

OntoGuide-4WS: Uma Abordagem Sistemática de Descrição de Serviços Web para fins de Descoberta Dinâmica

Linair Maria Campos

Universidade Federal do Rio de Janeiro
Instituto de Matemática
Núcleo de Computação Eletrônica
Mestrado

Maria Luiza Machado Campos, PhD
Paulo de Figueiredo Pires, DSc

Rio de Janeiro
2004

FICHA CATALOGRÁFICA

C1980 Campos, Linair Maria.

OntoGuide-4WS : uma abordagem sistemática de descrição de serviços web para fins de descoberta dinâmica. - - Rio de Janeiro : UFRJ / IM / NCE, 2004.

xii, 178 p.

Dissertação (mestrado) - - Universidade Federal do Rio de Janeiro. Instituto de Matemática, Núcleo de Computação Eletrônica, 2004.

Orientadores : Maria Luiza Machado Campos e Paulo de Figueiredo Pires

1. Serviços web. 2. Descoberta dinâmica. 3. Ontologias.

I. Título. II. UFRJ

CDD. 004.6

Aos meus pais, pelo exemplo do entusiasmo em aprender e a determinação de enfrentar dificuldades de forma positiva. Ao Nilson, que sempre me apóia e incentiva em todas as minhas iniciativas, e ao Pedro, meu filho, que torna mais alegres os meus dias.

AGRADECIMENTOS

Esta dissertação não seria possível sem a ajuda de alguns, e não seria conduzida de forma tão prazerosa sem a ajuda de outros. Como são muitas as pessoas que tenho a agradecer, escolho alguns apenas, deixando claro no entanto, que de todos guardo o reconhecimento de suas contribuições.

Agradeço aos meus colegas de turma de mestrado, em especial ao Edmundo, pela companhia agradável e sincera, e também pelos muitos cafés que tomamos juntos, à Cristina pelo bom humor e à Glenda pela companhia nos congressos.

Também agradeço ao Pablo Mendes, pelas inspiradas discussões sobre ontologias e ao Serra pela oportunidade de escrevermos artigos em dupla.

Em especial, agradeço à paciência, dedicação e carinho dos meus orientadores, Paulo e Maria Luiza, que me ajudaram muito não só na condução da escrita desse trabalho, como também na motivação a pesquisar, aprender e partilhar novos conhecimentos.

Agradeço também ao Alberto Davila, que me proporcionou as noções básicas de Bioinformática que utilizei neste trabalho, e que de forma amigável e sempre prestativo e atencioso, me motivou a ir mais longe em meus estudos nessa área.

Agradeço ainda à Ligia, que me ajudou a apresentar e a organizar minha pesquisa e que sempre respondeu meus e-mails, fazendo-me sentir muito importante pelo carinho e atenção das suas respostas.

Por fim, para não me estender demais, agradeço à Maria Luiza Almeida, pelas muitas dúvidas que me tirou e pela inspiração que foi para mim a sua tese.

Por tudo, considero que foi para mim um privilégio concluir este trabalho nesta instituição.

Um cronópio pequenininho procurava a chave da porta da rua na mesa de cabeceira, a mesa de cabeceira no quarto de dormir, o quarto de dormir na casa, a casa na rua. Por aqui parava o cronópio, pois para sair à rua precisava da chave da porta.

Julio Cortazar.

RESUMO

CAMPOS, Linair Maria. **OntoGuide-4WS: Uma Abordagem Sistemática de Descrição de Serviços Web para fins de Descoberta Dinâmica**. Orientadores: Maria Luiza Machado Campos e Paulo de Figueiredo Pires. Rio de Janeiro: UFRJ/IM, 2004. Dissertação (Mestrado em Informática).

Com a crescente adoção da tecnologia de serviços Web, a questão da descoberta dinâmica de serviços se torna uma questão relevante, considerando a grande variedade de serviços que podem ser fornecidos. Vários trabalhos têm sido propostos visando estender o poder semântico de descrição de serviços Web com o objetivo de viabilizar a automatização do processo de descoberta. Esses trabalhos baseiam-se em descrições genéricas a serem estendidas para cada domínio de aplicação, sem que seja fornecida, no entanto, uma orientação adequada de como esta extensão deve ser feita. O objetivo do presente trabalho é apresentar uma abordagem sistemática para elaboração ou extensão de descritores genéricos de serviços Web, partindo-se da aplicação de um conjunto de diretrizes de descrição. Para avaliar a utilidade dessa abordagem, propusemos uma ferramenta de software baseada nas diretrizes de descrição propostas e conduzimos um experimento para estender uma ontologia de serviços Web para a área de Bioinformática.

ABSTRACT

CAMPOS, Linair Maria. **OntoGuide-4WS: Uma Abordagem Sistemática de Descrição de Serviços Web para fins de Descoberta Dinâmica**. Orientadores: Maria Luiza Machado Campos e Paulo de Figueiredo Pires. Rio de Janeiro: UFRJ/IM, 2004. Dissertação (Mestrado em Informática).

As Web services utilization grows, Web services dynamic discovery become a relevant issue, considering the multitude of service offerings that can be provided. Several works have been conducted aiming to improve the semantics of web Services description in order to provide automatic web Services discovery. Those works are based on generic descriptions to be extended according to each application domain. However, they do not provide adequate guidance on how the extension should be done. The goal of this dissertation is to present a systematic approach to create or extend generic Web services descriptors, using a set of description guidelines. To demonstrate the guidelines utilization we have proposed a software tool based on them, and also have conducted an experiment to extend a Web services ontology to the Bioinformatics area.

Índice

| | |
|--|----|
| CAPÍTULO 1 | 13 |
| 1. Introdução..... | 13 |
| 1.1. Caracterização do Problema | 14 |
| 1.2. Hipótese | 16 |
| 1.3. Objetivos da Dissertação | 17 |
| 1.4. Enfoque de Solução | 18 |
| 1.5. Organização | 19 |
| | |
| CAPÍTULO 2 | 21 |
| 2. Descrição, Publicação e Descoberta de Serviços Web..... | 21 |
| 2.1. Arquitetura de Serviços Web..... | 22 |
| 2.1.1. Padrão de Descrição de Serviços Web | 23 |
| 2.1.2. Padrão de Publicação de Serviços Web..... | 25 |
| 2.2. Descoberta Dinâmica de Serviços Web..... | 27 |
| 2.3. Características dos Serviços Web..... | 29 |
| 2.3.1. Atributos Funcionais dos Serviços Web..... | 30 |
| 2.3.2. Atributos Não-Funcionais dos Serviços Web..... | 31 |
| 2.3.3. Características Levantadas a partir da Análise de Serviços Web Existentes . | 34 |
| 2.4. Sistematização das Características dos Serviços Web | 40 |
| 2.5. Limitação dos Padrões de Descrição e Publicação de Serviços Web..... | 44 |
| | |
| CAPÍTULO 3 | 46 |
| 3. Iniciativas Relacionadas à Descrição | 46 |
| 3.1. Iniciativas de Descrição Baseadas em Dialectos XML..... | 47 |
| 3.1.1. WSEL | 47 |
| 3.1.2. WSOL..... | 48 |
| 3.1.3. Restrições das Iniciativas de Descrição Baseadas em Dialectos XML..... | 50 |
| 3.2. Iniciativas de Descrição Baseadas em Ontologias | 51 |
| 3.2.1. DReggie | 53 |
| 3.2.2. OWL-S | 54 |
| 3.2.3. Restrições das Iniciativas de Descrição Baseadas em Ontologias | 55 |
| 3.3. Tabela Comparação entre Iniciativas de Descrição de Serviços Web..... | 55 |
| 3.4. Abordagens Teóricas de Apoio à Descrição..... | 57 |
| 3.4.1. Organização de Domínios de Conhecimento | 58 |
| 3.4.2. Organização de Descritores | 59 |
| 3.4.3. Definição de Descritores | 62 |
| 3.4.4. Outras Questões relativas à Semântica de Ontologias..... | 62 |
| 3.4.5. Descrição do Propósito do Serviço Web | 63 |

| | |
|--|---------|
| CAPÍTULO 4 | 65 |
| 4. Diretrizes de Descrição de Serviços Web..... | 65 |
| 4.1. Diretrizes para a Descrição de Serviços Web..... | 66 |
| 4.1.1. Etapa 1 - Definição do Escopo e Modo de Organizar os Domínios..... | 69 |
| 4.1.1.1. Determinação dos Domínios e Sub-Domínios de Conhecimento | 69 |
| 4.1.1.2. Determinação dos Vocabulários a serem Reutilizados..... | 70 |
| 4.1.1.3. Determinação do Método de Raciocínio | 71 |
| 4.1.1.4. Determinação das Categorias Fundamentais e Relações Básicas | 71 |
| 4.1.1.5. Tratamento dos Vocabulários a serem Reutilizados..... | 72 |
| 4.1.2. Etapa 2 - Definição das UC e suas Relações | 73 |
| 4.1.2.1. Definição das UC..... | 73 |
| 4.1.2.2. Determinação do Nível de Compromisso Ontológico das UC..... | 76 |
| 4.1.2.3. Estabelecimento das Relações entre as UC | 78 |
| 4.1.3. Etapa 3 - Detalhamento das UC | 80 |
| 4.1.3.1. Aspectos da Não Ambigüidade do Contexto dos Descritores | 80 |
| 4.1.3.2. Aspectos do Escopo de Uso das Operações | 81 |
| 4.1.3.3. Detalhamento da Associação de Parâmetros | 82 |
| 4.1.3.4. Definição do Propósito das Operações..... | 83 |
| 4.1.3.5. Detalhamento da Categorização do Uso..... | 86 |
| 4.1.4. Etapa 4 - Revisão das Definições | 87 |
| 4.2. Documentação das Diretrizes de Descrição de Serviços Web | 87 |
| CAPÍTULO 5 | 88 |
| 5. A Ferramenta OntoGuide-4WS..... | 88 |
| 5.1. Descrição do Uso da Ferramenta OntoGuide-4WS..... | 89 |
| 5.1.1. Fase 1: Configuração da Ferramenta | 90 |
| 5.1.2. Fase 2: Tratamento das Ontologias | 93 |
| 5.1.3. Fase 3: Extensão da Ontologia de Aplicação | 95 |
| 5.1.4. Fase 4: Geração da Ontologia Estendida em OWL..... | 100 |
| 5.1.5. Fase 5: Geração da Documentação da Ontologia Estendida em XML | 101 |
| 5.1.6. Observações Sobre o uso da Ferramenta OntoGuide-4WS..... | 101 |
| 5.2. Exemplo de Busca por Serviços de Bioinformática | 103 |
| 5.3. Aspectos de Implementação da Ferramenta de Busca..... | 110 |
| 5.4. Descrição do Diagrama de Classes da Ferramenta OntoGuide-4WS..... | 111 |
| 5.5. Aspectos de Implementação da Ferramenta OntoGuide-4WS | 112 |
| CAPÍTULO 6 | 113 |
| 6. Conclusão | 113 |
| 6.1. Contribuições..... | 114 |
| 6.2. Limitações e Dificuldades Encontradas | 116 |
| 6.3. Trabalhos Futuros..... | 118 |

Referências Bibliográficas 121

Anexos 137

| | |
|---|-----|
| A-1 – Exemplo de Documento WSDL..... | 138 |
| A-2 – Exemplo de Documento WSOL..... | 139 |
| A-3 – Ontologia DReggie..... | 140 |
| A-4 – Exemplo de Serviço Web Descrito em OWL-S..... | 144 |

Apêndices 148

| | |
|--|-----|
| B-1 – Documentação OntoGuide | 149 |
| B-1.1 Elementos de Documentação Oriundos da Etapa 1..... | 149 |
| B-1.2 Elementos de Documentação Oriundos da Etapa 2..... | 150 |
| B-1.2 Elementos de Documentação Oriundos da Etapa 3..... | 150 |
| B-2 – Diagramas de Colaboração OntoGuide | 154 |
| B-3 – Fonte da Ontologia de Aplicação em OWL | 157 |
| B-3 – Ontologia de Aplicação na Ferramenta Protégé..... | 173 |

Índice de Figuras

| | |
|--|-----|
| Figura 1 - SOA – Arquitetura Orientada a Serviços | 22 |
| Figura 2 – Ontologias de acordo com o nível de generalidade. | 59 |
| Figura 3 - Diagrama de Atividades das Diretrizes | 59 |
| Figura 4 - Diagrama de Classes (exemplo) para serviços Web de Bioinformática | 69 |
| Figura 5 - Diagrama de Classes da Ferramenta OntoGuide-4WS | 111 |
| Figura 6 – Diagrama de Classes dos Domínios Utilizados | 90 |
| Figura 7 – Escolhendo Ontologias na Ferramenta OntoGuide-4WS | 91 |
| Figura 8 - Configurando Categorias Fundamentais na Ferramenta OntoGuide-4WS .. | 92 |
| Figura 9 - Configurando Relações Básicas na Ferramenta OntoGuide-4WS | 93 |
| Figura 10 - Tratando Categorias Fundamentais de Ontologias em OntoGuide-4WS ... | 94 |
| Figura 11 - Tratando Relações Básicas de Ontologias em OntoGuide-4WS | 95 |
| Figura 12 - Definindo Conceitos na Ferramenta OntoGuide-4WS | 96 |
| Figura 13 – Definindo Relações em OntoGuide-4WS | 98 |
| Figura 14 – Revendo a Definição de Termos em OntoGuide-4WS | 99 |
| Figura 15 - Revendo Relações na Ferramenta OntoGuide-4WS | 100 |
| Figura 17 - Diagrama de Colaboração da Etapa 1 de OntoGuide-4WS | 154 |
| Figura 18 - Diagrama de Colaboração da Etapa 2 de OntoGuide-4WS | 155 |
| Figura 19 - Diagrama de Colaboração da Etapa 3 de OntoGuide-4WS | 156 |
| Figura 20 – Ontologia de Aplicação de Serviços Web – Telas Protégé | 178 |

Índice de Tabelas

| | |
|--|-----|
| Tabela 1 - características dos serviços de tempo pesquisados em XMethods.com. | 36 |
| Tabela 2 - Principais aspectos dos níveis de especificação de Fettke. | 42 |
| Tabela 3 - Aspectos de descrição dos serviços Web, para descrição de atributos. | 43 |
| Tabela 4 - Aspectos de descrição dos serviços Web, para a organização de atributos . | 44 |
| Tabela 5 - Comparação das iniciativas de descrição de serviços Web. | 56 |
| Tabela 6 – Etapas X Aspectos de Descrição Cobertos | 67 |
| Tabela 7 - Exemplos de Definição de UC | 75 |
| Tabela 8 - Práticas Recomendadas para Relações de Serviços Web | 78 |
| Tabela 9 - Serviços de Bioinformática utilizados para análise filogenética. | 105 |
| Tabela 10 - Conceitos usados para definir o propósito dos serviços. | 106 |

CAPÍTULO 1

Introdução

Nos últimos anos, com a difusão do uso da Internet, as empresas vêm gradualmente utilizando esse meio como forma de divulgar e expandir seus negócios. Em um primeiro momento do comércio eletrônico o objetivo da empresa é interagir com seus clientes e parceiros de negócio através de uma interface, onde o diálogo ocorre entre essa interface e uma pessoa. Um bom exemplo é o sítio de comércio eletrônico da Amazon (2003), que lida com clientes e fornecedores. Com a evolução da tecnologia da Internet, o amadurecimento das relações de negócio e o crescimento da competição, novas necessidades surgem, dando origem a transações comerciais onde o foco das interações não está em pessoas. As interações entre parceiros de negócio se dão de forma automatizada, entre programas de computador, e envolve questões como a descoberta de novos parceiros e a integração de dados entre eles (GISOLFI, 2001). Com base nisso, algumas soluções têm sido exploradas, notadamente através de modelos de objetos distribuídos.

Os modelos de objetos distribuídos tais como CORBA (1995), DCOM (1995) e EJB (2001), definem interfaces de comunicação para objetos que implicam em um acordo em relação à infra-estrutura tecnológica, pois seus protocolos de comunicação e modelos de interação não são compatíveis entre si. Essa dependência tecnológica, somada ao fato da complexidade desses modelos, faz com que essa solução não seja a ideal para a integração de dados entre plataformas distintas. Para resolver esse problema, tem-se buscado a adesão a protocolos e padrões que facilitem essa integração e que permitam um novo modelo de software, mais flexível.

Com o surgimento do padrão XML (2003), a melhoria dos meios de transmissão de dados, permitindo o tráfego de maiores volumes de dados característicos dos documentos XML, e a adesão de grandes fornecedores de software tais como IBM, Microsoft, Hewlett-Packard e SUN (ROMMEL, 2001), abriu-se o caminho para um

novo modelo de software: o software como serviço, interoperável, independente de plataforma e linguagens, disponível via Web, utilizando protocolos XML: o serviço Web.

Embora não haja uma única definição, os serviços Web são assim descritos pelo W3C (WSDEF, 2002):

“Um serviço Web é uma aplicação de software identificada por URI, cujas interfaces e ligações são capazes de serem definidas, descritas e descobertas através de artefatos em XML e que suportam interações diretas com outras aplicações de software utilizando mensagens baseadas em XML via protocolos baseados na internet.”

A grande vantagem do uso dos serviços Web está na sua interoperabilidade, obtida devido à adesão a protocolos padrão, amplamente difundidos na Internet, e à existência de uma arquitetura padrão, que define os mecanismos necessários para a sua utilização. Esses atributos vão justamente ao encontro das novas necessidades do comércio eletrônico atual. Dessa forma, é natural esperar que os serviços Web se multipliquem, devido à relativa facilidade de seu desenvolvimento e de suas características de interoperabilidade. De fato, hoje já há no mercado uma tendência do uso crescente dos serviços Web como forma de disponibilizar aplicações de software na Web (IBM WSAT, 2000) (DERTOUZOS, 2002) (TIDWELL, 2000) (TEHAN, 2002). Entretanto, alguns aspectos dessa tecnologia ainda precisam ser amadurecidos para atender esta demanda crescente, especialmente em relação à descrição de serviços voltada para a sua descoberta.

1.1. Caracterização do Problema

A descoberta de serviços Web buscando atender necessidades específicas tem como desafio a diversidade de suas características, muitas vezes complexas de serem descritas (DUMAS et al, 2001). Esses serviços, ainda, possuem características dinâmicas, ou seja, podem estar ou não disponíveis ou podem possuir características diferentes (por exemplo, o custo), dependendo da data em que são procurados. Por isso, dizemos que sua descoberta deve ser *dinâmica*, isto é, deve considerar a natureza mutável e também a riqueza de suas características.

Para atender a necessidade dessa descoberta dinâmica é necessário fornecer descritores precisos para os serviços Web. No contexto da descoberta dinâmica não buscamos responder a perguntas simples do tipo: “quais os serviços Web fornecidos pela Fundação Oswaldo Cruz que pertencem à área de Bioinformática”. Buscamos respostas para perguntas que envolvem aspectos diversos, tais como: “quais os serviços Web de Bioinformática que fazem alinhamento de seqüência de nucleotídeos em bancos de dados genômicos, utilizando o algoritmo Blast, gratuitos, com desempenho médio de 1.0 ms, que utilizem o banco de dados GenBank, disponíveis hoje, domingo, e que retornem a resposta no formato XML”.

Como podemos observar, o escopo dos descritores a considerar é amplo e complexo. Eles precisam ser organizados, de forma que a busca possa ser encarada sob diversas perspectivas, tais como: tempo, espaço, qualidade, entidades envolvidas, recursos utilizados, dentre outros. Entretanto, os padrões atuais relacionados aos serviços Web possuem limitações.

Os serviços Web são geralmente descritos através de um dialeto XML chamado WSDL (Web Services Description Language) (WSDL, 2001) e publicados em registros que estão de acordo com um padrão chamado UDDI (Universal Description, Discovery and Integration) (UDDI, 2001). Porém, os mecanismos de categorização e busca fornecidos pelo UDDI e os atributos de descrição presentes na linguagem WSDL são insuficientes para apoiar a descoberta precisa de serviços Web. WSDL tem seu foco na definição de atributos funcionais, voltados para a chamada do serviço, não considerando informações importantes para uma descoberta precisa, como, por exemplo, QoS (qualidade do serviço), custo ou regras de utilização (definidas por acordos de uso de serviço). As buscas no UDDI, por sua vez, são limitadas a poucas opções, tais como: empresa, nome ou categoria do serviço e área geográfica. Além disso, em ambos os casos, WSDL e UDDI, não fornecem uma categorização adequada das características dos serviços Web e suas relações.

Buscando preencher esta lacuna deixada pelos padrões, outras abordagens vêm sendo propostas, como será discutido no Capítulo 3, mas estas também carecem de uma organização sistemática, que permita perceber o serviço sob diversas facetas, como é necessário para uma descoberta precisa

A organização dos atributos dos serviços em categorias viabiliza essa percepção multifacetada, na medida em que sintetiza os atributos em níveis de abstração mais abrangentes. Entretanto, é preciso uma forma sistemática de organizar os atributos. Quando a categorização é muito genérica, ela estimula esforços, muitas vezes redundantes, de criação de uma grande variedade de vocabulários não-uniformes, dificultando o consenso entre eles. De acordo com Lenci (2000), a definição vaga de classes muito genéricas pode levar a interpretações equivocadas entre aplicações, aumentando o problema da compatibilidade entre vocabulários.

Além disso, os serviços Web estão sempre ligados a uma área de aplicação, tal como Comércio Eletrônico, Turismo ou Bioinformática, necessitando não só de um vocabulário básico de serviços Web para a sua descrição, como também de vocabulários específicos da área a que pertencem. A necessidade de estender um vocabulário básico de serviços Web para a área de aplicação do serviço traz agravantes, tais como: a possibilidade da criação indiscriminada de vocabulários estendidos, a falta de um guia de como gerar esses vocabulários, a falta de divulgação de vocabulários já existentes, a identificação de quais vocabulários devem ser utilizados evitando a criação de vocabulários sobrepostos.

1.2. Hipótese

Os padrões atuais de descrição de serviços Web podem ter sua precisão melhorada através do uso de vocabulários de extensão baseados em *ontologias*¹. Esses vocabulários partem de um núcleo básico que possui diversos atributos funcionais e não funcionais que devem ser estendidos para diferentes áreas de aplicação.

O uso de *ontologias* provê maior semântica às descrições possibilitando inferências mais ricas. Entretanto, as ontologias devem ser utilizadas em um ambiente que apóie o seu reuso e extensão, e que facilite a definição e organização de conceitos.

Com base nisso, o trabalho elaborado nessa dissertação parte das seguintes hipóteses:

¹ O conceito de Ontologia é apresentado no Capítulo 3. Por ora, é suficiente considerar ontologia como um conjunto de conceitos e seus relacionamentos.

- O uso de um esquema de categorização favorece a organização dos descritores de forma mais bem estruturada, possibilitando analisá-los sob diferentes perspectivas.
- A utilização de uma abordagem sistemática para estender uma ontologia básica de serviços Web favorece a criação de vocabulários mais uniformes e precisos.
- A utilização de uma ferramenta de software onde vocabulários podem ser manipulados de acordo com uma abordagem sistemática, contribui para incentivar o reuso destes.

1.3. Objetivos da Dissertação

O objetivo desta dissertação é apresentar uma abordagem sistemática para a definição e organização de descritores de serviços Web, permitindo criá-los ou estendê-los a partir do reuso de vocabulários já existentes. Essa abordagem é expressa através de um conjunto de diretrizes que é apoiada por uma solução tecnológica, através de um protótipo de uma ferramenta de software, OntoGuide-4WS. Esta proposta visa auxiliar na descoberta *dinâmica* de serviços Web com base em seus atributos funcionais e não funcionais, de acordo com a área de aplicação do serviço. Para isso, espera-se permitir o emprego de diferentes tipos de ontologias e para diversas áreas de aplicação, incentivando o seu reuso através da ferramenta OntoGuide-4WS.

Dessa forma, buscamos responder aos seguintes questionamentos:

- *Como organizar melhor as características dos serviços Web, e ao mesmo tempo, possibilitar o uso de mecanismos de inferência que apoiem a sua descoberta mais precisa?*
- *Como facilitar a extensão e criação dos vocabulários utilizados para descrever os serviços e como incentivar o reuso de vocabulários já existentes?*

Dada a variedade e, muitas vezes, a complexidade das características dos serviços Web, não é objetivo deste trabalho descrevê-los de forma completa, nem acreditamos que isso seja viável. Entretanto, é necessário levantar essas características

e prever mecanismos de extensão que possam ser incorporados ao ambiente proposto em trabalhos futuros, visando atender essas questões.

Nossa proposta baseia-se em fornecer uma solução adequada que permita apoiar as tarefas de criação e extensão de vocabulários, com o foco na descrição de serviços Web. A partir daí, espera-se minimizar a complexidade da tarefa de descrever esses serviços no contexto de seu domínio ou área de aplicação, obtendo assim ontologias mais uniformes e com maior chance de reuso.

Por fim, a descoberta *dinâmica* de serviços implica em fornecer dados precisos para a sua localização e chamada. Nesse sentido, nossa abordagem é utilizada de forma complementar, podendo ser aplicada para estender quaisquer iniciativas de descrição de serviços Web, idealmente já aproveitando os padrões existentes, sempre que possível. Por exemplo, estendendo os descritores utilizados no UDDI, que já permitem apontar para o documento que descreve os dados de chamada do serviço.

1.4. Enfoque de Solução

Para atender os objetivos que buscamos, o enfoque de solução adotado nesse trabalho parte do estudo das principais características dos serviços Web e também de iniciativas de organização de domínios de conhecimento, utilizadas na área da Ciência da Informação

A identificação das características genéricas é o ponto de partida para a elaboração de um vocabulário básico de serviços Web que pode ser estendido para diversas áreas de aplicação. Dessa forma, como resultado do estudo dos serviços, obtivemos um vocabulário constituído por um grupo básico de descritores, que deve ser estendido de acordo com a área de aplicação do serviço.

Esse vocabulário é implementado como uma ontologia, pela maior semântica que esta oferece. Entretanto, nossa proposta não é a construção de uma nova ontologia de serviços, mas sim apontar os seus atributos básicos e uma forma de organizá-los.

Nesse sentido, entendemos que o uso de um esquema de categorização bem planejado é uma boa forma de atingir este objetivo. Porém, ainda que este esquema seja um importante requisito para uma boa descrição de serviços, ele não é suficiente. Desta

forma, é preciso uma abordagem sistemática de como criar e estender conceitos e suas relações e uma maneira prática de utilizá-la.

Para isso, desenvolvemos um conjunto de diretrizes que fornece uma abordagem sistemática para a descrição de serviços Web. Embora o foco escolhido tenha sido em serviços Web, algumas das diretrizes são de uso genérico, podendo ser aplicadas em outros domínios, o que é necessário, uma vez que vamos fazer uso delas para descrever a área de aplicação dos serviços.

Para apoiar seu uso, desenvolvemos dois protótipos: uma ferramenta de software baseada nas diretrizes propostas, chamada OntoGuide-4WS (CAMPOS, PIRES, CAMPOS, 2004a) e uma ferramenta de busca, para localizar serviços.

O protótipo da ferramenta de busca por serviços Web parte do pressuposto da já existência de uma ontologia de descrição de serviços Web, bem como de um conjunto de serviços Web descritos com esta ontologia. Sua finalidade é demonstrar que descritores mais ricos, obtidos a partir da utilização das diretrizes propostas nesse trabalho, se traduzem em vantagens na busca por serviços.

A experiência com as diretrizes propostas mostrou que os descritores resultantes são mais adequados para a descoberta precisa, pois apresentam uma representação dos conceitos e suas relações mais próximas do ponto de vista do consumidor dos serviços, permitindo que a busca possa ser encarada sob diversas perspectivas, que este percebe como importantes.

Para a implementação de ambas as ferramentas, adotamos o enfoque da utilização de protocolos neutros e de software livre, além da utilização de padrões já existentes, sempre que possível. Desta forma, minimiza-se a dependência de plataforma e amplia-se a chance de evolução e aceitação das ferramentas.

1.5. Organização

Esta dissertação encontra-se organizada conforme a seguir.

O Capítulo 2 apresenta uma revisão sobre objetos distribuídos, serviços Web e tecnologias padrão relacionadas à descrição e publicação de serviços Web.

No Capítulo 3, descrevem-se outras iniciativas relacionadas à descrição e publicação de serviços Web que não as propostas como padrão, incluindo as que utilizam ontologias. São discutidos também os diferentes aspectos dos serviços Web, derivando assim os atributos necessários para a sua descrição mais precisa. Além disso, são apresentadas abordagens complementares, que vão dar o subsídio necessário sobre como tratar a organização dos atributos dos serviços Web.

O Capítulo 4 apresenta a solução proposta por esse trabalho, detalhando o conjunto de diretrizes utilizado para descrever serviços Web de forma mais precisa.

O Capítulo 5 discute a ferramenta OntoGuide-4WS, que implementa este conjunto de diretrizes através de um ambiente para reuso e extensão de ontologias. Neste capítulo, ilustramos ainda a viabilidade da solução proposta através de um exemplo de aplicação na área de Bioinformática.

Por fim, o Capítulo 6 conclui este trabalho retomando o problema abordado, e apresentando os resultados alcançados e contribuições, limitações, problemas encontrados e possibilidades de trabalhos futuros.

CAPÍTULO 2

Descrição, Publicação e Descoberta de Serviços Web

O uso de serviços Web é apoiado por uma arquitetura padrão, que fornece mecanismos para a sua descrição publicação e invocação (GRAHAM et al., 2002). Entretanto, o uso generalizado e ubíquo dos serviços Web demanda necessidades de descoberta cada vez mais flexíveis e dinâmicas.

Para isso, é importante conhecer de forma precisa e detalhada as características dos serviços Web, de modo que a sua descrição reflita as necessidades que o usuário espera identificar, e que possam ser entendidas por este através de um vocabulário que seja inequívoco.

Este vocabulário, dada a complexidade das características levantadas, deve ser criteriosamente organizado, de modo a estimular o reuso de vocabulários já existentes e de favorecer o uso de inferências, que venham a ampliar a abrangência dos serviços descobertos.

Estas questões, entretanto, possuem pontos em comum com a do reuso de componentes de software, o que é de se esperar, uma vez que serviços Web também podem ser considerados como tal. Dessa forma, é importante explorar iniciativas já adotadas para descrição de componentes e analisar a sua adequação às questões relativas aos serviços Web.

Nesse sentido, neste capítulo vamos iniciar apresentando a arquitetura de serviços Web. Em seguida, vamos fazer uma análise integrada das características dos serviços Web e sua organização, e levantar até que ponto os padrões atuais, presentes na sua arquitetura, contemplam essas necessidades. A partir daí, vamos apontar as limitações dos padrões existentes, delineando um cenário que permite compreender a importância de outras iniciativas que abordam estas limitações.

2.1. Arquitetura de Serviços Web

A arquitetura padrão de serviços Web, orientada a serviços, é chamada de SOA (Service Oriented Architecture). De acordo com Tsalgatidou e Pilioura, (2002), uma arquitetura orientada a serviços deve apoiar a descrição, publicação, descoberta, invocação do serviço (envolvendo aspectos como protocolos de envio e transporte de mensagens pela rede), e ainda, a sua remoção, caso ele não esteja mais disponível. A SOA possui um conjunto de especificações que norteiam a utilização dos serviços em relação a esses aspectos.

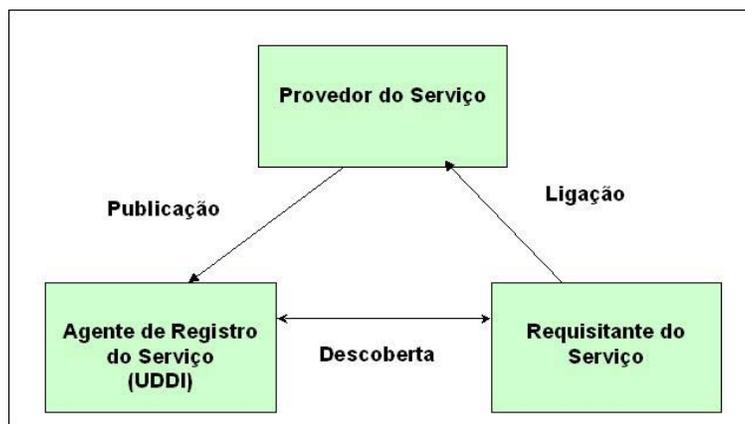


Figura 1-SOA – Arquitetura Orientada a Serviços

A arquitetura possui três componentes básicos: o provedor do serviço, o agente de registro e o requisitante do serviço.

O *provedor do serviço* divulga ou publica no agente de registro os serviços que disponibiliza e responde a pedidos de uso dos mesmos. O provedor é ainda responsável por descrever as informações de ligação do serviço, usadas para chamá-lo. Estas informações estão contidas em um documento XML escrito na linguagem padrão WSDL (Web Service Description Language) (WSDL, 2001).

O *agente de registro do serviço (broker)* fornece facilidades para armazenar os dados básicos dos serviços, tais como seu nome, provedor e categoria e permite buscas por esses atributos. O padrão adotado na SOA para registro é o UDDI (Universal Description, Discovery and Integration) (UDDI, 2001).

O requisitante do serviço pesquisa o agente de registro de serviço para descobrir² o serviço desejado e buscar informações sobre como efetuar a sua chamada, ou seja, quais métodos estão disponíveis, que parâmetros estes possuem, qual o protocolo utilizado e em que endereço o serviço está disponível.

É importante observar a diferença entre um registro e um repositório. O registro redireciona o usuário para os recursos, enquanto que o repositório é a fonte real da informação. O UDDI é um registro. Dessa forma, as buscas ao UDDI conduzem a uma interface ou a uma implementação localizada em um outro servidor (SEELY, 2001). Da mesma forma, o registro também aponta e não incorpora o documento WSDL com as descrições de chamada dos serviços. Essa característica possui como vantagem uma maior independência em relação às implementações e descrições dos serviços, uma vez que estas não estão contidas no registro. Por outro lado, o documento WSDL não descreve atributos que identifiquem inequivocamente o propósito do serviço e suas características não funcionais. Além disso, as buscas no UDDI não são feitas com base nos atributos do documento WSDL. Elas são feitas com base em alguns descritores do serviço presentes no UDDI.

Como se pode observar, as informações que descrevem o serviço estão separadas em locais distintos e, com propósitos distintos. As informações que estão armazenadas no agente de registro estão mais voltadas para permitir localizá-lo enquanto que as outras estão voltadas para a sua chamada. Além disso, seus padrões são definidos de forma independente. Neste trabalho chamamos o primeiro (UDDI) de padrão de publicação e o segundo (WSDL) de padrão de descrição.

2.1.1. Padrão de Descrição de Serviços Web

Os provedores de serviços, em geral, descrevem os serviços Web através da linguagem WSDL. Embora o uso do WSDL não seja obrigatório, ele é recomendado pelo W3C, sendo, na prática, a linguagem de descrição mais utilizada para serviços Web. O documento escrito em WSDL tem por finalidade descrever um serviço Web através de suas interfaces, ou seja, os métodos que são disponibilizados pelo serviço Web, e também os parâmetros recebidos, a resposta enviada. O documento WSDL

² Consideramos aqui a descoberta como sendo a busca bem-sucedida por um serviço Web. A descoberta é apoiada pelo agente de registro, padrão de publicação, UDDI.

também descreve os dados de ligação (*binding*), informando detalhes sobre protocolos de comunicação e endereço físico de chamada do serviço. Um documento WSDL é composto pelos seguintes elementos:

- **Type.** Definição de tipos de dados XML que podem ser definidos pelo usuário e utilizados depois nos elementos *message*.
- **Message.** Uma definição abstrata dos dados que são transmitidos. É formada por um conjunto de elementos *parts*, que permitem que a mensagem possa ser dividida em partes menores. Cada elemento *message* pode ser uma mensagem de entrada, saída ou de erro.
- **Operation.** Uma definição abstrata de uma função realizada pelo serviço. Define um método em um serviço Web, com seu nome, parâmetros de entrada e o tipo de retorno do método.
- **Port Type.** Encapsula um conjunto abstrato de elementos *operation* disponibilizados em um ou mais pontos terminais (porta).
- **Binding.** Informa ao requisitante do serviço como formatar a mensagem em um protocolo concreto, para um determinado *port type*.
- **Port.** Um ponto terminal, ou porta, este elemento possui um atributo com dados de ligação (*binding*) e um sub-elemento contendo o endereço de rede (URL). Indica a localização real do serviço.
- **Service.** Uma coleção de elementos *port* relacionados.

No documento WSDL serviços são coleções de pontos terminais de rede, ou portas (ports). A definição abstrata desses serviços em *port types* no documento WSDL permite desacoplar a descrição do serviço da sua implementação física, permitindo o reuso dessas definições abstratas em várias implementações distintas.

No anexo A-1, temos um exemplo de Tsalgatidou e Pilioura (2002), de um documento WSDL.

2.1.2. Padrão de Publicação de Serviços Web

O modelo de publicação de serviços Web utilizado atualmente, o UDDI, é patrocinado pela UDDI.org (UDDIORG, 2004), uma organização composta por mais de duzentos importantes fabricantes e líderes de negócios eletrônicos (e-business).

O UDDI é uma especificação formada por um conjunto de documentos XML com informações sobre os serviços e sobre as empresas que os fornecem, e uma interface de programação (API) utilizada para descoberta e registro de serviços Web (UDDIEXEC, 2001).

Algumas empresas, tais como IBM (2004), Microsoft (2004) e SAS (2004), implementaram esta especificação, fornecendo um serviço de descoberta de serviços Web através de pesquisa no diretório UDDI, que é replicado e espelhado entre as diversas implementações.

Os provedores de serviço registram ou publicam seus serviços Web nesses diretórios e um mecanismo de busca descobre o serviço Web para um cliente, mediante uma pesquisa por palavras-chave. É interessante observar, no entanto, que o UDDI pode ser utilizado para publicar serviços diversos e não apenas programas.

O UDDI possui um registro de “páginas brancas”, “páginas amarelas” e “páginas verdes”. As páginas brancas contêm informações sobre a empresa fornecedora do serviço, tais como seu nome e informações de contato. As páginas amarelas contêm informações genéricas sobre o serviço, tais como seu nome, categoria do serviço³ e localização geográfica. Já as páginas verdes contêm ponteiros para informações técnicas envolvidas na interação com o serviço, tais como parâmetros e métodos de chamada, cujas definições abstratas estão em outro lugar, geralmente sob a forma de um documento WSDL. Um ponteiro no UDDI provê uma indicação para o local físico, ou URI (Universal Resource Identification), onde se encontra esse documento.

A comunicação com o UDDI se dá normalmente através de mensagens SOAP (2003) e, após a localização do serviço, são obtidas as informações necessárias para a chamada ou ligação com o serviço Web (binding), tais como: URL, nome dos métodos e tipos de argumentos (VASUDEVAN, 2001).

³ Categorias de serviços são geralmente organizadas de acordo com taxonomias padrão da indústria como a NAICS - North American Industry Classification System (NAICS, 2003) e servem para agrupar os serviços Web de acordo com a área de negócio a que estes pertencem.

As informações que descrevem os serviços estão em documentos XML, que definem quatro tipos de estruturas de dados (HENDRICKS, 2002):

- ***business entities***: correspondem às páginas brancas. Descrevem informações sobre a empresa fornecedora e seus serviços oferecidos. Seus dados estão divididos conceitualmente em informações de contato, informações sobre os tipos e localização dos serviços e detalhes de como chamar os serviços oferecidos pela empresa fornecedora. Esta estrutura é formada por elementos `businessServices`.
- ***businessServices***: correspondem às páginas amarelas. Descrevem uma coleção de serviços Web oferecidos. Contém o nome, descrição textual do serviço, assinatura digital (se for o caso). Além disso, esta estrutura é formada por elementos `bindingTemplates` e `categoryBags`. Estes últimos utilizados para categorizar o serviço de acordo com taxonomias padrão utilizadas na indústria, tais como a Dun & Bradstreet Data Universal Numbering System (DUNS, 2004).
- ***bindingTemplates***: correspondem às páginas verdes. Informam a forma de acesso ao serviço, geralmente através de uma URL⁴, e contêm referências para os elementos `tModels`.
- ***tModels***: este elemento permite que grupos de indústrias e organismos de padronização especifiquem definições abstratas e reutilizáveis de tipos de serviços, que podem ser utilizadas e combinadas por outros. O `tModel` pode conter o elemento `overviewDoc` que, por sua vez, pode conter o elemento `overviewURL` que faz referência a uma URL correspondente a um documento WSDL (no caso de serviços Web), que descreve o serviço. O `tModel` pode conter ainda os elementos `categoryBag` e `identifierBag`, ambos utilizados para categorizar o serviço de acordo com taxonomias padrão utilizadas na indústria, descrevendo aspectos tais como: indústria,

⁴ Note-se que nem todo serviço publicado no UDDI é um serviço Web. Pode haver, por exemplo, um serviço convencional de entrega de pizzas por telefone. Nesse caso, o endereço de acesso pode ser um telefone, ou uma página na Web.

categoria do produto ou região geográfica, da mesma forma que os elementos `businessService`.

A versão atual das implementações públicas do UDDI é a versão 2, mas a versão 3 já se encontra em fase final de aprovação na UDDI.org (UDDIORG, 2004). Aspectos importantes na versão 3 são o apoio a assinaturas digitais, a melhoria das pesquisas ao registro através de consultas aninhadas e o uso de caracteres genéricos de busca (wildcards). Podem-se ainda associar palavras-chave aos `tModels` criados, para serem usadas nas buscas.

Neste item foram apresentados padrões relacionados à descrição e publicação de serviços Web. Para podermos analisá-los em relação às necessidades de buscas precisas de serviços é importante compreender o que se entende exatamente por descoberta precisa, que doravante chamamos de *descoberta dinâmica*, e quais são os seus requisitos.

2.2. Descoberta Dinâmica de Serviços Web

Os serviços Web podem ser usados individualmente, mas também, devido à modularidade e independência de tecnologia, podem ser combinados. Dessa forma, novos serviços podem ser criados por meio da integração e composição dos serviços de uma empresa com outros oferecidos pelos parceiros, agregando valor ao negócio (CASATI et al, 2000). Considerando a velocidade com que os negócios mudam e a competitividade aumenta, é natural esperar que serviços antes existentes deixem de existir ou que novos serviços venham a ser oferecidos com mais vantagens que os dos antigos parceiros, ou ainda que mudanças no cenário econômico, político ou de mercado obriguem a empresa a mudar seus objetivos também. Dessa forma, é preciso uma adaptação rápida a essas mudanças, possibilitando que novos serviços possam ser descobertos e combinados de forma ágil e flexível, em tempo de execução⁵, de acordo com as novas características desejadas.

A composição de serviços pressupõe que os serviços possam ser publicados, localizados e seu funcionamento e características compreendidos pelos interessados na

⁵ Fazer a ligação com o serviço em tempo de execução implica em não fixar em tempo de programação os dados de sua chamada, deixando esta tarefa para o momento de execução do serviço.

composição, de forma que se possa decidir se um dado serviço ofertado se encaixa nos requisitos de seu próprio negócio e atende às suas necessidades

Para que essa decisão se dê de forma mais ágil, com o mínimo de intervenção humana, é necessário dispor de um mecanismo de descoberta preciso e automatizado, que chamamos *descoberta dinâmica dos serviços*. Por descoberta dinâmica de serviços, nesse trabalho, entende-se o mecanismo de software que permite *buscar*, *localizar*, *selecionar* e, alternativamente, *compor* serviços a partir de um determinado local onde estes são registrados, de acordo com parâmetros de busca.

Nessa definição, o termo *dinâmica* se refere ao fato de que a localização física do serviço ou dos dados que descrevem a sua chamada não são conhecidos à priori e não são fixos, ou seja, podem mudar. É conhecido apenas um local de referência, com informações de vários serviços Web, a partir do qual a descoberta é feita.

Assim, *buscar*, no contexto dessa definição, consiste em procurar por serviços, a partir de parâmetros fornecidos que restringem o escopo da busca; *localizar* consiste em achar os serviços que atendem aos requisitos de busca informados por aquele que deseja descobrir os serviços; *selecionar* é escolher o serviço mais adequado de acordo com os critérios de desempate⁶ (FONER, 1996) escolhidos por aquele que deseja descobrir os serviços e, por fim, *compor* serviços, no processo da descoberta dinâmica, não é o mesmo que combinar dois ou mais serviços que já têm suas características individuais identificadas. Há casos em que um único serviço não atende aos requisitos de descoberta individuais desejados, mas que, alternativamente, dois ou mais serviços combinados podem atender. O processo de descoberta e encadeamento desses serviços de forma que funcionem de forma transparente, como se fossem um só, é o que chamamos aqui de *composição dinâmica* de serviços. Esta envolve mais do que simplesmente achar serviços que combinados executem a função desejada. É preciso tratar também questões não relacionadas à função do serviço, tais como integridade de transações e substituição eventual de serviços não disponíveis. E ainda, tratar atributos que, quando combinados, podem variar, tais como o tempo médio de execução, o custo conjunto dos serviços e contratos que regulam a sua utilização. Por exemplo, um determinado serviço *x* pode ser combinado com os serviços *y* e *z* para fornecer um

⁶ Critérios de desempate servem para escolher o serviço mais adequado dentre os localizados. Exemplos são: custo e tempo de execução, dentre outros.

serviço de acompanhamento de entrega de mercadoria. Porém o serviço x pode ser executado em paralelo ao serviço y . Desta forma, o tempo médio de execução total não é a média dos tempos de x , y e z , neste caso. Embora possa vir a ser, caso se empreguem serviços que não sejam executados em paralelo.

Ao minimizar a intervenção humana no processo da descoberta dinâmica, cresce a necessidade de aumentar o nível de detalhe e precisão da descrição dos serviços. Sem a possibilidade do julgamento humano, descritores vagos podem trazer o que se chama de *falsos positivos*, retornando resultados equivocados, ou ainda *falsos negativos*, não se conseguindo localizar os serviços de acordo com as perspectivas que seriam desejadas. Por exemplo, na área de Bioinformática, devido ao sigilo das informações tratadas e do seu grande volume, muitas vezes pode ser conveniente publicar um serviço não só para que este possa ser utilizado pela Internet, mas também para que, alternativamente, este possa ser baixado livremente para ser implementado em uma Intranet em um instituto de pesquisa. Neste caso, se este tipo de serviço é o que nos interessa, é importante que a característica de sua licença de uso seja descrita juntamente com a função oferecida por ele. Caso contrário, se a função do serviço nos atende, mas não a sua licença de uso, a seleção deste serviço em uma busca é um falso positivo. Outrossim, se nossa intenção principal é a busca pelo tipo de licença e este descritor não está disponível, temos de recorrer a outros que se aproximem. Nesse exemplo, poderíamos procurar pelo custo, assumindo que talvez um serviço gratuito permitisse ser baixado livremente, mas isso não é verdade necessariamente, e assim, poderíamos estar deixando de selecionar vários serviços que nos atenderiam, caracterizando o falso negativo.

Dessa forma, para entendermos o nível de detalhes que pode estar envolvido na descrição de serviços Web, com vistas ao processo da descoberta dinâmica, é importante analisar as características desses serviços.

2.3. Características dos Serviços Web

O estudo das características dos serviços Web é feito a partir de duas fontes de pesquisa: a análise de iniciativas de descrição presentes na literatura e a análise de serviços Web existentes. A primeira fonte nos dá os detalhes de **quais** atributos, funcionais e não-funcionais, devem ser descritos. A segunda nos dá indícios de **como**

eles devem ser descritos, devido às diferentes maneiras que os serviços são implementados.

No escopo desse trabalho, atributos funcionais são os voltados para descrever a parte operacional ou funcional do serviço, necessária para que este possa ser executado. Por exemplo, quais as operações disponíveis, seus parâmetros, protocolos e endereço de chamada do serviço.

Já os atributos não-funcionais dizem respeito a aspectos não diretamente ligados à execução do serviço. Por exemplo, aspectos de qualidade e custo monetário, e ainda o propósito do serviço e suas restrições de acesso. Estes atributos são relevantes para se identificar o serviço desejado e não para executá-lo.

A maneira como os serviços são implementados influi na forma como devemos descrever esses atributos funcionais e não-funcionais. Por exemplo, um serviço de informe sobre o tempo pode fornecer uma única operação que aceita um parâmetro de entrada indicando o nome da cidade para o qual é dado o informe. Alternativamente, o mesmo serviço pode ser implementado através de n operações distintas, uma para cada cidade para a qual é dado o informe. Embora o serviço em si seja o mesmo, ele deve ser descrito de forma diferente para cada uma das implementações.

O levantamento dos atributos funcionais e não-funcionais e as questões envolvidas em como descrevê-los, obtidas a partir da análise de serviços já existentes (XMETHODS, 2004), são discutidos a seguir.

2.3.1. Atributos Funcionais dos Serviços Web

Um serviço Web possui uma definição funcional abstrata e uma física. A definição funcional abstrata diz respeito à forma de uso do serviço, sem levar em conta aspectos da implementação física tais como protocolos e endereços físicos de chamada. Sob este ponto de vista nos interessa saber as **operações abstratas** do serviço, ou seja, as funções que o serviço oferece, descrita pelos seus nomes, tipo (entrada, saída ou erro) e parâmetros.

Podemos também pensar o serviço Web como sendo uma implementação física daquela operação abstrata que executa uma atividade. A implementação física diz respeito a como o provedor torna disponível o serviço. Sob este ponto de vista nos

interessa saber duas coisas: **como** vamos **chamar** o serviço, ou seja, qual os protocolos de empacotamento das mensagens e de transporte; **onde** o serviço é **fornecido**, ou seja, em que endereço físico (URL) está localizado o serviço.

Em relação aos itens **como** e **onde**, cabe uma questão: se um serviço é implementado de duas formas diferentes, por exemplo, utilizando diferentes protocolos de comunicação e empacotamento, tais como HTTP com SOAP e usando HTTP com GET, cada uma destas implementações é um outro serviço? Na nossa interpretação, vamos considerar que sim. Afinal, dependendo do local e do protocolo utilizado, pode haver mudanças em alguns atributos de qualidade e custo monetário, caracterizando para efeitos de descoberta, uma atividade com características distintas, embora derivada da mesma operação abstrata. Por exemplo, um determinado serviço pode ser implementado em dois endereços físicos distintos. Um dos endereços é localizado em um servidor de alto desempenho, com controle de acesso restrito e cobrado. O outro endereço é localizado em um servidor de baixo desempenho, acesso liberado e gratuito. A função oferecida, entretanto, é a mesma, embora seus atributos de qualidade e custo sejam diferentes, dependendo do local de sua implementação física.

Os atributos funcionais de um serviço Web estão contemplados satisfatoriamente no documento WSDL que o descreve, conforme apresentado na Seção 2.1.1.

2.3.2. Atributos Não-Funcionais dos Serviços Web

Os atributos não-funcionais dos serviços Web são de diferentes naturezas. Alguns dizem respeito à qualidade do serviço, outros a seu custo monetário. Todos, no entanto, estão envolvidos em descrever características do serviço, sem, entretanto, descrever aspectos da sua implementação física.

Esses atributos, em seus diferentes aspectos, são: qualidade, regulação do uso, custo monetário, contexto ou escopo de uso, pré e pós-condições de uso, propósito e reputação.

No escopo deste trabalho, *qualidade de serviço* (QoS) é encarada como um conjunto de atributos não funcionais do serviço que o qualificam para ser avaliado em relação a três núcleos básicos: *execução*, *informação* e *reputação*. Qualidade de

execução contém dados relacionados à execução do serviço, tais como: tempo de resposta, prioridade, disponibilidade do acesso e confiabilidade. Qualidade de *informação* contém dados relacionados à informação fornecida, tais como: privacidade, integridade e segurança. Qualidade da *reputação* contém informações relacionadas a como os seus usuários percebem a sua qualidade.

A qualidade do serviço se desdobra em várias facetas. Um relatório técnico (CAMPOS, PIRES, CAMPOS, 2004b) fornece uma discussão mais detalhada sobre elas, abordando aspectos tais como privacidade, integridade e segurança, tratados em diversas iniciativas (WSRELIA, 2003) (XAML, 2000) (XKMS, 2001) (XMLENC, 2001) (BOSWORTH et al., 2003) (BOX et al., 2003) (CARDOSO et al.) (MIKALSEN, ROUVELLOU, TAI 2001) (MIKALSEN, TAI, ROUVELLOU, 2002) (MYERSON, 2002) (MYERSON, 2003), e também de outros atributos discutidos resumidamente a seguir.

Embora não haja um padrão de descrição de serviços Web que contemple atributos de qualidade de serviço de forma geral, algumas iniciativas que já foram feitas em respeito a essa questão WSOL (TOSIC, PATEL, PAGUREK, 2002), WSIL (MODI, 2002), WSQD (AZEVEDO JR, 2003), WSEL (WSEL, 2001), WSLA (KELLER, LUDWIG, 2002), DAML-S (ANKOLEKAR et al., 2001) e ainda as contribuições de Sahai, Durante e Machiraju (2002), e Mani e Nagarajan (2002), que descrevem de maneira mais detalhada os aspectos de qualidade de um serviço Web.

Além dos atributos de qualidade, da mesma forma que qualquer aplicativo de software disponível publicamente, mesmo que não pago, é natural que se implementem mecanismos para regular o uso dos serviços Web. Para isso, além de informar se o serviço está em conformidade com um SLA (vide item 2.3.2.1), é importante descrever as cláusulas desse SLA. Sahai, Durante e Machiraju (2002) desenvolveram na HP uma linguagem baseada em XML com o objetivo de descrever SLAs para serviços Web. Esta linguagem, referida por Patel (2003) como WSML, permite a descrição dos atributos de qualidade e suas métricas de modo preciso, flexível e formal.

Quanto ao custo monetário do serviço, ainda não há um modelo padrão para fixá-lo ou de estabelecer a cobrança de serviços Web, embora haja vários trabalhos abordando a questão de custo de serviços fornecidos pela Internet (STILLER et al.,

2001) (BRADY-MYEROV et al., 2001) (EIBACH, KUEBLER, 2001) (GERKE, FLURY, STILLER, 2000), alguns citando os modelos de cobrança utilizados na telefonia celular (BRADY-MYEROV et al., 2001) (STILLER et al., 2001).

Há também a questão do contexto (DEY, 2001). De acordo com Brézillon (1999), “Contexto é o que limita a resolução de um problema, sem interferir explicitamente com ele “. Como exemplo podemos citar uma restrição geográfica para a realização do serviço, ou seja, o serviço só pode ser executado em um determinado país ou região. Esta limitação, no entanto, não interfere explicitamente com a função realizada por ele.

O contexto influencia as pré e pós-condições de uso do serviço, pois estas limitam a resolução de um problema, sem interferir explicitamente com ele. De acordo com Frankel e Parodi (2002):

“Uma pré-condição é uma restrição que afirma que alguma coisa deve ser verdade para que uma operação seja executada. Uma pós-condição é uma restrição que afirma que alguma coisa tem de ser verdade após o término da execução da operação”).

Pré-condições são importantes para especificar as características de um serviço de forma mais precisa, e são relevantes para o processo de descoberta e composição de serviços (LARA et al., 2003) (ERMOLAYEV, 2003). O mesmo se pode dizer das pós-condições.

O contexto ainda, ajuda a definir o propósito do serviço. Por exemplo, dois serviços de Bioinformática, de sequenciamento de proteínas, podem possuir os mesmos parâmetros de entrada e de saída, realizar a mesma tarefa, porém utilizando algoritmos diferentes. Descrever qual procedimento é utilizado para resolver um problema pode ser importante para que se possa comparar resultados com o mesmo enfoque (BREZILLON, 1999). Cabe ressaltar que detalhes de procedimentos usados pelo serviço ou o seu propósito não estão necessariamente ligados aos parâmetros de entrada ou saída do serviço⁷.

⁷ O propósito do serviço é abordado com mais detalhes no Capítulo 5, ao se estudar exemplos para uma área de aplicação.

Por fim, existe o aspecto da reputação. O estudo da *reputação de serviços* é bem ilustrado em Maximilien e Singh (2001, 2002), Yu e Singh (2002) e em Zacharia e Maes (2002), com vários exemplos aplicados nas comunidades online, tais como salas virtuais de conversa pela Internet, lista de discussão, e também em sítios de comércio eletrônico. De acordo com Zacharia e Maes (ZACHARIA e MAES, 2002 APUD MARSH, 1994), reputação se define como sendo a quantidade de confiança inspirada por uma pessoa em uma área específica de interesse. Se adaptarmos essa definição a um serviço Web, podemos definir que a reputação de um serviço Web é a quantidade de confiança inspirada por esse serviço Web nos membros de uma comunidade virtual (softwares) que o utilizem.

Nossa análise até aqui evidencia uma grande variedade de descritores, entretanto há outras questões a avaliar que não o levantamento das características que devemos descrever. Um mesmo serviço pode ser implementado de diversas formas, dependendo da forma como o provedor do serviço projeta a sua interface. Isso pode influir na forma como o serviço deve ser descrito, e é estudado a partir da análise de serviços Web existentes, como discutimos a seguir.

2.3.3. Características Levantadas a partir da Análise de Serviços Web Existentes

A análise de serviços Web já existentes nos fornece uma visão complementar da descrição de serviços. Além de saber o que devemos descrever, é importante avaliar até que ponto a forma como a interface do serviço é implementada influi na sua descrição.

Ainda que muitos dos serviços Web estejam implementados em mercados privados e não estejam disponíveis ao público, há também um mercado em expansão que urge para expor seus negócios a parceiros e clientes potenciais (AKIHA, 2003). Uma análise dos diretórios públicos de serviços Web tais como IBM UDDI (IBM UDDI, 2003) e XMethods (XMETHODS, 2004), nos mostra que a disponibilidade de serviços está aumentando, e ainda, que é muito diversa. Ela varia de simples utilitários como conversores de moeda até aplicações complexas de Bioinformática como o BLAST (ALTSCHUL et al., 1990).

Esses serviços possuem características comuns: possuem um provedor, manipulam dados, têm um propósito executado por uma ou mais ações, podem fazer

uso de outros recursos, e todas essas características podem ter atributos de qualidade associados a elas. Entretanto, a análise mais atenta desses serviços nos mostra alguns importantes aspectos, que influenciam na sua descrição.

Para ilustrar esses aspectos vamos supor que estamos procurando por um serviço Web que informe as condições do tempo de uma cidade de um país. Para isso, vamos considerar três serviços Web que estudamos em nossa pesquisa. Estes serviços estão publicados em XMethods.com (XMETHODS, 2004), quando de nossa análise. Todos três informam condições de tempo para cidades, mas suas interfaces são diferentes. Para simplificar, vamos considerar que todos apresentam os dados de saída da mesma forma, embora na verdade elas variem um pouco.

O primeiro é chamado *Global Weather* (GWEATHER, 2004) e fornece dados das condições de tempo para as principais cidades do mundo. Ele oferece duas operações: uma para informar as cidades de um país e outra para informar o tempo em uma determinada cidade de um país. A primeira operação, *GetCitiesByCountry*, pede como parâmetro de entrada o nome do país, devolvendo como resposta uma tabela com os nomes das cidades do país. A segunda operação, *GetWeather*, pede como parâmetro de entrada o nome do país e a cidade, devolvendo como resposta uma tabela com os dados de tempo. Esse serviço é gratuito e não envolve nenhum contrato de uso.

O segundo é chamado *Weather Services from EJSE, Inc.* (EJSE, 2004) e fornece dados das condições de tempo para regiões dos Estados Unidos, indicadas por um código de endereçamento postal, e também para algumas cidades do Iraque. Como nosso interesse é no tempo de cidades, vamos analisar apenas as operações relacionadas a cidades e não a códigos de endereçamento postal. Neste serviço há duas, uma com dados mais genéricos e outra com dados mais completos. Como não estamos nos importando com o formato da saída, vamos considerar apenas uma delas: *GetIraqWeatherInfo*, que pede como parâmetro de entrada o nome de uma cidade. Esse serviço é gratuito e não envolve nenhum contrato de uso.

O terceiro é chamado *DotFastWeather* (DOTFASTW, 2004) e fornece dados das condições de tempo para regiões dos Estados Unidos, indicadas por um código de endereçamento postal, e também para algumas cidades dos Estados Unidos. No caso do tempo das cidades, existe a operação *GetWeatherByCityState*, para a qual deve-se

informar o nome do estado e o nome da cidade, e ainda um código de identificação do usuário, pois o serviço não é gratuito sendo regido por um contrato de uso.

A Tabela 1 resume as características dos serviços acima.

Tabela 1 - características dos serviços de tempo pesquisados em XMethods.com.

| Serviço Web | Operações | Entradas | Saída |
|----------------------------|-----------------------|--|--------------|
| Global Weather | GetCitiesByCountry | país | idades |
| Global Weather | GetWeather | país, cidade | tempo |
| Weather Services from EJSE | GetIraqWeatherInfo | cidade | tempo |
| DotFastWeather | GetWeatherByCityState | estado, cidade, código de identificação | tempo |

O primeiro serviço atende a nossa necessidade (saber as condições do tempo de uma cidade de um país) com a operação `GetWeather`, mas atende também necessidades de quem esteja querendo saber cidades de um país, com a operação `GetCitiesByCountry`, o que não está necessariamente ligado a informes de tempo.

Dependendo do país, o segundo ou o terceiro serviço atende à nossa necessidade. Entretanto, nesses serviços o nome do país está implícito, enquanto que no primeiro serviço ele é um dos parâmetros de entrada da operação. Isso nos leva a concluir que não podemos apenas confiar nos parâmetros para saber se um serviço atende ou não aos nossos interesses.

E ainda, em relação aos parâmetros, há que se tomar cuidado com descritores que significam a mesma coisa, mas com contextos diferentes. Por exemplo, dois descritores que informem o preço de uma mercadoria, sendo um com o frete embutido e outro sem⁸. Neste caso, se formos utilizar o preço retornado como um fator de opção de escolha do serviço, o contexto passa a ser bastante relevante.

A análise da interface dos serviços de tempo revela ainda que um serviço pode ser composto por operações com propósitos distintos, mas que dizem respeito a uma área de aplicação, delineando o escopo de um propósito mais amplo, como um propósito geral. Já o propósito mais específico vai estar associado a cada operação do serviço e sua descrição vai depender da sua área de aplicação. Dessa forma, se

⁸ Outros atributos mais complexos podem estar envolvidos, tais como a definição de métricas, para definir, por exemplo, descritores como o tempo de resposta de uma operação.

desejamos encontrar um serviço que informe o tempo em Bagdá, primeiro selecionamos os serviços de informe de tempo, para depois analisar cada uma de suas operações à procura de alguma cujo propósito seja informar tempos de cidades de um país, ou de cidades do país Iraque, ou ainda especificamente da cidade de Bagdá.

De maneira análoga, há determinados atributos que se aplicam ao serviço como um todo enquanto outros dizem respeito a cada operação e não ao serviço. O provedor do serviço, da sua data de criação, e às vezes o preço de uso⁹, se aplicam ao serviço como um todo. Entretanto, o tempo de execução, por exemplo, depende de que tarefa está sendo executada. Se um serviço possui duas operações, uma simples de consulta e outra mais complexa de efetivação de venda por cartão de crédito, certamente elas têm tempos de execução diferentes. Dessa forma, esses atributos se aplicam a cada operação individualmente.

Entretanto, há casos em que os atributos do serviço ou de suas operações não informam se o serviço nos atende e só vamos saber isso após executá-lo. Isso pode acontecer quando o escopo de uso do serviço não está explícito. Por exemplo, podemos informar de alguma forma uma lista das treze cidades do Iraque que o serviço de tempo atende, ou omitir essa informação. Nesse último caso, se o serviço não informa o tempo de Tikrit, só vamos saber disso após tê-lo executado. Se estivermos utilizando serviços pagos, saber que o serviço não nos atende somente após executá-lo não parece ser razoável.

A partir dessa análise podemos apontar alguns pontos relevantes que influem em relação ao que vamos descrever:

- Um mesmo serviço Web pode apresentar operações com propósitos diferentes, embora pertencentes à mesma funcionalidade.
- Há serviços Web que possuem operações que podem servir para atender necessidades de diferentes áreas de aplicação.
- Serviços Web podem apresentar operações com interfaces diferentes para realizar tarefas semelhantes. Nessas operações há dados que são importantes para a descrição do seu propósito e outros que não. Esses dados podem estar

⁹ O preço de uso pode ser cobrado pelo uso individual de cada operação, mas também pode ser cobrado por mês, por exemplo, englobando todas as operações disponíveis pelo serviço.

implícitos ou explícitos como parâmetros na sua interface e podem representar conceitos iguais em contextos diferentes.

- Existem propósitos gerais, associados ao serviço como um todo, e específicos, associados a cada operação do serviço.
- Existem casos em que só sabemos se o serviço nos atende após executá-lo, pois o seu escopo de uso não está explícito.

Temos de considerar ainda a questão de *como* descrever os aspectos discutidos acima, pois há mais de uma forma possível.

Há que se analisar se descritores que representam conceitos iguais em contextos diferentes vão ser representados como dois descritores distintos ou como um descritor só, porém com algum atributo que os diferencie. Por exemplo: vamos criar os descritores `preçoComFrete` e `preçoSemFrete`, ou apenas o descritor `preço`, porém com uma métrica associada?

Há que se analisar *como* e *se* são descritas as restrições de escopo. Restrições de escopo de uso podem estar associadas a descritores aos quais a restrição se aplica. Por exemplo, no serviço de tempo das cidades do Iraque, poderia estar associada ao descritor `cidade`, através de uma lista das cidades que o serviço atende. Cabe avaliar, no entanto, até que ponto é viável descrever esse tipo de restrição. Nos casos das restrições serem listas muito grandes, talvez não seja conveniente descrevê-lo, pois pode tornar a descrição do serviço muito longa e, conseqüentemente, mais demorada para tratar. Pode-se, entretanto, considerar nesses casos restrições de exceção, ou seja, que descrevem o escopo que o serviço não atende, em vez de descrever o que ele atende. Mesmo assim, pode haver casos em que a lista das exceções pode ser igualmente longa.

Há que se analisar de que forma é descrito o propósito de cada operação do serviço. Naturalmente, devido à natureza diversa das áreas de aplicação dos serviços, os dados necessários para descrever o propósito das operações podem variar muito, evidenciando a necessidade de definição de vocabulários consensuais para descrevê-los.

Há ainda uma outra questão, relacionada com a redundância dos descritores: os parâmetros de entrada e de saída, que são importantes para descrever o propósito serviço, também são importantes para descrever seus aspectos funcionais, estando

presentes no documento WSDL do serviço, associados a aspectos de implementação física. Dessa forma, se vamos descrevê-los separadamente, há que se pensar em uma maneira de relacioná-los.

A questão da redundância dos descritores nos remete a uma discussão mais ampla, que é a da descrição genérica de *atributos complexos*. Por atributos complexos, consideramos os aspectos da descrição que necessitam do uso de vocabulários consensuais para descrevê-los. O propósito e as métricas são casos particulares de atributo complexo, mas podem existir outros, tais como a descrição de um recurso que é usado por uma operação. Por exemplo, ao definir o descritor “banco de dados genômico situado no Japão”, vamos ter de considerar como definir o que é banco de dados, genômico e Japão. Ou simplesmente assumir que “banco de dados genômico situado no Japão” é um conceito conhecido pelos que vão fazer uso da linguagem. Aqui há que se tomar especial cuidado, pois nem sempre esses conceitos vão poder ser separados. Muitas vezes um conceito tem um sentido quando ligado a outro e este sentido se perde ao se separá-los. É o caso do sentido metafórico. Por exemplo, ao descrevermos uma *árvore filogenética* em Biologia, não é conveniente destacar o conceito de *árvore*, pois este é usado ali como um sentido metafórico para denotar a organização dos conceitos em ramos, à semelhança de uma árvore. Já se tivéssemos diversos tipos de *árvore*, que quiséssemos diferenciar por seus atributos, aí seria o caso de ter o conceito definido separadamente.

Essas questões permeiam a forma de descrever os atributos dos serviços Web como um todo, estando ligadas à forma como estes são organizados.

É importante lembrar, também, que muito provavelmente os serviços são desenvolvidos à revelia das linguagens criadas para descrevê-los. Sendo assim, estas têm de considerar que suas interfaces podem estar construídas de muitas maneiras diferentes. Deve-se levar em consideração inclusive que pode haver casos de serviços que não vão ser possíveis de descrever com um determinado vocabulário, mesmo pertencendo à área de aplicação do vocabulário. Outros ainda podem pertencer a mais de uma área de aplicação. Neste caso, cabe avaliar se é interessante para o provedor reformular a sua interface, descrevê-lo em uma das áreas ou em ambas.

Com base no que foi discutido até aqui, podemos resumir as questões levantadas nessa seção nos seguintes tópicos:

- É preciso avaliar se um serviço é descrito como pertencendo a mais de uma área de aplicação.
- É preciso diferenciar os descritores que dizem respeito ao serviço como um todo dos que dizem respeito a cada operação do serviço.
- É preciso determinar, quando o contexto influencia em um descritor, se a diferenciação vai se dar através de métricas ou de descritores distintos.
- É preciso determinar como será a descrição do propósito. Se esta envolve mais de um conceito deve-se pensar se é conveniente ou não evidenciar esses conceitos separadamente.
- É preciso determinar como relacionar, se for o caso, as descrições de parâmetros de entrada e saída com as do documento WSDL.
- É preciso avaliar se os descritores vão permitir que o escopo de uso de cada operação do serviço esteja explícito ou não, e, caso esteja, se vai ser de forma afirmativa ou por exceções.
- É preciso avaliar de que forma os descritores são organizados em categorias e de que forma essa organização influencia a descrição dos serviços.

O que constatamos é que é importante estar ciente dessas questões, deixando claro que elas existem, e que devem ser levadas em consideração ao se criar ou estender vocabulários de descrição de serviços¹⁰. De que forma essas questões vão ser abordadas, no entanto, fica a critério dos responsáveis pela definição dos vocabulários.

2.4. Sistematização das Características dos Serviços Web

Como observamos até aqui, diversas são as características dos serviços Web. A partir dessa constatação, evidencia-se a necessidade de uma forma de organizá-los, que nos permita a sua identificação e reuso. Para isso, fomos buscar apoio em trabalhos oriundos da área de componentes de software.

¹⁰ Idealmente, seria interessante que os próprios fornecedores também as considerassem, embora isto esteja fora do escopo desse trabalho.

Os serviços Web possuem muitos aspectos em comum com os componentes de software. De acordo com Fellner e Turowski (2003), componentes são unidades de software *autocontidas* isto é, fornecem uma função específica, e reutilizáveis, expondo sua função através de uma interface bem definida, podendo, além disso, serem comercializadas. Por interface bem definida entenda-se um canal abstrato através do qual são passados parâmetros, o serviço é invocado e uma resposta é retornada, após a sua execução. Esta definição se encaixa nas características dos serviços Web, e, então, é natural esperar que as iniciativas de organização das características dos componentes de software possam ser aproveitadas para os serviços Web.

Vários trabalhos abordam a especificação de componentes (LEACH, 1997) (JACOBSON, 1997) (FETTKE e LOOS, 2002) (HAN, 1999) (MELING et al., 2000) (FROLUND e KOISTINEN, 1998) (WOOD, SOMMERVILLE, 1998) (VARADARAJAN et al., 2002) (CRNKOVIC, 2000) (SZYPERSKI, 1997) (MOTTA et al., 1999) (NAKKRASAE e SOPHATSATHIT, 2002) (PENIX, BARAONA, ALEXANDER, 1995) (PENIX, ALEXANDER, 1998) (PODURSKI, PIERCE) (PRIETO-DIAZ, 1990) (PRIETO-DIAZ, 1991) (SAMETINGER, 1996) (YACOUB, AMMAR, MILI, 1999) (ZAREMSKI, WING, 1995) (ZAREMSKI, WING, 1997), apresentando diferentes vieses para a sua descrição. Alguns desses trabalhos propõem mecanismos de categorização.

Meling et al. (2000) propõem um ambiente para classificação e gerenciamento de componentes, onde as descrições dos componentes são feitas por um administrador e são baseadas em uma árvore de classificação. Nessa árvore, classes pré-definidas são usadas para agrupar componentes com características semelhantes. A árvore tem muitos ramos, cobrindo aspectos funcionais e não-funcionais tais como a área de negócio, preço e tipo de licença de uso (por exemplo: por uso, por mês).

Varadarajan et al. (2002) propuseram a linguagem CDML (Component Description Markup Language), baseada em XML, usada para descrever características funcionais e não-funcionais, que são divididas em quatro grupos: sintáticas, não-funcionais, comerciais e de licença de uso.

Fetke e Loos (2002) propõem uma organização mais abrangente das características de componentes, visando descrever um componente através de sete

aspectos ou níveis de especificação: marketing, tarefa, terminologia, qualidade, interação, comportamento e interface. Estes níveis cobrem aspectos funcionais, não-funcionais, de qualidade, descrevem as interfaces dos componentes e seus parâmetros, as pré e pós-condições do processamento do componente, conceitos do domínio de negócio e possíveis dependências entre componentes.

Tabela 2 - Principais aspectos dos níveis de especificação de Fettke.

| Nível | Aspectos de Especificação |
|----------------------|--|
| Interface | Serviços fornecidos e requeridos, Parâmetros, Tipos Públicos de Dados, Erros possíveis. |
| Comportamento | Pré e pós-condições. |
| Interação | Seqüência de serviços dependentes do mesmo componente e entre serviços de diferentes componentes. |
| Qualidade | Critérios de qualidade: sugere-se o padrão ISO 9126 (ISO 9126, 1991), que estabelece os itens de portabilidade e eficiência, entre outros. |
| Terminologia | Definição de conceitos do domínio do negócio e outros termos necessários, que podem ser utilizados nos outros níveis. |
| Tarefa | Tarefas do domínio do negócio que ele oferece. Ex: verificar código do banco; procurar banco. |
| Marketing | Características detalhadas da área de negócio e da organização que o fornece; requisitos técnicos e características diversas. |

A Tabela 2 resume os principais aspectos dos sete níveis propostos por Fettke.

A proposta de Fettke permite analisar com mais detalhes as diversas características dos componentes, já introduzindo aspectos de negócio, importantes para a descrição de componentes com objetivos comerciais. Entretanto, de acordo com Villela (2000), ainda não há um método de desenvolvimento baseado em componentes que contemple a captura de abstrações das características dos componentes de modo a facilitar o seu entendimento e conseqüente reutilização.

De modo a tornar mais detalhada a captura das características dos serviços Web, com foco em seu reuso e entendimento, propomos adaptar e estender a proposta de Fettke e aplicá-la à descrição dos serviços Web, criando os aspectos de descrição dos serviços Web, ilustrados nas Tabelas 3 e 4.

A adaptação é feita com base no estudo das características dos serviços Web, realizada na Seção 2.3.

Os aspectos presentes na tabela 3 se referem a características a serem descritas. Estas características foram levantadas nas seções 2.3.1 e 2.3.2. Os aspectos da tabela 4 se referem a decisões que devem ser tomadas ao se organizar os atributos dos serviços Web. Estes aspectos são explorados na Seção 2.3 e não são abordados na proposta de Fettke.

Além disso, na tabela 3, alguns itens foram acrescentados em relação à proposta de Fettke (representados hachurados), enquanto que o nível de interação de Fettke não foi considerado, pois está mais voltado para a especificação de *workflow* de composição de serviços, o qual não é o foco desse trabalho.

Tabela 3 - Aspectos de descrição dos serviços Web, voltados para descrição de atributos.

| Atributos Descritos | Aspectos de Descrição | Tipo¹¹ (F/NF) | Dep I (S/N)¹² | Dep D (S/N)¹³ |
|----------------------------|---|-------------------------------------|-------------------------------------|-------------------------------------|
| Interface Abstrata | Operações e seus parâmetros de entrada e saída. | F | N | S |
| Acesso Físico | Dados do acesso físico ao serviço, tais como protocolo, porta e mensagens. | F | S | N |
| Qualidade de Informação | Atributos de qualidade relacionados aos dados manipulados pelo serviço Web. Ex: confiabilidade, integridade de transação e segurança. | NF | S | N |
| Qualidade de Execução | Atributos de qualidade relacionados à execução do serviço Web. Ex: disponibilidade, acessibilidade e desempenho. | NF | S | N |
| Qualidade de Reputação | Atributos de qualidade relacionados à reputação do serviço, como percebida pelos usuários. | NF | S | N |
| Escopo de uso Local | Restrições associadas ao local de utilização do serviço. Ex: cobertura geográfica. | NF | N | N |
| Escopo de uso Temporal | Restrições associadas a quando o serviço pode ser usado. Ex: de segunda à sexta-feira. | NF | N | N |
| Propósito do Serviço | Áreas genéricas de atuação do serviço. | NF | N | N |
| Propósito de cada Operação | Propósito específico de cada operação do serviço. | NF | N | S |
| Recursos | Recursos utilizados pelas operações do serviço que não são passados como parâmetros. | NF | N | S |
| Pré e Pós-condições de uso | Condições que devem ser satisfeitas antes e depois da execução do serviço. Ex: cód. de acesso que autoriza o uso. | NF | N | S |
| Custo Monetário | Dados de valor e moeda cobrados pelo uso do serviço. | NF | S | N |
| Cobrança | Dados de licença de uso: Ex: por uso, por mês. | NF | S | N |
| Provedor | Dados do fornecedor do serviço Web. | NF | N | N |
| SLA - Contratos | Termos e condições que regulam a utilização do serviço Web. | NF | S | N |

¹¹ Tipo **F** é Funcional. Tipo **NF** é Não-Funcional.

¹² Indica se o atributo pode depender ou não da implementação física do serviço.

¹³ Indica se o atributo pode depender ou não do Domínio.

Tabela 4 - Aspectos de descrição dos serviços Web, voltados para a organização de atributos.

| Item de Organização | Aspectos de Descrição |
|---|--|
| Vocabulários necessários | Vocabulários necessários para descrever o serviço e suas operações. |
| Organização dos descritores em categorias. | Determinar de que forma os descritores são organizados em categorias e como se relacionam. |
| Não-Ambigüidade do Contexto dos descritores | Determinar como diferenciar conceitos semelhantes, minimizando ambigüidades na sua descrição. Ex: métricas. |
| Descrição do escopo do uso de operação | Avaliar se o escopo de uso de cada operação do serviço vai estar explícito ou não, e, caso esteja, se vai ser de forma afirmativa ou por exceções. |
| Associação de parâmetros com WSDL | Determinar como relacionar, se for o caso, as descrições de parâmetros de entrada e saída com as do documento WSDL. |
| Definição do propósito de cada operação | Definir de se o propósito vai estar fatorado ou aglutinado. |
| Categorização do uso do serviço | Determinar se o serviço é descrito como pertencendo a mais de uma categoria. |
| Diferenciação do escopo do uso | Diferenciar a descrição do uso do serviço como um todo do uso de cada operação do serviço. |

Como podemos perceber, descrever um serviço Web de forma abrangente não é tarefa trivial. Por outro lado, se nosso objetivo é que mecanismos de software possam escolher serviços Web de forma automática, é necessário que a descrição desses serviços seja criteriosa. Os aspectos de descrição de serviços Web apresentados nas Tabelas 3 e 4 se aplicam a este propósito e são utilizados mais adiante como um critério de avaliação das iniciativas de descrição de serviços.

2.5. Limitação dos Padrões de Descrição e Publicação de Serviços

Web

Considerando a análise dos aspectos de descrição dos serviços Web resumidos nas Tabelas 3 e 4, podemos observar que os padrões atuais de descrição e busca possuem limitações.

O documento WSDL descreve apenas os aspectos funcionais e de acesso dos serviços, o que não atende os propósitos da descoberta dinâmica.

Por sua vez, os mecanismos de classificação e busca de serviços no UDDI são insuficientes para apoiar buscas dinâmicas. Buscas por palavras-chave são ambíguas e os elementos de categorização de serviço, tais como seu nome, provedor, categoria do serviço, localização geográfica e até mesmo o `tModel`, são muito genéricos para

permitir identificação inequívoca de serviços. Além disso, a maior parte das características da Tabela 3 não são abordadas, não sendo possível fazer buscas por elas. Por exemplo, não há uma descrição formal do propósito das operações dos serviços, atributos de qualidade, custo, recursos etc.

CAPÍTULO 3

Iniciativas Relacionadas à Descrição de Serviços e Abordagens Teóricas de Apoio

Para minimizar as limitações dos padrões de publicação e descrição de serviços, outras iniciativas estão surgindo, procurando dar maior precisão à descoberta de serviços através de descrições mais detalhadas. Dentre elas, podemos citar: ebXML (ebXML, 2003) , UDDIe (SHAIKHALI et al., 2003), UDDI-M (MILES et al., 2003), UX (CHEN et. al., 2003), e a proposta de Ran (2003).

Algumas dessas iniciativas são apoiadas pelo modelo de descrição presente no registro de serviços. Dentre essas, a maioria estende os padrões existentes, enquanto outras propõem novas arquiteturas para publicação e descoberta. Estas abordagens, embora ofereçam mais precisão nas buscas que o padrão UDDI, ainda assim resultam em pouco detalhe para descritores, ou não fornecem documentação adequada, ou ainda necessitam consideravelmente de intervenção humana envolvida no processo de descrição e busca de serviços. De um modo geral, elas buscam viabilizar a descrição de atributos a partir de extensões de descritores genéricos, sem oferecer uma forma sistemática de proceder esta descrição. Desta forma, não cabe aqui sua avaliação.

Outras iniciativas são apoiadas por linguagens de descrição baseadas em dialetos XML. Consideramos estas como sendo as linguagens que descrevem serviços em XML, porém sem procurar estabelecer relações ou restrições de domínio entre esses elementos.

Por fim, há as iniciativas de linguagens baseadas em ontologias, que procuram estabelecer relações entre os descritores dos serviços, permitindo inferências mais ricas, embora ainda limitadas, sobre essas descrições.

Essas iniciativas, porém, não cobrem os aspectos relacionados à organização dos descritores em categorias e possuem outras limitações.

Neste capítulo, são analisadas e comparadas essas iniciativas de descrição, discutindo-se também suas limitações, especialmente no aspecto da organização dos descritores por elas propostos. Buscando preencher a lacuna das iniciativas estudadas em relação a esses aspectos de organização, analisamos ainda abordagens complementares, que tratam dessas questões.

3.1. Iniciativas de Descrição Baseadas em Dialeto XML

XML é uma linguagem utilizada para criar outras linguagens, através de elementos descritores (tags) que ajudam a descrever os dados associados a estes descritores. Ao utilizar este tipo de recurso é possível associar também um significado aos dados (COYLE, 2002). Esse significado, entretanto, é consensual, ou seja, é preciso que haja um acordo entre os usuários da linguagem sobre o que ele significa. Por exemplo, se existe um elemento chamado `título` este pode representar o título de um livro, ou o título de uma palestra, ou outra coisa qualquer dependendo do que for convencionalizado.

Entretanto, a sua portabilidade (um arquivo XML é armazenado em formato de texto) e facilidade de uso tornam o XML atrativo para criar novas linguagens baseadas em XML, ou seja, dialetos XML, a serem utilizados em diferentes áreas, dentre elas a descrição de serviços Web. Com a vantagem de que esses dialetos oferecem alguma semântica em relação a um texto simples, mesmo que seja uma semântica consensual.

É o caso do padrão de descrição WSDL. Entretanto, a linguagem WSDL tem seu foco voltado para a descrição da interface do serviço Web, não tratando atributos não funcionais do serviço. Dessa forma, outras iniciativas mais ricas de descrição de serviços Web baseadas em dialetos XML foram desenvolvidas, sendo abordadas a seguir.

3.1.1. WSEL

WSEL é uma extensão do WSFL (LEYMANN, 2001), este um dialeto XML utilizado para descrever o fluxo das interações de negócio envolvendo serviços Web. Essas interações entre parceiros de negócio têm características tais como obrigações legais entre os parceiros, custo de cada tarefa, duração máxima, aspectos de segurança e

confidenciabilidade, dentre outras, cuja definição está no escopo da linguagem WSEL. Entretanto, apenas uma descrição genérica desses atributos está disponível atualmente (LEYMANN, 2001). Contudo, como esta proposta é iniciativa de um grande fabricante, IBM, e ter sido submetida ao W3C, optamos por citá-la aqui.

3.1.2. WSOL

Web Service Offerings Language (WSOL) é um dialeto XML que permite descrever formalmente características funcionais e não funcionais dos serviços Web, tais como qualidade de serviço e direitos de acesso, onde os serviços Web são agrupados em classes (TOSIC, PATEL, PAGUREK, 2002). A linguagem é voltada para os aspectos de contratos associados às características descritas. Entretanto, como o escopo da sua descrição é amplo, consideramos como trabalho relacionado, mesmo sabendo que o foco não é a descrição do serviço para efeitos de descoberta dinâmica.

WSOL descreve um serviço Web, através dos seguintes recursos:

- Restrições (constraints): Uma restrição especifica alguma condição que deve ser satisfeita antes ou depois da execução do serviço. Podem ser pré-condições, pós-condições e condições futuras, restrições de atributos de qualidade ou de direito de acesso (restringindo o acesso a operações de um serviço). São sentenças lógicas, assumindo os valores verdadeiro ou falso;
- Afirmações (statements): Uma afirmação permite a especificação de características como atributos de qualidade, preço por subscrição, preço por uso, penalidades ou multas e responsabilidade pelo gerenciamento das afirmações. Este último descreve qual entidade é responsável pela verificação de uma restrição, grupo de restrição ou a completa oferta do serviço, para ver se estes estão sendo fornecidos como prometido. São também referenciadas como afirmações de gerenciamento (management statements).

- Grupos de restrição e modelos de grupo de restrição: São construções que permitem agrupar restrições e afirmações, permitindo reuso em ofertas de serviço. Modelos de grupo são semelhantes aos grupos de restrição, porém aceitam parâmetros.
- Oferta de serviço: São construções que permitem o agrupamento de restrições, afirmações, grupos de restrição e instâncias de grupos de restrição, constituindo-se em uma representação formal de uma única classe de serviço de um serviço Web. Por classe de serviço entende-se uma variação discreta de um serviço e da qualidade do serviço, fornecida por um serviço Web, referindo-se à mesma funcionalidade, mas diferindo em restrições e afirmações, tais como privilégios de uso, tempos de resposta ou prioridade.
- Relações dinâmicas entre ofertas de serviço: São associações entre ofertas de serviço, permitindo que serviços alternativos do mesmo provedor que pertençam à mesma classe de um determinado serviço possam ser utilizados para substituí-lo, caso este esteja indisponível por algum motivo. Esta facilidade permite que os consumidores possam dinamicamente adaptar o serviço utilizado, sem ter que substituí-lo por outro serviço, possivelmente de outro provedor.

Um exemplo da definição de uma oferta de serviço em WSOL (TOSIC, PATEL, PAGUREK, 2003) é mostrado no anexo A-2. Cabe ressaltar que nesse trabalho são apontados alguns vocabulários que precisam ser definidos e que podem ser usados para estender as descrições da linguagem.

A linguagem WSOL oferece uma forma rica de descrição de serviços Web, porém deixa em aberto a definição de atributos presentes na Tabela 3, deixando que estes sejam definidos através de descritores de extensão previstos na linguagem. Além disso, não trata os aspectos de organização da descrição, presentes na Tabela 4.

3.1.3. Restrições das Iniciativas de Descrição Baseadas em Dialeto XML

Independente do nível de detalhe das linguagens de descrições de serviços baseadas em XML, elas apresentam restrições inerentes à linguagem XML.

A linguagem XML fornece apenas interoperabilidade sintática, mas não semântica. A interoperabilidade sintática permite entendimento a partir de um dialeto comum, mas não permite que se faça deduções ou inferências a partir da comparação de características comuns. Já a interoperabilidade semântica implica que diferentes softwares são capazes de interpretar uma descrição da mesma forma (KONEN e KLAPSING, 2001), permitindo partilhar não só o dialeto comum, mas o significado. Desta forma, a interoperabilidade sintática, por exemplo, nos permite recuperar um objeto do tipo quadrado ou triângulo baseado nos atributos número de lados e figura geométrica, mas não permite inferir que quadrado e triângulo são objetos do tipo polígono, pois na interoperabilidade sintática não temos como expressar relacionamentos e restrições de domínio.

Uma limitação das especificações em XML é que um mesmo documento XML pode significar coisas distintas, dependendo do contexto de uso. De acordo com Paolucci (PAOLUCCI et al., 2002), isto é uma grande limitação para as buscas por funcionalidades de serviços (*capabilities matching*) onde é importante uma maior flexibilidade para se comparar o que se procura com o que é ofertado. Pode-se, por exemplo, estar procurando por um serviço de cotação de ações e este estar descrito como fornecedor de notícias financeiras. Nesses casos, faz-se necessário um método de descrição que permita relações entre conceitos, como por exemplo, de generalização e especialização (ARTALE et al.), e que permita também a restrição de propriedades.

Para que a descoberta dinâmica de serviços possa ser implementada com flexibilidade e acurácia é preciso ir além da interoperabilidade sintática, buscando um melhor entendimento do significado real de um descritor, ou vai-se incorrer em retornos imprecisos e indesejados.

3.2. Iniciativas de Descrição Baseadas em Ontologias

Para minimizar o problema da precisão na descrição de serviços Web, foram criadas propostas de definição de *ontologias* para serviços Web. Algumas dessas propostas vêm apoiadas por ambientes de publicação. Dessa forma, essas iniciativas são estudadas em conjunto (descrição e publicação), nos itens seguintes.

O termo *ontologia* é originário da filosofia, onde é utilizado para estudar a *natureza da existência*, ou seja, os tipos e estruturas dos objetos, suas propriedades e outros aspectos da realidade (WELTY e GUARINO, 2001).

Para a comunidade de representação do conhecimento, em uma definição dada por Gruber (1993b), ontologia é uma “especificação formal e explícita de uma conceituação compartilhada”. Ding e Foo (2002) explicam os significados de: *conceituação, explícita, formal e compartilhada*. Segundo Ding, por *conceituação* entende-se um modelo abstrato de um fenômeno no mundo, tendo-se identificado os conceitos relevantes desse fenômeno. Por *explícita* entende-se que o tipo dos conceitos usados e suas restrições estão explicitamente definidos. Por *formal* entende-se que a ontologia deve ser capaz de ser interpretada por máquinas. Por *compartilhada* entende-se que a ontologia deve capturar o conhecimento aceito por consenso pelas comunidades que dela fazem uso.

Ainda de acordo com Tim Berners-Lee, na *Web semântica*¹⁴ (BERNERS-LEE, HENDLER, LASSILA, 2001) o termo ontologia é empregado como sendo uma taxonomia e um conjunto de regras de inferência. A taxonomia é formada por classes e sub-classes e possui um conjunto de relações que relacionam as classes umas às outras. As regras normalmente são expressas como restrições sobre os domínios das classes (HENDLER, 2002).

Cabe aqui ressaltar que o significado do termo *semântica*, no contexto desse trabalho, é o entendimento comum, pelo homem e pela máquina, de termos e conceitos envolvidos na definição de um item de informação. Para que isso seja possível, existem alguns termos básicos, ou *primitivas*, empregados nesta definição, os quais têm o

¹⁴ A Web Semântica, de acordo com Berners-Lee é uma extensão da Web atual, onde é atribuído um significado bem definido às informações, permitindo uma melhor cooperação entre homens e máquinas.

contexto de seu uso implícito pelos homens que venham a utilizá-lo e pelas máquinas, não havendo a necessidade de uma definição anterior (CUI e O'BRIEN, 2000). A partir da utilização desses termos básicos, outros termos podem ser criados, ampliando o escopo da definição e permitindo expressar relacionamentos entre os termos e também restringir o escopo de sua utilização.

Para que a busca semântica seja implementada, vem se buscando o uso de ontologias. Entretanto, uma ontologia que se aplique a um domínio genérico de conhecimento é dificilmente atingível, ao menos com os recursos existentes atualmente. Para que isso fosse possível, deveria haver um consenso entre todas as comunidades sobre os diferentes conceitos existentes, seus significados, definições e relacionamentos. Dessa forma, ontologias existem normalmente associadas a uma área específica ou domínio, e, ainda assim, há várias ontologias definidas para cada domínio, cada uma expressando diferentes visões ou necessidades da comunidade envolvida em sua utilização.

Dentre estas, pode-se destacar uma ontologia voltada para a descrição de serviços Web, OWL-S (ANKOLEKAR et al., 2001), criada a partir da DAML+OIL (HARMELEN, PATEL-SCHNEIDER, HORROCKS, 2001). Esta ontologia “descreve as propriedades e funcionalidades dos serviços Web de uma forma não ambígua e compreensível pelos computadores” (ANKOLEKAR et al., 2001).

Descrevendo o serviço com uma linguagem voltada para elaboração de ontologias na Web, como por exemplo OWL, os mecanismos de busca por software podem usar recursos como hierarquia (e restrições de propriedade) para recuperar informação, de acordo com os critérios informados em uma pesquisa (BERNERS-LEE, 1998). Esses recursos possibilitam inferência do valor dos atributos, permitindo que buscas mais ricas sejam feitas.

Considere-se, como exemplo, um serviço financeiro que divulgue a cotação de ações por empresa em uma determinada data, em uma determinada bolsa de valores. Hoje, se quiséssemos localizar tal serviço pelo UDDI, teríamos que saber seu nome, ou saber a categoria de serviço em que este se encaixa. Categorias de serviço, no entanto, são muito genéricas para que uma localização mais precisa seja obtida. A utilização de uma ontologia permitiria a busca a partir de conceitos como “ações”, “empresa”, “bolsa

de valores”, e ainda aplicar restrições a eles. Por exemplo: empresa Microsoft, ou bolsa de valores de São Paulo. Esse tipo de busca é notadamente mais rico e flexível que o oferecido por uma categoria de serviços.

Existem várias propostas relacionadas à descrição de serviços baseadas em ontologias. Algumas dessas (LÉGER et al., 2000) (SOLLAZZO et al., 2002) propõem também uma arquitetura para dar subsídio à descoberta de serviços, porém elas ainda estão em seu início, ou possuem pouca documentação. Outras ainda, não descrevem com maiores detalhes os atributos não funcionais (SYCARA et al., 2002) (BENJAMINS et al., 1999) (FENSEL et al., 1999) (FENSEL, 2000a), (FENSEL, 2000b). Devido a esse fato, elas não serão comentadas com detalhes. Optamos, alternativamente, por destacar as abordagens DReggie (CHAKRABORTY et al., 2001) e OWL-S (ANKOLEKAR et al., 2001), descrevendo-as nas Seções 3.2.1 e 3.2.2, considerando a sua divulgação e relação com a descrição semântica de serviços.

3.2.1. DReggie

DReggie (Semantic Service Discovery for M-Commerce Applications) (CHAKRABORTY et al., 2001), é uma infra-estrutura para descoberta de serviços baseada em Jini (JINI, 2003), que utiliza DAML para fornecer mais semântica ao processo de descoberta. Jini é uma arquitetura orientada para serviços distribuídos criada pela SUN Microsystems e que possui um serviço de diretório com informações dinâmicas sobre serviços presentes em uma federação normalmente composta por dispositivos móveis. No projeto DReggie foi criada uma ontologia (PERICH, CHAKRABORTY, 2001), para a descrição de serviços, levando em conta os atributos do modelo de serviços Jini. Entretanto, ela é voltada para a descrição de serviços móveis.

DReggie aborda poucos dos atributos considerados na Tabela 3 e inclui outros que são específicos de serviços móveis. A ênfase nesse tipo de serviço, no entanto, torna a linguagem difícil de estender para outros mais genéricos.

Um exemplo de descrição de serviço em DReggie é mostrado no anexo A-3.

3.2.2. OWL-S

OWL-S (OWL-S, 2003) é uma ontologia que ainda continua em desenvolvimento e que faz parte do programa DAML - the DARPA Agent Markup Language (HENDLER, McGUINNESS, 2000). OWL-S é uma nova versão, escrita em OWL (Ontology Web Language), da DAML-S (ANKOLEKAR et al., 2001), e foi lançada em dezembro de 2003 como recomendação do W3C. Esta versão apresenta alguns refinamentos em relação a DAML-S e é utilizada para descrever serviços Web. Utilizando OWL-S um agente de software é capaz de interpretar a descrição do serviço e saber quais parâmetros de entrada são necessários, qual saída é gerada pelo serviço Web e como executá-lo automaticamente. OWL-S descreve os serviços em termos de seus aspectos funcionais e não funcionais, através de classes. A classe *Service* é definida no topo da ontologia e se relaciona a três outras classes: *ServiceProfile*, *ServiceModel* e *ServiceGrounding*.

A classe *ServiceProfile* descreve os *atributos* funcionais e não funcionais do serviço Web tais como: área geográfica, tipo do serviço (ex: B2B, B2C), grau de qualidade (aferido de acordo com padrões da indústria de Web services), garantias de qualidade (ex: tempo de resposta menor que 2 milisegundos), categoria do serviço, protocolo de comunicação.

A classe *ServiceModel* descreve como o serviço Web *funciona*, estando voltada para o modelo de fluxo do serviço, descrevendo seus parâmetros de entrada, as pré e pós condições de execução do serviço e também seus estados, como por exemplo: iniciando, em execução ou terminado.

A classe *ServiceGrounding*, ainda em desenvolvimento, descreve como o serviço Web pode ser *chamado*. Para isso especifica detalhes como o formato de mensagem e protocolo de transporte, assumindo um papel semelhante ao do WSDL.

OWL-S, entretanto, não descreve todos os atributos considerados na Tabela 3. Também não faz distinção entre os descritores do serviço e os das suas operações. É verdade que a linguagem também pode ser estendida para contemplar esses atributos. Por exemplo, a classe *ServiceParameters* pode ser estendida para descrever qualquer atributo de qualidade, mas uma vez que os atributos não estão previstos, essa

extensão pode se dar de muitas maneiras diferentes, gerando vocabulários não uniformes. Além disso, OWL-S também não trata os aspectos de organização de descritores apresentados na Tabela 4, e discutidos na Seção 2.3, pelo menos não de forma explícita.

Um exemplo de descrição de serviço em OWL-S é mostrado no anexo A-4.

3.2.3. Restrições das Iniciativas de Descrição Baseadas em Ontologias

Os trabalhos baseados em ontologias aqui descritos, embora apoiados em alguma forma expressiva de semântica, não apresentam de forma clara o que seria uma definição mais completa de serviços Web, sob os diferentes aspectos que estes podem ser encarados, detalhando atributos de qualidade do serviço e aspectos funcionais. Cabe ressaltar, no entanto, que dentre as propostas de descrição de serviços baseadas em ontologias o projeto OWL-S é o que mais se aproxima do propósito dessa dissertação, na medida em que define serviços Web de forma mais completa. Ainda assim, OWL-S poderia se beneficiar de um estudo mais elaborado para a construção de sua ontologia, passando a abordar de maneira menos genérica as características dos serviços Web, conforme discutido na Seção 2.3 e resumido na Tabela 3. Uma abordagem de descrição genérica sem dúvida é mais flexível, na medida em que pode ser estendida de maneira mais ampla. Entretanto, se ela é por demais genérica, isso pode ser um problema, pois cada um vai estender de uma forma diferente, gerando vocabulários redundantes e não uniformes. É preciso chegar a um meio termo entre o nível de detalhamento e a flexibilidade desejada. E é preciso fornecer alguma orientação de como se espera que a linguagem seja estendida.

3.3. Tabela Comparação entre Iniciativas de Descrição de Serviços Web

A tabela comparativa foi elaborada a partir da análise de serviços Web discutida na Seção 2.3 e resumida na Tabela 3. Outros atributos poderiam ter sido escolhidos, mas optamos pelos que, no nosso entender, fornecem os requisitos mínimos para que se possa descrever um serviço em termos de seus atributos funcionais e não funcionais, permitindo que sejam identificados pelo seu propósito e sua qualidade. Algum grau de

subjetividade, no entanto, foi adotado na escolha. Entendemos que isso é inevitável, uma vez que, citando Noy e McGuinness (2000) “a melhor solução sempre depende da aplicação que temos em mente e das extensões que podemos antecipar”.

Tabela 5 - Comparação das iniciativas de descrição de serviços Web.

| Atributos | DReggie | WSOL | DAML-S |
|----------------------------|---------|------|--------|
| Interface Abstrata | S | S | S |
| Acesso Físico | N | N | S |
| Qualidade de Informação | N | N | N |
| Qualidade de Execução | S | N | N |
| Qualidade de Reputação | N | N | N |
| Escopo de uso Local | N | N | N |
| Escopo de uso Temporal | N | N | N |
| Propósito do Serviço | N | N | S |
| Propósito de cada Operação | N | N | N |
| Recursos | S | N | N |
| Pré e Pós-condições de uso | N | S | S |
| Custo Monetário | S | S | N |
| Cobrança | N | S | N |
| Provedor | N | S | S |
| SLA - Contratos | N | S | N |

De uma maneira geral, a partir da análise da Tabela 5, podemos observar que os trabalhos analisados são complementares entre si, e que nenhum deles fornece os meios para uma descrição abrangente dos atributos de um serviço Web¹⁵.

Nenhum dos trabalhos analisados descreve o propósito nem as qualidades de reputação e de informação do serviço. Apenas WSOL trata do contexto dos descritores e apenas DReggie detalha atributos de qualidade de execução e recursos. Com exceção de WSOL, nenhum desses trabalhos considera ainda os aspectos de organização de descritores, presentes na Tabela 4, pelo menos de forma explícita, e mesmo WSOL, só considera um desses aspectos, o da escolha de vocabulários necessários.

Cabe observar que os aspectos de organização de descritores está relacionado à forma de pensá-los dentro do domínio em que estes estão inseridos. Embora os trabalhos relacionados à descrição específica de serviços Web não explorem esses aspectos, existem abordagens de outras áreas que tratam destas questões e que serão discutidas a seguir.

¹⁵ Cabe ressaltar que não utilizamos os atributos de Acesso Físico (vide Tabela 3) como itens de comparação, pois entendemos que eles são satisfatoriamente cobertos pelo padrão WSDL.

3.4. Abordagens Teóricas de Apoio à Descrição

A falta de uma abordagem sistemática de apoio para a descrição de serviços Web, faz com que as linguagens propostas apresentem problemas relacionados ao escopo dos descritores, como os discutidos na Seção 3.3 e que envolvem os aspectos apontados na Tabela 3. Além desses, existem também problemas relacionados à sua organização. Estes últimos envolvendo os aspectos apontados na Tabela 4.

Nesse sentido, a escolha dos vocabulários necessários envolve não apenas identificar os domínios de conhecimento, mas também enxergá-los sob uma ótica que facilite a sua compreensão e reuso.

A organização dos descritores em categorias, por sua vez, envolve escolher um modo de pensar essas categorias e a própria maneira de definir os descritores, uma vez que o processo de categorização está diretamente relacionado às características do objeto que se quer categorizar.

Já os aspectos que envolvem aumentar a semântica dos descritores, tais como os do contexto e da associação de parâmetros WSDL devem ser planejados criteriosamente, envolvendo outras questões relativas à semântica de ontologias, que se vai adotar.

Por fim, a descrição do propósito envolve aspectos particulares em relação à semântica, podendo ser exploradas sob a perspectiva do uso de recursos lingüísticos.

Dessa forma, essa dissertação visa propor uma abordagem sistemática para a descrição dos serviços Web, de modo a contemplar essas questões aqui levantadas. Para tanto, fomos buscar apoio em diretrizes propostas, encontrando-a na tese de Campos (CAMPOS, 2001a), que propõe requisitos para a elaboração de metodologias para modelos conceituais de hiperdocumentos. Em seu trabalho, Campos faz um estudo de diversas abordagens de organização de informação, colhendo contribuições de diversas teorias, tais como: Teoria do Conceito (DAHLBERG, 1978), Teoria da Classificação Facetada (TUDHOPE et al., 2002), Ontologia Formal (GUARINO, 1998a), dentre outras.

Partindo deste trabalho de Campos, utilizado como linha mestra, desconsideramos alguns aspectos específicos de sua proposta, voltados para

hiperdocumentos, e fomos buscar ainda outras abordagens que auxiliaram em pontos específicos voltados para o domínio de serviços Web.

Estes aspectos são explorados nas sub-seções seguintes.

3.4.1. Organização de Domínios de Conhecimento

Os domínios de conhecimento são os assuntos básicos ou áreas mais abrangentes do conhecimento, tais como Comércio Eletrônico, Finanças, Informática (CAMPOS, 2001a). Determinar quais os domínios que vão ser trabalhados é o primeiro passo para a construção de uma ontologia. Entretanto, mesmo sendo esta uma tarefa que depende do objetivo e do escopo que se deseja atingir, é a partir do reconhecimento dos domínios que vamos determinar os vocabulários que são necessários e que vão ser reutilizados ou construídos.

Falbo (1998) sugere que algum mecanismo de decomposição deve ser usado para facilitar este processo de construção. Sugere que se adote a abordagem de construção de ontologias em níveis, isto é, construir ontologias básicas, centrais para o domínio em estudo e, a partir destas, estendê-las adicionando novos elementos. Isto vai ao encontro da proposta de Guarino (1997), que propõe que ontologias sejam construídas segundo seu nível de generalidade, e para isso destaca diversas categorias de ontologia, de acordo com seu nível de generalidade: genéricas, de domínio, de tarefa, de aplicação (GUARINO, 1998a). A Figura 2 ilustra a categorização de ontologias proposta por Guarino.

Ontologias genéricas ou de *alto nível* descrevem conceitos gerais, tais como: tempo espaço, matéria, objeto, evento, ação, e são independentes de um domínio particular, ou seja, se aplicam a mais de um domínio (GUARINO, 1998b). Algumas ontologias genéricas já se acham disponíveis, tais como a *Standard Upper Ontology* (SUO WG, 2004), a *Suggested Upper Merged Ontology* (SUMO) (NILES, PEASE, 2001) e a *Cyc Upper Ontology* (CYC, 2004).

Ontologias de domínio expressam conceitos de domínios particulares, descrevendo o vocabulário relacionado a este domínio. Por exemplo, para os domínios de Biologia e Informática, poderia incluir conceitos tais como proteína e célula, ou

ainda banco de dados e estrutura de dados, respectivamente. São especializações de uma ontologia genérica.

Ontologias de tarefas expressam conceitos sobre a resolução de problemas, isto é, descrevem o vocabulário relacionado a uma atividade ou tarefa genérica, tal como, diagnose ou vendas, sendo também especializações de uma ontologia genérica.

Ontologias de aplicação descrevem conceitos dependentes de um domínio específico e de tarefa, geralmente especializando ambas as ontologias. Em nossa abordagem, por exemplo, a ontologia de serviços Web proposta é uma ontologia de aplicação.

Entretanto, a construção de ontologias em níveis, por si só, não é suficiente no processo de tratamento ou construção de uma ontologia para reuso: é preciso ter um conhecimento profundo do domínio de aplicação e ainda um método de pensar e organizar o conhecimento de forma estruturada e precisa.

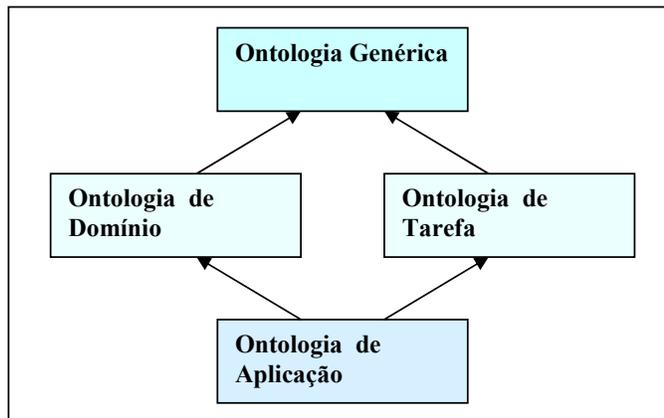


Figura 2 – Ontologias de acordo com o nível de generalidade. Adaptado de Guarino (GUARINO, 1998a).

3.4.2. Organização de Descritores

Para se organizar o conhecimento, o método de raciocínio que buscamos é o analítico-sintético. Nele se adota uma abordagem top-down (análise) e bottom-up (síntese) como forma de organização das unidades de conhecimento. As unidades de

conhecimento¹⁶ (doravante denominadas por UC) correspondem à menor unidade de informação em um domínio de conhecimento (CAMPOS, 2001a) e são utilizadas para entender o domínio.

No método de raciocínio analítico-sintético começamos com a análise, olhando para o domínio de conhecimento e procurando entender a sua natureza em classes de mais alto nível, ou seja, sabemos que existem categorias de conceitos às quais pertencem as UC, mas ainda não sabemos quais são essas categorias. A partir da escolha das UC que serão consideradas para descrever o domínio e de suas características comuns, parte-se então para a síntese das UC em uma hierarquia de categorias. O que vai determinar a escolha de um conjunto de características comuns para formar uma categoria é o enfoque que se pretende dar ao domínio, ou seja, os principais temas, ou a estrutura que desejamos ressaltar e abordar esse domínio.

Depois de categorizadas as UC, a hierarquia é expandida em classes e subclasses, estas últimas onde pertencem as UC.

Porém, é preciso considerar também os vocabulários já existentes. Para facilitar o processo de estruturar os conceitos nesses vocabulários, buscamos categorias fundamentais, a fim de organizar inicialmente as suas UC. As categorias escolhidas para isso são as propostas por Ranganathan (1967).

Ranganathan elaborou a Teoria da Classificação Facetada na qual apresenta princípios para a organização de conceitos hierarquicamente estruturados (CAMPOS, 1995). A Classificação Facetada parte da identificação dos domínios de conhecimento que se quer representar. A partir daí, dentro de cada domínio, este é analisado e seus elementos básicos identificados e agrupados em classes de acordo com características comuns. Essas classes são chamadas de facetas. Cada faceta, por sua vez, pertence a uma categoria fundamental. As categorias propostas por Ranganathan são referenciadas pelo acrônimo PMEST (*Personality, Mater, Energy, Space, Time*): Personalidade, Matéria, Energia, Espaço e Tempo.

¹⁶ A autora adota para Unidades do Conhecimento a definição de Dahlberg, que formulou a Teoria do Conceito, que propõe que o conceito seja definido como “unidades do conhecimento”, pois conhecimento pressupõe um entendimento mais objetivo de algo observável (CAMPOS apud DAHLBERG, 2001a).

De acordo com Ranganathan, *Matéria* se refere à manifestação intrínseca de materiais e propriedades do conteúdo sendo descrito (CAMPOS APUD RANGANATHAN, 2001a, p.60).

Entretanto, para Ranganathan, as propriedades não fazem parte de uma categoria, sendo consideradas uma manifestação de uma das categorias¹⁷ (CAMPOS, 2001a, p. 131). Assim, por exemplo, a cor de uma cadeira não pertence à categoria de Matéria, como uma manifestação isolada, mas ocorre ligada a uma entidade, no caso, a cadeira.

Energia se refere aos seus processos, *Espaço* a aspectos de sua localização, *Tempo* aos seus atributos temporais, e, finalmente, *Personalidade* às entidades relacionadas, suas partes ou o que não pertence às outras categorias.

Para facilitar o processo de estruturar os relacionamentos, utilizamos categorias básicas de relações, para organizá-las de forma genérica. Para isso buscamos apoio na Teoria da Terminologia, que fundamenta o padrão ISO-704 (1987). Aqui as relações entre conceitos são desmembradas em *relações lógicas* e *relações ontológicas*. As lógicas se dividem ainda em *relações de comparação* e de *combinação*, e as ontológicas, por sua vez, se dividem em relações de *contato* e de *causalidade*. Essas categorias cobrem diversos tipos de relações de *agregação* e *especialização* e também outros tipos tais como relações que constituem conceitos compostos, por exemplo: indo-europeu, analista-programador, e ainda relações de *parentesco* (várias), de *instrumento e seu uso*, de *encadeamento*, dentre outras. Para complementar esse conjunto de relações, consideramos ainda a de *equivalência*. Esta é utilizada não para conceitos, mas para especificar diferentes formas de se denominar um mesmo conceito, como os sinônimos em um dicionário (CAMPOS, 2001a).

Cabe ressaltar que para estabelecer as relações entre as UC, vamos procurar pensar essas relações a partir das categorias de relações da Teoria da Terminologia, mas as relações estão ligadas de forma muito próxima às categorias fundamentais às quais pertencem às UC. Por exemplo, UC pertencentes a categorias diferentes não podem nunca ter uma relação de *gênero-espécie*, *todo-parte* ou *equivalência*, minimizando a

¹⁷ Já Dahlberg, no âmbito da Ciência da Informação, atribui às propriedades o caráter de categoria.

ocorrência de erros lógicos no estabelecimento de relações entre classes (CAMPOS, 2001b).

3.4.3. Definição de Descritores

De acordo com a proposta de Campos, a definição das UC envolve a associação de um rótulo a cada uma delas, que denominamos daqui por diante como *termo*. Para cada definição de UC devem ainda ser esgotadas todas as suas características e propriedades, de forma que uma UC só possa ser organizada em uma classe e não em outra, dentro de um determinado contexto ou domínio. Deve-se proporcionar assim um entendimento inequívoco do que é o conceito, para que se saiba e quais são seus elementos constitutivos (CAMPOS, 2001b).

Para isso, o conceito é formado através de uma série de afirmativas, feitas passo a passo, de forma a ilustrar as características do conceito. Tanto de forma geral, mostrando o que tem em comum com outros de sua classe, como também de forma individual, ressaltando o que o diferencia de outros de sua classe.

Além da definição do conceito, há a definição do termo que o representa. Para isso, pode-se utilizar a abordagem da lingüística, através do uso de palavras fatoradas. Esta abordagem, no entanto, torna a semântica menos precisa, pois as soluções ficam apenas no plano da língua e não do verdadeiro significado do conceito. Desta forma, é importante ressaltar que, embora incentivemos em nossa proposta o uso de termos fatorados, estes devem ser utilizados com cautela. Termos devem ser fatorados sempre que seus elementos constituírem-se em termos com significado próprio dentro do contexto a que pertencem, sem dar margem a metáforas. Desta forma, aliamos flexibilidade, permitindo a junção de conceitos para formar uma nova descrição, à coerência do significado de cada conceito, que é assim inequivocamente conhecido pelos que dele se utilizam.

3.4.4. Outras Questões relativas à Semântica de Ontologias

Diversos trabalhos abordam questões de modelagem e semântica em ontologias (GRUBER, 1993a) (GUARINO, 1992) (GUARINO, 1997) (NOY e MCGUINNESS, 2000). Embora não seja o foco deste trabalho aprofundar tais questões, entendemos que,

dentro do contexto da descoberta dinâmica, este é um ponto de especial atenção devido à complexidade de seus descritores e devido ao fato de as buscas estarem implementadas de forma automatizada, com o mínimo de intervenção humana.

Partindo dessas premissas, buscamos compreender as melhores práticas na construção de ontologias, ressaltando um aspecto em particular, chamado de *nível de compromisso ontológico* (GRUBER, 1993a). Através do critério do compromisso ontológico (*ontological commitment*), Gruber aponta que uma ontologia deve fazer o mínimo possível de imposições sobre o mundo que está sendo modelado, permitindo que utilizadores da ontologia fiquem livres para especializar e instanciar a ontologia de acordo com seus propósitos. Isso implica em se modelar a ontologia de uma forma bem genérica. Entretanto, como o próprio Gruber ressaltava, deve-se ter cuidado com a clareza, ou seja, minimizar ambigüidades na definição de conceitos. Mas como conciliar uma ontologia genérica com a clareza? De acordo com Gruber, deve-se adotar um escopo genérico quanto à granularidade dos conceitos, permitindo que sejam mais facilmente estendidos, mas não em relação ao seu formalismo. Este, deve estar bem explícito. Desta forma, no domínio dos serviços Web, reiteramos a importância de se *pensar*¹⁸ a forma de descrever a não-ambigüidade do contexto dos descritores, o escopo do uso de cada operação e a associação de parâmetros com WSDL, de forma genérica, mas precisa, minimizando ambigüidades. E, de uma maneira geral, buscar o respaldo do uso das melhores práticas em construção de ontologias.

3.4.5. Descrição do Propósito do Serviço Web

Para descrever o propósito do serviço Web, buscamos apoio em Mizoguchi, (MIZOGUCHI, VANWELKENHUYSEN, IKEDA, 1995). De acordo com Mizoguchi, quando temos um processo de resolução de problemas que envolve buscas através de linguagem natural, por exemplo, uma ontologia de tarefa pode ser utilizada com um sistema de vocabulário semântico para representar o significado da sentença. Em nosso caso, como o propósito do serviço Web está relacionado a um processo, aproveitamos a proposta de Mizoguchi para que o propósito possa ser descrito a partir de conceitos de uma ontologia de tarefa, da seguinte forma:

¹⁸ Cabe ressaltar que não é o foco deste trabalho o estudo de linguagens de expressão do formalismo em ontologias.

- Nomes genéricos representando objetos que possuem um papel no processo de resolução do problema.
- Verbos genéricos representando atividades unitárias no processo de resolução do problema.
- Adjetivos que representam características dos objetos.

Outros trabalhos na área de Lingüística (IKEDA et al., 1998) (BINKERT, 2003) (HIRST, 2004), também exploram o uso de estruturas de linguagem na representação da semântica de descrições de texto, porém está fora do escopo deste trabalho aprofundar essas questões. Cabe entretanto lembrar que, apesar de explorarmos características lingüísticas para representar o propósito, consideramos como pré-requisito que os nomes, verbos e adjetivos citados acima sejam utilizados como conceitos, com significado inequívoco dentro do vocabulário a que pertencem, e não apenas como construtos de linguagem.

Por fim, para concluir, após os estudos das iniciativas de descrição de serviços Web e das abordagens complementares para apoio à sua descrição, a fundamentação teórica que levantamos nos remete a alguns questionamentos.

Podemos utilizar os atributos resumidos na Tabela 3 para estender linguagens já existentes ou para propor uma nova, que contemple todos esses atributos e que permita que estes sejam estendidos, prevendo necessidades específicas que possam vir a surgir. Entretanto cabe refletir: como exatamente definir essa linguagem e seus vocabulários? E ainda, como aproveitar vocabulários já existentes, incentivando seu reuso e poupando esforços de criação de novos vocabulários, muitas vezes conflitantes? Essas questões serão discutidas em detalhes no capítulo seguinte.

CAPÍTULO 4

Diretrizes de Descrição de Serviços Web para fins de Descoberta Dinâmica

A descrição detalhada e precisa de um serviço Web é um requisito importante para a sua descoberta dinâmica, uma vez que permite que se conheça com precisão as suas características, e, portanto, se possam fazer buscas com mais chances de sucesso em relação à intenção original da pesquisa. Por isso estudamos as características e as iniciativas de descrição dos serviços Web nos Capítulos 2 e 3. Esse estudo é a base para que se possa ter uma visão geral dos aspectos que podem estar envolvidos no uso dos serviços e, a partir dessa visão geral, criar ou estender vocabulários de descrição para eles, utilizando os aspectos que sejam de interesse.

Entretanto, é importante que as características sejam descritas através de uma abordagem sistemática, de forma a tirar o maior proveito de seu uso quando da busca e recuperação dos serviços Web.

Cabe observar que o conceito de abordagem sistemática aqui foi inspirado na definição de Bjorner (1987) no contexto da engenharia de software. De acordo com Bjorner, uma abordagem sistemática prevê algumas regras pré-estabelecidas associadas a cada fase do desenvolvimento do software. Estas regras norteiam a passagem de uma fase para outra no processo de desenvolvimento. Embora essas regras não possuam necessariamente um formalismo nem se apresentem provas formais de sua eficácia, entendemos que esta abordagem é um avanço em relação à abordagem informal, onde existem poucas ou nenhuma regra pré-estabelecida. Este avanço se traduz em algumas vantagens, tais como: definição clara dos descritores de serviço, documentação do processo de extensão destes, maior uniformidade dos vocabulários estendidos e conseqüente incentivo ao reuso dos mesmos.

Assim, neste capítulo vamos propor uma possível abordagem sistemática para a descrição de serviços Web. Esta abordagem visa especificar como devem ser descritos

os serviços Web, para que depois essa especificação possa servir de base para a elaboração de uma ontologia de descrição de serviços Web, ou para a extensão de uma já existente.

Nossa abordagem consiste em um conjunto de seis etapas, cujo objetivo final é cobrir as características levantadas nas Tabelas 3 e 4. Após uma visão geral, ilustrada na Figura 2, detalhamos cada etapa, explicando do seu uso, sua finalidade em relação às Tabelas 3 e 4, e ainda os artefatos¹⁹ produzidos. Cabe observar que nossa abordagem é voltada para a especificação dos descritores e não tem o objetivo de cobrir todos os aspectos envolvidos na elaboração de uma ontologia de serviços. Estes aspectos tratam de questões sobre linguagens e ferramentas escolhidas para representar e validar os descritores, e ainda a sua compatibilização com outras ontologias, o que está fora do escopo deste trabalho.

Por fim, após apresentar a abordagem, discutimos uma forma de documentar o que vai sendo definido em cada uma de suas etapas. Nosso propósito, com isso, é incentivar a divulgação de todo o processo de construção da descrição dos serviços, que culmina com a elaboração ou extensão de uma ontologia. Esta ontologia pode ser reutilizada, estendida e facilmente compreendida pelos que dela venham a se utilizar.

4.1. Diretrizes para a Descrição de Serviços Web

A proposta é formada por quatro etapas. Cada etapa das diretrizes envolve uma série de atividades, conforme ilustrado na Figura 3.

A etapa 1 (E1 - Definição do Escopo e Modo de Organizar os Domínios) envolve as seguintes atividades: determinar os domínios do conhecimento, os vocabulários a serem reutilizados, o método de raciocínio, as categorias fundamentais e relações básicas a serem utilizadas e tratar os vocabulários a reutilizar.

A etapa 2 (E2 - Definição das UC e suas Relações) envolve as atividades de definir as UCs, seu nível de compromisso ontológico e suas relações.

A etapa 3 (E3 – Detalhamento das UC), envolve definir: o contexto, a associação de parâmetros, o propósito e a categorização do uso.

¹⁹ Por artefatos consideramos não só documentos tangíveis, como por exemplo diagramas, como também decisões sobre como proceder em relação à construção dos descritores.

Na etapa 4 (Revisão das Definições) devem-se rever as categorias, definições e relacionamentos, pois o processo é evolutivo e não linear. Ao longo do processo, documenta-se o que é produzido.

Cabe lembrar que um vocabulário de descrição de serviços não é monolítico. Ele faz uso de outros vocabulários, que eventualmente vão ser estendidos. Dessa forma, algumas atividades da etapa 3 das diretrizes se aplicam a todos os vocabulários utilizados, e não apenas especificamente ao de serviços Web. Por outro lado, outras atividades da etapa 3 dizem respeito apenas a aspectos ligados aos serviços Web e não se aplicam aos outros vocabulários.

Por fim, é interessante notar que as etapas cobrem os aspectos levantados nas Tabelas 3 e 4. A Tabela 6 ilustra esse relacionamento.

Tabela 6 – Etapas X Aspectos de Descrição Cobertos

| Etapa | Aspectos das Tabelas 3 e 4 |
|---|--|
| 1 - Definição do Escopo e Modo de Organizar os Domínios | Escolha / Vocabulários Necessários, Organização de Descritores da Tabela 4. |
| 2 - Definição das UC e suas Relações | Descrição de atributos, da Tabela 3. |
| 3 - Detalhamento das UC | Não ambigüidade do contexto dos descritores e a Descrição do escopo do uso de cada operação, da Tabela 4. |
| 4 – Revisão das Definições | Descrição de atributos, da Tabela 3, Organização de Descritores, Não ambigüidade do contexto dos descritores e a Descrição do escopo do uso de cada operação, da Tabela 4. |

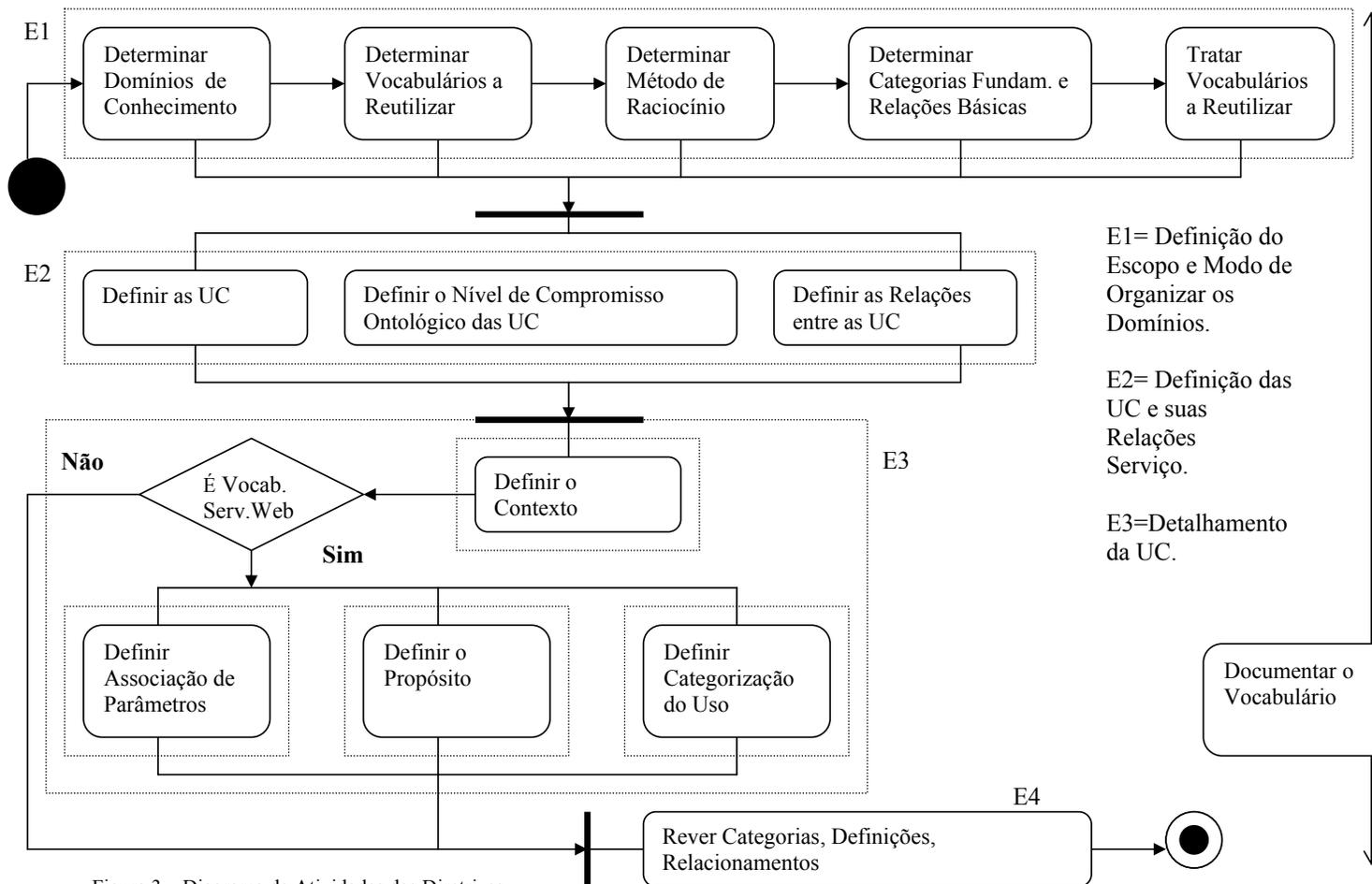


Figura 3 – Diagrama de Atividades das Diretrizes

4.1.1. Etapa 1 - Definição do Escopo e Modo de Organizar os Domínios

Essa etapa atende os pré-requisitos básicos que são necessários ao longo do processo de extensão ou criação de vocabulários. Suas atividades são descritas a seguir.

4.1.1.1. Determinação dos Domínios e Sub-Domínios de Conhecimento

Nesta atividade começamos com a análise dos domínios a serem considerados, serviços Web e outros, escolhendo as UC que vamos considerar, de acordo com o nível de detalhe e o enfoque com que desejamos abordar o domínio. Os sub-domínios, de acordo com nossa proposta, são sub-áreas de conhecimento dentro de um domínio. Por exemplo, o domínio de Biologia possui o sub-domínio de Biologia Molecular.

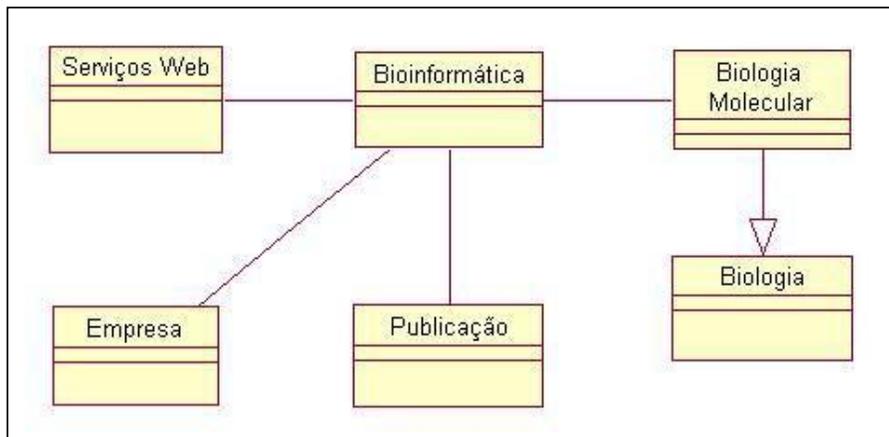


Figura 4 - Diagrama de classes (exemplo) para serviços Web de Bioinformática

Nesta atividade se busca chegar a um acordo sobre o corte do nível de detalhe que é dado aos descritores. Através da determinação de quais domínios e sub-domínios estamos considerando, é possível compreender melhor o escopo desses descritores, fazendo com que se estabeleça um entendimento sobre como o assunto deve ser abordado e quais os seus pontos de interseção com outras áreas. Depois de obtido o consenso sobre os domínios e sub-domínios desejados, estes são representados através de um diagrama de classes. A Figura 4 ilustra um possível diagrama com esta representação.

Este tipo de entendimento dos domínios e sub-domínios é de grande importância ao escolhermos os vocabulários já existentes e que podem ser reutilizados, como veremos na Seção 4.2.1.2, pois vai nos permitir identificar melhor que tipo de descritores precisamos, contribuindo, desta forma, para a escolha de vocabulários com o nível mais adequado de detalhamento.

Como artefatos produzidos nessa atividade tem-se: um diagrama de classes em UML contendo a representação gráfica dos domínios e sub-domínios de aplicação envolvidos.

4.1.1.2. Determinação dos Vocabulários a serem Reutilizados

Um serviço Web pode ser descrito de várias formas: sem reuso de outros vocabulários, com reuso e estendendo os vocabulários reutilizados, ou ainda incorporando os descritores de outros vocabulários dentro da linguagem de descrição dos serviços, descaracterizando-se de onde estes são aproveitados e, assim, desconsiderando eventuais evoluções destes vocabulários. Como nosso objetivo é incentivar o reuso, aconselhamos que não se utilize a incorporação dos vocabulários, sempre que possível.

Para a escolha dos vocabulários deve-se utilizar como base o diagrama de classes representando os domínios, descrito na Seção 4.2.1.1, e a classificação de ontologias de Guarino (1997), discutida na Seção 3.4.1.

Deve-se então pesquisar na Web por vocabulários existentes que possam ser reutilizados. Naturalmente, no estágio atual da Web semântica, essa tarefa não é muito fácil. Não pelo excesso, mas antes pela falta de vocabulários disponíveis e de divulgação dos existentes. No caso desses vocabulários desejados não existirem, devemos defini-los separadamente e utilizá-los como vocabulários auxiliares, colocando-os disponíveis para outros, e não incorporá-los à linguagem que está sendo definida.

Devemos então nessa atividade não só escolher os vocabulários auxiliares mas ainda decidir de que maneira vamos utilizar cada um deles. Dessa forma, como artefatos produzidos, tem-se:

- A identificação dos vocabulários utilizados e se estes vão ser criados, estendidos, incorporados ou apenas referenciados dentro da linguagem.
- A atualização do diagrama de classes dos domínios produzido na Seção 4.2.1.1, para que este incorpore estereótipos, indicando qual o tipo do vocabulário (tarefa, domínio ou genérico) e de que forma o domínio ou sub-domínio é utilizado (criado, estendido, incorporado ou referenciado). Por exemplo: “tarefa-criado” ou “domínio-estendido”.

4.1.1.3. Determinação do Método de Raciocínio

Nessa atividade deve-se escolher o método de raciocínio utilizado para representar as UC dos serviços Web. O método de raciocínio sugerido é o analítico-sintético. Este método de raciocínio fornece-nos uma diretriz para organizar os termos de vocabulários novos em categorias. [Para vocabulários já existentes, deve-se buscar apoio nas categorias fundamentais a serem definidas na atividade seguinte.](#)

4.1.1.4. Determinação das Categorias Fundamentais e Relações Básicas

Em nossa proposta, apesar de as categorias não serem fixas, utilizamos como base as categorias fundamentais propostas por Ranganathan [para os vocabulários já existentes, de modo a poder enxergá-los de forma mais uniforme.](#) Porém, vamos considerar que a categoria de Matéria, proposta por Ranganathan, no escopo deste trabalho, passa a agrupar também as propriedades que Ranganathan deixa associadas às outras categorias, tais como cor, ou qualidades de um modo geral (vide Seção 3.4.2).

Além disso, alguns dos nomes originais das categorias de Ranganathan, são aqui adaptados para os seguintes: Personalidade passa a ser Objeto; Matéria passa a ser Característica; Energia passa a ser Processo; Tempo e Espaço são mantidos como no original.

Caso seja necessário, pode-se ainda criar novas categorias desmembrando-se as categorias fundamentais. Dessa forma, doravante, as categorias de Ranganathan, que adaptamos, são referenciadas como *categorias fundamentais de Ranganathan adaptadas*.

No que diz respeito às relações básicas, considerando os estudos apresentados na Seção 4.1.3, nossa proposta sugere que se adotem algumas relações padrão, adaptadas da Teoria Geral da Terminologia²⁰, e que estas sejam estendidas de acordo com as necessidades observadas para a descrição de serviços Web.

Estas relações padrão, que chamamos de *relações básicas* podem ainda possuir restrições associadas, de forma que uma determinada relação se aplique apenas entre UC de determinadas categorias fundamentais, como por exemplo, as de *Objeto* e *Processo*; ou ainda, mais particularmente, entre UC de duas classes específicas. Assim, estamos minimizando ainda mais a ocorrência de erros lógicos nos relacionamentos e, de maneira análoga às categorias fundamentais das UC, estamos facilitando a identificação de uma relação e, portanto, incentivando o seu uso de forma mais consistente.

Como artefatos produzidos nessa atividade tem-se:

- A relação das categorias fundamentais escolhidas e a descrição de como são usadas para agrupar os conceitos.
- A relação das categorias de relações escolhidas e a descrição de como são usadas para relacionar os conceitos.

4.1.1.5. Tratamento dos Vocabulários a serem Reutilizados

Nesta atividade vamos tratar cada vocabulário a ser reutilizado²¹, de forma que os seus conceitos possam ser bem entendidos sob a perspectiva das categorias fundamentais básicas escolhidas. Para isso, deve-se fazer, dentro do possível, uma correspondência entre cada conceito e uma (e somente uma) categoria fundamental básica. Esse tratamento facilita o reuso dos conceitos uma vez que, ao organizá-los nas categorias fundamentais, já eliminamos uma série de reutilizações inválidas. Por

²⁰ Cabe observar, no entanto, que, para simplificar, não ilustramos as relações lógicas, da Terminologia, usadas por exemplo para especializar relações do tipo é um, ou para relacionar conceitos semelhantes, tais como: avião/navio.

²¹ Cabe ressaltar que ao considerarmos uma ontologia para reuso, não precisamos trabalhar com ela por inteiro. Existem ferramentas (CRUBÉZY et al., 2004) que permitem fazer um corte da ontologia desejada, extraindo apenas um sub-conjunto de descritores relacionados.

exemplo, especializações entre conceitos de categorias distintas. Além disso, ao organizá-los, estamos contribuindo para o seu melhor entendimento.

Para facilitar o processo de mapeamento dos conceitos em categorias, sugerimos que estes sejam mapeados com base nas categorias fundamentais de Ranganathan adaptadas. Entretanto, pode ocorrer que o vocabulário utilizado já esteja organizado segundo outras categorias fundamentais, ou que julgemos conveniente utilizar outras. Neste caso, devemos informar a correspondência entre as categorias fundamentais utilizadas para agrupar os conceitos e as fundamentais básicas.

Como artefatos produzidos tem-se:

- A organização dos conceitos de cada vocabulário em categorias fundamentais.
- Uma descrição das categorias fundamentais utilizadas em cada vocabulário e sua correspondência com as categorias fundamentais utilizadas como base.

4.1.2. Etapa 2 - Definição das UC e suas Relações

Esta etapa é o núcleo do processo de extensão ou criação de um vocabulário, determinando o seu conteúdo. Suas atividades são comentadas a seguir.

4.1.2.1. Definição das UC

A partir da definição dos conceitos, agrupamos em classes e, se for necessário, em subclasses²², as UC com atributos comuns. Por fim, enquadramos preferencialmente essas subclasses e classes nas categorias fundamentais de Ranganathan adaptadas, que, por sua vez, podem ser estendidas de acordo com nossa necessidade, formando uma hierarquia de conceitos.

Cabe aqui ressaltar que utilizamos o método de raciocínio analítico-sintético, e que, desta forma, as categorias fundamentais nascem da análise das características comuns das classes que englobam as UC, e, portanto, não são conhecidas de antemão. Entretanto, achamos conveniente que, antes de criar categorias, verifiquemos se as

²² Essa hierarquia de classe e subclasse pode se repetir por vários níveis, de acordo com a necessidade.

categorias fundamentais de Ranganathan adaptadas são suficientes, ou se podem ser estendidas, o que tornaria os vocabulários criados mais homogêneos e fáceis de mapear.

Definir um conceito, no entanto, não é uma tarefa trivial. Dizer *o que é* um objeto depende das perspectivas sob as quais desejamos enxergá-lo. Por exemplo, ao definir *alinhamento local*, podemos dizer que ele é *método de alinhamento de cadeias de caracteres usado para alinhar seqüências biológicas*. E ainda, de acordo com o método, o alinhamento pode ser local ou global e também múltiplo ou pareado; alinhamento local é portanto um tipo de método de alinhamento.

Da mesma forma, ao definir para que serve um conceito, devemos caracterizar seu propósito de acordo com as características que julgamos relevantes para o domínio, da forma como se pretende enxergá-lo. Por exemplo, um serviço pode servir para *alinhamento local de seqüências de proteínas*, mas também pode ser definido como servindo para *alinhar seqüências de proteínas com resultados mais amplos, menos precisos e mais rápidos*.

Diversos trabalhos exploram métodos de se definir conceitos a partir de suas características (DAHLGBERG, 1978), (GOMES, 1990), (CAMPOS, 2001a), (CAMPOS, 2001b), mas está fora do escopo desta dissertação aprofundar essa questão. O que queremos ressaltar é que existe a necessidade de se estabelecer um modo de se definir os conceitos em ontologias, de forma que essa definição siga alguns critérios. Esta questão já está sendo tratada por Garcia (2004), em sua tese de doutorado.

Os critérios aqui sugeridos, de maneira simplificada, são: definir as UC, e associar um termo a cada uma, procurando determinar o que é, para que serve e quais as partes de cada UC (CAMPOS, 2001a), tendo o cuidado de caracterizá-las de forma inequívoca. Para melhor compreensão dessa etapa, vamos exemplificar de que forma organizamos os atributos de qualidade dos serviços Web.

Vamos supor que desejamos criar uma linguagem com os seguintes atributos de qualidade: Acessibilidade, Confiabilidade, Disponibilidade, Granularidade da Resposta, Integridade, Privacidade, Reputação, Segurança e Tempo de Resposta. Uma vez identificadas, passamos a definir as unidades selecionadas. Para isso devemos descrever o que são, para que servem e quais são as suas partes, se for o caso. A Tabela 7, abaixo, ilustra as definições das UC referentes aos atributos de qualidade acima.

Tabela 7 - Exemplos de Definição de UC

| Unidade de Conhecimento | O que é, para que serve / suas partes |
|--------------------------------|---|
| Confiabilidade | É uma medida de qualidade da execução de um serviço Web que informa o número de falhas na execução em um dado período de tempo. Serve para medir a qualidade da execução de um serviço em relação à correção dos dados informados. |
| Disponibilidade | É uma medida de qualidade da execução de um serviço Web que informa a probabilidade de o serviço estar disponível e pronto para execução imediata. Serve para medir a qualidade da execução de um serviço em relação à disponibilidade. |
| Tempo de Resposta | É uma medida de qualidade da execução de um serviço Web que informa o tempo gasto entre o envio de uma requisição e o recebimento da resposta (tempo de latência). Serve para medir a qualidade da execução de um serviço em relação ao seu desempenho. |
| Granularidade da Resposta | É uma medida de qualidade da informação de um serviço Web que informa a quantidade média de dados a serem transmitidos como resposta a uma solicitação de execução de uma determinada operação deste serviço. Serve para medir a qualidade da informação de um serviço em relação à quantidade de dados que trafegam pela rede. |
| Integridade | É um indicador de qualidade da informação de um serviço Web que informa como o serviço garante o tratamento de transações. Serve para indicar a qualidade da informação de um serviço em relação à sua integridade. |
| Reputação | É uma medida de qualidade da reputação de um serviço Web que informa o valor médio da reputação deste serviço de acordo com critérios estabelecidos por uma entidade aferidora. Serve para medir a qualidade da reputação de um serviço como esta é percebida por terceiros. |

A análise das definições das UC permite-nos perceber diferentes facetas da qualidade dos serviços: a qualidade da informação, a qualidade da execução e a qualidade da reputação. Podemos então criar facetas para agrupar as UC de acordo com essa perspectiva. Por fim, devemos associar essas facetas às categorias fundamentais que elas pertencem. No caso, todas elas pertencem à categoria de *Característica*. Seja por exemplo a UC denominada pelo termo *Integridade*. Ela estaria, por exemplo, organizada de acordo com a seguinte hierarquia:

Categoria fundamental: *Característica*.

- o Classe: *qualidade*
 - Subclasse: *qualidade de informação*
 - *Integridade*²³
 - Subclasse: *qualidade de execução*
 - Subclasse: *qualidade de reputação*

A análise dos outros descritores de serviços Web vai eventualmente revelar outras categorias fundamentais que podem ser consideradas, de acordo com o enfoque que desejamos ter do domínio. Por exemplo, destacamos a categoria de *Instituição* como importante o desmembramento da categoria fundamental de *Objeto*. Esta categoria fundamental é útil para representar aspectos específicos relacionados ao provedor do serviço, ou de outras entidades que possam estar relacionadas a um serviço Web, como por exemplo, autoridades aferidoras de reputação, provedores de recursos e assim por diante. Nossa proposta é que ela seja sempre usada, além das categorias fundamentais de Ranganathan adaptadas.

É importante também lembrar que o processo de definição das UC pode se aplicar aos vocabulários auxiliares discutidos na Seção 4.2.1.2, sempre que estes forem estendidos ou criados.

Como artefatos produzidos nesta atividade tem-se:

- A lista das UC utilizadas na ontologia e sua definição precisa.

4.1.2.2. Determinação do Nível de Compromisso Ontológico das UC

Nessa etapa devem-se ponderar quais os conceitos presentes nas definições das UC que vão estar explícitos na nossa ontologia de serviços Web e quais vão estar implícitos, e ainda que nível de granularidade desejamos oferecer ao detalhar um conceito.

Na construção de uma ontologia para descrição de serviços Web, podemos fazer uso de diversas outras ontologias, como vimos na Seção 4.1.2.

²³ Optamos, para simplificar, representar apenas uma UC.

Nossa abordagem é que a *fatoração criteriosa* dos termos da ontologia de serviços Web nos ajuda a perceber onde se encaixam essas outras ontologias, estimulando o seu reuso e tornando a descrição de serviços mais fácil e flexível. Por fatoração criteriosa entendemos uma análise cuidadosa da definição de um conceito, procurando identificar se na definição deste há outros conceitos embutidos. Caso haja, nossa proposta é que os outros conceitos entrem como conceitos associados através de uma relação, sempre que possível, evitando assim a aglutinação de conceitos. Esta questão pode ser evidenciada, por exemplo, ao se estruturar o conceito de “alinhamento de seqüências de proteínas”.

Uma possível forma de estruturar esse conceito é partir da categoria fundamental de *Processo*, uma vez que é um processo, e organizá-lo hierarquicamente, por exemplo como: alinhamento / alinhamento de seqüências de proteínas. Mas o que é um alinhamento de seqüências de proteínas? Pode ser um conceito implícito e, desta forma, sem nenhum formalismo que não seja o de sabermos que é um tipo de alinhamento. Neste caso, os utilizadores da ontologia devem saber de antemão o que este conceito significa e que ele está relacionado com seqüências de proteínas, pois isso está implícito.

Por outro lado, podemos minimizar a informalidade definindo alinhamento de seqüências de proteínas como sendo um tipo de alinhamento que está relacionado a seqüências de proteínas, destacando o conceito de *seqüência de proteínas* além do conceito de *alinhamento*. Desta forma, além de podermos fazer inferências mais ricas sobre esse tipo de alinhamento, também estamos promovendo o reuso do conceito *seqüência de proteínas*, que pode estar relacionado a outros conceitos. Naturalmente, para determinar se este fatoramento é conveniente devemos olhar para o domínio que estamos descrevendo e analisar até que ponto o fatoramento traz vantagens. Se não houver reuso de conceitos, nem se vislumbre essa possibilidade em extensões da ontologia, então o aglutinamento de conceitos pode ser uma alternativa mais simples.

A importância dessa fase se desdobra em duas: determinar o nível de compromisso ontológico e justificá-lo, de modo que outros possam compreender o

porquê da solução adotada e, ao reutilizar a ontologia, fazê-lo de forma coerente com a sua forma de construção.

4.1.2.3. Estabelecimento das Relações entre as UC

As principais relações são propostas como um conjunto de práticas recomendadas, conforme ilustrado na Tabela 8. Outras relações que não estejam previstas pelas práticas recomendadas podem ser acrescentadas de acordo com a necessidade, sempre que possível sendo respaldadas pelos estudos apontados na Seção 4.1.3. Entretanto, não está no escopo desse trabalho um estudo aprofundado de relações entre UC, nem julgamos que seja possível prever todas as situações possíveis de relacionamentos entre elas.

Tabela 8 - Práticas Recomendadas para Relações de Serviços Web

| Objetivo | Associação entre Categorias | Natureza da Relação | Nome da Relação |
|---|---|--|--------------------------|
| Associar uma qualidade genérica a um descritor. | Associa uma UC da categoria fundamental de Característica a uma UC de qualquer outra categoria. | Qualificação/ /Característica /Adjetivação | hasObjectQualifier. |
| Associar uma qualidade não funcional (informação, execução, custo) a uma operação do serviço. | Associa uma UC da categoria fundamental de Característica a uma UC de qualquer outra categoria. | Qualificação/ /Característica /Qualidade | hasQualifierAspect |
| Associar uma qualidade de medida a um descritor. | Associa uma UC da categoria fundamental de Característica a uma UC de qualquer outra categoria. | Qualificação/ /Característica /Medida | hasMetrics, hasValue. |
| Associar as partes de um descritor ²⁴ | Associa uma UC da categoria fundamental de Objeto a uma UC dessa mesma categoria. | Ontológica/Contato /Coordenação Ontológica /Subordinação Partitiva | hasPart. |
| Associar a quem pertence uma parte de um descritor. | Associa uma UC da categoria fundamental de Objeto a uma UC dessa mesma categoria. | Ontológica/Contato /Coordenação Ontológica /Subordinação Partitiva | isPartOf. |

²⁴ Partes de descritores são também descritores. Por exemplo: os descritores núcleo e citoplasma são partes de um descritor chamado célula.

| | | | |
|---|--|---|--|
| Associar uma ação ligada a uma operação do serviço. | Associa uma UC da categoria fundamental de Objeto a uma UC da categoria de Processo. | Ontológica /Causalidade /Serviço /Tarefa | hasTask. |
| Associar a transitividade de uma ação ligada a uma operação do serviço. | Associa uma UC da categoria fundamental de Processo a uma UC da categoria de Objeto. | Ontológica /Causalidade /Serviço /Assunto | hasDirectObject, hasIndirect Object. |
| Associar os objetos que são resultado de uma ação ligada a uma operação do serviço. | Associa uma UC da categoria fundamental de Processo a uma UC da categoria de Objeto. | Ontológica /Causalidade /Serviço /Objeto | producesObject. |
| Associar os objetos usados em uma ação ligada a uma operação do serviço. | Associa uma UC da categoria fundamental de Processo a uma UC da categoria de Objeto. | Ontológica /Causalidade /Serviço /Objeto | usesObject. |
| Associar dados do provedor de um Serviço Web | Associa uma UC da categoria fundamental de Objeto a uma UC da categoria fundamental de Instituição. | Ontológica /Causalidade /Descendência /Genealógica | hasProvider Information |
| Associar o escopo de uso (afirmativo) de um descritor | Associa uma UC da categoria fundamental de Objeto a uma UC da mesma categoria. | Restritiva /Restrição Afirmativa | isAppliedTo Object |
| Associar o escopo de uso (negativo) de um descritor | Associa uma UC da categoria fundamental de Objeto a uma UC dessa mesma categoria. | Restritiva /Restrição Negativa | isNotAppliedTo Object |
| Contextualizar um descritor no tempo | Associa uma UC de uma categoria fundamental de personalidade ou Processo a uma UC da categoria fundamental de Tempo. | Ontológica/Contato /Encadeamento /Tempo | hasTimeScope. |
| Contextualizar um descritor no espaço | Associa uma UC de uma categoria fundamental de Objeto ou Processo a uma UC da categoria fundamental de Espaço. | Ontológica/Contato /Encadeamento /Espaço | hasSpaceScope. |

É importante observar que o processo de determinação das relações entre as unidades de conteúdo é evolutivo, ou seja, à medida que as outras etapas vão sendo cumpridas, as relações vão sendo atualizadas, para refletir as decisões de projeto adotadas como padrão.

Como artefatos produzidos tem-se: uma lista das relações entre as UC, sua descrição e as restrições de seu domínio (por exemplo: entre que categorias), quando for o caso.

4.1.3. Etapa 3 - Detalhamento das UC

A finalidade dessa etapa é voltada para abordar dois dos aspectos de descrição relacionados ao contexto: a não ambigüidade do contexto dos descritores e a descrição do escopo do uso de cada operação.

Formatted: Bullets and Numbering

4.1.3.1. Aspectos da Não Ambigüidade do Contexto dos Descritores

Aqui buscamos definir de que forma desejamos expressar a semântica de um dado descritor, quando a ele se aplicam métricas, de modo a minimizar a ambigüidade da sua definição. Por exemplo, seja o descritor custo de inicialização, definido por Azevedo Junior²⁵ (2003), como “Intervalo de tempo máximo para que um determinado serviço seja inicializado. Ele inclui tempo para autenticação de usuários, verificação de permissões e inicialização dos itens de hardware e software necessários”. De acordo com nossa proposta podemos especificá-lo com a métrica explícita, assim:

- Criar uma classe para o custo de inicialização.
- Criar uma classe para definir a métrica do custo de inicialização. Um exemplo de métrica é: “Somatório do tempo para autenticação de usuários, verificação de permissões e inicialização de itens de hardware e software”.
- Criar uma classe para definir a unidade de medida da métrica. Um exemplo de unidade de medida é: “Milisegundos”.
- Criar um relacionamento entre a classe do custo de inicialização e sua métrica. Por exemplo: `hasMetrics`.
- Criar um relacionamento entre a classe da métrica do custo de inicialização e a da unidade de medida da métrica. Por exemplo: `hasMeasure`.

²⁵ O autor não define a unidade de medida.

- Criar um atributo para a classe do `custo de inicialização` que defina seu valor. Por exemplo: `hasValue`.

Cabe ressaltar que, caso este descritor seja usado com uma métrica associada, então esta só faz sentido se os conceitos envolvidos na sua definição também estão descritos. No exemplo acima, teríamos de definir descritores para o tempo para autenticação de usuários, verificação de permissões e inicialização de itens de hardware e software, pois o custo de inicialização é definido em função deles. O uso de métricas, como vimos, embora forneça uma semântica mais precisa, pode se tornar uma opção complexa.

Uma alternativa é o uso de descritores em que a métrica está embutida na definição do termo, de forma consensual. Neste caso, podemos optar por especificar apenas a classe `custo de inicialização`, deixando seu significado totalmente implícito.

Devemos decidir qual abordagem vamos adotar. Dessa forma, como artefatos produzidos relativos à não ambigüidade do contexto, tem-se: a decisão de optar por deixar a métrica explícita ou implícita, no caso de descritores que possuem métricas, e a atualização das UC relacionadas para que sua definição contemple a métrica utilizada.

4.1.3.2. Aspectos do Escopo de Uso das Operações

A decisão de informar a restrição do escopo de uso das operações do serviço é pertinente à fase de instanciação da descrição do mesmo, ficando a cargo do seu provedor. Entretanto, a decisão de se permitir ou não essa possibilidade, e de que forma, é pertinente à fase de especificação dos descritores do serviço. Por isso, o que queremos aqui é deixar três formas disponíveis: informar o escopo através de uma lista as opções que são atendidas pela operação (forma afirmativa); informar o escopo através de uma lista de opções que *não* são atendidas pela operação (forma negativa) ou não informar o escopo. Cabe observar que a lista de opções é representada na ontologia através de uma associação com uma lista de classes.

Dependendo do serviço Web a ser descrito, pode ser mais vantajoso que o escopo esteja explícito de forma afirmativa, negativa ou, alternativamente, não estar explícito. Por exemplo, seja o descritor `cidade` do `Iraque`, utilizado em uma

possível descrição de um serviço Web sobre informe de tempo para algumas cidades do Iraque, conforme descrito na Seção 2.3.3. Caso o escopo esteja explícito sugere-se que este seja especificado de duas formas, respectivamente:

- Criar uma relação `isAppliedToObject` ligando a classe que queremos restringir, no caso cidade do Iraque, com subclasses²⁶ dessa classe que representem as cidades do Iraque para as quais a operação do serviço fornece o informe do tempo.
- Criar uma relação `isNotAppliedToObject` ligando a classe que queremos restringir, no caso cidade do Iraque, com subclasses dessa classe que representem as cidades do Iraque para as quais a operação do serviço não fornece o informe do tempo.

Devemos decidir qual abordagem vamos adotar. Dessa forma, como artefatos produzidos relativos à descrição do escopo do uso de cada operação tem-se:

- A decisão de se optar por deixar o escopo explícito de forma afirmativa, negativa ou não deixá-lo explícito.
- Caso se decida por deixar o escopo explícito, então se devem atualizar as UC e suas relações, para que estas reflitam um possível relacionamento de restrição, afirmativo ou negativo, para os descritores pertinentes.

4.1.3.3. Detalhamento da Associação de Parâmetros

Algumas propostas de descrição de serviços, por exemplo, OWL-S (ANKOLEKAR et al., 2001), permitem incorporar, traduzindo para a sua linguagem e forma de descrever, os parâmetros, as operações dos serviços e seus aspectos de ligação. Desta forma, o documento WSDL do serviço não é necessário. Neste caso, no nosso entender, não há por que se associar um conjunto de descritores OWL-S com os do WSDL. Entretanto, há casos em que na linguagem utilizada para descrever o serviço existe uma redundância dos descritores relacionados aos parâmetros de entrada em relação ao WSDL, porém o documento WSDL ainda se faz necessário para descrever os aspectos de ligação. Neste caso, é preciso estabelecer uma correspondência entre esses

²⁶ A associação poderia ser entre a classe e suas instâncias. Entretanto, na nossa proposta, não modelamos os descritores como instâncias e sim como subclasses, i.e., na ontologia gerada então não há instâncias.

descritores, caso se deseje aproveitar a semântica da linguagem de descrição de serviços para a chamada automatizada de um serviço Web por ela descrito.

Por exemplo, seja o descritor *cidade do Iraque*, já citado anteriormente. Dependendo do serviço Web a ser descrito, pode ser que este descritor seja um parâmetro de entrada em uma operação do serviço no documento WSDL com o nome de *city*, na primeira posição de uma operação chamada *GetIraqWeatherInfo*. Consideremos ainda que ele é também um descritor utilizado pela linguagem de descrição de serviços sendo especificada, como o nome *cidade*. Neste caso, nossa proposta para especificação dessa correspondência é a seguinte:

- Criar um descritor para cada parâmetro da operação desejada que exista documento WSDL. Esse descritor, referenciado aqui como *descritor de destino*, contém dois atributos: nome da operação no documento WSDL e posição onde ocorre o descritor nesta operação.
- Criar uma relação para associar o descritor desejado ao descritor de destino. Sugere-se, neste caso, que a relação tenha o nome de *hasTargetWSDLObject*.

Como artefatos produzidos tem-se:

- A decisão de se optar ou não por fazer a associação de um descritor com o seu correspondente no documento WSDL, quando for o caso e o modo pelo qual essa associação é feita .
- Caso se decida por fazer a associação de um descritor com o seu correspondente no documento WSDL, então se devem atualizar as UC e suas relações, para que estas reflitam essa associação.

4.1.3.4. Definição do Propósito das Operações

Uma opção para definir o propósito é fazê-lo de forma fatorada, isto é, em termos de uma ação, do objeto dessa ação (direto e/ou indireto) e de seus qualificadores. Cabe ressaltar que por qualificadores entendemos os descritores da categoria fundamental de *Característica*.

Com a forma fatorada buscamos uma analogia com a estrutura de uma sentença afirmativa da língua portuguesa, formada pelo sujeito (no caso o propósito), o verbo e seu complemento, que pode ser um objeto direto e/ou indireto, de acordo com a transitividade do verbo²⁷. Por exemplo, vamos considerar uma operação de um serviço hipotético de Bioinformática que faça:

“alinhamento local de seqüências de nucleotídeos em bancos de dados genômicos do Japão, utilizando o algoritmo Blast”.

Seu propósito é “alinhamento local de seqüência de nucleotídeos”. Observe-se que consideramos os bancos de dados genômicos e o algoritmo Blast como tipos de recurso, e, por isso, eles não fazem parte da descrição do propósito, assim como Japão, que designa um local.

Por hipótese, vamos considerar que a operação aceita como parâmetro de entrada uma seqüência de nucleotídeos. Vamos ainda considerar que já temos definidos os descritores SeqüênciaNucleotídeo e banco-de-dados, ambos da categoria fundamental de Objeto, o descritor alinhamento, da categoria fundamental de Processo, os descritores genômico e local, da categoria fundamental de Característica, e, por fim, o descritor Japão, da categoria fundamental de Espaço. De acordo com nossa proposta, há duas maneiras sugeridas de se especificar o propósito. Uma dessas maneiras é fazê-lo de modo fatorado, utilizando-se as categorias fundamentais, assim:

- Associar uma classe que pertença à categoria fundamental de Processo à classe à qual se deseja associar um propósito. Utilizar para isso a relação hasTask. Por exemplo: classe AbstractMessage (classe que se deseja associar um propósito) hasTask (relação) comparação (classe de Processo).

²⁷ No caso da descrição do propósito consideramos que não há verbos intransitivos. Mesmo assim, o complemento é opcional, caso haja alguma situação em que se aplique um verbo intransitivo para descrever o propósito.

- Associar essa classe de `Processo` a uma ou mais classes da categoria fundamental de `Objeto`. Para isso usa-se a relação `hasDirectObject` e/ou `hasIndirectObject`, de acordo com a transitividade do verbo utilizado para representar a subclasse. Por exemplo: `alinhamento` `hasDirectObject` `SeqüênciaNucleotídeo`.
- Opcionalmente, pode-se associar um qualificador às classes de `Processo` e de `Objeto` usadas para definir o propósito. Esse qualificador deve pertencer à categoria fundamental de `Característica`. A relação utilizada para fazer essa associação é a `hasObjectQualifier`. Por exemplo: `alinhamento` `hasObjectQualifier` `local`. Deve-se entretanto avaliar se esta alternativa é vantajosa ou não. No caso, se o qualificador `local` não tiver o seu sentido inequívoco (o que não parece ser o caso) e não for utilizado com o mesmo sentido em outro conceito, então talvez seja melhor definir o propósito como `alinhamento_local`. Cabe ressaltar que só pode haver um descritor da categoria fundamental de `Processo` por propósito, ao contrário dos descritores derivados das outras categorias.

Deve-se observar ainda que descritor usado para descrever o propósito da operação é o mesmo usado para descrever o parâmetro de entrada da mesma, quando isso se aplica, evitando redundâncias. Assim por exemplo, `SeqüênciaNucleotídeo` é usado tanto para descrever o parâmetro de entrada da operação quanto para descrever seu propósito.

Uma opção alternativa para descrever o propósito da operação é estabelecer que este se constitui em descritores independentes, formados por aglutinação de outros descritores. Por exemplo: `Alinhamento-Local-de-Seqüência-de-Nucleotídeos`. Neste caso, não se aproveitam os descritores usados para descrever os parâmetros de entrada da operação na definição do propósito.

Cabe lembrar essa é uma diretriz geral, ou seja, todas as operações dos serviços devem ser descritas ou de uma forma ou de outra. O que equivale dizer que há apenas uma forma de descrever o propósito na linguagem.

Devemos decidir qual abordagem vamos adotar. Dessa forma, como artefatos produzidos relativos à descrição do propósito, tem-se: a definição da forma como se escolhe definir o propósito das operações dos serviços, fatorada ou aglutinada, e a atualização das UC relacionadas, para que estejam de acordo com a forma escolhida.

4.1.3.5. Detalhamento da Categorização do Uso

Um serviço Web poderia ter um propósito associado, cuja descrição poderia ser feita da mesma forma como a descrição de suas operações, como proposto na Seção 4.2.4. Entretanto, devido à forma flexível como um serviço pode ser implementado, com operações cujos propósitos nem sempre são muito semelhantes, essa opção não é aconselhada. Nossa proposta é que o propósito do serviço como um todo seja descrito de forma mais abrangente, através de uma categoria de uso, representada por uma terminologia²⁸, da forma como hoje já é feito no UDDI.

A decisão de informar a categoria do serviço é pertinente à fase de instanciação da descrição do mesmo, ficando a cargo do seu provedor. Entretanto, a decisão de se permitir ou não que haja múltiplas categorias possíveis de se associar a um único serviço é pertinente à fase de especificação dos descritores do serviço. Por isso, o que queremos aqui é deixar duas opções disponíveis: categorização múltipla ou única.

A primeira opção fornece uma maior flexibilidade na descrição de serviços cujas operações tenham propósitos menos homogêneos, ao passo que estimula que os provedores de serviço não se preocupem com este aspecto na implementação dos seus serviços. A segunda tem o efeito contrário: menos flexibilidade, mas estimulando a implementação de serviços com propósitos mais homogêneos.

Devemos decidir qual abordagem vamos adotar. Dessa forma, como artefato produzido relativos à categorização do serviço, tem-se: a definição de qual a opção adotada para categorizar os serviços na linguagem, única ou múltipla, e a atualização das UC relacionadas.

²⁸ Uma terminologia é um vocabulário controlado organizado como uma hierarquia de conceitos (REIMER, 2001).

4.1.4. Etapa 4 - Revisão das Definições

O processo de definição dos descritores não é linear, ou seja, existe sempre a necessidade de se rever o que já foi descrito e, a partir daí refinar as definições das UC, categorias e relacionamentos. Nessa etapa pode haver a necessidade de definir novas UC, relações e até mesmo novas categorias. Essa etapa deve ser realizada tantas vezes quantas forem necessárias, até que se tenha segurança de que os conceitos definidos estão de acordo com o escopo e a precisão desejados para o vocabulário.

4.2. Documentação das Diretrizes de Descrição de Serviços Web

A documentação das opções adotadas ao construir a ontologia de serviços ajuda outras pessoas a compreenderem de que forma ela foi projetada, motivando o seu reuso e contribuindo para que suas extensões se dêem de forma mais uniforme. Essa atividade se dá ao longo do processo de uso das diretrizes, de forma gradual e evolutiva.

Nossa proposta é que essa documentação esteja disponível na Web e, por isso, acreditamos que o formato XML é uma boa opção, devido à sua portabilidade e relativa expressividade em relação ao HTML. Além disso, pode ser facilmente manipulado e formatado por algum aplicativo que o utilize para divulgar o seu conteúdo.

O documento XML possui os diversos elementos, oriundos das diversas etapas de nossa abordagem sistemática, conforme exemplificado no apêndice B-1. Seu propósito é o registro da documentação do processo de descrição dos serviços Web.

O próximo capítulo apresenta a implementação das diretrizes propostas na ferramenta de software Ontoguide-4WS

CAPÍTULO 5

A Ferramenta OntoGuide-4WS

As diretrizes propostas cobrem aspectos abrangentes da descrição e organização dos descritores de serviços Web. Muitos desses aspectos estão relacionados ao reuso e extensão de vocabulários, sendo conveniente o uso de uma ferramenta que auxilie essa tarefa. Nesse sentido, a ferramenta OntoGuide-4WS foi desenvolvida para apoiar não só o reuso e extensão, mas também a compreensão dos vocabulários utilizados, a partir do seu tratamento uniforme, de acordo com a perspectiva das categorias fundamentais.

Para compreender melhor de que forma a ferramenta implementa o uso das diretrizes, é apresentado um exemplo prático, onde se procura definir, como o apoio destas, um problema da área de Bioinformática que envolve um conjunto de serviços Web.

As diretrizes propostas estão expressas na ferramenta OntoGuide-4WS através de 3 fases principais e duas secundárias, respectivamente: (i) configuração de ontologias reutilizáveis, (ii) tratamento dessas ontologias, (iii) extensão da ontologia básica de serviços Web, (iv) geração de uma ontologia estendida em OWL e (v) geração de documentação da ontologia estendida em XML.

A partir da ontologia estendida criada vamos buscar por serviços Web descritos com ela, evidenciando que esta permite buscas mais ricas desses serviços do que as que hoje são possíveis com os padrões e propostas atuais. Para ilustrar essa vantagem implementamos um módulo de busca de serviços Web que faz uso da ontologia criada, permitindo que se informem os conceitos definidos na ontologia para se achar os serviços desejados.

A descrição da ferramenta OntoGuide-4WS e a busca por serviços a partir da ontologia criada são descritas a seguir. No final desse capítulo, é apresentada também

uma visão geral do projeto da ferramenta através de seu modelo de classes, seguido dos aspectos de sua implementação.

5.1. Descrição do Uso da Ferramenta OntoGuide-4WS

Esta seção mostra, através de um exemplo a extensão de uma ontologia básica de serviços Web para a área de Bioinformática, escolhida devido à existência de ontologias mais ricas nessa área.

Vamos considerar agora que, nesta ontologia, definimos um conceito relacionado ao propósito de um serviço Web. Este conceito é o *alinhamento local de pares de seqüências de proteínas*. Um exemplo típico de serviço com este propósito é um programa de Bioinformática chamado `BLAST` (ALTSCHUL et al., 1990).

De acordo com nossa proposta, devemos pesquisar na literatura (ou entre especialistas da área) uma definição detalhada do que é nosso conceito. De acordo com um glossário de Bioinformática (WIKIPEDIA, 2004) encontramos a seguinte definição:

“Pairwise sequence alignment methods are concerned with finding the best-matching piecewise (local) or global alignments of protein / amino acid or dna / nucleic acid sequences”.

Ainda, no glossário, temos que um alinhamento de pares de seqüências de proteínas pode ser global e local, o que está de acordo com a nossa definição.

Além disso, um especialista em Bioinformática pode estabelecer que serviços que utilizam este tipo de alinhamento o fazem em bancos de dados de genomas, e que estes serviços conseguem retornar mais pares de seqüências semelhantes que os de alinhamento global, sendo mais rápidos que estes. Por outro lado, seus resultados não são tão precisos em relação à similaridade por inteiro das seqüências comparadas.

Com uma definição mais detalhada como esta, podemos fazer buscas mais ricas por serviços, de acordo não só com o seu propósito, mas também pelas suas características mais genéricas, tais como: grau de precisão, velocidade, escopo do resultado.

Para iniciar, no Menu File, são informados os dados de cada projeto, tais como seu nome e a pasta onde são armazenadas as ontologias utilizadas. A partir da abertura

do projeto, inicia-se o processo de extensão da ontologia de serviços Web, cujas fases são explicadas a seguir.

5.1.1. Fase 1: Configuração da Ferramenta

A primeira fase consiste na configuração da ferramenta. Inicialmente, a ontologia de aplicação básica de serviços Web a ser estendida deve ser definida. A ferramenta já vem com uma padrão, construída a partir do estudo das características dos serviços Web, e da análise de serviços já existentes, conforme relatado na Seção 2.3.3. Após a escolha dessa ontologia básica, devem ser escolhidas as ontologias que vão ser utilizadas para descrever a área de aplicação do serviço, ou outras que se façam necessárias. A Figura 7 ilustra essa atividade.

Cabe aqui uma observação importante: como a ferramenta, na sua versão atual, não apóia a elaboração do diagrama de classes dos domínios, comentado na Seção 4.1.1.1, sugerimos que o mesmo seja feito em alguma ferramenta, antes de se começar a escolha das outras ontologias, complementares à de aplicação.

No exemplo simples explorado nessa seção, teríamos o diagrama ilustrado na Figura 6, servindo de apoio.

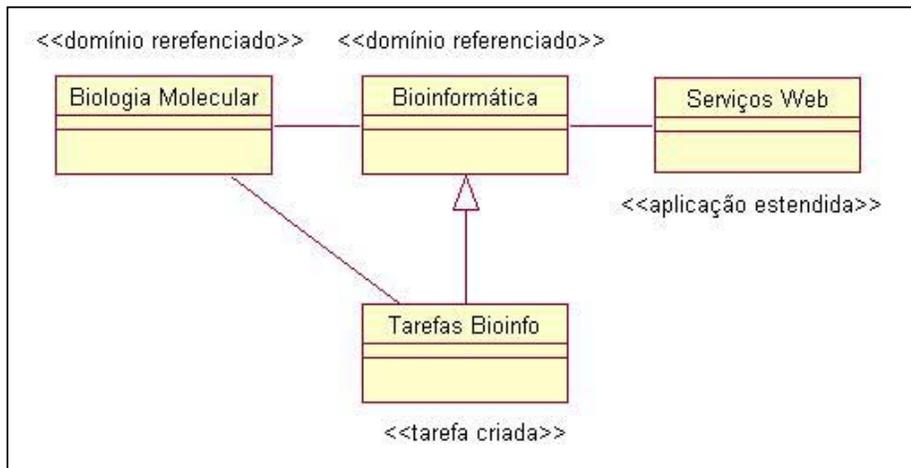


Figura 6 – Diagrama de Classes dos Domínios Utilizados

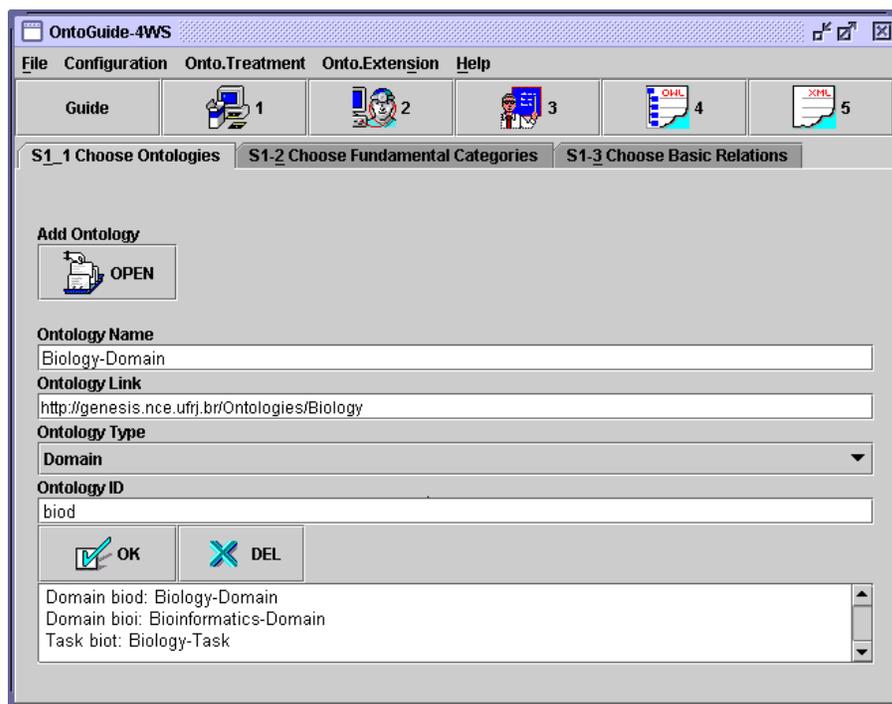


Figura 7 – Escolhendo Ontologias na Ferramenta OntoGuide-4WS

Após identificar as ontologias necessárias, elas devem ser carregadas para o projeto, através do botão *Add Ontology*. Para cada ontologia deve-se informar o seu tipo (genérica, domínio, tarefa ou aplicação) e o seu identificador. Essa identificação tem por objetivo facilitar o reconhecimento da ontologia para fins de reuso na ferramenta. No exemplo, já escolhemos algumas ontologias a reutilizar: uma de domínio, de Biologia Molecular (biot:Biology-Domain), que define conceitos básicos tais como proteína, ou nucleotídeo; uma de tarefa (biot:Biology-Task), que define atividades como alinhamento de seqüências e elaboração de árvore filogenética; e a de aplicação, que no caso é a ilustrada no apêndice B-3. Podemos ainda usar outras ontologias, como por exemplo uma de Bioinformática (bioi:Bioinformatica-Domain), para definir conceitos relativos aos aplicativos utilizados (Blast, Clustalw).

Após a escolha das ontologias, devem ser informadas as categorias fundamentais que são utilizadas para cada uma delas. Além disso, estabelecemos uma correspondência entre as categorias fundamentais de cada ontologia com o conjunto de

categorias fundamentais adotadas como padrão pela ferramenta. Essa correspondência, no entanto, é opcional. Caso não se queira informá-la, basta que se escolha *None* na caixa de seleção *Maps to Basic Fundamental Category*.

A Figura 8 ilustra como se configuram as categorias fundamentais em OntoGuide-4WS.

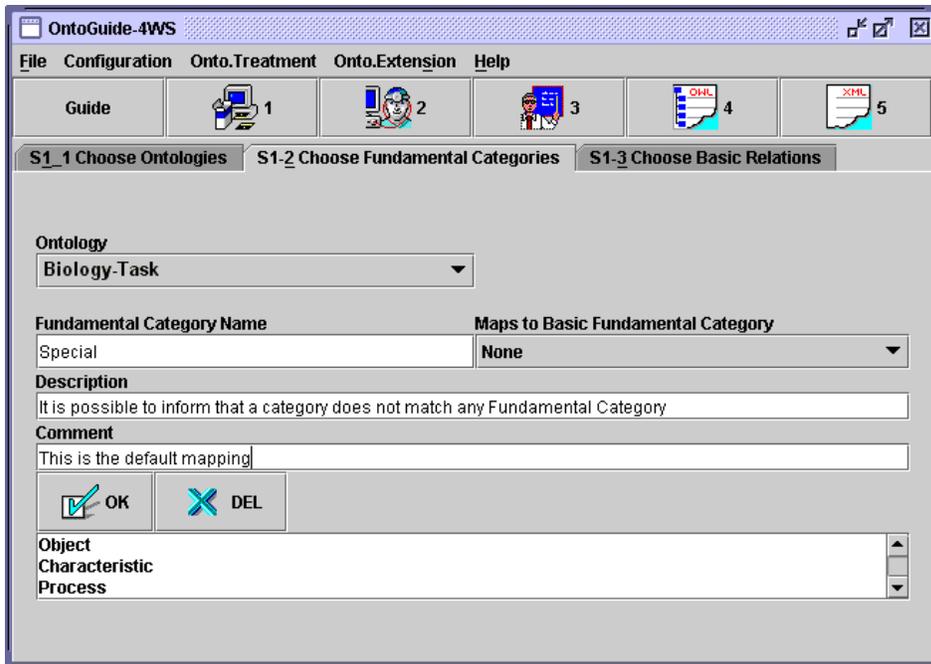


Figura 8 - Configurando Categorias Fundamentais na Ferramenta OntoGuide-4WS

Após a escolha das ontologias, devem ser informadas as relações básicas (*basic relations*) que são utilizadas para cada uma delas. Além disso, estabelecemos uma correspondência entre as relações básicas de cada ontologia com o conjunto de relações básicas adotadas como padrão pela ferramenta. Essa correspondência, no entanto, é opcional.

A Figura 9 ilustra como se configuram as relações básicas em OntoGuide-4WS.

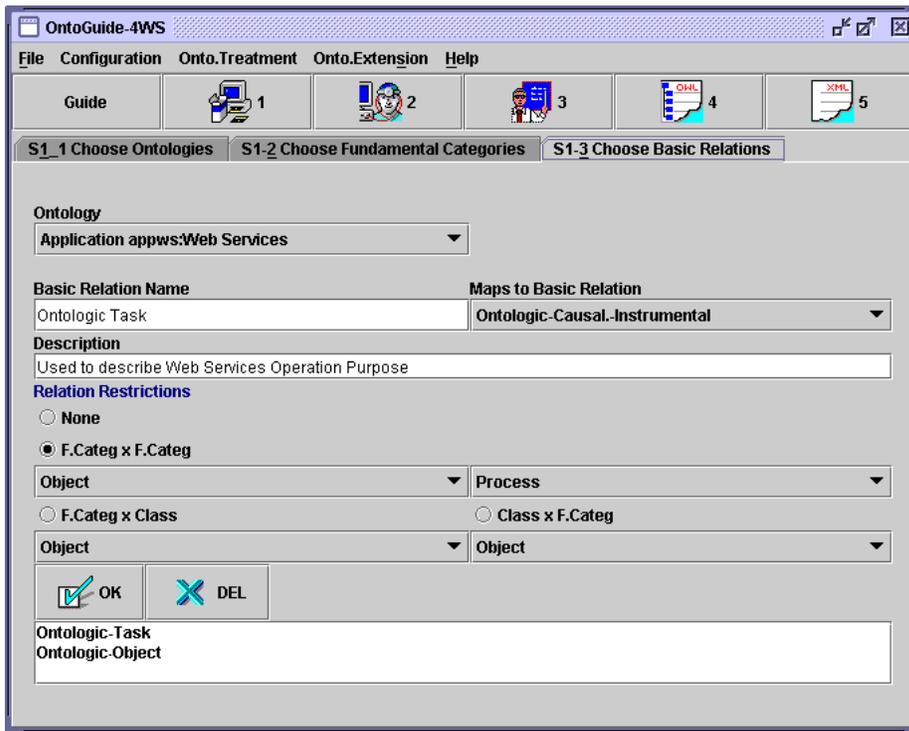


Figura 9 - Configurando Relações Básicas na Ferramenta OntoGuide-4WS

No caso, estamos informando que a relação `Ontologic_Task`, da ontologia de aplicação, corresponde à relação básica `Ontologic_Causal_Instrumental`. E ainda, que esta relação só pode ser utilizada para relacionar classes da categoria fundamental de Objeto (`Object`) a classes da categoria fundamental de Processo (`Process`).

5.1.2. Fase 2: Tratamento das Ontologias

Na segunda fase vamos tratar cada ontologia escolhida. Isso envolve tratar seus termos, associando-os a categorias fundamentais na aba `Map. F. Categories` e, a seguir, na aba `Map F. Relations`, tratar suas relações, associando-as às relações básicas.

Na aba `Map. F. Categories`, para cada termo de uma ontologia, vamos informar a que categoria fundamental este pertence. Esta fase é importante para que se possa ter uma visão mais uniforme do conjunto de descritores que utilizamos para descrever os serviços Web. Assim, por exemplo, evitamos que se utilizem termos de

categorias fundamentais distintas para caracterizar um relacionamento do tipo *é um* ou *faz parte de*.

Para isso, primeiro escolhe-se a ontologia a tratar. Depois, como esta atividade pode ser feita aos poucos, pode-se optar por tratar todos os termos da ontologia escolhida, ou apenas os que ainda não se tenha associado uma categoria fundamental.

A ferramenta então percorre as classes (Term) da ontologia sendo tratada e as vai apresentando, uma de cada vez, para que se informe a categoria fundamental associada. Pode ocorrer, no entanto, que o utilizador das ontologias não consiga ou não deseje informar a categoria fundamental associada. Neste caso, pode informar None na caixa de seleção *Belongs to Fundamental Category*. A figura 10 ilustra essa atividade.

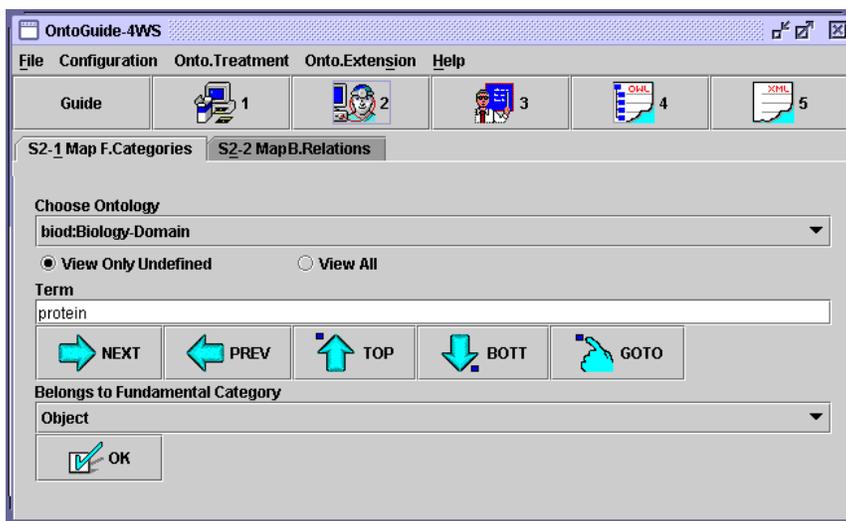


Figura 10 - Tratando Categorias Fundamentais de Ontologias em OntoGuide-4WS

No caso, estamos tratando o conceito *proteína* (protein) da ontologia de domínio de Biologia Molecular (biol:Biology-Domain), informando que este pertence à categoria fundamental de *Objeto* (Object).

A seguir, de forma análoga, na aba *Map. F. Relations*, vamos informar para cada relação de cada ontologia, a que relação básica ela corresponde.

A Figura 11 ilustra o tratamento das relações básicas de uma ontologia.

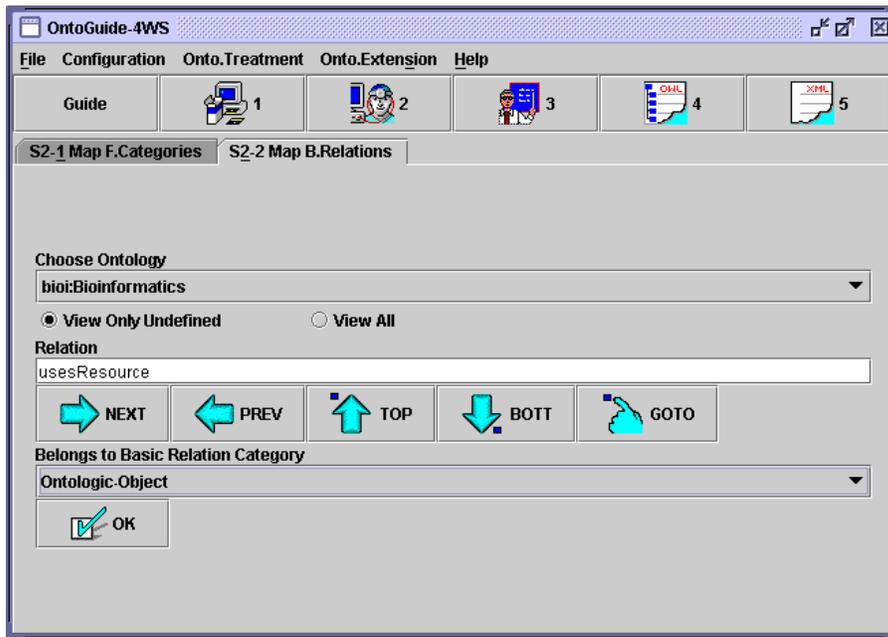


Figura 11 - Tratando Relações Básicas de Ontologias em OntoGuide-4WS

No exemplo da Figura 11, estamos informando que a relação `usesResource` da Ontologia de Bioinformática (`bio:Bioinformatics`), pertence à categoria de relação básica `Ontologic Object`.

Esta fase, juntamente com a Fase 1, implementa a parte de escolha dos domínios, de determinar a forma de organização dos descritores em categorias e as categorias de relações, presentes na Etapa 1 das diretrizes propostas (Seção 4.2.1).

5.1.3. Fase 3: Extensão da Ontologia de Aplicação

Na terceira fase vamos estender a ontologia de aplicação de serviços Web. Para isso, dispomos de quatro atividades: definir conceitos, definir relações, rever definição de conceitos e, por fim, rever a definição de relações.

Para definir conceitos utilizamos a aba `Define Terms`, da opção 3 (`Ontology Extension`) do Menu. A Figura 12 ilustra essa atividade em OntoGuide-4WS.

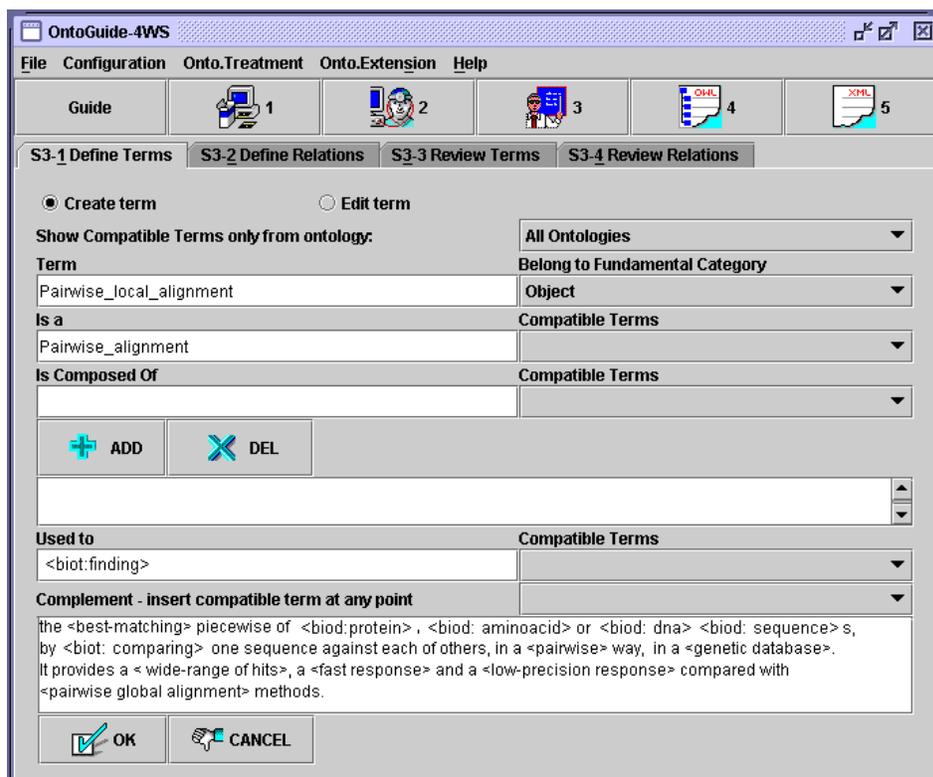


Figura 12 - Definindo Conceitos na Ferramenta OntoGuide-4WS

Nesta opção, a ferramenta permite que se referencie termos das outras ontologias configuradas na primeira fase, filtrando a exibição de seus termos de acordo com as categorias fundamentais que fizer sentido utilizar, dependendo do termo que estiver sendo definido na ontologia de aplicação. A definição do termo é feita informando-se a que categoria fundamental este pertence, o que ele é (ou seja, se é uma especialização de um outro termo ou se está na raiz de uma categoria fundamental), as suas partes (por exemplo: o núcleo e o citoplasma são partes da célula), e para que serve. Ao definir o termo a ferramenta permite que se faça uma descrição estruturada, onde os termos escolhidos de outras ontologias são marcados por caracteres delimitadores (“<” e “>”), precedidos pelo qualificador abreviado da ontologia. Também se permite marcar termos que se reconhecem como conceitos a serem definidos mais tarde, mas que são importantes para compor a definição do termo corrente. A Figura 12 ilustra um exemplo prático dessa funcionalidade, na área de texto `complement`.

A ferramenta OntoGuide-4WS ajuda a formar a definição já a partir do momento em que indicamos a que categoria fundamental o termo pertence. Na Figura 12, na caixa de seleção *Compatible Terms*, associada à caixa de texto *Is a*, por exemplo, só aparecem termos da categoria fundamental de *Processo (Process)* da ontologia de aplicação. Caso exista algum termo já definido que se encaixe na descrição desejada, basta selecioná-lo da caixa de seleção. No caso, o termo sendo descrito, *pairwise local alignment*, pertence à categoria de *Processo (Process)*. Desta forma, o termo que o generaliza, *pairwise alignment*, tem de pertencer à mesma categoria fundamental (no caso, *Processo*). Procederíamos da mesma forma, se o termo que estamos definindo fosse composto por outros conceitos (*is composed of*), representados por outros termos.

Ao definir a utilidade de um serviço, começamos com a definição de uma ação, provavelmente obtida da categoria fundamental de *Processo (Process)* de uma ontologia de tarefa. Na Figura 12, na caixa de seleção *Compatible Terms* associada à caixa de texto *Used to*, são apresentados termos da categoria de *Processo* de ontologias reutilizadas no projeto. No exemplo, selecionamos o termo *<finding>* da ontologia de tarefa qualificada como *biot*.

A maior flexibilidade, no entanto, vem na formação do complemento da definição do termo, informada na área de texto *Complement*. Nesta podemos ir compondo a definição do termo em linguagem natural, com a inserção de termos quaisquer de outras ontologias reutilizadas no projeto, selecionadas na caixa de seleção *Compatible Terms* associada. Neste caso, os termos selecionados na caixa de seleção são inseridos entre caracteres especiais, sendo que o caractere especial de início é seguido pelo identificador da ontologia. Por exemplo, selecionamos *<biot:protein>*, reutilizando o conceito *protein* da ontologia de domínio da Biologia Molecular, qualificada como *biot*. Podemos ainda, marcar um termo entre caracteres especiais, porém sem um identificador de ontologia. Neste caso, a ferramenta entende que é um termo que sabemos ser importante definir, tendo-se optado por fazê-lo mais tarde. Assim, em um momento posterior, podem-se conferir os termos que ficaram faltando definir na aba *Review Terms*.

Para definir relações, utilizamos a aba *Define Relations*, da opção 3 (Ontology Extension) do Menu. A Figura 13 ilustra essa atividade em *OntoGuide-4WS*.

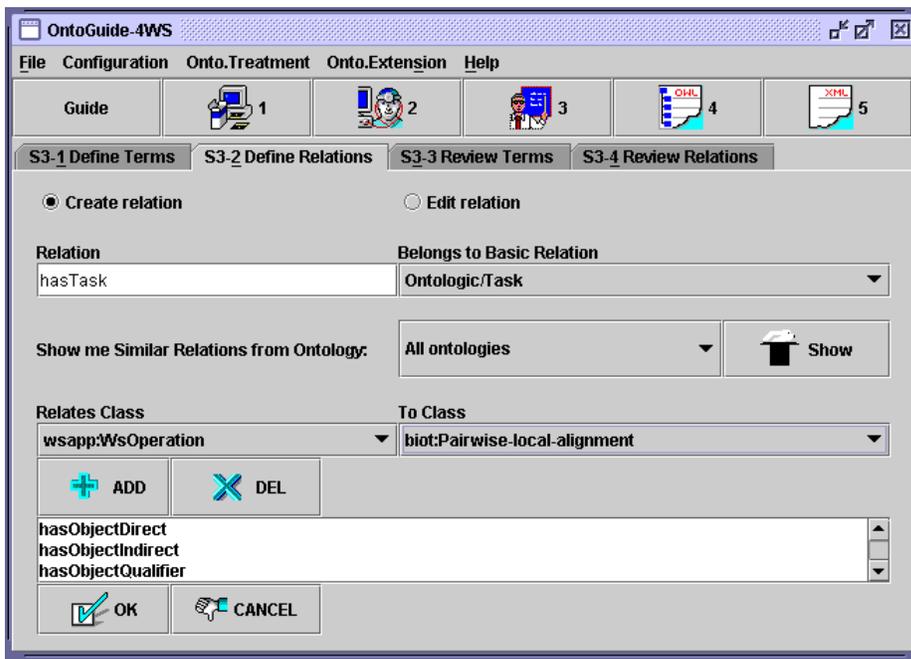


Figura 13 – Definindo Relações em *OntoGuide-4WS*

Nesta opção, a ferramenta permite que se crie relações, permitindo informar apenas as classes associadas que fizer sentido utilizar, de acordo com as classes permitidas para a relação básica a qual pertence a relação sendo definida. Por exemplo, na Figura 13, a relação *hasTask* só pode associar classes da categoria fundamental de Objeto a classes da categoria fundamental de Tarefa, pois informamos que *hasTask* pertence à relação básica *Ontologic / Task*, e esta só permite associar classes destas duas categorias fundamentais (vide Figura 9). Para auxiliar ainda mais a criação de relações, a ferramenta permite, ao pressionar o botão *Show*, que sejam exibidos os dados das relações de ontologias reutilizadas que pertencem à mesma relação básica da relação sendo criada. Essa facilidade tem como objetivo que o usuário possa observar alguma relação já existente nessas ontologias e utilizá-la como fonte de comparação ao criar a sua própria relação, trazendo mais segurança no processo de criação desta.

O processo de definir conceitos é evolutivo, desta forma, após a atividade de definição inicial, devem-se rever as definições feitas neste primeiro ciclo. Essa atividade deve ser feita tantas vezes quantas forem necessárias, até que o processo de refinamento das definições seja considerado satisfatório. A ferramenta OntoGuide-4WS ajuda o processo de revisão através da aba `Review Terms`, conforme ilustrado na Figura 14.

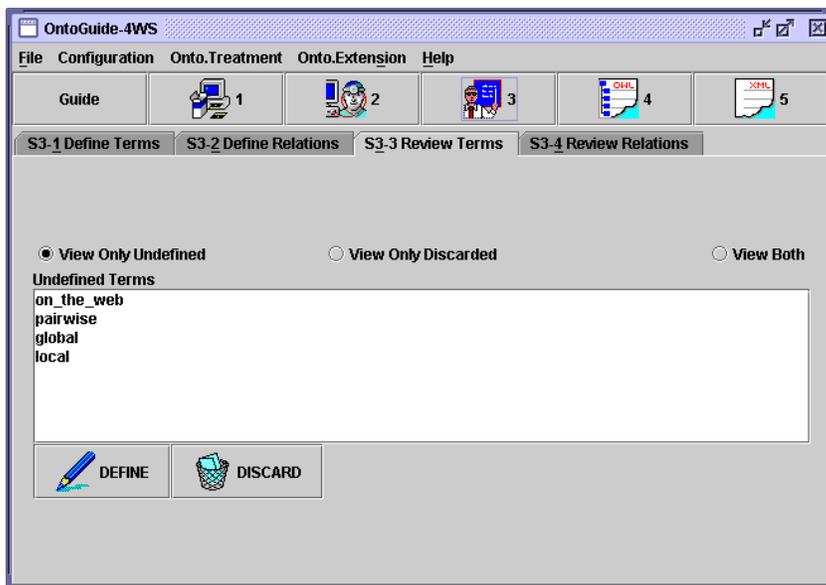


Figura 14 – Revendo a Definição de Termos em OntoGuide-4WS

Na aba `Review Terms` podemos definir um termo marcado entre delimitadores (`< >`), que são indicados para serem definidos posteriormente, ou descartá-lo como conceito. Permite-se rever apenas esses termos (`View Only Undefined`), ou revisar os que foram descartados nessa atividade em um momento anterior, ou ainda revisar ambos ao mesmo tempo.

Por fim, na aba `Review Relations` podemos rever relações a partir de inferências obtidas pela ferramenta a partir dos conceitos definidos, conforme ilustrado na Figura 15.

Nesta atividade, para cada termo, são sugeridas possíveis relações que podem ser definidas ou descartadas. Caso se opte por definir uma relação, a ferramenta abre a tela de definição de relações.

No caso de sugerir relações, os termos marcados são uma importante fonte de raciocínio. Assim, por exemplo, para o caso do termo da Figura 12 (Pairwise-local-alignment), podemos sugerir que ele provavelmente tem algum relacionamento com os termos <pairwise global alignment>, <wide-range of hits> e <fast-response>.

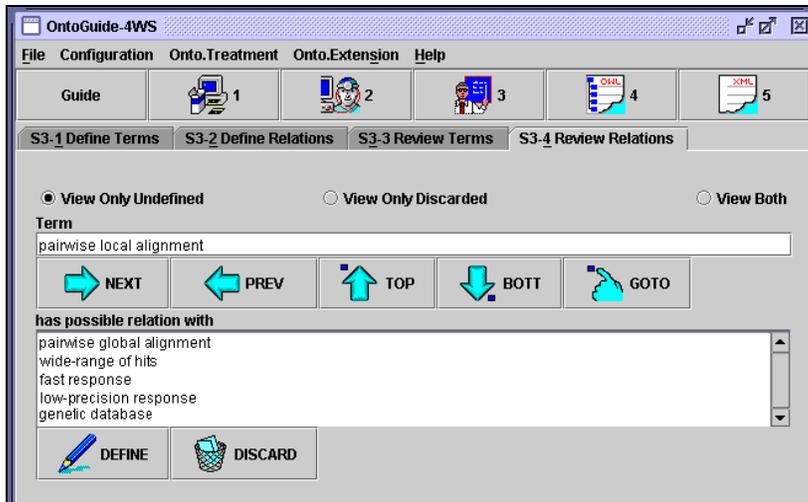


Figura 15 - Revendo Relações na Ferramenta OntoGuide-4WS

Esta fase implementa a parte de descrição precisa de descritores e suas relações, propostas na Etapa 1 das diretrizes, e também dá apoio para a implementação das Etapas 2, 3 e 4 das diretrizes, uma vez que estas estão relacionadas com a definição de conceitos e relacionamentos. Na Seção 5.6 um exemplo de aplicação ilustra com mais detalhes esta fase.

5.1.4. Fase 4: Geração da Ontologia Estendida em OWL

Na quarta fase da ferramenta, vamos gerar o documento da ontologia em OWL, de acordo com os termos e relacionamentos informados. Esta ontologia, entretanto, não é considerada acabada. Ela é considerada um ponto de partida, a partir do qual a ontologia é aperfeiçoada através da introdução de um maior formalismo, como por exemplo, do estabelecimento de relações inversas e de equivalência. Essa fase ajuda ainda a documentar graficamente os descritores e suas relações, uma vez que há ferramentas para isso, a partir de documentos OWL (OWL, 2003).

5.1.5. Fase 5: Geração da Documentação da Ontologia Estendida em XML

Na quinta fase, pode-se gerar um documento XML com uma documentação da ontologia, onde os conceitos marcados na terceira fase geram ponteiros ligando os conceitos, como num documento hipertexto. Além da definição estruturada de cada termo, com esses ponteiros, outros dados são fornecidos, tais como as categorias fundamentais utilizadas e seus significados. O objetivo é que esse documento seja publicado na Web, ajudando o entendimento da ontologia pelos usuários que desejarem utilizá-la. Esta documentação atende à proposta de documentação sugerida pelas diretrizes, sendo que os descritores implementados seguem os sugeridos pela proposta e ilustrados no apêndice B-1.

5.1.6. Observações Sobre o uso da Ferramenta OntoGuide-4WS

Como podemos observar, a partir de uma definição textual acurada conseguimos de forma simples introduzir na ferramenta uma série de conceitos reutilizados de outras ontologias e inferir uma estrutura básica da ontologia e seus relacionamentos.

De uma forma guiada, é possível que um especialista em Bioinformática, mas não em construção de ontologias, consiga gerar uma versão preliminar da ontologia já em OWL, bem como uma documentação em XML. Dessa forma, a partir daí, um especialista em construção de ontologias, com algum conhecimento de Bioinformática, pode acrescentar mais formalismos na ontologia, como por exemplo, através do estabelecimento de relações inversas, e gerar uma ontologia mais completa. Com isso aproveitamos o melhor da especialidade de cada profissional e favorecemos o reuso e a melhor uniformidade na definição dos conceitos de um domínio.

Como contraste, vamos analisar os mesmos conceitos aqui exemplificados, porém definidos na ontologia de Bioinformática presente no projeto myGrid, em sua versão de 03/11/2003 (WROE et.al., 2003). Este projeto foi escolhido por ser o mais avançado na aplicação de uma ontologia de serviços Web a uma área de aplicação específica.

Nesta ontologia temos que o conceito `pairwise global alignment` é um `global alignment` e que `pairwise local alignment` é um `local alignment`, sendo necessário definir esses quatro termos. Ao contrário do que

encontramos na literatura, onde temos uma divisão de alinhamentos em `pairwise` e `multiple`. Neste caso, definimos apenas três termos relacionados a alinhamento `pairwise`: `pairwise alignment`, que se divide em `global alignment` e `local alignment`. Embora seja um exemplo simples, nos indica que pensar o domínio com base em seus conceitos traz vantagens que se traduzem em hierarquias mais bem construídas.

Um outro exemplo, extraído da mesma ontologia, é a definição de serviços que utilizam alinhamento local e global. Neste caso, na ontologia do `myGrid`, existe a definição do termo `global sequence alignment service`, mas não a do termo `local sequence alignment service`, como seria de se esperar. Este último é definido pelo termo `BLAST_service`. Cabe observar que, ainda que os serviços BLAST sejam os mais usados em Bioinformática para alinhamento local, não são os únicos, ou podem surgir novos em um futuro próximo. Dessa forma, seria mais conveniente defini-los como um `local sequence alignment service`. Além disso, o termo `BLAST_service` é definido, dentre outras formas, em função de uma tarefa que executa, no caso, `pairwise local aligning`. Porém, como não há uma descrição mais detalhada do que este significa, não temos como inferir que este tipo de função permite resultados mais amplos, menos precisos e mais rápidos. Desta forma, perdemos flexibilidade e abrangência nas buscas, pois temos de procurar por `BLAST_service` ou por `pairwise local aligning`, para achar serviços com as características mencionadas, quando, ao nosso ver, seria mais natural para um Biólogo procurar por características como escopo, precisão e rapidez e não por conceitos mais restritivos e agregados como `pairwise local aligning`.

A descrição que utilizamos como exemplo é bem simples, podendo ser incrementada com outros atributos que caracterizem melhor a aplicabilidade do conceito associado ao termo `pairwise local aligning`. Entretanto, a ferramenta `OntoGuide-4WS` incentiva que esta descrição seja feita, pois facilita que o Biólogo pense livremente, sem estar limitado por restrições tecnológicas de uma ferramenta tradicional de construção de ontologias.

A próxima seção mostra um exemplo de buscas por serviços descritos com a ontologia estendida, desenvolvida com o apoio da ferramenta `OntoGuide-4WS`.

5.2. Exemplo de Busca por Serviços de Bioinformática

A descoberta dinâmica, no presente trabalho, é ilustrada com um protótipo, partindo da definição de um conjunto de serviços Web da área de Bioinformática e de um conjunto de descritores elaborados com as diretrizes propostas e implementadas na ferramenta OntoGuide-4WS. A Tabela 9 ilustra esses serviços.

Utilizamos um protótipo de uma ferramenta de busca para procurar pelos serviços desejados. Para isso experimentamos dois conjuntos de descritores distintos: um subconjunto baseado na ontologia myGrid e o outro em uma ontologia elaborada com as diretrizes propostas e desenvolvida na ferramenta OntoGuide-4WS. Cabe observar que a ferramenta de busca permite carregar dinamicamente qualquer ontologia compatível com OWL-DL (OWL, 2003).

Os serviços da Tabela 9 foram selecionados com o objetivo de resolver um problema de elaboração de uma análise filogenética de seqüências. Esta análise tem como objetivo reconstruir a relação de parentesco entre espécies, a partir das seqüências de DNA ou de proteínas de diferentes organismos, usadas como fonte de informação (BIONET, 2004). Geralmente, tal análise tem como resultado a elaboração de uma árvore filogenética (WIKIPEDIA, 2004), onde os organismos evolutivamente mais próximos são agrupados em ramos adjacentes.

Para resolver este problema, o biólogo precisa seguir algumas etapas:

1. Definir o tipo de análise que deseja realizar. Por exemplo entre organismos próximos evolutivamente ou entre organismos mais distantes. No primeiro caso, poderíamos estar analisando por exemplo um conjunto de indivíduos da mesma família. No segundo caso, poderíamos estar comparando indivíduos de grupos diferentes tais como bactérias e mamíferos. Dependendo do tipo de análise que se deseja, diferentes organismos e genes são escolhidos.
2. Buscar as seqüências de nucleotídeos dos genes dos organismos que serão utilizados para as análises.

3. Obter um conjunto de seqüências semelhantes (de nucleotídeos ou de aminoácidos) às dos genes dos organismos utilizados. Essas seqüências pertencem a outros organismos.
4. Alinhar globalmente este conjunto de seqüências, para verificar se estão adequadas à análise que se deseja fazer. Por exemplo, se estão relativamente próximas ou se estão muito divergentes.
5. A partir de um conjunto de seqüências alinhadas, elaborar a árvore filogenética, através do uso de um algoritmo adequado. Por exemplo: máxima parcimônia.

Neste problema, o item 1 é subjetivo e depende do propósito que o biólogo tem em mente. Para os demais itens existem ferramentas de software para ajudar o biólogo a atingir seu objetivo. Estas ferramentas manipulam seqüências biológicas (nucleotídeos, aminoácidos) e oferecem a opção do uso de vários algoritmos diferentes para a resolução dos problemas biológicos a que se destinam.

Com base no exposto, selecionamos um conjunto de serviços para os quais devemos descrever seu propósito, o tipo de dados que manipula e o algoritmo que utiliza. Com isso esperamos atender as necessidades do biólogo na busca por ferramentas que venham a lhe auxiliar a resolver seu problema²⁹.

²⁹ Outros atributos são importantes, tais como o formato de entrada e saída dos dados. Optamos por omiti-los para simplificar o exemplo.

Tabela 9 - Serviços de Bioinformática utilizados para análise filogenética.

| Descrição | Propósito | Tipo de dados | Algoritmo |
|-----------------|---|---|-------------------|
| Entrez | Busca por genes | Entrada: palavras-chave Saída: seq. de nucleotídeos. | - |
| Blastn | Alinhamento local de pares de seqüências de nucleotídeos utilizando banco de dados de nucleotídeos retornando as seqüências de nucleotídeos mais semelhantes. | Entrada: seq. de nucleotídeos Saída: seq. de nucleotídeos. | BLOSUM62 |
| Blastp | Alinhamento local de pares de seqüências de aminoácidos utilizando banco de dados de aminoácidos retornando as seqüências de aminoácidos mais semelhantes. | Entrada: seq. de aminoácidos Saída: seq. de aminoácidos. | BLOSUM62 |
| ClustalW | Alinhamento global de um conjunto de seqüências de aminoácidos ou nucleotídeos. | Entrada: seq. de aminoácidos ou nucleotídeos Saída: seq. de aminoácidos ou nucleotídeos. | - |
| Phylip-Protpars | Elaboração de árvore filogenética. | Entrada: seq. de aminoácidos Saída: árvore filogenética. | Máxima parcimônia |
| Phylip-Dnapars | Elaboração de árvore filogenética. | Entrada: seq. de nucleotídeos Saída: árvore filogenética. | Máxima parcimônia |

Para descrever os serviços, buscamos apoio nas diretrizes propostas no Capítulo 4 dessa dissertação, definindo os conceitos necessários e utilizando as categorias fundamentais e relações básicas como ajuda. Além da definição dos conceitos, as diretrizes apóiam a decidir a forma de definição do propósito dos serviços. No caso, optamos pela forma composta por ser mais flexível.

Como resultado obtemos uma ontologia simples, que é utilizada na ferramenta de busca para localizar os serviços desejados. O objetivo aqui é apenas ilustrar a utilidade de nossa proposta e não elaborar uma ontologia completa para a área de Bioinformática. O código fonte da ontologia está disponível no apêndice B-3. Os conceitos definidos e as categorias fundamentais associadas a eles são descritos na Tabela 10.

Tabela 10 - Conceitos usados para definir o propósito dos serviços.

| Conceito | Categoria Fundamental |
|--------------------------------|-----------------------|
| Busca | Processo |
| Elaboração | Processo |
| Alinhamento | Processo |
| Alinhamento Local Pareado | Processo |
| Alinhamento Global | Processo |
| Gene | Objeto |
| Seqüência de Nucleotídeos | Objeto |
| Banco de Dados Genético | Objeto |
| Banco de Dados de Nucleotídeos | Objeto |
| Banco de Dados de Aminoácidos | Objeto |
| Árvore Filogenética | Objeto |
| Algoritmo | Objeto |
| Máxima parcimônia | Objeto |
| BLOSUM62 | Objeto |
| Palavra-chave | Objeto |
| Mais semelhante | Característica |

Como vimos na Seção 4.2.4.1 a descrição do propósito pode ser feita de várias formas. Aqui optamos por defini-lo em função de uma tarefa, um objeto e seus qualificadores, usando as relações propostas na Tabela 8 da Seção 4.2.1.4. Por exemplo, o propósito do serviço `blastn` é definido como:

```
hasTask = alinhamento_local_pareado, hasDirectObject =
SeqüênciaNucleotídeo, producesObject = SeqüênciaNucleotídeo a qual
hasObjectQualifier = mais_semelhante. Cabe observar, no entanto, que o
conceito alinhamento_local_pareado pode ser definido em função de outros
conceitos, como é ilustrado na Seção 5.6. Desta forma, o conceito em si não é
aglutinado.
```

É importante observar que às vezes procuramos pelo que o serviço faz, e às vezes pelo que ele produz. No caso do serviço `blastn`, por exemplo, vamos achá-lo ao procurar pelo que ele produz (`SeqüênciaNucleotídeo mais_semelhante`), e também pelo que ele faz (`alinhamento_local_pareado`). A segunda opção pode não parecer evidente, mas uma vez que definimos `alinhamento_local_pareado` (`pairwise local alignment`) como sendo o processo de “*alinhamento local* que *acha* as *seqüências mais-semelhantes* de aminoácidos ou *nucleotídeos*”, mecanismos de inferência podem localizar o serviço com base nesses atributos.

Ainda, cabe acrescentar que as categorias e relações fundamentais ajudam na definição dos conceitos permitindo achar com mais facilidade conceitos já definidos e as relações a serem utilizadas para uni-los, de acordo com a natureza da relação que vamos descrever. Por exemplo: para definir o conceito `nucleotide sequences` (seqüência de nucleotídeos), vamos procurar por conceitos da categoria fundamental de `Objeto`, de uma ontologia da área de Biologia Molecular. Não sabemos de antemão se este conceito já está definido nesta ontologia, mas sabemos a sua natureza (`Objeto` de Biologia Molecular) e podemos procurá-lo dentro desse escopo mais restrito, facilitando a sua localização. Da mesma forma, ao procurarmos associar o serviço `Web` a seu propósito, sabemos que queremos descrever o resultado da ação de um serviço, `alinhamento local pareado` (`pairwise local alignment`), mas não sabemos ainda qual relação usar. Ao pensar em mais alto nível podemos ver as relações que já planejamos e utilizar a mais adequada, de acordo com o objetivo da categoria de relações básicas a que ela pertence. No caso: descrever uma ação - Ontológica/Causalidade/Instrumental/Tarefa/ `hasTask`.

Após descrever os serviços com a ontologia criada, vamos utilizar a ferramenta de busca para verificar se o conjunto de descritores criados é eficaz. A Figura 16 ilustra a busca pelo serviço `blastn`, através da ferramenta de busca.

| Property | Range Class | Specify Range | Confirm |
|----------------|---------------|--------------------|---------|
| producesObject | BioinfoObject | NucleotideSequence | OK >> |

AbstractMessage, producesObject, NucleotideSequence
 NucleotideSequence, hasObjectQualifier, BestMatching

Send

Figura 16 - Buscando por serviços descritos com a ontologia criada com OntoGuide

A caixa de seleção `Class` contém todas as classes da ontologia estendida de serviços `Web` para a área de Bioinformática. A partir da escolha de uma classe, é possível escolher suas relações com outras classes, montando uma sentença de busca.

Para escolher uma relação associada à classe escolhida, utilizamos a caixa de seleção `Property`. Ali estão contidas todas as relações associadas à classe escolhida na caixa de seleção `Class`.

Cabe observar que na ferramenta de busca `Range Class` é pedida quando o domínio de uma relação é uma classe (poderia ser um valor alfa-numérico). Se essa classe tem filhos, pede-se ainda que se especifique qual vamos usar (`Specify Range`). Se o domínio for um valor a ser informado, então em `Specify Range` é exibida uma caixa de texto, onde o usuário digita o valor desejado.

No caso, ao procurar pelo serviço `Blastn`, vamos utilizar a classe `WsOperation`, pois vamos fornecer atributos relacionados à sua operação. Sobre essa operação, sabemos que ela produz uma seqüência de nucleotídeos. Desta forma, escolhemos a relação `producesObject`, que está associada à classe `BioinfoObject`. Basta então escolher qual `BioinfoObject` desejamos. No caso, `NucleotideSequence`. Mas é preciso ainda restringir o escopo de busca, pois sabemos que não desejamos qualquer `NucleotideSequence`, mas sim uma que tenha como característica ser `BestMatching`. Desta forma, acrescentamos outra sentença de busca, complementar à primeira, onde informamos que a classe `NucleotideSequence` tem uma relação (`Property`) `hasObjectQualifier`, cuja `Range Class` é `BioinfoPrecisionMatching` e cujo domínio desejado (`Specify Range`) é `BestMatching`.

Constatamos que a partir da ontologia elaborada é possível achar todos os serviços desejados. A partir dos dados informados no caso da Figura 16, por exemplo, encontramos um serviço que possui uma operação que produz uma seqüência de nucleotídeos semelhantes (`producesObject (BestMatching (NucleotideSequence))`).

Em contrapartida, ao utilizarmos o conjunto de descritores da ontologia `myGrid` (WROE et al., 2003), com a mesma ferramenta de busca, não é possível localizar o serviço `blastn`, da maneira como seu propósito é descrito pelo biólogo no item 3 do enunciado de seu problema. Da mesma forma, não é possível localizar os programas `protpars` e `dnapars`, comumente utilizados para elaborar árvores filogenéticas, e que utiliza o algoritmo de máxima parcimônia.

Essa diferença de resultados de busca reafirma o que foi comentado na Seção 5.6 em relação ao uso de conceitos agregados. É também um indício da maior flexibilidade de nossa abordagem para criar descritores mais ricos, partindo de uma definição mais precisa de conceitos e do uso de categorias fundamentais que ajudam a pensar e organizar o domínio de aplicação do serviço.

De maneira semelhante, podemos responder à pergunta feita na introdução dessa dissertação:

“Quais os serviços Web de Bioinformática que fazem alinhamento de seqüência de nucleotídeos em bancos de dados genômicos, utilizando o algoritmo Blast, gratuitos, com desempenho médio de 1.0 ms, que utilizem o banco de dados GenBank, disponíveis hoje, domingo, e que retornem a resposta no formato XML ?”.

Para isso, utilizamos as seguintes sentenças de busca:

```

AbstractMessage hasTask Alignment
AbstractMessage hasDirectObject
    InputNucleotideSequence
AbstractMessage usesObject BlastAlgorithm
AbstractMessage usesObject GenBank
AbstractMessage producesObject
    OutputNucleotideSequence
OutputNucleotideSequence hasObjectQualifier OutputFormat
OutputFormat hasValue XMLFormat
AbstractMessage hasQualifierAspect DefaultCost
DefaultCost hasCostValue 0
AbstractMessage hasQualifierAspect ResponseTime
ResponseTime hasMetrics miliseconds
ResponseTime hasValue 10
AbstractMessage hasQualityAspect AvailabilityDays
AvailabilityDays isAppliedToObject Sunday

```

Cabe observar que o GenBank é um banco de dados genômico. Desta forma, não há necessidade de informar esta sentença de busca, pois a mesma pode ser inferida.

De maneira análoga, `InputNucleotideSequence` e `OutputNucleotideSequence` são ambos `NucleotideSequence`. Esta especialização se faz necessária pois podemos ter atributos usados para qualificar uma e outra de forma diferente e precisamos diferenciá-las. Afinal, são seqüências distintas como indivíduos, embora ambas sejam originadas da mesma classe: `NucleotideSequence`.

Também cabe ressaltar que a resposta do serviço não é descrita em função de seus parâmetros de saída, como no WSDL, e sim através da relação `producesObject`. Isto se dá pois pode haver casos de serviços cujos resultados não são enviados no corpo da mensagem de resposta. Por exemplo, podem consistir na geração de um banco de dados que fica disponível para *download*.

De maneira semelhante, `usesObject` é utilizado quando queremos descrever recursos que são utilizados pelo serviço mas não são parâmetros de entrada de uma operação.

5.3. Aspectos de Implementação da Ferramenta de Busca

A interface de busca é implementada através de um conjunto de páginas jsp que se comunicam com um módulo de busca de serviços. O módulo de busca é implementado como uma classe Java e efetua as buscas em arquivos em formato texto escritos em OWL. Esses arquivos contêm as descrições dos serviços feitas com ontologias estendidas de serviços Web, sendo um arquivo para cada ontologia. A interface de busca assume que as duas ontologias estendidas e as respectivas descrições de serviços já estejam disponíveis em arquivos com nomes pré-definidos.

O protótipo é implementado utilizando Java Struts (2004), AXIS (2004), Tomcat (2004) e a API do Jena para OWL com o uso da linguagem de busca RDQL (JENA, 2004).

5.4. Descrição do Diagrama de Classes da Ferramenta

OntoGuide-4WS

O diagrama de classes de OntoGuide-4WS é apresentado na Figura 5 e no apêndice B-2 são apresentados os principais diagramas de colaboração.

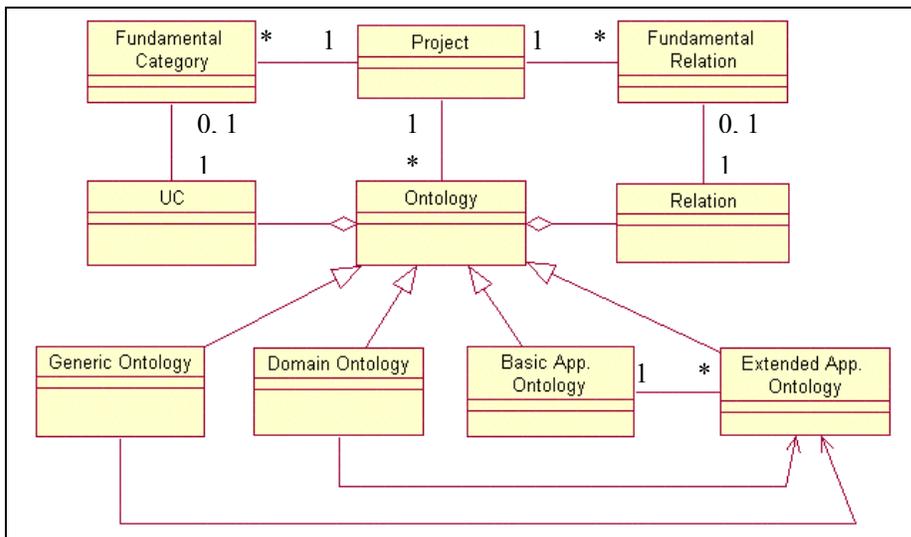


Figura 5 - Diagrama de Classes da Ferramenta OntoGuide-4WS

A classe `Project` contém os dados de configuração de cada projeto. Possui como classes associadas: `Fundamental Category`, com os dados das Categorias Fundamentais; `Fundamental Relation`, com os dados das Relações Básicas, e por fim, `Ontology`, com os dados das ontologias utilizadas. `Ontology` pode ser de quatro tipos: `Generic`, quando a ontologia é genérica; `Domain`, quando é de domínio; `Basic Application`, quando se trata da ontologia básica de serviços Web que se deseja estender e, por fim, `Extended Application Ontology`, quando se trata da ontologia básica estendida. Cada `Ontology` possui um conjunto de `UC`, as quais pertencem a uma `Fundamental Category`, e um conjunto de `Relation`, as quais pertencem a uma `Fundamental Relation`.

5.5. Aspectos de Implementação da Ferramenta OntoGuide-4WS

Para a sua implementação utilizou-se um conjunto de classes Java, e a API do Jena para OWL (JENA, 2004). Esta última fornecendo um conjunto de objetos para manipular ontologias escritas em OWL, permitindo que os descritores criados possam ser exportados para OWL, assim como as ontologias reutilizadas possam ser trazidas e entendidas pela ferramenta.

A escolha dessa API se dá pela sua farta documentação, pela existência de uma lista de discussão ativa e participante, e ainda pela sua constante evolução. Quanto à escolha de OWL, esta se dá pela sua ampla aceitação e evolução dentro do W3C e pelo seu crescente uso como linguagem para construção de ontologias na Web.

As classes utilizadas estão empacotadas em um aplicativo que é instalado como cliente na máquina de cada usuário. É requisito para seu uso a instalação do *Java Runtime Environment* versão v 1.4.2_04 ou superior.

É importante observar que as ontologias que venham a ser importadas pela ferramenta devem ser compatíveis com OWL-DL (ANTONIOU e HARMELEN, 2004), um conjunto mais restrito da linguagem OWL.

CAPÍTULO 6

Conclusão

O uso crescente e o aumento da oferta de serviços Web, muitos deles complexos, tem gerado desafios para a sua busca e descoberta. Para que estas possam se dar de forma automatizada, vocabulários precisos que descrevam esses serviços são necessários. Iniciativas têm sido conduzidas nesse sentido, porém têm resultado na criação de vocabulários que são muito específicos, ou que são muito genéricos. Vocabulários muito genéricos são fáceis de serem estendidos, porém a liberdade de extensão, favorece a criação de descritores não uniformes e incompatíveis, muitas vezes com a repetição de esforços para definir conceitos já existentes em outros vocabulários. Por outro lado, vocabulários muito específicos não atendem às necessidades de diferentes áreas de aplicação e não favorecem o reuso.

Porém, não basta apenas descrever os serviços de forma abrangente. É preciso permitir que estes sejam buscados através de diversas perspectivas e favorecer que mecanismos de busca possam inferir informações mais precisas sobre os serviços Web associados a estas perspectivas. Esta questão envolve o problema de *como* organizar os descritores e seus relacionamentos.

No âmbito dessas questões, nossa proposta procura conciliar a descrição precisa dos serviços Web com uma técnica que apóia a extensão desses descritores para a área de aplicação do serviço e sua organização em categorias. Espera-se assim que o vocabulário resultante possua a precisão adequada para a descoberta dinâmica e favoreça o seu reuso e o de outros que venham a ser identificados como necessários para a descrição abrangente do serviço.

6.1. Contribuições

Esse trabalho apresenta quatro contribuições principais:

- um estudo detalhado das características dos serviços Web;
- um conjunto de diretrizes que fornece uma abordagem sistemática para a criação, extensão e organização de descritores de serviços Web;
- uma ferramenta que viabiliza a aplicação das diretrizes;
- uma forma automatizada de documentar os vocabulários criados.

O estudo detalhado das características dos serviços Web resultou em um conjunto básico de descritores, o qual representa a ontologia de aplicação de serviços Web, utilizada em nossa proposta.

O conjunto de diretrizes foi obtido a partir da análise dessas características e do uso de uma técnica, adaptada de iniciativas oriundas da área de Ciência da Informação e da Terminologia. O estudo das características e as diretrizes podem ser utilizados em quaisquer iniciativas de descrição de serviços Web, auxiliando-as no processo de detalhamento de atributos e de sua extensão.

Essas diretrizes são apoiadas pela ferramenta OntoGuide-4WS, mas podem ser utilizadas de forma independente desta. O protótipo desta ferramenta foi parcialmente implementado e encontra-se em desenvolvimento através de um projeto de final de curso de graduação em Informática.

As diretrizes auxiliam no processo de definição de conceitos e extensão de vocabulários, estimulando o reuso de outros já existentes. Também proporcionam, através do uso de categorias básicas de descritores e relacionamentos, uma melhor organização e identificação da natureza dos conceitos a criar ou reutilizar, permitindo pensá-los em mais alto nível e, dessa forma, definindo-os de modo mais coerente. A ferramenta OntoGuide-4WS apóia essas tarefas, fornecendo um conjunto de categorias padrão para termos e relacionamentos, usadas para nivelar as ontologias utilizadas, ajudando assim na definição de conceitos mais uniformes.

A ferramenta OntoGuide-4WS, por sua vez, espelhando as diretrizes e adotando uma forma intuitiva e guiada para a definição de novos conceitos, contribui ainda para

minimizar a dependência entre especialistas no domínio de aplicação do serviço (analistas de domínio) e na construção de vocabulários (analista de tecnologia).

Os primeiros são beneficiados por uma interface voltada para a sua maneira de descrever conceitos: utilizando uma descrição textual, onde, como fonte de ajuda, é possível introduzir conceitos já definidos de outros vocabulários ou marcar os conceitos considerados importantes. A ferramenta OntoGuide-4WS possibilita também alguma inferência sobre as definições, guiando o uso de termos semelhantes de outras ontologias, e sugerindo possíveis relacionamentos entre os termos definidos.

Os analistas de tecnologia, por sua vez, são beneficiados por já iniciarem seu trabalho a partir de um conjunto de descritores já previamente definidos pelo analista de domínio. Além disso, a ferramenta OntoGuide-4WS permite exportar esse vocabulário inicial como uma ontologia em OWL. Dessa forma, o analista de tecnologia pode iniciar dali o seu trabalho, dirigindo seu foco para refiná-lo, introduzindo formalismos adicionais, tais como relações inversas, de equivalência ou de restrição de domínios, possibilitando inferências mais ricas sobre ela, que permitem, respectivamente, encontrar conceitos inversos, equivalentes e semanticamente mais próximos do conceito que se busca.

Por fim, a partir das descrições introduzidas na ferramenta OntoGuide-4WS, permite-se gerar uma documentação automática, que pode ser introduzida em uma página Web, explicando de que forma a ontologia é categorizada, quais os conceitos que ela define, em linguagem natural, e permitindo navegar entre os conceitos, como em um sistema de hipertexto.

Dessa forma, como fruto do uso das diretrizes propostas nesta dissertação, esperamos ter vocabulários mais precisos, detalhados e compatíveis com outros já existentes, adequados para a descrição de serviços Web voltada para a descoberta dinâmica, e ainda bem documentados, o que facilita a sua compreensão e extensão.

Buscando comprovar a viabilidade de nossa proposta, realizamos um experimento, contemplando as seguintes etapas:

- Utilizamos as diretrizes para estender o conjunto básico de descritores³⁰ para a área de Bioinformática, criando uma ontologia estendida simples.
- Descrevemos um conjunto de serviços de Bioinformática, sob o ponto de vista de um Biólogo, com a ontologia estendida, utilizando para isso a ferramenta Protégé 2000.
- Descrevemos o mesmo conjunto de serviços de Bioinformática com um subconjunto de descritores de uma ontologia da área, pertencente ao projeto myGrid (WROE et.al., 2003).
- Utilizamos o protótipo do módulo de busca por serviços Web, para localizar os serviços descritos.
- A partir de nossa ontologia estendida foi possível mostrar que, através do protótipo da ferramenta de busca, para um mesmo conjunto de serviços Web disponível, o universo de serviços Web localizados a partir da utilização de nossa ontologia estendida é um superconjunto do universo de serviços Web localizados a partir da utilização da ontologia de myGrid.

Isso posto, cabe um questionamento: será essa uma evidência tão clara da viabilidade de nossa proposta? Para responder esta questão, é preciso entender o momento que cerca as iniciativas de construção de ontologias, dentro do contexto maior da Web semântica, e as conseqüentes dificuldades enfrentadas ao longo dessa dissertação.

6.2. Limitações e Dificuldades Encontradas

Um dos principais obstáculos para a pesquisa relacionada à Web semântica é o estágio inicial em que esta se encontra. Apesar do grande interesse nesta iniciativa, ainda são poucos os exemplos de organizações ou comunidades que utilizem amplamente as tecnologias a ela relacionadas, tais como ontologias indexando conceitos em páginas Web ou serviços Web mais complexos e disponíveis para utilização. Para ilustrar essa afirmativa, podemos recorrer ao sítio do desafio da Web Semântica

³⁰ Vide apêndice B3.

(CHALLENGE, 2004), onde, por um período de cinco anos, são fornecidos prêmios para os que conseguirem oferecer soluções inovadoras nessa área.

De acordo com os organizadores do desafio, apesar de muitas questões relevantes terem sido apresentadas em eventos relacionados, ainda não existe um exemplo integrado que deixe uma idéia clara do que a Web Semântica pode oferecer. De fato, o que constatamos é que poucos trabalhos são divulgados de forma completa e documentada. Ao buscar por ontologias para reuso em nossa aplicação, por exemplo, constatamos que poucas ontologias existem disponíveis para uso. Dentre essas, muitas são apropriadas como exemplos para aprendizado, mas não para uso efetivo. Outras, disponíveis no decorrer dessa pesquisa, tiveram seus fontes retirados da Web (WROE et al., 2004) (PERICH, CHAKRABORTY, 2001), provavelmente denotando sua necessidade de evolução.

Dessa forma, torna-se muito difícil fazer uma análise comparativa ou um ensaio baseado em vocabulários reais. Apesar das discussões em torno de ontologias de tarefa na literatura, não se encontram exemplos disponíveis para uso, pelo menos, até onde pudemos perceber. Da mesma forma, a falta de documentação muitas vezes dificulta sobremaneira o entendimento de ontologias que conseguimos utilizar, tornando difícil delinear o escopo de seus descritores em relação aos domínios que estes se propõem a descrever, como é o caso da ontologia do projeto myGrid (WROE et. al, 2003).

Por outro lado, é verdade que existe uma comunidade atuante e cooperativa, disposta a se ajudar e a compartilhar suas dúvidas e experiências. Foi o que constatamos ao interagir com as listas de discussão dos usuários Jena (2004) e Protégé (CRUBÉZY et al., 2004). Essa interação, apesar de profícua, revelou que ainda há questões em aberto na comunidade que constrói ontologias, como por exemplo, a questão do reuso. Apesar de muito se discutir esse assunto, até onde podemos perceber, não existe ainda uma abordagem sistemática para o projeto de ontologias com o foco em reutilização. Nesta linha, diversas questões ainda necessitam de maior discussão. Por exemplo: devemos incentivar a incorporação de trechos de outras ontologias na nossa, ou devemos apenas referenciá-las? E ainda, que comunidades (de modo geral), ou que ontologias (de modo específico), estão reutilizando vocabulários, e de que forma? De nossas interações com a comunidade que desenvolve ontologias, o que nos foi apontado

como reuso são, na verdade, ontologias de grande porte como a Gene Ontology (GO, 2004), ou léxicos como o Wordnet (MILLER, 1995), onde, se há reuso, este certamente não é evidenciado com clareza.

Dessa forma, ficou claro, ao longo dessas interações, que não há muitas certezas ao se buscar construir ou estender uma ontologia. No caso de ontologias voltadas para serviços Web uma visão sintagmática se impõe como pré-requisito, apresentando diversas dificuldades. Conhecer o domínio de serviços Web; levantar suas características; conhecer um domínio com riqueza de descritores, onde estes serviços se aplicam, como a Bioinformática; conhecer por fim, as questões relativas à organização de descritores, aliando-as às questões de construção de ontologias, procurando promover o seu reuso: esses têm sido nossos desafios.

Assim, ao responder a pergunta que lançamos sobre a viabilidade dessa proposta, o que podemos dizer é que, dentro do contexto das dificuldades encontradas, julgamos ter dado um passo no sentido de organizar e definir conceitos mais precisos e também apontar meios para facilitar a descrição de serviços Web, levantando algumas questões relevantes, como a do reuso.

6.3. Trabalhos Futuros

Para concluir, trabalhos futuros podem vislumbrar outras possibilidades com base no proposto nessa dissertação:

- Explorar o uso de diferentes abordagens propostas na área de Ciência da Informação e da Terminologia, que tratam de questões relacionadas à definição de conceitos, esquemas de categorização de termos e de relacionamentos, e ainda as que tratam de introduzir um maior formalismo na definição de conceitos. Essas abordagens, dentre as quais citamos a tese de doutorado de Garcia (GARCIA, 2004), podem permitir o aproveitamento mais preciso de conceitos compatíveis de outras ontologias na definição de termos da ontologia de aplicação de serviços Web.

- Conjugar o uso da ferramenta em outros sistemas voltados para a construção de ontologias. Por exemplo, adaptando OntoGuide-4WS para ser usado como um *plugin* da Protégé (CRUBÉZY et al., 2004). Desta forma, proporciona-se um ambiente integrado onde o analista de domínio trabalha junto com o analista de tecnologia, cada um com a sua visão das ontologias, embora podendo compartilhá-las um com o outro.
- Adaptar o módulo de exportação da ontologia em OWL para outras linguagens com maior formalismo, possivelmente baseadas em lógica de primeira ordem, ou ainda que implementem regras (RULEML, 2004). Isso pode implicar em introduzir outras convenções na ferramenta, quando da definição de conceitos. Em vez de apenas utilizar os marcadores “<” e “>”, podem-se criar outros para definir regras, como por exemplo: “[se]” e “[então]”.
- Evoluir o protótipo da ferramenta de busca para uso com essas linguagens, e os vocabulários com elas criados, incorporando-os em um projeto que tenha o foco em fornecer um ambiente para publicação e descoberta dinâmica de serviços Web. Nesse ambiente, a ferramenta de busca se transforma em uma API, capaz de interagir com agentes que encaminham pedidos de localização de serviços e com um repositório onde estão armazenadas as descrições de serviços e as ontologias disponíveis para essas descrições.
- Incorporar à ferramenta um diagrama que permita visualizar graficamente as ontologias sendo criadas ou reutilizadas, de modo a facilitar seu entendimento.
- Explorar o levantamento das características dos serviços Web (CAMPOS, PIRES, CAMPOS, 2004b) como um guia para a elaboração de vocabulários mais precisos e detalhados, que podem ser utilizados em iniciativas de composição automática de serviços, como as de Costa, Pires e Matoso (COSTA, PIRES, MATTOSO, 2004) e Pires (2002), ou de buscas, tais como as propostas de extensão do UDDI (SHAIKHALI, 2003) (MILES et al., 2003) (CHEN et. al., 2003) (RAN, 2003). Nesses

casos, ressaltamos o detalhamento do propósito do serviço como benefício direto da aplicação do levantamento.

- Explorar este como um guia para a elaboração de logs de serviços Web mais precisos e detalhados, como o proposto por Cruz et al. (2004), que podem ser utilizados para acompanhar o perfil de uso de clientes ou para fins administrativos. Descritores mais ricos de contexto, explorando aspectos legais, podem revelar que um determinado produto oferecido por um serviço tem seu consumo diferenciado dependendo do local onde é feito o pedido, possivelmente por causa das diferentes taxas de imposto cobradas por local. Por exemplo, as taxas de imposto em New York, são maiores que as de New Jersey. O que pode levar um consumidor a preferir serviços Web que atendam a áreas onde o imposto é menor.

Referências Bibliográficas

AKIHA, N. *Web Services at Work*, DM Review, 2003. Disponível em: http://dmreview.com/editorial/dmreview/print_action.cfm?EdID=6513. Último acesso em: 02 nov 2003.

ALTSCHUL, S .F.; GISH, W.; MILLER, W.; MYERS, E.W.; LIPMAN, D.J. *Basic local alignment search tool*, Journal of Molecular Biology, 1990, v. 215, pp. 403-410.

AMAZON, *Amazon.com Homepage*. Disponível em: <http://www.webservices.org/index.php/article/view/959/>. Último acesso em : 10 mai 2003.

ANKOLEKAR, A.; BURSTEIN, M.; HOBBS, J.; LASSILA, O.; MARTIN, D.; McILRAITH, S.; NARAYANAN, S.; PAOLUCCI, M.; PAYNE T., SYCARA, K.; ZENG, H. *DAML-S: Semantic Markup for Web Services*, Proceedings of the International Semantic Web Working Symposium (SWWS), 2001.

ANTONIOU, G.; HARMELEN, F. *Handbook on Ontologies in Information Systems, Web Ontology Language: OWL*, Springer-Verlag, 2004. pp. 67-92.

ARTALE, A.; FRANCONI, E.; GUARINO, N.; PAZZI, L. *Part-whole relations in object-centered systems: An overview*. Data & Knowledge Engineering, v.20, n.3, pp. 347-383, 1996.

AXIS, *Apache AXIS*, 2004. Disponível em: <http://ws.apache.org/axis/>.

AZEVEDO JR, V., *Webtransact-EM: um modelo para execução dinâmica de serviços Web semanticamente equivalentes*. Dissertação de Mestrado, UFRJ, COPPE, 2003.

BENJAMINS, V. R.; WIELINGA, B.; WIELEMAKER, J.; FENSEL, D *Towards Brokering Problem-Solving Knowledge on the Internet*. EKAW-99, Dagstuhl Castle, Germany, LNCS, Springer-Verlag, 1999.

BERNERS-LEE, T. *Semantic Web Roadmap*, 1998. Disponível em www.w3c.org/DesignIssues/Semantic.html. Último acesso em 07 nov 2002.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. *The Semantic Web*, Scientific American, Mai 2001.

BINKERT, P. *Linguistic Analysis: Lecture Notes and Workbook for Lin 180*. The Langtech Corporation. Milford, Michigan, 2003.

BIONET, *Rede Paranaense de Bioinformática - Tutorial de Bioinformática da PUCPR*, 2004. Disponível em: <http://www.bionet.pucpr.br/tutoriais.php>. Último acesso em 30 mai 2004.

BJORNER, D. *On the use of formal methods in software development*, Proceedings of the 9th international conference on Software Engineering, p.17-29, Monterey, California, United States, 1987.

BOSWORTH, A.; BOX, D.; CHRISTENSEN, E.; CURBERA, F.; FERGUSON, D.; FREY, J.; KALER, C.; LANGWORTHY, D.; LEYMAN, F.; LUCCO, S.; MILLET, S.; MUHKE, N.; NOTTINGHAM, M.; ORCHARD, D.; SHEWCHUK, J.; STOREY, T.; WEERAWARANA, S. *Ws-Addressing*, 2003. Disponível em: <ftp://www6.software.ibm.com/software/developer/library/ws-addressing.pdf>. Último acesso em 03 mar 2003.

BOX, D.; CABRERA, F.; CHRISTENSEN, E.; CURBERA, F.; FERGUSON, D.; FERRIS, C.; FREUND, T.; FREY, J.; HONDO, M.; IBBOTSON, J.; KALER, C.; LANGWORTHY, D.; LEYMAN, F.; LIMPRECHT, R.; LUCCO, S.; MILLET, S.; NADALIN, A.; NIELSEN, H. F.; SHEWCHUK, J.; STOREY, T.; WEERAWARANA, S. *Reliable Message Delivery in a Web Services World: A Proposed Architecture and Roadmap*, 2003. Disponível em: <http://msdn.microsoft.com/webservices/default.aspx?pull=/library/en-us/dnglobspec/html/ws-rm-exec-summary.asp>. Último acesso em 05 mar 2003.

BRADY-MYEROV, A.; CHARLES, W.; JARRETT, T.; KOTTAHACHI, B. *Web Services: the next IT revolution?*, 2001. Disponível em: http://web.mit.edu/buddhika/www/15.567/deliverables/Web_Services_Final.pdf. Último acesso em 01 mar 2003.

BRÉZILLON, P. *Context in problem solving: A survey*. The Knowledge Engineering Review, v.14, n.1, pp. 1-34, 1999. Disponível em: <http://www-poleia.lip6.fr/~brezil/Pages2/Publications/KER99.pdf>. Último acesso em 02 mar 2003.

CAMPOS, L.M.; PIRES, P.F.; CAMPOS, M.L.M. *OntoGuide-4WS: Uma Abordagem Sistemática para Descrição de Serviços Web*, WebMedia/LA-Web, 2004a, Ribeirão Preto, São Paulo, Brasil. A ser publicado.

CAMPOS, L.M.; PIRES, P.F.; CAMPOS, M.L.M. *Atributos para Descrição de Serviços Web*, Relatório Técnico, NCE-UFRJ, 2004b.

CAMPOS, M.L.A. *Perspectivas para o Estudo da Área de Representação da Informação*, Ciência da Informação, v. 25, n. 2, 1995.

CAMPOS, M.L.A. *A Organização de Unidades do Conhecimento em Hiperdocumentos: o modelo conceitual como um espaço comunicacional para a realização da autoria*. 2001a. 198p. Tese de Doutorado – Universidade Federal do Rio de Janeiro, Escola de Comunicações, Instituto Brasileiro de Informação em Ciência e Tecnologia, Rio de Janeiro. Disponível em: <http://genesis.nce.ufrj.br/dataware>. Último acesso em 01 dez 2003.

CAMPOS, M.L.A. *Linguagem Documentária: Teorias que fundamentam sua elaboração*, EdUFF, Niterói, RJ, 2001b.

CARDOSO, L.; MILLER, J.; SHETH, A.; ARNOLD, J. *Modeling Quality of Service for Workflows and Web Service Processes*, in The VLDB Journal, 2002.

CASATI, F.; ILNICKI S., JIN; L.; KRISHNAMOORTHY, V.; SHAN, M. *Adaptive and Dynamic Service Composition in eFlow*. In Proc. of the International Conference on Advanced Information Systems Engineering, Stockholm, Sweden, 2000.

CHAKRABORTY, D.; PERICH, F.; AVANCHA, S.; JOSHI, A. *DReggie: Semantic Service Discovery for M-Commerce Applications*, Workshop on Reliable and Secure Applications in Mobile Environment, 20th Symposium on Reliable Distributed Systems, New Orleans, LA, 2001. Disponível em: <http://daml.umbc.edu/papers/dreggie.pdf>. Último acesso em 07 mar 2003.

CHALLENGE, Semantic Web Challenge. Disponível em: <http://challenge.semanticweb.org>. Último acesso em 20 jun 2004.

CHEN, Z.; CHIA; LIANG-TIEN.; SILVERAJAN, B.; LEE, BU-SUNG *UX- An Architecture Providing QoS-Aware and Federated Support for UDDI*. Proceedings of the International Conference on Web Services, ICWS '03, Las Vegas, Nevada, USA, 2003, pp 171-176, CSREA Press.

CHIU, E. *ebXML Simplified. A Guide to the New Standard for Global E-Commerce*. Jonh Wiley & Sons, Inc., New York, NY, 2002.

CONEN, W.; KLAPSING, R. *Exchanging semantics with RDF*, In: Information Age Economy Fifth International Conference of Wirtschaftsinformatik Augsburg, 2001. Disponível em: <http://citeseer.nj.nec.com/442959.html>. Último acesso em 08 mar 2003.

CONWAY, A. *Web Services and Distributed Objects: Competing or Complementary?* E-servQ homepage institucional, 10 dez. 2001. Disponível em: http://e-serv.ebizq.net/wbs/conway_1.html. Último acesso em: 6 jun 2002.

CORBA, Object Management Group, *The Common Object Request Broker: Architecture and Specification*, 1995.

COSTA, L. A. G.; PIRES, P. F.; MATTOSO, M. L. *Automatic Composition of Web Services with Contingency Plans* In: 2004 IEEE International Conference on Web Services (ICWS 2004), 2004, San Diego.

COYLE, F.P. *XML, Web Services, and the Data Revolution*. Addison-Wesley Pub Co; New York, 1st edition, 2002.

CRNKOVIC, I. et al, *Building Reliable Component-Based Software Systems – Artech House*; 1st edition, 2002. Disponível em: <http://www.idt.mdh.se/cbse-book/>. Último acesso em: 04 mar 2003.

CRUBÉZY, M.; DAMERON, O.; FERGERSON, R.; KNUBLAUCH, H.; MUSEN, M.; NOY, N.; RUBIN, D.; TU, S.; VENDETTI, J. *Protege Ontology Editor*, 2004, the Stanford University School of Medicine. Disponível em: <http://protege.stanford.edu/>. Último acesso em 25/05/2004.

CRUZ, S. M. S.; PIRES, P. F.; CAMPOS, L. M.; CAMPOS, M. L. M. *Monitoring E-Business Web Services Usage through a Log Based Architecture* In: 2004 IEEE International Conference on Web Services (ICWS 2004), 2004, San Diego.

CUI, Z.; O'BRIEN, P. *Domain Ontology Management Environment*, Proceedings of the 33rd Hawaii International Conference on System Sciences, Hawaii, 2000.

CYC, *Cyc Upper Ontology. The open source version of the Cyc technology*. Disponível em: <http://www.cyc.com/cyc/opencyc/overview>. Último acesso em 02 abr 2004.

DAHLBERG, I. *A referent-oriented analytical concept theory of interconcept*. International Classification, v. 5, n. 3, pp.142-150, 1978.

DCOM, Microsoft Corp., Digital Equipment Corp., *The Component Object Model Specification*, 1995. Disponível em: <http://www.opengroup.org/pubs/catalog/ax01.htm>. Último acesso em 20 abr 2003.

DERTOUZOS, M. *A Revolução Inacabada*; tradução de Claudia Lopes 1. ed. São Paulo: Futura, 2002.

DEY, A., K., *Understanding and Using Context*, Personal and Ubiquitous Computing Journal, v. 5, n.1, 2001, pp. 4-7. Disponível em: <http://www.cc.gatech.edu/fce/ctk/pubs/PeTe5-1.pdf>. Último acesso em 06 mar 2003.

DING, Y., Foo, S., *Ontology research and development. Part 1 – a review of ontology generation*. Journal of Information Science, 28 (2) 2002, pp. 123-136.

DOTFASTW, *DotFastWeather - Dynamic On Time Services*, 2004. Disponível em: <http://www.serviceobjects.com/products/default.asp?bhcp=1>. Último acesso em 10 mar 2003.

DUNS, *Dun & Bradstreet Data Universal Numbering System*, 2004. Disponível em: <http://www.abnsearch.com.au/about.htm>. Último acesso em: 26 mai 2004.

DUMAS, M. O'SULLIVAN, J., HERAVIZADEH, M., EDMOND, D., HOFSTEDE, TER A., *Towards a Semantic Framework for Service Description*, in Proc. of the 9th Int. Conf. on Database Semantics, Hong-Kong, 2001. Kluwer Academic Publishers. Disponível em: <http://citeseer.nj.nec.com/455961.html>. Último acesso em 06 mar 2003.

EbXML Registry, 2003. Disponível em <http://www.ebxml.org>. Último acesso em 08 mar 2003.

EIBACH, W., KUEBLER, D., *Metering and accounting for Web services, A dynamic e-business solution*, 2001. Disponível em: <http://www-106.ibm.com/library/ws-maws/?dwzone=webservices>. Último acesso em 06 mar 2003.

EJB, SUN Microsystems, 2001, *Enterprise JavaBeans Specification 2.0*. Disponível em: <http://java.sun.com/products/ejb/docs.html>. Último acesso em 20 abr 2003.

EJSE, *Weather Services from EJSE, Inc. Forecasts for a valid U.S. zipcode and 13 Iraq cities*, 2004. Disponível em: <http://www.ejse.com/WeatherService/>. Último acesso em 01 mar 2004.

ERMOLAYEV, V. *Towards Cooperative Distributed Service Composition on the Semantic Web*. IFI, University of Klagenfurt, Austria, 2003.

FALBO, R. A. *Integração de conhecimento em um ambiente de desenvolvimento de software*. Rio de Janeiro: COPPE/UFRJ, 1998. 81 p. Tese de D.Sc.

FELLNER, K.J., TUROWSKI, K., *Classification Framework for Business Components*, Proceedings of the 33rd Hawaii International Conference on System Sciences, Hawaii , 2000. Disponível em: <http://www.computer.org/proceedings/hicss/0493/04938/04938047.pdf?SMSESSION=NO>. Último acesso em 03 mar 2003.

FENSEL, D.; BENJAMINS, V.R.; MOTTA, E.; WIELINGA, B *UPML: A Framework for knowledge system reuse*. Proceedings of the International Joint Conference on AI (IJCAI-99), Stockholm, Sweden, 1999. Disponível em: <http://www.cs.vu.nl/~upml/>. Último acesso em 14 dez 2003.

FENSEL, D. *The semantic Web and its languages*. IEEE Intelligent Systems, 2000a. Disponível em <http://www.cs.man.ac.uk/~horrocks/Publications/download/2000/faqs-on-oil.pdf> . Último acesso em 10 nov 2003.

FENSEL, D. *Ontologies: Silver Bullet for knowledge Management and Eletronic Commerce*, 2000b, Springer-Verlag, Berlin. Disponível em <http://www.cs.vu.nl/~dieter>. Último acesso em 12 nov 2003.

FETTKE, P., LOOS, P. *Specification of Business Components, Objects, Components, Architectures, Services, and Applications for a Networked World* , International Conference NetObjectDays, NODe 2002, Erfurt, Germany, 2002. Disponível em: <http://www.netobjectdays.org/pdf/02/papers/node/0071.pdf>. Último acesso em 07 mar 2003.

FONER, L.N. *A Multi-Agent Referral System for Matchmaking*, The First International Conference on the Practical Applications of Intelligent Agents and Multi-Agent Technology, London, UK, 1996. Disponível em: <http://citeseer.nj.nec.com/foner96multiagent.html>. Último acesso em: 08/03/2003.

FROLUND, S., KOISTINEN, J. *Quality-of-service specifications in distributed object systems*, Distributed Systems Engineering, IEEE, n. 5, pp. 179-202, UK, 1998.

FRANKEL, D.; PARODI, J. *Using Model-Driven Architecture to Develop Web Services*, White paper, IONA Technologies PLC, Second Edition, April, 2002.

GARCIA, S.S., *Processo de Criação de Ontologias em Ciência da Computação: usando teorias e conceitos de Ciência da Informação*. Tese de Doutorado, UFRJ, 2004, a ser publicada.

GERKE, J.; FLURY, P.; STILLER, B. *The Design of a Charging and Accounting System for the Internet*, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology (ETH) Zurich, Version 1, TIK-Report N. 91, 2000.

GISOLFI, D. *Web Services Architect Part 1: An Introduction to Dynamic e-Business*, Disponível em: <http://www-106.ibm.com/developerworks/webservices/library/ws-arc1/>, 2001. Último acesso em 04 abr 2003.

GO, *The Gene Ontology Consortium*, 2004. Disponível em: <http://www.geneontology.org/>. Último acesso em 03 jun 2004.

GOMES, H. E. (coord.) *Manual de Elaboração de Tesouros Monolíngües*. Brasília, Programa Nacional de Bibliotecas das Instituições de Ensino Superior, 1990.

GUARINO, N. *Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration*, 1997, in: M. Pazienza, (Ed.) *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*, International Summer School, SCIE-97, Frascati, Italy, pp. 139-170.

GUARINO, N. *Formal ontology and information systems*. In: FOIS '98, 1, 1998a, Trento, Italy. Proceedings. Amsterdam: IOS Press; Tokyo: Omsa, 1998a. p. 3-15.

GUARINO, N. *Some Ontological Principles for Designing Upper Level Lexical Resources*, First International Conference on Language Resources and Evaluation, Granada, Spain, May, 1998b.

GUARINO, N. *Concepts, Attributes and Arbitrary Relations: Some Linguistic and Ontological Criteria for Structuring Knowledge Bases*, Data & Knowledge Engineering, pp. 249-261, 1992.

GRAHAM, S., SIMEONOV, S., BOUBEZ, T., DAVIS, D., DANIELS, G., NAKAMURA, Y. & NEYAMA, R., *Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI*, Sams Publishing, Indianapolis, 2002.

GRUBER, T. R. *Toward principles for the design of ontologies used for knowledge sharing*. International Journal of Human-Computer Studies, special issue on Formal Ontology in Conceptual Analysis and Knowledge Representation, 1993a. Disponível em: <http://citeseer.nj.nec.com/gruber93toward.html>. Último acesso em: 19 dez 2002.

GRUBER, T. R., *A Translation Approach to Portable Ontology Specifications*. Knowledge Acquisition, 1993b, v. 5, n.2, pp. 199-220. Disponível em: <http://www-ksl.stanford.edu/people/gruber/publications.html>. Último acesso em: 15 mai 2003.

GWEATHER, *Global Weather - Current weather and weather conditions for major cities around the world*, 2004. Disponível em: <http://www.webservicex.net/WS/default.aspx>. Último acesso em 16 mai 2004.

HAN, J., *An Approach to Software Component Specification*, Proceedings of 1999 International Workshop on Component Based Software Engineering, Los Angeles, USA, May 1999. Disponível em: www.sei.cmu.edu/cbs/icse99/papers/25/25.pdf. Último acesso em 05 mar 2003.

HARMELEN, F.; PATEL-SCHNEIDER; P. F, HORROCKS, I. *Reference description of the DAML+OIL language, 2001*. Disponível em: <http://www.daml.org/2001/03/reference.html>. Último acesso em 08 mar 2003.

HENDLER, J., *Agents and the Semantic Web*, IEEE Intelligent Systems Journal, 2002, disponível em <http://www.cs.edu/users/hendler/AgentWeb.html>. Último acesso em 11 jul 2002.

HENDLER, J., MCGUINNESS, D., *The DARPA Agent Markup Language*, IEEE Intelligent Systems, 15, N. 6, pp. 67-73, 2000.

HENDRICKS, M.; GALBRAITH, B.; IRANI, R.; MILBERNY, J.; MODI, T.; TOST, A.; TOUSAAINT, A.; BASHA, J.; CABLE, S. *Professional Java Web Services*, Editora Alta Books, Rio de Janeiro, 2002.

HIRST, G. *Ontology and the Lexicon*. In Staab, Steffen and Studer, Rudi (eds) *Handbook on Ontologies in Information Systems*, Berlin: Springer, 2004, pp. 209-229.

IBM Corporation Homepage, 2004. Disponível em: <http://www.ibm.com/us/>. Último acesso em 26 jun 2004.

IBM UDDI *Business Test Registry*, 2003. Disponível em: <https://uddi.ibm.com/testregistry/find>.

IBM WSAT, Web Services Architecture Team, *Web Services Architecture Overview - The next stage of evolution for e-business*, 2000. Disponível em: <http://www-106.ibm.com/developerworks/webservices/library/w-ovr/>. Último acesso em 07 mar 2002.

IKEDA, M.; SETA, K.; KAKUSHO, O.; MIZOGUCHI, R. *An Ontology for Building A Conceptual Problem Solving Model*, Proceedings of ECAI98 Workshop on Applications of Ontologies and Problem-solving Model, pp. 126-133, 1998.

ISO-704. *Principles and Methods of Terminology*, 1987. Disponível em: <http://www.iso.ch/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=31696&ICS1=1>. Último acesso em 10 nov 2003.

ISO 9126, *Information technology -- Software product evaluation -- Quality characteristics and guidelines for their use*, 1991, ISO, Geneva.

JACOBSON, I., Griss, M., Jonsson, P., *Software Reuse, Architecture, Process and Organization for Business Success*, 1997. ACM Press.

JAVASTRUTS, *The Apache Struts Framework*, 2004. Disponível em: <http://jakarta.apache.org/struts/>. Último acesso em 17 mai 2004.

JENA - *A Semantic Web Framework for Java*, 2004. Disponível em: <http://jena.sourceforge.net/>. Último acesso em 17 mai 2004.

JINI, *Jini Network Technology*, 2003. Disponível em: <http://www.sun.com/software/jini/>. Último acesso em 12 abr 2003.

KELLER, A.; LUDWIG, H., *The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services*. IBM Research Report RC22456. Yorktown Heights, 2002. Disponível em: <http://www.research.ibm.com/people/h/hludwig/publications/publications.html>. Último acesso em 15 mai 2003.

KONEN, W./ KLAPSING, R., *Exchanging semantics with RDF*, 2001. Disponível em: <http://citeseer.nj.nec.com/442959.html>. Último acesso em 08/03/2003.

LARA, R.; LAUSEN, H.; ARROYO, S.; BRUIJN, J.; FENSEL, D. *Semantic Web Services: description requirements and current technologies*. International Workshop on Electronic Commerce, Agents, and Semantic Web Services, In conjunction with the Fifth International Conference on Electronic Commerce (ICEC 2003), Pittsburgh, PA, 2003.

LEACH, R.J., *Software reuse: methods, models, and cost*. McGraw-Hill, new York, 1997.

LÉGER, A., MICHEL, G., BARRETT, P., GITTON, S., GOMÉZ-PÉREZ, A., LEHTONA, A., MOKKILA, K., RODRIGEZ, S., SALLENIN, J., VARVARIGOU, T., VINESSE, J., *Multilingual Knowledge Based European Electronic Market place*, 2000. Disponível em: <http://www.mkbeem.com>. Último acesso em 07/03/2003.

LENCI, A. *Building an Ontology for the Lexicon: Semantic Types and Word Meaning*, Workshop on Ontology-Based Interpretation of Noun Phrases, Kolding-Denmark, 2000.

LEYMANN, F. *Web Services Flow Language (WSFL) Version 1.0*, 2001. Disponível em: www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf. Último acesso em 08 mar 2003.

MANI, A., NAGARAJAN, A. *Understanding quality of service for Web services - Improving the performance of your Web services*, 2002. Disponível em: <http://www106.ibm.com/developerworks/library/ws-quality.html>. Último acesso em 20 dez 2003.

MAXIMILIEN, E. M., SINGH, M., P. *Reputation and Endorsement for Web Services*. ACM SIGecom Exchanges, v. 3, n. 1, pp. 24 - 31, 2001.

MAXIMILIEN, E. M., SINGH, M., P. *Conceptual Model of Web Service Reputation*. ACM SIGMOD Record, v.31, n.4, 2002.

MELING, R.; MONTGOMERY, E.J.; PONNUSAMY, P.S.; WONG, E.B.; MEHANDJISKA, D. *Storing and Retrieving Software Components: A Component Description Manager*, Australian Software Engineering Conference, 2000, Gold Coast, Queensland, Australia. Disponível em: <http://www.computer.org/proceedings/aswec/0631/06310107abs.htm?SMSESSION=N> O. Último acesso em 03 mar 2003.

MICROSOFT Corporation Homepage, 2004. Disponível em: <http://www.microsoft.com>. Último acesso em 26 mai 2004.

MIKALSEN, T., ROUVELLOU, I., TAI, S., *Reliability of Composed Web Services - From Object Transactions to Web Transactions*, 2001. Disponível em: http://www.research.ibm.com/AEM/pubs/web_services_oopsla2001.pdf. Último acesso em 03 mar 2003.

MIKALSEN, T.; TAI, S.; ROUVELLOU, I. *Transactional Attitudes: Reliable Composition of Autonomous Web Services*, 2002. Disponível em: <http://www.research.ibm.com/AEM/pubs/wstx-WDMS-DSN2002.pdf>. Último acesso em 06/02/2003.

MILES, S.; PAPAY, J.; DIALANI, V.; LUCK, M.; DECKER, K.; PAYNE, T.; MOREAU, L. *Personalised Grid Service Discovery* in Stephen A. Jarvis, editor, Nineteenth Annual UK Performance Engineering Workshop (UKPEW 39;03), pp. 131-140, University of Warwick, Coventry, England, 2003.

MILLER, G. A. *WordNet: a lexical database for English*. In: Communications of the ACM 38 (11), November 1995, pp. 39 - 41. Disponível em: <http://www.acm.org/pubs/articles/journals/cacm/1995-38-11/p39-miller/p39-miller.pdf>. Último acesso em 04 jun 2004.

MIZOGUCHI, R.; VANWELKENHUYSEN, J.; IKEDA, M. *Task Ontology for reuse of problem solving knowledge*. In N.J.I. Mars. Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing. IOS Press, 1995.

MODI, T., *WSIL: Do we need another Web Services Specification?*, 2002, Web Services Architect, Jan, 2002. Disponível em: <http://www.webservicesarchitect.com/content/articles/modi01.asp>. Último acesso em: 01 abr 2003.

MOTTA, E.; FENSEL, D.; GASPARI, M.; BENJAMINS, R. *Specifications of Knowledge Components for Reuse*. Eleventh International Conference on Software Engineering and Knowledge Engineering (SEKE '99) , 1999.

MYERSON, J., M., *Guarantee your Web service with an SLA Introduction, architecture, and testing mechanisms*, 2002. Disponível em: <http://www-106.ibm.com/developerworks/webservices/library/ws-sla/>. Último acesso em 07/03/2003.

MYERSON, J., M., *Web services architectures – how they stack up*, 2003. Disponível em: <http://www.webservicesarchitect.com/content/articles/myerson01.asp>. Último acesso em 08 mar 2003.

NAICS - *North American Industry Classification System*, 2003. Disponível em: <http://www.census.gov/epcd/www/naics.html>. Último acesso em: 26 mai 2004.

NAKKRASAE, S. ; SOPHATSATHIT, P., *A formal approach for specification and classification of software components*, Proceedings of the 14th international conference on Software engineering and knowledge engineering, 2002, Ischia, Italy.

NILES, I., PEASE, A. *Towards a Standard Upper Ontology*, in Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001), Chris Welty and Barry Smith, eds, Ogunquit, Maine, 2001.

NOY, N. F., MCGUINNESS, D L., *Ontology Development 101: A Guide to Creating Your First Ontology*, 2000. Disponível em: http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html. Último acesso em 26 mai 2004.

OWL - *Web Ontology Language*, 2003. Disponível em: <http://www.w3.org/TR/2003/PR-owl-ref-20031215/>. Último acesso em 22 jul 2003.

OWL-S, *OWL-based Web Service Ontology (OWL-S)*, 2003. Disponível em: <http://www.daml.org/services/owl-s/1.0/>. Último acesso em 10 mai 2004.

PAOLUCCI, M., KAWAMURA, T., PAYNE, T., R., SYCARA, K., *Semantic Matching of Web Services Capabilities*, First Int. Semantic Web Conference, ISWC2002, Cagliari, Italy, 2002.

PATEL, K., *XML Grammar and Parser for the Web Service Offerings Language*, Master of Applied Science in Electrical Engineering Thesis, Carleton University, Ottawa, Ontario, Canada, 2003.

PENIX, J., BARAONA, P., ALEXANDER, P., *Classification and Retrieval of Reusable Components Using Semantic Features*, The 10th Knowledge-Based Software Engineering Conference, 1995, Boston, Massachusetts. Disponível em <http://www.computer.org/proceedings/kbse/7204/72040131abs.htm>. Último acesso em 08 mar 2003.

PENIX, J., ALEXANDER, P., *Using Formal Specifications for Component Retrieval and Reuse*. Proceedings of the Thirty-First Hawaii International Conference on System Sciences, v.3, pp.356-65, Hawaii , 1998.

PERICH, F.; CHAKRABORTY, D., *Initial ontology for D-Reggie, a DAML-ized version of Reggie/Jini framework*, 2001. Disponível em: <http://www.daml.org/ontologies/153>. Último acesso em 07 mar 2003.

PIRES, P., F., *WebTransact: A Framework for Specifying and Coordinating Reliable Web Services Composition*. Tese de D. Sc., Relat. Tec. ES-578/02, COPPE/UFRJ, Brasil, 2002.

PODGURSKI, A.; PIERCE, L. *Behaviour sampling: A technique for automated retrieval of reusable components*. In Proceedings of the 14th International Conference on Software Engineering, pages 349-360, 1992.

PRIETO-DIAZ, R., *Domain Analysis: An Introduction*. ACM SIGSoft Software Engineering Notes, v.15, n. 2, pp. 47-54, 1990. Disponível em: <http://www.cs.jmu.edu/users/prietorx/RubenPubs/RubenPubs.htm>. Último acesso em 15 mai 2003.

PRIETO-DIAZ, R., *Implementing faceted classification for software reuse*, Communications of the ACM, Special issue on Software engineering, v. 34, n. 5, pp. 88-97, 1991. Disponível em: <http://portal.acm.org/citation.cfm?doid=103167.103176>. Último acesso em 07 mar 2003.

RAN, S., *A Model for Web Services Discovery With QoS*, ACM SIGecom Exchanges, v.4, n.1, March 2003.

RANGANATHAN, S. R. *Prolegomena to library classification*. Bombay: Asia Publ. House, 1967.

RDQL, *A Query language for RDF - w3c*, 2004. Disponível em: <http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/>.

ROMMEL, 2001 Rommel J., *Will Web Services jump-start the software slump?* Java World, out. 2001. Disponível em: http://www.javaworld.com/javaworld/jw-08-2001/jw-0831-webservice_p.html. Último acesso em: 24 mai. 2002.

RULEML, The Rule Markup Initiative, 2004. Disponível em: <http://www.ruleml.org>. Último acesso em 02/07/2004.

SAHAI, A., DURANTE, A., MACHIRAJU, V., *Towards Automated SLA Management for Web Services*. Software Technology Laboratory, HP Laboratories Palo Alto, July, 2002. Disponível em: Último acesso em: 20 abr 2003.

SAMETINGER, J., *On a Taxonomy for Software Components*, Workshop on Component-Oriented Programming (WCOP-96), Linz, July 1996. Disponível em: <http://www.swe.uni-linz.ac.at/publications/abstract/TR-SE-96.11.html>. Último acesso em 12 mai 2003.

SAS Corporation Homepage, 2004. Disponível em: <http://www.sas.com>. Último acesso em 26 mai 2004.

SEELY, S., *Web Services Description and Discovery Using UDDI, Part II*, 2001. Disponível em: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnservice/html/service10172001.asp>. Último acesso em 26 mai 2004.

SHAIKHALI, A., RANA, O. F. , AL-ALI, R., WALKER, D.W. *UDDIe: An Extended Registry for Web Services*, Symposium on Applications and the Internet Workshops (SAINT'03 Workshops) , IEEE, 2003, Orlando, Florida.

SOAP Specification – W3C, 2003. Disponível em: <http://www.w3.org/TR/soap/>. Último acesso em: 26 mai 2004.

SOLLAZZO, T., HANDSCHUH, S., STAAB, S., FRANK, M. *Semantic Web Service Architecture - Evolving Web Service Standards toward the Semantic Web*. Proc. of the 15th International FLAIRS Conference. Pensacola, Florida, May 16-18, 2002. AAAI Press. Disponível em: <http://www.aifb.uni-karlsruhe.de/WBS/sst/Research/Publications/sub-flairs2002.pdf>. Último acesso em: 07 mar 2003.

STILLER, B., FANKHAUSER, G., PLATTNER, B., WEILER, N., *Charging and accounting for integrated internet services - state of the art, problems, and trends*. In Proceedings of INET '98, Geneva, Switzerland, July 1998. Disponível em: <http://citeseer.nj.nec.com/stiller98charging.html>. Último acesso em 02 abr 2003.

STILLER, B., GERKE, J., REICHL, P., FLURY, P.: *A Generic and Modular Internet Charging System for the Cumulus Pricing Scheme*. Journal of Network and Systems Management, v. 3, n. 9, pp. 293-325, 2001.

SUO WG, Standard Upper Ontology Working Group, *Standard Upper Ontology* - Disponível em: <http://suo.ieee.org/>. Último acesso em 12 mai 2004.

SYCARA, K. et al., *LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace*, Autonomous Agents and Multi-Agents Systems, v.5, pp.173-203, 2002.

SZYPERSKI, C., *Component Software*, Addison Wesley, 1997.

TEHAN, R., *Internet Statistics: Explanation and Sources*, 2002, Congressional Research Service - The Library of Congress. Disponível em: <http://164.109.48.86/usa/infousa/tech/reports/rl31270.pdf>. Último acesso em: 07 mar 2003.

TIDWELL, D., *Web services: the Web's next revolution*, November 2000. Disponível em: <http://www-105.ibm.com/developerworks/education.nsf/webservices-onlinecourse-bytitle/BA84142372686CFB862569A400601C18?OpenDocument>. Último acesso em 02 fev 2003.

TOMCAT, *Apache TOMCAT*, 2004. Disponível em: <http://jakarta.apache.org/tomcat/>. Último acesso em 05 jan 2004.

TOSIC, V., PATEL, K., PAGUREK, B. *WSOL - Web Service Offerings Language*. In Proc. of the Workshop on Web Services, e-Business, and the Semantic Web (WES) at CaiSE'02 Toronto, Canada, May 2002. Springer-Verlag, Lecture Notes in Computer Science. Disponível em: <http://citeseer.nj.nec.com/550601.html>. Último acesso em 07 abr 2003.

TOSIC, W., PATEL, K., PAGUREK, B., *Reusability Constructs in the Web Service Offerings Language (WSOL)*, Research Report SCE-03-08, april 2003. Disponível em: Último acesso em: 20 abr 2003.

TSALGATIDOU, A.; PILIOURA, T. *An Overview of Standards and Related Technology in Web Services*, International Journal of Distributed and Parallel Databases, Special Issue on E-Services, v.12, n.2, pp.135-162, 2002. Disponível em: <http://cgi.di.uoa.gr/~afrodite/publications.html>. Último acesso em 14 mai 2003.

TUDHOPE, D., BINDING, C., BLOCKS, D., CUNLIFFE, D. *Representation and Retrieval in Faceted Systems : Challenges in knowledge representation and organization for the 21st century: integration of knowledge across boundaries*: Proceedings of the the Seventh International ISKO Conference, Granada, Spain, 2002. Advances in Knowledge Organization; v. 8, pp. 191-197, 2002.

UDDIORG, Uddi.org Homepage, 2004. Disponível em : <http://www.uddi.org/>. Último acesso em : 26 mai 2004.

UDDI Specification, 2001. Disponível em: <http://www.uddi.org/specification.html>. Último acesso em : 26 mai 2004.

UDDIEXEC, UDDI Executive White Paper. Nov, 14, 2001. Disponível em: www.uddi.org/pubs/UDDI_Executive_White_Paper.pdf. Último acesso em 07 mar 2003.

VARADARAJAN, S., KUMAR, A., GUPTA, D., JALOTE, P. *ComponentXchange: An E_Exchange for Software Components*, 2002. Disponível em: <http://citeseer.nj.nec.com/sharma02componentxchange.html>. Último acesso em 20 abr 2003.

VASUDEVAN, V., *A Web Services Primer*, apr. 2001 Disponível em: <http://www.xml.com/pub/a/2001/04/04/webservices/index.html>. Último acesso em 23 jul. 2002.

VILLELA, R.M.B., *Busca e reutilização de componentes em ambientes de reutilização de software*. COPPE / UFRJ, Tese de D. Sc., Engenharia de Sistemas e Computação, 2000.

WELTY, C., GUARINO, N., *Supporting Ontological Analysis of taxonomic relationships*. Elsevier Data and Knowledge Engineering 39 (2001) 51-74. Disponível em: <http://ontology.ip.rm.cnr.it/Papers/dke2001.pdf>. Último acesso em 12 dez 2002.

WIKIPEDIA, *Wikipedia Encyclopedia*, 2004. Disponível em: <http://www.wikipedia.org>.

WOOD, M.; SOMMERVILLE, I. *An Information Retrieval System for Software Components*, in SIGIR Forum, v.22, n.3, pp 11-25, 1988.

WROE, C; STEVENS, R.; GOBLE, C.; ROBERTS, A.; GREENWOOD, M. *A suite of DAML+OIL ontologies to describe bioinformatics web services and data*. International Journal of Cooperative Information Systems, v.12, n. 2 , pp.197-224, 2003.

WSDEF, *Web Services Architecture Requirements*, W3C Working Draft 19 August 2002. Disponível em: <http://www.w3.org/TR/2002/WD-wsa-reqs-20020819#IDAIO2IB>. Último acesso em 10 jul 2004.

WSDL, *WSDL Specification*, 2001 – W3C. Disponível em: www.w3.org/TR/wsdl.

WSEL, *Web Services Endpoint Language* - W3C, 2001. Disponível em: http://www.w3.org/2001/04/wsws-proceedings/rod_smith/img13.htm. Último acesso em 12 jan 2003.

WSRELIA, *Reliable Message Delivery in a Web Services World: A Proposed Architecture and Roadmap*, march 2003, Disponível em: <http://msdn.microsoft.com/webservices/default.aspx?pull=/library/en-us/dnglobspec/html/ws-rm-exec-summary.asp>. Último acesso em 05 mar 2003.

XAML, *Transaction Authority Markup Language*, 2000. Disponível em: <http://www.oasis-open.org/cover/xaml.html>. Último acesso em 17 mai 2003.

XML - *Extensible Markup Language*, 2003 – W3C. Disponível em <http://www.w3.org/XML/>.

XKMS, *XML Key Management Specification*, 2001. Disponível em: <http://www.w3.org/TR/xkms/>. Último acesso em: 26 mai 2004.

XMETHODS, *Web Services Directory Site*, 2004. Disponível em: www.xmethods.com.

XMLENC, *XML Encryption WG*, 2001. Disponível em: <http://www.w3.org/Encryption/2001/>. Último acesso em: 26 mai 2004.

YACOUB, S., AMMAR, H., MILI, A.: *Characterizing a Software Component*. Disponível em: <http://www.sei.cmu.edu/cbs/icse99/papers/icse99-papers.pdf>. Último acesso em 02 abr 2003.

YU, B., SINGH, M.P., *Distributed Reputation Management for Electronic Commerce*. Computational Intelligence v.18, n.4, pp. 535-549, 2002.

ZACHARIA, G., MAES, P., *Trust Management Through Reputation Mechanisms*, Applied Artificial Intelligence, v.14, pp: 881-907, 2002.

ZAREMSKI, A.M., WING, J.M., *Specification Matching of Software Components*, ACM Transactions on Software Engineering and methodology, v. 6, n. 4, pp. 333-369, 1997. Disponível em: <http://citeseer.nj.nec.com/zaremski95specification.html>. Último acesso em 08 mar 2003.

ZAREMSKI, A.M., WING, J.M., *Signature Matching: A Tool for Using Software Libraries*, ACM Transactions on Software Engineering and Methodology, v. 4, n. 2, pp. 146-170, 1995. Disponível em: <http://citeseer.nj.nec.com/zaremski95specification.html>. Último acesso em 05/03/2003.

Anexos

A-1 – Exemplo de Documento WSDL

```

<types>
  <schema targetNamespace="http://example.com/stockquote.xsd"
    xmlns="http://www.w3.org/1999/XMLSchema">
    <element name="GetLastTradePrice">
      <complexType>
        <all> <element name="symbol" type="string"/> </all>
      </complexType>
    </element>
    <element name="GetLastTradePriceResponse">
      <complexType>
        <all> <element name="Price" type="float"/> </all>
      </complexType>
    </element>
  </schema>
</types>

<message name="GetLastTradePriceInput">
  <part name="body" element="xsd1:GetLastTradePrice"/>
</message>
<message name="GetLastTradePriceOutput">
  <part name="body" element="xsd1:GetLastTradePriceResponse"/>
</message>

<portType name="StockQuotePortType">
  <operation name="GetLastTradePrice">
    <input message="tns:GetLastTradePriceInput"/>
    <output message="tns:GetLastTradePriceOutput"/>
  </operation>
</portType>

<binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="GetLastTradePrice">
    <soap:operation soapAction="http://example.com/stockquote.xsd"/>
    <input>
      <soap:body use="encoded" namespace="http://example.com/stockquote.xsd"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </input>
    <output>
      <soap:body use="encoded" namespace="http://example.com/stockquote.xsd"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </output>
  </operation>
</binding>

<service name="StockQuoteService">
  <documentation>My first service</documentation>
  <port name="StockQuotePort" binding="tns:StockQuoteBinding">
    <soap:address
      location="http://www.stockquoteserver.com/StockQuote"/>
  </port>
</service>

```

A-2 – Exemplo de Documento WSOL

```

<wsol:serviceOffering name = "SO1" service = "buyStock:buyStockService"
  accountingParty = "WSOL-SUPPLIERWS" >
  <wsol:constraint name = "QoScons2" xsi:type = "qosSchema:QoSconstraint"
    service = "WSOL-ANY" portOrPortType = "WSOL-EVERY" operation =
      "WSOL-EVERY" >
    <expressionSchema:booleanExpression>
      <expressionSchema:arithmeticWithUnitExpression>
        <expressionSchema:QoSmetric metricType = "QoSMetricOntology:
          ResponseTime" service = "WSOL-ANY" portOrPortType = "WSOL-ANY"
          operation = "WSOL-ANY" measuredBy = "WSOL_INTERNAL" />
        </expressionSchema:arithmeticWithUnitExpression>

        <expressionSchema:arithmeticComparator type = "&lt;" />

        <expressionSchema:arithmeticWithUnitExpression>
          <wsol:numberWithUnitConstant>
            <wsol:value> 0.3 </wsol:value>
            <wsol:unit type = "QoSMeasOntology:second" />
          </wsol:numberWithUnitConstant>
        </expressionSchema:arithmeticWithUnitExpression>
      </expressionSchema:booleanExpression>
    </wsol:constraint>

    ...
    <wsol:managementResponsibility name = "MangResp1" >
      <wsol:supplierResponsibility scope = "tns:AccRght1" />
      <wsol:consumerResponsibility scope = "tns:Precond3" />
      <wsol:independentResponsibility scope = "tns:QoScons2" entity =
        "http://www.someThirdParty.com" />
    </wsol:managementResponsibility>
  </wsol:serviceOffering>

```

A-3 – Ontologia DReggie

```

<?xml version="1.0" ?>
: <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
  xmlns="http://daml.umbc.edu/ontologies/dreggie-ont#">
: <daml:Ontology rdf:about="">
  <daml:versionInfo>Filip Perich and Dipanjan Chakraborty
  </daml:versionInfo>
  <rdfs:comment>Anamika Primary Ontology</rdfs:comment>
  <daml:imports rdf:resource="http://daml.umbc.edu/ontologies/dreggie-ont" />
</daml:Ontology>
: <daml:Class rdf:ID="Component">
  <rdfs:label>Component</rdfs:label>
</daml:Class>
: <daml:Property rdf:ID="Description">
  <rdfs:domain rdf:resource="#Component" />
  <rdfs:range rdf:resource="#DescriptionClass" />
</daml:Property>
<daml:Class rdf:ID="DescriptionClass" />
: <daml:Property rdf:ID="ServiceName">
  <rdfs:domain rdf:resource="#DescriptionClass" />
  <rdfs:range rdf:resource="http://www.daml.org/2001/03/daml+oil#Literal" />
</daml:Property>
: <daml:Property rdf:ID="ServiceAlias">
  <rdfs:domain rdf:resource="#DescriptionClass" />
  <rdfs:range rdf:resource="http://www.daml.org/2001/03/daml+oil#Literal" />
</daml:Property>
: <daml:Property rdf:ID="ClientName">
  <rdfs:domain rdf:resource="#DescriptionClass" />
  <rdfs:range rdf:resource="http://www.daml.org/2001/03/daml+oil#Literal" />
</daml:Property>
: <daml:Property rdf:ID="Capability">
  <rdfs:domain rdf:resource="#DescriptionClass" />
  <rdfs:range rdf:resource="#CapabilityClass" />
</daml:Property>
: <daml:Property rdf:ID="Requirements">
  <rdfs:domain rdf:resource="#DescriptionClass" />
  <rdfs:range rdf:resource="#RequirementsClass" />
</daml:Property>

```

```

- <daml:Property rdf:ID="Cost">
  <rdfs:domain rdf:resource="#DescriptionClass" />
  <rdfs:range rdf:resource="#CostClass" />
</daml:Property>
- <daml:Property rdf:ID="Mobility">
  <rdfs:domain rdf:resource="#DescriptionClass" />
  <rdfs:range rdf:resource="#MobilityClass" />
</daml:Property>
- <daml:Property rdf:ID="Input">
  <rdfs:domain rdf:resource="#DescriptionClass" />
  <rdfs:range rdf:resource="#InputClass" />
</daml:Property>
- <daml:Property rdf:ID="Output">
  <rdfs:domain rdf:resource="#DescriptionClass" />
  <rdfs:range rdf:resource="#OutputClass" />
</daml:Property>
<daml:Class rdf:ID="CapabilityClass" />
- <daml:Property rdf:ID="ClientCapability">
  <rdfs:domain rdf:resource="#CapabilityClass" />
  <rdfs:range rdf:resource="#Type" />
</daml:Property>
- <daml:Property rdf:ID="ServiceCapability">
  <rdfs:domain rdf:resource="#CapabilityClass" />
  <rdfs:range rdf:resource="#Type" />
</daml:Property>
<daml:Class rdf:ID="RequirementsClass" />
- <daml:Property rdf:ID="CPURequirement">
  <rdfs:domain rdf:resource="#RequirementsClass" />
</daml:Property>
- <daml:Property rdf:ID="MemoryRequirement">
  <rdfs:domain rdf:resource="#RequirementsClass" />
</daml:Property>
- <daml:Property rdf:ID="DiskRequirement">
  <rdfs:domain rdf:resource="#RequirementsClass" />
</daml:Property>
- <daml:Property rdf:ID="OSRequirement">
  <rdfs:domain rdf:resource="#RequirementsClass" />
  <rdfs:range rdf:resource="#VersionName" />
</daml:Property>
- <daml:Property rdf:ID="SoftwareRequirement">
  <rdfs:domain rdf:resource="#RequirementsClass" />
  <rdfs:range rdf:resource="#VersionName" />
</daml:Property>
<daml:Class rdf:ID="CostClass" />
- <daml:Property rdf:ID="Local">
  <rdfs:domain rdf:resource="#CostClass" />
  <rdfs:range rdf:resource="#AmountUnit" />
</daml:Property>
- <daml:Property rdf:ID="Remote">
  <rdfs:domain rdf:resource="#CostClass" />
  <rdfs:range rdf:resource="#AmountUnit" />
</daml:Property>

```

```

<daml:Class rdf:ID="MobilityClass" />
: <daml:Property rdf:ID="ClientMobility">
  <rdfs:domain rdf:resource="#MobilityClass" />
  <rdfs:range rdf:resource="#Type" />
</daml:Property>
: <daml:Property rdf:ID="ServiceMobility">
  <rdfs:domain rdf:resource="#MobilityClass" />
  <rdfs:range rdf:resource="#Type" />
</daml:Property>
<daml:Class rdf:ID="InputClass" />
: <daml:Property rdf:ID="ServiceInputType">
  <rdfs:domain rdf:resource="#InputClass" />
  <rdfs:range rdf:resource="#Type" />
</daml:Property>
: <daml:Property rdf:ID="ClientInputType">
  <rdfs:domain rdf:resource="#InputClass" />
  <rdfs:range rdf:resource="#Type" />
</daml:Property>
<daml:Class rdf:ID="OutputClass" />
: <daml:Property rdf:ID="ServiceOutputType">
  <rdfs:domain rdf:resource="#OutputClass" />
  <rdfs:range rdf:resource="#Type" />
</daml:Property>
: <daml:Property rdf:ID="ClientOutputType">
  <rdfs:domain rdf:resource="#OutputClass" />
  <rdfs:range rdf:resource="#Type" />
</daml:Property>
: <daml:Property rdf:ID="Property">
  <rdfs:domain rdf:resource="#Component" />
  <rdfs:range rdf:resource="#PropertyClass" />
</daml:Property>
<daml:Class rdf:ID="PropertyClass" />
: <daml:Property rdf:ID="ServiceProperty">
  <rdfs:domain rdf:resource="#PropertyClass" />
  <rdfs:range rdf:resource="#PropertyType" />
</daml:Property>
: <daml:Property rdf:ID="ClientProperty">
  <rdfs:domain rdf:resource="#PropertyClass" />
  <rdfs:range rdf:resource="#PropertyType" />
</daml:Property>
<daml:Class rdf:ID="PropertyType" />
: <daml:Property rdf:ID="Name">
  <rdfs:range rdf:resource="http://www.daml.org/2001/03/daml+oil#Literal" />
</daml:Property>
: <daml:Property rdf:ID="CpuType">
  <rdfs:range rdf:resource="#Type" />
</daml:Property>
: <daml:Property rdf:ID="Availability">
  <rdfs:range rdf:resource="http://www.daml.org/2001/03/daml+oil#Thing" />
</daml:Property>
: <daml:Property rdf:ID="Speed">
  <rdfs:range rdf:resource="#AmountUnit" />
</daml:Property>

```

```

: <daml:Property rdf:ID="Size">
  <rdfs:range rdf:resource="#AmountUnit" />
</daml:Property>
: <daml:Property rdf:ID="Memory">
  <rdfs:range rdf:resource="#AmountUnit" />
</daml:Property>
: <daml:Property rdf:ID="FileSystem">
  <rdfs:range rdf:resource="#AmountUnit" />
</daml:Property>
: <daml:Property rdf:ID="OperatingSystem">
  <rdfs:range rdf:resource="#AmountUnit" />
</daml:Property>
: <daml:Property rdf:ID="Software">
  <rdfs:range rdf:resource="#AmountUnit" />
</daml:Property>
: <daml:Class rdf:ID="AmountUnit">
  <daml:subClassOf
    rdf:resource="http://www.daml.org/2001/03/daml+oil#Literal" />
</daml:Class>
: <daml:Property rdf:ID="Amount">
  <rdfs:range
    rdf:resource="http://www.w3.org/2000/10/XMLSchema#nonNegativeI
neger" />
  <rdfs:domain rdf:resource="#AmountUnit" />
</daml:Property>
: <daml:Property rdf:ID="Unit">
  <rdfs:range rdf:resource="http://www.daml.org/2001/03/daml+oil#Literal"
/>
  <rdfs:domain rdf:resource="#AmountUnit" />
</daml:Property>
: <daml:Class rdf:ID="VersionType">
  <daml:subClassOf rdf:resource="#Type" />
</daml:Class>
: <daml:Class rdf:ID="VersionName">
  <daml:subClassOf
    rdf:resource="http://www.daml.org/2001/03/daml+oil#Thing" />
</daml:Class>
: <daml:Property rdf:ID="Version">
  <rdfs:range rdf:resource="http://www.daml.org/2001/03/daml+oil#Literal"
/>
</daml:Property>
: <daml:Class rdf:ID="Type">
  <daml:subClassOf
    rdf:resource="http://www.daml.org/2001/03/daml+oil#Thing" />
</daml:Class>
</rdf:RDF>

```

A-4 – Exemplo de Serviço Web Descrito em OWL-S

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE uridef (View Source for full doctype...)>
- <!--
  This document uses entity types as a shorthand for URIs.
  Download the source for a version with unexpanded entities.
  -->
- <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:service="http://www.daml.org/services/owl-s/1.0/Service.owl#"
  xmlns:process="http://www.daml.org/services/owl-s/1.0/Process.owl#"
  xmlns:profile="http://www.daml.org/services/owl-s/1.0/Profile.owl#"
  xmlns:actor="http://www.daml.org/services/owl-s/1.0/ActorDefault.owl#"
  xmlns:addParam="http://www.daml.org/services/owl-
s/1.0/ProfileAdditionalParameters.owl#"
  xmlns:profileHierarchy="http://www.daml.org/services/owl-
s/1.0/ProfileHierarchy.owl#" xmlns="http://www.daml.org/services/owl-
s/1.0/BravoAirProfile.owl#">
- <!--
Monika 4/10/03: added namespaces "actor" and "addParam" to take into account
the shift of some of the properties to other namespaces.
-->
- <owl:Ontology rdf:about="">
  <owl:versionInfo>$Id: BravoAirProfile.owl,v 1.19 2003/12/18 02:10:14
  martin Exp $</owl:versionInfo>
  <rdfs:comment>DAML-S Coalition: BravoAir Example for OWL-S Profile
  description</rdfs:comment>
  <owl:imports rdf:resource="http://www.daml.org/services/owl-
s/1.0/Service.owl" />
  <owl:imports rdf:resource="http://www.daml.org/services/owl-
s/1.0/Profile.owl" />
  <owl:imports rdf:resource="http://www.daml.org/services/owl-
s/1.0/ActorDefault.owl" />
  <owl:imports rdf:resource="http://www.daml.org/services/owl-
s/1.0/ProfileAdditionalParameters.owl" />
  <owl:imports rdf:resource="http://www.daml.org/services/owl-
s/1.0/Process.owl" />
  <owl:imports rdf:resource="http://www.daml.org/services/owl-
s/1.0/BravoAirService.owl" />
  <owl:imports rdf:resource="http://www.daml.org/services/owl-
s/1.0/BravoAirProcess.owl" />
  <owl:imports rdf:resource="http://www.daml.org/services/owl-
s/1.0/Concepts.owl" />
  <owl:imports rdf:resource="http://www.daml.org/services/owl-
s/1.0/Country.owl" />
  <owl:imports rdf:resource="http://www.daml.org/services/owl-
s/1.0/ProfileHierarchy.owl" />
</owl:Ontology>

```

```

- <!--
#####
-->
- <!--
# Instance Definition of BravoAir Reservation Agent Advertisement #
-->
- <!--
#####
-->
  : <!--

    BraviAur is defined as a AirlineTicketing service. It inherits from
    the ontology of services that is an eCommerce service, and that
    the product it sells products that are restricted to be
    CommercialAirlineTravel.

    The same service could be specified outside the hierarchy of
    services by declaring it an instance of Profile, and by adjusting
    the relevant properties accordingly.

-->
: <profileHierarchy:AirlineTicketing rdf:ID="Profile_BravoAir_ReservationAgent">
- <!--
  reference to the service specification
-->
  <service:presentedBy rdf:resource="http://www.daml.org/services/owl-
    s/1.0/BravoAirService.owl#BravoAir_ReservationAgent" />
- <!--
  reference to the process model specification
-->
  <profile:has_process rdf:resource="http://www.daml.org/services/owl-
    s/1.0/BravoAirProcess.owl#BravoAir_Process" />
  <profile:serviceName>BravoAir_ReservationAgent</profile:serviceName>
  <profile:textDescription>This service provide flight reservations based on
  the specification of a flight request. This typically involves a
  departure airport, an arrival airport, a departure date, and if a return
  trip is required, a return date. If the desired flight is available, an
  itinerary and reservation number will be
  returned.</profile:textDescription>
  : <!--
  specification of contact information.
  There are two contacts specified here:
  1. to a reservation department
  2. to John Doe that is a sale representative

  The two conctacs are related to the profile through different
  instances of the contactInfo relation

-->

```

```

- <profile:contactInformation>
  - <actor:Actor rdf:ID="BravoAir-reservation">
    <actor:name>BravoAir Reservation department</actor:name>
    <actor:title>Reservation Representative</actor:title>
    <actor:phone>412 268 8780</actor:phone>
    <actor:fax>412 268 5569</actor:fax>
    <actor:email>Bravo@Bravoair.com</actor:email>
    <actor:physicalAddress>Airstrip 2, Teetering Cliff Hights,
      Florida 12321, USA</actor:physicalAddress>
    <actor:webURL>http://www.daml.org/services/daml-
      s/2001/05/BravoAir.html</actor:webURL>
    </actor:Actor>
  </profile:contactInformation>
- <!--
other contact
-->
- <profile:contactInformation>
  - <actor:Actor rdf:ID="BravoAir-information">
    <actor:name>John Doe</actor:name>
    <actor:title>Sale Representative</actor:title>
    <actor:phone>412 268 8789</actor:phone>
    <actor:fax>412 268 5569</actor:fax>
    <actor:email>John_Doe@Bravoair.com</actor:email>
    <actor:physicalAddress>Airstrip 2, Teetering Cliff Hights,
      Florida 12321, USA</actor:physicalAddress>
    <actor:webURL>http://www.daml.org/services/daml-
      s/2001/05/BravoAir.html</actor:webURL>
    </actor:Actor>
  </profile:contactInformation>
  - <!--
description of Geographic radius as a service parameter.
rather than a direct property of profile as in version 0.6
-->
-->
- <profile:serviceParameter>
  - <addParam:GeographicRadius rdf:ID="BravoAir-geographicRadius">
    <profile:serviceParameterName>BravoAir Geographic
      Radius</profile:serviceParameterName>
    <profile:sParameter
      rdf:resource="http://www.daml.org/services/owl-
      s/1.0/Country.owl#UnitedStates" />
    </addParam:GeographicRadius>
  </profile:serviceParameter>
- <!--
specification of quality rating for profile
-->
- <profile:qualityRating>
  - <profile:QualityRating rdf:ID="BravoAir-goodRating">
    <profile:ratingName>SomeRating</profile:ratingName>
    <profile:rating rdf:resource="http://www.daml.org/services/owl-
      s/1.0/Concepts.owl#GoodRating" />
    </profile:QualityRating>
  </profile:qualityRating>
- <!--

```

Specification of the service category using NAICS

```
-->
- <profile:serviceCategory>
  - <addParam:NAICS rdf:ID="NAICS-category">
    <profile:value>Airline reservation services</profile:value>
    <profile:code>561599</profile:code>
  </addParam:NAICS>
</profile:serviceCategory>
- <!--
```

Specification of the service category using UN-SPSC

```
-->
- <profile:serviceCategory>
  - <addParam:UNSPSC rdf:ID="UNSPSC-category">
    <profile:value>Travel Agent</profile:value>
    <profile:code>90121500</profile:code>
  </addParam:UNSPSC>
</profile:serviceCategory>
- <!--
```

Descriptions of IOPEs

```
-->
<profile:hasInput rdf:resource="http://www.daml.org/services/owl-
s/1.0/BravoAirProcess.owl#DepartureAirport_In" />
<profile:hasInput rdf:resource="http://www.daml.org/services/owl-
s/1.0/BravoAirProcess.owl#ArrivalAirport_In" />
<profile:hasInput rdf:resource="http://www.daml.org/services/owl-
s/1.0/BravoAirProcess.owl#OutboundDate_In" />
<profile:hasInput rdf:resource="http://www.daml.org/services/owl-
s/1.0/BravoAirProcess.owl#InboundDate_In" />
<profile:hasInput rdf:resource="http://www.daml.org/services/owl-
s/1.0/BravoAirProcess.owl#RoundTrip_In" />
<profile:hasInput rdf:resource="http://www.daml.org/services/owl-
s/1.0/BravoAirProcess.owl#PreferredFlightItinerary_In" />
<profile:hasOutput rdf:resource="http://www.daml.org/services/owl-
s/1.0/BravoAirProcess.owl#AvailableFlightItineraryList_Out" />
<profile:hasInput rdf:resource="http://www.daml.org/services/owl-
s/1.0/BravoAirProcess.owl#AcctName_In" />
<profile:hasInput rdf:resource="http://www.daml.org/services/owl-
s/1.0/BravoAirProcess.owl#Password_In" />
<profile:hasInput rdf:resource="http://www.daml.org/services/owl-
s/1.0/BravoAirProcess.owl#ReservationID_In" />
<profile:hasInput rdf:resource="http://www.daml.org/services/owl-
s/1.0/BravoAirProcess.owl#Confirm_In" />
<profile:hasOutput rdf:resource="http://www.daml.org/services/owl-
s/1.0/BravoAirProcess.owl#PreferredFlightItinerary_Out" />
<profile:hasOutput rdf:resource="http://www.daml.org/services/owl-
s/1.0/BravoAirProcess.owl#AcctName_Out" />
<profile:hasOutput rdf:resource="http://www.daml.org/services/owl-
s/1.0/BravoAirProcess.owl#ReservationID_Out" />
<profile:hasEffect rdf:resource="http://www.daml.org/services/owl-
s/1.0/BravoAirProcess.owl#HaveSeat" />
</profileHierarchy:AirlineTicketing>
</rdf:RDF>
```

Apêndices

B-1 – Documentação OntoGuide

B-1.1 Elementos de Documentação Oriundos da Etapa 1

Considerando o exposto nesse item, de acordo com nossa proposta, a abordagem adotada para a especificação dos vocabulários auxiliares usados na descrição é:

```
<vocabulario-auxiliar tipo="xxx" estendido="yyy">
  <url>
    Namespace do vocabulário, ou endereço de uma
    página Web onde pode-se achar uma documentação
    sobre ele.
  </url>
  <dominio>
    Descrição textual do domínio ao qual pertence o
    vocabulário.
  </dominio>
  <sub-dominio>
    Descrição textual do sub-domínio ao qual
    pertence o vocabulário.
  </sub-dominio>
</vocabulario-auxiliar >
```

Tipo="xxx" especifica o tipo de vocabulário, onde xxx assume os valores: "tarefa", "domínio" e "genérico", e é baseado na classificação de ontologias de Guarino. Essa representação é usada mesmo nos casos em que o vocabulário usado for uma terminologia e não uma ontologia.

Estendido = "yyy" especifica se o vocabulário utilizado é estendido, não é estendido ou se é incorporado dentro da linguagem, onde yyy assume os valores "verdadeiro", "falso" ou "incorporado", respectivamente.

B-1.2 Elementos de Documentação Oriundos da Etapa 2

De acordo com nossa proposta, os termos da ontologia são documentados em um glossário, como se segue:

```
<glossario>
  <termo>
    <e-um> classe pai ou categoria fundamental <e-um>
    <partes> relacionar os termos parte <partes>
    <usado-para> descrição do uso do termo <usado-para>
  </termo>
</glossario>
```

Onde <termo> deve ser substituído pelo nome do termo.

B-1.2 Elementos de Documentação Oriundos da Etapa 3

Dependendo do serviço Web a ser descrito, pode ser mais vantajoso que o contexto esteja explícito de forma afirmativa, negativa ou, alternativamente, não estar explícito. Isso pode ser documentado de três formas, respectivamente:

```
<contexto-termo explícito="afirmativa">
```

O descritor xxx, que ocorre múltiplas vezes, contém os nomes das cidades as quais o serviço atende. A propriedade yyy associa o descritor cidades do Iraque ao descritor xxx.

```
</contexto-termo>
```

ou então,

```
<contexto-termo explícito="negativa">
```

O descritor xxx, que ocorre múltiplas vezes, contém os nomes das cidades as quais o serviço não atende. A propriedade zzz associa o descritor cidades do Iraque ao descritor xxx.

```
</contexto-termo>
```

No último caso, nada é informado. Nos dois primeiros casos, deve-se substituir xxx pelo nome do descritor que contém a restrição, e yyy e zzz pelos nomes das propriedades utilizadas para fazer a associação da restrição, afirmativa e negativa, respectivamente, ao termo que é restrito.

Ainda, de acordo com a etapa 3 de nossa proposta, a associação de um descritor da ontologia a um descritor do documento WSDL é documentada como:

```
<associacao-wsdl>
  <descritor-wsdl>
    <cidade>
      Esse descritor corresponde a um descritor de uma operação no
      documento WSDL. A propriedade yyy associa o descritor cidade ao
      descritor xxx. O descritor xxx contém dois atributos: nome da
      operação no documento WSDL e posição onde ocorre o descritor nesta
      operação.
    </cidade>
  </descritor-wsdl>
  <xxx/>
  <yyy/>
</associacao-wsdl>
```

Deve-se substituir xxx pelo nome do descritor da ontologia que equivale ao descritor do documento WSDL, e yyy pelo nome da propriedade utilizada para fazer a associação dos descritores. Cabe ressaltar que esta função não está implementada na ferramenta OntoGuide-4WS, na versão atual.

Também de acordo com a etapa 3, devemos documentar de que forma se dá a descrição do propósito. Uma possível descrição deste utilizando as categorias fundamentais é especificada como:

```
<proposito-operacao tipo="fatorado">
  <processo>
  </processo>
  <objeto-direto>
    <tem-complemento-qualidade>
      <caracteristica/>
    </tem-complemento-qualidade>
  </objeto-direto>
  <objeto-indireto>
    <tem-complemento-qualidade>
      <caracteristica/>
    </tem-complemento-qualidade>
  </objeto-indireto>
</proposito-operacao>
```

ou então,

```
<proposito-operacao tipo="aglutinado">
  <descriptor-proposito/>
</proposito-operacao>
```

Onde o `descriptor-proposito` deve ser substituído pelo descritor escolhido para o propósito, por exemplo, `preverTempoCidadePaís`, por ocasião da descrição do serviço.

Cabe ressaltar que só pode haver um descritor da categoria fundamental de processo por propósito, ao contrário dos descritores derivados das outras categorias.

Por fim, de acordo com a etapa 3 de nossa proposta, a abordagem adotada para a categorização do uso do serviço é especificada como:

```
<categoriaçao-uso-servico tipo="unica">  
</categoriaçao-uso-servico>
```

ou então,

```
<categoriaçao-uso-servico tipo="multipla">  
</categoriaçao-uso-servico>
```

B-2 – Diagramas de Colaboração OntoGuide

Etapa 1 – Configuração do Projeto

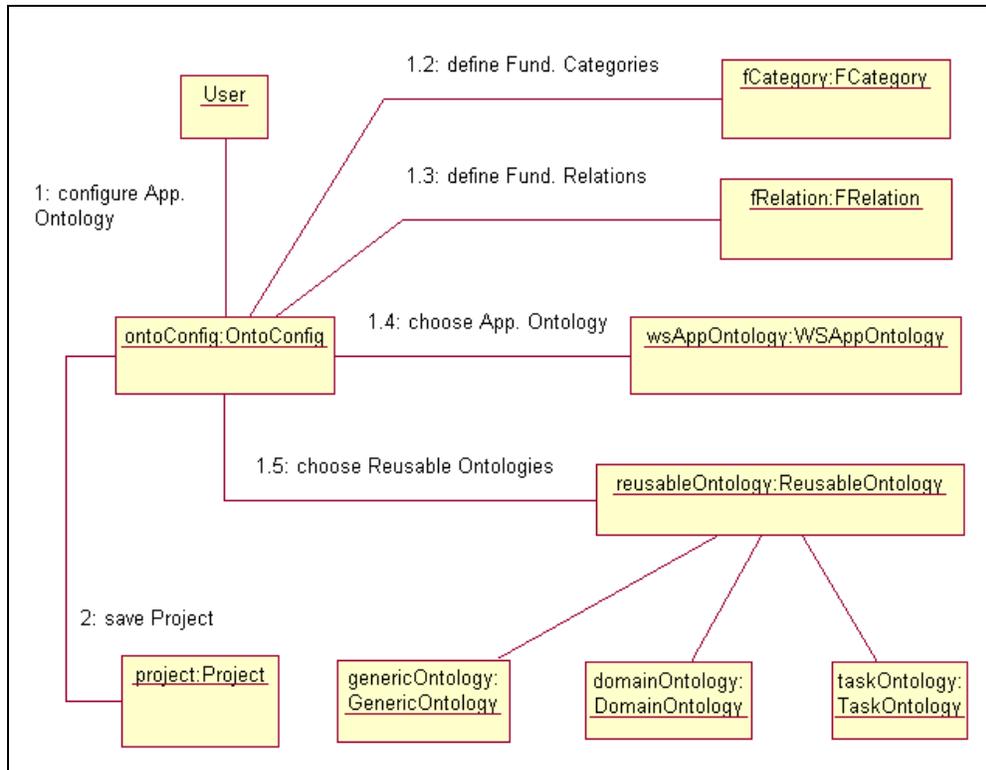


Figura 17 - Diagrama de Colaboração da Etapa 1 de OntoGuide-4WS

Etapa 2 – Tratamento das Ontologias

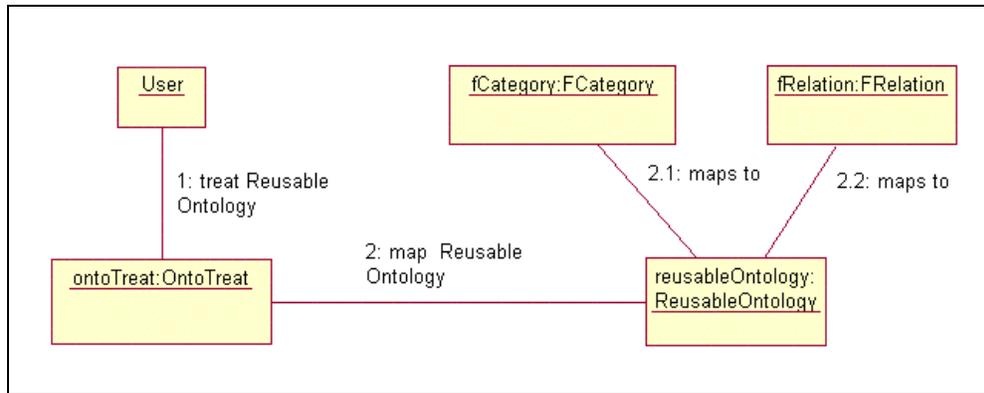


Figura 18 - Diagrama de Colaboração da Etapa 2 de OntoGuide-4WS

Etapa 3 – Extensão da Ontologia de Aplicação

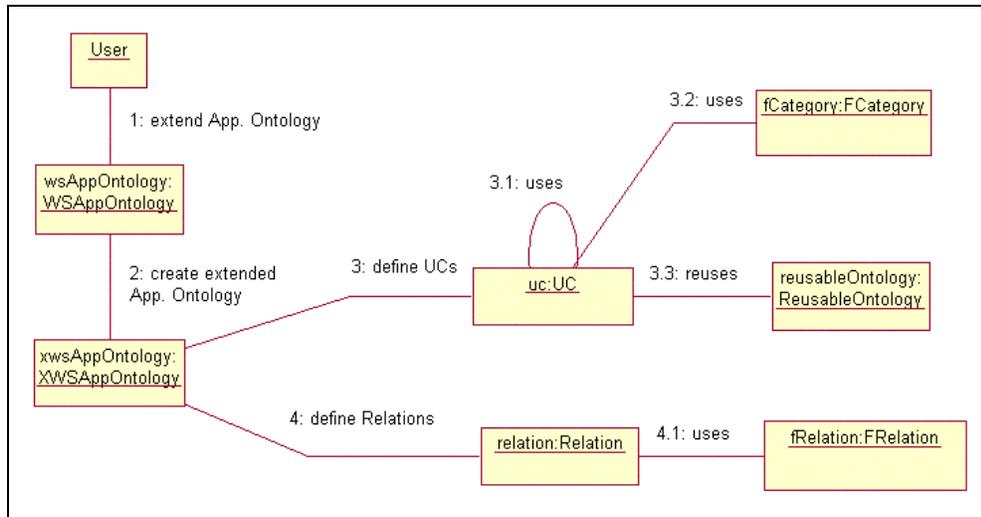


Figura 19 - Diagrama de Colaboração da Etapa 3 de OntoGuide-4WS

B-3 – Fonte da Ontologia de Aplicação em OWL

Esta ontologia contém, a título de exemplo, algumas classes que indicam a necessidade de vocabulários auxiliares. Estas classes têm seu nome precedido por “_WS”. Da mesma forma, para exemplificar sua utilidade, algumas classes foram estendidas com alguns elementos, como é o caso de `GenericTask`, `GenericObject`, `_BioinfoTask`, `_BioinfoObject`.

```

<rdf:RDF xmlns:rss="http://purl.org/rss/1.0/"
  xmlns="http://owl.protege.stanford.edu#"
  xmlns:jms="http://jena.hpl.hp.com/2003/08/jms#"
  xmlns:j.0="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#"
  xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <owl:Ontology rdf:about="">
    <owl:imports
      rdf:resource="http://protege.stanford.edu/plugins/owl/protege" />
  </owl:Ontology>
  <owl:DataRange>
    <owl:oneOf rdf:parseType="Resource">
      <rdf:first>RootedTree</rdf:first>
      <rdf:rest rdf:parseType="Resource">
        <rdf:first>UnrootedTree</rdf:first>
        <rdf:rest>
          <rdf:List rdf:about="http://www.w3.org/1999/02/22-rdf-
            syntax-ns#nil" />
        </rdf:rest>
      </rdf:rest>
    </owl:oneOf>
  </owl:DataRange>
  <owl:DataRange>
    <owl:oneOf rdf:parseType="Resource">
      <rdf:first>input</rdf:first>
      <rdf:rest rdf:parseType="Resource">
        <rdf:first>output</rdf:first>
        <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-
          syntax-ns#nil" />
      </rdf:rest>
    </owl:oneOf>
  </owl:DataRange>
  <owl:Class rdf:ID="BestMatching">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#BioinfoPrecisionMatching" />
    </rdfs:subClassOf>
  </owl:Class>

```

```

: <owl:Class rdf:ID="toSelect">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#GenericTask" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="Milliseconds">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#Qof_Execution_Measure" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="BioinfoDataQuality">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#_BioinfoObjectQualifier" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="TModel">
  <rdfs:comment>TModel como registrado no UDDI</rdfs:comment>
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#ServiceInterface" />
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="GenericSpace" />
: <owl:Class rdf:ID="_BioinfoObject">
  <rdfs:comment>Fixed here just for experimentation
  purposes</rdfs:comment>
</owl:Class>
: <owl:Class rdf:ID="Email">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#GenericResource" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="ProteinStructureDatabase">
  <rdfs:subClassOf rdf:resource="#_BioinfoObject" />
</owl:Class>
: <owl:Class rdf:ID="Qof_Execution_Measure">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#_WS_Qof_Execution_Vocabulary" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="Days">
  <rdfs:subClassOf rdf:resource="#Qof_Execution_Measure" />
</owl:Class>
: <owl:Class rdf:ID="Nanoseconds">
  <rdfs:subClassOf rdf:resource="#Qof_Execution_Measure" />
</owl:Class>
: <owl:Class rdf:ID="OutputNucleotideSequence">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#NucleotideSequence" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="SLA">
  <rdfs:comment>Service Level Agreement. Used to indicate the URL of a
  contract, if the service has one.</rdfs:comment>
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#GenericOperationQualifier" />
  </rdfs:subClassOf>
</owl:Class>

```

```

: <owl:Class rdf:ID="GenBankFormat">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#BioinfoDataFormat" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="GeneralPurposeSequenceDatabase">
  <rdfs:subClassOf rdf:resource="#_BioinfoObject" />
</owl:Class>
: <owl:Class rdf:ID="toDelete">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#GenericTask" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="ConcretImplementation">
  <rdfs:comment>Pointer to WSDL document of the
    service</rdfs:comment>
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#ServiceInterface" />
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="_WS_Time_Zone_Vocabulary" />
: <owl:Class rdf:ID="toCopy">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#GenericTask" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="File">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#GenericResource" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="_WS_Qof_Reputation_Terminology">
  <rdfs:comment>Quality of Reputation Terminology used to describe
    Quality of Reputation attributes.</rdfs:comment>
</owl:Class>
: <owl:Class rdf:ID="ServiceInterface">
  <rdfs:comment>Describes the interface of the service.</rdfs:comment>
</owl:Class>
: <owl:Class rdf:ID="ProteinSequence">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#BiologicalSequence" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="Algorithm">
  <rdfs:subClassOf rdf:resource="#_BioinfoObject" />
</owl:Class>
: <owl:Class rdf:ID="Search">
  : <rdfs:subClassOf>
    <owl:Class rdf:ID="_BioinfoTask" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="toExecute">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#GenericTask" />
  </rdfs:subClassOf>
</owl:Class>

```

```

: <owl:Class rdf:ID="Availability">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#Quality_of_Execution" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="MultipleAlignment">
  <rdfs:comment>Multiple alignment is an extension of pairwise alignment
to incorporate several sequences. Instead of searching a database,
multiple alignment methods take a few sequences and find
common regions between them all.</rdfs:comment>
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#Alignment" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="Idiom">
  : <rdfs:subClassOf>
    : <owl:Restriction>
      : <owl:onProperty>
        <owl:DatatypeProperty rdf:about="#hasValue" />
      </owl:onProperty>
      : <owl:allValuesFrom>
        <owl:Class rdf:ID="_WS_Idiom_Terminology" />
      </owl:allValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#GenericOperationQualifier" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="ProteinSequenceDatabase">
  <rdfs:subClassOf rdf:resource="#GeneralPurposeSequenceDatabase" />
</owl:Class>
: <owl:Class rdf:ID="Monday-Friday">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#Days_of_Week" />
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="_WS_Country_Vocabulary" />
: <owl:Class rdf:ID="ResponseGranularity">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#Quality_of_Information" />
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="_WS_Currency_Terminology" />
: <owl:Class rdf:ID="toSend">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#GenericTask" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="StructuralAlignment">
  <rdfs:comment>Protein structural alignment is a form of alignment
which tries to establish equivalences between two or more protein
structures based on their fold.</rdfs:comment>
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#Alignment" />
  </rdfs:subClassOf>
</owl:Class>

```

```

: <owl:Class rdf:ID="toRetrieve">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#GenericTask" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="toCompare">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#GenericTask" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="toTransform">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#GenericTask" />
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="GenericTime" />
: <owl:Class rdf:ID="xls">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#FileFormat" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="DefaultCost">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#GenericOperationQualifier" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="toCalculate">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#GenericTask" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="CuratedData">
  <rdfs:subClassOf rdf:resource="#BioinfoDataQuality" />
</owl:Class>
: <owl:Class rdf:ID="OtherService">
  <rdfs:comment>Services that are not Web services. E.g.
  Scripts</rdfs:comment>
  : <rdfs:subClassOf>
    <owl:Class rdf:ID="Service" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="toInsert">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#GenericTask" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="PhylogeneticTree">
  <rdfs:subClassOf rdf:resource="#_BioinfoObject" />
</owl:Class>
: <owl:Class rdf:ID="toLogin">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#GenericTask" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="NucleotideSequenceDatabase">
  <rdfs:subClassOf rdf:resource="#GeneralPurposeSequenceDatabase" />
</owl:Class>

```

```

: <owl:Class rdf:ID="GlobalAlignment">
  <rdfs:comment>A global alignment between two sequences is an
    alignment in which all of the characters in both sequences
    participate in the alignment. Global alignments are useful mostly
    for finding closely-related sequences.</rdfs:comment>
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#PairwiseAlignment" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="Days_of_Week">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#_WS_Qof_Execution_Vocabulary" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="BioinfoDataFormat">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#_BioinfoObjectQualifier" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="Quality_of_Reputation">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#GenericOperationQualifier" />
  </rdfs:subClassOf>
  : <rdfs:subClassOf>
    : <owl:Restriction>
      : <owl:onProperty>
        : <owl:ObjectProperty rdf:about="#hasMetrics" />
      </owl:onProperty>
      <owl:allValuesFrom
        rdf:resource="#_WS_Qof_Reputation_Terminology" />
    </owl:Restriction>
  </rdfs:subClassOf>
  : <rdfs:subClassOf>
    : <owl:Restriction>
      : <owl:onProperty>
        : <owl:ObjectProperty rdf:about="#hasMeasure" />
      </owl:onProperty>
      <owl:allValuesFrom
        rdf:resource="#_WS_Qof_Reputation_Terminology" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="_WS_Region_Vocabulary" />
: <owl:Class rdf:ID="LocalAlignment">
  <rdfs:comment>Local alignment methods find related regions within
    sequences - in other words they can consist of a subset of the
    characters within each sequence (e.g. positions 20-40 of sequence
    A might align with positions 50-70 of sequence B).</rdfs:comment>
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#PairwiseAlignment" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="AvailabilityDays">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#Quality_of_Execution" />
  </rdfs:subClassOf>
</owl:Class>

```

```

: <owl:Class rdf:ID="ResponseTime">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#Quality_of_Execution" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="SWISSPROT">
  <rdfs:subClassOf rdf:resource="#ProteinSequenceDatabase" />
</owl:Class>
: <owl:Class rdf:ID="FileFormat">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#GenericObjectQualifier" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="Sunday">
  <rdfs:subClassOf rdf:resource="#Days_of_Week" />
</owl:Class>
: <owl:Class rdf:ID="OutputProteinSequence">
  <rdfs:subClassOf rdf:resource="#ProteinSequence" />
</owl:Class>
: <owl:Class rdf:ID="GenericTask">
  <rdfs:comment>Main verb used to indicate the web services purpose
    and also its abstract message purpose. Process (action) related
    aspects, according to Ranganathans Energy fundamental category.
    It represents a way to distinguish one service from the others
    according to its purpose. It is complemented by the
    Object.</rdfs:comment>
</owl:Class>
: <owl:Class rdf:ID="GenBank">
  <rdfs:subClassOf rdf:resource="#NucleotideSequenceDatabase" />
</owl:Class>
: <owl:Class rdf:ID="WebService">
  <rdfs:comment>Web Service to be described. It is the main class of this
    ontology.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#Service" />
</owl:Class>
: <owl:Class rdf:ID="csv">
  <rdfs:subClassOf rdf:resource="#FileFormat" />
</owl:Class>
: <owl:Class rdf:ID="toUpdate">
  <rdfs:subClassOf rdf:resource="#GenericTask" />
</owl:Class>
: <owl:Class rdf:ID="_BioinfoObjectQualifier">
  <rdfs:comment>Fixed here for experimentation purposes</rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="_WS_Service_Category_Terminology" />
: <owl:Class rdf:ID="GenericObject">
  <rdfs:comment>Entities and other aspects not classified in the
    remaining fundamental categories, according to Ranganathans
    Personality fundamental category. Should be
    extended</rdfs:comment>
</owl:Class>
: <owl:Class rdf:ID="_WS_Qof_Information_Vocabulary">
  <rdfs:comment>Quality of Information Vocabulary used to describe
    Quality of Information attributes.</rdfs:comment>
</owl:Class>

```

```

: <owl:Class rdf:ID="ProviderName">
  <rdfs:comment>Company responsible for providing and maintaining a
    Web service or a resource. It may be considered as an object, but
    as they fulfill a particularly important role concerning objects, we
    decided to model it apart from them.</rdfs:comment>
  : <rdfs:subClassOf>
    <owl:Class rdf:ID="_WS_Company_Vocabulary" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="Elaboration">
  <rdfs:subClassOf rdf:resource="#_BioinfoTask" />
</owl:Class>
: <owl:Class rdf:ID="AbstractMessage">
  <rdfs:comment>A set of parameters, used to communicate with the Web
    service. The abstract message can be used as a input, output or
    exception message. Parameters are part of a
    message.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#ServiceInterface" />
</owl:Class>
: <owl:Class rdf:ID="BioinfoPrecisionMatching">
  <rdfs:subClassOf rdf:resource="#_BioinfoObjectQualifier" />
</owl:Class>
: <owl:Class rdf:ID="BlastAlgorithm">
  <rdfs:subClassOf rdf:resource="#Algorithm" />
</owl:Class>
: <owl:Class rdf:ID="NucleotideSequence">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#BiologicalSequence" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="Reliability">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#Quality_of_Information" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="PairwiseAlignment">
  <rdfs:comment>Pairwise sequence alignment methods are concerned
    with finding the best-matching piecewise (local) or global
    alignments of protein (amino acid) or DNA (nucleic acid)
    sequences.</rdfs:comment>
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#Alignment" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="Security">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#Quality_of_Execution" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="XMLFormat">
  <rdfs:subClassOf rdf:resource="#BioinfoDataFormat" />
</owl:Class>
: <owl:Class rdf:ID="FastaFormat">
  <rdfs:subClassOf rdf:resource="#BioinfoDataFormat" />
</owl:Class>

```

```

: <owl:Class rdf:ID="Quality_of_Information">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#GenericOperationQualifier" />
  </rdfs:subClassOf>
  : <rdfs:subClassOf>
    : <owl:Restriction>
      : <owl:onProperty>
        <owl:ObjectProperty rdf:about="#hasMetrics" />
      </owl:onProperty>
      <owl:allValuesFrom
        rdf:resource="#_WS_Qof_Information_Vocabulary" />
    </owl:Restriction>
  </rdfs:subClassOf>
  : <rdfs:subClassOf>
    : <owl:Restriction>
      : <owl:onProperty>
        <owl:ObjectProperty rdf:about="#hasMeasure" />
      </owl:onProperty>
      <owl:allValuesFrom
        rdf:resource="#_WS_Qof_Information_Vocabulary" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="MaximumParcimonyAlgorithm">
  <rdfs:subClassOf rdf:resource="#Algorithm" />
</owl:Class>
: <owl:Class rdf:ID="toProvide">
  <rdfs:subClassOf rdf:resource="#GenericTask" />
</owl:Class>
: <owl:Class rdf:ID="Privacy">
  <rdfs:subClassOf rdf:resource="#Quality_of_Information" />
</owl:Class>
: <owl:Class rdf:ID="_WS_Qof_Execution_Vocabulary">
  <rdfs:comment>Quality of Execution Vocabulary used to describe
    Quality of Execution attributes.</rdfs:comment>
</owl:Class>
: <owl:Class rdf:ID="Abrangency">
  <rdfs:subClassOf rdf:resource="#GenericSpace" />
</owl:Class>
: <owl:Class rdf:ID="Alignment">
  <rdfs:comment>Sequence alignment is concerned with the relationships
    between biological sequences (e.g. protein sequences or DNA
    sequences). Two major types exist: pairwise and multiple sequence
    alignments.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#_BioinfoTask" />
</owl:Class>
: <owl:Class rdf:ID="toMove">
  <rdfs:subClassOf rdf:resource="#GenericTask" />
</owl:Class>
: <owl:Class rdf:ID="txt">
  <rdfs:subClassOf rdf:resource="#FileFormat" />
</owl:Class>
: <owl:Class rdf:ID="Seconds">
  <rdfs:subClassOf rdf:resource="#Qof_Execution_Measure" />
</owl:Class>
: <owl:Class rdf:ID="jpeg">
  <rdfs:subClassOf rdf:resource="#FileFormat" />
</owl:Class>

```

```

: <owl:Class rdf:ID="TextFormat">
  <rdfs:subClassOf rdf:resource="#BioinfoDataFormat" />
</owl:Class>
: <owl:Class rdf:ID="GenericOperationQualifier">
  <rdfs:comment>Describes generic quality attributes of services
    operations. It represents a way to distinguish one operation from
    the others according to its qualities. They should be extended
    according to specific needs.</rdfs:comment>
  </owl:Class>
  <owl:Class rdf:ID="_WS_Application_Area_Terminology" />
: <owl:Class rdf:ID="InputNucleotideSequence">
  <rdfs:subClassOf rdf:resource="#NucleotideSequence" />
</owl:Class>
: <owl:Class rdf:ID="BiologicalSequence">
  <rdfs:subClassOf rdf:resource="#_BioinfoObject" />
</owl:Class>
: <owl:Class rdf:ID="Priority">
  : <rdfs:subClassOf>
    <owl:Class rdf:about="#Quality_of_Execution" />
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="Saturday">
  <rdfs:subClassOf rdf:resource="#Days_of_Week" />
</owl:Class>
: <owl:Class rdf:ID="GenericObjectQualifier">
  <rdfs:comment>Quality of objects</rdfs:comment>
</owl:Class>
: <owl:Class rdf:ID="PreliminarData">
  <rdfs:subClassOf rdf:resource="#BioinfoDataQuality" />
</owl:Class>
: <owl:Class rdf:ID="toCombine">
  <rdfs:subClassOf rdf:resource="#GenericTask" />
</owl:Class>
: <owl:Class rdf:ID="GenericResource">
  <rdfs:subClassOf rdf:resource="#GenericObject" />
</owl:Class>
: <owl:Class rdf:ID="Database">
  <rdfs:subClassOf rdf:resource="#GenericResource" />
</owl:Class>
: <owl:Class rdf:ID="toValidate">
  <rdfs:subClassOf rdf:resource="#GenericTask" />
</owl:Class>
: <owl:Class rdf:ID="Qof_Execution_Metrics">
  <rdfs:subClassOf rdf:resource="#_WS_Qof_Execution_Vocabulary" />
</owl:Class>
: <owl:Class rdf:ID="doc">
  <rdfs:subClassOf rdf:resource="#FileFormat" />
</owl:Class>
: <owl:Class rdf:ID="GMT">
  <rdfs:subClassOf rdf:resource="#_WS_Time_Zone_Vocabulary" />
</owl:Class>
: <owl:Class rdf:ID="DateInformation">
  <rdfs:comment>Web services related dates.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#GenericTime" />
</owl:Class>
  <owl:Class rdf:ID="_WS_Payment_Type_Terminology" />

```

```

: <owl:Class rdf:ID="Quality_of_Execution">
  <rdfs:subClassOf rdf:resource="#GenericOperationQualifier" />
  : <rdfs:subClassOf>
    : <owl:Restriction>
      : <owl:onProperty>
        <owl:ObjectProperty rdf:about="#hasMetrics" />
      </owl:onProperty>
      <owl:allValuesFrom
        rdf:resource="#_WS_Qof_Execution_Vocabulary" />
    </owl:Restriction>
  </rdfs:subClassOf>
  : <rdfs:subClassOf>
    : <owl:Restriction>
      : <owl:onProperty>
        <owl:ObjectProperty rdf:about="#hasMeasure" />
      </owl:onProperty>
      <owl:allValuesFrom
        rdf:resource="#_WS_Qof_Execution_Vocabulary" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
: <owl:Class rdf:ID="InputProteinSequence">
  <rdfs:subClassOf rdf:resource="#ProteinSequence" />
</owl:Class>
<owl:Class rdf:ID="_WS_License_of_Use_Terminology" />
: <owl:Class rdf:ID="gif">
  <rdfs:subClassOf rdf:resource="#FileFormat" />
</owl:Class>
: <owl:ObjectProperty rdf:ID="hasIndirectObject">
  <rdfs:range rdf:resource="#_BioinfoObject" />
  <rdfs:domain rdf:resource="#AbstractMessage" />
</owl:ObjectProperty>
: <owl:ObjectProperty rdf:ID="hasObjectQualifier">
  <rdfs:domain rdf:resource="#_BioinfoObject" />
  <rdfs:range rdf:resource="#_BioinfoObjectQualifier" />
</owl:ObjectProperty>
: <owl:ObjectProperty rdf:ID="hasCurrency">
  <rdf:type
    rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty" />
  <rdfs:comment>Name of the currency used. ISO 4217 may be
    used.</rdfs:comment>
  <rdfs:range rdf:resource="#_WS_Currency_Terminology" />
  <rdfs:domain rdf:resource="#DefaultCost" />
</owl:ObjectProperty>
: <owl:ObjectProperty rdf:ID="hasTimeZone">
  <rdf:type
    rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty" />
  <rdfs:domain rdf:resource="#DateInformation" />
  <rdfs:range rdf:resource="#_WS_Time_Zone_Vocabulary" />
</owl:ObjectProperty>

```

```

<owl:ObjectProperty rdf:ID="hasSpaceScope">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#AbstractMessage" />
        <owl:Class rdf:about="#WebService" />
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <rdfs:range rdf:resource="#GenericSpace" />
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAbragencyRegion">
  <rdfs:comment>Regions of the country for which the Web service can be provided, e.g., in a pizza delivery service, the Abragency_Region indicates the region of the country in which the pizza delivery can be done. It may refer to a state, a city, a zip code or a set of one of those. Regions may be defined by a terminology.</rdfs:comment>
  <rdfs:range rdf:resource="#_WS_Region_Vocabulary" />
  <rdfs:domain rdf:resource="#Abragency" />
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasTask">
  <rdfs:comment>Task is used to define more accurately the purpose of a task.</rdfs:comment>
  <rdfs:domain rdf:resource="#AbstractMessage" />
  <rdfs:range rdf:resource="#_BioinfoTask" />
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isAppliedToObject">
  <rdfs:domain rdf:resource="#AvailabilityDays" />
  <rdfs:range rdf:resource="#Days_of_Week" />
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isNotAppliedToObject" />
<owl:ObjectProperty rdf:ID="hasPosCondition">
  <rdfs:domain rdf:resource="#WebService" />
  <rdfs:range rdf:resource="#GenericTask" />
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasServiceCategory">
  <rdf:type
    rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty" />
  <rdfs:comment>Category of the web service. Standards like NAICS may be used here.</rdfs:comment>
  <rdfs:domain rdf:resource="#WebService" />
  <rdfs:range rdf:resource="#_WS_Service_Category_Terminology" />
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasMeasure">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Quality_of_Information" />
        <owl:Class rdf:about="#Quality_of_Execution" />
        <owl:Class rdf:about="#Quality_of_Reputation" />
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
</owl:ObjectProperty>

```

```

: <owl:ObjectProperty rdf:ID="hasParameter">
  <rdfs:range rdf:resource="#_BioinfoObject" />
  <rdfs:domain rdf:resource="#AbstractMessage" />
</owl:ObjectProperty>
: <owl:ObjectProperty rdf:ID="hasDirectObject">
  <rdfs:range rdf:resource="#_BioinfoObject" />
  <rdfs:domain rdf:resource="#AbstractMessage" />
</owl:ObjectProperty>
: <owl:ObjectProperty rdf:ID="hasLicenseOfUse">
  <rdfs:comment>Type of license. E.g. per use, by month, by number of
    connections.</rdfs:comment>
  <rdfs:type
    rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty" />
  <rdfs:domain rdf:resource="#DefaultCost" />
  <rdfs:range rdf:resource="#_WS_License_of_Use_Terminology" />
</owl:ObjectProperty>
: <owl:ObjectProperty rdf:ID="hasTModel">
  <rdfs:type
    rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty" />
  <rdfs:domain rdf:resource="#WebService" />
  <rdfs:range rdf:resource="#TModel" />
</owl:ObjectProperty>
: <owl:ObjectProperty rdf:ID="hasQualifierAspect">
  <rdfs:domain rdf:resource="#AbstractMessage" />
  <rdfs:range rdf:resource="#GenericOperationQualifier" />
</owl:ObjectProperty>
: <owl:ObjectProperty rdf:ID="usesObject">
  <rdfs:range rdf:resource="#_BioinfoObject" />
  <rdfs:domain rdf:resource="#AbstractMessage" />
</owl:ObjectProperty>
: <owl:ObjectProperty rdf:ID="hasPreCondition">
  <rdfs:domain rdf:resource="#WebService" />
  <rdfs:range rdf:resource="#GenericTask" />
</owl:ObjectProperty>
: <owl:ObjectProperty rdf:ID="hasProviderInformation">
  <rdfs:comment>Provider information about a service or
    resource.</rdfs:comment>
  <rdfs:range rdf:resource="#_WS_Company_Vocabulary" />
</owl:ObjectProperty>
: <owl:ObjectProperty rdf:ID="hasTimeScope">
  <rdfs:domain>
    : <owl:Class>
      : <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#AbstractMessage" />
        <owl:Class rdf:about="#WebService" />
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <rdfs:range rdf:resource="#GenericTime" />
</owl:ObjectProperty>
: <owl:ObjectProperty rdf:ID="hasAbstractMessage">
  <rdfs:domain rdf:resource="#WebService" />
  <rdfs:range rdf:resource="#AbstractMessage" />
</owl:ObjectProperty>
: <owl:ObjectProperty rdf:ID="producesObject">
  <rdfs:range rdf:resource="#_BioinfoObject" />
  <rdfs:domain rdf:resource="#AbstractMessage" />
</owl:ObjectProperty>

```

```

<owl:ObjectProperty rdf:ID="hasMetrics">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Quality_of_Information" />
        <owl:Class rdf:about="#Quality_of_Execution" />
        <owl:Class rdf:about="#Quality_of_Reputation" />
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAbrangencyCountry">
  <rdfs:comment>Countries for which the Web service can be provided,
  e.g., in a pizza delivery service, the Abrangency_Country indicates
  the country in which the pizza delivery can be done. ISO Standard
  is used to describe the country codes.</rdfs:comment>
  <rdfs:range rdf:resource="#_WS_Country_Vocabulary" />
  <rdfs:domain rdf:resource="#Abrangency" />
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="hasMessageType">
  <rdfs:range>
    <owl:DataRange>
      <owl:oneOf rdf:parseType="Resource">
        <rdfs:rest rdf:parseType="Resource">
          <rdf:first>output</rdf:first>
          <rdfs:rest rdf:parseType="Resource">
            <rdf:rest
              rdf:resource="http://www.w3.org/1999/02/22
              -rdf-syntax-ns#nil" />
            <rdf:first>fault</rdf:first>
          </rdf:rest>
        </rdf:rest>
        <rdf:first>input</rdf:first>
      </owl:oneOf>
    </owl:DataRange>
  </rdfs:range>
  <rdfs:type
    rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty" />
  <rdfs:comment>Type of message: input, output, fault.</rdfs:comment>
  <rdfs:domain rdf:resource="#AbstractMessage" />
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasID">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
  <rdfs:domain rdf:resource="#TModel" />
</owl:DatatypeProperty>

```

```

<owl:DatatypeProperty rdf:ID="hasValue">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
  <rdf:type
    rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty" />
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Idiom" />
        <owl:Class rdf:about="#Quality_of_Reputation" />
        <owl:Class rdf:about="#ResponseTime" />
        <owl:Class rdf:about="#Security" />
        <owl:Class rdf:about="#Priority" />
        <owl:Class rdf:about="#Availability" />
        <owl:Class rdf:about="#Quality_of_Information" />
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
</owl:DatatypeProperty>
<owl:FunctionalProperty rdf:ID="hasURI">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
  <rdfs:comment>URL of the service provider, where additional
    information about it may be provided.</rdfs:comment>
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#ProviderName" />
        <owl:Class rdf:about="#SLA" />
        <owl:Class rdf:about="#ConcretImplementation" />
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="hasCreationDate">
  <rdfs:comment>Web Services date of creation.</rdfs:comment>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
  <rdfs:domain rdf:resource="#DateInformation" />
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="hasType">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
  <rdfs:comment>Pode ser, input, output ou fault</rdfs:comment>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="hasVersionDate">
  <rdfs:comment>Web services date of current version.</rdfs:comment>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
  <rdfs:domain rdf:resource="#DateInformation" />
</owl:FunctionalProperty>

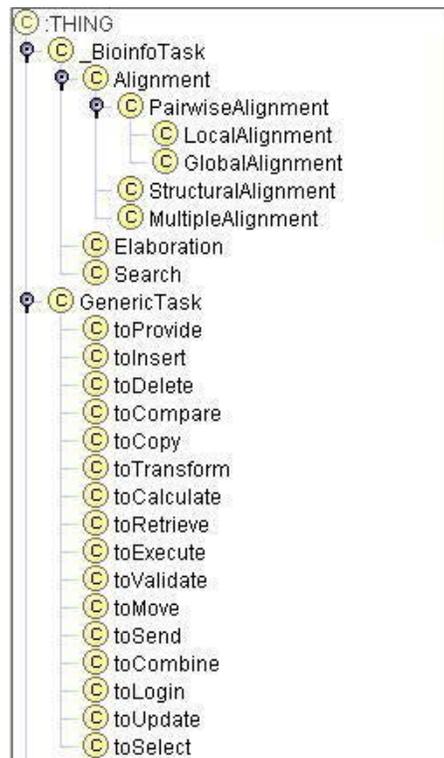
```

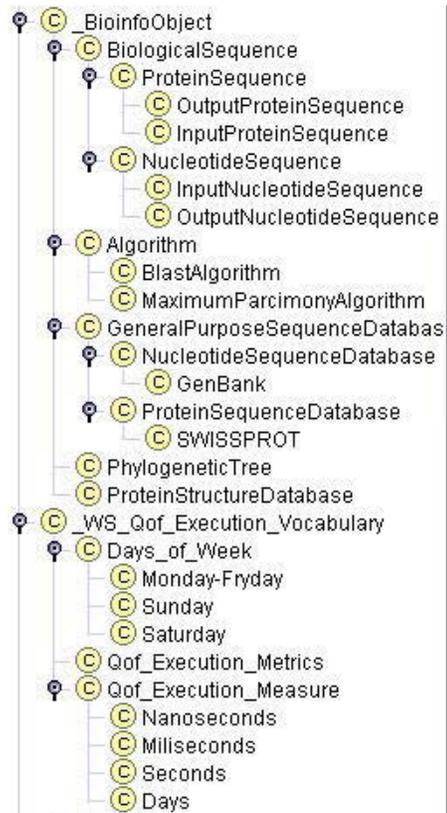
```

- <owl:FunctionalProperty rdf:ID="hasApplicationArea">
  <rdfs:comment>Code of the Web services application area. Used to
  provide a more specific category to classify the
  service.</rdfs:comment>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"
  />
  <rdfs:range rdf:resource="#_WS_Application_Area_Terminology" />
  <rdfs:domain rdf:resource="#WebService" />
</owl:FunctionalProperty>
- <owl:FunctionalProperty rdf:ID="hasCostValue">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"
  />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float" />
  <rdfs:comment>Default monetary value to be paid when using the
  service, when there is no contract associated.</rdfs:comment>
  <rdfs:domain rdf:resource="#DefaultCost" />
</owl:FunctionalProperty>
- <owl:FunctionalProperty rdf:ID="hasPaymentType">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"
  />
  <rdfs:comment>Type of payment. E.g. by credit card,
  cash.</rdfs:comment>
  <rdfs:range rdf:resource="#_WS_Payment_Type_Terminology" />
  <rdfs:domain rdf:resource="#DefaultCost" />
</owl:FunctionalProperty>
<LocalAlignment rdf:ID="WS_LocalAlignment_Task" />
</rdf:RDF>

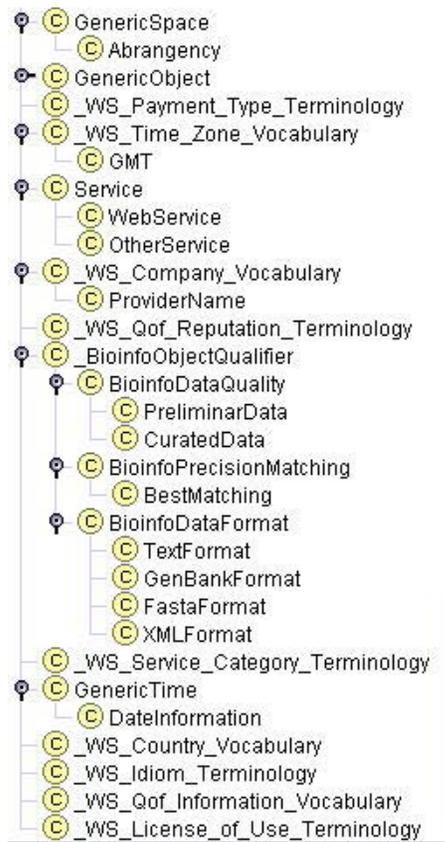
```

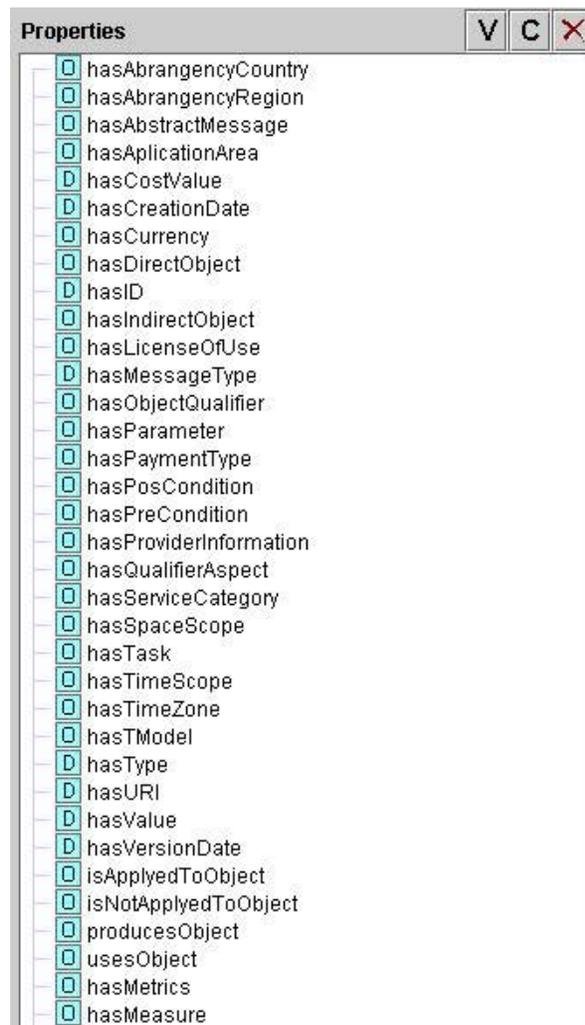
B-3 – Ontologia de Aplicação na Ferramenta Protégé











| P Properties at Class WebService | | | | |
|--|----------|-------------|--|--|
| Name | Type | Cardinality | Other Facets | |
| <input type="checkbox"/> hasPreCondition | Instance | multiple | classes={GenericTask} | |
| <input type="checkbox"/> hasPosCondition | Instance | multiple | classes={GenericTask} | |
| <input type="checkbox"/> hasAbstractMessage | Instance | multiple | classes={AbstractMessage} | |
| <input type="checkbox"/> hasSpaceScope | Instance | multiple | classes={GenericSpace} | |
| <input type="checkbox"/> hasServiceCategory | Instance | single | classes={_WS_Service_Category_Terminology} | |
| <input type="checkbox"/> hasApplicationArea | Instance | single | classes={_WS_Application_Area_Terminology} | |
| <input type="checkbox"/> hasConcreteImplementation | Instance | single | classes={ConcreteImplementation} | |
| <input type="checkbox"/> hasTimeScope | Instance | multiple | classes={GenericTime} | |
| <input type="checkbox"/> hasTModel | Instance | single | classes={TModel} | |
| <input type="checkbox"/> hasProviderInformation | Instance | multiple | classes={_WS_Company_Vocabulary} | |

| P Properties at Class AbstractMessage | | | | |
|--|----------|-------------|-------------------------------------|--|
| Name | Type | Cardinality | Other Facets | |
| <input checked="" type="checkbox"/> hasMessageType | Symbol | single | allowed-values={input,output,fault} | |
| <input type="checkbox"/> usesObject | Instance | multiple | classes={_BioinfoObject} | |
| <input type="checkbox"/> producesObject | Instance | multiple | classes={_BioinfoObject} | |
| <input type="checkbox"/> hasSpaceScope | Instance | multiple | classes={GenericSpace} | |
| <input type="checkbox"/> hasIndirectObject | Instance | multiple | classes={_BioinfoObject} | |
| <input type="checkbox"/> hasQualifierAspect | Instance | multiple | classes={GenericOperationQualifier} | |
| <input type="checkbox"/> hasParameter | Instance | multiple | classes={_BioinfoObject} | |
| <input type="checkbox"/> hasDirectObject | Instance | multiple | classes={_BioinfoObject} | |
| <input type="checkbox"/> hasTimeScope | Instance | multiple | classes={GenericTime} | |
| <input type="checkbox"/> hasTask | Instance | single | classes={_BioinfoTask} | |

Figura 20 – Ontologia de Aplicação de Serviços Web – Telas na Ferramenta Protégé