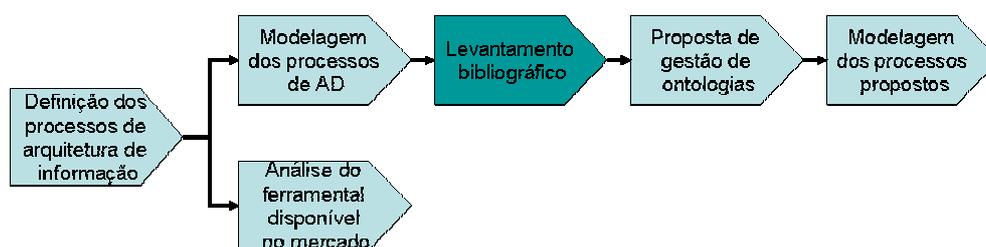


Figura 1 – Metodologia do projeto

Observando-se a necessidade de se considerar a arquitetura de TI da empresa de modo a permitir que as iniciativas da área estejam em consonância com o planejamento estratégico, partiu-se para uma definição dos processos de arquitetura de informação, uma vez que considerava-se a perspectiva de dados/informações. Assim, para propor uma gestão de ontologias alinhada às atividades realizadas na GDIEP, buscou-se também entender melhor a área e seus processos, em especial através das atividades transacionais.

Deste modo, a etapa de Modelagem dos Processos de Administração de Dados teve o intuito de promover o entendimento da equipe em relação aos principais processos adotados na Petrobras em relação à gestão de dados para possibilitar uma proposta de gestão de ontologias mais consistente com a realidade da empresa.

A etapa de Levantamento Bibliográfico visa promover o entendimento atual de gestão de ontologias, com enfoque na literatura disponível no meio acadêmico. Optou-se por não utilizar um levantamento no campo empresarial considerando o baixo grau de maturidade desta abordagem nas organizações em geral e o tempo necessário para desenvolver tal estratégia. Assim, esta primeira etapa consistiu em realizar uma busca bibliográfica avançada que percorreu os principais autores e trabalhos relacionados à gestão de ontologias, especialmente os mais recentes, que envolvem a Web. A busca bibliográfica fundamentou a criação de um quadro conceitual sobre a gestão de ontologias, sendo o ponto de partida para a proposta de processos para a gestão de ontologias da BDIEP/GDIEP.

Considerando a ênfase atual em Arquitetura de Tecnologia da Informação, buscou-se um alinhamento da Proposta de Gestão de Ontologias para a BDIEP/GDIEP também com a abordagem de gestão da informação seguindo os preceitos de Arquitetura de Informação. Assim, os processos e atividades de gestão de ontologias estão sendo revistos e re-estruturados conforme esta abordagem de Arquitetura de Informação. Estes novos processos serão apresentados e discutidos com especialistas da aca-

demia na área de Gestão de Ontologias, sob o enfoque de Representação de Conhecimento e Raciocínio, Gerência de Conhecimento, Modelagem de Processos, Administração de Dados e Banco de Dados, através da participação de diversos pesquisadores da UNIRIO.

A etapa de Modelagem dos Processos Propostos compreenderá a modelagem de processos da Gestão de Ontologias da BDIEP/GDIEP, de modo a possibilitar sua implantação.

Em paralelo às etapas acima, a etapa de Análise do Ferramental Disponível no Mercado teve o objetivo de analisar os produtos disponíveis no mercado para apoiar as atividades modeladas nos processos de Gestão de Ontologias.

1.5 Estrutura do Relatório

O relatório está estruturado em elementos pré-textuais, 4 (quatro) capítulos, e elementos pós-textuais. Na primeira parte há itens como agradecimento, sumário e índices. Em seguida, estão os capítulos que estão brevemente descritos a seguir, e, por fim, os elementos pós-textuais.

O capítulo 1 explica o projeto de gestão de ontologias, de forma breve. Apresenta a motivação para a pesquisa, os objetivos e a metodologia que descreve o projeto.

O capítulo 2 apresenta a conceituação formada sobre a gestão de ontologias, com a apresentação de definições, de classificação de ontologias, bem como princípios básicos a serem seguidos na construção de ontologias e um resumo de passos para desenvolver uma ontologia.

O capítulo 3 apresenta os resultados da primeira etapa do projeto, abordando a busca bibliográfica realizada e o quadro conceitual formado sobre o tema gestão de ontologias.

O final do relatório ainda apresenta as referências bibliográficas utilizadas no desenvolvimento do documento.

2 Ontologias

Segundo Corcho *et al.* [2003a], a palavra ontologia vem da Filosofia, onde significa uma explicação sistemática do ser (ou da Existência). Várias definições sobre o significado de uma ontologia e sua evolução podem ser encontradas. Segundo Neches *et al.* [1991], “uma ontologia define os termos e relações básicas que compreendem o vocabulário de uma área tópico assim como as regras para combinar termos e relações para definir extensões ao vocabulário”. Esta definição descritiva diz o que fazer para construir uma ontologia e inclui não apenas os termos que são explicitamente definidos, mas também o conhecimento que pode ser inferido dela. Esta definição foi seguida pela área de Inteligência Artificial.

Poucos anos depois, Gruber [1993] definiu uma ontologia como “uma especificação explícita de uma conceitualização”, sendo esta definição a mais referenciada na literatura. Baseada nesta definição de Gruber, várias outras definições foram propostas.

Borst [1997] modificou esta definição dizendo que “Ontologias são definidas como uma especificação formal de uma conceitualização compartilhada”. As defini-

ções de Gruber e Borst foram agregadas e explicadas por Studer *et al.* [1998]: “Conceitualização se refere a um modelo abstrato de algum fenômeno do mundo pela identificação de conceitos relevantes daquele fenômeno. Explícito significa que o tipo de conceitos usados e as restrições sobre seu uso são explicitamente definidos. Formal se refere ao fato de uma ontologia ser processável por máquina. Compartilhado reflete a noção que uma ontologia captura conhecimento consensual, isto é, não é particular a algum indivíduo, mas aceito por um grupo”.

Guarino *et al.* [1995] consideram uma ontologia como “uma teoria lógica que dá um relato explícito, parcial de uma conceitualização”, onde conceitualização é basicamente uma idéia do mundo que uma pessoa ou grupo de pessoas pode ter.

Segundo Corcho *et al.* [2003b], também existe outro grupo de definições baseadas no processo seguido para construir a ontologia. Uma vez que ontologias são amplamente utilizadas para diferentes propósitos (processamento de linguagem natural, gerência de conhecimento, comércio eletrônico, integração inteligente de informação, web semântica etc.) em diferentes comunidades (engenharia de conhecimento, inteligência artificial, banco de dados e engenharia de software), Uschold e Jasper [1999] proveram uma nova definição da palavra ontologia: “uma ontologia pode ter uma variedade de formas, mas necessariamente inclui um vocabulário de termos e alguma especificação de seu significado. Isto inclui definições e uma indicação de como conceitos estão interrelacionados, que coletivamente impõem uma estrutura no domínio e restringem as interpretações possíveis dos termos”. Embora na comunidade de banco de dados, assim como na comunidade de orientação a objetos, também se constroem modelos de domínio usando conceitos, relações, propriedades etc., mas na maioria das vezes ambas as comunidades impõem menos restrições semânticas do que aquelas consideradas em ontologias (mais complexas).

Diferentes definições provêm pontos de vista diferentes e complementares da mesma realidade. Alguns autores provêm definições que são independentes dos processos seguidos para construir a ontologia e também independentes de seu uso em aplicações, enquanto outras definições são influenciadas pelo processo de desenvolvimento de ontologias. Segundo Corcho *et al.* [2003b], ontologias capturam conhecimento consensual de modo genérico e formal e que pode ser reutilizado e compartilhado entre aplicações (software) e por grupos de pessoas. Ontologias são usualmente construídas cooperativamente por um grupo de pessoas em diferentes locais.

Recentemente ontologias são desenvolvidas nas estações de trabalho e são cada vez mais utilizadas. Muitos departamentos/áreas desenvolvem ontologias padronizadas que especialistas de domínio podem usar para compartilhar e anotar informação em suas áreas. Ontologias de propósito geral, ou seja, que podem ser utilizadas em diversos domínios por serem mais genéricas, também estão aparecendo.

2.1 Classificação de Ontologias

Os tipos mais comuns de ontologias são:

- Meta-ontologias, também chamadas de Ontologias Genéricas ou Ontologias Fundamentais, são reutilizáveis (ou aplicáveis) em diferentes domínios. Os exemplos mais representativos poderiam ser ontologias que estabelecem relações lógicas entre parte e todo, vocabulário relacionado a eventos, tempo, espaço, comportamento, etc.
- Ontologias de domínio são reutilizáveis em um dado domínio. Elas provêm

vocabulários sobre os conceitos dentro de um domínio e seus relacionamentos, sobre as atividades que envolvem este domínio e sobre as teorias e princípios elementares que governam aquele domínio.

- Ontologias de tarefas provêm um vocabulário sistematizado dos termos usados para resolver problemas associados com tarefas que podem ou não ser de um mesmo domínio. Estas ontologias provêm um conjunto de termos para genericamente descrever como resolver um tipo de problema. Incluem nomes genéricos (por exemplo, plano, objetivo, restrição), verbos genéricos (por exemplo, atribuir, classificar, selecionar), adjetivos genéricos (por exemplo, atribuído) e outros nas tarefas agendadas.
- Ontologias de aplicações contêm o conhecimento necessário para modelar situações específicas de uma tarefa em um domínio particular.

Todos estes tipos de ontologias podem ser combinados para construir uma nova ontologia.

2.2 Princípios Básicos a Serem Seguidos na Construção de Ontologias

Segundo Natalya Noy e Deborah McGuinness [2001], como todo exercício de modelagem, também não existe uma única forma ou metodologia para modelar ontologias. Existem, porém, algumas regras fundamentais a serem seguidas durante um projeto de ontologias que podem, em muitos casos, ajudar durante as decisões de projeto:

- Não existe uma única forma de modelar um domínio – o que existem são alternativas viáveis ou não. A melhor solução quase sempre depende da aplicação que você tem em mente e das extensões do domínio que você pode antecipar.
- O desenvolvimento de ontologias é necessariamente um processo iterativo.
- Os conceitos na ontologia deveriam estar próximos de objetos (físicos ou lógicos) e seus relacionamentos no seu domínio de interesse. Assim, existe uma grande probabilidade de serem substantivos (objetos) ou verbos (relacionamentos) nas sentenças que descrevem seu domínio.

Decidir para que a ontologia será usada e o quão detalhada ou geral a ontologia necessita ser é fundamental para guiar muitas das decisões de modelagem [Noy e McGuinness, 2001]. Entre as muitas alternativas viáveis para a modelagem, é necessário determinar qual alternativa seria melhor empregada para a tarefa projetada, qual seria mais intuitiva, mais extensível e manutenível.

Gómez-Pérez [1999b] apresenta alguns princípios a serem seguidos na construção de ontologias. Esses princípios são reafirmados por Gruber [2007]:

- Clareza e objetividade, que significa que a ontologia deve prover o significado dos termos definidos através de definições objetivas e também de documentação em linguagem natural.
- Completude, que significa que uma definição expressa por condições necessárias e suficientes é preferida em relação a uma definição parcial.
- Coerência, para permitir inferências que são consistentes com as definições.

- Maximização de extensibilidade monotônica¹, que significa que os novos termos gerais ou especializados devem ser incluídos na ontologia de tal modo que não requeira a revisão de definições existentes.
- Compromissos ontológicos mínimos, que significa fazer o mínimo possível de afirmações sobre o mundo sendo modelado, de modo que a ontologia deve especificar o mínimo possível sobre o significado de seus termos, dando aos envolvidos na manipulação da ontologia liberdade para especializar e instanciar a ontologia à medida que seja necessário.
- Princípio de distinção ontológica, que indica que as classes em uma ontologia devem ser disjuntas. O critério usado para isolar o núcleo de propriedades consideradas invariáveis para uma instância de uma classe é chamado de Critério de Identidade.
- Diversificação de hierarquias para aumentar o poder provido por múltiplos mecanismos de herança. Se conhecimento suficiente é representado na ontologia e se tantos critérios quanto possível forem usados, será mais fácil acrescentar novos conceitos (uma vez que eles podem ser facilmente especificados de conceitos e critérios de classificação pré-existentes) e herdar propriedades de diferentes pontos de vista.
- Modularidade para possibilitar/promover o reuso da ontologia.
- Minimizar a distância semântica entre conceitos irmãos. Conceitos similares são agrupados e representados como subclasses de uma classe e devem ser definidos usando as mesmas primitivas, uma vez que conceitos que são menos similares são representados mais distantes na hierarquia.
- Nomes padronizados, sempre que possível.

É necessário lembrar que uma ontologia é um modelo da realidade do mundo e os conceitos na ontologia devem refletir essa realidade. Depois de definida a versão inicial da ontologia, é necessário avaliar e conferir esse modelo utilizando-o em aplicações ou métodos para resolução de problemas, discutindo-o com os especialistas do domínio, ou ambos [Noy e McGuinness, 2001]. A versão inicial de uma ontologia deverá ser alterada após uma revisão. Este processo de projeto iterativo provavelmente continuará durante todo o ciclo de vida da ontologia.

2.2.1 Modelagem de Ontologias

Conceitos em uma ontologia podem estar relacionados a objetos (físicos ou lógicos) e a relacionamentos no seu domínio de interesse. Conceitos são prováveis de serem identificados como substantivos (objetos) ou verbos (relacionamentos) em sentenças que descrevem o domínio.

Algumas idéias de projeto de ontologia se originam do projeto de orientação a objetos [Rumbaugh *et al.*, 1991], [Booch *et al.*, 1999]. Contudo, o desenvolvimento de ontologias é diferente da modelagem de classes e relacionamentos numa modelagem orientada a objetos (representada em UML). Conforme salientam Noy & McGuinness [2001], a modelagem orientada a objetos se foca principalmente sobre métodos em classes — um analista toma decisões baseado em propriedades operacionais de uma

¹ A lógica monotônica é aquela em que o conjunto de teoremas de uma teoria é sempre um subconjunto dos teoremas de qualquer extensão desta teoria [Russell e Norvig, 1995].

classe, enquanto um engenheiro de ontologias toma decisões baseado em propriedades estruturais de uma classe, explicitando suas relações e formalizando os conceitos. Como resultado, uma estrutura de classe e relações em uma ontologia pode ser diferente de uma estrutura em uma modelagem orientada a objetos para um domínio similar [Noy e McGuinness, 2001].

A distinção entre a modelagem de ontologias e a modelagem orientada a objetos em UML pode ser detalhada nos seguintes aspectos [IBM, 2008b]:

- Ontologias são fundamentalmente baseadas em lógica. Enquanto ontologias permitem raciocínio automatizado (inferências), um modelo orientado a objetos (OO) não permite.
- No tratamento das propriedades, ontologias permitem herança entre propriedades, enquanto a modelagem orientada a objetos não permite.
- Propriedades de conceitos em ontologias são sempre herdadas pelos conceitos-filhos e não existem propriedades ocultas, como privadas, protegidas, etc.
- Ontologias permitem adicionar propriedades a relacionamentos como simétricos, transitivos e inversos para que sejam usados em inferências. O mesmo não é verdadeiro na modelagem OO.
- Ontologias permitem múltiplas heranças entre classes e também entre propriedades. A modelagem OO permite múltipla herança entre classes somente, porém não é uma prática comum dado a limitação de algumas linguagens de programação.

Apesar dessas diferenças, a modelagem OO fornece um primeiro passo para a criação de ontologias. Estas ontologias normalmente precisam ser validadas (vide seção 3.5) e enriquecidas com mais dados semânticos sobre o domínio para serem melhor aproveitadas em máquinas de inferência.

2.2.2 Passos para Desenvolver uma Ontologia

Para os propósitos deste documento, uma ontologia é uma descrição formal explícita de conceitos em um domínio de discurso (classes - algumas vezes chamadas de conceitos), atributos do conceito (slots - algumas vezes chamados de papéis ou propriedades) e restrições sobre as propriedades (facetadas - algumas vezes chamadas de restrições).

Uma ontologia, juntamente com um conjunto de instâncias das classes, constitui uma base de conhecimento. Neste documento, considera-se o desenvolvimento de ontologia correspondente ao próprio esquema conceitual/contextual de dados. Sendo assim, classes descrevem conceitos de um domínio. Uma classe pode ter subclasses que representam conceitos que são mais específicos do que as superclasses. *Slots* descrevem propriedades de classes e instâncias. Em termos práticos, desenvolver uma ontologia inclui:

- Definir classes na ontologia;
- Organizar estas classes em uma hierarquia taxonômica (subclasses - superclasses);
- Definir atributos e suas restrições (descrever valores permitidos para estes atributos);
- Criar a base de conhecimento, definindo instâncias individuais destas classes

preenchendo uma informação do valor de um atributo específico e restrições adicionais do atributo.

Esses passos são detalhados em diferentes abordagens na seção 3.1, onde são apresentadas as principais metodologias para construção de ontologias existentes na literatura.

3 Propostas Acadêmicas Relacionadas à Gestão de Ontologias

Corcho *et al.* [2003b] explicam que ao construir uma nova ontologia, várias questões básicas aparecem relacionadas às metodologias, ferramentas e linguagens a serem utilizadas no processo de desenvolvimento. Estas questões podem ser adaptadas às necessidades do projeto:

- Quais métodos e metodologias podem ser utilizados para construir ontologias desde o início ou reutilizando ontologias já disponíveis? Que atividades são realizadas ao construir ontologias com uma determinada metodologia? Que metodologias apóiam a construção cooperativa de ontologias? Qual é o ciclo-de-vida de uma ontologia?
- Que ferramentas apóiam o processo de desenvolvimento de uma ontologia? Como as ontologias são armazenadas (em banco de dados ou arquivos) e depois recuperadas, visualizadas e manipuladas? As ferramentas têm algum mecanismo de inferência? As ferramentas têm tradutores para diferentes linguagens e formatos de ontologias? Qual é a qualidade das traduções? Como aplicações podem interoperar com servidores da ontologia e/ou usar as ontologias que foram desenvolvidas?
- Que linguagem deve ser utilizada para implementar a ontologia? Qual expressividade tem uma determinada linguagem de ontologia? Quais são os mecanismos de inferência de uma linguagem de ontologia? A ferramenta de desenvolvimento de ontologias considera a linguagem? A linguagem escolhida é apropriada para trocar informação entre aplicações diferentes? A linguagem facilita a integração da ontologia a uma aplicação? A linguagem é compatível com outra linguagem utilizada para representar conhecimento e informação na Web? A aplicação requer a implementação da ontologia em linguagens diferentes? A aplicação requer a integração de ontologias que já foram implementadas em linguagens diferentes? Existem tradutores que podem transformar a ontologia implementada em uma linguagem fonte para uma linguagem destino? E finalmente, como tais tradutores minimizam a perda de conhecimento no processo de tradução?

O presente relatório apresenta um levantamento bibliográfico visando possibilitar responder a estas e outras perguntas e permitir uma proposta de gestão de ontologias bem fundamentada e adequada à Petrobras.

3.1 Metodologias para a Construção de Ontologias

Para diminuir os riscos e aumentar o entendimento dos requisitos de um *software*, são necessários alguns passos e metodologias para sua construção [Corcho *et al.*, 2003a],

[Cristani e Cuel, 2005]. Uma vez que a construção de ontologias é um processo custoso e, em parte, se assemelha com o levantamento de requisitos de um domínio [Felicíssimo *et al.*, 2003], então também é importante minimizar os riscos de uma construção errônea. Como o tema engenharia de ontologias ainda é recente, cada grupo de pesquisadores utiliza seu próprio método, processo ou metodologia para a criação e atualizações de ontologias [Cardoso, 2007]. A meta aqui é apresentar um resumo das diferentes abordagens para o desenvolvimento de ontologias, destacando a existência de um processo de compatibilização (ou integração) de conceitos dentro dessas metodologias. As metodologias abaixo são propostas acadêmicas e relatos da experiência adquirida por alguns pesquisadores em projetos de pesquisa aplicada.

3.1.1 Metodologia Uschold e King

Esta metodologia é baseada no contexto do desenvolvimento da *Enterprise Ontology*, uma ontologia para processos de modelagem de empresas [Uschold e King, 1995]. Ela fornece quatro diretrizes principais para o desenvolvimento de ontologias [Fernández-López e Gómez-Pérez, 2002], que são descritas a seguir:

- *Identificação do propósito e escopo* – onde se detalha o porquê da construção da ontologia e onde ela será utilizada.
- *Construção da ontologia* - Esta fase é dividida em três passos, que são:
 - *Captura da ontologia*: onde ocorre a identificação dos conceitos e relacionamentos chave para o domínio de interesse, focando no sentido concreto destes para produzir definições textuais não ambíguas e a identificação dos termos para identificar tais conceitos e relacionamentos. Para a identificação dos conceitos chave são propostas três possíveis estratégias: partindo dos conceitos mais concretos para os mais abstratos (*bottom-up*), dos mais abstratos para os mais concretos (*top-down*), ou partindo dos conceitos mais relevantes para os mais abstratos e para os mais concretos (*middle-out*).
 - *Codificação*: nesta etapa há a representação do conhecimento adquirido no passo anterior, que é explicitado em linguagem formal.
 - *Integração de ontologias existentes*: durante os processos de captura e codificação surge a questão de se é realmente necessário usar ontologias já existentes e como isto é feito.
- *Avaliação* - aqui é necessária a referência à especificação e análise dos requisitos, às questões de competência e/ou ao mundo real.
- *Documentação* - recomenda-se que diretrizes sejam estabelecidas para documentar as ontologias, possivelmente diferindo de acordo com o tipo e o propósito da ontologia. Tudo deve ser documentado, desde os conceitos principais até as primitivas usadas para definir a ontologia.

3.1.2 Metodologia Gruninger & Fox

Esta metodologia é baseada na experiência de desenvolvimento de uma ontologia – TOVE (*TO*ronto *V*irtual *E*nterprise) [Kim *et al.*, 1999], [Gruninger *et al.*, 2000] – dentro do domínio de processos de negócios e modelagem de atividades.

Ela se baseia na construção de um modelo lógico do conhecimento a ser especificado através de uma ontologia. Este modelo não é construído diretamente, já que é feita uma descrição informal da especificação do que deve constar na ontologia para então ser formalizada. As seguintes etapas são propostas nesta metodologia:

- *Captura dos cenários de motivação* - identificação de problemas e/ou exemplos que não são adequadamente tratados por ontologias existentes e fornecimento de um conjunto de possíveis soluções para os problemas do cenário no qual se está trabalhando.
- *Formulação de questões de competência informais* - essas questões são baseadas nos cenários obtidos na etapa anterior e podem ser consideradas como requisitos de expressividade que estão em forma de questões. A ontologia tem que ser capaz de representar essas questões usando sua terminologia e caracterizar as respostas para estas questões usando axiomas e definições.
- *Especificação da terminologia da ontologia numa linguagem formal* - os termos definidos nesta etapa devem ser suficientes para responder as questões de competência formuladas na etapa anterior. Estes termos servirão como base para especificar a terminologia em uma linguagem formal, tal como lógica de primeira ordem ou na linguagem equivalente KIF (*Knowledge Interchange Format*) [Ginsberg, 1991], [Genesereth e Fikes, 1992]. Estes termos permitirão que definições e restrições usadas sejam expressas por meio de axiomas.
- *Formulação das questões de competência formais* - as questões de competência são reformuladas usando a linguagem lógica de primeira ordem. Uma vez que estas questões foram propostas e a terminologia foi definida, as questões de competência são definidas formalmente.
- *Especificação dos axiomas* - nesta fase a ontologia é formalmente codificada em uma linguagem lógica. Os axiomas especificam as definições dos termos da ontologia e das restrições em sua interpretação. Eles são definidos em sentenças de primeira ordem para definir os termos e restrições na ontologia, definindo a semântica, ou significado, destes termos.
- *Verificação dos teoremas de completude da ontologia* - uma vez que as questões de competência foram formalmente declaradas, é necessário definir as condições sob as quais as soluções para as questões são completadas.

Uma vez que a ontologia é descrita em lógica de primeira ordem, a metodologia se torna muito formal por se beneficiar da robustez da lógica clássica. Uma aplicação para essa metodologia é utilizá-la para transformar cenários informais em modelos computáveis.

3.1.3 Metodologia 101

Esta metodologia foi desenvolvida por autores envolvidos na criação de ambientes de edição de ontologias como Protege-2000, Ontolingua e Chimaera [Noy e McGuinness, 2001]. Eles propõem um guia simples baseado num processo iterativo que ajuda os desenvolvedores, mesmo que não sejam especialistas em Engenharia de Ontologias, a criarem uma ontologia usando estas ferramentas.

A seqüência de passos para desenvolver uma ontologia usando essa metodologia é resumida abaixo:

- *Escolha do domínio e escopo da ontologia* - Determine quais perguntas a ontologia deverá responder. Perguntas de competência são muito importantes nesse domínio, elas permitem que o desenvolvedor entenda quando a ontologia contém informação suficiente e quando ela possui o nível esperado de detalhes ou representação. Também nessa fase o desenvolvedor deve se preocupar com quem irá manter a ontologia, uma vez ela seja posta em uso.
- *Reutilização de ontologias existentes* - Procure por ontologias que definem o domínio. Existem bibliotecas de ontologias reusáveis na Web e na literatura (por exemplo: biblioteca de ontologias Ontolingua, DAML, UNSPSC, RosettaNet e DMOZ). Estes modelos de dados são fontes primárias importantes para a coleta dos termos relevantes do domínio.
- *Enumeração dos termos importantes na ontologia* - Escreva uma lista de todos os termos usados na ontologia e descreva esses termos, seus significados e suas propriedades.
- *Definição das classes e hierarquias de classes* - A definição da hierarquia das classes pode seguir a abordagem *top-down* (especialização dos conceitos), *bottom-up* (abstração dos conceitos) ou *middle-out* (uma abordagem híbrida, mais comum durante processos criativos).
- *Definição das propriedades das classes* - Adicione todas as propriedades e informações necessárias para que a ontologia responda as perguntas de competência.
- *Definição das restrições das propriedades* - Defina os valores permitidos para cada propriedade, sua cardinalidade, seu domínio e alcance.
- *Criação de instâncias* - Crie instâncias das classes na hierarquia.

3.1.4 Método KACTUS

O método KACTUS foi desenvolvido como parte do projeto *Esprit KACTUS*, para construções de ontologias no domínio de redes elétricas [Bernaras *et al.*, 1996]. Um dos objetivos do projeto KACTUS é investigar a viabilidade do reuso de conhecimento em sistemas técnicos complexos.

Este método para desenvolver ontologias está condicionado ao desenvolvimento de aplicações e tem como base uma aplicação da base de conhecimento, por meio de um processo de abstração (isto é, seguindo uma estratégia *bottom-up*). À medida que uma aplicação vai sendo construída, a ontologia que representa seu conhecimento também é construída. Desta forma, quanto mais aplicações são construídas, mais geral se torna a ontologia, que evolui e se adapta às novas aplicações e representará o conhecimento consensual necessário por todas as aplicações. Esta ontologia pode ser desenvolvida pela reutilização de outras e pode ser integrada a ontologias futuras. Logo, a cada aplicação desenvolvida, os seguintes passos são tomados:

- *Especificação da aplicação* - fornece o contexto da aplicação e uma visão dos componentes que a aplicação tenta modelar.
- *Projeto preliminar* - os conceitos relevantes de alto nível da ontologia são definidos a partir das listas de termos e de tarefas desenvolvidos na etapa anterior.
- *Refinamento e estruturação da ontologia* - é nesta etapa que se objetiva chegar ao projeto definitivo, de modo que ele seja o mais coerente possível.

3.1.5 Método SENSUS

Esta metodologia é baseada na ontologia SENSUS, que foi criada para fornecer uma estrutura conceitual ampla para desenvolver tradutores automatizados [Swartout *et al.*, 1996]. Segundo os autores, ela pode ser considerada uma ontologia de ampla cobertura, possuindo tanto conceitos de alto nível quanto específicos e tem mais de 70 mil conceitos organizados em hierarquias de acordo com seu nível de abstração. Ao invés de ter sido construída do zero, a SENSUS foi desenvolvida pela extração e união de informações de diferentes recursos eletrônicos, criando a ontologia através da combinação de diferentes ontologias.

A representação da ontologia da SENSUS pode ser descrita na forma de uma árvore, onde a raiz representa o termo mais abstrato e as folhas os mais específicos [Brauner *et al.*, 2003]. A partir daí, quando uma ontologia for construída para um determinado domínio, os seguintes passos devem ser seguidos:

- São extraídos os termos específicos do domínio e colocados como folhas da ontologia.
- São incluídos todos os conceitos que vão desde as folhas até a raiz da SENSUS na nova ontologia.
- São adicionados os termos relevantes para o domínio. Se um termo que pode ser relevante ao domínio não apareceu, então é adicionado manualmente e o passo anterior é realizado novamente, até que nenhum termo esteja ausente.
- Finalmente, para todos os nós que têm um grande número de caminhos entre eles, a sub-árvore sob eles é adicionada, baseada na idéia de que se muitos dos nós na sub-árvore foram considerados relevantes, então os outros nós na árvore também devem ser relevantes.

Como mostra os passos descritos acima, a metodologia SENSUS possui um enfoque *top-down* para derivação de ontologias específicas de domínio de ontologias maiores. Além disso, o enfoque de reutilização de conceitos de alto nível promove o compartilhamento do conhecimento, uma vez que a mesma ontologia base é utilizada para desenvolver ontologias em domínios particulares.

O *software* utilizado para construir ontologias seguindo este método é o Ontosaurus [Swartout *et al.*, 1996]. Ontosaurus é um servidor que pode ser acessado através da Web e representa o conhecimento na linguagem de programação LOOM [Ding, 2001], permitindo tradução para Ontolingua, KIF e em classes C++.

3.1.6 Metodologia On-To-Knowledge

A metodologia On-To-Knowledge (OTK) [York Sure e Studer, 2002], [Sure, 2003] tem como principal interesse a construção de grandes sistemas de gestão do conhecimento. O processo de desenvolvimento da ontologia nesta metodologia consiste das fases seguintes.

- *Meta-Processo de Conhecimento* - Esta é a fase de introdução de uma aplicação baseada em ontologia, onde os documentos e o conhecimento necessários são identificados. Este processo consiste das fases mostradas a seguir:
 - 1ª Fase - Estudo de viabilidade - nesta etapa ocorre a identificação de problemas e suas possíveis soluções, determinando as viabilidades eco-

nômicas, técnicas e de projeto, para saber se a ontologia deve ser construída ou não.

- 2ª Fase – Inicialização - nesta etapa são realizadas atividades tais como identificação dos requisitos, onde o objetivo, o domínio e o escopo da ontologia são definidos através de questões de competência. É aqui também que pode ser examinada a possibilidade de haver uma integração com ontologias existentes.
- 3ª Fase – Refinamento - nesta etapa ocorre a extração do conhecimento, utilizando o método de extração mais aplicável (*top-down*, *middle-out* e *bottom-up*) à fonte de conhecimento disponível. É realizada a construção de uma ontologia modelo contendo os conceitos relevantes e descrevendo os relacionamentos entre estes os conceitos. Esta ontologia será utilizada como base para se chegar à ontologia desejada, a qual será descrita numa linguagem formal.
- 4ª Fase – Avaliação - nesta fase a ontologia é checada de acordo com o documento de requisitos, sendo que esta fase funciona de maneira iterativa com a anterior, podendo, inclusive, detectar falhas na representação da ontologia. Assim, vários ciclos podem ser executados até a ontologia ser considerada boa o bastante.
- 5ª Fase – Aplicação e Evolução - durante esta etapa ocorre a aplicação da ontologia em sistemas baseados em ontologia. É aqui onde o engenheiro recebe a recomendação de práticas para lidar com a evolução/atualização da ontologia, a manutenção, que faz com que mudanças no mundo real reflitam em mudanças na ontologia.
- *Processo de Conhecimento* - É um processo iterativo da gestão de conhecimento que é aplicado na organização. Neste processo são realizadas as seguintes atividades:
 - Atividade 1 – Criação ou importação de dados: o conhecimento deve ser criado ou adaptado para se moldar aos padrões da organização.
 - Atividade 2 – Captura do conhecimento: inclusão dos dados que se referem aos conceitos da ontologia.
 - Atividade 3 – Recuperação e acesso ao conhecimento: visões adicionais do conhecimento podem ser derivadas, permitindo a inferência de relacionamentos e descrições através de um mecanismo de inferência apropriado.
 - Atividade 4 – Utilização do conhecimento: a utilização e a reutilização do conhecimento são facilitadas pela ontologia, sendo que este conhecimento pode ser derivado ou adicionado de alguma outra fonte para que as metas do usuário sejam alcançadas.

3.1.7 METHONTOLOGY

A METHONTOLOGY possibilita a construção de ontologias e inclui a identificação do seu processo de desenvolvimento, um ciclo de vida baseado na evolução de protótipos e técnicas particulares para cumprir cada atividade [Blázquez *et al.*, 1998].

Esta metodologia engloba o processo de criação da ontologia desde seu início, passando pelos estágios de especificação, conceitualização, formalização, integração, implementação e manutenção, como mostra a Figura 2. O planejamento deve ser feito antes do início destas etapas e a aquisição do conhecimento, a documentação e a avaliação ao longo de todo o ciclo. As fases são distinguidas como segue:

- *Especificação* - deve conter o propósito da ontologia, os possíveis usuários finais, seu nível de formalidade, seu escopo, incluindo o conjunto de termos a ser representado, e a escolha da linguagem.
- *Aquisição do Conhecimento* - há várias fontes de conhecimento que podem ser usadas, tais como análise de texto, troca de informações entre grupos, entrevistas com peritos, através de outras ontologias etc.
- *Conceitualização* - um completo glossário de termos deve ser construído e em seguida os conceitos e verbos devem ser separados e distribuídos em árvores de classificação de conceitos e diagramas de verbos, havendo também a construção de tabelas de condições e regras.
- *Integração* - para reusar definições em outras ontologias é necessário inspecionar as meta-ontologias (conceitos usados para modelar a ontologia) para selecionar a que melhor se ajusta à conceitualização e tentar encontrar que outras ontologias contêm definições de termos coerentes com os da conceitualização e quais usam as definições mais apropriadas.
- *Implementação* - a ontologia é codificada em uma linguagem formal.
- *Avaliação* - verificação da corretude da ontologia e sua validação no sistema.
- *Documentação* - ao fim de cada fase um documento deve ser criado, contendo tudo o que foi feito nesta fase.

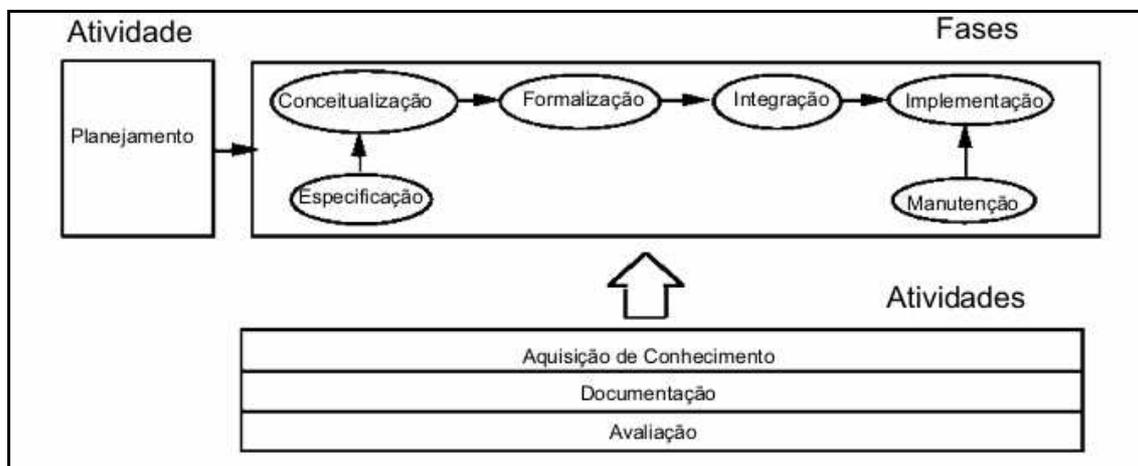


Figura 2 – Fases e atividades do método METHONTOLOGY, adaptado de [Férrandez et al., 1997]

Para o processo de desenvolvimento da ontologia são descritas algumas atividades, que são divididas em:

- *Atividades de Gerência de Projeto* - inclui planejamento, controle e garantia de qualidade.
- *Atividades Orientadas ao Desenvolvimento* - inclui especificação, conceitualização,

formalização e implementação.

- *Atividades de Suporte* - incluem uma série de atividades que são executadas paralelamente às atividades de desenvolvimento da ontologia, sem as quais a ontologia não poderia ser construída. Elas incluem aquisição de conhecimento, avaliação, integração, documentação e gerenciamento de configuração, que registra todas as versões da documentação.

O ambiente para construção de ontologias que utiliza esta metodologia é o ODE (*Ontology Design Environment*) [Blázquez *et al.*, 1998]. Seu objetivo é dar suporte aos engenheiros de ontologia durante o ciclo de vida do processo de desenvolvimento, automatizando cada atividade e integrando os resultados de cada fase com as entradas da fase seguinte.

A METHONTOLOGY é uma das mais abrangentes metodologias estudadas, já que engloba a maioria (não todos) dos aspectos do processo de desenvolvimento de *software* padronizado pelo IEEE. Ela é baseada na idéia de prototipação e evolução do ciclo de vida da ontologia e é altamente recomendada para o reuso de ontologias existentes. Esta metodologia serviu de base para a proposta de gestão de ontologias para a Petrobras.

3.1.8 Processo KUP

O KUP (*Knowledge Unified Process*) é um processo unificado proposto em [Orlean, 2003] para desenvolvimento de ontologias e bases de conhecimento, que deverão ser utilizadas em aplicações para a Web Semântica. Para garantir a qualidade de desenvolvimento de ontologias, ele foi baseado nos critérios de avaliação de metodologias para desenvolvimento de ontologias propostos por (Fernández-López e Gómez-Pérez, 2002) e faz a adaptação e integração das melhores práticas de desenvolvimento de *software* apresentadas pelo RUP (*Rational Unified Process*) [Rational, 1998]. Baseia-se também nas atividades do padrão definido pelo IEEE para a construção de processos de ciclo de vida e desenvolvimento de *software* [IEEE, 1995] tendo em vista atender aos critérios para avaliação de metodologias e processos de desenvolvimento de ontologias e bases de conhecimento.

Para se resolver o problema do risco que se mantém alto durante todo o processo de desenvolvimento de *software* foram identificadas seis práticas principais, citadas em [Orlean, 2003]: desenvolvimento iterativo, gerência de requisitos, uso de componentes arquiteturais, modelagem visual, verificação contínua de qualidade e gerência de mudanças. Apesar de estas melhores práticas terem sido propostas para solucionar problemas que ocorrem durante desenvolvimentos de *software*, elas foram aproveitadas no KUP para unificação das atividades propostas no processo.

Diferente da maneira como são executadas as tarefas durante o processo de desenvolvimento de *software* em cascata, o KUP é composto de três fases que são realizadas de forma iterativa ao longo de todo o processo de desenvolvimento e não apenas nas fases finais do projeto. As fases são as seguintes:

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

- *Fase de Evolução* - foco na integração de novos requisitos ao projeto, incluindo novos conceitos, relações e axiomas na ontologia.

Durante as fases do processo artefatos são gerados a partir de atividades executadas nas disciplinas. Uma disciplina agrupa atividades para integrar um processo específico a um determinado tema e um artefato é tudo o que é produzido, consumido ou modificado por uma atividade, podendo ser um documento, um modelo, um código fonte ou mesmo programas inteiros. O diagrama da Figura 3 descreve as iterações entre as atividades de cada disciplina nas fases propostas pelo KUP.

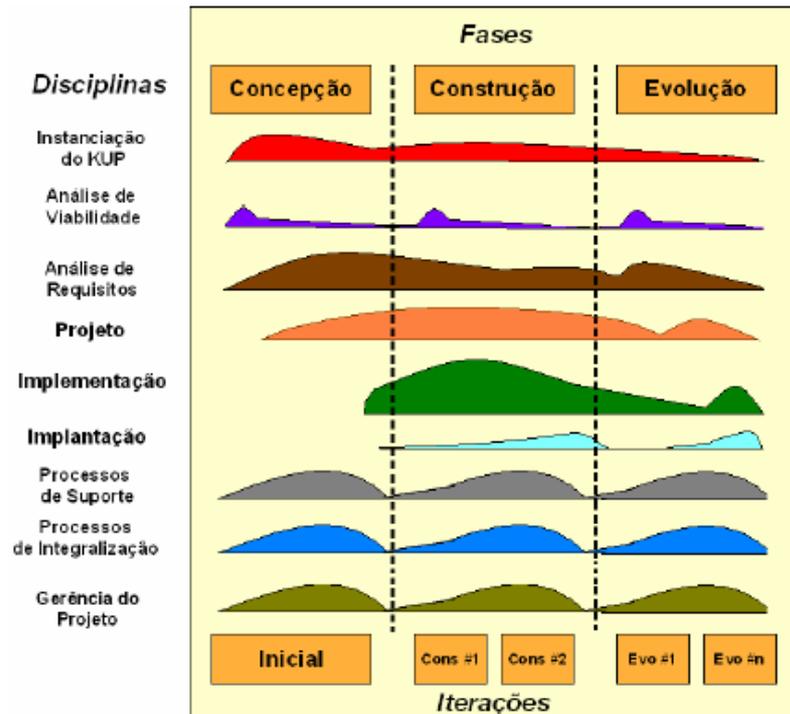


Figura 3 – Diagrama de iterações entre disciplinas e fases do KUP [Orlean, 2003]

Para deixar clara a possibilidade de reuso de ontologias nesta metodologia, uma breve descrição das disciplinas e artefatos do KUP se faz necessária. Mais detalhes podem ser obtidos em [Orlean, 2003].

- *Instanciação do KUP* - aqui é proposta a Instanciação do Framework de Processos, que inclui a definição das fases, disciplinas, papéis, atividades e artefatos que farão parte do processo concreto que será usado no projeto. Devem ser escolhidos os elementos que garantam a expressividade necessária para a execução do processo e a facilidade de execução; os critérios de monitoramento e controle e de gerência de qualidade do processo devem ser definidos.
- *Análise de viabilidade* - nesta etapa se faz uma identificação prévia de possíveis oportunidades que podem acelerar o processo de desenvolvimento ou de possíveis problemas que podem aumentar o seu risco. Ela deve focar em aspectos técnicos, econômicos e de projeto, de forma a minimizar os riscos nas fases seguintes.
- *Análise de requisitos* - compreende atividades iterativas que permitem desenvolver a especificação de requisitos da ontologia ou das aplicações que farão uso dela. Esta análise se subdivide em três atividades:
 - Identificação de propósito e escopo: onde será definido o propósito da

construção da ontologia e seu uso pretendido, se na comunicação, interoperabilidade ou Engenharia de Sistemas.

- Captura dos cenários motivacionais: identificação de possíveis situações e problemas que não são cobertos pelas ontologias existentes no domínio em questão. A partir daí serão inferidas possíveis soluções, que fornecerão uma semântica informal dos conceitos e relações que serão posteriormente incluídos na ontologia.
- Elicitação de requisitos e atores: são levantados os requisitos para o desenvolvimento do projeto, as fontes de informação e os atores dos cenários de uso da ontologia.
- Projeto: compreende atividades que serão executadas com o objetivo de apresentar uma representação coerente e consistente da ontologia que atenda aos requisitos especificados. É nesta etapa que ocorrerá a geração da ontologia em uma linguagem formal ou um modelo conceitual que permita sua implementação, através de seus conceitos, relações e axiomas.
- Conceitualização: é proposta a extração dos termos a partir do artefato que descreve as questões de competência informais. Com base no que é extraído são formuladas as questões de competência formais, a partir das quais se gera um Modelo Conceitual para a ontologia, construindo-se sua taxonomia, identificando suas relações de especialização ou generalização entre os conceitos e é feita a adição das relações não taxonômicas para os conceitos.
- Formalização: é feita a integração das ontologias identificadas na etapa anterior, a linguagem de representação é escolhida, ocorre a especificação formal da terminologia, axiomas e questões de competência.
- Avaliação da ontologia projetada e Refinamento: atividade executada depois da Formalização para que seja possível identificar omissões e erros no artefato gerado. A ontologia deve ser refinada até que todas as omissões ou erros sejam corrigidos e as questões de competência respondidas.
- Implementação: apresenta atividades que irão transformar a representação conceitual da ontologia em sua implementação. Para tal objetivo é preciso escolher a linguagem de representação de conhecimento e o ambiente de engenharia de ontologias mais adequados ao projeto. Após tais escolhas a ontologia é codificada na linguagem escolhida.
- Implantação: compreende os esforços para integrar a ontologia desenvolvida às diversas aplicações identificadas anteriormente.
- Processos de suporte: são propostas atividades de operação, suporte, manutenção e arquivamento da ontologia e suas aplicações.
- Processos de integralização: são propostas atividades de aquisição de conhecimento, verificação e validação, gestão de configuração da ontologia, documentação e treinamento.
- Gerência do projeto: realizada paralelamente às outras disciplinas do KUP, para assegurar o gerenciamento no decorrer do ciclo de vida da ontologia, propondo atividades de gerenciamento tais como: definição da equipe, definição de prazos e cronograma.

3.1.9 Metodologia DILIGENT

A metodologia DILIGENT [Tempich *et al.*, 2004] tem por objetivo criar um conjunto de ontologias para que o usuário possa compartilhá-las e, ao mesmo tempo, possa expandir seu uso local, seja por seu desejo ou necessidade individual. O objetivo dessa metodologia é superar erros de entendimento das ontologias que são criadas por um pequeno grupo de pessoas (os engenheiros de ontologias e os especialistas do domínio que representam os usuários), mas são utilizadas por um grande número de usuários. Segundo os autores, a engenharia de ontologia precisa lidar com um cenário onde a evolução das ontologias se dá de forma distribuída e sem grande controle dessa evolução.

Com essa metodologia, os autores fornecem um modelo de processo para engenharia distribuída das estruturas de conhecimento e planejam torná-la mais completa e fortemente testada.

Ao analisar esta metodologia, pode-se dizer que ela é baseada em cinco fases de alto nível que são descritas abaixo:

- *Construção* - A definição de uma ontologia inicial. A equipe envolvida na construção da ontologia inicial deve ser relativamente pequena para que facilite o consenso de uma primeira versão dessa ontologia compartilhada. Não é requerido que a ontologia criada possua completude.
- *Adaptação local* - Usuários trabalham em uma ontologia núcleo e a adaptam para seus negócios e necessidades locais. As adaptações locais são registradas e coletadas por um controle central.
- *Análise* - O controle analisa as ontologias locais e as modificações coletadas e tenta identificar similaridades nas ontologias dos usuários. Uma das principais atividades do controle é decidir quais mudanças deveriam ser introduzidas na ontologia compartilhada. No final dessa fase, a ontologia compartilhada se encontra numa versão mais refinada.
- *Revisão* - O controle regularmente revisa as ontologias locais e a compartilhada, e avalia as ontologias do ponto de vista técnico e de domínio. Outra tarefa do controle nessa fase é assegurar algumas compatibilidades entre as versões da ontologia compartilhada.
- *Atualização local* - Com a nova versão da ontologia compartilhada, os usuários atualizam suas ontologias locais para melhor se ajustarem à nova versão da ontologia compartilhada e suas necessidades locais.

A metodologia considera importante que os usuários envolvidos na construção colaborativa da mesma ontologia sejam especialistas com competências diferentes e complementares.

3.1.10 Metodologia CO4

CO4 [Euzenat, 1996] é um protocolo para chegar a um consenso entre várias bases de conhecimento (BCs), que são organizadas em uma árvore. As folhas são chamadas de BCs do usuário e os nós intermediários de BCs de grupo. As BCs de usuários não precisam ter conhecimento consensual. Em cada nó intermediário, há conhecimento de

consenso entre todos seus filhos e irmãos. Conhecimento consensual é obtido através da troca de mensagens entre usuários.

3.1.11 Metodologia Cyc

O método utilizado para construir a base de conhecimento Cyc [Lenat e Guha, 1990] consistiu de três fases: a primeira fase consistiu na codificação manual de artigos e partes de conhecimento, na qual o conhecimento de senso comum que está implícito em fontes diferentes foi extraído. A segunda e a terceira fases consistiram na aquisição de novo conhecimento de senso comum utilizando linguagem natural ou ferramentas de aprendizagem de máquina. Na segunda fase esta aquisição foi auxiliada por ferramentas, mas foi principalmente realizada por humanos, enquanto na terceira fase a aquisição foi realizada principalmente pelas ferramentas.

3.1.12 Considerações finais

A tabela abaixo resume, em termos gerais, as metodologias apresentadas anteriormente. A tabela apresenta quatro características avaliadas em cada metodologia: a forma de criação da ontologia (se esta pode ser criada manualmente de um rascunho, ou seja, sem ser obrigatório utilizar outras ontologias como base, ou a partir da integração com outras ontologias, ou ambas), a dependência de aplicação, a existência de uma fase de validação da ontologia criada e a identificação de uma fase que considere a evolução da ontologia.

Metodologia	Criação	Dependência de aplicação	Validação	Evolução
Ushold & King	Rascunho ou integração	Independente	Sim	Não
Grüninger & Fox	Rascunho	Semi-dependente	Sim	Não
101	Rascunho ou integração	Independente	Não	Não
KACTUS	Rascunho	Dependente	Não	Não
SENSUS	Integração	Semi-dependente	Não	Não
On-To-Knowledge	Rascunho ou integração	Dependente	Sim	Sim
METHONTOLOGY	Rascunho ou integração	Independente	Sim	Sim
KUP	Rascunho	Independente	Sim	Sim
DILIGENT	Rascunho e integração	Independente	Sim	Sim
CO4	Integração	Independente	Não	Não
CYC	Rascunho	Não se aplica	Não	Não

Tabela 1 – Comparação entre metodologias para construção de ontologias

Segundo o que foi exposto, Corcho *et al* [2003b] indicam que no método KACTUS, a ontologia é construída por meio de um processo de abstração de uma base de conhecimento inicial, enquanto no método Sensus, a ontologia é automaticamente ge-

rada de outras ontologias. Nos outros métodos e metodologias as ontologias podem ser construídas de um rascunho ou reusando outras ontologias.

Além disto, conforme explicado por Corcho *et al* [2003b], o método usado no projeto KACTUS e a metodologia On-To-Knowledge são dependentes de aplicação, uma vez que a ontologia é construída com base em uma dada aplicação; a metodologia de Grüninger e Fox e o método baseado no Sensus são semi-dependentes de aplicação (não são dependentes de uma aplicação específica, mas também não podem ser usadas em qualquer aplicação); e os métodos Cyc e de Uschold e King, bem como as metodologias METHONTOLOGY, KUP e DILIGENT são independentes da aplicação, uma vez que o processo de desenvolvimento de ontologia é totalmente independente dos usos da ontologia.

3.2 Linguagens de representação

Um modelo é uma abstração da realidade de acordo com uma certa conceitualização. Uma vez representada como um artefato concreto, um modelo pode apoiar a comunicação, aprendizagem e análise de aspectos relevantes do domínio.

Como um artefato concreto, um modelo representado deve ser expresso em alguma linguagem “adequada”. Tal linguagem deve estar de acordo com o domínio da realidade que ela representa.

Abstrações de uma dada porção da realidade são construídas em termos de conceitos, i.e., representações abstratas de certos aspectos de entidades que existem naquele domínio. Assim, conceitualização corresponde a um conjunto de conceitos utilizados para articular abstrações do estado de coisas em um dado domínio. A abstração de uma dada porção da realidade articulada de acordo com uma conceitualização do domínio é chamada de modelo.

Segundo Guizzardi [2005], conceitualizações e modelos são entidades abstratas que existem somente na mente do usuário ou de uma comunidade de usuários de uma linguagem. De modo a ser documentada, comunicada e analisada, estas entidades devem ser capturadas em termos de algum artefato concreto. A representação de um modelo conceitual é chamada de especificação. Além disto, é necessária uma linguagem de especificação (ou modelagem) para representar uma especificação.

Uma linguagem pode ser vista como algo que determina todas as especificações possíveis (i.e., todas especificações válidas gramaticalmente) que podem ser construídas usando aquela linguagem. Igualmente, uma conceitualização pode ser vista como algo que determina todos os modelos possíveis (significando os estados de coisas) admissíveis naquele domínio [Guarino, 1998].

Assim, uma linguagem de modelagem deve ser suficientemente expressiva para caracterizar corretamente a conceitualização de seu domínio de assunto e a semântica das especificações produzidas devem ser claras, i.e., deve ser fácil para um projetista da especificação reconhecer o que os construtores da linguagem significam em termos de conceitos de domínio. A especificação produzida usando a linguagem deve facilitar ao usuário o entendimento e raciocínio sobre os estados representados das coisas.

Atualmente, existem muitas linguagens que são usadas para o propósito de criar representações de conceitualizações do mundo real. Estas linguagens são em alguns casos chamadas de linguagens de modelagem de domínio (ex.: LINGO), linguagens de modelagem conceitual de dados (ex.: ER), linguagens de representação de ontologias (ex.: OWL), entre outros termos.

Embora algumas destas linguagens sejam empregadas na prática para modelagem conceitual, não foram projetadas com o propósito específico de estar de acordo com a realidade. Por exemplo, LINGO [Falbo *et al.*, 1998], [Falbo *et al.*, 2002] foi projetada com o objetivo específico de chegar a uma negociação positiva entre o poder de expressividade da linguagem e a habilidade de facilitar a quebra da barreira entre os níveis conceitual e de implementação. Esta preocupação também parece estar presente na proposta original de Peter Chen sobre diagramas ER [Chen, 1976]. OWL [Horrocks *et al.*, 2003] foi projetada com o propósito principal de atingir eficiência computacional em um processo de raciocínio automático. Outras linguagens como Z [Spivey, 1988] e técnicas de Ciência da Computação [Dijkman *et al.*, 2001], levam em consideração a simplicidade de um arcabouço matemático bem definido de teoria de conjuntos. Finalmente outras linguagens usadas atualmente para modelagem conceitual foram criadas para diferentes propósitos, a mais notória sendo UML [OMG, 2003], que inicialmente ficou muito enfocada em projeto de software.

O enfoque deste trabalho está em ontologias, de modo a capturar conhecimento sobre algum domínio de interesse. Uma ontologia descreve os conceitos do domínio e também os relacionamentos entre estes conceitos. Diferentes linguagens de ontologia provêm diferentes facilidades. O desenvolvimento mais recente em linguagem padrão de ontologia é OWL, da W3C (*World Wide Web Consortium*).

Segundo Corcho *et al.* [2003b], no início da década de 90, foi criado um conjunto de linguagens de implementação de ontologias baseadas em Inteligência Artificial. Basicamente o paradigma de representação de conhecimento destas linguagens de ontologia era baseado em lógica de primeira ordem (ex. KIF), em frames combinados com lógica de primeira ordem (ex. Ontolingua, OCML e FLogic) ou em Lógica Descritiva (ex. Loom).

A explosão da Internet levou à criação de linguagens de ontologia que exploram as características da Web. Estas linguagens são chamadas de linguagens de ontologias baseadas na web ou linguagens de marcação de ontologias.

SHOE [Heflin *et al.*, 1999] foi construída como uma extensão de HTML, permitindo a inserção de ontologias em documentos HTML. SHOE combina frames e regras e permite a representação apenas de conceitos, suas taxonomias, relações n-árias, instâncias e regras de dedução, que são utilizadas por sua ferramenta de inferência para obter novo conhecimento.

XOL [Karp *et al.*, 1999] foi desenvolvida a partir de XML representando um pequeno subconjunto de primitivas do protocolo OKBC². É uma linguagem muito restrita, onde apenas conceitos, taxonomias de conceitos e relações binárias podem ser especificados. Não há mecanismos de inferência agregados.

RDF [Lassila e Swick, 1999] foi desenvolvido pela W3C (o *World Wide Web Consortium*) como uma linguagem baseada em uma rede semântica para descrever recursos Web. RDF Schema [Brickley e Guha, 2004] foi construída pela W3C como uma extensão a RDF com primitivas baseadas em frame. A combinação de ambos, RDF e RDF Schema é normalmente conhecida como RDF(S) e não é muito expressiva. RDF(S) permite apenas a representação de conceitos, taxonomias de conceitos e relações binárias. Algumas ferramentas de inferência foram criadas, principalmente para checagem de restrições.

² *Open Knowledge Base Connectivity* [Chaudhri *et al.*, 1998] é um protocolo *web* para apoiar o tráfego de ontologias e bases de conhecimento na internet, permitindo o fluxo de ontologias entre as plataformas e dando maior flexibilidade aos construtores e projetistas [Botelho *et al.*, 2002].

OIL [Fensel *et al.*, 2001] adiciona primitivas de representação de conhecimento baseadas em *frame* a RDF(S) e sua semântica formal é baseada em Lógica Descritiva. O classificador FaCT é utilizado para realizar classificações automáticas de conceitos.

DAML+OIL [Connolly *et al.*, 2001] também adiciona primitivas de representação de conhecimento baseadas em Lógica Descritiva às primitivas de RDF(S). Assim como OIL, DAML+OIL permite representar conceitos, taxonomias, relações binárias, funções e instâncias.

Finalmente, OWL [Dean, 2006] foi proposta como uma evolução de DAML+OIL. A OWL Web Ontology Language emerge como um novo padrão que é apoiado pela W3C para a definição de ontologias na web semântica. Segundo Smith *et al.* [2004], a OWL visa prover uma linguagem que pode ser usada para descrever as classes e relacionamentos entre elas, que estão implícitos nos documentos web e aplicações. Esta linguagem pode ser usada para formalizar um domínio através da definição de classes e suas propriedades; definir indivíduos e afirmar propriedades sobre eles e prover raciocínios lógicos sobre estas classes e indivíduos de acordo com o grau permitido pela semântica formal da OWL.

OWL provê um conjunto de vocabulário mais rico do que o encontrado em RDF para melhor restringir o conjunto de triplas que podem ser representadas. Um documento OWL pode incluir um cabeçalho opcional de ontologia, classes, propriedades e descrições de indivíduos ou axiomas.

Ontologias em OWL podem ser categorizadas em três espécies ou sub-linguagens: OWL-Lite, OWL-DL e OWL-Full. A diferença entre cada sub-linguagem é sua expressividade e computabilidade. OWL-Lite é a menos expressiva, enquanto a OWL-Full é a mais expressiva, porém menos computável. A expressividade da OWL-DL fica entre as duas, o que permite uma boa representação enquanto possibilita inferências.

Apesar de existirem outras propostas para representação dos conceitos e seus relacionamentos em um nível mais abstrato, tais como ER e OntoUML [Guizzardi, 2005], optou-se por seguir a proposta da W3C por uma questão de adoção de padrões web e portanto promover a interoperabilidade. Abstrações sugeridas para o entendimento dos conceitos e seus possíveis relacionamentos e restrições tais como aquelas propostas em [Guizzardi, 2005] podem ser consideradas na metodologia para construção e manutenção da ontologia.

3.3 Linguagens de consulta de ontologias

As linguagens para consultas em ontologias são relativamente novas e podem ser encontradas muitas propostas distintas na literatura. Podem ser divididas em propostas baseadas em linguagens de consulta de banco de dados (isto é, SQL, OQL), propostas baseadas em linguagens de regras (isto é, PROLOG, LISP) e propostas baseadas em expressões de caminho (principalmente linguagens para consultas em documentos XML, como XPATH e XQUERY).

Nesta seção, um número reduzido de linguagens é apresentado. Como não existe um padrão *de facto* e é impraticável descrever todas as propostas existentes na literatura neste relatório, foi utilizada a seguinte metodologia para escolha das linguagens a serem descritas nesta seção: (1) pesquisar linguagens reconhecidas e/ou definidas pela W3C, por ser este um consórcio internacional que objetiva desenvolver padrões para a Web e (2) realizar uma pesquisa empírica para selecionar as linguagens

mais citadas em artigos acadêmicos. Com essa metodologia, espera-se selecionar, das linguagens existentes, as mais utilizadas e que possuem maior possibilidade de se tornar um padrão futuro.

Assim, nessa seção, são descritas três linguagens para consultas em ontologias: nRQL, OWL-QL e SPARQL. As duas primeiras foram escolhidas pela ocorrência na maior parte dos artigos acadêmicos pesquisados e a última foi escolhida por ser uma linguagem reconhecida pela W3C. Outras linguagens para consultas em ontologias podem ser observadas em [Gómez-Pérez *et al.*, 2002], [Magkanaraki *et al.*, 2002], [Haase *et al.*, 2004].

3.3.1 nRQL

A linguagem nRQL [Haarslev *et al.*, 2004] faz parte do sistema RACER (*Renamed Abox and Concept Expression Reasoner*), um processador otimizado de consultas para OWL-DL criado para ser altamente efetivo e eficiente [Haarslev e Möller, 2003]. Uma consulta nRQL é composta de um cabeçalho e um corpo. O corpo da consulta consiste de uma expressão enquanto o cabeçalho corresponde às variáveis referenciadas no corpo da consulta, as quais serão exibidas no resultado.

A nRQL permite a formulação de consultas conjuntivas. Em uma consulta nRQL, as variáveis de consulta são usadas para representar os indivíduos ABox³ que satisfazem a consulta. As consultas irão fazer uso de conceitos arbitrários. Os elementos TBox⁴ (ou definições de classes OWL) definem o vocabulário de domínio a ser explorado na consulta.

Uma variável em nRQL somente é ligada a um indivíduo ABox se este indivíduo satisfaz a consulta. Satisfazer uma consulta significa que o resultado encontrado ao substituir todas as variáveis com suas ligações é uma implicação lógica da base de conhecimento.

Para um exemplo de uso da linguagem nRQL, considere a figura abaixo, onde está modelado, de forma muito simples, duas relações de parentesco (*tem_filho* e *tem_mãe*) entre três indivíduos (Alice, Betty e Charles).

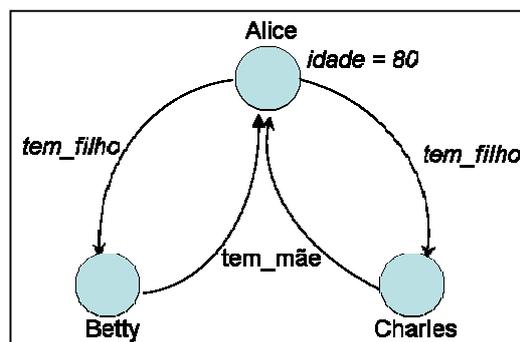


Figura 4 – Relações de parentesco

A Figura 5 exemplifica uma consulta nRQL simples.

³ Elementos ABox são fatos associados a uma terminologia dentro de uma base de conhecimento. Num ontologia, os elementos ABox estão associados a instâncias de um conceito.

⁴ Elementos TBox são componentes terminológicos, ou seja, um vocabulário associado a um conjunto de fatos ABox. Em uma ontologia, elementos TBox estão associados a classes e propriedades.

```

(Retrieve (?x ?y)
  (and
    (tem_filho ?z ?x)
    (tem_filho ?z ?y)
  )
)

```

Figura 5 – Consulta nRQL simples

A consulta acima, quando aplicada no modelo da Figura 4, retorna os elementos descritos na Figura 6, ou seja, elementos que possuem mesma mãe (tem_filho ?z ?x e tem_filho ?z ?y), isto é, elementos irmãos.

```

(
  ( (?x Charles) (?y Betty) )
  ( (?x Betty) (?y Charles) )
)

```

Figura 6 – Resultado da consulta nRQL

A linguagem nRQL também possui capacidade de lidar com restrições, comparações de elementos (modificador “*same-as*”), negações de elementos e propriedades, união de consultas, operações de projeção, restrições numéricas, etc.

Uma das características singulares do nRQL é a sua capacidade de processar negação por falha [Palazzo, 1997] para o formalismo OWL [Racer, 2005]. Este tipo de negação é útil para medir o grau de completude de uma modelagem de domínio na base de conhecimento. Por exemplo, é possível realizar uma consulta que recupere os indivíduos do ABox que implicam ser mulheres, mas que não implicam ser mães.

3.3.2 OWL-QL

Quando se trabalha com OWL, por ter sua sintaxe baseada em RDF/XML, pode-se aplicar qualquer linguagem de consulta compatível com RDF/XML, como SPARQL ou XQuery. Entretanto, uma ontologia muitas vezes necessita de um poder expressivo maior do que RDF/XML pode oferecer e por isso as linguagens de consulta baseadas nesse modelo têm a limitação de não conseguir capturar a semântica dos elementos OWL que estão além das definições do RDF/XML. Por outro lado, espera-se encontrar na Web Semântica uma grande variedade de informações representadas em muitos formatos. Uma linguagem para fazer consulta a ontologias, que explore todo o poder expressivo da OWL deve ser capaz de suportar a heterogeneidade da Web Semântica, além de oferecer suporte ao raciocínio automático. Por isso a linguagem OWL-QL [Fikes *et al.*, 2004] (“OWL Query Language”) é uma forte candidata à recomendação pelo W3C [Drumond e Girardi, 2006].

A linguagem OWL-QL foi proposta para ser, ao mesmo tempo, uma linguagem de consulta e um protocolo para diálogos entre agentes computacionais da Web Semântica utilizando o conhecimento representado em OWL. A OWL-QL é uma linguagem formal que especifica os relacionamentos semânticos entre uma consulta, uma resposta da consulta e uma base de conhecimento usada para produzir a resposta obtida.

A formulação de respostas para consultas em OWL-QL pode ser feita através de diálogos onde um agente pode utilizar métodos automatizados de inferência (*reasoning*) para derivar as respostas. O conhecimento a ser usado ao responder uma consulta pode estar em múltiplas bases de conhecimento na Web Semântica, podendo ou não essas bases de conhecimentos ser especificadas pelo agente de consulta. Neste caso, o conjunto de respostas de uma consulta pode conter um tamanho imprevisível e pode requerer uma quantidade de tempo igualmente imprevisível para ser respondida [Fikes *et al.*, 2004].

Consultas em OWL-QL geralmente contêm muitas linhas de descrição em XML. Por esse motivo, utiliza-se neste relatório o exemplo apresentado em [Drumond e Girardi, 2006], onde o XML foi convertido em uma sintaxe abstrata da linguagem OWL-QL. Esta sintaxe especifica a consulta em linguagem natural, além de um padrão de consulta, expresso por uma série de sentenças do tipo *propriedade - objeto - valor* com algumas variáveis, cujos valores serão retornados ou não como resposta (conforme especificado abaixo do padrão de consulta) e um padrão que especifica o formato esperado da resposta.

Uma consulta OWL-QL simples é um objeto que contém um padrão de consulta com várias sentenças OWL dentro das quais existem algumas referências a URIs que podem ser variáveis [Fikes *et al.*, 2004]. A Figura 7 apresenta um exemplo de consulta simples, onde o cliente deseja saber o tipo (lei ordinária, lei complementar, decreto, etc) do instrumento jurídico normativo de número 10856. Para responder à consulta *Qual o tipo do instrumento jurídico normativo de nº 10856?* o servidor iria procurar entre as ontologias disponíveis, por uma que tratasse dos instrumentos jurídico normativos, provavelmente hospedada em um sítio governamental, e procuraria pela instância de número 10856 para então fornecer a resposta [Drumond e Girardi, 2006].

```
Consulta: (Qual o tipo do instrumento jurídico normativo de nº
10856?)
Padrão de Consulta: {(numero ?inst 10.856) (tipo ?inst ?tipo)}
Lista de variáveis que devem ser retornadas: (?tipo)
Lista de variáveis que podem ser retornadas: ()
Lista de variáveis que não devem ser retornadas: (?inst)
Padrão de resposta: {(tipo 10.856 ?tipo )}
Resposta: ( O tipo do instrumento jurídico normativo de nº 10856 é
Lei Ordinária )
Instância do Padrão de Resposta: {(tipo 10.856 Lei Ordinária )}
```

Figura 7 – Exemplo de consulta simples em OWL-QL [Drumond e Girardi, 2006]

Também é possível fazer consultas do tipo *se-então*. Neste tipo de consulta, a mesma vem precedida por premissas que podem ser bases de conhecimento ou referências para outras bases. Deste modo facilita-se o uso de um operador de implicação, inexistente em OWL. A Figura 8 ilustra uma consulta com premissa, no caso, a premissa é “*João é uma pessoa*” e a consulta, “*João tem pai?*”. Através da informação contida na premissa de que João é uma pessoa e de informações escritas em ontologias de que toda pessoa tem pai e mãe, um agente dotado de capacidade de raciocínio responderia afirmativamente a consulta [Drumond e Girardi, 2006].

```

Consulta: Se João é uma pessoa, então João tem pai?
Premissa: {(type João pessoa)}
Padrão de Consulta: {(pai João ?pai)}
Lista de variáveis que devem ser retornadas: (?pai)
...

```

Figura 8 – Exemplo de consulta com premissa em OWL-QL [Drumond e Girardi, 2006]

3.3.3 SPARQL

A linguagem SPARQL [Prud'hommeaux e Seaborne, 2004] é uma linguagem recomendada pela W3C para consulta RDF, o que a torna candidata a um futuro padrão *de facto*. Por possuir uma sintaxe semelhante à da linguagem SQL, as consultas em SPARQL são intuitivas e de mais fácil compreensão para profissionais com experiência em bancos de dados. A SPARQL pode ser usada para expressar consultas entre diversas fontes de dados, podendo ser o dado armazenado como um documento RDF ou visualizado como RDF via um *middleware*. A linguagem contém capacidades de consultar padrões opcionais ou obrigatórios de grafos com suas conjunções e disjunções, além de possibilitar expressar restrições nos termos RDF que aparecerão no resultado da consulta. O resultado das consultas pode ser um conjunto de grafos em RDF.

Na figura abaixo, está exemplificada uma consulta simples, onde são listados somente elementos que possuem o atributo "name" e "mbox".

```

:Dados originais:
@prefix foaf: <http://xmlns.com/foaf/0.1/>

_:a foaf:name "Johnny Lee Outlaw"
_:a foaf:mbox <mailto:jlow@example.com>
_:b foaf:name "Peter Goodguy"
_:b foaf:mbox <mailto:peter@example.org>
_:c foaf:mbox <mailto:carol@example.org>

:Consulta:
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE
{
  ?x foaf:name ?name
  ?x foaf:mbox ?mbox }

```

Figura 9 – Consulta simples em SPARQL [Prud'hommeaux e Seaborne, 2004]

Na tabela abaixo, o resultado da consulta acima, contendo os únicos dois elementos que possuem os atributos desejados. Note que o elemento `_:c` possui somente o atributo "mbox" e, por esse motivo, não aparece na tabela de resultado.

Name	Mbox
"Johnny Lee Outlaw"	<mailto:jlow@example.com>
"Peter Goodguy"	<mailto:peter@example.org>

Tabela 2 – Resultado da consulta em SPARQL [Prud’hommeaux e Seaborne, 2004]

A seguir, é apresentado um exemplo de consulta em que são listados todos os elementos que possuem atributo “name” e que podem ou não possuir atributos “mbox” ou “hpage”.

```

:Dados originais:
@prefix foaf:      <http://xmlns.com/foaf/0.1/>

_:a  foaf:name      "Alice"
_:a  foaf:homepage  <http://work.example.org/alice/>

_:b  foaf:name      "Bob"
_:b  foaf:mbox      <mailto:bob@work.example>

:Consulta:
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox ?hpage
WHERE {
  ?x foaf:name ?name
      OPTIONAL { ?x foaf:mbox ?mbox }
      OPTIONAL { ?x foaf:homepage ?hpage }
}

```

Figura 10 – Consulta SPARQL com atributos opcionais [Prud’hommeaux e Seaborne, 2004]

Na Tabela 3 é apresentado o resultado da consulta acima.

name	mbox	hpage
"Alice"		<http://work.example.org/alice/>
"Bob"	<mailto:bob@work.example>	

Tabela 3 – Resultado da consulta em SPARQL com atributos opcionais [Prud’hommeaux e Seaborne, 2004]

Além dos modificadores de consultas apresentados anteriormente, o SPARQL possui em sua sintaxe modificadores para realizar união de resultados, selecionar *namespaces*, ordenar e limitar o resultado, prover resultados únicos, contar e agrupar resultados, etc.

3.3.4 Considerações finais

A tabela abaixo resume as características principais das linguagens de consulta de ontologias descritas anteriormente.

Linguagem	Formalismo	Inferência	Recomendação W3C	Sintaxe
nRQL	OWL-DL	Sim	Não	Baseada em LISP
OWL-QL	OWL	Sim	Não	Baseada em XML

SPARQL	RDF	Não	Sim	Baseada em SQL
--------	-----	-----	-----	----------------

Tabela 4 – Comparação entre linguagens de consulta de ontologias

Entre as linguagens citadas, a linguagem SPARQL é provavelmente a mais utilizada por ser a única com recomendação da W3C. Embora essa linguagem não seja específica para OWL, ela é utilizada para ler ontologias descritas em OWL, uma vez que este formalismo possui construtores de RDF. Contudo, ao não ser capaz de ler os construtores de OWL, a linguagem não é capaz de realizar inferências. Porém, alguns trabalhos propõem uma extensão da linguagem para trabalhar com OWL-DL [Glimm, 2007], [Sirin e Parsia, 2007].

3.4 Reuso de Ontologias

Uma vez que ontologias possuem a característica de prover interoperabilidade semântica entre fontes de informação, é comum encontrarmos a palavra *integração* sendo usada para descrever tanto a interoperabilidade semântica resultante de seu uso, quanto a compatibilidade entre as possíveis ontologias distintas usadas para gerar essa interoperabilidade semântica (sendo elas de um mesmo domínio ou não).

Contudo, no cenário onde mais de uma ontologia são utilizadas e, por isso, é necessário compatibilizá-las encontrando suas similaridades e diferenças, a *integração* é somente uma das aproximações que podem ser aplicadas para o processo de compatibilidade de ontologias. Sendo assim, o termo *integração* é utilizado nesse documento como um dos mecanismos para compatibilizar ontologias. O processo de compatibilização de ontologias permite a reutilização da estrutura de conhecimento, seja em diferentes aplicações, seja na construção de novas estruturas de conhecimento.

Descrevemos nas seções abaixo os principais mecanismos para compatibilizar ontologias. São eles: (1) combinação de ontologias [Noy e Musen, 2001], (2) alinhamento de ontologias [Noy e Musen, 1999], (3) integração de ontologias (PINTO *et al.*, 1999) e (4) mapeamento de ontologias [Noy e Musen, 2003]. Outros mecanismos podem ser encontrados em [Klein, 2001].

3.4.1 Combinação

A combinação é o processo de construção de uma ontologia de um domínio, reutilizando duas ou mais ontologias diferentes daquele domínio. Neste processo, as ontologias-fontes são unificadas em uma única ontologia e, dessa forma, se torna difícil identificar na ontologia resultante regiões que foram retiradas das ontologias combinadas e que foram deixadas mais ou menos inalteradas. Normalmente as ontologias originais descrevem domínios similares ou de alguma sobreposição.

A Figura 11 ilustra um exemplo de combinação de ontologias, onde os dois conceitos compatíveis, **carro** da ontologia **O1** e **veículo** da ontologia **O2**, são combinados, isto é, unidos, na ontologia única **O**.

Como pôde ser visto na figura abaixo, um conjunto de ontologias (pelo menos duas) vão ser combinadas (O_1, O_2, \dots, O_n) e ter como produto a ontologia resultante **O**. O objetivo é criar uma ontologia mais geral sobre um assunto ao reunir em uma única estrutura coerente, o conhecimento de várias outras ontologias sobre aquele mesmo

assunto. O assunto S das ontologias combinadas resultante é o mesmo, apesar de algumas ontologias serem mais gerais que outras.

Deve-se salientar que no processo de combinação as ontologias-fontes são verdadeiramente ontologias diferentes e não apenas revisões, melhoramentos ou variações da mesma ontologia.

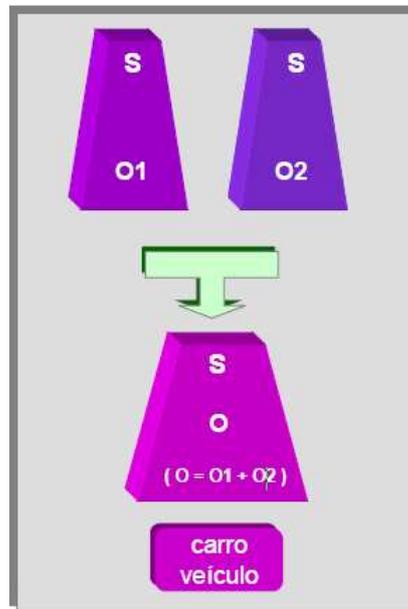


Figura 11 – Combinação de ontologias, adaptado de [Felicíssimo, 2004]

3.4.2 Alinhamento

O alinhamento de ontologias estabelece ligações entre duas ontologias para permitir que as ontologias alinhadas reusam informação umas das outras. Essa ligação se torna um acordo mútuo entre as ontologias para que se tornem consistentes e coerentes [Klein, 2001]. Uma vez que a ontologia alinhada está reusando o conhecimento representado em outra ontologia através de uma ligação, esta ontologia se torna dependente do conhecimento representado nesta ontologia.

No alinhamento tem-se como entrada duas ontologias que normalmente descrevem domínios complementares e, como resultado, têm-se as duas ontologias originais separadas, mas nestas são adicionadas as ligações entre seus termos complementares [Felicíssimo, 2004], conforme ilustra a Figura 12.

Na figura abaixo, os conceitos da ontologia O1 reusam conceitos da ontologia O2. No exemplo, o conceito carro da ontologia O1 é ligado ao conceito veículo da ontologia O2 por uma relação de hierarquia, ou seja, um carro é um tipo de veículo. Assim, o conceito carro da ontologia O1 herda as definições do conceito veículo da ontologia O2, enriquecendo a ontologia semanticamente. Contudo, as ontologias se tornam dependentes, ou seja, para se entender o conceito “carro”, sempre será necessário conhecer o conceito “veículo” o qual ele herda suas características.

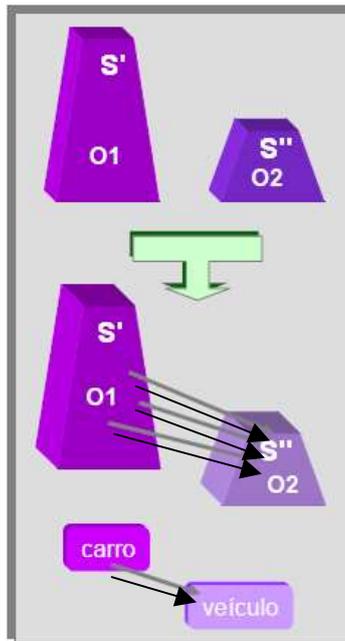


Figura 12 – Alinhamento de ontologias, adaptado de [Felicíssimo, 2004]

A Figura 13 ilustra um exemplo de alinhamento de ontologias, onde os conceitos **gás**, **água**, **óleo**, **petróleo** e **petróleo retirado** da ontologia de **Fluidos** são alinhados, i.e., ligado, ao conceito **Fluidos** da ontologia de **Poços**, herdando sua definição.

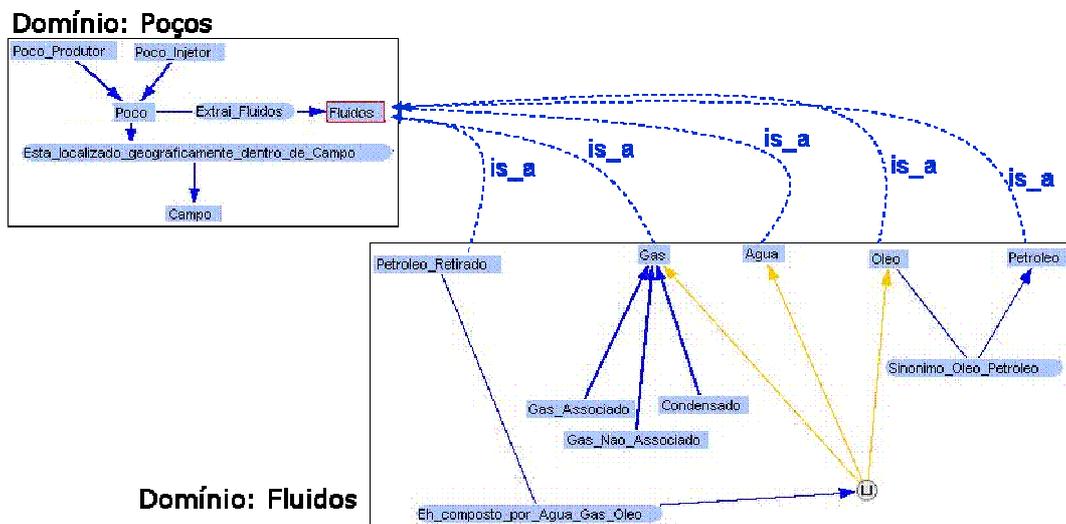


Figura 13 – Alinhamento de ontologias

3.4.3 Integração

Neste processo, as ontologias-fonte (geralmente de domínios distintos) são agregadas, combinadas para formar a ontologia resultante (com um domínio mais amplo ou genérico), possivelmente depois as ontologias reutilizadas terão sofrido alguma mudança, como extensão, especialização ou adaptação. Em um processo de integração, podem-se identificar na ontologia resultante regiões que foram retiradas das ontologias integra-

das. O conhecimento descrito nestas regiões permanece praticamente inalterado. A Figura 14 ilustra um exemplo de integração de ontologias, onde os conceitos **carro** de **O1**, **veículo** de **O2**, **automóvel** de **O3** e **meio de transporte terrestre** de **O4** são integrados, i.e., unidos, na ontologia única **O**.

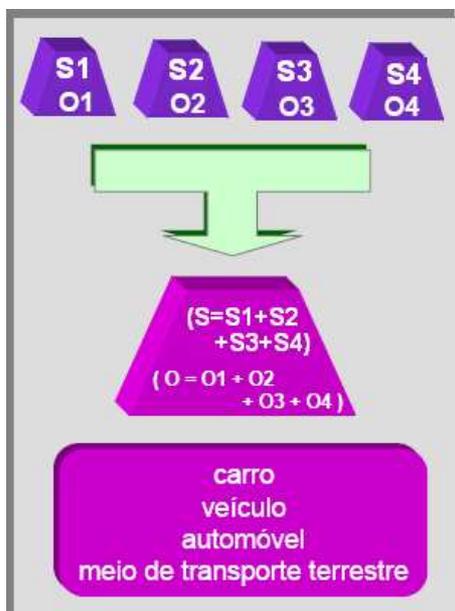


Figura 14 – Integração de ontologias, adaptado de [Felicíssimo, 2004]

Como ilustra a figura acima, tem-se ontologias que serão integradas (O_1, O_2, \dots, O_n), e a ontologia resultante do processo de integração. As ontologias integradas são aquelas que serão reutilizadas. Elas são partes da ontologia resultante. A ontologia resultante do processo de integração é a que deseja-se construir e, apesar dela ser referenciada como uma ontologia, ela pode ser composta de vários “módulos”, que são (sub)ontologias. Isto acontece não só na integração, mas também quando se começa a construir uma ontologia sem reutilizar conhecimento previamente estruturado.

Quando a ontologia integrada é reutilizada pela ontologia resultante, os conceitos integrados podem ser, entre outras coisas [Resnik *et al.*, 2005]: (1) utilizados como estão, (2) adaptados (ou modificados), (3) especializados (levando a uma ontologia mais específica no mesmo domínio) ou (4) acrescidos de novos conceitos (ou por conceitos mais gerais ou por conceitos do mesmo nível).

Os domínios das ontologias integradas usualmente são diferentes entre si (representados na figura através de assuntos diferentes S_1, S_2, S_3 e S_4), ou seja, cada ontologia que foi integrada à ontologia resultante geralmente diz respeito a um domínio diferente, mas esses domínios são relacionados de alguma forma.

Na integração, a ontologia resultante não deveria se assemelhar a nenhuma outra ontologia previamente existente, caso contrário dever-se-ia simplesmente reutilizar esta ontologia já existente. As ontologias reutilizadas são escolhidas entre as disponíveis nas bibliotecas de ontologias que se encaixam com uma série de requisitos, por exemplo, domínio, abstração, tipo, generalidade, modularidade, e outros mais [Rodríguez e Egenhofer, 2003].

Apesar do resultado final tanto da combinação quanto da integração de ontologias ser uma ontologia única, constituída pela união dos termos das ontologias origi-

nais, a principal diferença entre estes dois mecanismos é que, no primeiro, as ontologias tratam do mesmo assunto, o que não acontece necessariamente no segundo.

3.4.4 Mapeamento

No mapeamento de ontologias tem-se como resultado uma estrutura formal com expressões que relacionam conceitos ou relações similares de uma fonte diferente para a outra através de relações de equivalência. Essas relações de equivalência expressam a similaridade dos conceitos (por exemplo, está contido, contém, é disjunto, é igual, etc) e não representam uma dependência entre as duas ontologias, uma vez que as estruturas de conhecimento **não estão** estendendo ou reutilizando conceitos uma da outra. Este mapeamento pode ser usado para transferir instâncias de uma ontologia para a outra, criar esquemas de integração e de combinação, e outras tarefas similares. No mapeamento não existe dependência entre os modelos e a similaridade entre os conceitos é representada por uma ligação (normalmente, utilizando noções da teoria de conjuntos: é igual, diferente, composto, disjunto, está contido, contém, etc.), enquanto no alinhamento há o reuso de conceito de outras ontologias **através de ligações ontológicas** (is-a, part-of, propriedades de classes ontológicas). A Figura 15 ilustra um exemplo de mapeamento de ontologias, onde os conceitos **carro** de **O1** e **veículo** de **O2** são mapeados como, por exemplo, conceitos equivalentes.

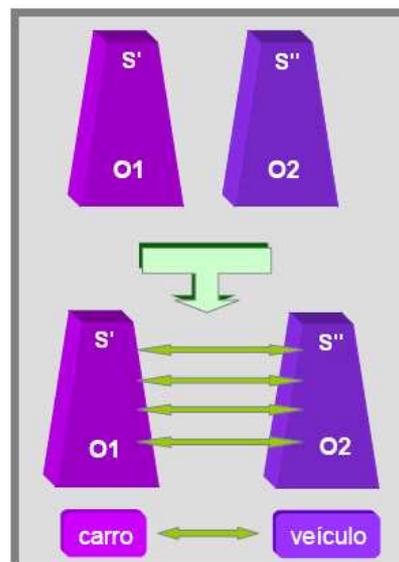


Figura 15 – Mapeamento de ontologias, adaptado de [Felicíssimo, 2004]

No exemplo acima, os conceitos da ontologia O1 são mapeados aos conceitos da ontologia O2. Esses mapeamentos podem ser unidirecionais ou bidirecionais. Além disso, esses mapeamentos podem conter regras de transformação que são utilizadas quando as instâncias forem transferidas de uma ontologia para outra. Essas regras de transformação são, normalmente, cálculos dos valores de propriedades das instâncias para transformação de unidades de medida (metro para milha, celsius para fahrenheit, etc), valores (real para dólar), etc. No exemplo acima, o conceito Carro da ontologia O1 é mapeado no conceito veículo da ontologia O2 utilizando, por exemplo, um mapeamento de “equivalência”. Ou seja, as instâncias de Carro podem ser incluídas na ontologia O2 como instâncias de Veículo. Note que esses dois conceitos não são exatamente o mesmo conceito (carro é uma especialização de veículo). Porém, uma vez que onto-

logias podem ter diferentes granularidades, esse tipo mapeamento é viável e permite que as instâncias da ontologia O1 possam ser transferidas para a ontologia O2 sem perda de consistência.

Existem pelo menos duas formas diferentes de ser fazer esse mapeamento entre ontologias. Na primeira delas, as duas ontologias a serem mapeadas compartilham uma ontologia de referência comum, como mostrado na Figura 16. Ontologias genéricas⁵, como SUMO [Niles e Pease, 2003], OpenCyc [Bunningen, 2004] e DOLCE [Gangemi *et al.*, 2003], às vezes são usadas em algumas arquiteturas de ontologias mapeadas pois facilitam o compartilhamento de conhecimento. As duas primeiras ontologias são padrões do grupo de trabalho de ontologias genéricas do IEEE - *Institute of Electrical and Electronics Engineers* [Suowg, 2006].

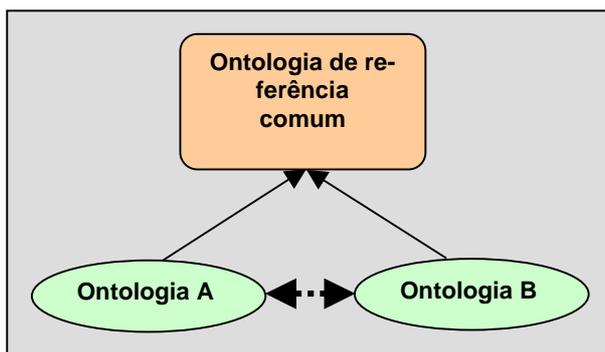


Figura 16 – Mapeamento de ontologias utilizando uma ontologia de referência

Em [Gruninger e Kopena, 2005], os autores propõem um mapeamento de ontologias que é baseado na idéia de uma linguagem compartilhada, na qual não existem mapeamentos diretos entre as ontologias, mas apenas para a linguagem, como mostra a figura seguinte.

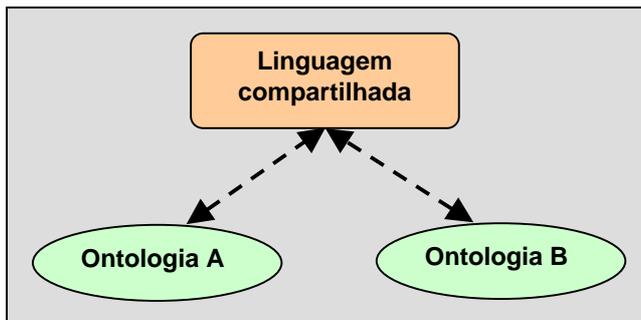


Figura 17 – Mapeamento de ontologias utilizando uma linguagem compartilhada

Quando não existe uma ontologia compartilhada, pode-se usar ferramentas como QOM [Ehrig e Staab, 2004], Similarity Flooding [Melnik *et al.*, 2002] ou Prompt [Noy e Musen, 2003] para realizar um mapeamento diretamente de uma ontologia à outra, como ilustra a Figura 18.

⁵ Ontologias genéricas, ou *upper ontologies*, são aquelas que possuem descrições mais gerais, ou seja, limitam-se a conceitos que são meta, genéricos, abstratos e filosóficos. Conceitos específicos de um dado domínio não são incluídos nas ontologias genéricas. Assim, estas ontologias fornecem estrutura e conceitos genéricos o suficiente para serem utilizados, em um nível elevado, na construção de outras ontologias de várias áreas de domínio [Suowg, 2006].

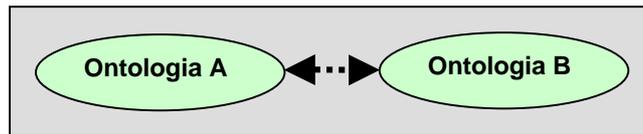


Figura 18 – Mapeamento entre ontologias sem recursos compartilhados

A interação com o usuário é outra fonte importante de informação. Muitos pesquisadores acreditam que um mapeamento completamente automático é impossível de ser feito e, assim, é necessário que haja alguma intervenção do usuário durante o processo. Essa interação pode incluir: alimentar o algoritmo de mapeamento com um conjunto inicial de pares, verificar os pares mapeados que o algoritmo produz ou configurar os cálculos a serem usados [Mitra *et al.*, 2000], [Mcguinness *et al.*, 2003], [Noy e Musen, 2003].

3.4.5 Divergências encontradas durante o reuso de ontologias

Encontrar correspondências entre ontologias é um dos problemas mais difíceis na pesquisa sobre ontologias. Esse problema surge sempre que duas ou mais ontologias com alguma sobreposição de termos necessitem ser usadas numa única aplicação ou por um único agente de *software*. No mundo ideal, existiriam ontologias padrões descrevendo modelos de diferentes domínios: uma ontologia para cada área da medicina, uma para processos de negócios, uma para aplicações de viagens, e assim por diante. Contudo, não apenas este não é o cenário atual, onde várias ontologias descrevem o mesmo cenário, como também é provável que a situação piore no futuro: quanto mais ontologias forem sendo desenvolvidas, mais ontologias com conteúdo similar ou sobreposto existirão. Não é razoável esperar que as pessoas concordarão com um pequeno conjunto de ontologias com pequenas ou nenhuma sobreposição. As razões vão das práticas (diferentes aplicações requerem diferentes visões de um domínio) às institucionais e sociais (uma ontologia desenvolvida em outro lugar pode não ser tão boa quanto à ontologia desenvolvida por nós para nossas finalidades).

Todavia, aplicações que usam diferentes ontologias para descreverem seus domínios ainda necessitam interoperar. E, para isso, necessita-se encontrar correspondências entre diferentes ontologias. Dado duas ontologias, é preciso identificar onde estão as similaridades e diferenças entre elas, e expressar essas correspondências (com um mapeamento entre as ontologias, por exemplo) de uma maneira processável por máquinas.

Quando duas ontologias são comparadas, podem ser identificadas algumas divergências entre elas. É importante identificar quais tipos de divergências ocorrem entre duas ontologias, a fim de resolvê-las durante o reuso de ontologias. A classificação dessas divergências também é importante para denotar o que pode ser resolvido com o mapeamento de um formalismo (linguagem de descrição de ontologias) e o que pode ser resolvido com a ajuda de um algoritmo para cálculo de similaridade.

Klein [2001] identificou dois níveis de divergências entre ontologias. O primeiro nível é o da linguagem de descrição ou meta-modelo. As divergências inclusas neste nível são as diferenças sintáticas, diferenças no significado das primitivas nas diferentes linguagens e diferenças na expressividade das linguagens. O segundo nível de divergências é o da ontologia ou modelo, chamadas aqui de nível da conceitualização. Essas divergências são detalhadas nas seções seguintes.

3.4.5.1 Divergências no nível da linguagem

Normalmente, os mapeamentos entre ontologias requerem que as duas ontologias sejam representadas na mesma linguagem. A tradução de uma linguagem para outra pode resolver a maioria dos problemas que podem ocorrer com a diferença de representação. Divergências típicas no nível de linguagem são a sintaxe, a representação lógica, a semântica das primitivas e a expressividade da linguagem [Klein, 2001]. Como pode ser lido em [Predoiu *et al.*, 2006], os métodos e ferramentas para mapeamento necessitam que as ontologias sejam representadas no mesmo formalismo e, mesmo que exista uma tradução de qualquer linguagem ontológica para o formalismo desejado, a preservação da semântica pode ainda não ser garantida, pois toda tradução pode gerar perdas semânticas.

Diferenças na sintaxe ocorrem quando linguagens diferentes são usadas na representação de cada ontologia. Por exemplo, para representar o conceito “carro” na linguagem LOOM [Loom, 2006] usa-se a expressão (defconcept Carro). O mesmo conceito usando a linguagem RDF Schema seria representado com a construção <rdfs:Class ID=“Carro”>. Tal divergência pode ser resolvida através da tradução da ontologia para uma representação qualquer, desde que, no final do processo, ambas as ontologias usem a mesma sintaxe. Essa é uma das divergências mais simples de serem contornadas, porém essa divergência quase nunca ocorre sozinha [Jakoniene, 2006]. Uma diferença que pode ser consequência dessa divergência sintática é a encontrada na representação lógica, quando estruturas sintaticamente diferentes, mas logicamente equivalentes, são usadas para representar a mesma coisa. Um exemplo dessa divergência pode ser visto ao representar duas classes disjuntas. Em algumas linguagens é possível representar a disjunção de forma explícita (por exemplo, disjoint A B), em outras é necessário usar negações nas subclasses (por exemplo, A subclass-of (NOT B), B subclass-of (NOT A)). Nota-se que a divergência não está na capacidade de expressar algo, pois as estruturas são equivalentes, mas quais construtores da linguagem deveriam ser usados para realizar essa explicitação. Além disso, nota-se que essa divergência não diz respeito à representação dos conceitos, mas à representação das noções lógicas. Este tipo de divergência pode ser resolvido ao usar regras de tradução de uma representação lógica para outra.

Quando a semântica das primitivas é diferente nas diferentes linguagens para descrição de ontologias, isto é, um construtor sintaticamente equivalente tem um significado diferente nas diferentes linguagens, a tradução para uma representação comum precisa levar isto em conta. Por exemplo, existem diferentes interpretações para A equalTo B [Klein, 2001]. Esta divergência também pode ser resolvida com a tradução para uma representação comum, uma vez que, se duas ontologias já utilizam uma representação comum e esta representação não permite estruturas ambíguas, então esta divergência não ocorrerá. A tradução para uma representação comum também é solução para as divergências na expressividade das linguagens, que ocorre quando uma linguagem é capaz de expressar verdades que uma outra linguagem não pode. Por exemplo, algumas linguagens possuem construtores para expressar negação, listas, conjuntos, valores padrão, etc; e outras linguagens podem não possuir alguma dessas capacidades. Essa é a divergência que pode causar maior impacto no processo de tradução, principalmente quando a expressividade da linguagem comum de representação não for um superconjunto da linguagem da ontologia-fonte. Neste caso, alguma semântica pode se perder na tradução [Mitra *et al.*, 2000].

Como pôde ser visto, a tradução de linguagens é uma solução plausível para a maioria dos problemas que surgem com a divergência no nível da linguagem. Neste

trabalho, considera-se que as duas ontologias a serem mapeadas estão representadas por uma mesma linguagem de representação (OWL), uma vez que este problema já está sendo bastante explorado pela comunidade acadêmica [Visser *et al.*, 1997], [Bundy *et al.*, 2003], [Mcneill *et al.*, 2004], [Mcneill *et al.*, 2005].

3.4.5.2 Divergências no nível da conceitualização

Enquanto as divergências no nível da linguagem incluem diferenças na codificação e significado dos construtores das linguagens, as divergências no nível do modelo incluem diferenças no significado ou codificação dos conceitos nas diferentes ontologias. Essas divergências acontecem quando serão combinadas duas ou mais ontologias que descrevem domínios com alguma sobreposição e podem ocorrer quando as ontologias estão escritas na mesma linguagem ou em linguagens diferentes [Klein, 2001].

Essas divergências podem ser divididas entre (1) divergências de conceitualização, (2) divergências na explicação (ou estruturação) e (3) divergências terminológicas.

Uma **divergência de conceitualização** é uma diferença no modo em que um domínio é interpretado (conceitualizado), o que resulta em diferentes conceitos ou diferentes relações entre esses conceitos. Nas divergências que ocorrem na conceitualização, tem-se a diferença de escopo que ocorre quando duas classes parecem representar o mesmo conceito, mas não possuem exatamente as mesmas instâncias, apesar de alguma interseção. O exemplo comum para essa divergência é a classe “empregado” [Klein, 2001]: empresas podem usar conceitos ligeiramente diferentes de “empregado”. Por exemplo, como foi levantado em [Wiederhold, 1994], empregados podem ser nomeados como **funcionários** ou **pessoas** nos domínios de **folha de pagamento** e **recursos contratados**, respectivamente. Neste exemplo, a divergência ocorre quando se verifica que no domínio de **recursos contratados** podem-se incluir **pessoas** de outras instituições que não serão inseridas na **folha de pagamento**. Ainda, no domínio de **folha de pagamento**, talvez possa constar auxílios-escola como benefício para filhos de **funcionários**, porém estas crianças, apropriadamente, não estão inseridas no domínio de **recursos contratados**.

Outra divergência se encontra na abrangência do modelo e sua granularidade. Esta divergência é a diferença na parte do domínio que é representado por cada ontologia e no nível de detalhe no qual o domínio é modelado. A Figura 19 ilustra a diferença de granularidade entre a ontologia A, que representa diferentes tipos de computadores pessoais, e a ontologia B, que representa todos os computadores pessoais como uma única classe da ontologia. Chalupsky [2000] exemplifica essa divergência com uma ontologia sobre carros: uma ontologia pode modelar carros, mas não caminhões. Outra pode representar caminhões, mas apenas classificá-los em algumas poucas categorias, enquanto uma terceira ontologia pode explicitar várias distinções entre os tipos de caminhões baseados na sua estrutura física, peso, finalidade, etc.

Uma **divergência de explicação** é a diferença no modo em que a conceitualização é especificada. Ela pode ocorrer em divergências nas definições, termos ou combinação dos dois, ocorrendo por diferenças no estilo de modelagem usado. Quando diferentes paradigmas são usados para explicar um mesmo conceito, tem-se uma dessas divergências de explicação. Por exemplo, uma ontologia pode representar tempo usando intervalos, enquanto outra ontologia pode usar pontos. O uso de diferentes ontologias genéricas (*upper ontologies*) também é um exemplo desse tipo de divergência.

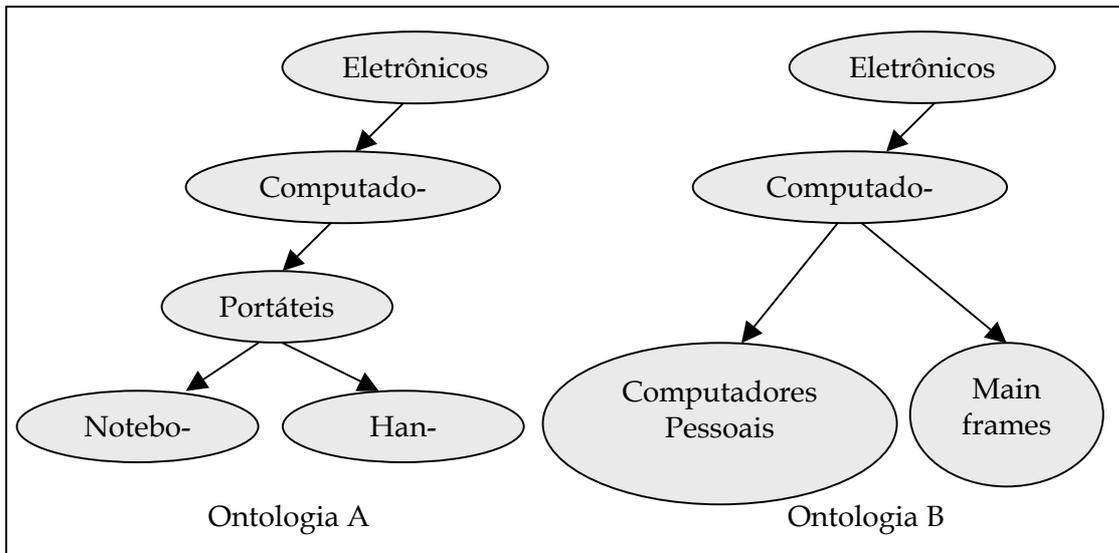


Figura 19 – Diferença de granularidade entre ontologias

Podem existir também divergências no modo em que um conceito é descrito como, por exemplo, as distinções entre duas classes: podem ser modeladas usando uma propriedade que as diferencia ou introduzindo uma classe separada. A Figura 20 exemplifica essa diferença: a ontologia A possui os conceitos Vinho Branco e Vinho Tinto para diferenciar tipos de Vinho, porém a ontologia B representa essa diferenciação nos tipos de vinhos adicionando o atributo “cor” ao conceito Vinho.

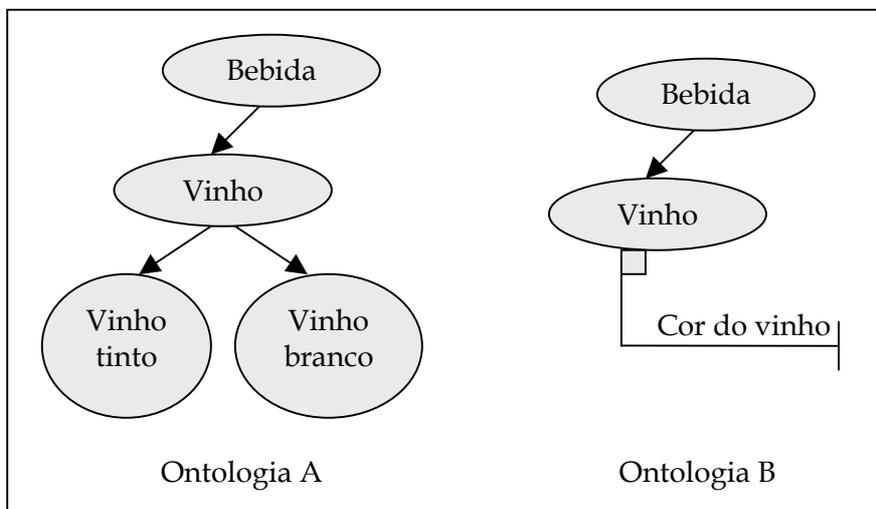


Figura 20 – Divergência no modo no qual o conceito é descrito

Por último, tem-se as divergências terminológicas, quando aparecem termos sinônimos⁶ ou homônimos⁷. Também é considerada uma divergência terminológica as diferenças na codificação. Essa diferença se dá quando valores nas ontologias podem ser codificados em formatos diferentes. Por exemplo, uma data pode ser representada

⁶ Conceitos com mesmo significado, mas com nomes diferentes.

⁷ Conceitos com mesmo nome, mas com significados diferentes em outros contextos. Por exemplo, o termo “condutor” quando usado no domínio sobre música tem um significado diferente do usado no domínio sobre engenharia elétrica.

como “dd/mm/aaaa” ou “mm-dd-aa”, distâncias podem ser medidas por quilômetros ou milhas, etc.

As diferenças encontradas na conceitualização não podem ser resolvidas automaticamente, segundo Klein [2001]. Por isso, requerem conhecimento e decisões de especialistas no domínio. Mesmo que as soluções técnicas para resolver as divergências terminológicas pareçam relativamente simples (uso de *thesauri* para reconhecer sinônimos, transformações para resolver diferenças na codificação de datas e unidades de medidas), a integração de ontologias com sinônimos ou de línguas também requerem esforço humano, pois apresentam vários problemas semânticos. Em especial as inconsistências com homônimos são muito difíceis de resolver computacionalmente e também exigem conhecimento humano para solucionar essa ambigüidade.

3.5 Avaliação da Qualidade de Ontologias

Ontologias são estruturas de dados fundamentais para conceitualização do conhecimento. Contudo, uma vez que é possível construir ontologias distintas que conceitualizam uma mesma parte do conhecimento, deve ser possível dizer quais dessas ontologias poderiam ser mais bem aplicadas levando em consideração algum critério predefinido e, antes disso, analisar a estrutura ontológica e identificar erros, inconformidades ou inconsistências quanto à semântica descrita na ontologia.

Segundo Brank *et al.* [2005], a avaliação de ontologias é o problema de pontuar uma dada ontologia do ponto de vista de um dado critério, tipicamente com a finalidade de determinar quais das muitas ontologias disponíveis poderiam melhor se encaixar a um propósito particular. A definição de Brank *et al.* está voltada para a aplicação da ontologia. Uma outra definição mais importante para a construção e manutenção de ontologias poderia ser “avaliar ontologias é o problema de validar estruturas ao expor escolhas inapropriadas e inconsistentes de modelagem”, inspirada na abordagem metodológica de Guarino e Welty [2002]. Essa definição trata de aspectos filosóficos, expondo a necessidade de se realizar uma análise metafísica das noções ontológicas mais gerais para avaliar ontologias.

Propostas para a avaliação de ontologias são encontradas na literatura, mas parecem existir poucas metodologias formais. Contudo, com uma análise das técnicas para avaliação de ontologias encontradas na literatura pode-se dividir as abordagens de avaliação naquelas que tratam de aspectos filosóficos e as que tratam de aspectos de aplicação, ou seja, seleção de ontologias para aplicações levando em consideração aspectos estruturais (hierarquia e relações não hierárquicas), léxicos e abrangência da ontologia (termos sintaticamente corretos e que abrangem todo o domínio que se pretende conceitualizar).

Gómez-Perez [1999a] apresenta critérios que podem ser utilizados para avaliar ontologias. Os passos apresentados para a avaliação focalizam-se sobre os conceitos e definições que compõem a ontologia:

- Verificar a estrutura ou arquitetura da ontologia: as definições são construídas seguindo os critérios de projeto?
- Verificar a sintaxe das definições: existem estruturas ou palavras-chave sintaticamente incorretas nas definições?
- Verificar o conteúdo das definições: o que a ontologia define ou não? O que define incorretamente? O que pode ser inferido e o que não pode?

Um método para avaliar ontologias lexicamente e sobre sua abrangência é apresentado em [Maedche e Staab, 2002], onde a similaridade entre dois termos é medida com base na distância de edição de Levenshtein [Euzenat *et al.*, 2004] normalizada para produzir resultados na faixa [0,1]. Uma medida de similaridade entre dois conjuntos de palavras é então definida ao selecionar cada palavra do primeiro conjunto, encontrar suas similaridades em relação à palavra mais similar do segundo conjunto, e ao calcular a média dessas similaridades entre todas as palavras do primeiro conjunto. Pode-se analisar o conjunto de todas as palavras usadas como identificadores de conceitos na ontologia avaliada e compará-las a um conjunto de “padrão dourado” (*golden standard*) que são considerados uma boa representação dos conceitos para o domínio sob consideração. Este padrão poderia ser outra ontologia ou poderiam ser consideradas palavras encontradas estatisticamente em um conjunto de documentos sobre o domínio ou, até mesmo, preparado por especialistas.

O método intitulado OntoMetric [Lozano-Tello, 2002], [López-Pérez, 2004] ajuda os engenheiros do domínio a escolher uma ontologia apropriada para um novo projeto; para realizar esse trabalho, o engenheiro necessita comparar a importância dos objetivos, e estudar cuidadosamente as características das ontologias. O método é baseado em um grupo de processos que ajudam a escolher a ontologia mais apropriada para ser reutilizada em um projeto em particular. O método provê uma medição sobre a conformidade de um conjunto de ontologias candidatas a serem incorporadas ao novo projeto. O método é baseado no Processo Analítico Hierárquico (AHP – *Analythic Hierarchy Process*) [Saaty, 1964], um método de decisão multicritério, adaptando alguns processos para o reuso de ontologias. Os critérios básicos de decisão do método, também chamados de dimensões, são os aspectos fundamentais a serem considerados pelo usuário antes de escolher a ontologia. As dimensões especificam características como o conteúdo representado pela ontologia, a linguagem implementada, a metodologia usada para sua criação, o ambiente de software usado, os custos de utilizar a ontologia no sistema, etc.

O2 e oQual são meta-ontologias que caracterizam uma ontologia como elementos semióticos [Gangemi *et al.*, 2006] e permitem realizar tarefas de avaliação e validação. Como semiótica, entende-se uma ontologia constituída de um objeto de informação (p.ex. uma versão OWL-DL da ontologia FOAF⁸ – *Friend Of A Friend*) e uma conceitualização desejada (p.ex. a relação conceitual entre pessoas, seus endereços e o conhecimento das outras pessoas) estabelecida dentro de um ambiente de comunicação (p.ex. uma tarefa, aplicação ou contexto de uso do FOAF) [Gangemi *et al.*, 2005]. As ontologias são analisadas pelos seus grafos e elementos formais, requisitos funcionais e perfis de anotação. A meta-ontologia oQual é uma ontologia sobre qualidade de ontologias que pretende fornecer o melhor conjunto de critérios para escolher uma ontologia no contexto de um dado projeto.

A metodologia OntoClean [Guarino e Welty, 2002] é baseada em noções formais, as quais são gerais o bastante para serem utilizadas em qualquer ontologia, independentemente de um domínio particular. Os autores utilizam-se dessas noções para definir um conjunto de metapropriedades que são usadas para caracterizar aspectos relevantes do significado pretendido das propriedades, classes e relações que compõem a ontologia. Essas metapropriedades impõem restrições na estrutura taxonômica da ontologia, ajudando na avaliação das escolhas feitas na modelagem. Entre essas metapropriedades, estão:

⁸ <http://www.foaf-project.org>

- **Essência:** Uma propriedade de uma entidade é essencial para a entidade se a entidade sempre preserva a propriedade. Por exemplo, “ser duro” é uma propriedade essencial para entidade machado, mas não para entidade esponja, mesmo que algumas esponjas sejam duras em algumas ocasiões.
- **Rigidez:** Esta é uma forma especial de essência que se refere a uma propriedade que é essencial a todas as instâncias. Por exemplo, todas as entidades que são pessoas terão a propriedade “ser humano”, e não haverá entidades que podem ser humanas, mas não o são.
- **Identidade:** Refere-se ao problema de ser capaz de reconhecer entidades no mundo como sendo a mesma entidade (ou uma entidade diferente).
- **Unidade:** Refere-se ao problema de ser capaz de reconhecer todas as partes que formam uma entidade.

Com a verificação dessas metapropriedades e as restrições que elas impõem a cada entidade, a OntoClean provê um meio formal e consistente para explicitar erros comuns durante a modelagem conceitual em relação à taxonomia e a especificação de relações.

Haase e Sure [2005] discutem a avaliação de ontologias durante sua evolução em diferentes cenários. A avaliação da ontologia se dá pelo resultado de funções matemáticas definidas pelos autores que consideram, principalmente, os pesos atribuídos em conceitos e propriedades por cada usuário da ontologia na ontologia pré- e pós-mudança, mapeando assim as alterações ocorridas na estrutura ontológica e analisando a profundidade da mudança. Algumas das funções de avaliação que foram criadas se utilizam das metapropriedades do OntoClean.

Um resumo sobre os métodos de avaliação apresentados é apresentado na tabela abaixo.

Método	Tipo de avaliação
(MAEDCHE e STAAB, 2002)	Avaliação léxica
OntoMetric	Avaliação por objetivo
oQual	Avaliação por objetivo
OntoClean	Avaliação semântica
(HAASE e SURE, 2005)	Avaliação semântica

Tabela 5 – Comparação entre métodos de avaliação de ontologias

3.6 Evolução de Ontologias

Além das instâncias de uma ontologia mudarem ao longo do tempo, através de inclusões, alterações e exclusões, sua própria estrutura pode necessitar de revisão, seja porque o entendimento dos conceitos existentes não foi capturado e modelado adequadamente, seja pela própria modificação do cenário da ontologia [Liang, 2005]. Também pode ser necessário alterar o formalismo da ontologia ou alterar o domínio de representação. Assim, essas mudanças não são simplesmente com o objetivo de reparar erros de modelagem omitidos durante a fase de construção da ontologia. A maioria das mudanças é consequência dos novos requisitos (sejam necessidade de novos conhecimentos, alteração da abrangência de domínio ou alteração da granularidade da ontologia) que são gerados em resposta a um novo entendimento da conceitualização do mundo [Klein e Fensel, 2001].

As mudanças ocorridas em uma ontologia podem ser divididas em quatro fases, como ilustra a figura seguinte.

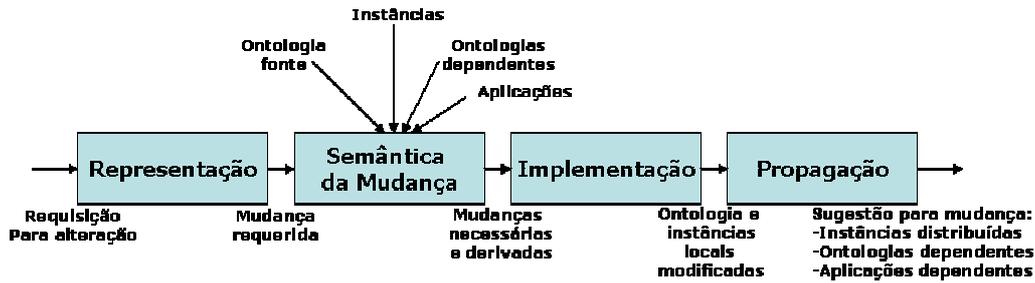


Figura 21 – Fases de uma mudança, traduzido de [Stojanovic *et al.*, 2002]

Uma mudança numa ontologia pode induzir a inconsistências em outras partes da ontologia. Essa inconsistência gerada pode ser tanto semântica quanto sintática. A inconsistência sintática surge quando são usados elementos não definidos na ontologia ou quando as restrições na ontologia são invalidadas. Por exemplo, a remoção de um conceito que é o único elemento do conjunto de domínio de uma dada propriedade resulta em uma inconsistência de sintaxe. A inconsistência semântica surge quando o significado de uma entidade é alterado por causa das mudanças realizadas anteriormente. Quanto maior uma ontologia, mais complexo é compreender completamente a extensão e o significado que cada mudança pode gerar [Stojanovic *et al.*, 2002]. A implementação da mudança acontece na codificação da dada mudança na linguagem formal utilizada. Por fim, essa mudança é propagada para as ontologias ou aplicações dependentes. As instâncias da ontologia necessitam ser alteradas para preservar a consistência ontológica.

As mudanças em uma ontologia acontecem desde a sua criação até a sua utilização, sendo parte de todo o processo evolutivo da ontologia. Essas mudanças fazem com que o processo de evolução da ontologia seja considerado um processo cíclico, onde, a cada novo requisito de negócio ou descoberta de conhecimento, todo o processo precise ser novamente realizado. Uma vez que a validação de uma mudança realizada pode (automaticamente) induzir novas mudanças a fim de obter um modelo consistente ou que satisfaça as expectativas do usuário, o processo de evolução pode ser reiniciado. A Figura 22 ilustra o processo de evolução de uma ontologia.

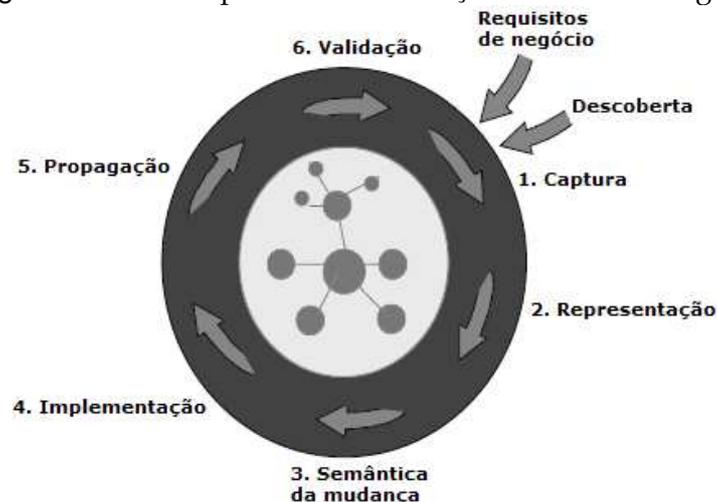


Figura 22 – Processo de evolução de ontologias, traduzido de [Stojanovic *et al.*, 2002]

A variedade de causas e conseqüências nas alterações em uma ontologia faz com que a evolução de ontologia seja uma operação muito complexa que deveria ser considerada não só um processo técnico, mas também organizacional [Staab *et al.*, 2001].

Dos tipos de mudança existentes, as mudanças no domínio e na conceitualização são vistas como alterações de manutenção. A implicação dessas mudanças é discutida na seção 3.6.1 . A seção discute também sobre a mudança na codificação, isto é, a mudança na linguagem de representação da ontologia. A propagação das mudanças para os elementos dependentes da ontologia é vista como parte do problema de versionamento ontológico, discutido na seção 3.6.2 .

3.6.1 Manutenção

A manutenção de ontologias é realizada não só para implementar mudanças oriundas de novos requisitos de negócio, mas também para melhorar a qualidade da ontologia. Segundo Welty [2004], a gerência da qualidade de uma ontologia pode melhorar o desempenho em pesquisas em 18%, por exemplo. Dessa forma, mudanças na ontologia podem surgir para modificar sua riqueza, correteza, organização, consistência, aumentar sua cobertura do domínio ou melhorar a integração com outras ontologias.

Ainda segundo Welty, assim como na vida de um *software*, a fase de manutenção equivale a 80% do custo de uma ontologia. Esse custo pode se tornar o maior problema na utilização das ontologias conforme a ontologia venha ser cada vez mais usada. O custo de manutenção pode ser mais elevado quando uma análise do projeto não é realizada de modo adequado, o que gera retrabalho para consertar erros de conceitualização ou codificação. Esses erros podem gerar inconsistências e imprecisões no resultado de inferências. Ontologias com uma alta qualidade, ou seja, ontologias consistentes, eficientes e facilmente extensíveis são mais difíceis de construir, porém mais fáceis de manter [Welty, 2004].

Para descoberta de erros em ontologias, é recomendável a aplicação de testes. Esses testes podem ser testes de regressão ou de remoção. Pode-se também utilizar um conjunto de meta-tags para teste de conteúdo. Algumas ferramentas podem ser usadas nessa tarefa, como ferramentas para visualização hierárquica de classes, propriedades e relações entre classes, máquinas de inferência e *reasonings* de consistência. Para verificar consistência, podem também ser utilizados os métodos discutidos na seção 3.5 .

Quando a mudança necessária implica na troca de formalismo, é preciso levar em consideração a capacidade de representação de cada linguagem. Por exemplo, uma linguagem pode ser capaz de representar herança múltipla e outra não. Esses problemas de expressividade geram alteração na estrutura ontológica e podem gerar também anomalias semânticas, alterando o significado dos conceitos descritos na nova linguagem. É aconselhável a realização de testes após codificação da ontologia no novo formalismo.

3.6.2 Versionamento

As pesquisas relacionadas à evolução e versionamento de ontologias se focam em como ontologias lidam com as mudanças internas e externas do ambiente. De acordo com Noy e Klein [2004], o versionamento e evolução de ontologias é definido como “a habilidade de manter as mudanças da ontologia e seus efeitos ao criar e manter dife-

rentes variantes da ontologia”. Esta habilidade inclui métodos para distinguir e reconhecer versões, especificações de relacionamento entre versões, procedimentos de atualização e mudanças, além de mecanismos de acesso que combinem versões de uma ontologia com os dados correspondentes.

Na última década, a maioria das atividades de pesquisa na área da engenharia de ontologias se focou na construção de ontologias, enquanto a necessidade de encontrar meios para lidar com a manutenção e mudanças das ontologias tenha sido uma área de estudo praticamente abandonada [Liang, 2005]. As metodologias existentes para evolução de ontologias são ainda imaturas e exigem muitos estudos [Stojanovic, 2004]. Ainda que a evolução de ontologias seja amplamente reconhecida como um requisito para uma aplicação de sucesso, existe muito trabalho de pesquisa nessa área e muitos problemas a serem resolvidos [Stojanovic *et al.*, 2003].

Alguns pesquisadores, contudo, têm proposto várias direções para dar suporte ao versionamento e evolução de ontologias. Uma dessas abordagens é baseada na utilização de arquivos de *logs* de mudanças. Olivier *et al.* [1999] discutem o tipo de mudanças que ocorrem em terminologias médicas e propõem o modelo de conceitos CONCORDIA para lidar com essas mudanças. Eles especificam um conjunto de mudanças e descrevem mecanismos baseados em arquivos de *logs* para sincronizar diferentes versões. A principal característica do CONCORDIA é que todos os conceitos possuem um identificador (ID) permanente e único, o que faz com que seja dado um estado de “aposentado” para os conceitos ao invés de serem fisicamente apagados. Assim, as ligações especiais com estado de “aposentado” são mantidas, sendo possível rastrear os relacionamentos de herança que não estão mais sendo utilizados. Contudo, esta abordagem não poderia ser aplicada na Web Semântica, uma vez que não existe a possibilidade de controlar todo o processo de evolução [Liang, 2005]. Stojanovic *et al.* [2002] introduziu estratégias de evolução no ambiente KAON [Bozsak *et al.*, 2002], um conjunto de ferramentas para gerência de ontologias, as quais permitem aos desenvolvedores especificar efeitos complexos de mudança no ambiente. O KAON utiliza arquivos de *logs* de mudanças para versionamento de informação. Outro trabalho baseado em arquivos de *log* pode ser lido em [Ognyanov e Kiryakov, 2002], onde os autores propõem um método para rastrear mudanças em repositórios RDF.

Enquanto existem sistemas e ferramentas provendo *logs* de mudanças entre várias versões de ontologias para dar suporte ao processo de evolução de ontologias, não há interação ou compartilhamento de informação entre esses sistemas. Ter um *framework* geral para versionamento e evolução de ontologias que permita suportar diferentes tarefas de evolução e compartilhar essa informação de mudança com outras ferramentas e ontologias, irá tornar o processo de evolução muito mais eficiente, conforme é discutido em [Klein e Noy, 2003].

Outra abordagem é baseada na comparação de diferentes variantes ou versões de uma mesma ontologia. As ferramentas Protégé [Noy *et al.*, 2004] e OntoView [Klein *et al.*, 2002] são dois exemplos que consideram essa abordagem. O OntoView é um sistema para versionamento de ontologias que se baseia na comparação de duas versões de uma ontologia a fim de detectar mudanças. A ferramenta realiza uma comparação entre pares dos conjuntos de elementos RDF que formam a versão antiga e a nova versão das definições de classes e propriedades de uma ontologia. Assim, as mudanças na sintaxe e na representação do RDF são ignoradas. Existem ainda muitas formas de transferir uma ontologia para uma nova versão e o OntoView gera uma solução baseada em um conjunto fixo de heurísticas [Ognyanov e Kiryakov, 2002].

A análise do suporte de versionamento e evolução de ontologias no Protégé é feito através do módulo de comparação chamado PROMPT [Noy e Musen, 2003]. O PROMPT compara duas versões de uma mesma ontologia exibindo a hierarquia e destacando as mudanças encontradas. Com essa visão, o engenheiro pode aceitar ou rejeitar as mudanças feitas na ontologia. Além disso, o Protégé permite armazenar cada mudança realizada na ontologia dentro do próprio arquivo de projeto (com extensão .pjrn), permitindo que essas informações possam ser lidas e alguma informação de interesse possa ser posteriormente recuperada.

3.7 Ferramentas de apoio

Segundo [Gómez-Pérez *et al.*, 2002], nesta década surgiu um grande número de ferramentas para ontologias com o objetivo de dar suporte ao processo de desenvolvimento de ontologias (construção, anotação, integração, etc) e ao uso de ontologias em aplicações. A maioria dessas ferramentas é distribuída como *software* livre ou vendida por empresas pequenas recém criadas, como o Protégé [Noy *et al.*, 2001] e o OntoBroker [Decker *et al.*, 1999], respectivamente. Contudo, grandes empresas de *software* começam a incluir no seu portfólio ferramentas que oferecem serviços semânticos através de ontologias, o que é o caso da Oracle [Oracle, 2008] e IBM [IBM, 2004], [IBM, 2006], [IBM, 2008a].

Uma listagem detalhada das ferramentas existentes atualmente, pode ser encontrada em vários trabalhos da literatura, tais como: [Denny, 2002], [Gómez-Pérez *et al.*, 2002], [Welty, 2004], [Cardoso, 2007], [Thibodeau, 2008]. Neste trabalho, apresentaremos uma descrição dos tipos de ferramentas existentes para resumir as características principais de cada grupo de ferramentas.

As ferramentas para manipulação de ontologias podem ser categorizadas nos seguintes tipos:

- Ferramentas para desenvolvimento de ontologias: Este grupo inclui ferramentas, ambientes e suítes que podem ser usadas para construir novas ontologias ou reusar ontologias existentes. Além das funcionalidades comuns de edição e navegação, estas ferramentas também incluem funcionalidades para documentação, exportação e importação de diferentes formatos, diferentes visualizações da ontologia construída, bibliotecas de ontologias, máquinas de inferências, etc. Alguns exemplos de ferramentas para desenvolvimento de ontologias são: Protégé, OILED, OntoStudio, Ontolingua Server, LinkFactory, etc.
- Ferramentas para compatibilização de ontologias: Estas ferramentas tentam resolver o problema de combinar e integrar ontologias. Esta necessidade aparece, por exemplo, quando duas organizações se unem numa só, ou quando é necessário obter uma ontologia de qualidade superior (neste caso, mais abrangente) utilizando ontologias existentes do mesmo domínio. Alguns exemplos de ferramentas para integração e combinação de ontologias são: Chimaera, PROMPT, ODEMerge, etc.
- Ferramentas para avaliação de ontologias: Estas ferramentas são utilizadas como ferramentas de suporte para garantir que duas ontologias possuam um dado nível de qualidade. A garantia de qualidade é extremamente importante pois diminui os problemas durante a integração de ontologia e em aplicações industriais que se utilizam dessas estruturas de conhecimento. Alguns exemplos de ferramentas para avaliação de ontologias são: OntoAnalyser, OntoGenerator, ONE-T, etc.

- Ferramentas para armazenamento e consulta de ontologias: Estas ferramentas têm sido criadas para permitir a utilização e consulta de ontologias de forma mais simples. Algumas dessas ferramentas utilizam ou fazem parte de SGBD's, como é o caso da solução da Oracle [Oracle, 2008]. Outros exemplos de ferramentas para armazenamento e consulta de ontologias são: Cerebra, Sesame, Redland, etc.
- Ferramentas de aprendizagem (ou de construção/preenchimento semi-automático): Estas ferramentas são utilizadas para derivar (semi)automaticamente ontologias de textos em linguagem natural. Esta funcionalidade normalmente é utilizada para facilitar o processo de construção de ontologias, criando-se uma ontologia inicial que será posteriormente revisada manualmente. Alguns exemplos de ferramentas para aprendizagem são: OntoLearn, Text-To-Onto, OntoBuilder, etc.

Geralmente, as ferramentas existentes são altamente especializadas e, assim, se encaixam em somente um dos grupos descritos. Contudo, algumas suítes oferecem funcionalidades que tentam abranger todas ou a maioria dos grupos acima, como é o caso do Protégé [Noy *et al.*, 2001], que, além do editor, possui ferramentas para compatibilização de ontologias, consultas, entre outras, e do KAON, que possui um servidor para acesso a ontologias, ferramentas para edição e visualização, entre outras [Bozsak *et al.*, 2002].

4 Conclusão

Este relatório apresentou as principais questões relativas à gestão de ontologias, abordando temas como metodologias de construção, linguagens para representação de ontologias, linguagens para consultas em ontologias, editores de ontologias, formas de compatibilização, métodos de avaliação de qualidade e questões relativas à evolução das ontologias existentes.

As questões apresentadas neste relatório servem como base para a proposta de Gestão de Ontologias para a GDIEP/BDIEP.

Agradecimentos

Este trabalho não seria possível sem a contribuição de pesquisadores em Sistemas de Informação. Em especial, agradecemos aos professores e alunos que colaboraram nas discussões e desenvolvimento de pesquisas, testes e desenvolvimentos necessários ao projeto. Dentre os agradecimentos à academia, se destaca o papel dos profissionais do NP2Tec⁹ que contribuíram, técnica ou administrativamente, para o sucesso de nossas atividades.

A condução e os resultados deste trabalho são uma exemplar evidência de como a relação entre as universidades e as empresas pode contribuir para a geração de conhecimento útil e, desta forma, contribuir para nossa sociedade.

⁹ Site do NP2Tec: <http://www.uniriotec.br/~np2tec>

Referências Bibliográficas

- BERNARAS, A., LARESGOITI, I., CORERA, J. Building and Reusing Ontologies for Electrical Network Applications. In: EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE, 12., 1996, p. 298-302.
- BLÁZQUEZ, M., FERNÁNDEZ-LÓPEZ, M., GARCÍA-PINAR, J.M., et al. Building Ontologies at the Knowledge Level Using The Ontology Design Environment. In: WORKSHOP ON KNOWLEDGE ACQUISITION, MODELING AND MANAGEMENT, 11., 1998, Banff, Canada.
- BLOMMESTEIN, F.V. REA as an e-business ontology. In: INTERNATIONAL REA TECHNOLOGY WORKSHOP, 2., 2006, Santorini Island, Greece.
- BOOCH, G., RUMBAUGH, J., JACOBSON, I. **The unified modeling language user guide**. Addison-Wesley, 1999.
- BORST, W.N. **Construction of Engineering Ontologies**. 1997. PhD thesis, Centre for Telematica and Information Technology, University of Twente, Enschede, HL.
- BOTELHO, F.C., TORRES, K., BRAGA, J.L. Reengenharia e Visualização de Conceitos no WordNet, Revista Eletrônica de Iniciação Científica da SBC (REIC), v. 2, n. 2, 2002, p. 1-16.
- BOZSAK, E., EHRIG, M., HANDSCHUH, S., et al. KAON - Towards a large scale Semantic Web. In: INTERNATIONAL CONFERENCE ON E-COMMERCE AND WEB TECHNOLOGIES EC-WE, 3., v. 2455, 2002, Aix-en-Provence, France, p. 304-313.
- BRANK, J., GROBELNIK, M., MLADENIC, D. A survey of ontology evaluation techniques. In: INTERNATIONAL MULTICONFERENCE INFORMATION SOCIETY IS, 8., 2005, p. 166-169.
- BRAUNER, D.F., BRANDÃO, A.A.F., CUNHA, L.M., et al. **Um Estudo de Caso para Avaliação do Knowledge Unified Process para o Desenvolvimento de Ontologias**. Rio de Janeiro: PUC-RIO, Departamento de Informática da Pontifícia Universidade Católica, 2003.
- BRICKLEY, D., GUHA, R.V. **RDF Vocabulary Description Language 1.0: RDF Schema**. Relatório, W3C Recommendation, 2004.
- BUNDY, A., MCNEILL, F., SCHORLEMMER, M. Dynamic ontology refinement. In: WORKSHOP ON PLAN EXECUTION (ICAPS), 2003, Trento, Itália.
- BUNNINGEN, A.H.V. **Augmented Trading: Predicting stocks with OpenCyc**. 2004. Dissertação de Mestrado, Department of Computer Science, University of Twente, Enschede, Holanda.
- CAPPELLI, C., BAIÃO, F., SANTORO, F., et al. Uma abordagem de construção de ontologia de domínio a partir do modelo de processos de negócio. In: WORKSHOP ON ONTOLOGIES AND METAMODELING IN SOFTWARE AND DATA ENGINEERING (WOMSDE), 2., 2007, João Pessoa, Brasil.

- CARDOSO, J. The Semantic Web Vision: Where are we? *IEEE Intelligent Systems*, v. 22, n. 5, 2007, p. 84-88.
- CHALUPSKY, H. OntoMorph: A translation system for symbolic logic. In: *INTERNATIONAL CONFERENCE ON PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING*, 7., 2000, San Francisco, CA, USA, p. 471-482.
- CHAUDHRI, V.K., FARQUHAR, A., FIKES, R., *et al.* OKBC: a programmatic foundation for knowledge base interoperability. In: *NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE (AAAI)*, 15., 1998, Madison, Wisconsin, p. 600-607.
- CHEN, P.P., The Entity-Relationship Model-Toward a Unified View of Data, *ACM Transactions on Database Systems*, v. 1, n. 1, 1976, p. 9-36.
- CONNOLLY, D., ARMELEN, F.V., HORROCKS, I., *et al.* DAML+OIL Reference Description, 2001. Disponível em <<http://www.w3.org/TR/daml+oilreference>>. Acesso em: 21 jan. 2008.
- CORCHO, O., FERNÁNDEZ-LÓPEZ, M., GÓMEZ-PÉREZ, A. Methodologies, tools and languages for building ontologies. Where is their meeting point? *Data & Knowledge Engineering*, v. 46, n. 1, 2003, p. 41-64.
- CORCHO, O., GÓMEZ-PÉREZ, A., GUERRERO-RODRÍGUEZ, D.J., *et al.* Evaluation experiment of ontology tools' interoperability with the WebODE ontology engineering workbench. In: *WORKSHOP ON EVALUATION OF ONTOLOGY TOOLS (EON)*, 2003, Sanibel Island, Florida.
- CRISTANI, M., CUEL, R. A survey on ontology creation methodologies, *International Journal on Semantic Web & Information Systems*, v. 1, n. 2, 2005, p. 49-69.
- DAMJANOVIC, V., DEVEDŽIC, V., DJURIC, D., *et al.* Framework for Analyzing Ontology Development Tools, *AIS SIGSEMIS Bulletin*, v. 1, n. 3, 2004, p. 43-47.
- DEAN, M.S., G.; BECHHOFER, S.; HARMELLEN, F. V.; ENDLER, J.; HORROCKS, I.; MCGUINNESS, D. L.; PATEL-SCHNEIDER, P. F.; STEIN, L. A. OWL: Web Ontology Language Reference, 2006. Disponível em <<http://www.w3.org/TR/owl-ref/>>. Acesso em 18 jan. 2006.
- DECKER, S., ERDMANN, M., FENSEL, D., *et al.* Ontobroker: Ontology based access to distributed and semi-structured information. In: MEERSMAN, R., TARI, Z., STEVENS, S. (eds), *DS-8: Semantic Issues in Multimedia Systems*, 1999, Boston, Kluwer Academic Publishers.
- DENNY, M. Ontology Building: A Survey of Editing Tools, 2002. Disponível em <<http://www.xml.com/pub/a/2002/11/06/ontologies.html>>. Acesso em: 14 jun. 2008.
- DIJKMAN, R.M., PIRES, L.F., JOOSTEN, S.M.M. Calculating with concepts: a technique for the development of business process support. In: *UML WORKSHOP ON PRACTICAL UML-BASED RIGOROUS DEVELOPMENT METHODS*, 2001, Toronto, Canada, p. 87-98.

- DING, Y. Ontology: The enabler for the Semantic Web. A review of ontologies with the Semantic Web in view, *Journal of Information Science*, v. 27, n. 6, 2001, p. 377-384.
- DRUMOND, L.R., GIRARDI, R. Uma Análise das Técnicas e Ferramentas para o Desenvolvimento de Aplicações para a Web Semântica, *REIC - Revista eletrônica de iniciação científica*, v. 6, n. 1, 2006.
- EHRIG, M., STAAB, S. QOM - quick ontology mapping. In: *INTERNATIONAL SEMANTIC WEB CONFERENCE (ISWC)*, 3., 2004, Hiroshima, Japão, p. 683-696.
- EUZENAT, J. Corporate Memory through cooperative creation of knowledge based and hyper-documents. In: *KNOWLEDGE ACQUISITION FOR KNOWLEDGE-BASED SYSTEMS WORKSHOP (KAW)*, 10., 1996, Banff, Alberta, Canada, p. 1-20.
- EUZENAT, J., BACH, T.L., BARRASA, J., *et al.* *State of the Art on Ontology Alignment*, Relatório Técnico KWEB, 2004, D2.2.3, v 1.2, Knowledge Web.
- FALBO, R.A., MENEZES, C.S., ROCHA, A.R.C. A Systematic Approach for Building Ontologies. In: *IBEROAMERICAN CONFERENCE ON ARTIFICIAL INTELLIGENCE (IBERAMIA)*, 6., v. 98, 1998, Lisboa, Portugal.
- FALBO, R.D.A., GUIZZARDI, G., DUARTE, K.C. An ontological approach to domain engineering. In: *INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING (SEKE)*, 14., 2002, Ischia, Italy, p. 351-358.
- FELICÍSSIMO, C.H. **Interoperabilidade Semântica na Web: Uma Estratégia para o Alinhamento Taxonômico de Ontologias**. 2004. Dissertação de Mestrado, Programa de Pós-Graduação em Informática do Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil.
- FELICÍSSIMO, C.H., LEITE, J.C.P., BREITMAN, K.K., *et al.* Geração de Ontologias subsidiada pela Engenharia de Requisitos. In: *VI WORKSHOP EM ENGENHARIA DE REQUISITOS (WER)*, 6., 2003, Piracicaba, Brasil.
- FENSEL, D., HORROCKS, I., HARMELEN, F.V., *et al.* OIL: An ontology infrastructure for the semantic Web, *IEEE Intelligent Systems*, 2001.
- FERNÁNDEZ-LÓPEZ, M., GÓMEZ-PÉREZ, A. Overview and Analysis of methodologies for building ontologies, *Knowledge Engineering Review (KER)*, v. 17, n. 2 2002, p. 129-156.
- FERNÁNDEZ, M., GÓMEZ-PÉREZ, A., JURISTO, N. METHONTOLOGY: From Ontological Art Towards Ontological Engineering. In: *SPRING SYMPOSIUM SERIES ON ONTOLOGICAL ENGINEERING*, 1997, Stanford, EUA, p. 33-40.
- FIKES, R., HAYES, P., HORROCKS, I. OWL-QL – a language for deductive query answering on the Semantic Web, *Web Semantics: Science, Services and Agents on the World Wide Web*, v. 2, n. 1, 2004, p. 19-29.

- FREITAS, F., STUCKENSCHIMDT, H., NOY, N. *Ontology Issues and Applications*, Journal of the Brazilian Computer Society, v. 11, n. 2, 2005, p. 5-16.
- GANGEMI, A., CATENACCI, C., CIARAMITA, M., *et al.* **Ontology evaluation and validation: an integrated formal model for the quality diagnostic task**. Relatório, 2005, Laboratory for applied ontology, ISTC-CNR, Roma/Trento, Italy.
- GANGEMI, A., CATENACCI, C., CIARAMITA, M., *et al.* *Modelling Ontology Evaluation and Validation*. In: EUROPEAN SEMANTIC WEB CONFERENCE (ESWC), 3., 2006.
- GANGEMI, A., GUARINO, N., MASOLO, C., *et al.* *Sweetening wordnet with DOLCE*, Artificial Intelligence Magazine, v. 24, n. 3, 2003, p. 13-24.
- GENESERETH, M.R., FIKES, R.E. **Knowledge Interchange Format**, Version 3.0 Reference Manual, Relatório Técnico Logic-92-1, 1992, Computer Science Department, Stanford University.
- GINSBERG, M.L. *Knowledge Interchange Format: the KIF of Death*, AI Magazine, v. 12, n. 3, 1991, p. 57-63.
- GLIMM, B. *SPARQL and OWL: How to Answer Conjunctive Queries with an Expressive Ontology Language*, W3C Semantic Web Interest Group, Oxford University Computing Labs, 2007.
- GÓMEZ-PÉREZ, A. FERNÁNDEZ-LÓPEZ, M., FENSEL, D. **Deliverable 1.3: A survey on ontology tools**. Relatório Técnico IST-2000-29243, 2002, OntoWeb - Ontology-based information exchange for knowledge management and electronic commerce.
- GOMEZ-PEREZ, A. *Evaluation of Taxonomic Knowledge in Ontologies and Knowledge Bases*. In: BANFF KNOWLEDGE ACQUISITION FOR KNOWLEDGE-BASED SYSTEMS WORKSHOP, 12., 1999a, Banff, Alberta, Canada.
- GOMEZ-PEREZ, A. *Ontological engineering: A state of the art*, Expert Update. British Computer Society, v. 2, n. 3, 1999b, p. 33-43.
- GORDIJN, J. **D5.1: E-Business ontology**. Relatório Técnico IST-2001-33144, 2002, Free University Amsterdam, Amsterdam.
- GRUBER, T. **Grande Challenges for Ontology Design (or is it Vente?)**, 2007. Disponível em <http://tomgruber.org/writing/challenges-for-ontology-design.htm>. Acesso em: 14 abr. 2008.
- GRUBER, T.R. *A translation approach to portable ontology specifications*, Knowledge Acquisition, 1993.
- GRUNINGER, M., ATEFI, K., FOX, M.S. *Ontologies to support process integration in enterprise engineering*, Computational and Mathematical Organization Theory, Kluwer Academic Publishers, 2000.
- GRUNINGER, M., KOPENA, J. *Semantic Integration Through Invariations*, Artificial Intelligence Magazine, v. 26, n. 1, 2005, p. 11-20.

- GUARINO, N. Formal Ontology and Information Systems. In: INTERNATIONAL CONFERENCE ON FORMAL ONTOLOGIES IN INFORMATION SYSTEMS, 1., 1998, Trento, Italy, p. 3-15.
- GUARINO, N., PADOVA, I., GIARETTA, P. Ontologies and Knowledge Bases Towards a Terminological Clarification. In: MARS, N. (eds), Towards Very Large Knowledge Bases, Knowledge Building and Knowledge Sharing, 1995, Amsterdam, IOS Press.
- GUARINO, N., WELTY, C. Evaluating ontological decisions with OntoClean, Communications of the ACM, v. 45, n. 2, 2002, p. 61-65.
- GUIZZARDI, G. Ontological Foundations for Structural Conceptual Models, 2005, Centre for Telematics and Information Technology, Telematica Institute, University of Twente, Enschede, Holanda.
- HAARSLEV, V., MÖLLER, R. Racer: A Core Inference Engine for the Semantic Web. In: INTERNATIONAL WORKSHOP ON EVALUATION OF ONTOLOGY-BASED TOOLS, 2., 2003, Sanibel Island, Florida, USA, pp. 27-36.
- HAARSLEV, V., MÖLLER, R., WESSEL, M. Querying the Semantic Web with Racer+nRQL. In: WORKSHOP ON APPLICATIONS OF DESCRIPTION LOGICS, WORKSHOP ON APPLICATIONS OF DESCRIPTION LOGICS (ADL), 2004, Ulm, Germany.
- HAASE, P., BROEKSTRA, J., EBERHART, A., *et al.* A Comparison of RDF Query Languages. In: INTERNATIONAL SEMANTIC WEB CONFERENCE, 3., v. 3298, 2004, Hiroshima, Japan, p. 502-517.
- HAASE, P., SURE, Y. **Incremental Ontology Evolution - Evaluation**. Relatório, University Of Karlsruhe, Karlsruhe, 2005.
- HEFLIN, J., HENDLER, J., LUKE, S. **SHOE: A Knowledge Representation Language for Internet Applications**. Relatório Técnico CS-TR-4078, 1999.
- HORROCKS, I., PATEL-SCHNEIDER, P.F., VAN HARMELEN, F. From SHIQ and RDF to OWL: the making of a Web Ontology Language, Journal of Web Semantics: Science, Services and Agents on the World Wide Web, v. 1, n. 1, 2003, p. 7-26.
- IBM, **IBM Integrated Ontology Development Toolkit**, 2004. Disponível em <<http://www.alphaworks.ibm.com/tech/semanticstk>>. Acesso em: 05 jun. 2008.
- IBM, **IBM Web Ontology Manager**, 2006. Disponível em <http://www.alphaworks.ibm.com/tech/wom?open&S_TACT=105AGX59&S_CMP=GR&ca=dgr-lnxwd01awwom>. Acesso em: 05 jun. 2008.
- IBM, **IBM Semantic Layered Research Platform**, 2008a. Disponível em <<http://ibm-slrp.sourceforge.net/>>. Acesso em: 05 jun. 2008.

- IBM, **What is ontology?**, 2008b. Disponível em <http://www.alphaworks.ibm.com/contentnr/semanticsfaqs?open>. Acesso em: 19 jun. 2008.
- IEEE, **Guide for Software Quality Assurance Planning**, Relatório Técnico Std. 730.1-1995, IEEE Computer Society, New York.
- JAKONIENE, V. **Linköpings Universitet's Course on Logics for the Web: Ontology Integration**, 2006. Disponível em <http://www.ida.liu.se/labs/iislab/courses/LW/slides/ontologyIntegration.pdf>. Acesso em: 08 jul. 2006.
- KARP, P.D., CHAUDHRI, V.K., THOMERE, J. **XOL: An XML-based ontology exchange language - Version 0.4**, Relatório, 1999.
- KIM, H.M., FOX, M.S., GRUNINGER, M. An ontology for quality management - Enabling quality problem identification and tracing, *BT Technology Journal*, v. 17, n. 4, 1999, p. 131-140.
- KLEIN, M. Combining and Relating Ontologies: An Analysis of Problems and Solutions. In: WORKSHOP ON ONTOLOGIES AND INFORMATION SHARING AT THE 17TH INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 2001, Seattle, USA, p. 53-62.
- KLEIN, M., FENSEL, D. Ontology Versioning on the Semantic Web. In: INTERNATIONAL SEMANTIC WEB WORKING SYMPOSIUM (SWWS), 2001, Stanford University, California, USA, p. 75-91.
- KLEIN, M., FENSEL, D., KIRYAKOV, A., *et al.* Ontology versioning and change detection on the web, In: INTERNATIONAL CONFERENCE ON KNOWLEDGE ENGINEERING AND KNOWLEDGE MANAGEMENT (EKAW02), 13., 2002, p. 197-212.
- KLEIN, M., NOY, N.F. **A Component-Based Framework For Ontology Evolution**, Relatório Técnico IR-504, 2003, Department of Computer Science, Vrije Universiteit, Amsterdam.
- LASSILA, O., SWICK, R. **Resource description framework (RDF): model and syntax specification**, Relatório, W3C recommendation, 1999.
- LENAT, D.B., GUHA, R.V. **Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project**. Boston: Addison-Wesley, 1990.
- LIANG, Y. **Ontology Versioning and Evolution for Semantic Web-Based Applications**, report submitted as per the requirements for the continuation of study for the degree of MPhil/PhD, 2005, School of Electronics and Computer Science, University of Southampton, Southampton, UK.
- LOOM, **Loom Project Home Page**, 2006. Disponível em <http://www.isi.edu/isd/LOOM/LOOM-HOME.html>. Acesso em: 09 jun. 2006.

- LÓPEZ-PÉREZ, A. **Integración de la aplicación OntoMetric en la plataforma WebODE**. 2004. Proyecto fin de carrera, Facultad de Informática, Universidad Politécnica de Madrid, Madrid.
- LOZANO-TELLO, A. **Métrica de idoneidad de ontologías**. 2002. Tesis doctoral, Universidad de Extremadura, Extremadura.
- MAEDCHE, A., STAAB, S. Measuring similarity between ontologies. In: EUROPEAN CONFERENCE ON KNOWLEDGE ACQUISITION AND MANAGEMENT (EKAW), 2002, p. 251-263.
- MAGKANARAKI, A., KARVOUNARAKIS, G., ANH, T.T., *et al.* **Ontology Storage and Querying**. Relatório Técnico 308, 2002, Information Systems Laboratory, Foundation for Research and Technology Hellas.
- MCGUINNESS, D.L. Ontologies Come of Age. In: FENSEL, D., HENDLER, J., LIEBERMAN, H., *et al.* (eds), *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, 2005, MIT Press.
- MCGUINNESS, D.L., FIKES, R., RICE, J., *et al.* An environment for merging and testing large ontologies. In: INTERNATIONAL CONFERENCE ON PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING (KR2000), 7., 2003, Breckenridge, Colorado, USA, p. 483-493.
- MCNEILL, F., BUNDY, A., WALTON, C. Diagnosing and Repairing Ontological Mismatches. Starting AI Researchers' Symposium, Valencia, 2004.
- MCNEILL, F., BUNDY, L., WALTON, C. Planning from rich ontologies through translation between representations. In: ICAPS WORKSHOP ON THE ROLE OF ONTOLOGIES ON PLANNING AND SCHEDULING, Monterey, CA, 2005.
- MELNIK, S., GARCIA-MOLINA, H., RAHM, E. Similarity Flooding: A versatile graph matching algorithm and its application to schema matching. In: INTERNATIONAL CONFERENCE ON DATA ENGINEERING (ICDE), San Jose, CA, USA, 2002, p. 117-128.
- MITRA, P., WIEDERHOLD, G., KERSTEN, M. A Graph-Oriented Model for Articulation of Ontology Interdependencies. In: CONFERENCE ON EXTENDING DATABASE TECHNOLOGY (EDBT), v. 1777, Konstanz, Alemanha, 2000, p. 86-100.
- NECHES, R., FIKES, R., FININ, T.W., *et al.* Enabling Technology for Knowledge Sharing, AI Magazine, v. 12, n. 3, 1991, p. 36-56.
- NILES, I., PEASE, A. Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology. In: INTERNATIONAL CONFERENCE ON INFORMATION AND KNOWLEDGE ENGINEERING (IKE), Las Vegas, Nevada, 2003.
- NOY, N., MUSEN, M.A. Anchor-PROMPT: Using Non-Local Context for Semantic Matching. In: WORKSHOP ON ONTOLOGIES AND INFORMATION SHARING AT THE INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 17., v. 1, 2001, Seattle, EUA, p. 63-70.

- NOY, N., MUSEN, M.A. The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping, *International Journal of Human-Computer Studies*, v. 59, n. 6 2003, p. 983-1024.
- NOY, N.F., KLEIN, M. Ontology Evolution: Not the Same as Schema Evolution, *Knowledge and Information Systems*, v. 6, n. 4, 2004, p. 428-440.
- NOY, N.F., KUNNATUR, S., KLEIN, M., *et al*, Tracking changes during ontology evolution. In: INTERNATIONAL CONFERENCE ON THE SEMANTIC WEB (ISWC), 3., Hiroshima, Japão, 2004.
- NOY, N.F., MCGUINNESS, D.L. **Ontology development 101: A guide to creating your first ontology**, Relatório Técnico KSL-01-05, 2001, Stanford Knowledge Systems Laboratory, Stanford.
- NOY, N.F., MUSEN, M.A. SMART: Automated Support for Ontology Merging and Alignment. In: WORKSHOP ON KNOWLEDGE ACQUISITION, MODELING AND MANAGEMENT, 12., v. 4, 1999, Banff, Canadá, p. 1-20.
- NOY, N.F., SINTEK, M., DECKER, S., *et al*. Creating Semantic Web contents with Protege-2000, *Intelligent Systems, IEEE*, v. 16, n. 2, 2001, p. 60-71.
- OGNYANOV, D., KIRYAKOV, A., Tracking changes in RDF(S) repositories. In: INTERNATIONAL CONFERENCE ON KNOWLEDGE ENGINEERING AND MANAGEMENT, 13., Espanha, 2002.
- OLIVER, D.E., SHAHAR, Y., SHORTLIFFE, E.H., *et al*. Representation of change in controlled medical terminologies, *Artificial Intelligence In Medicine*, v. 15, n. 1, 1999, p. 53-76.
- OMG, **UML Version 2.1**, 2003. Disponível em <<http://www.omg.org/cgi-bin/doc?formal/07-11-01>>. Acesso em: 12 mar. 2003.
- ORACLE, **Semantic Technologies Center**, 2008. Disponível em <http://www.oracle.com/technology/tech/semantic_technologies/index.html>, acessado em junho de 2008.
- ORLEAN, D. **Um Processo Unificado para Engenharia de Ontologias**. 2003. Dissertação de Mestrado, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro.
- PALAZZO, L.A.M. **Introdução à programação Prolog**, 1 ed., Pelotas: Editora da Universidade Católica de Pelotas, 1997.
- PINTO, S.H., GÓMEZ-PÉREZ, A., MARTINS, J.P. Some Issues on Ontology Integration. In: WORKSHOP ON ONTOLOGIES AND PROBLEM SOLVING METHODS: LESSONS LEARNED AND FUTURE TRENDS (IJCAI), v. 18, 1999, p. 7-12, Stockholm, Suécia.
- PREDIOU, L., FEIER, C., SCHARFFE, F., *et al*. **State-of-the-art Survey on Ontology Merging and Aligning V2**, Relatório Técnico D4.2.1 (WP4), 2006, IST-2003-506826, SEKT Project.

- PRUD'HOMMEAUX, E., SEABORNE, A. **SPARQL Query Language for RDF**, 2004. Disponível em <<http://www.w3.org/TR/rdf-sparql-query/>>. Acesso em: 01 abr. 2008.
- RACER, RacerPro User's Guide Version 1.9, 2005. Disponível em <<http://www.racer-systems.com/dl.php?file=PDF%252Fusers-guide-1-9.pdf&typ=file&name=RacerPro-User-Guide.pdf>>. Acesso em: 05 mai. 2008.
- RATIONAL, **Rational Unified Process: Best Practices for Software development Teams**, Relatório Técnico TP026B, 1998, Rev 11/01, Rational Team.
- RESNIK, P., ELKISS, A., LAU, E., *et al.* The Web in Theoretical Linguistics Research: Two Case Studies Using the Linguist's Search Engine. In: MEETING OF THE BERKELEY LINGUISTICS SOCIETY, 31., Berkeley, Califórnia, USA, 2005.
- RODRÍGUEZ, A., EGENHOFER, M. Determining Semantic Similarity Among Entity Classes from Different Ontologies, *IEEE Transactions on Knowledge and Data Engineering*, v. 15, n. 2, 2003, p. 442-456.
- RUMBAUGH, J.R., BLAHA, M.R., LORENSEN, W., *et al.* **Object-oriented modeling and design**. Upper Saddle River: Prentice-Hall, 1991.
- RUSSELL, S.J., NORVIG, P. **Artificial intelligence: a modern approach**. Upper Saddle River: Prentice-Hall, 1995.
- SAATY, T.L. A Scaling Method for Priorities in Hierarchical Structures, *Journal of Mathematical Psychology*, v. 15, 1964, p. 234-281.
- SIRIN, E., PARSIA, B. SPARQL-DL: SPARQL Query for OWL-DL. In: OWL EXPERIENCES AND DIRECTIONS WORKSHOP (OWLED), 3., Innsbruck, Austria, 2007.
- SMITH, M.K., WELTY, C., MCGUINNESS, D.L. **OWL Web Ontology Language Guide**. Relatório, W3C Recommendation, 2004.
- SPIVEY, J.M., **Understanding Z: A Specification Language and Its Formal Semantics**. New York: Cambridge University Press, 1988.
- STAAB, S., STUDER, R., SCHNURR, H.P., *et al.* Knowledge Processes and Ontologies, Special Issue on Knowledge Management, *IEEE Intelligent Systems*, v. 16, n. 1, 2001.
- STOJANOVIC, L. **Methods and Tools for Ontology Evolution**. 2004. PhD Thesis, Fakultät fuer Wirtschaftswissenschaften, University of Karlsruhe, Karlsruhe, Germany.
- STOJANOVIC, L., MAEDCHE, A., MOTIK, B., *et al.* User-driven ontology evolution management. In: EUROPEAN CONFERENCE ON KNOWLEDGE ENGINEERING AND KNOWLEDGE MANAGEMENT, ONTOLOGIES AD THE SEMANTIC WEB, 13., 2002, p. 285-300.
- STOJANOVIC, L., MAEDCHE, A., STOJANOVIC, N., *et al.* Ontology evolution as re-configuration-design problem solving. In: INTERNATIONAL CONFERENCE

- ON KNOWLEDGE CAPTURE, 2., Sanibel Island, Florida, U.S.A., 2003, p. 162-171.
- STUDER, R., BENJAMINS, V.R., FENSEL, D. Knowledge engineering: Principles and methods, *Data & Knowledge Engineering*, v. 25, n. 1-2, 1998, p. 161-197.
- SUOWG, **Standard Upper Ontology Working Group**, 2006. Disponível em <<http://suo.ieee.org/>>. Acesso em: 11 set. 2006.
- SURE, Y. **Methodology, tools and case studies for ontology based knowledge management**. 2003. Phd Thesis, Department of Economics and Business Engineering, University of Karlsruhe, Karlsruhe.
- SWARTOUT, B., PATIL, R., KNIGHT, K., *et al.* Toward Distributed Use of Large-Scale Ontologies. In: *KNOWLEDGE ACQUISITION FOR KNOWLEDGE-BASED SYSTEMS WORKSHOP*, 10., Alberta, Canada, 1996.
- TEMPICH, C., PINTO, S., STAAB, S., *et al.* A case study in supporting distributed, loosely-controlled and evolving engineering of ontologies (DILIGENT). In: *INTERNATIONAL CONFERENCE ON KNOWLEDGE MANAGEMENT (I-KNOW)*, 4., 2004, Graz, Austria.
- THIBODEAU, T. **Semantic Web Development Tools**, 2008. Disponível em <<http://esw.w3.org/topic/SemanticWebTools>>. Acesso em: 12 jun. 2008.
- USCHOLD, M., GRUNINGER, M. Ontologies: Principles, methods and applications, *Knowledge Engineering Review (KER)*, v. 11, n. 2, 1996, p. 93-155.
- USCHOLD, M., JASPER, R. A Framework for Understanding and Classifying Ontology Applications. In: *WORKSHOP ON ONTOLOGIES AND PROBLEM-SOLVING METHODS*, Stockholm, Sweden, 1999.
- USCHOLD, M., KING, M. Towards a Methodology for Building Ontologies. In: *INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE*, Montréal, Québec, Canada, 1995.
- VISSER, P.R.S., JONES, D.M., BENCH-CAPON, T.J.M., *et al.* An analysis of ontological mismatches: Heterogeneity versus interoperability. In: *SPRING SYMPOSIUM ON ONTOLOGICAL ENGINEERING*, Califórnia, EUA, 1997, p. 164-172.
- WELTY, C. Ontology Maintenance Support: Text, Tools, and Theories. In: *INTERNATIONAL PROTÉGÉ CONFERENCE*, 7., Bethesda, MD, 2004.
- WIEDERHOLD, G. An algebra for ontology composition. In: *MONTEREY WORKSHOP ON FORMAL METHODS*, v. 5661, 1994, Monterey, CA, p. 56-61.
- YORK SURE; STUDER, R. **On-To-Knowledge Methodology**, Relatório Técnico EU IST-1999-10132, 2002, Institute AIFB, University of Karlsruhe, Karlsruhe.